Towards diversified recommendations

Lu Gan

To cite this version:

Lu Gan. Towards diversified recommendations. Information Retrieval [cs.IR]. Université de Lyon, 2022. English. NNT : 2022LYSEI047 . tel-03814902

HAL Id: tel-03814902
https://theses.hal.science/tel-03814902

Submitted on 14 Oct 2022

THESE DE DOCTORAT DE L'UNIVERSITE DE LYON

opérée au sein de

l'INSA de Lyon

Ecole Doctorale **N° 512**

Mathématiques et Informatique (InfoMaths)

Spécialité/discipline de doctorat : INFORMATIQUE

2022LYSEI047

Towards Diversified Recommendations

Soutenue publiquement le 30/05/2022, par :

Lu Gan

Devant le jury composé de :

Anne Boyer	Professeur des universités, Université de Lorraine	Rapporteur
Chantal Soulé-Dupuy	Professeur des universités, Université Toulouse 1 Capitole	Rapporteur
Patrice Bellot	Professeur des universités, Aix-Marseille Université	Examinateur
Jaap Kamps	Associate Professor, University of Amsterdam	Examinateur
Sylvie Calabretto	Professeur des universités, INSA Lyon	Directrice de thèse
Diana Nurbakova	Maître de conférences, INSA-Lyon	Co-directrice de thèse
Léa Laporte	Data Scientist, Cdiscount	Invitée

Département FEDORA – INSA Lyon - Ecoles Doctorales – Quinquennal 2016-2020

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	CHIMIE DE LYON http://www.edchimie-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr INSA : R. GOURDON	M. Stéphane DANIELE Institut de recherches sur la catalyse et l'environnement de Lyon IRCELYON-UMR 5256 Équipe CDFA 2 Avenue Albert EINSTEIN 69 626 Villeurbanne CEDEX directeur@edchimie-lyon.fr
E.E.A.	ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE http://edeea.ec-lyon.fr Sec. : M.C. HAVGOUDOUKIAN ecole-doctorale.eea@ec-lyon.fr	M. Gérard SCORLETTI École Centrale de Lyon 36 Avenue Guy DE COLLONGUE 69 134 Écully Tél : 04.72.18.60.97 Fax 04.78.43.37.17 gerard.scorletti@ec-lyon.fr
E2M2	ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION http://e2m2.universite-lyon.fr Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 INSA : H. CHARLES secretariat.e2m2@univ-lyon1.fr	M. Philippe NORMAND UMR 5557 Lab. d'Ecologie Microbienne Université Claude Bernard Lyon 1 Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69 622 Villeurbanne CEDEX philippe.normand@univ-lyon1.fr
EDISS	INTERDISCIPLINAIRE SCIENCES-SANTÉ http://www.ediss-lyon.fr Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 INSA : M. LAGARDE secretariat.ediss@univ-lyon1.fr	Mme Emmanuelle CANET-SOULAS INSERM U1060, CarMeN lab, Univ. Lyon 1 Bâtiment IMBL 11 Avenue Jean CAPELLE INSA de Lyon 69 621 Villeurbanne Tél : 04.72.68.49.09 Fax : 04.72.68.49.16 emmanuelle.canet@univ-lyon1.fr
INFOMATHS	INFORMATIQUE ET MATHÉMATIQUES http://edinfomaths.universite-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	M. Luca ZAMBONI Bât. Braconnier 43 Boulevard du 11 novembre 1918 69 622 Villeurbanne CEDEX Tél : 04.26.23.45.52 zamboni@maths.univ-lyon1.fr
Matériaux	MATÉRIAUX DE LYON http://ed34.universite-lyon.fr Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bât. Direction ed.materiaux@insa-lyon.fr	M. Jean-Yves BUFFIÈRE INSA de Lyon MATEIS - Bât. Saint-Exupéry 7 Avenue Jean CAPELLE 69 621 Villeurbanne CEDEX Tél : 04.72.43.71.70 Fax : 04.72.43.85.28 jean-yves.buffiere@insa-lyon.fr
MEGA	MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE http://edmega.universite-lyon.fr Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bât. Direction mega@insa-lyon.fr	M. Jocelyn BONJOUR INSA de Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69 621 Villeurbanne CEDEX jocelyn.bonjour@insa-lyon.fr
ScSo	ScSo* http://ed483.univ-lyon2.fr Sec. : Véronique GUICHARD INSA : J.Y. TOUSSAINT Tél : 04.78.69.72.76 veronique.cervantes@univ-lyon2.fr	M. Christian MONTES Université Lyon 2 86 Rue Pasteur 69 365 Lyon CEDEX 07 christian.montes@univ-lyon2.fr

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Abstract

Recommender systems (RS) have been widely applied in real life scenarios to constantly provide personalised recommendation to satisfy users' need. In classical top-N recommendation models, historical user-item interactions are collected and exploited to learn and predict for each user a top-N item list. But we are also aware that both user-side and item-side auxiliary information may help to improve the recommendation performance. At the same time, in terms of recommendation performance, accuracy has been the main research focus in recommender systems, though works have pointed out that an optimal accuracy is not equal to an optimal satisfaction of users towards recommendation. An accuracy-centric recommendation model may create an isolate, singular and redundant atmosphere when providing the service, thus it is essential to bring other goals in RS to alleviate these problems.

In this dissertation, we focus on bringing diversity along with accuracy as recommendation goals, as we argue that a diversified recommendation helps alleviate the problems suffered by accuracy-centric RS. We also take item-side auxiliary information into account for enhancing accuracy. Thus we propose diversity-aware top-N recommendations based on knowledge graph embedding to aim at achieving both high accuracy and high diversity in recommendation lists for users.

Our first contribution is DivKG, a diversified recommendation framework that combines knowledge graph embedding and determinantal point processes (DPP). We propose a new personalised DPP kernel matrix construction method that uses knowledge graph embedding results for DPP diversification.

Our second contribution is EMDKG, a diversified recommendation framework which encodes semantic diversity into item representations and achieve better trade-off compared to state-of-the art methods in terms of accuracy and diversity.

Keywords: Recommender Systems; Diversity; Accuracy; Knowledge Graph Embedding; Determinantal Point Processes.

Résumé

Les systèmes de recommandation (RS) sont largement appliqués dans la vie réelle pour fournir des recommandations personnalisées pour satisfaire les besoins des utilisateurs. Dans les modèles de recommandation top-N classiques, les historiques d'interactions entre utilisateur et éléments sont recueillies et exploitées pour apprendre et prédire des listes de top-N éléments. Mais nous sommes également conscients que les informations auxiliaires côté utilisateur et côté élément peuvent contribuer à améliorer la qualité des recommandations. En termes de mesures de performance de recommandation, la précision a été le principal objectif dans le domaine, bien que des travaux aient souligné qu'une précision optimale n'égale pas une satisfaction optimale des utilisateurs vis-à-vis de la recommandation. Un modèle de recommandation centré sur la précision peut renvoyer des résultats redondants lors de la fourniture du service. Il est donc essentiel d'apporter d'autres objectifs dans la recommandation pour atténuer ces problèmes.

Dans cette thèse, nous nous concentrons sur l'étude conjointe de la diversité et de la précision en tant qu'objectifs de recommandation, car nous considérons qu'une recommandation diversifiée aide à atténuer les problèmes générés par la recommandation centrée sur la précision. Nous prenons également en compte les informations auxiliaires côté élément pour améliorer la précision. Ainsi, nous proposons des recommandations diversifiées top-N basées sur le plongement de graphes de connaissances pour atteindre à la fois une haute précision et une grande diversité dans les listes de recommandations.

Notre première contribution est **DivKG**, un modèle qui combine le plongement de graphes de connaissances (KGE) et les processus ponctuels déterminantaux (DPP). Nous proposons une nouvelle méthode de construction de matrice de noyau DPP personnalisée qui utilise des résultats de KGE pour la diversification DPP.

Notre deuxième contribution est **EMDKG**, un modèle qui encode la diversité sémantique dans les représentations d'éléments et réalise un meilleur compromis par rapport aux méthodes de l'état de l'art en termes de précision et de diversité.

Mots-clés: Systèmes de recommandation; Diversité; Précision; Plongement des Graphes de Connaissance; Processus Ponctuels Déterminants.

Declaration by the candidate

I hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Where other sources of information have been used, they have been marked.

The work has not been presented in the same or a similar form to any other testing authority and has not been made public.

I hereby also entitle a right of use (free of charge, not limited locally and for an indefinite period of time) that my thesis can be duplicated, saved and archived by INSA-Lyon, LIRIS laboratory and Infomaths doctoral school.

Lyon, May 18, 2022

Remerciements

Ce manuscrit de thèse est le fruit d'un long travail de recherche qui a été riche en enseignements. Durant ces années, j'ai beaucoup appris tant du point de vue professionnel que personnel, plus que je n'aurais imaginé avant de commencer cette thèse. Pour tout cela, je souhaite remercier les personnes qui m'ont entouré durant cette période.

Je tiens tout d'abord à particulièrement remercier ma directrice Sylvie Calabretto et mon encadrante Diana Nurbakova avec qui j'ai eu la chance de pouvoir travailler durant toute cette thèse. Sylvie m'a apporté son expérience et son recul dans le domaine de la recherche et Diana m'a guidé et soutenu dans chacune des étapes de ce travail. Je remercie aussi à Dr. Léa Laporte qui m'a accompagné durant la thèse. J'ai beaucoup apprécié travailler avec vous et je vous remercie encore une fois ici pour vos conseils avisés, votre patience, votre bonne humeur et pour tout ce que j'ai pu apprendre à vos côtés.

Je remercie Anne Boyer et Chantal Soule-Dupuy qui ont rapporté ma thèse et qui ont accepté de faire partie du jury de thèse.

Je remercie aussi l'équipe LIRIS DRIM dans son ensemble. De nombreuses idées de ce travail sont le fruit de discussions avec les membres de l'équipe, dont certains sont devenus pour moi des amis proches.

Je remercie enfin ma famille et mes amis, leur soutien de loin ou de près m'a donné la motivation de terminer ce travail dans cette période difficile de pandémie et d'isolation sociale. Tout particulièrement, je remercie Nathan Blanken pour le soutien dans la vie quoditienne.

Contents

I Introduction 10

1 Context 11

2 Problem Statement 14

3 Contributions 15

3.1 RQ1: How can we incorporate knowledge graph embedding for a recommendation with multiple types of information? 15

3.2 RQ2: How can we combine knowledge graph embedding results with diversification methods? In particular, how can we incorporate knowledge graph embedding with determinantal point processes for a top-N recommendation? 16

3.3 RQ3: How can we evaluate the performance of a diversified recommendation? 16

4 Publications and Communications 16

4.1 International publications 17

4.2 National publications 17

4.3 Scientific Communications 17

5 Plan of the Dissertation 18

II State of the Art 19

6 Recommender Systems 20

6.1 Content-Based Filtering Recommender Systems 20

6.2 Collaborative Filtering Based Recommender Systems 23

6.2.1 Memory-Based Collaborative Filtering 24

6.2.2 Model-Based Collaborative Filtering 27

6.3 Hybrid Recommender Systems 30

7 Knowledge Graph Embedding and KGE Based Recommendation 31

7.1 Tensor Decomposition Based Knowledge Graph Embedding 33

7.2 Translation Based Knowledge Graph Embedding 37

7.3 Deep Neural Network Based Knowledge Graph Embedding 39

7.4 KGE-Based Recommendation 41

8 Diversity-Aware Information Retrieval and Recommendation 43

8.1 Diversity 44

8.2 Diversification 47

8.2.1 Query Disambiguation for Diversification 48

8.2.2 Post-processing Re-ranking for Diversification 48

8.2.3 Diversification Models 49

III Knowledge Graph Embedding Based Diversified Recommendations 51

9 Introduction 52

10 Background 54

10.1 Diversity Metrics for Recommendation 54

10.2 Determinantal Point Processes 57

10.3 Translation-Based Knowledge Graph Embedding for Recommendation . . . 58

11 DivKG: Improving Diversity for Recommendation Based on Knowledge Graphs 59

11.1 Knowledge Graph Embedding for Entity and Relation Representations . . 62

11.1.1 Representation Learning for Entities and Relations 62

11.1.2 Top-N Prediction Based on Knowledge Graph Embedding 63

11.2 Diversity-Aware Recommendation on Determinantal Point Processes 63

11.2.1 Construction of DPP Kernel Matrix 63

11.2.2 MAP Inference for Diversified Prediction 64

12 EMDKG: Diversity-Aware Representation Learning for a Better Trade-off Diversified Recommendation 65

12.1 General Overview 66

12.2 Diverse Item Sets Generation 67

12.3 Item Diversity Learning. 68

12.4 Modified Translation-Based Knowledge Graph Embedding for Recommendation. 70

12.5 Prediction 72

13 Conclusion 72

IV Evaluation 74

14 Introduction 75

15 Experimental Settings **75**

15.1 Datasets 75

15.2 Evaluation Protocol 80

16 Experiment Results **81**

16.1 Experiments on DivKG 82

16.1.1 Baselines 82

16.1.2 Diversity Metrics 83

16.1.3 Experimental Results 83

16.2 Experiments on EMDKG 84

16.2.1 Baselines 84

16.2.2 Performance Metrics 86

16.2.3 General results 86

16.2.4 Impact of trade-off parameter α. 89

16.2.5 Impact of candidate item set length M. 90

16.2.6 Discussion over diverse item set generation 91

17 Conclusions **91**

V Conclusion and Research Directions **97**

18 Conclusion **98**

19 Future Directions **99**

Bibliography **103**

Nomenclature

u, i	User, Item
U,I	The set of total users and total items.
h, t, r	Head entity, tail entity and relation
v_h, v_t, v_r	Head entity embedded vector, tail entity embedded vector, and relation embedded vector
$\odot$	Element-wise product (Hadmard product)

List of Acronyms

CBF	Content-Based Filtering
CF	Collaborative Filtering
CTR	Click-Through Rate
DNN	Deep Neural Network
DPP	Determinantal Point Process
KE	Knowledge Graph
KGE	Knowledge Graph Embedding
ML	Machine Learning

List of Figures

6.1 A content-based filtering method create user profiles from historical preferred items' content and a recommendation can be generated by finding similar items to user profile. 21

6.2 A collaborative filtering method learns from historical user-item interaction to find similar users(explicitly or implicitly). For generating personalized results, items that are preferred by similar users will constitute into a recommendation list for user. 23

6.3 A binarized rating matrix where each column represents the rating scores an item receives and each row represents the rating scores a user gives. The rating score 1 means a positive preference or like between a pair of user and item; and 0 means a negative preference or dislike between a pair of user and item. A '?' for rating score means no historical rating of this pair of user and item. 27

7.1 On the left, there is a knowledge graph for part of MovieLens data, including entities (movies, staff working for the movies, genres) and relations connecting the entities (InActorOf, IsInGenre etc.). On the right it is the latent embedding space where both entities and relations are embedded into. Knowledge graph embedding methods map elements of a knowledge graph (including both entities and relations) into the latent space while conserving the graph structure(node similarity etc.). 33

7.2 Simple illustrations of TransE and TransH. Source: [15]. 37

7.3 A simple illustration of TransD. Source: [69]. 37

8.1 An example of long-tail effect distribution chart. The horizontal axis represents the number of items in this distribution; and the vertical axis represents the occurrences of searches for items. The pink area on the left is the dominant "head" of the distribution while the green area shows a long tail of huge number of items with low searches. 45

11.1 DivKG: a diversity enhancing recommendation framework based on knowledge graph embedding methods. Datasets are preprocessed to extract entity-relation triplets for constructing knowledge graphs for recommendation. Then given such a knowledge graph, we learn the vectorial representations of entities and relations by applying different knowledge graph embedding methods. After obtaining the latent vectors of users, items, other entities and relations, personalized kernel matrix can be constructed for each user. Finally we render each user a diversified recommendation list of items by the fast greedy MAP inference algorithm on its corresponding kernel matrix. 61

11.2 An example of knowledge graph for movie recommendation, reflecting different relations (*Interaction*, *IsGenreOf*, *IsDirectorOf*, *IsPlayedBy*) between various entities (user, item, genre, director, actress). Such a knowledge graph is an extension of user-item interaction used by collaborative filtering methods. 63

11.3 The iterative process of selecting items by FastDPP. Y_g is the dynamic item set containing the items already selected by FastDPP and is initialized as an empty set $\emptyset$. When the number of items in Y_g has not reached N, FastDPP updates Y_g by selecting an item j among all the unselected candidate items which maximizes $log\det(L_{Y_g \cup \{i\}}) - log\det(L_{Y_g})$. 65

12.1 This is the general structure of EMDKG, composed of two modular learning: the Item Diversity Learning and Knowledge Graph Embedding. 67

12.2 A demonstration of how Item Diversity Learning module works. 68

15.1 Movie genre count over all items on dataset ML100K in descending order.(19 genres including Other in total) 76

15.2 Correlation of genres on dataset ML100K and Anime. (a) the correlation of 19 movie genres based on genre distribution of items on dataset ML100K; (b) the correlation of 44 genres based on genre distribution of items on dataset Anime. 77

15.3 Anime genre count over all items on dataset Anime in descending order. The anime genres are comedy, action, fantasy, sci-fi, drama, shounen, kids, romance, school, slice of life, hentai, supernatural, mecha, music, historical, magic, ecchi, shoujo, seinen, sports, mystery, super power, military, parody, space, horror, harem, demons, martial arts, dementia, psychological, police, game, samurai, vampire, thriller, cars, shounen ai, shoujo ai, josei, yuri, yaoi, and other. 78

15.4 The original metadata of datasets ML-100k and IMDb. 79

16.1 Comparison of EMDKG against state-of-the-art approaches on accuracy and diversity metrics when varying the trade-off parameter α on dataset ML100K-IMDb. (a) hit ratio@10 , (b) category coverage (CC)@10, (c) α-NDCG@10. 87
16.2 Comparison of EMDKG against state-of-the-art approaches on accuracy and diversity metrics when varying the candidate item list length (trade-off parameter: 0.5) on dataset ML-100K. (a) hit ratio@10 , (b) category coverage (CC)@10, (c) α-NDCG@10. 88

16.3 Pairwise comparison of EMDKG-E - DPP and EMDKG-E - MMR on accuracy and diversity metrics 93
16.4 Pairwise comparison of DivKG-E and TransKG-E - MMR on accuracy and diversity metrics 94
16.5 Pairwise comparison of EMDKG-H - DPP and EMDKG-H - MMR on accuracy and diversity metrics 95
16.6 Pairwise comparison of DivKG-H and TransKG-H - MMR on accuracy and diversity metrics 96

19.1 Categorical tree extracted from categorical information on dataset Amazon Movie & TV. 100
19.2 The category graph based on item categorical information on dataset Amazon Beauty. 102

List of Tables

7.1 List of Triplets in part of MovieLens data, corresponding to Figure. 7.1 . . 32

11.1 The construction of personalized kernel matrix for DPP. For a user having M candidate items, we construct a $M \times M$ dimension matrix where the diagonal element representing the affinities of each item towards the user (a_{u1} for example represents the affinity or relevance for item 1 towards user u) ;and non-diagonal element representing the pairwise similarity of items (d_{3M} for example represents the similarity between item 3 and item M). Due to the properties of DPP, such a kernel matrix will garantee a set of items with high affinity and diversity will be more likely to be selected from all candidate items. 65

16.1 Accuracy results before diversification with dimension=75, learning rate=0.001 on dataset ML100K-IMDb. 83
16.2 Diversified recommendation results with α=0.9 on dataset ML100K-IMDb. 84
16.3 Accuracy and diversity results before diversification method for `ml100k` extended datasets. Bold results are significantly higher than other results in the same column with $p = 0.01$. 89
16.4 Accuracy and diversity results before diversification method for `anime` datasets. Bold results are significantly higher than other results in the same column with $p = 0.01$. 89

Part I

Introduction

1 Context 11

2 Problem Statement 14

3 Contributions 15

3.1 RQ1: How can we incorporate knowledge graph embedding for a recommendation with multiple types of information? 15

3.2 RQ2: How can we combine knowledge graph embedding results with diversification methods? In particular, how can we incorporate knowledge graph embedding with determinantal point processes for a top-N recommendation? 16

3.3 RQ3: How can we evaluate the performance of a diversified recommendation? 16

4 Publications and Communications 16

4.1 International publications 17

4.2 National publications 17

4.3 Scientific Communications 17

5 Plan of the Dissertation 18

1 Context

Recommender systems have been prevalent for three decades in various domains, ranging from web personalisation [1], music [2,3], news [4], book recommendation [5] to e-commerce [6], group recommendation [7] and other domain-specific application. Recommender systems arise from the phenomenon of information overload on the internet, which makes it difficult for online consumers to receive relevant information. Thus, the existing systems have already proven the huge potential benefits of providing personalized information to their users. Indeed, immersing in such ample quantity of information of different kinds, whether on a news webpage or a social media platform for example, users are easily overwhelmed by the quantity of data or information and it is infeasible to go through all the information for them alone. Thus, it is necessary and of great interest for both users and information providers to apply personalization or recommender systems to filter unrelated information for the users.

In different contexts of applications, the information to be personalized is called differently and *item* as a term is widely used to refer to a unit of information to be recommended or personalized. Thus, a recommender system is essentially composed of two roles: *users* who are to receive recommendation items and *items* which are to be selected and recommended to the users. The method and process of recommendation or personalized selection is the bridge connecting these two by generating the recommendation through recommendation algorithms.

Unlike other information retrieval problems [8, 9] where queries are given to indicate what users are looking for, no direct and explicit requests (in forms of queries for example) are given in a scenario of recommendation. However, by analyzing and processing historical interactions between users and items and other available information relating users and items, we can detect to a certain degree the preference and taste of users towards items. Thus, recommendation algorithms aim to exploit the historical interactions and other available information to the maximum for generating good personalized recommendations for each user.

By intuition, a good recommendation is one that recommends suitable items to users according to each user's personal taste. However, users' personal tastes are not explicitly given by users themselves and should be inferred in recommendation algorithms given various types of available data. Traditionally, by utilizing various types of data through different approaches, three main paradigms of recommender systems have emerged in recommendation algorithm researches, namely collaborative filtering [10] methods, content-based filtering [11] methods and hybrid filtering methods [12, 13] combining the previous two.

Collaborative filtering (CF) methods derive from an assumption that a user may find an item interesting if this item is liked by another user who shares the tastes of the user in question. Departing from this assumption, CF methods aim at finding similar users for each user in order to recommend items preferred by similar users. It is obvious that CF methods can exploit user-item historical interaction data for the recommendation.

The second paradigm is *Content-based filtering (CBF)* methods which build the recommendations on another assumption. They consider the process from the item side instead of user side and assume that an item may interest a user if this item is similar to the items that are already liked by this user. That it to say, CBF methods compare items to each user's historical preferable items and recommend items most similar to the historical items as results. In order to find similar items for each item, item-side information will be used for measuring the similarities.

And the third paradigm *hybrid filtering* methods combine the collaborative and content-based filtering to leverage both user similarities and item similarities for better performance. To achieve this, both historical user-item interactions and item-side information are required. And different hybrid filtering methods exploit the available data to build more complex structures to benefit to the best.

Although huge quantities of innovative works in these three paradigms have been explored by researchers in the domain, the optimization of finding a better recommendation does not stop due to lack of theoretical upper bound for the problem and variety of available data in specific contexts. Particularly, given various types of data relating users and items which can potentially leverage a better recommendation result, there is still lack of a general and simple solution to fully exploit the knowledge hidden inside the data. Knowledge graph embedding methods [14–17] have recently attracted huge attention in research for many machine learning tasks, such as node classification, link prediction and graph completion. A few proposals [18, 19] using knowledge graph embedding combined with collaborative filtering for recommendation have also achieved better results than traditional approaches. In terms of knowledge graphs, users and items can be seen as entities, whereas preference of a user regarding an item can be seen as a kind of relation, link between corresponding entities. From this perspective, a recommendation task can be formulated as an instance of the link prediction problem. Besides, along with user-item preference information, item-side auxiliary information can also be represented in form of $(entity, relation, entity)$. Thus we consider it rather reasonable to build a model based on knowledge graph embedding to find a simple yet efficient solution to recommend taking into account multi-type information source.

When we go beyond intuition and reason what a good recommendation is, **accuracy** has been the first and in most cases the only criterion. Take a CF-based recommender system in industrial scenarios as an example. Such a system implemented on a certain platform consumes huge amount of interaction data between users and items available on the platform to predict users' potential need. The prediction or recommendation to users are expected to be accurate, as non-related item recommendation may disappoint user and thus discourage user from engaging to the platform.

However, recent works [20] have pointed out the inability of focusing solely on accuracy to satisfy users' interest. One example is that recommending two over-similar items to one user may produce redundancy in the result. Similar to an information retrieval scenario, a user expects a recommendation of items to satisfy her interests while still not being duplicates.

Another example is if the user only had a limited range of item interaction history, the recommendation based on such information will also generate highly similar items thus creating "*filter bubbles*" [21] for the users. The term "filter bubble" is brought by Eli Pariser, and designates the isolation effect resulting from the use of algorithms aiming to selectively guess what a user would like based on clicking and search history, separating users from different and opposing viewpoints. The viewpoint narrowing can result in users staying in their cultural or ideological bubbles, exposing them to fake news and the phenomenon of echo chambers, which potentially endangers the RS ecosystem or even the whole internet atmosphere.

Thus, in addition to accuracy, **diversity** of recommendation has been brought up as part of recommendation performance. Furthermore, various user studies [22–24] have confirmed that an exposure to diverse items in recommendation has a positive effect over item list attractiveness towards higher user satisfaction.

As recommendation accuracy is not equivalent to user satisfaction with recommender systems, diversity is introduced as another recommendation goal besides accuracy. Though it is intuitive that a diverse recommendation should provide a list of items one dissimilar to the other as much as possible without compromising its recommendation accuracy, more investigation into how to define and evaluate dissimilarity of items should be given.

Diversity of recommendation results can be understood on two levels, the *individual level* which corresponds to individual recommendation list diversity and the *group (or aggregate) level* which corresponds to the general item distribution over all users.

Aggregate diversity [25–27] targets at improving a more feature-balanced recommendation over all items towards user and alleviating the long-tail effect [6, 28] as much as possible. The long-tail effect is that there is a high frequency of very few well-recognized items exposed to users while large proportion of obscure items have extremely low chance to get exposed.

In contrast, individual diversity aims at providing a diversified personalized item list to each user to enhance personal experience and satisfactions of users with recommender systems. Improving the individual level diversity is a direct solution to redundancy and filter bubble issues. There exist different diversification methods aiming at improving individual diversity [29, 30] in information retrieval. These diversification approaches rely on pairwise dissimilarities among items, however the concept of dissimilarity is coupled with diversity, varying in specific scenarios. Thus, when applying a diversification model to a new accuracy-centered recommendation the effectiveness of such diversification still needs to be verified. More recently diversity models [31, 32] such as the ones based on *determinantal point processes* which capture the both diversity features and user-item relevance within models themselves have caught huge attention. Therefore, it is of great interest to investigate whether such diversity models can be employed for diversified recommendation.

2 Problem Statement

Given the context of recommender systems, our research conducted in this dissertation target at answering the following research questions.

> **RQ 1:** How can we incorporate knowledge graph embedding for a recommendation to take into account multiple types of information?

As we are interested in incorporating knowledge graph embedding methods into a top-N recommendation for processing both historical user-item interactions and item-side information, it is then the most fundamental question to decide how we can do this for both learning the representations and making top-N predictions based on learned models.

> **RQ 2:** How can we combine knowledge graph embedding results with diversification methods? In particular, how can we incorporate knowledge graph embedding with determinantal point processes for a top-N recommendation?

The second question corresponds to our determination to propose a diversified recommendation. Following **RQ 1**, as we decide to incorporate knowledge graph embedding for recommendation, we then need to conceive efficient diversification methods that make use of knowledge graph embedding results. As we are interested in the diversification model - determinantal point processes which can capture both user-item relevance and item diversity, it is then fundamental to propose an approach to effectively link the two models.

RQ 3: How can we evaluate the performance of a diversified recommendation?

The third research question is to answer how we can quantify the performance of diversified recommendation. As a diversified recommendation has two goals accuracy and diversity to achieve, it is not simply about to optimize the two separate goals. The correlations of these two goals must be taken into consideration for achieving a both accurate and diverse recommendation. On the other hand, the concept of individual diversity still varies according to a specific context, it is then essential to discuss whether a diversified recommendation can accommodate various definitions of individual diversities.

To resume, in this dissertation we target at improving individual diversity on a top-N recommendation problem. A top-N recommendation is to recommend each user a list of the length of N items from all the candidate items by learning from historical user-item interactions and other accessible auxiliary information. In addition to provide accurate recommendation results to the users, we consider another quality - the diversity measurement of each recommendation list as our optimization goal. In the end, for each user, we aim to provide a both accurate and diverse item list consisting of N items.

3 Contributions

The contributions of this thesis are two-fold: (1) **DivKG**: Improving Recommendation Diversity Based on Knowledge Graph; (2) **EMDKG**: Diversity-Aware Representation Learning for a Better Trade-off Diversified Recommendation. More precisely these two contributions answer our research question described below.

3.1 RQ1: How can we incorporate knowledge graph embedding for a recommendation with multiple types of information?

To answer this research question, we have established an approach for both DivKG and EMDKG to construct a recommendation knowledge graph which contains both user-item interactions and item-side auxiliary information. We note that user-item interactions can be regarded as a special type of relational triplets, where each instance resembles to $(user, preference, item)$ and $preference$ is the special relation between user and item. For item-side auxiliary information, relational triplets can also be extracted by identifying a relational instance composed of an item, a relation type and a related entity, denoted as $(item, relation, entity)$.

3.2 RQ2: How can we combine knowledge graph embedding results with diversification methods? In particular, how can we incorporate knowledge graph embedding with determinantal point processes for a top-N recommendation?

To address this research question, we first clarify that the combination of knowledge graph embedding with diversification methods can occur during the model learning phase and the diversified recommendation generation phase. In DivKG, our core contribution regarding this RQ is that we propose a novel approach of constructing personalized determinantal point process kernel matrix based on knowledge graph embedding during the recommendation generation phase. And in EMDKG, the diversification model incorporates both knowledge graph embedding learning phase and the diversified recommendation generation phase. This modification comparing to DivKG actually guarantees to encode explicitly a selected individual diversity into the item latent vectors, which can improve the diversity performance for final recommendation results.

3.3 RQ3: How can we evaluate the performance of a diversified recommendation?

After the proposal of diversified recommendation models, we need to support the models with experimental results based on real life datasets (MovieLens-IMDb datasets and Anime datasets) for assessing the effectiveness of the approaches. As the performance of a diversified recommendation considers both accuracy and diversity aspects, we use both accuracy metrics and diversity metrics for measuring the performance of DivKG and EMDKG. For the accuracy we employ two accuracy metrics Hit and NDCG on both DivKG and EMDKG. And for diversity metrics, we employ vector list dissimilarity ILMD and ILAD for DvKG and categorical information based metrics Category Coverage (CC) and α-NDCG in EMDKG.

4 Publications and Communications

These contributions were submitted and accepted by the following international/national conferences:

4.1 International publications

- **Lu Gan**, Diana Nurbakova, Léa Laporte & Sylvie Calabretto (2020). Enhancing Recommendation Diversity using Determinantal Point Processes on Knowledge Graphs. 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 25 July 2020, Virtual (Chine), pp. 2001-2004. doi : 10.1145/3397271.3401213. HAL : hal-02935150. (CORE A*)

- **Lu Gan**, Diana Nurbakova, Léa Laporte & Sylvie Calabretto (2021). EMDKG: Improving Accuracy-Diversity Trade-Off in Recommendation with EM-based Model and Knowledge Graph Embedding. The 20th IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 14 December2021, Melbourne (Australie). doi : 10.1145/3486622.3493925. HAL : hal-03518867. (CORE B)

4.2 National publications

- **Lu Gan**, Diana Nurbakova, Léa Laporte & Sylvie Calabretto (2021). Recommandation diversifiée via des processus ponctuels déterminantaux sur des graphes de connaissances. Extraction et Gestion des connaissances, EGC 2021, 25 January 2021, Montpellier (France), pp. 365-372. HAL : hal-03121974.

