SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Dipartimento di / Department of

Informatics, Systems and Communication

Dottorato di Ricerca in / PhD program: Computer sciences Ciclo / Cycle XXX

Curriculum in (se presente / if it is)

Topic modeling for scientific paper recommendation

Cognome / Surname: Amami Nome / Name: Maha

Matricola / Registration number: 799490

Tutore / Tutor: Prof. Francesca Arcelli Fontana

Supervisor: Prof. Gabriella Pasi

Supervisor: Prof. Rim Faiz

Coordinatore / Coordinator: Prof. Stefania Bandini

ANNO ACCADEMICO / ACADEMIC YEAR 2017

Acknowledgments

First of all, I want to thank my supervisor, Prof. Gabriella Pasi. I was honored to work with her. Not only does she have an astounding depth of knowledge, but with her example she inspires me. I have learnt so much from Prof Gabriella. Her encouragement, insight and feedback have been invaluable.

I want to thank also my supervisor Prof. Rim Faiz. I am very grateful to her for her tolerance in letting me to pursue my own academic path since without her counsel and early interest I probably would not have done any research at all. She provides me a lot of academic opportunities that would not have otherwise been possible.

I also want to thank Prof. Fabio Stella, whose help was fundamental in my exploration in topic