4.3 Scientific Communications

During this thesis, the following communications/presentations have been made on the topic of diversified recommendation:

Event	Date	Location	Title
WI-IAT 2021	14 - 17 Dec 2021	Melbourne, Australia (Virtual)	EMDKG: Improving Accuracy-Diversity Trade-off in Recommendation with EM-based Model and Knowledge Graph Embedding
IRIXYS workshop (winter)	29 Nov - 3 Dec 2021	Lyon, France	EMDKG: Incorporating Semantic Diversity to Achieve Better Accuracy-Diversity Trade-off on Knowledge Graph Embedding Based Recommendation
SIGIR 2020	25 - 30 July 2020	Virtual	Enhancing Recommendation Diversity using Determinantal Point Processes on Knowledge Graphs
IRIXYS workshop (summer)	12 - 17 Jun 2020	Virtual	Enhancing Recommendation Diversity using Determinantal Point Processes
IRIXYS workshop (winter)	Dec 2019	Passau, Germany	An Overview: Diversity-Aware Recommender Systems on Knowledge Graphs
IRIXYS workshop (summer)	Jun 2019	Lyon, France	Towards A Diversified and Explainable Recommendation

5 Plan of the Dissertation

This dissertation is organized as follows. In Part II, we introduce the state-of-the-art researches related to my research topics. First, we present recommender systems (Chapter 6) providing a general overview of the domain, then we describe knowledge graph embedding (KGE) techniques and KGE-based recommendation (Chapter 7), and finally, we discuss diversity-aware information retrieval and recommendation (Chapter 8). In Part III, we introduce the proposed models corresponding to our two contributions, DivKG and EMDKG. In Part IV, we provide the evaluation of our two contributions. Finally, in Part V, we present our conclusions and future directions.

Part II

State of the Art

6 Recommender Systems 20

6.1 Content-Based Filtering Recommender Systems 20

6.2 Collaborative Filtering Based Recommender Systems 23

6.3 Hybrid Recommender Systems 30

7 Knowledge Graph Embedding and KGE Based Recommendation 31

7.1 Tensor Decomposition Based Knowledge Graph Embedding 33

7.2 Translation Based Knowledge Graph Embedding 37

7.3 Deep Neural Network Based Knowledge Graph Embedding 39

7.4 KGE-Based Recommendation 41

8 Diversity-Aware Information Retrieval and Recommendation 43

8.1 Diversity 44

8.2 Diversification 47

6 Recommender Systems

Recommender systems have been prevalent for three decades, arising from overloading information and the need for filtering overwhelming quantity of information, on the one hand, and the interest of personalized information towards information consumers. Over the decades, various techniques and structures have been proposed to satisfy this personalization given a plethora of information related to both information consumers and recommended content itself. A recommender system is essentially composed of two type of components: items which are to be selected for constucting the personalized recommendation list and users who receive a list of items. Various types of information can be used for this personalization process, including user-item information, user-side auxiliary information and item-side information.

In this chapter we introduce the traditional three categories of recommender systems, namely *content-based filtering*, *collaborative filtering* and *hybrid methods*.

6.1 Content-Based Filtering Recommender Systems

Content-based filtering (CBF) methods [33, 34] are the first approaches of personalized information filtering to overcome information overload problems. As indicated by its name, the idea of content-based filtering is to build user profiles based on user preferred content and by matching new items' content with the user profile we can create recommendation lists. In Figure.6.1 we demonstrate the idea of content-based filtering for recommendation.

In order to build user profiles based on user preferred content, two tasks are required. The first task is to analyze historically preferred items' content (traditionally in textual data, denoted as $Content(i)$ for item i) to extract features or properties for representing the items. The second task is to collect and aggregate the features and properties of all items that are liked by a user to build and learn a corresponding user profile for item recommendation. Naturally both representing items and user profiles can employ different methods.

For the first task, as the items' content is usually textual data, vector space model (VSM) and extended language models, e.g.*term frequency-inverse document frequency* (TF-IDF) [35, 36], pLSA [37] and LDA [38], have been applied for item representation in CBF. For content-based methods which have employed TF-IDF, items are represented as weighted term vectors. More specifically, as items are originally represented as textual data or documents, the set of $|I|$ items is denoted as $I = \left\{i_1, ..., i_{|I|}\right\}$ and the content of the

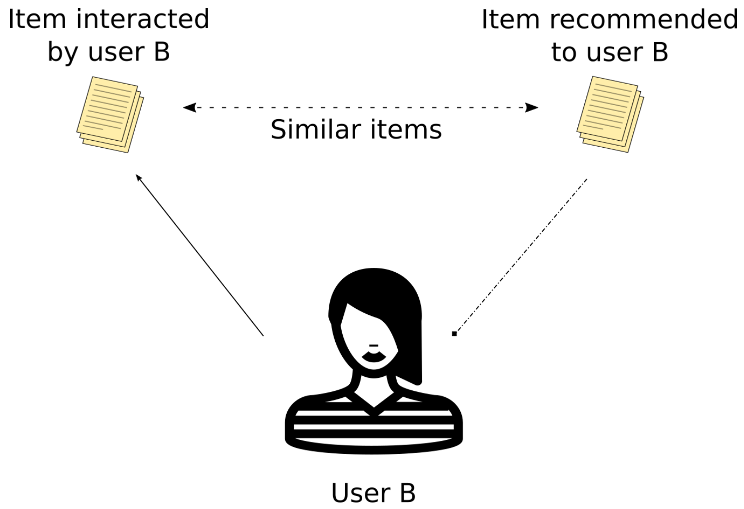


Figure 6.1: A content-based filtering method create user profiles from historical preferred items' content and a recommendation can be generated by finding similar items to user profile.

items is represented as the set $Content(I) = \left\{Content(i_1), Content(i_2), ..., Content(i_{|I|})\right\}$, and $Terms = \{t_1, t_2, ..., t_m\}$ is the set of terms appeared in all the documents/items or called as *dictionary*.

For each item i, the content is a collection of l terms, denoted as $Content(i) = \{t_{i,1}, t_{i,2}, ..., t_{i,l}\}$. The similarity between a term $t_{i,k}$ and an item content $Content(i)$ is denoted as $w_{i,k}, i \in \{1, 2, ..., |I|\}, k \in \{1, 2, ..., l\}$ and can be written as,

$$w_{i,k} = \frac{\mathrm{TF-IDF}(t_{i,k}, Content(i))}{\sqrt{\sum_{k=1}^{k=l} \mathrm{TF-IDF}(t_{i,k}, Content(i))^2}} \tag{6.1}$$

which is a normalized TF-IDF between the term $t_{i,k}$ and item content $Content(i)$. And the TD-IDF of $t_{i,k}$ and $Content(i)$ is formulated as,

$$\mathrm{TF-IDF}(t_{i,k}, Content(i)) = \frac{f_{k,i}}{\max_j f_{j,i}} \dot{\log} \frac{n}{n_k} \tag{6.2}$$

where $f_{k,i}$ is the frequency of term $t_{i,k}$ in item content $Content(i)$ and n_k is the number of documents in which the term $t_{i,k}$ has appeared at least once. Thus, an item content $Content(i)$ can be the represented as a vector of similarities with all terms, denoted as $Content(i) = \{w_{i,1}, w_{i,2}, ..., w_{i,l}\}$, where $w_{i,k}, k \in \{1, 2, .., l\}$ is given in Eq.(6.1).

For the second task to analyze and to learn user profiles, several ways have been proposed. The first and also intuitive way is to represent the user profile in a similar way as item content. In a similar fashion, the user profile for user u is composed of terms $Terms$, denoted as $ContentBasedProfile(u) = \{w_{u,1}, w_{u_2}, ..., w_{u,l}\}$. Then for recommendation the relevance between user u and item i can be calculated based on a score function based on the weighted term vectors in item content and user profile, written as,

$$relevance(u, i) = score(ContentBasedProfile(u), Content(i)) \tag{6.3}$$

Another way [39, 40] for learning user profiles is to regard the recommendation task as a binary text categorization task: given a new item with its content, the acquired user profile can match this item and predict whether it belongs to the positive or negative category, where *positive* means this item will compose the recommendation list and *negative* means the opposite. Thus, the learning of user profiles is to learn a two-class classification model with labeled historical items with their content represented by a collection of weighted term vectors as training dataset. One of the most common used methods for this two-class classification task is Naïve Bayes classifiers. In CBF settings, Naïve Bayes is a probabilistic model which estimates the *a posteriori* probability $P(c|d)$ of the a document d belonging to the class c, $c \in \{c_+, c_-\}.c_+$ and c_- represent separately the positive and negative category. According to the Bayes theorem, the *a posteriori* probability $P(c|d)$ can be written as,

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \tag{6.4}$$

where $P(d|c)$ is the probability of seeing document d given a class c and $P(c)$ is the probability of observing documents in category c. As $P(d)$ represents the probability of observing document d and is the same for either c_+ or c_-, so to decide which category document d falls in, we choose the category $\hat{c}$ by such rule,

$$\hat{c} = argmax_{c_i} P(c_i)P(d|c_i) \tag{6.5}$$

To estimate $P(d|c_i), c_i \in \{c_-, c_+\}$, the Naïve Bayes classifier uses an independence assumption: all terms in a document d are mutually independent, conditional on c_i. Thus,

$$P(c|d_i) = P(c|t_{i1}, t_{i2,..,t_{il}}) \tag{6.6}$$

Even though by recommending items with similar content can to a certain extend satisfy users' need because of the correlation between the content of items and users' preference, content-based filtering has been pointed out suffering from the following limitations [41]:

- Content-based filtering approaches in general supply a very shallow analysis of content [12] and can not provide items based on the assessment of quality or opinions. A highly rated item is no different to a poorly rated item if the content of them are highly similar or identical.

- Content-based approaches tend to produce overspecialization [42] in recommendation due to only comparing item content to historical items' content which may represent users' taste with bias.

- Coupling to the second point, content-based filtering approaches do not have a way to find serendipitous items for users. A serendipitous item to a user is one could satisfy the user's taste but is poorly relevant to user's historical interactions.

Due to these limitations, other filtering techniques have been proposed, and one type is particularly successful and has been widely used, which is *collaborative filtering*.

6.2 Collaborative Filtering Based Recommender Systems

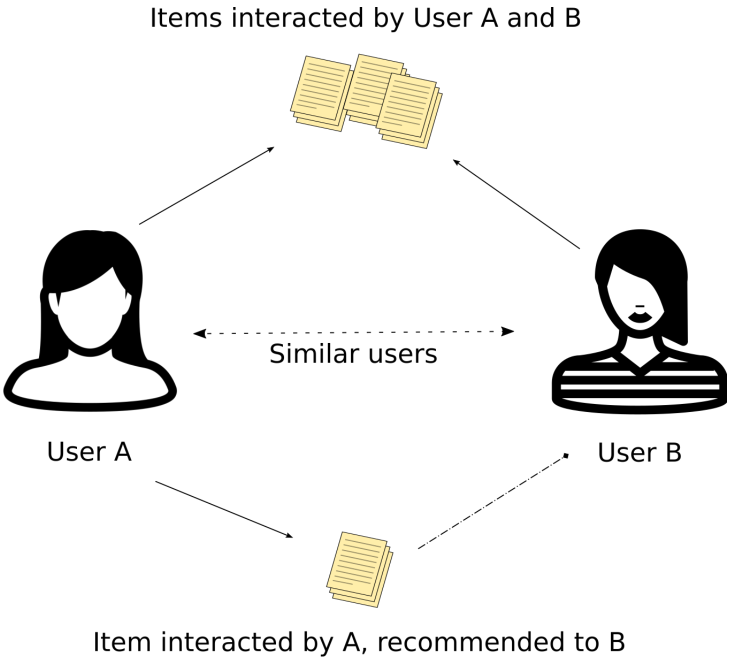

Items interacted by User A and B
Similar users
User A
User B
Item interacted by A, recommended to B

Figure 6.2: A collaborative filtering method learns from historical user-item interaction to find similar users(explicitly or implicitly). For generating personalized results, items that are preferred by similar users will constitute into a recommendation list for user.

Collaborative filtering (CF) is a category of techniques which are frequently used in recommendation and personalization and often achieve good performance in terms of accuracy. The main idea of CF is that a user's taste can be inferred through collecting and analyzing the tastes of similar users. To determine similar users of one particular user, the historical interactions between users and items can be used. There are various approaches to collect and use historical user-item interactions for inferring user's taste, and regarding the way of using the information, CF methods are in general categorized into *Memory-Based collaborative filtering* [10, 43, 44] and *Model-Based collaborative filtering* [45–50]. In Figure.6.2, we show the idea of collaborative filtering for recommending items to users.

6.2.1 Memory-Based Collaborative Filtering

As CF methods determine users' tastes by analyzing from historical user-item interactions, *memory-based collaborative filtering* [10, 43, 44, 51–53] directly use the historical rating data to calculate the relevance or similarity between users and items. This category of CF methods get their name by only needing to *memorize* the historical user-item interactions such as the rating data. We denote the historical rating data as $Rating(U) = \Big\{Rating(u_1), Rating(u_2), ..., Rating(u_{|U|})\Big\}$, where $Rating(u) = \{rating_{u,i} | i \in I_u\}$ represents the collection of ratings the user u has given towards different items I_u.

Thus, to predict the relevance between a new item i for user u, memory-based filtering provides an aggregation function based on user's historical ratings and can be written as,

$$relevance_{u,i} = \underset{u' \in Neighbor(u)}{\text{aggregate}} \; rating_{u',i} \tag{6.7}$$

where $Neighbor(u)$ is the set of users that are most similar to user u.

The selection of function aggregate is varied and here we give a few formulations,

$$relevance(u,i) = \frac{1}{|Neighbor(u)|} \sum_{u' \in Neighbor(u)} rating_{u',i} \tag{6.8}$$

$$relevance(u,i) = \frac{1}{\sum\limits_{u' \in Neighbor(u)} sim(u,u')} \sum_{u' \in Neighbor(u)} sim(u,u') * rating_{u',i} \tag{6.9}$$

$$\begin{aligned} relevance(u,i) = & \overline{rating_u} + \frac{1}{\sum\limits_{u' \in Neighbor(u)} sim(u,u')} \sum_{u' \in Neighbor(u)} * \\ & sim(u,u') * (rating_{u',i} - \overline{rating_{u'}}) \end{aligned} \tag{6.10}$$

where $sim(u,u')$ is a similarity measure between a pair of users u and u' and $\overline{rating_u}$ represents the average rating score a user u gave the items. Eq. (6.8) is a simple average of all ratings of similar users regarding an item. Eq. (6.9) is an extended version of Eq. (6.8) by including the similarity between users as weight to differentiate the influence of users. And Eq. (6.10) further extends Eq. (6.9) by weighing in average rating scores of users to consider user related score scale difference.

So the essential task for memory-based CF approaches is an appropriate solution of finding similar users for each user. *Mean Squared Difference (MSD)*, *Pearson Coefficient* and *Cosine Similarity* have been employed for this purpose to calculate similarity of users based on co-rated items. We represent the co-rated items between a pair of users u and u' as $CoR_{u,u'} = \{i | i \in I, \exists rating_{u,i} \exists rating_{u',i}\}$. For mean squared difference, the similarity between two users u and u' is given as,

$$sim_{MSD} = \sum_{i \in CoR_{u,u'}} (rating_{u,i} - rating(u', i))^2 \tag{6.11}$$

And Pearson coefficient for calculating the user similarity can be written as,

$$sim_{Pearson} = \frac{\sum\limits_{i \in CoR_{u,u'}} (rating_{u,i} - \overline{rating_u})(rating_{u',i} - \overline{rating_{u'}})}{\sqrt{\sum\limits_{i \in CoR_{u,u'}} (rating_{u,i} - \overline{rating_u})^2 \sum\limits_{i \in CoR_{u,u'}} (rating_{u',i} - \overline{rating_{u'}})^2}} \tag{6.12}$$

The cosine similarity on the other hand is given as,

$$sim_{cos} = \frac{\sum\limits_{i \in CoR_{u,u'}} rating_{u,i} * rating_{u',i}}{\sqrt{\sum\limits_{i \in CoR_{u,u'}} (rating_{u,i})^2 \sum\limits_{i \in CoR_{u,u'}} (rating_{u',i})^2}} \tag{6.13}$$

GroupLens proposes in [10] an architecture using Pearson coefficient and the a relevance given in Eq. (6.9) for generating personalized items to users. And Ringo is another memory-based CF approach proposed in [43] and has employed mean square difference and Pearson coefficient for finding similar users and predicting recommendation items. In [51] they have used both Pearson coefficient and cosine similarity with relevance function shown in Eq.(6.10) for making memory-based recommendation. Memory-based collaborative filtering relies heavily on finding similar users with similarity calculations $|U| \times |I|$ in total, which becomes expensive and time-consuming for scaling up. Some techniques [52] to prune the user profiles first before the similarity calculations thus have been proposed. In [52] they propose four different approaches to filter user profiles, accordingly by selecting new user profiles, rational user profiles, both new and rational profiles and user profiles with high utility score. By applying these approaches, the recommendation has been accelerated because of the reduction of calculating size whiling slightly improve the recommendation accuracy. Delgado et al. have argued that the proposed similarity functions shown as Eq. (6.11), (6.12) and (6.13) have no well founded theory and that learning weights through a combination of memory-based individual predictions and online weight-majority voting to replace the similarity is a better solution.

All the memory-based filtering methods above use aggregation function based on neighboring users' rating data, however, [44] and [53] replace the aggregation of ratings of similar user with aggregation of ratings of similar items. Thus to suit to change, the generic relevance function will be modified based on Eq.(6.7), written as,

$$relevance_{u,i} = \underset{i' \in I_u}{\text{aggregate}}\, rating_{u,i'} \tag{6.14}$$

where I_u is a list of N items that are most *coarsely* most relevant to the given user u. To obtain this list of items for each users, either a direct selection on rated items can be used or the similarity between the candidate items and rated items can be calculated and a top-N item list can be returned. The similarity function between candidate items and rated items can take form of aforementioned similarity metrics, namely, mean squared difference, Pearson coefficient and cosine similarity. An example of similarity functions for two items i and i' based on co-rating history are given as such. For a Pearson coefficient-based item similarity, it is written as,

$$sim_{Pearson} = \frac{\sum\limits_{u \in CoR_{i,i'}} (rating_{u,i} - \overline{rating_i})(rating_{u,i'} - \overline{rating_{i'}})}{\sqrt{\sum\limits_{u \in CoR_{i,i'}} (rating_{u,i} - \overline{rating_i})^2 \sum\limits_{u \in CoR_{i,i'}} (rating_{u,i'} - \overline{rating_{i'}})^2}} \tag{6.15}$$

where $CoR(i, i')$ is the set of users who have co-rated the two items. Then by applying Eq.(6.14), a refined and final item list for each user can be returned by sorting the relevance between user u and item i. Empirical results have shown a better performance compared to user memory-based collaborative filtering approaches.

Despite memory-based collaborative filtering methods have overcome some limits of content-based filtering, e.g. being able to provide quality-measured items to users and empirical results have shown certain success, there are still major shortcomings, namely,

- Memory-based collaborative filtering requires to calculate for each item a similarity function involved with a big chunk of all data to generating the recommendation. In consequence it presents serious scalability problems when applying memory-based methods. Particularly in online recommendation scenarios which are expected with short-time response, a memory-based CF would require complicated infrastructure to provide sufficient service.

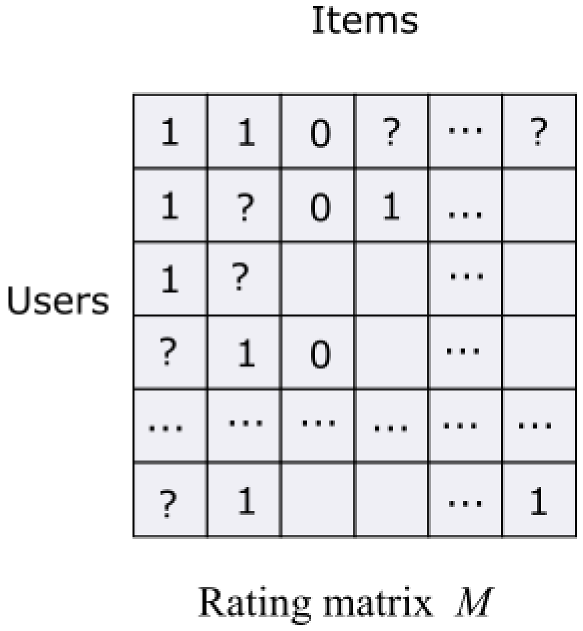
- Memory-based collaborative filtering relies directly on historical co-rating data for a pair of users or co-rated data for a pair of items, which in reality is highly scarce. That being said, each user only rates a small subset of total items, giving very little information for user item relevance calculation. And furthermore, even only considering users who have historically rated at least a fixed number of items and items which have been at least rated for a minimum times, the overlap of co-rated items or co-rating users will still present small in general. Therefore, based on only a few observations of correlation data (co-rating users or co-rated items), the computed relevance should not be considered as a reliable measure [45]. Besides, it is inevitable that memory-based collaborative filtering approaches will perform much worse when datasets present high sparsity.

- Memory-based collaborative filtering only considers the neighboring rating data of users or items, but it has been argued that ratings given by a user can still be useful to another user even if the user is not in another user's neighborhood [45].

Regarding these limits, model-based collaborative filtering methods have been proposed to partially alleviate the problems. And we give a more detailed introduction in next subsection.

6.2.2 Model-Based Collaborative Filtering

Model-based collaborative filtering approaches [45, 46, 48–50] also use historical rating data to recommend items that are preferred by other users with similar tastes. But instead of calculating the relevance directly with rating data, model-based CF methods build or learn a model first and the recommendation will be generated using this model under fewer calculations compared to memory-based CF. Various models from pattern recognition and machine learning tasks have been employed for recommendation, taking examples as *Singular Value Decomposition* [45], *Principal Component Analysis* [46], *Matrix Factorization* [47, 48] and *Neural Networks* [50].


Items
Users
Rating matrix M

1	1	0	?	···	?
1	?	0	1	···	
1	?			···	
?	1	0		···	
···	···	···	···	···	···
?	1			···	1

Figure 6.3: A binarized rating matrix where each column represents the rating scores an item receives and each row represents the rating scores a user gives. The rating score 1 means a positive preference or like between a pair of user and item; and 0 means a negative preference or dislike between a pair of user and item. A '?' for rating score means no historical rating of this pair of user and item.

The historical rating data can be represented as a rating matrix, demonstrated in Fig.6.3, and is in general highly sparse. The recommendation of items towards each user can be simplified as predicting the unknown rating score (either 1 or 0) and return the list of items with score 1 as final results. If we consider each user's preference (like and dislike) as a feature then the prediction problem is a typical classification problem in machine learning. However, the dimension of features is proportional to the number of users which may suffer from the curse of dimensionality [54] when the user count is too big. In the mean time, if treating unknown rating score as negative (which is a common practice during training procedure), each item's feature vector is mostly composed of 0 and thus also highly sparse. Some methods [45, 46] are therefore inspired by dimensionality reduction solutions in machine learning for this classification. D. Billsus et al. [45] presented a recommendation based on *Singular Value Decomposition (SVD)*. For a rectangular matrix $A \in \mathbb{R}^{b \times c}$ with rank m, SVD can decompose it into the product of three matrices,

$$A = U \Sigma V^{\top} \tag{6.16}$$

where U and V are orthonormal vectors and Σ is a diagonal matrix containing the singular values. In their proposal, all the feature vectors of items compose a new rectangular matrix where SVD can be applied to obtain singular values which can rescale the feature vectors for prediction. Eigentaste [46] is another dimensionality reduction-based recommendation where *Principal Component Analysis (PCA)* is applied on the item correlation matrix to get a fix number of eigen vectors for an 'eigen-plane'. The eigen-plane can be used to map rating matrix into lower continuous space for clustering and then generation of recommendation lists.

Matrix Factorization (MF) is another category of model-based CF methods where a rating matrix is directly learnt to map into a low-rank matrix and recommendation can be inferred from the low-rank matrix. Given a rating matrix M, Funk matrix factorization [55] proposes to predict the rating matrix $\widetilde{M}$ by learnt item latent vectors $V_I \in \mathbb{R}^{|I| \times d}$ and user latent vectors $V_U \in \mathbb{R}^{|U| \times d}$ by a matrix multiplication operation,

$$\widetilde{M} = V_U V_I^{\top} \tag{6.17}$$

To learn the item latent vectors and user latent vectors,

$$loss : arg \min_{V_U, V_I} \|M - \widetilde{M}\|^2 + \lambda_1 \|V_U\|^2 + \lambda_2 \|V_I\|^2 \tag{6.18}$$

where λ_1 and λ_2 are parameters for model regularization. SVD++ [56] is another matrix factorization-based model which also incorporates a neighborhood model, which considers both explicit feedback $Rating^{expl}$ and implicit feedback $Rating^{impl}$, written as

$$\begin{split}\widetilde{M}_{u,i} = b_{u,i} + V_i^{\top}(\sum_{j \in Rating^{expl}(u)} (M_{u,j} - b_{u,j}) x_j + \\ |Rating^{impl}(u)|^{-1/2} \sum_{j \in Rating^{impl}(u))} y_j)\end{split} \tag{6.19}$$

where $M_{u,i}$ is the element of row u and column i in M, $\widetilde{M}_{u,i}$ is the element of row u and column i in $\widetilde{M}$, V_i is the i-th row vector in V_I. $b_{i,j}$ is the bias vector for the pair of user u and item i. x_j and y_j are parameters to be learned for weighing different items. To accommodate the modification of the prediction of M the loss function is also modified accordingly, written as,

$$\begin{aligned} loss : & arg \min_{V_I, x_*, y_*, b_*} \sum_{(u,i)} (M_{u,i} - b_{u,i} \\ & - V_i^\top (\sum_{j \in Rating^{expl}(u)} (M_{u,j} - b_{u,j}) x_j + \frac{1}{\sqrt{|Rating^{impl}(u)|}} \sum_{j \in Rating^{impl}(u))} y_j))^2 \\ & + \lambda (V_u^2 + V_i^2 + b_{u,i}^2 + \sum_j x_j^2 + \sum_j y_j^2)^2 \end{aligned} \tag{6.20}$$

PMF [47] is also a matrix factorization-based CF method, where probabilistic graphical models are introduced to restrain the item latent vectors, user latent vectors and predicted rating matrix. The item, user latent vectors and predicted rating matrix are assumed to follow the Gaussian distribution, and the loss function is therefore the posterior distribution over users and items, written as,

$$\begin{aligned} loss : \ln p(V_U, V_I | M, \sigma, \sigma_U, \sigma_V) = & \| M - \widetilde{M} \|^2 \\ & + \lambda_1 \| V_U \|^2 + \lambda_2 \| V_I \|^2 \end{aligned} \tag{6.21}$$

All the matrix factorization-based methods above all optimize an element-wise RSME loss function with regard to the true explicit/implicit feedback of a pair of user and item. The choice of such a loss function corresponds to a regression/classification problem. However, item recommendation especially top-N recommendation is preferred to be regarded as a personalized ranking problem. Therefore to directly optimize a ranking a generic optimization criterion *BPR-OPT* is proposed in [48]. The BPR-OPT criterion is given as,

$$\begin{aligned} BPR - OPT & \doteq \ln p(\Theta | >_u) \\ & = \sum_{(u,i,j)} \ln \sigma(\widetilde{x}_{u,i,j}) + \lambda_\Theta \| \Theta \|^2 \end{aligned} \tag{6.22}$$

where Θ is the set of parameters in a model, σ is the sigmoid function and $>_u$ represents the preference (ranking) structure for a user u. The triple (u, i, j) is preference structure for user u with items i and j.

FISM [49] is also based on matrix factorization but adds a discount factor for the item and user latent vector multiplication. Besides, FISM applies both RMSE loss and the BPR-OPT criterion for parameter learning, and empirical results show a more stable result given by BPR-OPT criterion.

IRGAN [50] applies a *minimax* game to the matrix factorization-based CF. Inspired by *Generative Adversarial Nets (GAN)* [57], IRGAN is composed of a generative retrieval model (G) and discriminative retrieval model (D) which both are a matrix factorization-based CF model. The overall loss function for learning the generative and discriminative models is by a minimax optimization, written as,

$$\begin{aligned} J^{G,D} \doteq \min_{\theta} \max_{\phi} \sum_{u \in U} [\mathbb{E}_{i \sim p_{\text{true}}(i|u,Rating)}[\log \sigma(f_{\phi}(u,i))] + \\ \mathbb{E}_{i \sim p_{\theta}(i|u,Rating)}[\log(1 - \sigma(f_{\phi}(u,i)))]] \end{aligned} \tag{6.23}$$

where the generative model tries to estimate the real relevance distribution for user u towards item i as a conditional probability $p_{\text{true}}(i|u, Rating)$ given the historical user-item interactions $Rating$, and is denoted as $G(i|u) = p_{\theta}(i|u, Rating)$. The discriminative model D models and optimizes the probability of item i is preferred by user u, written in the sigmoid function $D(i|u) = \sigma(f_{\theta}(u, i))$. And both the generative model $G(i|u)$ and discriminative model $D(i|u)$ in item recommendation setting are score functions of matrix factorization-based methods, given as,

$$\begin{aligned} G(i,u) &= \sigma(s_{\theta}(u,i)) = \sigma(b_i^{\theta} + v_u^{\theta\top} v_i^{\theta}) \\ D(i,u) &= \sigma(s_{\phi}(u,i)) = \sigma(b_i^{\phi} + v_u^{\phi\top} v_i^{\phi}) \end{aligned} \tag{6.24}$$

Model-based collaborative filtering methods have been reported to have a better performance than memory-based collaborative filtering methods [51] and can alleviate the sparsity problem to some degree. However, with emerging issues such as cold start problems [58] which means the difficulty of handling new users with very few historical interactions with items or new items in the recommender system, new techniques are proposed through combining various types of recommendations to maximize the benefits of their advantages while limiting the disadvantages. These new techniques are categorized as hybrid recommender systems.

6.3 Hybrid Recommender Systems

As content-based filtering and collaborative filtering both have their own advantages and disadvantages, it is natural to think of combining the two methods for benefiting from the advantages while avoiding the weakness. Thus, various types of hybrid recommender systems (weighted [59], switching [60], mixed [61], feature combination [62], feature augmentation [63], cascade, meta-level [12]) have been proposed incorporating the two for achieving a more satisfying result [64].

In [59] they propose to compute a weighted score for final recommendation from separate CF and CBF learning components. NewsDude [60] is a switching-based hybrid method which chooses between CF and CBF recommendation for the final recommendation. In mixed hybrid method [61], recommendation results from different recommendation components are mixed and presented together to the users. The feature combination-based hybrid method [62] exploits features from various sources and combines them together to train a single recommendation model. While feature augmentation-based method [63] takes existed features to compute new features and use the new features as input for generating recommendation. Fab [12] proposes a meta-level hybrid system by maintaining both user profiles based on content analysis (CBF task) and comparing profiles to find similar users (memory-based CF task) at the same time. The recommendation generation will be a merge of the two recommendation results.

7 Knowledge Graph Embedding and KGE Based Recommendation

In previous chapter, we have seen several traditional recommendation techniques, using item textual content information, or user-item interactions or both. Empirical results have shown that using different sources of information from recommendation scenarios tend to boost recommendation. Hybrid methods have benefited from this combination of information but the way of traditional hybrid methods are in general either a simple aggregation of the two other methods lacking a more general and unified structure or too expensive and time consuming. Thus, given various kinds of information (user-item interactions, item-side auxiliary information etc) new approaches based on knowledge graph embedding have been proposed.

Knowledge graph embedding methods focus on learning latent representations of elements on knowledge graph. A knowledge graph is a graph-based data model, containing entities (vertices on the graph) and relations (links) connecting the entities for integrating knowledge and information. Various methods have been proposed to map knowledge graphs in latent vector spaces. The applications to link prediction, triple classification, entity recognition, clustering etc. have been proved to be successful in both academic works and industry.

A formal definition of a knowledge graph is given as such.

Definition 1. A knowledge graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$ is a set of entities $\mathcal{E}$, relations $\mathcal{R}$ and triplets $\mathcal{T}$. A triplet T is represented as (h, r, t) where $h \in \mathcal{E}$ is the head entity, $t \in \mathcal{E}$ is the tail entity and $r \in \mathcal{R}$ is the relation of the triplet. We denote ζ as the set of all correct combinations of triplets given $\mathcal{E}$ and $\mathcal{R}$. And the triplets $\mathcal{T}$ in the knowledge graph $\mathcal{G}$ is a subset of ζ.

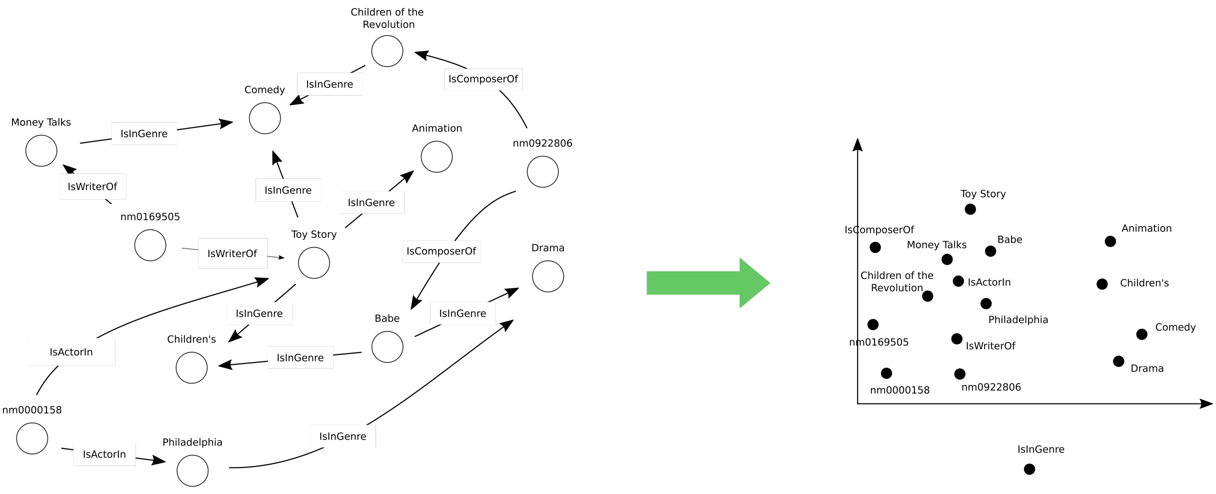
In Figure 7.1 we demonstrate a simple example of knowledge graph embedding for part of MovieLens data. In this example, the triplets of the knowledge graph shown on the left are listed as follows: And all the entities and relations shown in the Table. 7.1

Table 7.1: List of Triplets in part of MovieLens data, corresponding to Figure. 7.1

Head	Relation	Tail
Toy Story	IsInGenre	Comedy
Toy Story	IsInGenre	Animation
Toy Story	IsInGenre	Children's
nm0169505	IsWriterOf	Toy Story
nm0169505	IsWriterOf	Money Talks
nm0000158	IsActorIn	Toy Story
nm0000158	IsActorIn	Philedelphia
Money Talk	IsInGenre	Comedy
Philedelphia	IsInGenre	Drama
Babe	IsInGenre	Drama
Babe	IsInGenre	Children's
nm0922806	IsComposerOf	Babe
nm0922806	IsComposerOf	Children of the Revolution
Children of the Revolution	IsInGenre	Comedy

are mapped into latent space as latent vectors and shown in the right part of the Figure.7.1.

In this section we first introduce three main categories of prevalent knowledge graph embedding techniques: the tensor decomposition based knowledge graph embedding, translation based knowledge graph embedding and the most recent deep neural network based graph embedding. We then present the state-of-the-art methods for incorporating knowledge graph embedding methods for enhancing recommendation performance.



Knowledge graph structure

Embeddings on the latent space

Figure 7.1: On the left, there is a knowledge graph for part of MovieLens data, including entities (movies, staff working for the movies, genres) and relations connecting the entities (InActorOf, IsInGenre etc.). On the right it is the latent embedding space where both entities and relations are embedded into. Knowledge graph embedding methods map elements of a knowledge graph (including both entities and relations) into the latent space while conserving the graph structure(node similarity etc.).

7.1 Tensor Decomposition Based Knowledge Graph Embedding

Tensor decomposition based knowledge graph embedding methods are a group of methods using decomposed matrices to represent knowledge graph triplet information. That is to say the score function for measuring a triplet (h, r, t) is based on multiplications of matrices of entity and relation vectors. RESCAL [65], DisMult [16], *SimplE* [17], ComplEx [66] are typical bilinear tensor decomposition based knowledge graph embedding methods; and HolE [67] and TuckerER [68] are other linear tensor decomposition based knowledge graph embedding methods.

RESCAL [65] incorporates a tensor factorization model to represent the translation from entity h to entity t through relation r, and the score function of the triplet $(h, r, t)$is formulated as,

$$score_{RESCAL}(h, r, t) : v_h^\top M_r v_t \tag{7.1}$$

where $M_r \in \mathbb{R}^{d \times d}$ is the square matrix that models the interaction between entities and $v_h \in \mathbb{R}^d$ and $v_t \in \mathbb{R}^d$ are entity vectors for h and t separately. And d is the latent dimension for the embedding. We also point out that this square matrix M_r is by default asymmetric, which means the relation between a pair of entities does not have to be symmetric.

DistMult [16] proposes to interact the relation vector r with the entity h and t in an element-wise multiplication way, and the score function of DistMult for the triplet (h, r, t)

is formulated as,

$$score_{DistMult} : (v_h \odot v_r)^\top v_t \tag{7.2}$$

where $v_h \in \mathbb{R}^d$,$v_r \in \mathbb{R}^d$,and $v_t \in \mathbb{R}^d$ are embedded vectors for h ,r and t correspondingly and $\odot$ is the element-wise (Hadmard product) multiplication operator. We can easily tell the relation parameter size ($\mathbb{R}^d$) is much smaller than that of RESCAL ($\mathbb{R}^{d\times d}$). Though as the score function does not distinguish between head and tail entities, DistMult can only model symmetric relations.

ComplEx [66] confirms the effectiveness of dot product as the score function in knowledge graph embedding but it argues that extending real-valued vectors to complex-valued vectors will make the embedding more expressive while maintain a low parameter size. Thus, ComplEx uses a complex eigen decomposition form for representing the score function, formulated as,

$$score_{ComplEx} : \mathrm{Re}(v_h^\top w_r \overline{v_t}) \tag{7.3}$$

where $v_h \in \mathbb{C}^d$, $v_t \in \mathbb{C}^d$, $w_r \in \mathbb{C}^d$ and $\mathrm{Re}(\cdot)$ denotes the real part of a complex value. It is customary to represent the real part and imaginary part separately for a complex value, so for each entity e, we let $\mathrm{Re}(v_e) \in \mathbb{R}^d$ and $\mathrm{Im}(v_e) \in \mathbb{R}^d$ represent the real and imaginary parts. And for each relation r, we also let $\mathrm{Re}(v_r) \in \mathbb{R}^d$ and $\mathrm{Im}(v_r) \in mathbbR^d$ represent the real and imaginary parts of r. As we have in ComplEx,

$$v_e = \mathrm{Re}(v_e) + \mathrm{Im}(v_e)i, v_r = \mathrm{Re}(v_r) + \mathrm{Im}(v_r)i \tag{7.4}$$

and $i^2 = -1$,we can rewrite Eq. 7.3 into such,

$$\begin{aligned} score_{ComplEx} =& \mathrm{Re}(v_h^\top w_r \overline{v_t}) \\ =& \mathrm{Re}(\sum_{j=1}^{d} v_{hj} * w_{rj} * v_{tj}) \\ =& \mathrm{Re}[\sum_{j=1}^{d} (\mathrm{Re}(v_h) + \mathrm{Im}(v_h)i) * (\mathrm{Re}(v_r) + \mathrm{Im}(v_r)i) * (\mathrm{Re}(v_t) + \mathrm{Im}(v_t)i)] \\ =& \sum_{j=1}^{d} [\mathrm{Re}(v_{hj}) * \mathrm{Re}(v_{rj}) * \mathrm{Re}(v_{tj}) \\ & + \mathrm{Im}(v_{hj}) * \mathrm{Re}(v_{rj}) * \mathrm{Im}(v_{tj}) \\ & + \mathrm{Re}(v_{hj}) * \mathrm{Im}(v_{rj}) * \mathrm{Im}(v_{tj}) \\ & - \mathrm{Im}(v_{hj}) * \mathrm{Im}(v_{rj}) * \mathrm{Re}(v_{tj})] \\ =& \langle \mathrm{Re}(v_h), \mathrm{Re}(v_r), \mathrm{Re}(v_t) \rangle + \langle \mathrm{Im}(v_h), \mathrm{Re}(v_r), \mathrm{Im}(v_t) \rangle + \\ & \langle \mathrm{Re}(v_h), \mathrm{Im}(v_r), \mathrm{Im}(v_t) \rangle - \langle \mathrm{Im}(v_h), \mathrm{Im}(v_r), \mathrm{Re}(v_t) \rangle \end{aligned} \tag{7.5}$$

where $\langle v, w, x \rangle \doteq \sum_{j=1}^{d} v_j * w_j * x_j$ and v_j,w_j and x_j represents the j-th element of the corresponding vectors v,w and x.

SimplE [17] exploits another decomposition - the *canonical Polyadic (CP)* decomposition for knowledge graph embedding usage. Here, each entity e is represented as two vectors h_e and h_t instead of one single vector, and each relation r is represented as two vectors v_r and $v_{v^{-1}}$. For a triplet $(e_1$,r,$e_2)$, the similarity function is $\langle h_{e_1}, v_r, t_{e_2} \rangle$; while if we reconsider the reverse of this triplet $(e_2$, r^{-1}, $e1)$, the similarity function is $\langle h_{e_2}, v_{r^{-1}}, t_{e_1} \rangle$. And the translation function for two entities e_1 and e_2 related by relation r is defined as,

$$score_{SimplE} : \frac{1}{2}(\langle h_{e_1}, v_r, t_{e_2} \rangle + \langle h_{e_2}, v_{r^{-1}}, t_{e_1} \rangle) \tag{7.6}$$

which is the averaged similarities for $(e_1$,r,$e_2)$ and $(e_2$, r^{-1}, $e_1)$.

HolE [67] uses instead of multiplication operators a compositional operator - holographic operator which captures the idea of *circular correlation.* Given two vectors u and v, a holographic compositional operator $hol : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is defined as $hol(u, v) = u \star v = [h_1(u, v), ..., h_k(u, v), ..., h_d(u, v)]$ where,

$$h_k(u, v) = [u \star v]_k = \sum_{j=0}^{d-1} u_i v_{(i+k)\text{mod}d} \tag{7.7}$$

thus, the score function for a triplet (h, r, t) in holographic embedding is given as,

$$score_{HolE} : (v_h \star v_t) \cdot v_r \tag{7.8}$$

And in practice, the holographic operator can be efficiently calculated with fast Fourier Transform (FFT), denoted as $hol(u, v) = \mathcal{F}^{-1}(\overline{\mathcal{F}(u)} \odot \mathrm{F}(v))$, where $\odot$ represents the element-wise product, $\mathcal{F}$ represents the discrete Fourier transform and $\overline{(x)}$ represents the complex conjugate of x. Thus, the score function Eq. 7.8 can be rewritten as,

$$score_{HolE} : v_r^\top (\mathcal{F}^{-1}(\overline{\mathcal{F}(v_h)} \odot \mathcal{F}(v_t))) \tag{7.9}$$

,denoted as $hol(u, v) = \mathcal{F}^{-1}(\overline{\mathcal{F}(u)} \odot \mathrm{F}(v))$.

TuckerER [68] is another linear model which is based on Tucker decomposition. Given an original matrix $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, Tucker decomposition transforms the original matrix into a core tensor $\mathcal{Z} \in \mathbb{R}^{P \times Q \times R}$ and three other matrices $\mathbf{A} \in \mathbb{R}^{I \times P}$,$\mathbf{B} \in \mathbb{R}^{J \times Q}$, $\mathbf{C} \in \mathbb{R}^{K \times R}$, formalized as,

$$\mathcal{X} = \mathcal{Z} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \tag{7.10}$$

where $\times_1$,$\times_2$ and $\times_3$ are tensor products along the n-th mode, $n = 1, 2, 3$. The n-th mode tensor product of $\mathcal{Z}$ by $\mathbf{A}$(or $\mathbf{B}$ or $\mathbf{C}$) is denoted as $\mathcal{Z} \times_1 \mathbf{A}$ (or $\mathcal{Z} \times_2 \mathbf{B}$ or $\mathcal{Z} \times_3 \mathbf{C}$) with entries of each formulated as,

$$
\begin{aligned}
(\mathcal{Z} \times_1 \mathbf{A})(x_1, x_2, x_3) &= \sum_{i_1=1}^{P} \mathcal{Z}(i_1, x_2, x_3)\mathbf{A}(x_1, i_1) \\
(\mathcal{Z} \times_2 \mathbf{B})(x_1, x_2, x_3) &= \sum_{i_2=1}^{Q} \mathcal{Z}(x_1, i_2, x_3)\mathbf{B}(x_2, i_2) \\
(\mathcal{Z} \times_3 \mathbf{C})(x_1, x_2, x_3) &= \sum_{i_3=1}^{R} \mathcal{Z}(x_1, x_2, i_3)\mathbf{C}(x_3, i_3)
\end{aligned}
\tag{7.11}
$$

where $x_1, x_2, x_3 \in R$ are entry indexes of each dimension. The dimensions P, Q, R of core tensor $\mathcal{Z}$ are smaller than the dimensions I, J, K, thus Tucker decomposition can be considered a compressed version of X. Based on all these, TuckerER defines the score function for a triplet (h, r, t) as,

$$
score_{TuckerER} : \mathcal{W} \times_1 v_h \times_2 v_r \times_3 v_t \tag{7.12}
$$

where $v_h, v_t \in \mathbb{R}^{d_e}$, $v_r \in \mathbb{R}^{d_r}$, $\mathcal{W} \in \mathbb{R}^{d_e \times d_r \times d_e}$ and d_e and d_r are the dimensions of entity and relation vectors separately.

As for loss functions for the learning process, both penalized itemwise and pairwise loss functions are used for different tensor decomposition based knowledge graph embedding. For RESCAL, as the score function is formulated as 7.1, they employ a penalized squared loss represented as such,

$$
loss_{RESCAL} \doteq \sum_{k} \|\mathrm{Y_k} - V_E M_{r_k} V_E^{\top}\|_{\mathrm{F}}^2 + \lambda_1 \|V_E\|_{\mathrm{F}}^2 + \lambda_2 \sum_{k} \|M_{r_k}\|_{\mathrm{F}}^2 \tag{7.13}
$$

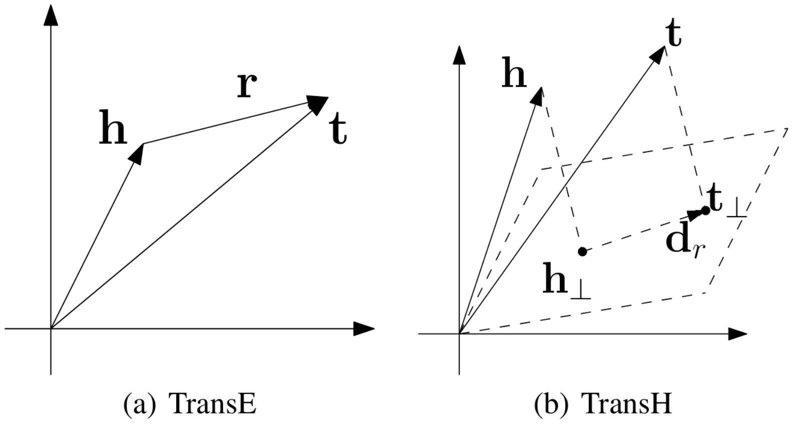
where λ_1 and λ_2 are the regulation parameters and Y_k is the label vector corresponding to the k-th relation.

ComplEx, SimplE, HolE and TuckerER share a same penalized loss function but they have employed a negative log-likelihood of the logistic model in their form. The general form of this loss function can be written as such,

$$
loss_{log_logistic} \doteq \sum_{(h,r,t)} \log(1 + \mathrm{e}^(-\mathrm{Y}_{h,r,t}\phi(h, r, t, \Theta))) + \lambda \|\Theta\|_2^2 \tag{7.14}
$$

where Θ represents all the learnable parameters (latent vectors) and $\phi(h, r, t, \Theta)$ is the score function for the triplet (h, r, t). λ is the regulation parameter. DistMult on the other hand uses a margin-based loss function shared with translation-based knowledge graph embedding methods. And we will give the margin-based loss function in the next section.

7.2 Translation Based Knowledge Graph Embedding



(a) TransE (b) TransH

Figure 7.2: Simple illustrations of TransE and TransH. Source: [15].

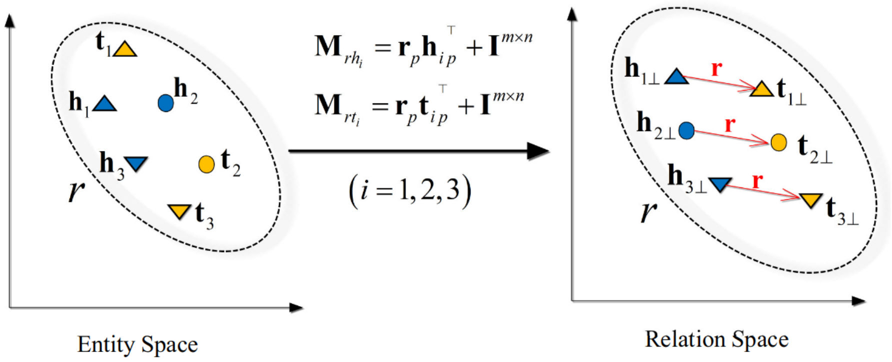


Figure 7.3: A simple illustration of TransD. Source: [69].

A *translation-based knowledge embedding* basically considers a semantic translation from one entity (head, h) to another entity (tail, t) by a specific relation r, i.e. $v_h + v_r \approx v_t$ when the triplet (h, r, t) holds. Different translation-based models [14, 15, 69, 70] vary in the projected low-dimensional continuous space of entities and relations.

TransE [14] maps both relations and entities lay on the same space and represents the translation function of a triplet (h, r, t) simply as,

$$translation_{TransE} : \|v_h + v_r - v_t\|_2 \tag{7.15}$$

where $v_h, v_r, v_t \in \mathbb{R}^d$ are latent vectors for h, r and t separately.

TransH [15] however projects the entities to a hyperplane by a norm vector $w_r \in \mathbb{R}^d$ for each type of relation first then considers the projected entities and original relations on the same latent space. For entities h and t, after being projected to a hyperplane noted by its direction vector w_r corresponding to the relation r, they are written as

$v_{h_\perp} = v_h - w_r^\top v_h w_r, v_{h_\perp} \in \mathbb{R}^d$ and $v_{t_\perp} = v_t - w_r^\top v_t w_r, v_{t_\perp} \in \mathbb{R}^d$ separately, thus the translation for TransH can be written as,

$$\begin{aligned} translation_{TransH} \doteq & \|v_{h_\perp} + v_r - v_{t_\perp}\|_2 \\ = & \|(v_h - w_r^\top v_h w_r) + v_r - (v_t - w_r^\top v_t w_r)\|_2 \end{aligned} \tag{7.16}$$

By mapping entities to a relation-related hyperplane, TransH improves performance compared to TransE over link prediction, node classification and other common knowledge embedding applied tasks, while the training parameter size is also slightly increased. In Figure 7.2 we give simple illustrations of TransE and TransH on latent space.

TransR [70] projects entities and relations to separate spaces through a relational projection matrix $M_r \in \mathbb{R}^{d\times d}$. Thus the translation for TransR can be written as,

$$translation_{TransR} : \|v_h M_r + v_r - v_t M_r\| \tag{7.17}$$

TransD [69] also projects entities to separate spaces but comparing to TransR, TransD uses a dynamic mapping matrix for each entity. That is to say, for each triplet (h, r, t), we also have entity-related project vector $h_p \in \mathbb{R}^d$ and t_p and relation-related project vector $r_p \in \mathbb{R}^d$ and the dynamic mapping matrices for h and t are defined correspondingly as,

$$M_{rh} = r_p h_p^\top + \mathrm{I} \tag{7.18}$$

$$M_{rt} = r_p t_p^\top + \mathrm{I} \tag{7.19}$$

where $\mathrm{I} \in \mathbb{R}^{d\times d}$ is the identity matrix having the same dimensions as $r_p h_p^\top$. And the translation function for TransD can be written as,

$$translation_{TransD} \doteq \|v_h M_{rh} + v_r - v_t M_{rt}\| \tag{7.20}$$

By comparing the two equations Eq.(7.17) and Eq.(7.20), there is high similarity in translation definition as entities h and r are both transformed through a mapping matrix before applying the simple translation form. However, the difference lies in the fact that in TransD, as the mapping matrix is dynamic and is decomposed into $r_p \in \mathbb{R}^d$ and $e_p \in \mathbb{R}^d$, the size of trainable parameter are greatly reduced compared to an unknown matrix $M_r \in \mathbb{R}^{d\times d}$. In Figure 7.3 we give a simple illustration of TransD, mapping from entity space to relation space.

TransE, TransH, TranR and TransD all have employed $\|\cdot\|_2$ a euclidean metric for the definition of translation function. TransA [71], however, argues that by replacing this oversimplified euclidean metric by a more flexible Mahalanobis distance, the translation of a triplet (h, r, t) can be expressed in a more adaptive way thus gaining in performance.

The translate function of TransA is given as such,

$$translation_{TransA} \doteq (\|v_h + v_r - v_t\|^\top) W_r (\|v_h + v_r - v_t\|) \tag{7.21}$$

where $W_r \in \mathbb{R}^{d \times d}$ is a relation-related non-negative weighted matrix corresponding to the adaptive distance.

In order to learn the entity/relation vectors and projection vector/matrix, translation-based knowledge graph embedding employs a margin-based pairwise loss function based on score function $f_r(h,t)$, which refers to different translation functions simulating the idea of $v_h + v_r \approx v_t$. The margin-based ranking loss is optimized to distinguish golden triplets $\mathcal{T}$ (those hold under historical interactions) and incorrect triplets $\mathcal{T}'$ (those unhold under historical interactions or unknown triplets), and is written as,

$$loss_{KGE} \doteq \sum_{(h,r,t)\in\mathcal{T}} \sum_{(h',r',t')\in\mathcal{T}'} [f_r(h,t) + \gamma - f_{r'}(h',t')]_+ \tag{7.22}$$

where (h', r', t') is a negative triplet degenerated from the golden triplet (h, r, t) and γ is the margin parameter. $[x]_+$ is absolute value of x.

Compared to tensor decomposition based knowledge graph embedding methods, translation-based knowledge graph embedding usually exhibit simpler forms and less complicated operations, while works have reported better performance of tensor decomposition based methods for complicated tasks (involving asymmetric relations and n-n relations).

7.3 Deep Neural Network Based Knowledge Graph Embedding

Deep neural networks have shown huge success in various machine learning (ML) domains, such as computer vision and natural language processing. Inspired by the successful applications in ML, works incorporating neural networks have been proposed by various researches for knowledge graph embedding learning. In neural network based knowledge graph embedding methods, entities and relations are still represented in latent vectors or tensors, while the score function for representing the relation of a triplet (h, r, t) is replaced with a neural network layer.

The Neural Tensor Network (NTN) [72] model proposes to use a bilinear tensor layer to relate two entity vectors before wrapping the unit in a non-linear activation function. And the score function of a triplet (h, r, t) is given as,

$$score_{NTN} \doteq u_r^\top \tanh(v_h^\top W_r^{[1:k]} v_t + V_r \begin{bmatrix} v_h \\ v_t \end{bmatrix} + b_r) \tag{7.23}$$

where $v_h, v_t \in \mathbb{R}^d$, $W_r^{[1:k]} \in \mathbb{R}^{d \times d \times k}$ is a tensor related to relation r, and $V_r \in \mathbb{R}^{k \times 2d}$,

$u_r \in \mathbb{R}^k$ and $b_r \in \mathbb{R}^k$ are all parameters related to relation r to express interaction of the entities through relation in the triplet.

Special types of deep neural networks have been developed in fascinating fashions and there is much spotlight on Convolutional Neural Networks (CNNs) particularly. Derived from CNNs, Graph Neural Networks (GCNs) [73] use multi-layer node embedding and spectral convolution on graphs to propagate the embedding between layers. The recent development in GCNs helps deep neural networks perform well in many tasks on graph structures. And it is natural to expand this line of work to knowledge graphs, and several works have been proposed for combining GCN in knowledge graph embedding.

In vanilla GCN, each layer l of neural networks contains all the entities representations, and from layer l to layer $l+1$, the propagation is given by the formula below,

$$v_e^{l+1} = \sigma(W^l(m_e^{l+1} + v_e^l)) \tag{7.24}$$

where v_e^l is the entity representation in layer l and v_e^{l+1} is the entity representation in layer $l+1$ for entity e. $\sigma(\cdot)$ is an activation function, and W_l is the model parameter for layer l. m_e^{l+1} is given by the formula as such,

$$m_e^{l+1} = \sum_{e' \in \mathcal{N}(e)} v_e^l \tag{7.25}$$

where $\mathcal{N}(\cdot)$ represents the neighbors (other entities with an edge with entity e in the original graph structure) of entity e and it represents the average of neighbor nodes of entities e in layer l.

Inspired by Eq.7.24, CompGCN [74] propagates both entity and relation in each layer by using a composition operator relating the entity and relation vectors. And the propagation from layer l to $l+1$ for an entity e is given as such,

$$v_e^{l+1} = \sigma(m_e^{l+1} + W_0^l v_e^l) \tag{7.26}$$

where m_e^{l+1} is related to relation r and written as,

$$m_e^{l+1} = \sum_{(e',r) \in \mathcal{N}_{in}(e)} W_r^l \phi_{in}(v_{e'}^l, v_r^l) + \sum_{(e',r) \in \mathcal{N}_{out}(e)} W_r^l \phi_{out}(v_{e'}^l, v_r^l) \tag{7.27}$$

where $\phi_{in}, phi_{out} : \mathbb{R}^{d_l} \times \mathbb{R}^{d_l} \rightarrow \mathbb{R}^{d_l}$ are the composition operators corresponding to entity e as tail or head in the triplet, and v_r^l is propagated as,

$$v_r^{l+1} = W_{rel}^l v_r^l \tag{7.28}$$

$W_0^l, W_r^l, W_{rel}^l \in \mathbb{R}^{d_l \times d_{l+1}}$ are all model parameters to transfer entity/relation vectors from layer l to $l+1$.

KE-GCN [75], compared to CompGCN, replaces the composition operators in m_e^{l+1} update to a differential form and the item propagate from layer l to $l+1$ is given as,

$$h_e^{l+1} = \sigma_{ent}(m_e^{l+1} + W_0^l v_e^l) \tag{7.29}$$

and m_e^{l+1} is formulated as,

$$m_e^{l+1} = \sum_{(e',r)\in\mathcal{N}_{in}} W_r^l \frac{\partial f_{in}(v_{e'}^l, v_r^l, v_e^l)}{\partial v_e^l} + \sum_{(e',r)\in\mathcal{N}_{out}} W_r^l \frac{\partial f_{out}(v_e^l, v_r^l, v_{e'}^l)}{\partial v_e^l} \tag{7.30}$$

where f_{in} and f_{out} are the score function used in non-DNN based knowledge graph embedding. In a similar fashion the relation vector is propagated from layer l to $l+1$ as such,

$$v_r^{l+1} = \sigma_{rel}(W_{rel}^l(m_r^{l+1} + v_r^l)) \tag{7.31}$$

and m_r^{l+1} is given as,

$$m_r^{l+1} = \sum_{(e,e')\in\mathcal{N}(r)} \frac{\partial f_r(v_e^l, v_r^l, v_{e'}^l)}{\partial v_r^l} \tag{7.32}$$

In terms of loss functions for deep neural network based knowledge graph embedding, both margin-based ranking loss function in Eq. (7.22) and log logistic loss function Eq. (7.14) are used for different tasks. We point out that in GCN-based knowledge graph embedding, the score function for a triplet (h, r, t) is simply given as,

$$f(h, r, t) = \|v_h - v_t\| \tag{7.33}$$

as v_h and v_t are updated at each layer through a relation-related form as in Eq.(7.29) and Eq.(7.31).

7.4 KGE-Based Recommendation

We have seen in previous sections different models of knowledge graph embedding methods, employing various score functions to capture structured knowledge information. Having witnessed the success in other ML tasks such as node classification and link prediction, and based on assumption that auxiliary information may be used to enhance the performance of recommendation, many works [18, 19, 76–79] have been dedicated to incorporate knowledge graph embedding into recommender systems.

CKE [18] leverages three types of information - structural content, textual content and visual content to jointly learn knowledge graph embedding and collaborative filtering representations. More particularly, for processing structural content composed of knowledge triplets $(head, relation, tail)$, the probabilistic translation-based knowledge graph embedding TransR [70] with a log logistic loss function is applied, which also helps the construction of item latent vector.

Deep knowledge-aware network (DKN) [76] is a content-based model which combines convolutional neural networks for textual information and knowledge graph embedding for structural contextual information forming a $multi-channel$ and $word-entity-aligned$ representation for click-through rate (CTR) prediction. DKN also applies an attention network [80] after the representation embedding layer for further enhancing the prediction performance while accentuating more influencing contexts which leads to a certain prediction result.

RippleNet [77] is also a CTR prediction which implements preference propagation on the knowledge graph based KGE methods. More specifically, RippleNet first extracts k-click ripple sets defined as the set of triplets starting from $(k-1)$-click reachable entities. Then similar to a content- based filtering model, the user representation v_u is the sum of all preferences reflected by k-click triplets and the predicted clicking probability of user u toward item i is given as,

$$\hat{y}_{u,i} = \text{sigmoid}(v_u^\top v_i) \tag{7.34}$$

RCF [19] departs from item-based collaborative filtering (ICF) methods and jointly learns a user-item preference model and an item-item relational data model. And in user-item preference model, they propose a two-level hierarchy attention mechanism to capture the interactions between user embedding and relation types and the weights between users and historical items with specific relation values. The loss function for this ICF-based user-item preference model is given as,

$$loss_{rcf-rec} \doteq -\sum_{(u,i,k)\in\mathcal{D_I}} \ln \text{sigmoid}(MLP(m_{u,i} \odot q_i) - MLP(m_{u,k} \odot q_j)) \tag{7.35}$$

where MLP is a multilayer perceptron (MLP), $\mathcal{D_I}$ is the triplet of user u, positive item i and negative item j, q_i and q_j are item embedding vectors for item i and item j separately. $m_{u,i}$ and $m_{u,j}$ are user embedding vectors with the two-level hierarchy attention mechanism. And we give the formulation of the target-aware user embedding $m_{u,i}$ as such,

$$m_{u,i} = p_u + \sum_{t\in\mathcal{T}} \text{softmax}(a(p_u, x_t)) \cdot s_{u,i}^t \tag{7.36}$$

where $\text{softmax}(a(p_u, x_t))$ is the first-level attention between user embedding p_u and relation type embedding x_t; and $s_{u,i}^t$ is the second-level attention between user and items in terms of specific relation values for each relation type. Besides, the item-item relational data models treats item auxiliary information as knowledge graph and applies DistMult [16] for knowledge graph embedding. To link the user-item preference model and the item-item relational model together, the relation embedding vector v_r in item-item relational model for relation type r is the combination of the relation type latent vector x_t and relation value vector z_v in user-item preference model and is written as $v_r = x_t + z_v$. And the loss function of the item-item relational model is given as,

$$loss_{rcf-rel} \doteq \sum_{(h,r,t,t') \in \mathcal{D}_{\mathcal{R}}} \ln \text{sigmoid}(f_r(h,t) - f_r(h,t')) \tag{7.37}$$

where $\mathcal{D}_{\mathcal{R}}$ denotes a pair of triplets where (h, r, t) is the golden triplet in knowledge graph and (h, r, t') is the corresponding negative triplet.

BEM [78] is an Bayesian approach of learning representations integrating knowledge graphs and behavior graphs for various knowledge graph tasks and item recommendation. BEM pre-trains separate embeddings for a knowledge graph (KG) and a behavior graph (BG) of a dataset with translation-based KGE methods, and then uses a bayesian framework to adjust the representations for both KG and BG representations. KGNN-LS [79] proposes a *Label Smoothness regularization* method to incorporate GCN-based knowledge graph embedding into recommendation.

8 Diversity-Aware Information Retrieval and Recommendation

Information retrieval (IR) [81] is a process to retrieve documents or information relevant to the demand, in forms of queries. A query is typically a piece of information (in texts, for example) given by the user to represent the characteristics or topics of potential information. Web search engines is one of the most common applications of information retrieval service on the Internet. Similar to recommender systems, the quality of rendering only relevant documents cannot fully achieve users' satisfaction due to the fact that the interpretation of a query can be vague and ambiguous. And focusing on maximizing one way of calculating the relevance between a query and the potential documents can lead to misunderstanding of users' intent or exposing too much redundant information. Thus, result diversification has been brought to IR to deal with this problem.

Recommendation problems, however, are highly similar to information retrieval in terms of goals both should attain to. Although in recommender systems, an explicit query in text or other formats is not given, it is expected to retrieve relevant items for each user by inferring from historical user behavior data and other available information. And as briefly presented in introduction, over-concentrating on predicting and returning relevant items deviates from the goal of maximizing user satisfactions. So it is essential to take diversity into account for achieving a better recommendation. In reality, due to a highly resembling goal shared by IR and recommendation, various methods for diversification in both areas have inspired one the other.

Because of the nature and their similarity of diversity related topics in information retrieval and recommendation, we present an overall view of works dedicated in diversity-aware IR/recommendation in this section.

The roadmap of this section is that we will present the concept of diversity and its various definitions from different perspective. Then we will present various diversification methods proposed in existed works.

8.1 Diversity

Diversity has been brought up in information retrieval and recommender systems for over a decade due to various limits and issues reported in research works [20–22, 24]. Despite the importance of diversity in IR and recommendation tasks, the definition of diversity in existent literature varies. As a matter of fact, the concept of diversity only refers to retrieving heterogeneous information or items [82], which leaves huge flexibility to researchers in academics and industry to define diversity differently according to the configuration required in various scenarios.

From the perspective of diversifying scale, diversity can be considered in either aggregated level or individual level. **Aggregate diversity** [25–27] targets at improving a more balanced recommendation over all items towards users and alleviating the *long-tail effect* [6, 28] as much as possible. The long-tail effect is a distribution with high occurrences that are far from the concentrated "head" and those high occurrences appear to be a long tail for such a distribution. And in Figure 8.1 we give an example of such effect. The latter actually reflects a systematic bias towards a huge quantity of items in IR or recommender systems and it is especially critical in retail business, where the aggregate diversity is named as **sales diversity**.

In contrast, **individual diversity** aims at providing a diversified personalized item list to each user to enhance personal experience and satisfactions of users in recommender systems. We focus on individual diversity in recommender systems in this thesis, thus we use simply the term diversity to refer to individual diversity in the following text.

A widely used definition of diversity in recommender systems and other information retrieval relies on semantic information, i.e. item categories [83–85] or tags/topics [22, 86–89]. The former assumes there is an existing taxonomy of information related to the retrieval/recommendation candidates, which serves to define or measure the returning diversity of results through relevance-related metrics. And the latter tends to use language

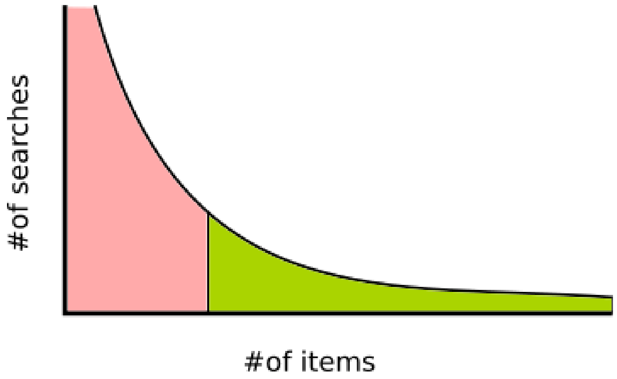


Figure 8.1: An example of long-tail effect distribution chart. The horizontal axis represents the number of items in this distribution; and the vertical axis represents the occurrences of searches for items. The pink area on the left is the dominant "head" of the distribution while the green area shows a long tail of huge number of items with low searches.

models (such as PLSA [37], LDA [38]) to represent words or latent topics and based on that, the methods can use similarity/dissimilarity metrics for diversity purpose.

In [83] the diversity corresponds to various intents of a given query, and is based on categorical information. Thus the measurement of result diversity relies on the probability of intent (category) of a query and the intent-dependent quality measure (NDCG,MRR,MAP), formulated as,

$$\begin{aligned} \mathrm{NDCG} - IA(D,n) &= \sum_c P(c|q)\mathrm{NDCG}(D,n|c) \\ \mathrm{MRR} - IA(D,n) &= \sum_c P(c|q)\mathrm{MRR}(D,n|c) \\ \mathrm{MAP} - IA(D,n) &= \sum_c P(c|q)\mathrm{MAP}(D,n|c) \end{aligned} \tag{8.1}$$

where D is a list of document, n the cut-off number and c is a category or intent.

[84] also proposes a diversity measure based on categorical information. Inspired by binomial distribution, this diversity metric considers for each item i with its categories $C(i)$ and a randomly sampled category c, whether c belongs to $C(i)$ as an independent Bernoulli trial. And they model the probability of how adequate a genre c is covered in a top-N recommendation list, denoted as p_c or $P(X_c)$. Thus, they propose $BinomDiv$ composed of two parts: $Coverage(\cdot)$ and $NonRed(\cdot)$, formulated separately as,

$$Coverage(D) = \prod_{c \notin C(D)} P(X_c = 0)^{1/|C|} \tag{8.2}$$

$$NonRed(D) = \prod_{c \in C(D)} P(X_c >= k_c^R | X_c > 0)^{1/C(D)} \tag{8.3}$$

And finally *BinomDiv* is given as,

$$BinomDiv(D) = Coverage(D) \cdot NonRed(D) \tag{8.4}$$

Their diversity measure given in Eq.(8.4) considers both categorical coverage and redundancy, and measures the genre diversity of the recommended items assuming that the recommendation is the most diverse when its coverage of categories is proportional to categorical popularity in the system. In [85], they define their diversity function directly on categories, denoted as function f, given as,

$$f(D) = \sum_{c \in C} \mathbb{1}\left\{\exists d \in D, d \text{ covers } c\right\} \tag{8.5}$$

In [87], the diversity measurement is not directly defined based on topical information. Rather, the item list diversity is evaluated as a redundancy estimation regarding the pre-selected relevant items, where the estimation is based on three language weighting methods. Given a query q and a set of pre-selected items $\overline{D}$, the item list diversity at this iterative step for a document d is defined as,

$$\begin{aligned} P(d, \overline{D}|q) &= \sum_{q_i \in Q} P(q_i|q) P(d, \overline{D}|q_i) \\ &= \sum_{q_i \in Q} P(q_i|q) P(d|q_i) P(\overline{D}|q_i) \end{aligned} \tag{8.6}$$

where q_i is the sub-query of query q, Q is the collection of sub-queries corresponding to the original query q, and $P(d|q_i)$ measures the coverage of document d with regard to sub-query q_i and $P(\overline{D}|q_i)$ measures the novelty of the document d regarding q_i.

[90] also does not define an explicit diversity metric but a utility function for a candidate item/document d over the original ranking results R_q in terms of sub-queries, formulated as,

$$U(d|R_{q_i}) = \sum_{d' \in R_{q_i}} \frac{1 - cosine(d, d')}{rank(d', R_{q_i})} \tag{8.7}$$

where q_i is a specialized query (sub-query) of the original query q, and R_{q_i} corresponds to the ranking of documents given the sub-query q_i. And the discounted factor $1 - cosine(d, d')$ is the diversity consideration in this utility function for each new document d.

[88] proposes to combine latent topics of textual information learnt from *latent dirichlet analysis (LDA)* [38] and category information as auxiliary information to build user profiles in question recommendation. The user profile actually is represented as a three level probability distribution tree, namely a $top - category - distribution$ level, a $model - distribution$ level and a $feature - distribution$ level. The branches of such a tree structure correspond to the diverse intent recorded in the user profiles. In measuring the diversity

both categorical information and LDA topics are considered to be covered.

[89] considers the diversity between a new document x_i and pre-selected documents S as a relational function $h_S(\cdot)$. This relational function takes various diversity feature vectors Div_{ij} as input, and can be defined as the different distances between the new document x_i and x_j in pre-selected documents S. The different distances are namely Minimal Distance, Average Distance and Maximal Distance, formulated correspondingly as,

$$h_S(x_i) = (\min_{x_j \in S} Div_{ij1}, ..., \min_{x_j \in S} Div_{ijl}) \tag{8.8}$$

$$h_S(x_i) = (\mathrm{avg}_{x_j \in S} Div_{ij1}, ..., \mathrm{avg}_{x_j \in S} Div_{ijl}) \tag{8.9}$$

$$h_S(x_i) = (\max_{x_j \in S} Div_{ij1}, ..., \max_{x_j \in S} Div_{ijl}) \tag{8.10}$$

where $Div_{ijk}, k \in \{1, ..., l\}$ are diversity feature vectors incorporating semantic diversity information. These various pairwise semantic diversities include pairwise euclidean distance based subtopic diversity given latent subtopic vectors learnt from *Probabilistic Latent Semantic Analysis (PLSA)*, formulated as,

$$Div_{ij}^{topic} = \sqrt{\sum_{k=1}^{m} (p(z_k|x_i) - p(z_k|x_j))^2} \tag{8.11}$$

where $z_k, k \in \{1, .., m\}$ are the latent topics learnt from PLSA, and x_i and x_j are different documents. They also include pairwise cosine dissimilarity text diversity based on weighted term vector representations (TF-IDF), formulated as,

$$Div_{ij}^{text} = 1 - \frac{d_i \cdot d_j}{\|d_i\| \|d_j\|} \tag{8.12}$$

where d_i and d_j are the TF-IDF document vectors for documents x_i and x_j.

However, [91] points out semantic information may be incomplete and thus unreliable, leading to exploit latent item vectors learnt from explicit features (ratings etc.) or implicit features (clicking, viewing etc.) to measure diversity of the returned list. It is argued that latent vectors learnt from these features reflect the underlying features of items thus the more orthogonal two latent vectors are the more dissimilar they are.

8.2 Diversification

Traditional IR/recommendation tasks have concentrated on estimating a precise relevance or affinity of the documents and candidate items, who will cause redundancy and dissatisfaction of the rendered results. Thus, it is essentially to apply diversification in

IR/recommendation tasks for a both accurate and diverse result. As we have discussed in previous section (8.1), diversity does not have a universal definition and thus various diversification methods may be possible according to a specific setting of diversity. Corresponding to various diversities in information retrieval problems, various diversification methods [29, 30, 87, 91] have been proposed. In this section, we first present query reformulation based diversification whose main idea is to unravel user's different intents for diversified results. Then we introduce post-processing re-ranking methods which consider an optimization of linearly combined accuracy and diversity function for achieving both accuracy and diversity. Finally we present diversification models which in models themselves have combined accuracy and diversity in an intelligent way.

8.2.1 Query Disambiguation for Diversification

In information search scenarios, search result diversification targets query disambiguation, leading to query reformulation methods for diversification [87], and intent-based [83, 92, 93] diversification. Query reformulation is a process to generate representations of query aspects in the form of sub-queries [87], and there exist different techniques for this process. The term *aspect* refers to different interpretations of ambiguous queries, which can be extracted from each query. The term *intent* refers to the type of a query / the purpose of a user, informational or navigational for example. xQuAD [87] defines for a candidate document d the diversity score shown in Eq. (8.6). The original query q is reformulated into $Q = \{q_i | i \in |Q|\}$. The xQuAD framework returns a diversified document list by iteratively selecting from the candidate documents a document which can maximize the mixture of relevance and diversity estimation. However, in case of recommendation, *intent* or *aspect* of each user are not explicitly available.

8.2.2 Post-processing Re-ranking for Diversification

Different post-processing diversification techniques [29, 30, 91] have been proposed in information retrieval tasks to re-rank the documents. MMR (Maximal Marginal Relevance) [29] defines a marginal relevance that linearly combines the relevance and accumulated dissimilarity, and maximizes this marginal relevance to diversify the ranking by taking the form,

$$\omega_i^* = arg \max_{\omega_i \subseteq X \setminus S} [\lambda Sim_1(\omega_u, \omega_i) - (1-\lambda) \max_{\omega_j \in S} Sim_2(\omega_i, \omega_j)] \tag{8.13}$$

where S is the subset of already selected items, λ is the parameter to adjust the trade-off between relevance and diversity. Sim_1 measures the relevance between the item and the user and Sim_2 measures the similarity between two items. And λ adjusts the proportion of relevance and accumulated dissimilarity, giving a standard recommendation list when $\lambda = 1$ and a purely diversified list when $\lambda = 0$.

Maxsum [30] describes the diversification problem as to optimize the linear combination of a submodular quality function $f(S)$ over the result list S and the max-sum dispersion $\sum_{u,v \in S} d(u,v)$, which can be formulated as

$$arg \max \Big(f(S) + \lambda \sum_{\{u,v\}:u,v \in S} d(u,v) \Big) \tag{8.14}$$

subject to $\mid S \mid = p$, with $d(\cdot,\cdot)$ a metric distance function on item sets. Then they propose one greedy-based solution and one local search solution for this task with an approximation ratio of 2.

Entropy [91] takes probabilistic matrix factorization (PMF) method as base algorithm and assumes user latent vectors and item latent vectors both have the Gaussian distributions. And they propose to takes a linear form combining the quality measuring of the recommendation list and diversity promoting regularizer defined by different entropy of the assumed distribution of ratings, noted as,

$$arg \max_{S:|S|<K} f(S) \equiv R(S) + \lambda g(S) \tag{8.15}$$

where $R(S)$ and $g(S)$ are separately represented as,

$$R(S) \equiv \sum_{\omega \in S} (\mathbb{E}[r_\omega \mid r_\Omega, V] - c) \tag{8.16}$$

$$g(S) = h(r_S \mid r_\Omega, V) = \frac{1}{2} \mid S \mid log(2\pi e) + \frac{1}{2} logdet(\Sigma_S) \tag{8.17}$$

where Ω the set of rated items of the user and S the unrated items, r_ω predicted rating of the item, Σ the covariance matrix of items, V the item feature matrix and c a constant. The proposed differential entropy $g(S)$ is submodular and will be maximized when item latent vectors are orthogonal to each other (if that is possible).

8.2.3 Diversification Models

[83] presumes the existence of a taxonomy information with both queries and documents categorized using this information. They denote the set of categories belonging to a query q as $C(q)$ and that belonging to a document d as $C(d)$. Besides they also presume knowing the distribution of a given query belonging to a given category, denoted as $P(c|q)$ and the likelihood of a document d satisfying the intent c for the query q as $V(d|q,c)$. Based on these presumptions they define the top-N diversification problem in a probabilistic form over a set of documents D for a query q as such,

$$P(S|q) = \sum_c P(c|q)(1 - \prod_{d \in S} (1 - V(d|q,c))) \tag{8.18}$$

where S is the top$N$ diversified result and $|S| = N$.

Recently, determinantal point processes [94] (DPPs) have been widely applied to various diversification tasks in machine learning. DPPs are equipped with a positive semi-definite kernel matrix L. Following Kuelsza et al. [94], this kernel matrix can be decomposed as Gram matrix, with each entry denoted as $L_{ij} = q_i\phi_i^\top\phi_j q_j$, where q_i and q_j measure the quality of item i and j and ϕ_i and ϕ_j are normalized feature vectors to help measure the similarity of items by applying $\phi_i^\top\phi_j$. However, in personalized recommendation settings, the quality of items varies from one user to the other and thus the normalized feature vectors of items may not directly be deduced from CF results.

PDGAN [95] is a method that combines the diversification model DPP with *generative adversarial network (GAN)*. However the base model for the discriminator and generator model in GAN framework is based on a matrix factorization methods (PMF [47]) which only considers the user-item interactions, therefore the performance of PDGAN does not benefit from auxiliary information for better accuracy.

Part III

Knowledge Graph Embedding Based Diversified Recommendations

9 Introduction **52**

10 Background **54**

10.1 Diversity Metrics for Recommendation 54

10.2 Determinantal Point Processes 57

10.3 Translation-Based Knowledge Graph Embedding for Recommendation . . . 58

11 DivKG: Improving Diversity for Recommendation Based on Knowledge Graphs **59**

11.1 Knowledge Graph Embedding for Entity and Relation Representations . . 62

11.2 Diversity-Aware Recommendation on Determinantal Point Processes 63

12 EMDKG: Diversity-Aware Representation Learning for a Better Trade-off Diversified Recommendation **65**

12.1 General Overview 66

12.2 Diverse Item Sets Generation 67

12.3 Item Diversity Learning. 68

12.4 Modified Translation-Based Knowledge Graph Embedding for Recommendation. 70

12.5 Prediction 72

13 Conclusion **72**

9 Introduction

Recommender systems (RS) have been intensively studied over the last three decades and have reached a remarkable effectiveness. Amazon, Netflix, Facebook: all these applications and e-commerce sites make a very intensive use of recommender systems. Top-N recommender systems, as one of main recommendation applications, exploiting user-item interactions have achieved amazing recommendation results.

Top-N recommendation aims at selecting a set of N items that represents the highest relevance to a user. Among various kinds of recommendation techniques, collaborative filtering methods (CF), in particular model-based matrix factorization methods such as [47, 96], have been widely suggested due to their predictive power in terms of accuracy. They make use of user-item interactions in order to determine item relevance towards users, which in general generate rather accurate predictions.

However, the use of only one type of relations (user-item interaction) lacks explicit semantics and implies the search for latent user-item relations. Besides, apart from user-item interactions (among which rating is commonly used), there exist various relations between items and other entities that could be helpful for a better understanding of the users' behaviour. All these relationships can be modeled as a graph structure that provides richer information about the users, items, and their interactions. Mind that this graph may also contain the direct user-item interaction as one of its relations and therefore, can be seen as an extension of CF model.

Furthermore, knowledge graph embedding methods [14, 15, 69, 70], naturally capturing and conserving different types of relations among various kinds of entities including users, items and others, can provide a promising model for this purpose. F. Zhang et *al.* propose the framework CKE [18] incorporating one translation-based embedding method Bayesian TransR for recommendation. X. Xin et *al.* [19] propose a two-layer relational collaborative filtering method RCF to exploit knowledge graph embedding for top-N recommendation. Both of them have revealed improved performance of recommendation accuracy due to exploiting structural information on knowledge graphs.

However, accuracy should not be the ultimate goal of the recommendation task as it results in returning to the user highly similar items, ignoring the relations between them, and finally, decreasing user's satisfaction with the provided service. For example, in E-commerce, after detecting a user's interest in laptops, a recommender system returning a list purely composed of laptops is inefficient as the user is very unlikely to purchase more than one model at a time. In movie recommendation, users may get bored after watching several theme-alike movies sequentially. Thus, a returned list of items should be diverse enough, implying both, redundancy reduction and novelty increase.

Despite the importance of diversity, it has received much less attention than accuracy in top-N recommendation domain. Carbonell and Goldstein provide a diversification method Maximal Marginal Relevance (MMR) [29] for re-ordering the items through an iterative process by selecting the most dissimilar item to the existent item list. Further, A. Borodin *et al.* [30] provide its extended version and redefine the problem as max sum diversification problem in order to give a theoretically provable solution. However, these diversification solutions adopt pairwise similarity which tends to be sub-optimal as it ignores the correlation within the item list.

A recent emergence of *determinantal point processes* (DPPs) brings new potential to enhancing diversity in multiple machine learning problems, such as extractive summarization [32] and basket completion [97,98]. DPPs are probabilistic models of sets parameterized with positive semi-definite matrix which characterizes naturally both element-wise relevance towards a query/user and the repulsiveness of subsets on the total candidate item set. DPP captures item similarity in an unified feature space and propose list-wise dissimilar items. Thus, using DPP in recommendation models improves their results in terms of diversity. The challenge here is to find an efficient way to construct the positive semi-definite kernel matrix in order to balance relevance and diversity of items.

Though, a bunch of work have been done recently to address the problem of diversity in recommendation (e.g. [29, 84, 87, 92, 95, 99–101]), achieving a good accuracy-diversity trade-off is still an open challenge. Thus, most of existing works confront with this trade-off assuming submodular feature of optimisation function. A few works [95, 102, 103] try to find solutions both diverse and accurate. However, there is still a lack of the understanding of a good trade-off between accuracy and diversity and when and how a better trade-off can be achieved. In our opinion, a diversity-aware recommendation algorithm should not only achieve both high accuracy and diversity, but also be robust under different parameter settings.

To address the accuracy-diversity trade-off problem, we propose two models: **DivKG** and **EMDKG** respectively in Chapter 11 and Chapter 12. DivKG is, to our best knowledge, the first method that combines knowledge graph embedding with determinantal point processes for a diversified recommendation. The combination of the two approaches KGE and DPP, we propose a new approach to construct the personalized DPP kernel matrix based on KGE results. EMDKG is another approach for diversified recommendation, which jointly optimizes an item diversity learning and knowledge graph embedding for obtaining better representations to capture semantic diversity.

The roadmap of the rest of this part is described as such. We first introduce the background of our proposals in Chapter 10. Then in Chapter 11 and Chapter 12 we respectively detail our two diversified recommendation models DivKG and EMDKG. And finally in Chapter 13 we conlude this part.

10 Background

The goal of our recommendation task is two-fold: given a set of users U, a set of items I, the user-item interactions R_0, item-side auxiliary information $R_i, i \in \{1, 2, ...\}$ the top-N recommendation should provide each user $u \in U$ with **relevant** yet **diverse** recommendations. As we have presented in Part II, diversity metrics play a fundamental part in defining our problem since the definition of the diversity itself usually lies on a diversity metric. Moreover, both our solutions **DivKG** and **EMDKG** lie on two ideas: Determinantal Point Processes and Translation-Based Knowledge Graph Embedding. Thus, to better introduce our models, first we present briefly these three parts in this section to provide the background.

10.1 Diversity Metrics for Recommendation

A **diverse** top-N recommendation means providing each user with a **diverse** list of N items, which requires a measurement of how diverse a given item list is. Two points are worthy noticing for the choice of this measurement: (1) in the state-of-the-art recommendation algorithms [19, 47–50], users and items are usually represented in continuous vector space or embedded space; thus the measurement of a item list should be more convenient to be operated on latent vectors in practice. (2) for users and more generally human beings, diversity of an item list is easier to understand when it is equipped with human understandable features or semantic information, namely categories, topics, co-activities etc. So it makes more senses to utilise the semantic information for providing diverse solutions.

Regarding the first point, as items are represented as latent vectors, it is necessary to define diversity measurement $Div(\cdot) : R^N \to R$ for a list of vectors $V = \{v_1, ..., v_N\}$. Different metrics for this have been proposed in existing works. One type of diversity measures for a list of vectors is inspired by pairwise vector diversity (dissimilarity). The definitions of pairwise similarity and pairwise diversity are then given as follows.

Definition 2. (Pairwise Similarity) A pairwise similarity Sim_{pair} is a real-valued function defining the similarity of two vectors. And it is usually related to the inverse of distance in vectorial space.

Definition 3. (Pairwise Diversity) A pairwise diversity is a real-valued function defining the diversity of two vectors. When measuring on a list of two items, the diversity equals to the dissimilarity or the inverse of pairwise similarity or distance in vectorial space.

More particularly, common pairwise similarity metrics include inner product and cosine similarity. And we formalize the two similarity metrics given two vectors v_i and v_j as such:

$$inner\_product : \langle v_i, v_j \rangle = v_i^\top \cdot v_j = \sum_{d=1}^{k} v_i^d * v_j^d \tag{10.1}$$

$$cosine\_similarity : cosine(v_i, v_j) = \frac{\langle v_i, v_j \rangle}{\|v_i\| * \|v_j\|} \tag{10.2}$$

As the pairwise diversity is the inverse pairwise similarity, it is apparent to choose $-Sim_{pair}$ for the convenience. Based on pairwise diversity, we can define diversity metric for a list of item vectors, named as pairwise based Intra-List Diversity (ILD), formulated as,

$$\text{ILD} : \sum_{i \in L} \sum_{j \in L \& j \neq i} -Sim_{pair}(v_i, v_j) \tag{10.3}$$

where L is the list of items.

Another category of vector diversity is directly listwise and based on the determinant of square matrix $V^\top V$, where $V = [v_1, ...nv_L]$ is the vector of item vectors.

$$\begin{aligned} Div_{det} : det(V^\top V) &= det\left(\begin{bmatrix} v_1 \cdot v_1 & ... & v_1 \cdot v_L \\ ... & & ... \\ v_L \cdot v_1 & ... & v_L \cdot v_L \end{bmatrix} \right) \\ &= def\left(\begin{bmatrix} \|v_1\|_2 & ... & sim(v_1, v_L) \\ ... & & ... \\ sim(v_L, v_1) & ... & \|v_L\|_2 \end{bmatrix} \right) \end{aligned} \tag{10.4}$$

The form of determinant of the item representation square matrix corresponds to a rather complicated relation in terms of vector pairwise similarity. And such a form actually can consider more than pairwise dissimilarity but also multi-item diversity (3, 4, up to $L - 1$).

On the other hand, to render diversity measurement more human comprehensible, diversity metrics relying on semantic information can be defined. The semantic information of items can be explicitly categories of items, labels or tags given to the items or implicitly a co-activity of two items [104]. Take movies as an example, genres or textual tags of movies are explicit features, while movies co-played or co-liked are implicit characteristics to evaluate the similar the movies. It is admittedly more straightforward, however, to consider the explicit features as rule of thumb for item semantic diversity. Thus we formalize the explicit features here. To be noted, different terms have been designated to refer item features, i.e. topics and categories, here we use these terms interchangeably.

Definition 4. (Item feature) An *item feature* f is an explicit description unit for an item i.

Two different item features may have correlations or not, but we can calculate the diversity based on item features (or **semantic diversity**) of an item list given each item's item feature information. And the semantic diversity is both motivation and evaluation criteria to achieve diversified recommendation.

Definition 5. (Item feature set) Given a ground feature set $\mathcal{F} = \{f_1, f_2, ..., f_M\}$ where the count of features is M, an item feature set is a subset of F for item i, denoted as $F_i \subseteq F$, meaning the item i possessing each feature $f \in F_i$.

Definition 6. (Semantic diversity) A semantic diversity is a quantitive measure over a set of N items $I = \{i_1, i_2, ..., i_N\}$ equipped with item features $\mathbf{F} = \{F_i | i \in I\}$, denoted as $div(\cdot)$. F_i is the i-th item's feature set.

Definition 6 gives a general concept and denotation but specific formulations of how to use item feature sets to calculate a given item list should be defined. Definition 7 is a list-wise measure of semantic diversity, calculating the percentage of categories covered by an item list.

Definition 7. (Category Coverage, CC) Given a ground item feature set $\mathcal{F}$ and a list of items I equipped with their item feature list $F_I = \{F_i | i \in I\}$, the category coverage (CC) is formulated as,

$$\text{CC} : \frac{\sum_{f \in \mathcal{F}} \mathbb{1}_f(\cup F_I)}{|\mathcal{F}|} \tag{10.5}$$

where, $|\cdot|$ represents the cardinality of a set and $\mathbb{1}$ is an indicator function.

Category Coverage as a diversity metric, however, only consider the presence of the features, but does not consider the repetitiveness or the position of the item in the list. Thus diversity metrics such as α-NDCG and DNG are brought to discount the repeating categories. α-NDCG@n [105] is formulated as,

$$\alpha\text{-NDCG@}n = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{\alpha\text{DCG}_u\text{@}n}{\alpha\text{IDCG}_u\text{@}n} \tag{10.6}$$

in which

$$\alpha\text{DCG}_u\text{@}n = \sum_{k=1}^{n} \frac{\sum_{l=1}^{L} J_{kl}^{u} (1-\alpha)^{q_{l,k-1}^{u}}}{log_2(1+k)} \tag{10.7}$$

where J^u_{kl} equals to the rating of the kth item in the list for user u if kth item belongs to the genre l otherwise 0. $q^u_{l,k-1}$ counts the number of items belonging to genre l up to the $k-1$ position in the list, which accompanying the constant α to modify the redundancy in the recommendation list. $\alpha\mathrm{IDCG}_u@n$ denotes the largest value of $\alpha\mathrm{DCG}_u@n$ which achieves the ideal diversification of recommendation lists.

And DNG [106] also takes a discounted factor for categories and is formulated as,

$$\mathrm{DNG}@N = \sum_{k=1}^{N} w_k G(k) \tag{10.8}$$

where w_k is a discount factor taking $w_k = 1/2^{k-1}$ and $G(k)$ denotes the number of categories of the kth item has while the first $k-1$ items don't.

Another category of diversity **Intra-List Diversity** measures base on pair-wise distance of any two items in the item list. Here Intra-List Diversity resembles tremendeously to the PB-ILD for vector diversity, but here distance function inside ILD is a function defined on binary representations, instead of real-values vectors.

Definition 8. (Intra-List Diversity, ILD) Given a list of items I and a pair-wise distance function $dist(\cdot,\cdot)$ defined over items, the intra-list diversity is formulated as,

$$\mathrm{ILD}_I = \sum_{i\in I} \sum_{j\in I \wedge i\neq j} dist(i,j) \tag{10.9}$$

10.2 Determinantal Point Processes

A determinantal point process (DPP) is a probabilistic model over selection of points. Originating from quantum physics, this model is characterized by its repulsiveness, which means a higher probability of a subset selection associates with more repelling points to each other in the subset [32]. DPP $\mathcal{P}$ over a discrete point set $\Omega = \{\omega_1, \omega_2, .., \omega_M\}$ is determined by a $M \times M$ positive semi-definite matrix L indexed by the elements from Ω and defining the probability of point selection. In our case, Ω is the set of items.

Given a discrete point set Ω, *a determinantal point process* $\mathcal{P}$ is a probability measure defined on 2^Ω, the set of all subsets of Ω, such that if $\mathsf{A} \sim \mathcal{P}$ is a random subset, then we get:

$$\mathcal{P}(\mathsf{A} = A) \propto \det(L_A) \tag{10.10}$$

where $L_A \equiv [L_{ij}]_{\omega_i,\omega_j \in A}$. The diagonal elements of L provide the probabilities of selecting individual items from Ω ($\mathcal{P}(\omega_i) \in W, i = 1, ...M$), while the off-diagonal elements of L reflect the negative correlations between item pairs. The larger the values of L_{ij}, the smaller the tendency of ω_i and ω_j to co-occur. The determinants of entries L_{ij} can be viewed as measurements of the similarity between ω_i and ω_j. Therefore, more similar items are less likely to get selected together. In the RS context , the diagonal elements L_{ii} can be seen as user's affinities towards an item ω_i.

When given the kernel matrix L for the determinantal processes, we can apply DPP sampling methods [100,107] based on kernel matrix to retrieve diversified top-N recommendation list. Thus, in RS context,the main task of using DPP for diversified recommendation goes to the construction or the learning of the personalized kernel matrix.

For the learning of kernel matrix, we consider a pairwise loss function between selected and all remaining items, and uniform sampling under the framework of Bayesian Personalized Ranking (BPR) [48]. For instance, given a user $u \in U$, the set of items I, the BPR loss function for the parameter vector of an arbitrary model class Θ is defined as:

$$\mathcal{L}_{BPR} = \sum_{(u,i,j) \in D_T} \ln \sigma(\hat{x}_{uij}) - \lambda_\Theta ||\Theta||^2 \tag{10.11}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the logistic sigmoid function, $\hat{x}_{uij}(\Theta)$ is a model specific function estimating a real value of preferences of user u and items i and j, λ_Θ are model specific regulation parameters, $D_T : U \times I \times I$, s.t.

$$D_S = \left\{(u, i, j) | i \in I_u^+ \wedge j \in I \setminus I_u^+\right\} \tag{10.12}$$

, and $(u, i, j) \in D_S$ denotes that the user u prefers the item i over j, $T \subseteq U \times I$ being the available user-item interactions and $I_u^+ = \{i \in I : (u, i) \in T\}$. The estimator $\hat{x}_{uij}$ is decomposed as $\hat{x}_{uij} = \hat{x}_{ui} - \hat{x}_{uj}$. We will specify the function used for optimisation in Section 12.3.

10.3 Translation-Based Knowledge Graph Embedding for Recommendation

A knowledge graph (KG) is a multi-relational graph $(V, Edges, R)$ consisting of nodes $v \in V$, i.e. *entities* such as users, items, genres, actors, etc., and edges $e \in E$ defining a *relation* between them $r \in R$ (e.g. user-item interaction, belonging to a category, being directed by a certain person, etc.). Thus, an edge $e \in Edges$ is a triplet of the head (or left) entity $h \in V$, the relation $r \in R$, and the tail (or right) entity $t \in V$, i.e. $e = (h, r, t)$. We denote by r_0 the affinity relation between a user and an item (user-item interaction), and by r_j any other relation between entities.

The main idea of translation-based embedding is to project the entities and the relations of the knowledge graph into the *embedding space*, i.e. a d-dimensional vector space $\mathbb{R}^d$. Thus, to each entity $h \in V$ (resp. $t \in V$) corresponds a vector $\mathbf{v_h} \in \mathbb{R}^d$ (resp. $\mathbf{v_t} \in \mathbb{R}^d$). A *score function* $f_r(\mathbf{h}, \mathbf{t})$ is defined to measure the plausibility that the triplet is incorrect. In other words, the relation should correspond to a translation of the embedding. Various score functions have been proposed for achieving better accuracy for tasks as link prediction, node classification and recommendation [108]. Here we use two forms of score functions from TransE and TransH as examples of KGE for recommendation. More advanced and complex forms of score functions are believed to provide a better performance. The score function of TransE [14] is given by:

$$f_r^E(\mathbf{h}, \mathbf{t}) = ||\mathbf{v_h} + \mathbf{v_r} - \mathbf{v_t}||_p, \;\; \mathbf{v_r} \in \mathbb{R}^d \tag{10.13}$$

.

TransH [15] projects the embeddings $\mathbf{v_h}$ and $\mathbf{v_t}$ to a relation-specific hyperplane $\mathbf{w}_r$ and considers the relation-specific translation vector $\mathbf{v}_r$ in $\mathbf{w}_r$. Its score function is defined as:

$$f_r^H(\mathbf{h}, \mathbf{t}) = \Big|\Big| \underbrace{(\mathbf{v_h} - \mathbf{w}_r^T \mathbf{v_h} \mathbf{w}_r)}_{\text{projection of } \mathbf{v_h} \text{ to } \mathbf{w}_r} + \;\mathbf{d}_r\; - \underbrace{(\mathbf{v_t} - \mathbf{w}_r^T \mathbf{v_t} \mathbf{w}_r)}_{\text{projection of } \mathbf{v_t} \text{ to } \mathbf{w}_r} \Big|\Big|_2^2 \tag{10.14}$$

where $\mathbf{w}_r, \mathbf{v}_r \in \mathbb{R}^d$. For training, the following margin-based ranking loss function [15] can be used:

$$\mathcal{L}_{KGE} = \sum_{(h,r,t)\in\Delta} \sum_{(h',r',t')\in\Delta'} \Big[f_r(\mathbf{h}, \mathbf{t}) + \gamma - f_{r'}(\mathbf{h}', \mathbf{t}') \Big]_+ \tag{10.15}$$

where $[x]_+ \triangleq \max(0, x)$, γ is the margin between positive and negative triplets, Δ and Δ' denote the sets of positive and negative triplets, respectively. The negative triplets (h', r', t') are the results of the corruption of (h, r, t).

Furthermore we give the definiton of **Diversty-Aware Knowledge Graph Embedding**, where the item representations should capture the semantic diversity.

Definition 9. Diversity-Aware Knowledge Graph Embedding. $(E, V, \mathbf{A}, \mathbf{R})$ is defined as a knowledge graph, with E a set of entities, $V \subset E \times E$ the edges connecting the entities E; $\mathbf{A}$ is a mapping function $\mathbf{A} : E \to T$ with T a finite set of entity types and $\mathbf{R}$ is also a mapping function $\mathbf{R} : V \to S$ with S a finite set of relation types. Knowledge graph embedding methods represent both entities and relations of the knowledge graphs into vectors, $v_e, e \in E$ and $v_r, r \in V$ accordingly.And a vectorial diversity measure $div_v : \mathbb{R}^{n \times d} \to \mathbb{R}$ defined on a subset of v_E is given for n $k-$dimension vectors. M is a subset of E where a semantic diversity measure is defined $div_s : 2^M \to \mathbb{R}$. An **optimum** diversity-aware knowledge graph embedding should obey the following rule: For subsets M_i,M_j of M, if $div_s(M_i) \leq div_s(M_j)$ then $div_v(v_{M_i}) \leq div_v(v_{M_j})$.

11 DivKG: Improving Diversity for Recommendation Based on Knowledge Graphs

In this chapter, we present our first diversity-enhancing recommendation framework - **DivKG** to make recommendations combining knowledge graph embeddings (step 1) and determinantal point processes (DPP) prediction (step 2). Given a knowledge graph for recommendation, we first apply knowledge graph embedding methods to learn the representations of entities engaged in the recommendation (including users and items) then we present a new approach to construction DPP personalized kernel matrix based on the learnt vectors from knowledge graph embedding. Finally we generate and provide top-N diversified recommendation by applying a fast greedy Maximum A Posteriori (MAP) Inference algorithm on each kernel matrix. We present a visualized procedure of our framework in Figure 11.1.

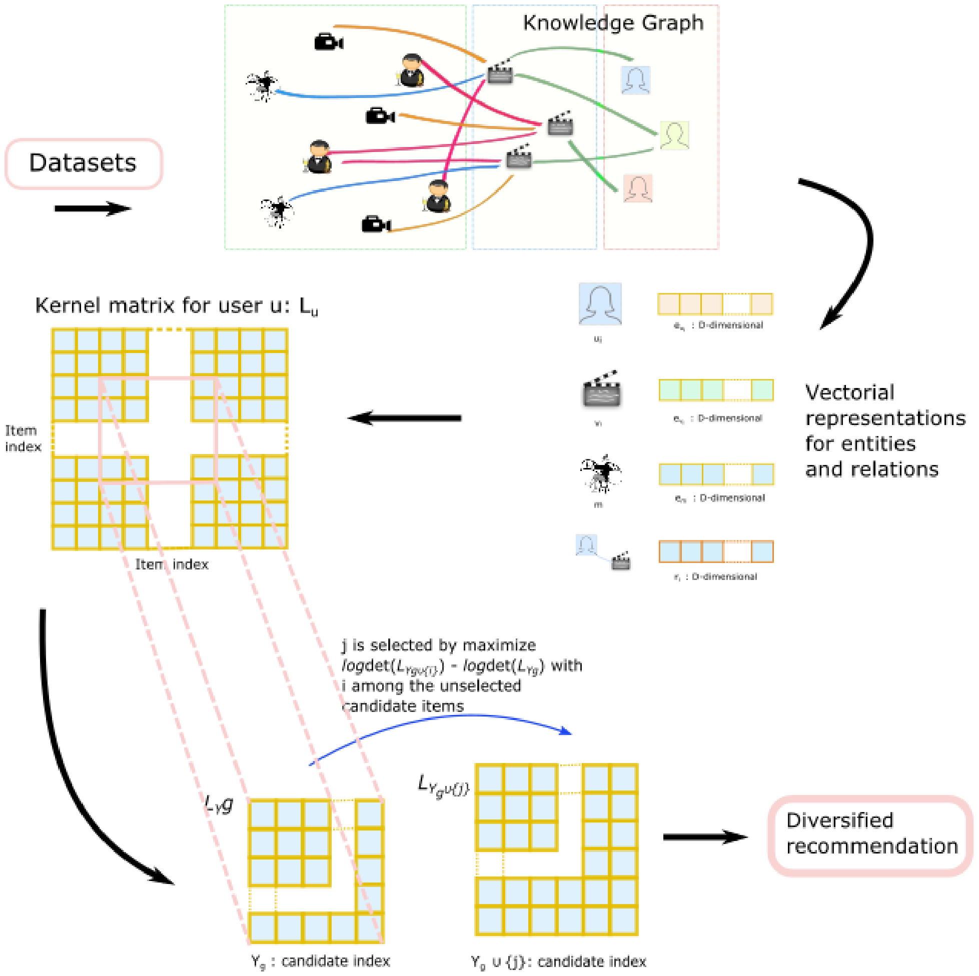

Knowledge Graph
Datasets
Kernel matrix for user u: L_u
Item index
Item index
e_{u_i} : D-dimensional
u_i
e_{v_i} : D-dimensional
v_i
e_{m} : D-dimensional
m
r_i : D-dimensional
Vectorial representations for entities and relations
j is selected by maximize $log\det(L_{Y_g \cup \{i\}})$ - $log\det(L_{Y_g})$ with i among the unselected candidate items
L_{Yg}
$L_{Y_g \cup \{j\}}$
Y_g : candidate index
$Y_g \cup \{j\}$: candidate index
Diversified recommendation

Figure 11.1: DivKG: a diversity enhancing recommendation framework based on knowledge graph embedding methods. Datasets are preprocessed to extract entity-relation triplets for constructing knowledge graphs for recommendation. Then given such a knowledge graph, we learn the vectorial representations of entities and relations by applying different knowledge graph embedding methods. After obtaining the latent vectors of users, items, other entities and relations, personalized kernel matrix can be constructed for each user. Finally we render each user a diversified recommendation list of items by the fast greedy MAP inference algorithm on its corresponding kernel matrix.

11.1 Knowledge Graph Embedding for Entity and Relation Representations

To improve the accuracy of recommendation, one can make use of additional information incorporated into collaborative filtering methods. Recently, using knowledge graphs to model this kind of data has been shown to enhance the recommendation [18, 19].

In this paper, we argue that knowledge graph is a robust and meaningful model that helps to blend multiple relations in one data structure. However, different from [19], where solely items are taken as vertices of the knowledge graph, we propose to consider all entities engaged in the recommender system, including users, items and other additional entities (*e.g.* genre, actor, etc.). Moreover, we consider user-item interaction used traditionally for CF-based methods just as a specific type of relation on the knowledge graph. More formally, we represent every relation instance as a triplet (h, r, t) having semantic interpretation, where h and t denote the head and tail entities linked by one relation r. Figure 11.2 demonstrates an example of such a knowledge graph structure in movie recommendation scenario.

11.1.1 Representation Learning for Entities and Relations

To apply embedding on such a knowledge graph, we represent each entity and each relation as a vector, *i.e.* h, r, t are represented as $\mathbf{v_h}$, $\mathbf{v_r}$,$\mathbf{v_t}$, respectively. We use translation-based embedding methods [14, 15] to interpret the translation semantics among vectors $\mathbf{v_h}, \mathbf{v_r}, \mathbf{v_t}$ which is $translation(\mathbf{v_h}, \mathbf{v_r}) \approx \mathbf{v_t}$. We use a margin-based loss function with margin γ to optimise the vector representation:

$$Loss_{KGE} = \sum_{(h,r,t)} \sum_{(h',r',t')} [f_r(h,t) + \gamma - f_{r'}(h',t')]_+ \tag{11.1}$$

The corrupted triplets (h', r', t') are derived from golden triplets (h, r, t) by (1) keeping the relation unvaried, *i.e.* $r = r'$, and (2) by either keeping unvaried the head entity and randomly selecting the tail entity, *i.e.* $h' = h,\ t' \neq t$, or keeping unvaried the tail entity and randomly selecting the head entity, *i.e.* $t' = t,\ h \neq h'$. And f_r denotes the translation function: TransE [14] takes

$$f_r(h,t) = \|\mathbf{v_h} + \mathbf{v_r} - \mathbf{v_t}\|_2 \tag{11.2}$$

TransH [15] takes

$$f_r(h,t) = \|(\mathbf{v_h} - \mathbf{w_r}^\top \mathbf{v_h} \mathbf{w_r}) + \mathbf{v_r} - (\mathbf{v_t} - \mathbf{w_r}^\top \mathbf{v_t} \mathbf{w_r})\|_2 \tag{11.3}$$

where $\mathbf{w_r}$ is a projection vector.

11.1.2 Top-N Prediction Based on Knowledge Graph Embedding

We obtain the vectors of all entities and relations through the optimization of the loss function 11.1. And based on these vectors, we can first generate a non-diversified top-N recommendation list for each user by measuring the similarity of users and items using the translation function (quality function). We notice that the translation function denotes the distance between two entities related by a relation and the further away two entities are the less similar these two entities are. Thus, to generate the top$N$ item list for each user, we select N items with lowest translation values as the top-N recommendation result.

11.2 Diversity-Aware Recommendation on Determinantal Point Processes

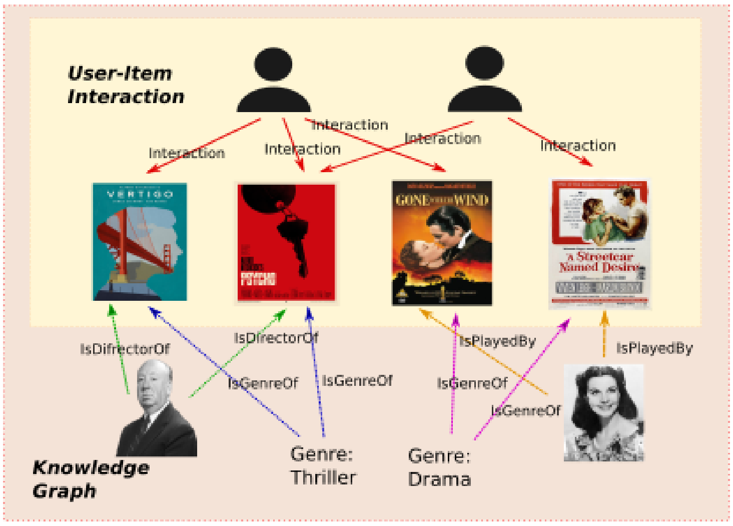


Figure 11.2: An example of knowledge graph for movie recommendation, reflecting different relations (*Interaction*, *IsGenreOf*, *IsDirectorOf*, *IsPlayedBy*) between various entities (user, item, genre, director, actress). Such a knowledge graph is an extension of user-item interaction used by collaborative filtering methods.

Here, we propose to exploit determinantal point processes (DPP) models to improve feature representation-based diversity, where feature representations are generated on the previous step, *i.e.* knowledge graph embedding (KGE). Note, that our framework is modular, and allows other quality estimation techniques to be used as input to the current step. However, we argue that the combination of KGE and DPP is the most efficient in terms of diversity-accuracy trade-off.

11.2.1 Construction of DPP Kernel Matrix

DPPs are a group of probability models to reflect the distribution of items from item list X over the set $Y, Y \subseteq 2^X$, where the selection of a subset $S, S \in Y$ of items is proportionate to the indexed determinant of the kernel matrix of DPP [31]. The kernel matrix of DPP is a positive semi-definite matrix which records the inherent affinity of each item appeared in the set Y and the similarities of every pair of two different items. More specifically the diagonal elements of the kernel matrix reflect the inherent affinity of each item and the non-diagonal elements reflect the pairwise similarity of the item set.

In order to construct a kernel matrix $\mathbf{L_u}$ for each user u for top-N diversified recommendation, we define two auxiliary matrices. The first is user's u affinity profile w.r.t. candidate items defined as a diagonal matrix $\mathbf{A}_u = \mathrm{diag}(a_1, ..., a_m)$, where m is the number of candidate items,

$$a_i = \frac{e^{-(f_r(u,i)-\delta)}}{\sum_{j \in X, j \neq i} e^{-(f_r(u,j)-\delta)}} \tag{11.4}$$

where $f_r(u,i)$ is the result of the item quality estimation function calculated in the previous step and δ is the average of $f_r(u,i)$ for u. We consider here the embedding translation function for user u, item i and translation type r. Here, we use the $softmin$ function to normalize the affinities of all items for each user as the item quality function $f_r(u,i)$ from the knowledge graph embedding measures the distance of a pair of user u and item i by the relation r. And the more similar between u and i the smaller the distance value is.

The second matrix reflects item pairwise and listwise similarity and is defined as $\mathbf{D}_u = [d_{ij}]^{m \times m}$, whose entries

$$d_{ij} = \frac{e^{-f_{r_0}(i,j)}}{\sum_{k \in X, k \neq i} e^{-f_{r_0}(i,k)}} \tag{11.5}$$

and $d_{ii} = 0$, where $f_{r_0}(i,j)$ is the result of the embeddings of items i and j and relation r_0, whose vector $v_{r_0} = \overrightarrow{0}$ if items i and j share the same relation value (category).

Finally, the kernel matrix $\mathbf{L_u}$ for user u can be defined as:

$$\mathbf{L_u} = \alpha \mathbf{A}_u + \mathbf{D}_u \tag{11.6}$$

where α is a parameter to adjust the trade-off between user's affinities and similarities among the items, or in other words, between accuracy and diversity. In table 11.1 we demonstrate a visualized kernel matrix under our construction approach.

11.2.2 MAP Inference for Diversified Prediction

After the construction of the kernel matrix for each user u, we aim at selecting a list S of size N of items from total candidate items, s.t.

$$S_{map} = \underset{S \in Y, |S|=N}{\mathrm{argmax}}\, log\det(\mathbf{L}_S) \tag{11.7}$$

	$item_1$	$item_2$	$item_3$	$\dots$	$item_M$
$item_1$	a_{u1}			$\dots$	
$item_2$		a_{u2}		$\dots$	
$item_3$			a_{u3}	$\dots$	d_{3M}
$\dots$				$\dots$	
$item_M$				$\dots$	a_{uM}

Table 11.1: The construction of personalized kernel matrix for DPP. For a user having M candidate items, we construct a $M \times M$ dimension matrix where the diagonal element representing the affinities of each item towards the user (a_{u1} for example represents the affinity or relevance for item 1 towards user u) ;and non-diagonal element representing the pairwise similarity of items (d_{3M} for example represents the similarity between item 3 and item M). Due to the properties of DPP, such a kernel matrix will garantee a set of items with high affinity and diversity will be more likely to be selected from all candidate items.

where $\mathbf{L}_S$ is the kernel matrix $\mathbf{L}_u$ indexed by items from S. We recall that the probability of selecting a subset S is proportionate to the determinant of the indexed kernel matrix and DPP promotes a diversified selection of items under its property by definition. Thus, the selected items with the maximum log determinant value theoretically determine the best related and diverse top-N items for user u. However, such an optimization problem has been proven to be NP-hard, thus we use the fast greedy algorithm proposed by [100] to retrieve an approximate top-N list as the diversified recommendation result. We refer to this DPP model with MAP inference as **FastDPP**. In Figure 11.3 we demonstrates the iterative procedure of how FastDPP selects items to return the top-N diverse items.

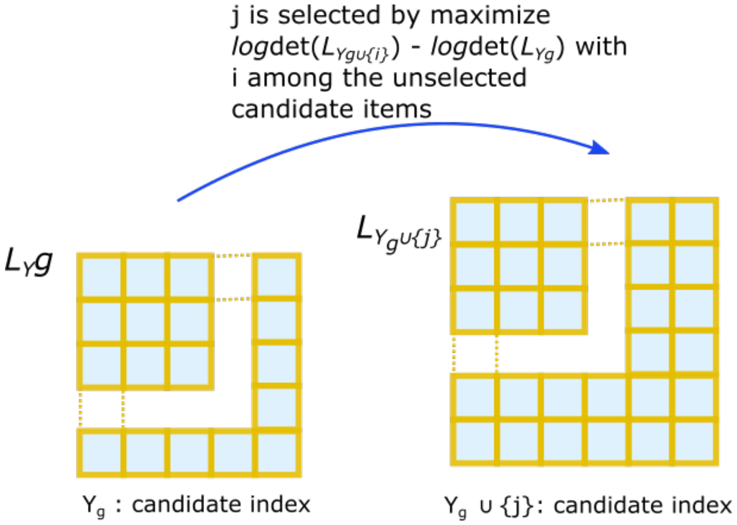

j is selected by maximize
logdet($L_{Y_g \cup \{i\}}$) - logdet(L_{Y_g}) with
i among the unselected
candidate items
L_{Yg}
$L_{Y_g \cup \{j\}}$
Y_g : candidate index
$Y_g \cup \{j\}$: candidate index

Figure 11.3: The iterative process of selecting items by FastDPP. Y_g is the dynamic item set containing the items already selected by FastDPP and is initialized as an empty set $\emptyset$. When the number of items in Y_g has not reached N, FastDPP updates Y_g by selecting an item j among all the unselected candidate items which maximizes $log\det(L_{Y_g \cup \{i\}}) - log\det(L_{Y_g})$.

12 EMDKG: Diversity-Aware Representation Learning for a Better Trade-off Diversified Recommendation

In this chapter, we describe our second diversified recommendation framework **EMDKG** which targets at optimizing both item diversity representations and knowledge graph embedding for top-N recommendations. Our previous model DivKG proposes a quite elegant approach to conbine knowledge graph embedding with determinantal point processes for a diversified recommendation. However, the key part of contructing personalized DPP kernel matrix relies on an important assumption that item representation can fully capture the similarity/disimilarity among them. But the assumption may not be true all the time. An example is given as such. The motivation of this framework is that the representations of items should reflect the semantic diversity of items given a specific semantic diversity definition. And we can only do this by explicitly encoding this semantic information into item vectors. In the following of this paper, EMDKG$_E$ (resp. EMDKG$_H$) denotes our solution EMDKG incorporating Trans$_E$ (resp. Trans$_H$) knowledge-graph embedding.

12.1 General Overview

We propose an EM-schemed representation learning for recommendation. Corresponding to the two objectives for diversity-aware translation-based recommendation, the E-step aims at optimising translation-based knowledge graph representations for users, items and entities with a modified margin-based ranking loss for taking into account the diversity of item vectors. And the M-step aims at learning a diversity-encoded item representations with a pairwise BPR-based loss. The general model alternates the learning processes of these two parts until it converges. We demonstrate the framework EMDKG in Figure 12.1.

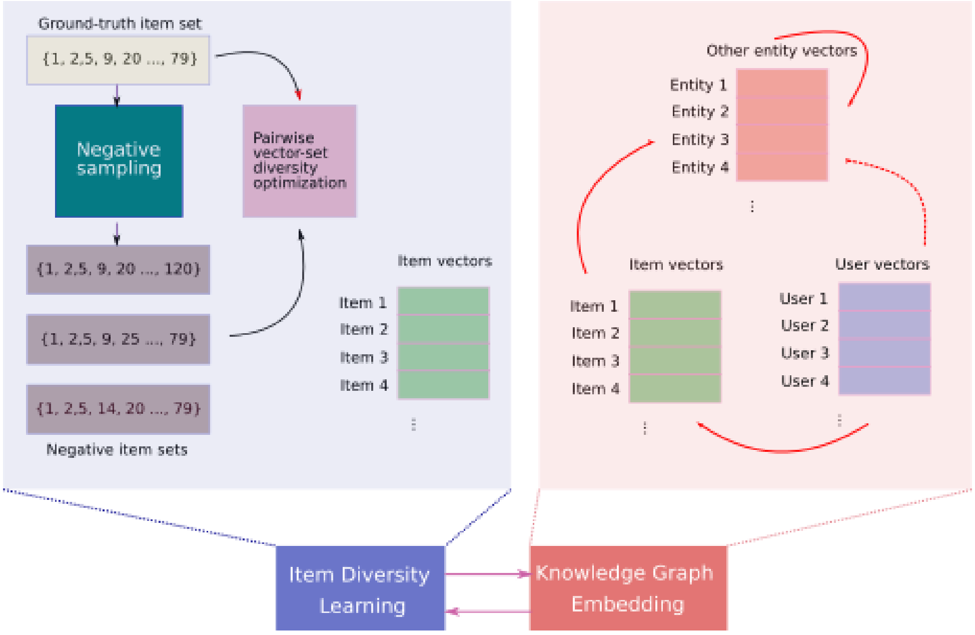


Figure 12.1: This is the general structure of EMDKG, composed of two modular learning: the Item Diversity Learning and Knowledge Graph Embedding.

12.2 Diverse Item Sets Generation

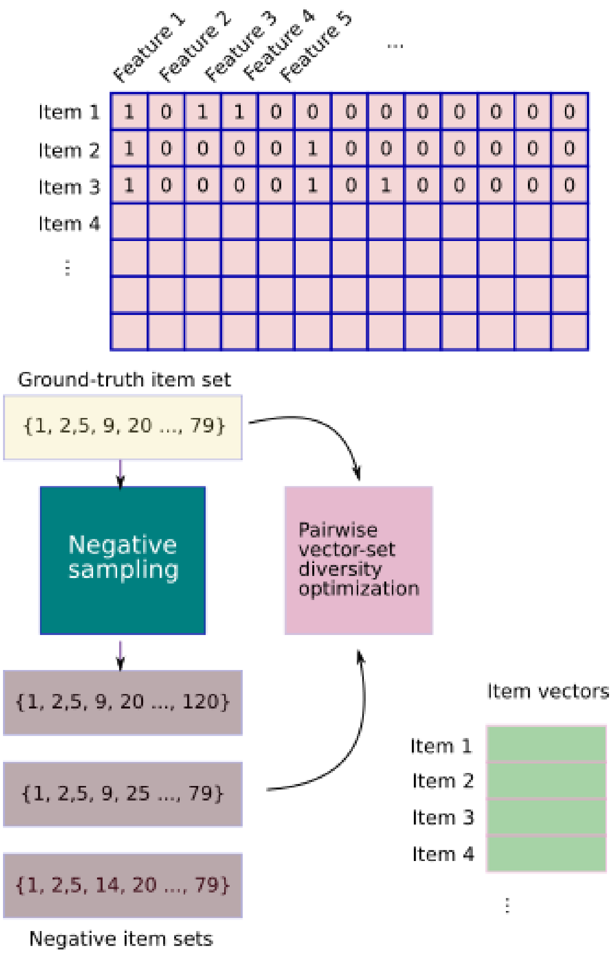
As we want to encode semantic diversity information into item representation, it is important to possess ground-truth diverse item sets as benchmarks. That is to say, given a specific diversity metric and items with their semantic information, a set of diverse item sets should be given or generated using available information. As we are interested in top-N recommendation lists for each user, we also prefer to obtain diverse item sets with a fixed length. As it is not common that ground-truth diverse sets are provided in datasets, we propose a procedure to randomly generate fixed length diverse item sets.

The procedure takes the historical user-item interactions and item categorical information, and generate Δ random historically interacted item sets. Given a semantic diversity $Div(\cdot)$ we can calculate for each generated item set its semantic diversity score. Then we select sort the Δ item sets by its diversity score in descending order and keep the highest as ground-truth diverse item sets. We note that Δ should be sufficiently big for selecting diverse item sets. And we show this procedure in Algo. 1. Here, we denote the historical interacted items for each user u as I^u_{candid} and item categorical information as C_I.

Algorithm 1 Diverse Item Set Generation

```
Inputs: $U$,$C_I$, $N$,$I_{candid}$, $Div(\cdot)$, $\Delta$
Outputs: T
for $u \in U$ do
    $RandomSet \leftarrow \emptyset$
    for $\Delta$ times do
        Randomly generate a set $T'$ of $N$ items from $I^u_{candid}$
        $Div(T') \leftarrow$ Calculate diversity score of the set $T'$
        Append $(T', Div(C_{T'}))$ to $RandomSet$
    end for
    Sort $RandomSet$ by $Div(\cdot)$ value in descend order
    Append top pairs from $RandomSet$ to T
end for
```

12.3 Item Diversity Learning.



	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	...							
Item 1	1	0	1	1	0	0	0	0	0	0	0	0	0
Item 2	1	0	0	0	0	1	0	0	0	0	0	0	0
Item 3	1	0	0	0	0	1	0	1	0	0	0	0	0
Item 4													
⋮													

Figure 12.2: A demonstration of how Item Diversity Learning module works.

Diversity can be considered in terms of categories. In this case, one-hot encoding [109] can be used to represent item features. However, such approach becomes infeasible when the definition of diversity goes wider or the category number increases significantly. To overcome these limitations, vector representations can be used. Indeed, with the growing popularity of matrix factorisation and embedding techniques, modern RS largely rely on user and item representations in continuous vector space. We argue that diversity should be encoded in vector representations of items. To do so, we propose an item diversity learning framework with negative sampling.

We first need to demarcate the concepts of (a) *semantic diversity* based on the available information about the items (take categories/movie genres as example), and (b) *vectorial diversity* based on item vector representations and calculated using item vectors. In the ***Item Diversity Learning (IDL)*** module, we make use of both concepts. Our motivation behind that is as follows. In traditional RS, item vectors are often learnt by optimising item relevance to a given user profile (user vector). At the same time, a general principle is that similar users tend to like similar items. Thus, there exists a correlation between the similarity of items and their vector representations. However, it is not always the case.

A diversity measure $div(\cdot)$, whether on discrete or continuous space, should be defined to characterise the diversity for a given list. For instance, one can use *pairwise vector dissimilarity metrics* like intra-list average distance (ILAD) [110], intra-list minimal distance (ILMD), or *set-level metrics* like category coverage (CC), α-NDCG and log-determinant of item-indexed kernel matrix employed in DPP [100] defining diversity in the vector space of the entire list of items. As semantic information is handled in one-hot encoding on discrete space, only ILAD, ILMD, CC and α-NDCG can be used to calculate this information. In contrast, learned vector representations are in continuous vector space, and metrics as determinant, ILAD and ILMD are the possible diversity measures here.

For a given user $u \in U$, the IDL-module aims at learning the representations of items which can reflect the semantic diversity based on the available information (features, relations). As input, it takes the set of ground-truth item sets, denoted $\{T_+\}$, each of the element T_+(one individual item set) having the same size (length) consisting of the items the user had interactions with. We consider the item sets in this set to be the most diverse for each given user.

Based on each element from the ground-truth set T_+ and the remaining items $I \setminus T_+$, the *negative sampling* is performed by randomly replacing all the items from T_+ except for one with the items from $I \setminus T_+$.

Note that the size of the item sets is the same, i.e. $|T_-| = |T_+|$. The items for substitution are selected so that the overall semantic diversity of these negative item sets is inferior to the one of the ground-truth. In other words, if $div(\cdot)$ is a list diversity measure, then $div(T_+) > div(T_-)$. At this step, we consider semantic diversity and suggest to apply dissimilarity measure on discrete space to calculate it. Thus, negative item sets T_- are generated.

Once the negative items sets are constructed, the IDL-module learns item vector representations by optimising the following loss function: $\mathcal{L}_{IDL} = -\log\left(1 + e^{-div(T_+)+div(T_-)}\right)$.

Here, we make use of vectorial diversity and apply log-determinant measure to calculate it. Thus, the loss function gets the following form:

$$\mathcal{L}_{IDL} = \sum_{A_+ \in \{T_+\}} \sum_{A_- \in \{T_-\}} - \log \Big(\sigma(\log \det(L_{A_+}) - \log \det(L_{A_-}) \Big) \tag{12.1}$$

where A_+ is the ground truth diverse item set, and A_- is one item set from all negative item sets, L_{A_+} and L_{A_-} are the kernel matrix of DPP indexed with the elements from A_+ and A_-, respectively.

This kernel matrix is built as follows. Given an item list $A \subset I$, we notate with $\mathbf{v}_A$ a vector representation of the item list A. Then $L_A = \mathbf{v}_A {\mathbf{v}_A}^\intercal$ is a positive semidefinite matrix. The larger the value of the determinant $\det(L_A)$ is, the more diverse the item set A is [32].

As the result of the IDL-module, we obtain vector representations of items that reckon with semantic similarity and vector similarity. And we demontrate the Item Diversity Learning module in Figure 12.2.

12.4 Modified Translation-Based Knowledge Graph Embedding for Recommendation.

Auxiliary semantic information related to items such as categories, film directors, actors, etc. can provide valuable assets for enhancing recommendation accuracy [111, 112]. Such auxiliary information can contain categorical information and relational knowledge with other entities which do not engage in the recommendation directly. It has been shown [108] that a translation-based knowledge graph embedding for recommendation can efficiently take into account (1) various entities that may affect user's preferences/choices for items and (2) different types of relations between them.

In this work, we take a similar approach to construct a knowledge graph as in DivKG, in terms of modelled relations, we distinguish between user-item interactions and any other relation between entities. We refer to the latter as auxiliary relations. Let r_0 be the relation between users and items reflecting a user-item interaction. As described in Chapter 7, for the triplet (u, r_0, i) we can define a translation-based score function $f_{r_0}(\mathbf{u}, \mathbf{i})$ to measure the affinity value between the user u and the item i. The smaller the value of $f_{r_0}(\mathbf{u}, \mathbf{i})$ is, the larger the affinity between the user u and item i is. Similarly, we can define the score function $f_{r_j}(\mathbf{i}, \mathbf{e})$ for any auxiliary relation $r_j, j \in \{1, 2, ...\}$ between the item i and the auxiliary entity $e \in E = V \setminus \{U \cup I\}$.

To learn the embedding accounting for both types of relations (user-item interactions r_0 and auxiliary relations $r_{j,j\neq 0}$), we can rewrite the margin-based ranking loss function from eq. 10.15 as:

$$\begin{aligned}\mathcal{L}_{KGE} = \sum_{(u,r_0,i)} \sum_{(u,r_0,i')} [-f_{r_0}(\mathbf{u},\mathbf{i}) + f_{r_0}(\mathbf{u},\mathbf{i}'),0]_{+}+ \\ \sum_{(i,r_j,e)} \sum_{(i,r_j,e')} [-f_{r_j}(\mathbf{i},\mathbf{e}) + f_{r_j}(\mathbf{i},\mathbf{e}'),0]_{+}.\end{aligned} \tag{12.2}$$

As stated above, such loss function aims at separating the golden triplets (both, historical user-item interactions and existent item-entity relations) from the negative triplets.

However, conventional KGE only considers the existent relations of entities in the graph and does not explicitly optimize the diversity representations of item vectors as we do in Section **??**. Thus we propose a modified knowledge graph embedding loss function to bridge KGE and Item Diversity Learning:

$$\mathcal{L}_{xKGE} = \mathcal{L}_{KGE} + KLDivergence\left(\mathbf{V}_I^{KGE}, \mathbf{V}_I^{IDL}\right) \tag{12.3}$$

where $\mathbf{V}_I^{KGE}$ and $\mathbf{V}_I^{IDL}$ represent correspondingly the item vectors in KGE and Item Diversity Learning. We minimize KL-divergence of the vector representations of items in two modules in order to resemble the two item representations. Finally, we alternate the learning of knowledge graph embedding and item diversity learning by alternating optimisation of functions (eq. 12.1) and (eq. 12.3) until the the learning converges. Thus, we can formalize our proposal EMDKG as a dual-goal optimization problem of the Diversity-Aware Translation-Based Recommendation as follows.

(Diversity-Aware Translation-Based Recommendation.) Given a set of users U, items I, other entities E, historical user-item interactions $H_{u,r_0,i}$ and item-side relation information triplets $H_{i,r_j,e}$, the diversity-aware translation-based recommendation aims at minimising two loss functions 12.1 and 12.3 simultaneously.

The process of co-learning is shown in Algorithm 2.

Algorithm 2 Co-learning of KGE and IDL

```
Inputs: $\mathbf{T} = \{T_+\}$, $\mathbf{P} = \{(u,i)\}$, $\mathbf{Q} = \{(i,r,e)\}$, $k$
Outputs: $V_I^{IDL}, V_I^{KGE}, V_U, V_R$
Initialize $V_I^{IDL}, V_I^{KGE}, V_U, V_R$
while not converge do
    for $k$ times do
        Get batch $\mathbf{p} \subseteq \mathbf{P}$ and $\mathbf{q} \subseteq \mathbf{Q}$
        Optimize Eq.(12.3) with $\mathbf{p}$ and $\mathbf{q}$
    end for
    Get batch $\mathbf{t} \subseteq \mathbf{T}$
    Using negative sampling to obtain $\mathbf{t}_-$ from $\mathbf{t}$
    Optimize Eq.(12.1) with $\mathbf{t}$ and $\mathbf{t}_-$
end while
```

12.5 Prediction

To make a top-N recommendation, we take the learned vectors of users and items and affinity relation r_0 from KGE and calculate the affinity score for each user u with any item i using the score function $f_{r_0}(\mathbf{u}, \mathbf{i})$. For each user $u \in U$, we sort the items by ascending score function values, and return the top-N items as the result.

To further adjust the recommendation list, we can also apply diversification methods to balance the accuracy-diversity trade-off of the list. In diversification methods, such as MMR [29], XQuAD [87], a threshold parameter that we denote α is used to adjust the trade-off between accuracy and diversity. For instance, MMR optimisation function is given by:

$$\max\left\{\alpha Sim_1(u,i) - (1-\alpha) max(Sim_2(i,j))\right\} \tag{12.4}$$

where $Sim_1(u,i)$ reflects the relevance of the item i to the user u, and $Sim_2(\cdot,\cdot)$ is a similarity measure between two items.

We propose the following MMR-like determinant-based trade-off equation:

$$\log\det(L_A) \propto \alpha \sum Q(u,i) + (1-\alpha) \cdot \log\det(L_A) \tag{12.5}$$

which is equivalent to the log-determinant of submatrix A on a new kernel

$$L_u = \mathrm{Diag}(exp(\beta Q(u))) \cdot L\mathrm{Diag}(exp(\beta Q(u))) \tag{12.6}$$

The change of operations from $-$ to $+$ is due to the semantics of $\log\det(\cdot)$, interpreted as the dissimilarity of the item list $\cdot$, contrary to the $Sim_2(\cdot,\cdot)$ being the similarity between any item pair. $Q(u,i)$ is the affinity measure between any user-item pair (u,i). Higher value of $Q(u,i)$ corresponds to closer affinity of the pair (u,i). However, in KGE setting, the lower the value of $f_{r_0}(\mathbf{u}, \mathbf{i})$ is, the higher the affinity is. Thus, we apply a monotonically decreasing function, i.e. e^{-x} to satisfy the requirements of $Q(u,i)$.

13 Conclusion

In this part, we have presented our two models to diversified recommendations based on knowledge graph embedding methods. Our first diversified recommendation model DivKG proposes a new approach to construct personalized DPP kernel matrix based on translation-based knowledge graph embedding results for diversified item lists. The new approach for the personalized DPP kernel matrix is composed of user-item affinity matrix and item-item similarity matrix, both derived of entity and relation vectors from knowledge graph embeddings. In DivKG we assume the item vectors learnt from knowledge graph embedding can capture well the co-relation with other items. Our second diversified recommendation EMDKG however does not rely on the assumption that item vectors solely based on knowledge graph embedding can capture good enough diversity of item lists. And we propose double-module framework for learning diversity-aware knowledge graph embedding for top-N recommendation and use a co-learning to optimize the representations.

Part IV

Evaluation

14 Introduction **75**

15 Experimental Settings **75**

15.1 Datasets 75

15.2 Evaluation Protocol 80

16 Experiment Results **81**

16.1 Experiments on DivKG 82

16.2 Experiments on EMDKG 84

17 Conclusions **91**

14 Introduction

In the previous part we have presented two top-N diversified recommendation models DivKG and EMDKG based on knowledge graph embedding methods and determinantal point processes. To quantify the effectiveness of our proposals, in this part we will present their evaluations based on public real-world datasets for diversified top-N recommendation tasks. Before presenting the details of evaluation, I would like to recall that a diversified top-N recommendation task aims at achieving both high accuracy and diversity at the same time. Furthermore, as optimal accuracy and optimal diversity may not be able to be achieved at the same time, it is then required to discuss whether we can achieve a good trade-off between the two aspects. Finally, as both re-ranking diversification methods (e.x. MMR) and our methods select a diversified N item lists as recommendation results, we would also like to know how robust the models are in different settings.

To demonstrate all these, we first introduce in Chapter 15 the basic experimental settings including datasets information and common evaluation protocols applied in our experiments. Then we demonstrate separately in Section 16.1 and Section 16.2 the experiments on our models DivKG and EMDKG against state-of-the-art baselines in terms of accuracy and diversity metrics. We also give discussions along the experiments for each model.

15 Experimental Settings

15.1 Datasets

For the evaluation purpose, we construct our first multi-relation dataset by combining two real-world datasets (Movielens-100K and IMDb datasets). Our first dataset is MovieLens-100K (denoted ML-100K) containing 100,000 user ratings ranging from score 1 to 5 from 943 users on 1,682 movies. Each user has rated at least 20 movies. However, the rating matrix of ML-100K is still highly sparse, with a sparsity of 93.70%. The second dataset for the multi-relation dataset is IMDb Dataset which is currently released on IMDb website[1]. It contains information including crews, principals, different releasing versions of more

[1] IMDb datasets link: https://datasets.imdbws.com/

than 947K films. We demonstrate a diagram of metadata of these two datasets in Figure. 15.4.

We process these two datasets as follows. For the rating information from ML-100K, we follow traditional idea to binarize explicit rating data by keeping the ratings score >= 4 or higher and interpret them as implicit feedback. And for item-side auxiliary information, we extract 13 categories of information from IMDb, namely: movie genre, self (where the actor/actress plays himself/herself in the movie), director, cinematographer, composer, producer, editor, actor, actress, writer, production designer, archive footage, and archive sound. We combine these data with MovieLens dataset for constructing multi-relation datasets, using the extracted categories to determine the relations within our knowledge graph. We denote the constructed multi-relation dataset as **ML100K-IMDb**. In Figure 15.1, we demonstrate the statistics of the genre information over total users, while observing a unbalanced distribution of genre information over total items. We also demonstrate in Figure 15.2(a), the correlation of different genres are shown in the heatmap, where the correlation between genres are in general not so significant.

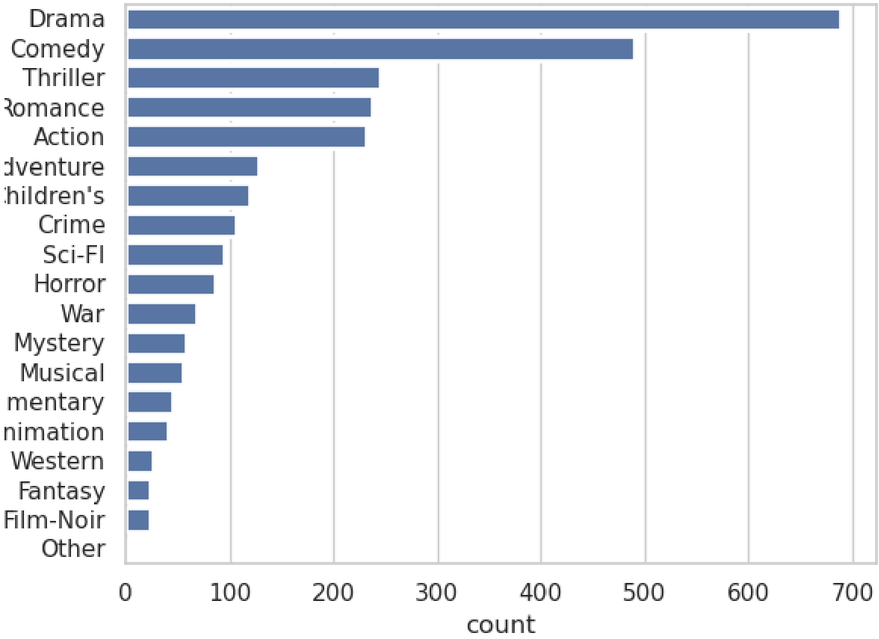


Figure 15.1: Movie genre count over all items on dataset ML100K in descending order.(19 genres including Other in total)

We also use **Anime** dataset[2] which contains 5+ million 10-grades ratings from 73,516 users on 12,294 animes. For Anime datasets, we are only given genre information as auxiliary information for constructing the knowledge graph; each anime may belong to one

[2] Anime datasets link: https://www.kaggle.com/datasets/CooperUnion/anime-recommendations-database

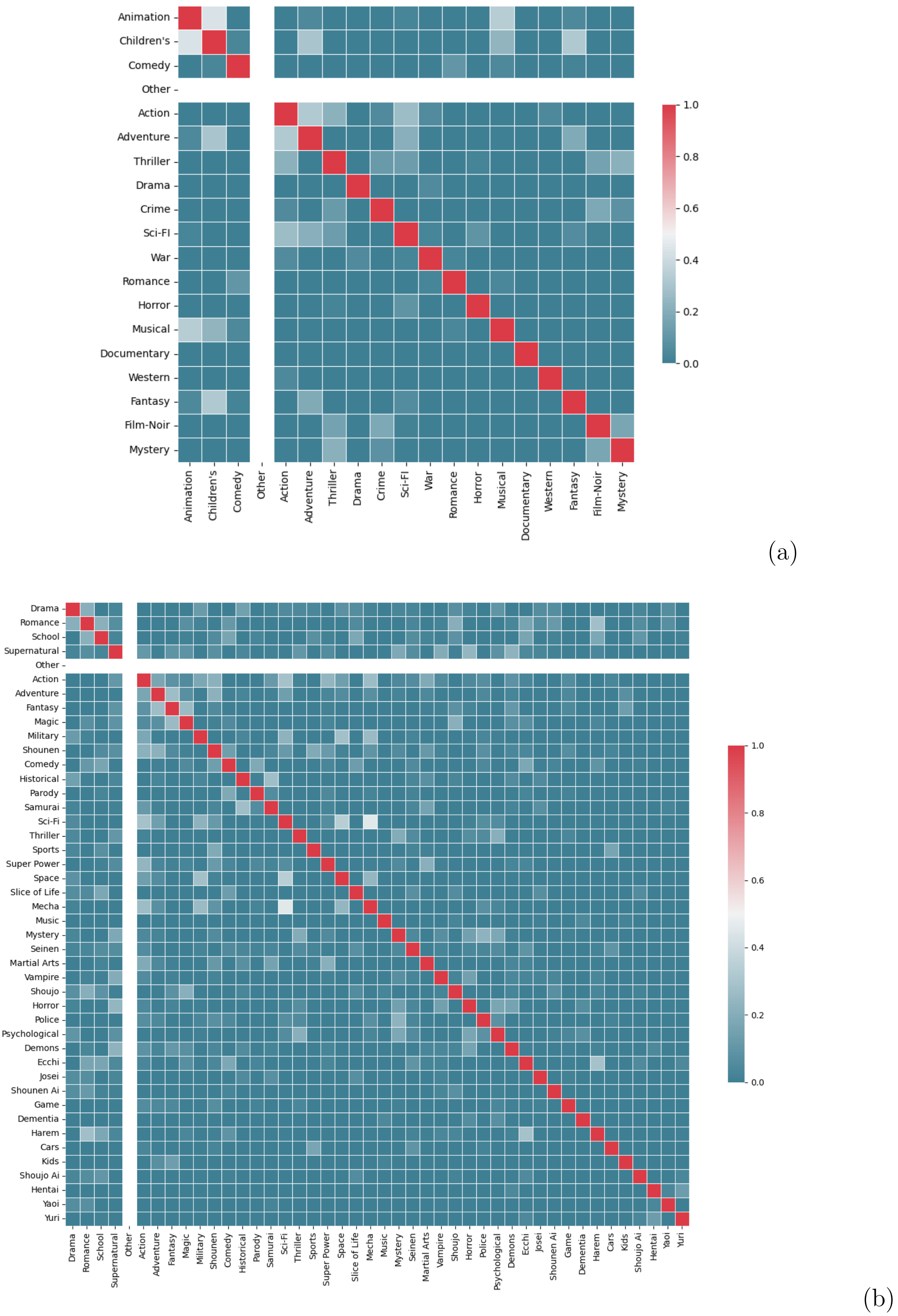


Figure 15.2: Correlation of genres on dataset ML100K and Anime. (a) the correlation of 19 movie genres based on genre distribution of items on dataset ML100K; (b) the correlation of 44 genres based on genre distribution of items on dataset Anime.

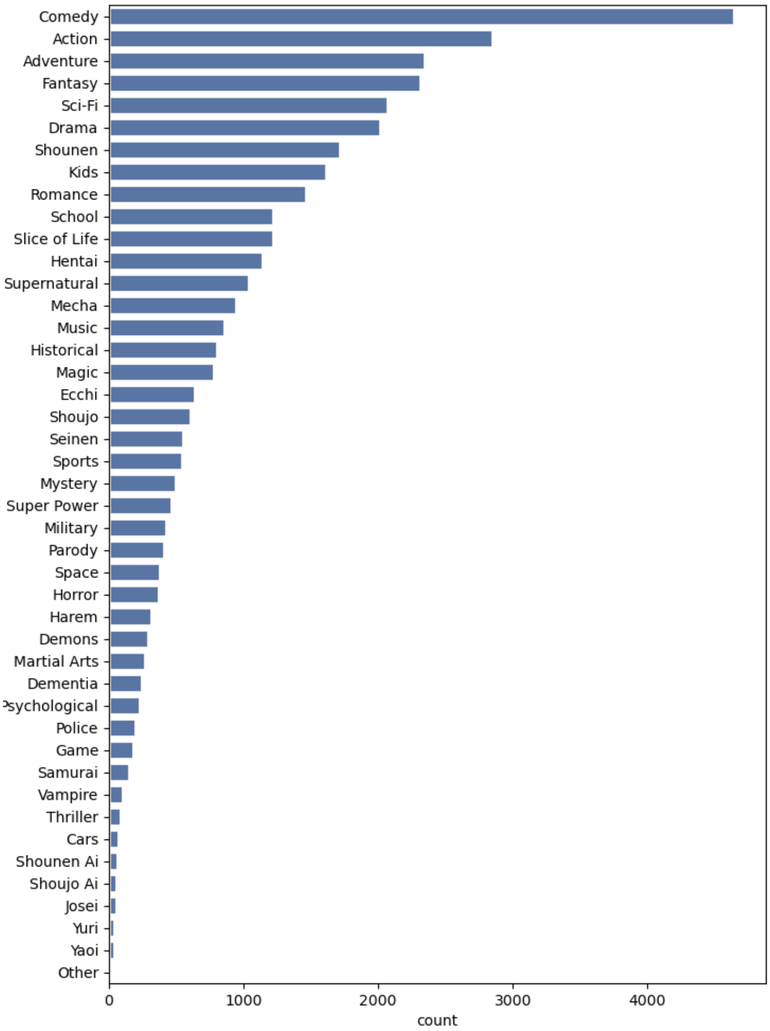


Figure 15.3: Anime genre count over all items on dataset Anime in descending order. The anime genres are comedy, action, fantasy, sci-fi, drama, shounen, kids, romance, school, slice of life, hentai, supernatural, mecha, music, historical, magic, ecchi, shoujo, seinen, sports, mystery, super power, military, parody, space, horror, harem, demons, martial arts, dementia, psychological, police, game, samurai, vampire, thriller, cars, shounen ai, shoujo ai, josei, yuri, yaoi, and other.

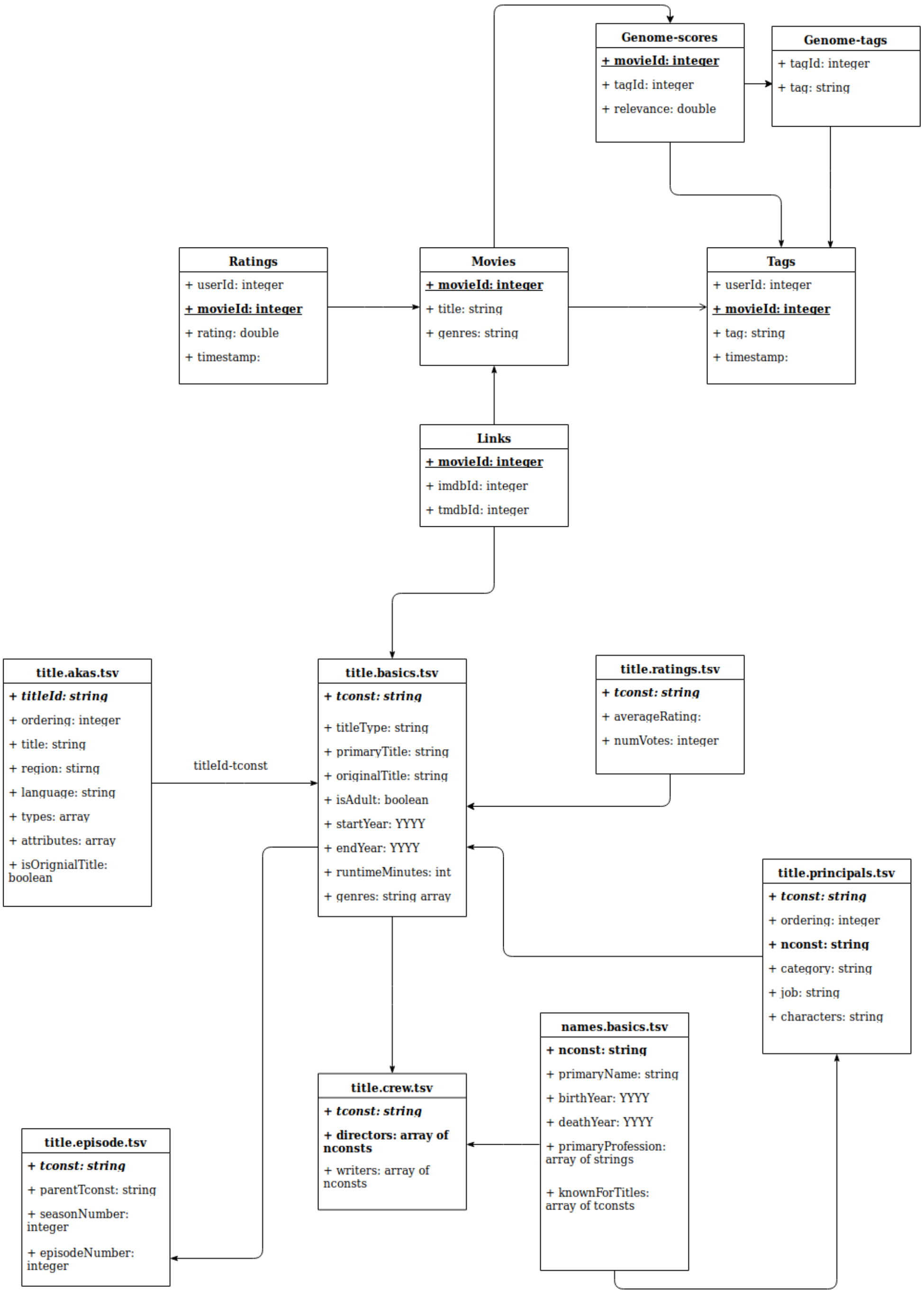


Figure 15.4: The original metadata of datasets ML-100k and IMDb.

or more of the 44 non-exclusive genres available in the dataset, namely comedy, action, fantasy, sci-fi, drama, shounen, kids, romance, school, slice of life, hentai, supernatural, mecha, music, historical, magic, ecchi, shoujo, seinen, sports, mystery, super power, military, parody, space, horror, harem, demons, martial arts, dementia, psychological, police, game, samurai, vampire, thriller, cars, shounen ai, shoujo ai, josei, yuri, yaoi, and other. The statistics of Anime dataset for the genre information is shown in Figure 15.3 and the correlation between genre based on item-genre information is demonstrated in Figure 15.2(b). For processing of Anime datasets, we follow the same procedure as for ML100K-IMDb dataset and treat the rating data as implicit feedback and consider ratings of 6/10 and higher as positive feedback, resulting in more than 1.8 million of positive interactions.

As we employ the *leave-one-out* strategy, we randomly divide rating information contained in both ML100K-IMDb and Anime datasets into training/ validation/ testing datasets. The random division into training/validation/testing datasets are done in five times with different initial random seeds. The experiments are operated over five different divisions of each dataset and the average results are reported. Item-side auxiliary information are added to training datasets as relational information.

For EMDKG model, we need to generate ground-truth diverse item sets for a given user by following the procedure described in Section 12.2. We use items from historical user-item interactions as candidates and the given categorical information (genres) of items to randomly generate item sets of length 10 and keep the top-100 item sets with the highest semantic diversity scores.

All the experiments are written mostly in Python (PyTorch framework for model learning and prediction), and part of the re-usable libraries for knowledge graph embedding is written in C++, callable in Python. Experiments presented in this dissertation were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

15.2 Evaluation Protocol

We evaluate our frameworks regarding the accuracy and diversity of the returned results. For both models the evaluation is performed in two steps. We first assess the quality of the embedding part, and then the results of diversification (DPP sampling) part.

For measuring accuracy of the top-N recommendation, we use two traditional metrics in information retrieval for document ranking:

1. **Normalized Discounted Cumulative Gain (NDCG)** [113]: NDCG takes the positions (ranks) of correct items into consideration. Given a top-N recommendation

list sorted in a descending order by a relative relevance, let $rel_{u,j}$ be the graded relevance of the recommendation at position j for the user u in the test set, and Z be the normalization constant. NDCG@N is then given by:

$$\text{NDCG@}N = \frac{1}{Z}\text{DCG@}N = \frac{1}{Z}\sum_{j=1}^{N}\frac{2^{rel_{u,j}} - 1}{log_2(j+1)}$$

.

2. **Hit ratio** *hit*@N which only considers whether a test item is in the returned recommendation list and it is formulated as,

$$hit@N = \begin{cases} 1 & \text{if } \exists j \in N, rel_{u,j} = 1 \\ 0 & \text{otherwise} \end{cases}$$

We calculate both metrics $hit@N$ and NDCG@N for each test user and report the average score.

To evaluate the performance of recommendation diversity, due to the differences of diversity definition in our models DivKG and EMDKG, we specify different diversity metrics for each method and the details of the diversity metrics are given in coming chapters for exhibiting experiments of each model.

For the two steps of evaluation, we apply the following evaluation procedure. To evaluate the accuracy performance of recommendation, we adopt the *leave-one-out* strategy which is widely used in literature [19] in both knowledge graph embedding and DPP processes. Thus, for knowledge graph embedding part (see Section 11.1 and Section 12.4), for each user, we randomly select one user-item interaction (rating) to constitute our *test set*, and then we randomly split the remaining interactions to *training set*, and *validation set* with ratio 80 : 20, respectively. We add all the other auxiliary information triplets to the training sets for representation learning purpose.

For the diversification part using DPP (see Section 11.2) and Section 12.5 , we randomly hold one user-item rating and mix with a fixed length M most similar items calculated from knowledge graph embedding results. The length of M will be specified along with experiment results.

16 Experiment Results

In this chapter we demonstrate the experiments on our two proposals separately in Section 16.1 and Section 16.2.

16.1 Experiments on DivKG

16.1.1 Baselines

To compare the performance of the first step of our framework, we use the following baseline algorithms:

- **BPRMF** [48]. This is a matrix factorization method optimised by a pairwise ranking loss (BPR-OPT, formulated in Eq. (6.22)) to learn from implicit feedback. This method only uses historical user-item interactions for learning and predicting recommendation and does not use relational data for learning process.
- **FISM** [49]. This is an item-based collaborative filtering method that considers a discount factor for the item and user latent vector multiplication. We also apply the pairwise ranking loss criterion BPR-OPT for the optimization of this method. We adopt and modify the implementation provided by [19].
- **RCF** [19]. This is a knowledge graph based method that considers both user-item interactions and other types of item relations and proposes a double-layer neural model for learning-to-rank top-N recommendation.

For the sake of a fair comparison, we combine the aforementioned models with two diversification models, namely our FastDPP (see Section 11.2.2) and the well-acknowledged diversification method MMR to compare recommendation performance both on accuracy and diversity.

Maximal Marginal Relevance (MMR) [29]. This is a re-ranking criterion to reduce redundancy while maintaining document relevance in the field of text summarization. Specifically, MMR iteratively chooses an item satisfying the following requirement:

$$\omega_i^* = \underset{\omega_i \subseteq X \setminus S}{\mathrm{argmax}} [\lambda r_{\omega_i} - (1-\lambda) \max_{\omega_j \in S} sim(\omega_i, \omega_j)] \tag{16.1}$$

where r_{ω_i} is the estimated rating of item ω_i, S is the subset of already selected items, λ is the parameter to adjust the trade-off between relevance and diversity and $sim(\cdot)$ is the similarity function between two items. We notify that for when applying recommendation methods with MMR diversification, the $sim(\cdot)$ function is selected according to the recommendation method. More specifically, for BPRMF and RCF, we use *cosine_similarity* of item latent vectors to replace the $sim(\cdot)$ and for translation-based knowledge graph representations, we use translation function $\|h - t\|$ (h and t to represent two item entities) to replace the $sim(\cdot)$ function.

16.1.2 Diversity Metrics

For DivKG, we assume the item latent vectors can capture the dissimilarity relation among each other well, i.e. the larger the similarity value of the item latent vectors the less diverse the corresponding items are. Therefore, to assess the diversity of the recommendation, we use two pairwise-based metrics defined on the item list used by [100], where S_{ij} denotes the similarity between i and j:

$$\text{ILAD} = \underset{u \in U}{\text{mean}} \underset{i,j \in R_u, i \neq j}{\text{mean}} (1 - S_{i,j}) \tag{16.2}$$

, and

$$\text{ILMD} = \underset{u \in U}{\text{mean}} \min_{i,j \in R_u, i \neq j} (1 - S_{i,j}) \tag{16.3}$$

. We calculate ILAD and ILMD for each result list and report the average score.

16.1.3 Experimental Results

Table 16.1: Accuracy results before diversification with dimension=75, learning rate=0.001 on dataset ML100K-IMDb.

	Hit			NDCG		
Metric	@5	@10	@20	@5	@10	@20
BPRMF	0.1394	0.2200	0.3240	0.0888	0.1150	0.1412
FISM	0.1182	0.2041	0.3160	0.0746	0.1023	0.1304
RCF	0.1442	0.2179	0.3261	0.0888	0.1123	0.1393
TransE	0.1879	0.2842	0.4087	0.1253	**0.1561**	0.1876
TransH	**0.1917**	**0.2861**	**0.4123**	**0.1257**	**0.1561**	**0.1878**

Recommendation Results for Applying Knowledge Graph Embedding Methods

To fairly compare the performance of different embedding methods, we train knowledge graph embedding by optimizing the margin loss with mini-batch Adam optimization [114]. The learning rate range is set as 0.001, 0.005, 0.01 and the batch size is set as 972 for ML-100K. The embedding size ranges from 50, 75, 100, 150. We have performed a grip search for finding the best results for each method. And in Table 16.1 we demonstrate the accuracy results of the methods used in DivKG (TransE and TransH) against baseline methods on metrics MRR and NDCG before diversification.

Table 16.2: Diversified recommendation results with α=0.9 on dataset ML100K-IMDb.

Metric	Hit@10	NDCG@10	ILAD	ILMD
BPRMF+MMR	0.2821	0.0789	0.9683	0.8867
BPRMF+FastDPP	0.3065	0.0848	0.9922	0.9726
RCF+MMR	0.2842	0.0785	0.9698	0.8886
RCF+FastDPP	0.3001	0.0798	0.9923	0.9729
TransE+MMR	0.2768	0.0774	0.9911	0.9183
DivKG$_E$	0.3160	0.1176	**0.9959**	**0.9768**
TransH+MMR	0.2693	0.0705	0.9898	0.8951
DivKG$_H$	**0.3175**	**0.1178**	0.9956	0.9690

Table 16.1 shows a general accuracy improvement using translation-based knowledge graph embedding regarding user-item interactions as one kind of relations. The two translation-based embedding methods we use here, TransE and TransH, outperform not only classic BPR-based CF method (BPRMF) and item-based matrix factorization method (FISM), but also outperform the start-of-art relation collaborative filtering method RCF which also takes relation information into account for accuracy enhancement. We attribute this lower performance of RCF to both the separation of user-item interactions and other relations in the knowledge graphs and its fixed types of relations encoded on the knowledge graph. Moreover, TransH generally outperforms TransE due to the projection of entity vectors to a relation-specific hyperplane which enhances the accuracy.

Diversified Recommendation

Table 16.2 shows diversified recommendation performance w.r.t. both accuracy and diversity metrics. It can be seen that our methods DivKG$_E$ and DivKG$_H$ that combine FastDPP with TransE and TransH respectively outperform baselines methods w.r.t. both, accuracy and diversity. Almost all accuracy-based methods combined with FastDPP outperform those combined with MMR.

16.2 Experiments on EMDKG

16.2.1 Baselines

We consider both diversified and not diversified baselines to evaluate EMDKG in terms of diversity, accuracy and diversity-accuracy trade-off:

- **BPRMF** [48] is a matrix factorization approach that uses a pairwise ranking loss to provide recommendations. Similar to EMDKG, it considers implicit feedback. However, it does not focus on diversity, nor it uses relational information from knowledge graph. We have used it as a baseline for evaluating DivKG.

- **FISM** [49] is an item-based collaborative filtering method that considers a discount factor for the item and user latent vector multiplication. We also apply the pairwise ranking loss criterion BPR-OPT for the optimization of this method. We adopt and modify the implementation provided by [19]. We have also used it as a baseline for evaluating DivKG.

- **TransKG$_{[.]}$**] [101] models users, items and all the associated entities in a knowledge graph, then uses the embedded vectors obtained with translation-based KG embedding, Trans$_E$ [14] (TransKG$_E$) or Trans$_H$ [15] (TransKG$_H$), to give the result. This actually corresponds to the knowledge graph embedding part in our previous model DivKG.

- **IRGAN**[1] [50] is a framework that makes compete a discriminative and a generative model in an adversarial way to solve several IR problems, including top-N recommendation.

- **RCF**[2] [19] is a hybrid method that combines item-based CF approach with knowledge graph embedding method to jointly learn a user-item preference model and an item-item relational data model. Specifically in user-item preference model, they use a two-level hierarchy attention mechanism to capture the interactions between user embedding and relation types and the weights between users and historical items with specific relation values.

These baselines have not been designed to promote diversity. We use them to evaluate the accuracy of our approach before applying diversification to the recommended list. To ensure a fair comparison when evaluating diversification ability, we adopt a similar approach as our DivKG. We thus combine each algorithm with two baselines diversification techniques, namely **MMR** [29] and the method from [100] that we denote **FastDPP**. MMR is a well-known re-ranking approach to promote diversity in recommendation, that iteratively re-ranks items to adjust the trade-off between accuracy and diversity. FastDPP promotes diversity by re-ranking the results using DPP and MAP inference. In total, we obtain 12 diversified baselines.

Finally, we also compare EMDKG with two recent baselines that consider DPP to provide diversity-promoting recommendations:

DivKG [101] (our proposal discussed above) is a re-ranking approach exploiting multiple relations of items using KGE to provide recommendations. It combines TransKG with FastDPP to enhance the diversity of the result list, but it does not explicitly encode diversity into the representations.

[1] The source code for IRGAN: https://github.com/geek-ai/irgan
[2] https://github.com/XinGla/RCF

PD-GAN [95] is a diversified recommendation method which combines determinantal point processes with GAN [57] framework to generate personalized and diverse recommendations without using the relations between items and other entities or knowledge graphs.

For reproducibility sake, we provide our source code[3].

16.2.2 Performance Metrics

To assess the accuracy of the proposed method and the baselines, we consider standard metrics widely used in the field (e.g. [49, 50]), namely Hit Ratio (Hit@n) and Normalized Discounted Cumulative Gain (NDCG@n). We take $n = \{5, 10, 20\}$.

As we explicitly consider categorical information for item diversity we evaluate diversity performance of recommendation results also using categorical information based metrics. Similar to [95, 103, 115], we evaluate the diversity through these two metrics, namely

1. Category Coverage (CC@n) given by:

$$\text{CC@}n = \frac{1}{|U|}\sum_{u=1}^{|U|}\frac{|C_u^n|}{|C|} \tag{16.4}$$

2. α-NDCG@n [105] given by:

$$\alpha\text{-NDCG@}n = \frac{1}{|U|}\sum_{u=1}^{|U|}\frac{\alpha\text{DCG}_u\text{@}n}{\alpha\text{IDCG}_u\text{@}n} \tag{16.5}$$

where

$$\alpha\text{DCG}_u\text{@}n = \sum_{k=1}^{n}\frac{\sum_{l=1}^{L} J_{kl}^u(1-\alpha)^{q_{l,k-1}^u}}{log_2(1+k)} \tag{16.6}$$

with J_{kl}^u equal to the rating of the kth item in the list for user u if kth item belongs to the genre l otherwise 0. $q_{l,k-1}^u$ counts the number of items belonging to genre l up to the $k-1$ position in the list, which accompanying the constant α to modify the redundancy in the recommendation list. $\alpha\text{IDCG}_u\text{@}n$ denotes the largest value of $\alpha\text{DCG}_u\text{@}n$ which achieves the ideal diversification of recommendation lists. For the length n of recommendation list we take $n = \{5, 10, 20\}$.

16.2.3 General results

First, we evaluate the modified knowledge graph embedding part of EMDKG against baselines (before applying DPP MAP inference diversification). Table 16.3 shows the results of accuracy and diversity for each method on ML100K-IMDb dataset. We can

[3] https://github.com/LGanShare/EMDKG_WI.git

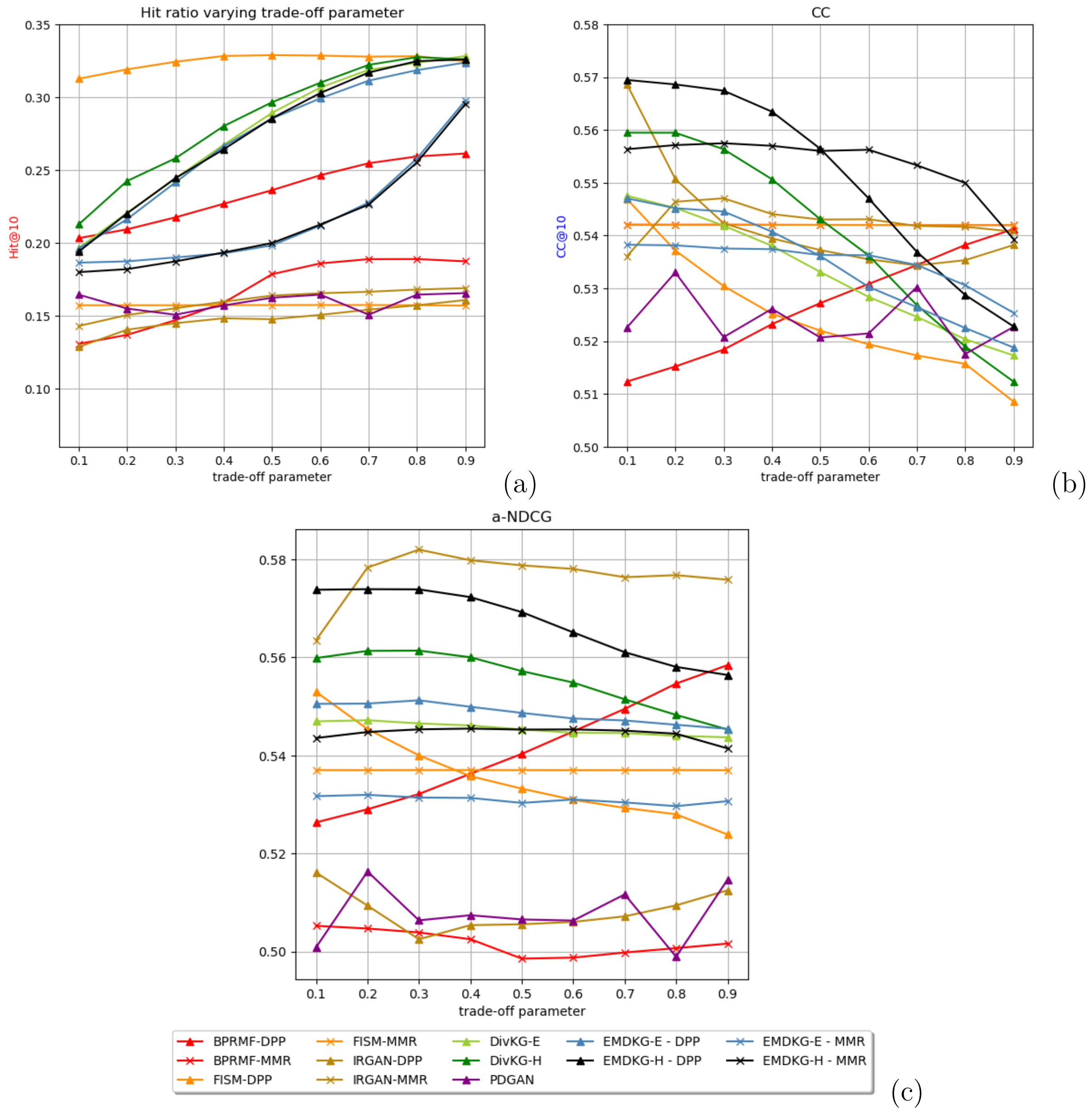


Figure 16.1: Comparison of EMDKG against state-of-the-art approaches on accuracy and diversity metrics when varying the trade-off parameter α on dataset ML100K-IMDb. (a) hit ratio@10 , (b) category coverage (CC)@10, (c) α-NDCG@10.

see that EMDKG-E and EMDKG-H perform much better for all accuracy and diversity metrics compared to BPRMF. Compared to FISM, although its performance in term of accuracy shows a slight advantage (not statistically significant), EMDKG-E and EMDKG-H bring a significant improvement in diversity. Although IRGAN and RCF show higher values in terms of CC@10 and CC@20, EMDKG-E and EMDKG-H still defeats these two methods in other diversity metrics and largely in accuracy. Compared to TransKGs, it is observed that TransKG$_H$ outperforms EMDKG-E and EMDKG-H in terms of accuracy but EMDKG-H compensates it by an obvious gain in diversity w.r.t. all diversity metrics. Besides, EMDKG-E shows both improvements in accuracy and diversity comparing to TransKG$_E$.

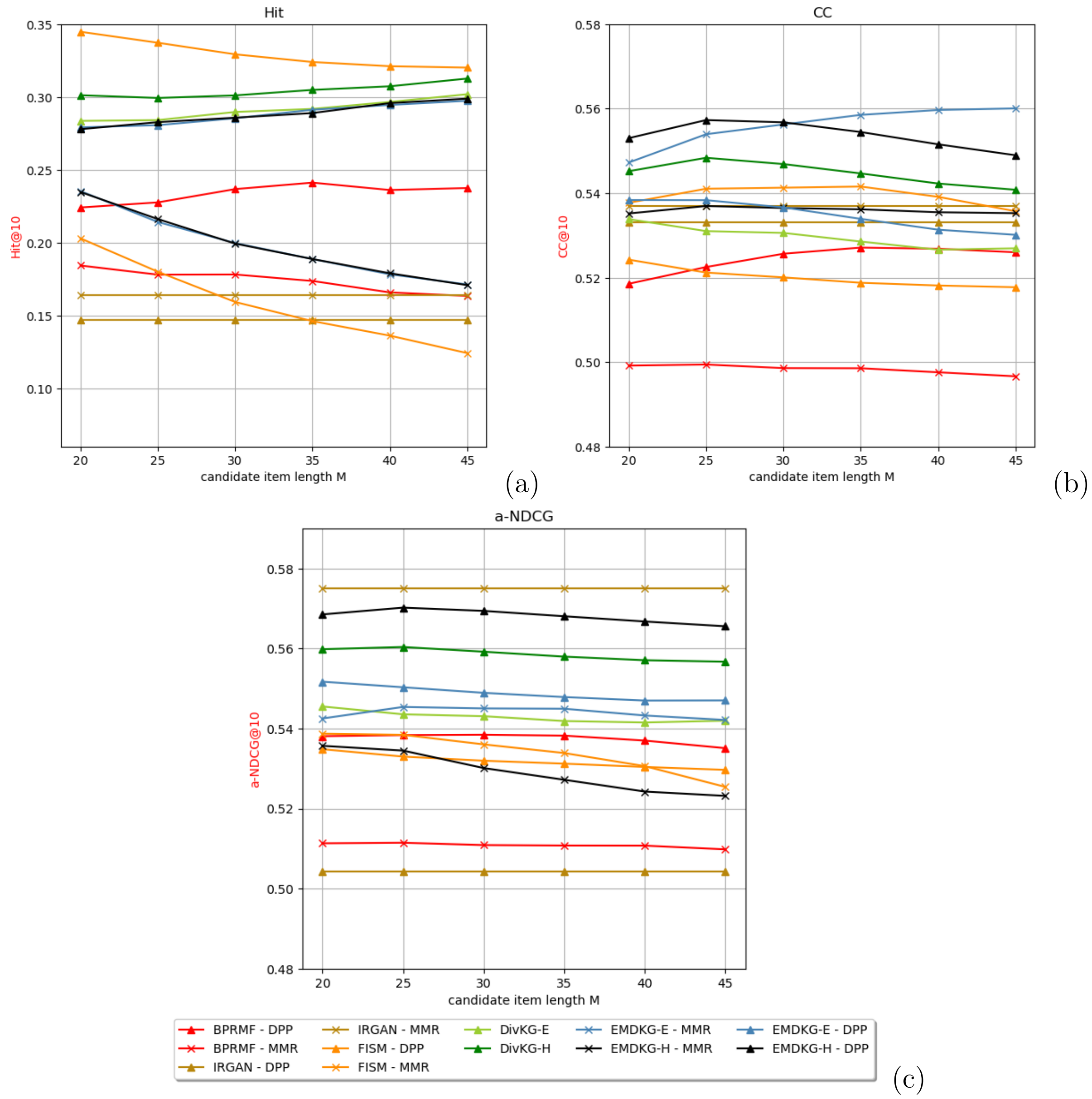


Figure 16.2: Comparison of EMDKG against state-of-the-art approaches on accuracy and diversity metrics when varying the candidate item list length (trade-off parameter: 0.5) on dataset ML-100K. (a) hit ratio@10 , (b) category coverage (CC)@10, (c) α-NDCG@10.

Table 16.4 shows the results of accuracy and diversity on dataset Anime. Our proposals EMDKG-E and EMDKG-H outperform BPRMF and FISM in all metrics of accuracy and diversity. Compared to IRGAN, although IRGAN show comparable results or slightly better results in terms of accuracy, EMDKG-E and EMDKG-H win with a margin for most diversity metrics except for CC@20. Compared to TransKG$_E$ and TransKG$_H$, EMDKG-E and EMDKG-H respectively have outperformed on accuracy and diversity. We have conducted pairwise t-test to confirm the result difference with confidence 99%.

Table 16.3: Accuracy and diversity results before diversification method for `ml100k` extended datasets. Bold results are significantly higher than other results in the same column with $p = 0.01$.

Metrics	Hit (%)			NDCG (%)			CC (%)			α-NDCG (%)		
	@5	@10	@20	@5	@10	20	@5	@10	@20	@5	@10	20
BPRMF	8.91	17.71	32.03	5.51	8.34	11.93	34.96	49.51	65.32	39.78	51.61	62.96
FISM	22.87	31.44	42.82	15.47	18.23	21.09	36.57	50.21	63.47	39.73	51.73	62.31
IRGAN	10.55	16.50	23.45	6.99	8.91	10.65	37.17	53.26	69.18	37.55	51.92	64.74
RCF	12.20	19.51	29.59	7.51	9.86	12.39	37.14	53.40	69.88	38.60	52.71	65.86
TransKG$_E$	22.07	32.43	46.49	14.25	17.57	21.13	36.18	50.87	65.73	41.83	53.85	65.14
TransKG$_H$	**23.38**	**34.36**	**47.01**	**15.70**	**19.24**	**22.44**	36.05	50.93	65.65	41.50	53.77	64.95
EMDKG-E	22.12	33.65	46.03	14.59	17.97	21.32	36.45	51.22	66.24	42.00	54.23	65.58
EMDKG-H	22.08	33.01	46.34	14.08	17.60	20.95	**37.34**	52.15	66.67	**43.27**	**55.50**	**66.66**

Table 16.4: Accuracy and diversity results before diversification method for `anime` datasets. Bold results are significantly higher than other results in the same column with $p = 0.01$.

Metrics	Hit (%)			NDCG (%)			CC (%)			α-NDCG(%)		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
BPRMF	8.13	11.56	17.81	5.13	6.25	7.83	22.95	34.23	46.46	23.95	32.06	40.21
FISM	11.72	17.03	22.97	7.29	8.98	10.48	24.95	37.31	50.43	25.63	34.26	42.97
IRGAN	14.84	19.38	24.06	9.79	11.30	12.46	26.28	35.63	**52.43**	27.97	34.89	44.95
RCF	13.02	16.37	19.22	7.46	9.01	11.20	26.80	38.99	51.97	28.10	34.94	45.33
TransKG$_E$	14.38	20.00	25.63	9.37	11.00	12.54	26.64	39.56	52.09	29.41	38.47	47.30
TransKG$_H$	13.44	20.00	26.88	9.06	11.02	12.89	26.70	39.77	52.21	29.52	38.66	47.31
EMDKG-E	14.84	**20.16**	26.25	9.72	**11.42**	12.89	26.65	39.40	51.50	29.14	38.08	46.77
EMDKG-H	14.84	19.22	**27.34**	9.64	11.05	**13.08**	**27.00**	**40.66**	51.60	29.49	**38.94**	47.14

16.2.4 Impact of trade-off parameter α.

We show the impact of parameter α in Figure 16.1. We choose Hit@10 as accuracy metric and CC@10 and α-NDCG@10 as diversity metrics for each method combined with one of diversification methods MMR or FastDPP. And we demonstrate the results on these three metrics respectively in Figure 16.1(a), Figure 16.1 (b) and Figure 16.1(c). In each chart, we present the results of the corresponding metric for every method by varying the trade-off parameter α. We can see that by increasing the value α, accuracy metrics tend to increase for most methods combined with any of the diversification method while diversity metrics have tendency of increasing in most diversification combinations. In terms of accuracy, EMDKG demonstrates a huge advantage under various α compared to other baseline methods, except for DivKG and FISM. Besides, we can tell that the combinations of EMDKG and FastDPP maintain better their accuracy while decreasing the α compared to those with MMR. While EMDKG and DivKG have comparable results in accuracy, EMDKG improves diversity in terms of both CC and α-NDCG for both diversification methods. Compared to FISM, EMDKG wins with a large margin in terms of both CC@10 and α-NDCG.

In terms of diversity, EMDKG-H shows competitive results in both diversity and accuracy results compared to all other methods. While the combinations of IRGAN+MMR gains advantage of α-NDCG, it also presents much lower accuracy for Hit@10 and lower diversity for CC@10. And compared to DivKG (noted as TransH+DPP) and TransH+MMR, although our proposal does not gain a huge margin in terms of accuracy (still in general no worse than both of them), EMDKG-H +DPP and EMDKG-H +MMR bring obvious improvements for both diversity metrics when varying trade-off parameter α correspondingly.

In particular, we demonstrate the trade-off points for EMDKG methods combined with MMR and FastDPP diversification in Figure 16.3, Figure 16.4, Figure 16.5 and Figure 16.6. Also to demonstrate the enhancing performance we also compare the TransKG methods combined with MMR diversification and DivKG method in the figures. We mark the accuracy results in red and diversity results in blue. In Figure 16.3(up) the trade-off point (between Hit@10 and CC@10) for EMDKG-E - DPP is when trade-off parameter $\alpha \doteq 0.27$, achieving Hit@10 0.23 and CC@10 0.545 while the trade-off point for EMDKG-E - MMR is when trade-off parameter $\alpha \doteq 0.59$ achieving a Hit@10 0.21 and CC@10 0.538. Actually we can see the trade-off point for EMDKG-E - DPP is above the trade-off point for EMDKG-E -MMR, beating both accuracy metric (Hit@10) and diversity metric (CC@10). In a similar fashion, in Figure 16.3(down) the trade-off point of DivKG-E sits above the trade-off point of TransKG-E - MMR, meaning both advantages in accuracy (Hit@10) and diversity metric (CC@10) for DivKG-E. We can observe a similar results in Figure 16.4 when the diversity metric is chosen as α-NDCG. For EMDKG-H and DivKG-H we demonstrate the trade-off points in Figure 16.5 and 16.6, where similar results of achieving better trade-off between an accuracy metric and a diversity metric can be found compared to applying MMR diversification. Furthermore, the trade-off point between EMDKG-E(h) - DPP also is higher than the trade-off point of DivKG-E(H) as the scale of the axis for the charts are the same. This also demonstrate a direct improvement of EMDKG framework over DivKG.

16.2.5 Impact of candidate item set length M.

We show the impact of candidate item set length M for applying diversification methods in Fig. 16.2. The range of candidate item set length M is $\{20, 25, 30, 35, 40, 45\}$. To speed up the re-ranking and keep the accuracy performance, we select the top-M ($M{>}N$) items from the prediction of each recommendation before diversification as candidate items for re-ranking. We can tell from these figures that EMDKG combined with FastDPP achieves stable results for both diversity and accuracy, while the combination with MMR may suffer from a loss of accuracy when increasing the size M of candidate item sets. Besides, EMDKG shows better results when varying the length M compared to DivKG and TransKG+MMR.

16.2.6 Discussion over diverse item set generation

The generation of diverse item set directly influences EMDKG co-learning representation results, and it is thus of importance to discuss this procedure in detail. Three points should be considered.

First, as we have described in Section 12.2, the procedure of diverse item set relies heavily on random selection, therefore to alleviate the randomizing effect, we set the random generation to a very large number (50000 times for each user, inspired by Monte-Carlo methods).

Another specific choice is the selection of the maximum number of diverse sets for each user. We have chosen 100 most diverse item sets for each user for two main reasons:

1. we only consider the most diverse item sets according to a diversity metric for our model. As this part of diversity information for item representation is most informative for learning item latent vectors.

2. for accelerating the process of EMDKG learning, the number of diverse item sets should not be too large.

Finally the diverse item set generation is influenced by the selection of diversity metrics. As we have presented in Section 12.2 the diversity metric can be selected among categorical-information based diversity metrics, e.g. CC, α-NDCG. We have conducted model learning with diverse item sets generated by both diversity metrics (CC and α-NDCG) and the reported results align with each other. Thus in figures shown above, we shown the results with diverse item sets generated with diversity metric CC.

17 Conclusions

In this part, we test two frameworks to address diversity-aware top-N recommendation problem. Our first proposal DivKG combines knowledge graph embedding methods (KGE) and DPP models for diversified prediction. The motivation behind using knowledge graphs lies in their ability to capture various relations between items, users, and auxiliary entities, providing a solid basis for understanding user's behaviour and improving recommendation quality. Moreover, knowledge graphs may facilitate the reasoning behind the recommendation process, making it more convincing. We leave this direction for our future work. In order to diversify the results of top-N recommendation, we further test our method on datasets to construct DPP kernels over KGE to facilitate diversified predictions. The construction of kernel provides an accuracy-diversity trade-off. Our evaluation results prove that such a combination is beneficial in terms of both accuracy and diversity.

In our second proposal EMDKG, we aim at learning representations for top-N recommendation to achieve better accuracy-diversity trade-off. From our perspective, the latter is not limited to only achieving both high accuracy and diversity, but should also be robust under different parameter settings. Thus, we propose a novel EM-schemed diversity-encoded knowledge graph embedding model EMDKG which incorporates Item Diversity Learning and Knowledge Graph Embedding for this purpose. We compare EMDKG with multiple state-of-art baseline methods before and after applying two diversification methods MMR and DPP. The results show that before diversification EMDKG can adjust accuracy and diversity to a better trade-off and after diversification EMDKG can outperform the baselines when varying the trade-off parameters and the candidate item set length. In all, EMDKG demonstrates better performance in terms of accuracy-diversity trade-off compared to competitive state-of-art works.

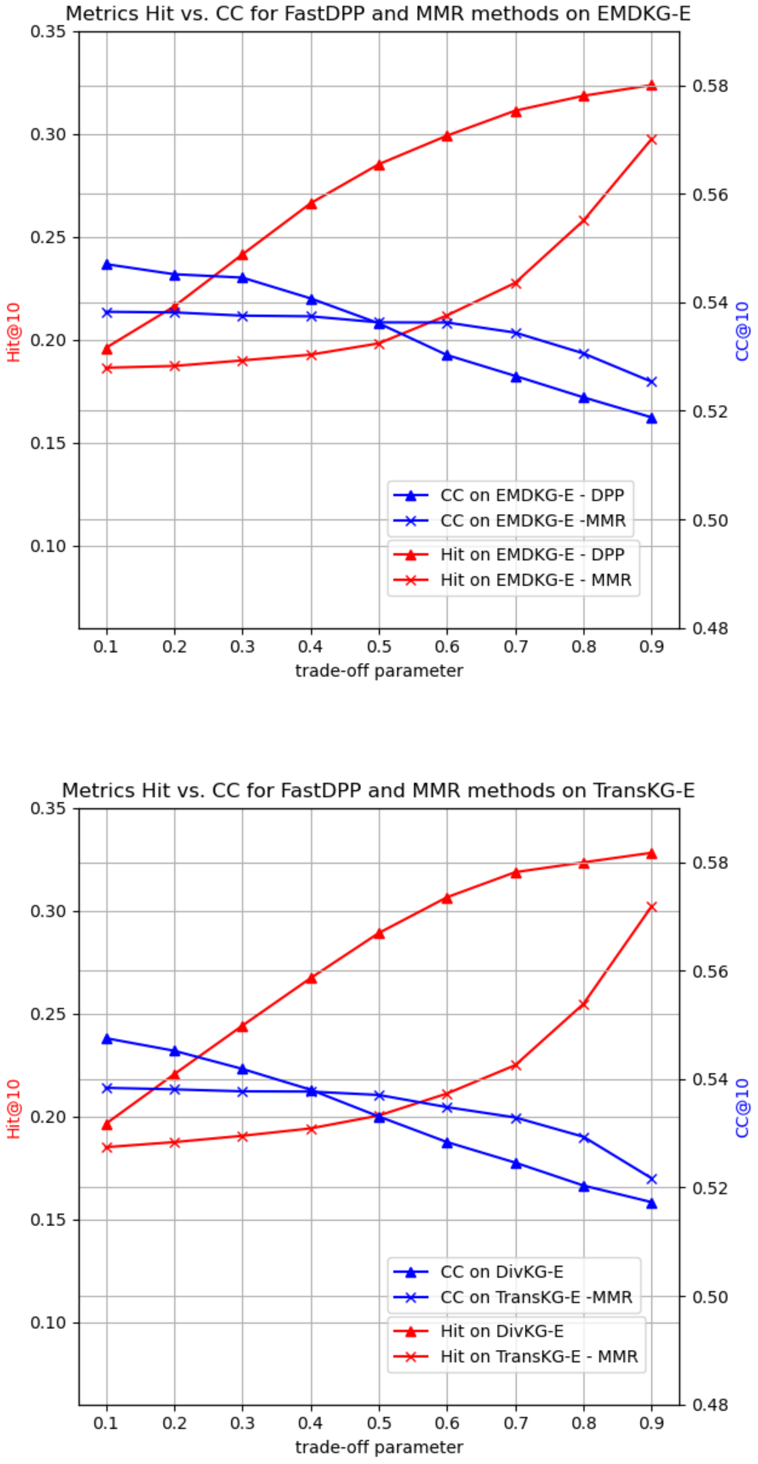


Figure 16.3: Pairwise comparison of EMDKG-E - DPP and EMDKG-E - MMR on accuracy and diversity metrics

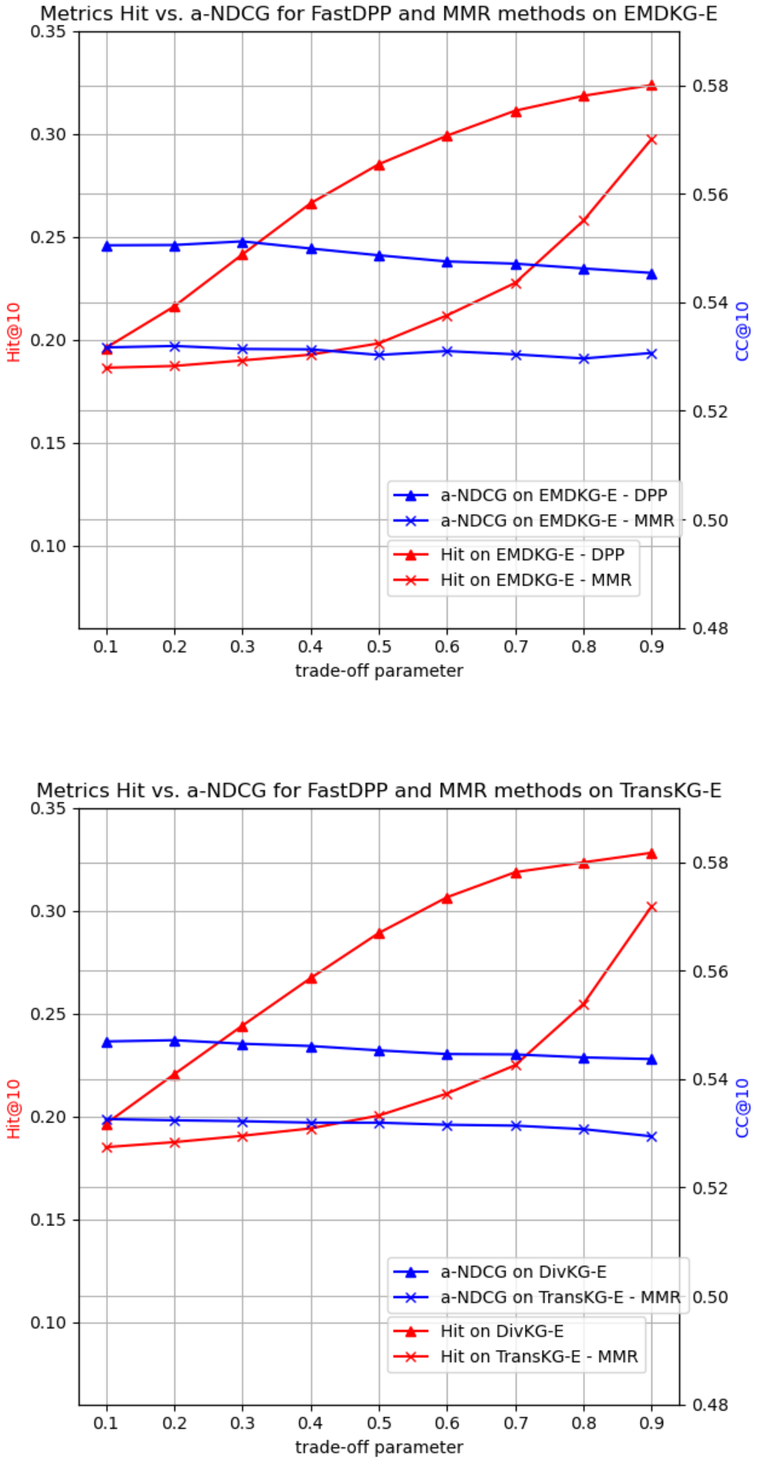


Figure 16.4: Pairwise comparison of DivKG-E and TransKG-E - MMR on accuracy and diversity metrics

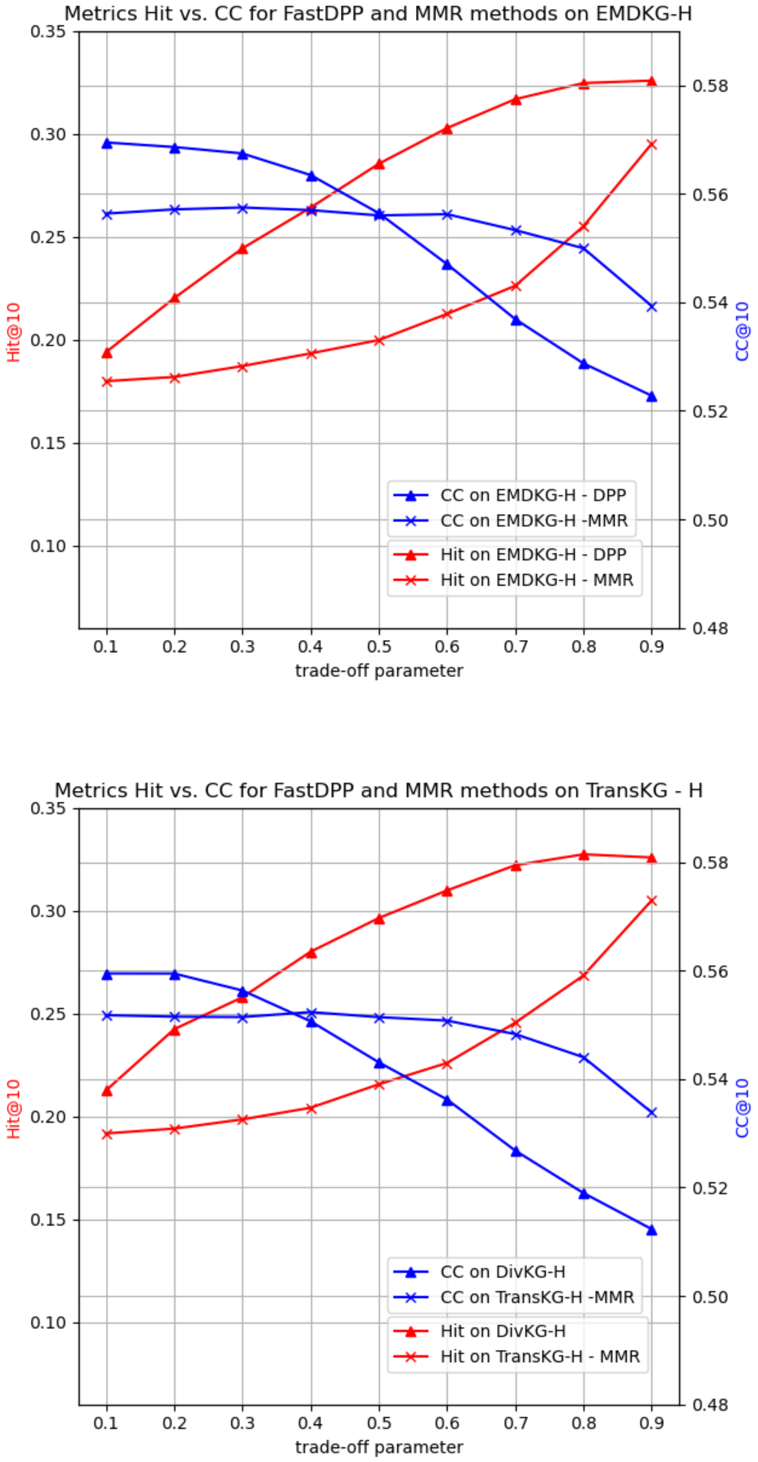
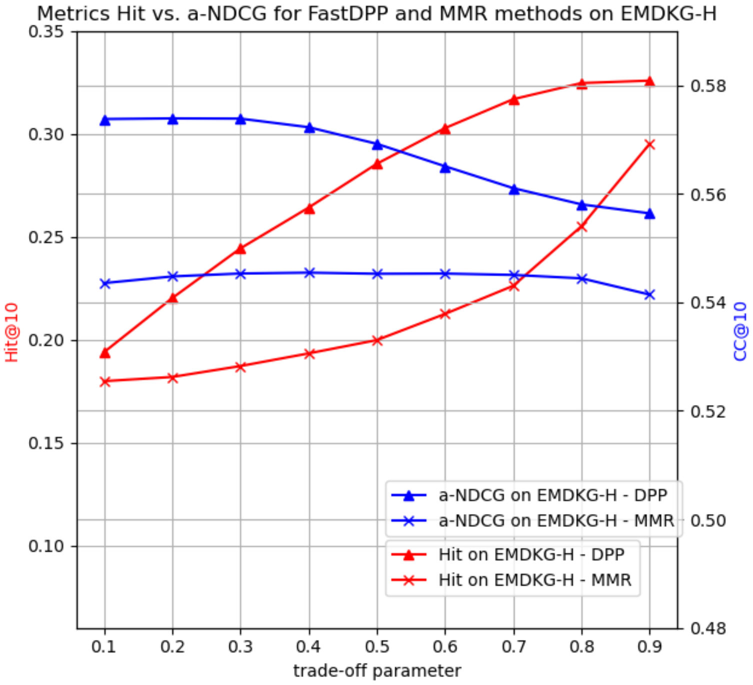


Figure 16.5: Pairwise comparison of EMDKG-H - DPP and EMDKG-H - MMR on accuracy and diversity metrics



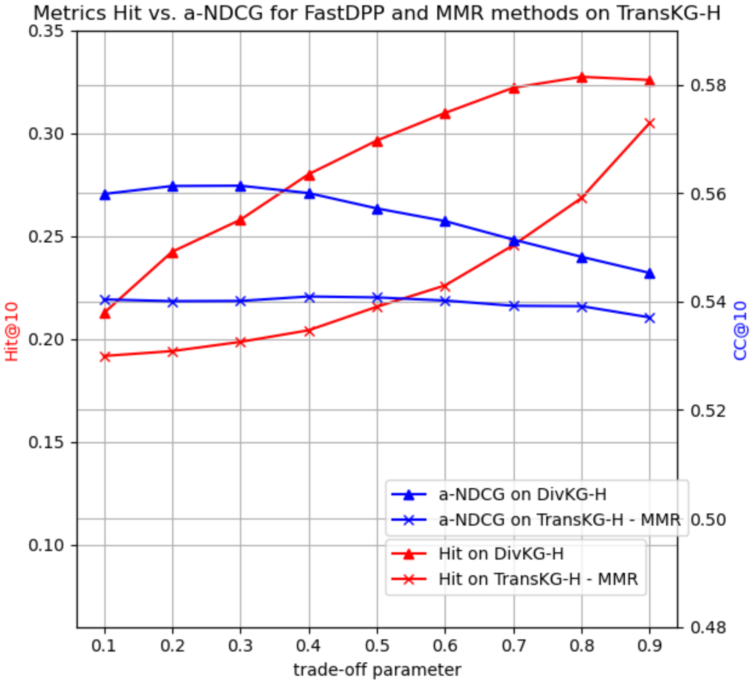


Figure 16.6: Pairwise comparison of DivKG-H and TransKG-H - MMR on accuracy and diversity metrics

Part V

Conclusion and Research Directions

18 Conclusion 98

19 Future Directions 99

Bibliography 103

18 Conclusion

Recommender systems have been prevalent for three decades and still remain among the active research topics. As accuracy-centered recommendations are not aligned completely with user satisfaction in reality, diversity has been brought in recommendation as anther goal to alleviate redundancy and filter bubble effects. In this dissertation, we concentrate on a diversified top-N recommendation problem, aiming at achieving recommendation with both high accuracy and high diversity for each user. With recent advancements in knowledge graph embedding methods where multi-type relations are mapped into latent vector spaces, capturing good graph structures and knowledge facts, we transfer this category of approaches into recommendation tasks. In the mean time, determinantal point processes have been widely studied and applied in various machine learning tasks for result diversification. Its natural advantage of both capturing item affinity and item diversity in the accompanying kernel matrix inspires us to incorporate it into knowledge graph embedding for achieving a both accurate and diverse top-N recommendation.

Therefore, in this dissertation we present our two contributions for diversified recommendations based on knowledge graph embedding methods and determinantal point processes. Our first diversified recommendation model DivKG combines knowledge graph embedding methods (KGE) and DPP models for diversified prediction. It proposes a novel approach to construct personalized DPP kernel matrix based on translation-based knowledge graph embedding for diversified top-N item lists. The new approach for the personalized DPP kernel matrix is a linear combination of user-item affinity matrix and item-item similarity matrix, both derived of entity and relation vectors from knowledge graph embedding representations. The motivation behind using knowledge graphs lies in their ability to capture various relations between items, users, and auxiliary entities, providing a solid basis for understanding user's behaviour and improving recommendation quality.

As we assume the item vectors learnt from knowledge graph embedding can capture well the correlation with other items in DivKG, our second diversified recommendation EMDKG however does not rely on this assumption and argues that item vectors solely based on knowledge graph embedding can not capture good enough diversity of item lists. Thus, we propose two module co-learning framework for learning diversity-aware knowledge graph embedding for top-N recommendation and suggest a learning procedure to optimize the representations.

We then test our models DivKG and EMDKG on real life datasets to prove the effectiveness of our proposals. The empirical results of DivKG shown in Section 16.1 demonstrate the capability of knowledge graph embedding methods to enhance recommendation accuracy. In order to diversify the results of top-N recommendation, we further test on datasets to construct DPP kernels over KGE to facilitate diversified predictions. The construction of kernel provides an accuracy-diversity trade-off. Our experimental results shown in Section 16.1 prove that such a combination is beneficial in terms of both accuracy and diversity.

For validating EMDKG, we evaluate whether the co-learning framework for top-N recommendation can achieve better accuracy-diversity trade-off. In our perspective, the latter is not limited to only achieve both high accuracy and diversity, but should also be robust under different parameter settings. We therefore compare EMDKG with multiple state-of-art baseline methods before and after applying two diversification methods MMR and FastDPP. The results in Section 16.2 show that before diversification EMDKG can adjust accuracy and diversity to a better trade-off and after diversification EMDKG can outperform the baselines when varying the trade-off parameters and the candidate item set length. In all, EMDKG demonstrates better performance in terms of accuracy-diversity trade-off compared to competitive state-of-art works.

19 Future Directions

For future directions, we first emphasize that there are still many interesting topics in diversified recommendation tasks, despite our contributions in this domain.

The first direction lies in extending the current offline evaluation to *online evaluation* using A/B testing [116]. In our evaluation for DivKG and EMDKG, we use offline evaluation, consisting of first using collected datasets, then dividing the datasets into training/validation/testing parts and using training and validation datasets for finding the optimal model and finally measuring the performance on testing datasets. The offline evaluation procedure can measure the recommendation performance in a rather easy-to-conduct and repeatable fashion and is widely used in academics. In contrast, online evaluation is preferred by industrial practitioners as they argue it will reflect the true utility of recommendation algorithms. Especially, when considering multiple-goal recommendation, it is significant to verify how the multiple goals affect the de-facto CTR performance.

The second direction is to add *aggregate diversity* as another goal for diversified recommendation. As we have briefly introduced aggregate diversity in Section 8.1, aggregate diversity targets at offering a more balanced recommendation of items in the system and can alleviate effectively the long-tail effect. Items which are seldom interacted in history may still be of great value to a particular group of users, therefore achieving additional goal - aggregate diversity will uncover the potentials of these items on the "tail". Thus, a multi-task recommendation regarding accuracy, individual diversity and aggregate diversity is also of great concern and will bring benefit to both users and items in the recommender systems.

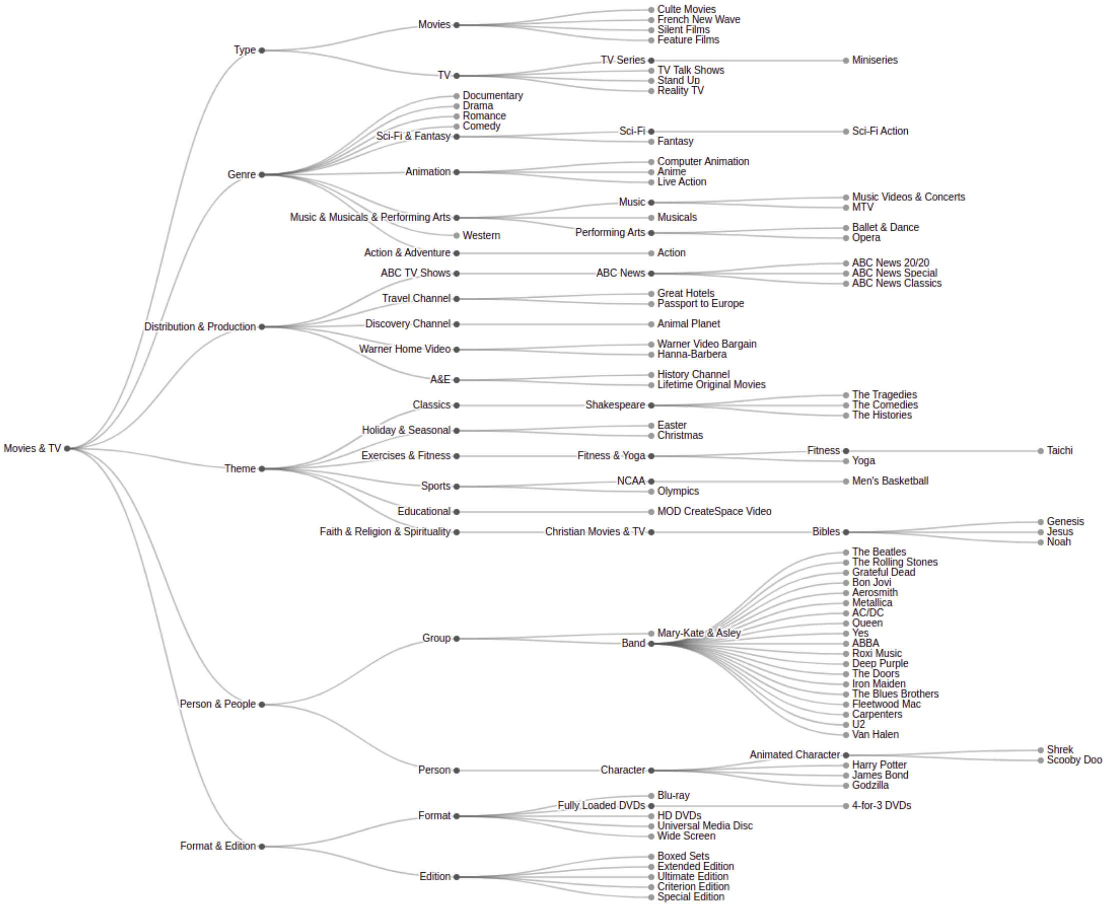


Figure 19.1: Categorical tree extracted from categorical information on dataset Amazon Movie & TV.

Another direction of potential future work is *explainability* in recommender systems. There is a recent burst of research interest for explainable AI as intelligent services offered to users should gain trust through getting more transparent and explainable. Thus, for recommendation tasks which also rely hugely on user satisfaction including understanding how the recommending mechanism works, it seems rather apparent that explainability in recommender system should be the trend. Actually there exist already several streams of explainable recommendations in recent literature [117–122].

A trendy solution to provide explanations for recommendation is to use textual review data [118, 119] as opinions can be extracted from textual reviews and by jointly learning an accuracy-based recommendation and a NLP classification a recommendation with viewpoints can be generated. Social relation information have also been used to provide explanations in recommendation settings [117, 118]. Apart from textual information and social relations, structural information are also taken into account in some works [120, 123, 124] for generating explanations. Structural information or more particularly tree-structure can help generate fine-grained explanations in different explanation styles (statement, comparative).

In Figure 19.1, we demonstrate such a hierarchical tree structure characterizing categorical information on dataset Amazon Movie & TV [1]. However such hierarchical structure requires heavy manual work as there exist no effective and accurate solutions to auto-generate such information. On the other hand, graphical structures including knowledge graphs are omnipresent and conserve a fine-grained explanation potentials as tree structure is one particular type of graphs. Furthermore, knowledge graphs naturally capture explainable units (the relational triplets) which is specially advantageous for a explainable recommendation. However, using graph structure to explain recommendations faces a huge challenge which lies on the complexity of graph structure itself, particularly when applying to a real-world scenario.

In Figure 19.2, we demonstrate an auto-generated category graph based on item categorical information on dataset Amazon Beauty. Amazon Beauty is a comparatively small dataset with already more than 100 categories (many of them overlap with others). Therefore, to leverage the graph information for recommendation requires an intelligent method to pinpoint non-overlapping elements (nodes and links) on the graph but we argue that the capability of graphs for conserving human-understandable knowledge weighs more and motivate us for finding a good solution to provide a more transparent and explanation recommendation.

[1] The Amazon dataset can be found here: https://nijianmo.github.io/amazon/index.html

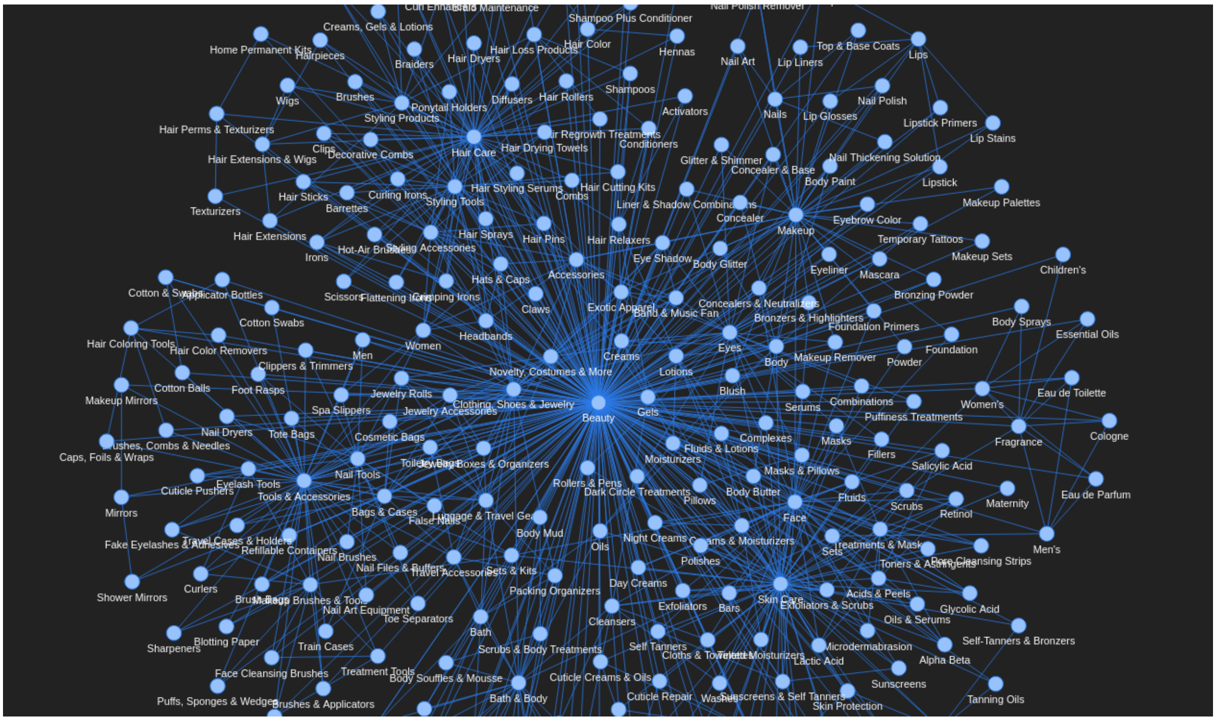


Figure 19.2: The category graph based on item categorical information on dataset Amazon Beauty.

Bibliography

[1] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on web usage mining. *Commun. ACM*, 43(8):142–151, aug 2000.

[2] Hung-Chen Chen and Arbee L. P. Chen. A music recommendation system based on music data grouping and user interests. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, page 231–238, New York, NY, USA, 2001. Association for Computing Machinery.

[3] Fang-Fei Kuo, Meng-Fen Chiang, Man-Kwan Shan, and Suh-Yin Lee. Emotion-based music recommendation by association discovery from film music. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, page 507–510, New York, NY, USA, 2005. Association for Computing Machinery.

[4] Christian Bomhardt. Newsrec, a svm-driven personal recommendation system for news websites. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '04, page 545–548, USA, 2004. IEEE Computer Society.

[5] Paula Cristina Vaz, David Martins de Matos, Bruno Martins, and Pavel Calado. Improving a hybrid literary book recommendation system through author ranking. In *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '12, page 387–388, New York, NY, USA, 2012. Association for Computing Machinery.

[6] Neel Sundaresan. Recommender systems at the long tail. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, page 1–6, New York, NY, USA, 2011. Association for Computing Machinery.

[7] Da Cao, Xiangnan He, Lianhai Miao, Yahui An, Chao Yang, and Richang Hong. Attentive group recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 645–654, New York, NY, USA, 2018. Association for Computing Machinery.

[8] C. J. van Rijsbergen. Information retrieval: New directions: Old solutions. In *Proceedings of the 6th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '83, page 264–265, New York, NY, USA, 1983. Association for Computing Machinery.

[9] Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, page 222–229, New York, NY, USA, 1999. Association for Computing Machinery.

[10] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, page 175–186, New York, NY, USA, 1994. Association for Computing Machinery.

[11] Pedro Cano, Markus Koppenberger, and Nicolas Wack. Content-based music audio recommendation. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, page 211–212, New York, NY, USA, 2005. Association for Computing Machinery.

[12] Marko Balabanović and Yoav Shoham. Fab: Content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, mar 1997.

[13] Luis Mateus Rocha. <i>talkmine</i> and the adaptive recommendation project. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, DL '99, page 242–243, New York, NY, USA, 1999. Association for Computing Machinery.

[14] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 2787–2795, 2013.

[15] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1112–1119, 2014.

[16] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[17] Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 4289–4300, Red Hook, NY, USA, 2018. Curran Associates Inc.

[18] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and*

Data Mining, KDD '16, page 353–362, New York, NY, USA, 2016. Association for Computing Machinery.

[19] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon Jose. Relational collaborative filtering: Modeling multiple item relations for recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 125–134, 2019.

[20] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, page 22–32, New York, NY, USA, 2005. Association for Computing Machinery.

[21] Eli Pariser. *The Filter Bubble: What the Internet Is Hiding from You.* Penguin Group , The, 2011.

[22] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. Exploring the filter bubble: The effect of using recommender systems on content diversity. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, page 677–686, New York, NY, USA, 2014. Association for Computing Machinery.

[23] Bruce Ferwerda, Mark P. Graus, Andreu Vall, Marko Tkalcic, and Markus Schedl. How item discovery enabled by diversity leads to increased recommendation list attractiveness. In *Proceedings of the Symposium on Applied Computing*, SAC '17, page 1693–1696, New York, NY, USA, 2017. Association for Computing Machinery.

[24] Zhijing Wu, Ke Zhou, Yiqun Liu, Min Zhang, and Shaoping Ma. Does diversity affect user satisfaction in image search. *ACM Trans. Inf. Syst.*, 37(3), may 2019.

[25] Saúl Vargas and Pablo Castells. Improving sales diversity by recommending users to items. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, page 145–152, New York, NY, USA, 2014. Association for Computing Machinery.

[26] Farzad Eskandanian and Bamshad Mobasher. *Using Stable Matching to Optimize the Balance between Accuracy and Diversity in Recommendation*, page 71–79. Association for Computing Machinery, New York, NY, USA, 2020.

[27] Farzad Eskandanian and Bamshad Mobasher. *Using Stable Matching to Optimize the Balance between Accuracy and Diversity in Recommendation*, page 71–79. Association for Computing Machinery, New York, NY, USA, 2020.

[28] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. Challenging the long tail recommendation. *Proc. VLDB Endow.*, 5(9):896–907, may 2012.

[29] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336, 1998.

[30] Allan Borodin, Aadhar Jain, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions, and dynamic updates. *ACM Trans. Algorithms*, 13(3), 2017.

[31] Alex Kulesza and Ben Taskar. k-dpps: Fixed-size determinantal point processes. In *Proceedings of the 28th International Conference on Machine Learning (ICML'11)*, pages 1193–1200, 2011.

[32] Alex Kulesza and Ben Taskar. Learning determinantal point processes, 2012.

[33] Peter W. Foltz and Susan T. Dumais. Personalized information delivery: An analysis of information filtering methods. *Commun. ACM*, 35(12):51–60, dec 1992.

[34] Pasquale Lops, Marco Degemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, 2011.

[35] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317, 1957.

[36] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[37] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, page 289–296, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[38] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, mar 2003.

[39] Ken Lang. Newsweeder: Learning to filter netnews. In *in Proceedings of the 12th International Machine Learning Conference (ML95*, 1995.

[40] Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, page 195–204, New York, NY, USA, 2000. Association for Computing Machinery.

[41] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. *Content-based Recommender Systems: State of the Art and Trends*, pages 73–105. Springer US, Boston, MA, 2011.

[42] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[43] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, page 210–217, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[44] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, page 285–295, New York, NY, USA, 2001. Association for Computing Machinery.

[45] Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 46–54, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[46] Ken Goldberg, Theresa Roeder, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4:133–151, 2001.

[47] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 1257–1264, 2007.

[48] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, 2009.

[49] Santosh Kabbur, Xia Ning, and George Karypis. Fism: Factored item similarity models for top-n recommender systems. In *Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 659–667, 2013.

[50] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proc. of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 515–524, 2017.

[51] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, page 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[52] Kai Yu, Xiaowei Xu, Jianjua Tao, Martin Ester, and Hans-Peter Kriegel. Instance selection techniques for memory-based collaborative filtering. In *SDM*, 2002.

[53] Mukund Deshpande and George Karypis. Item-based top-<i>n</i> recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, jan 2004.

[54] P. C. Hammer. Adaptive control processes: A guided tour (r. bellman). *SIAM Rev.*, 4(2):163, apr 1962.

[55] Simon Funk, Dec 2006.

[56] Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 426–434, New York, NY, USA, 2008. Association for Computing Machinery.

[57] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[58] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, ICUIMC '08, page 208–211, New York, NY, USA, 2008. Association for Computing Machinery.

[59] Mark Claypool, Anuja Gokhale, Tim Miranda, Paul Murnikov, Dmitry Netes, and Matthew M. Sartin. Combining content-based and collaborative filters in an online newspaper. In *SIGIR 1999*, 1999.

[60] Daniel Billsus and Michael J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2–3):147–180, feb 2000.

[61] B Smyth and P Cotter. A personalised tv listings service for the digital tv age. *Knowledge-Based Systems*, 13(2):53–59, 2000.

[62] Chumki Basu, Haym Hirsh, and William Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings*

of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence, AAAI '98/IAAI '98, page 714–720, USA, 1998. American Association for Artificial Intelligence.

[63] Derry O'Sullivan, David C. Wilson, and Barry Smyth. Preserving recommender accuracy and diversity in sparse datasets. In Ingrid Russell and Susan M. Haller, editors, *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference, May 12-14, 2003, St. Augustine, Florida, USA*, pages 139–143. AAAI Press, 2003.

[64] Robin Burke. *Hybrid Web Recommender Systems*, pages 377–408. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[65] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 809–816, Madison, WI, USA, 2011. Omnipress.

[66] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2071–2080, New York, New York, USA, 20–22 Jun 2016. PMLR.

[67] Yexiang Xue, Yang Yuan, Zhitian Xu, and Ashish Sabharwal. Expanding holographic embeddings for knowledge completion. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[68] Ivana Balazevic, Carl Allen, and Timothy Hospedales. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5185–5194, Hong Kong, China, November 2019. Association for Computational Linguistics.

[69] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, July 2015.

[70] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the*

Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, pages 2181–2187, 2015.

[71] Han Xiao, Minlie Huang, Yu Hao, and Xiaoyan Zhu. Transa: An adaptive approach for knowledge graph embedding. *ArXiv*, abs/1509.05490, 2015.

[72] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, page 926–934, Red Hook, NY, USA, 2013. Curran Associates Inc.

[73] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2016.

[74] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks, 2019.

[75] Donghan Yu, Yiming Yang, Ruohong Zhang, and Yuexin Wu. Knowledge embedding based graph convolutional network. In *Proceedings of the Web Conference 2021*, WWW '21, page 1619–1628, New York, NY, USA, 2021. Association for Computing Machinery.

[76] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Dkn: Deep knowledge-aware network for news recommendation, 2018.

[77] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018.

[78] Yuting Ye, Xuwu Wang, Jiangchao Yao, Kunyang Jia, Jingren Zhou, Yanghua Xiao, and Hongxia Yang. Bayes embedding (bem): Refining representation by integrating knowledge graphs and behavior-specific networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 679–688, New York, NY, USA, 2019. Association for Computing Machinery.

[79] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 968–977, New York, NY, USA, 2019. Association for Computing Machinery.

[80] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[81] Bernard J. Jansen and Soo Young Rieh. The seventeen theoretical constructs of information searching and information retrieval. *Journal of the American Society for Information Science and Technology*, 61(8):1517–1534, 2010.

[82] Michele Zanitti, Sokol Kosta, and Jannick Sørensen. A user-centric diversity by design recommender system for the movie application domain. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1381–1389, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[83] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, page 5–14, New York, NY, USA, 2009. Association for Computing Machinery.

[84] Saúl Vargas, Linas Baltrunas, Alexandros Karatzoglou, and Pablo Castells. Coverage, redundancy and size-awareness in genre diversity for recommender systems. In *Proc. of the 8th ACM Conference on Recommender systems*, pages 209–216, 2014.

[85] Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. Optimal greedy diversity for recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 1742–1748. AAAI Press, 2015.

[86] Yisong Yue and Thorsten Joachims. Predicting diverse subsets using structural svms. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1224–1231, New York, NY, USA, 2008. Association for Computing Machinery.

[87] Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 881–890, New York, NY, USA, 2010. Association for Computing Machinery.

[88] Idan Szpektor, Yoelle Maarek, and Dan Pelleg. When relevance is not enough: Promoting diversity and freshness in personalized question recommendation. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 1249–1260, New York, NY, USA, 2013. Association for Computing Machinery.

[89] Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Shuzi Niu. Learning for search result diversification. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, page 293–302, New York, NY, USA, 2014. Association for Computing Machinery.

[90] Gabriele Capannini, Franco Maria Nardini, Raffaele Perego, and Fabrizio Silvestri. Efficient diversification of web search results. *Proc. VLDB Endow.*, 4(7):451–459, apr 2011.

[91] Lijing Qin and Xiaoyan Zhu. Promoting diversity in recommendation by entropy regularizer. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, page 2698–2704. AAAI Press, 2013.

[92] Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. Intent-aware search result diversification. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, page 595–604, New York, NY, USA, 2011. Association for Computing Machinery.

[93] Sha Hu, Zhicheng Dou, Xiaojie Wang, Tetsuya Sakai, and Ji-Rong Wen. Search result diversification based on hierarchical intents. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 63–72, New York, NY, USA, 2015. Association for Computing Machinery.

[94] Alex Kulesza and Ben Taskar. Structured determinantal point processes. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, page 1171–1179, Red Hook, NY, USA, 2010. Curran Associates Inc.

[95] Qiong Wu, Yong Liu, Chunyan Miao, Binqiang Zhao, Yin Zhao, and Lu Guan. Pd-gan: Adversarial learning for personalized diversity-promoting recommendation. In *Proc. of the 28th International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3870–3876, 2019.

[96] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 549–558, 2016.

[97] Mike Gartrell, Ulrich Paquet, and Noam Koenigstein. Bayesian low-rank determinantal point processes. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 349–356, 2016.

[98] Romain Warlop, Jérémie Mary, and Mike Gartrell. Tensorized determinantal point processes for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 1605–1615, 2019.

[99] Marco Tulio Ribeiro, Anisio Lacerda, Adriano Veloso, and Nivio Ziviani. Pareto-efficient hybridization for multi-objective recommender systems. In *Proc. of the 6th ACM Conference on Recommender Systems*, page 19–26, 2012.

[100] Laming Chen, Guoxin Zhang, and Hanning Zhou. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 5627–5638, 2018.

[101] Lu Gan, Diana Nurbakova, Léa Laporte, and Sylvie Calabretto. Enhancing recommendation diversity using determinantal point processes on knowledge graphs. In *Proc. of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2001–2004, 2020.

[102] Jianing Sun, Wei Guo, Dengcheng Zhang, Yingxue Zhang, Florence Regol, Yaochen Hu, Huifeng Guo, Ruiming Tang, Han Yuan, Xiuqiang He, and Mark Coates. A framework for recommending accurate and diverse items using bayesian graph convolutional neural networks. In *Proc. of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 2030–2039, 2020.

[103] Peizhe Cheng and Shuaiqiang Wang. Learning to Recommend Accurate and Diverse Items. In *Proc. of the 26th International Conference on World Wide Web*, page 183–192, 2017.

[104] Zeinab Abbassi, Sihem Amer-Yahia, Laks V.S. Lakshmanan, Sergei Vassilvitskii, and Cong Yu. Getting recommender systems to think outside the box. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, page 285–288, New York, NY, USA, 2009. Association for Computing Machinery.

[105] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proc. of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 659–666, 2008.

[106] Yue Shi, Xiaoxue Zhao, Jun Wang, Martha Larson, and Alan Hanjalic. Adaptive diversification of recommendation results via latent factor portfolio. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, page 175–184, New York, NY, USA, 2012. Association for Computing Machinery.

[107] Jennifer Gillenwater, Alex Kulesza, Zelda Mariet, and Sergei Vassilvtiskii. A tree-based method for fast repeated sampling of determinantal point processes. In

Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2260–2268. PMLR, 09–15 Jun 2019.

[108] Ruining He, Wang-Cheng Kang, and Julian McAuley. Translation-based recommendation. In *Proc. of the Eleventh ACM Conference on Recommender Systems*, page 161–169, 2017.

[109] A. Zheng and A. Casari. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly, 2018.

[110] Mi Zhang and Neil Hurley. Avoiding monotony: Improving the diversity of recommendation lists. In *Proc. of the 2008 ACM Conference on Recommender Systems*, page 123–130, 2008.

[111] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The World Wide Web Conference*, page 151–161, 2019.

[112] Dongho Lee, Byungkook Oh, Seungmin Seo, and Kyong-Ho Lee. News recommendation with topic-enriched knowledge graphs. In *Proc. of the 29th ACM International Conference on Information and Knowledge Management*, page 695–704, 2020.

[113] Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, volume 51, pages 243–250, 2000.

[114] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[115] Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. Selectively diversifying web search results. In *Proc. of the 19th ACM International Conference on Information and Knowledge Management*, page 1179–1188, 2010.

[116] Giordano Tamburrelli and Alessandro Margara. Towards automated a/b testing. 08 2014.

[117] Haekyu Park, Hyunsik Jeon, Junghwan Kim, Beunguk Ahn, and U Kang. Uniwalk: Explainable and accurate recommendation for rating and network data. *ArXiv*, abs/1710.07134, 2017.

[118] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. Social collaborative viewpoint regression with explainable recommendations. In

Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM '17, page 485–494, New York, NY, USA, 2017. Association for Computing Machinery.

[119] Yichao Lu, Ruihai Dong, and Barry Smyth. Why i like it: Multi-task learning for recommendation and explanation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 4–12, New York, NY, USA, 2018. Association for Computing Machinery.

[120] Ninghao Liu, Xiao Huang, Jundong Li, and Xia Hu. On interpretation of network embedding via taxonomy induction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 1812–1820, New York, NY, USA, 2018. Association for Computing Machinery.

[121] Chenliang Li, Cong Quan, Li Peng, Yunwei Qi, Yuming Deng, and Libing Wu. A capsule network for recommendation and explaining what you like and dislike. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 275–284, New York, NY, USA, 2019. Association for Computing Machinery.

[122] Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. Transparent, scrutable and explainable user models for personalized recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 265–274, New York, NY, USA, 2019. Association for Computing Machinery.

[123] Yiyi Tao, Yiling Jia, Nan Wang, and Hongning Wang. The fact: Taming latent factor models for explainability with factorization trees. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 295–304, New York, NY, USA, 2019. Association for Computing Machinery.

[124] Eyal Shulman and Lior Wolf. Meta decision trees for explainable recommendation systems. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, page 365–371, New York, NY, USA, 2020. Association for Computing Machinery.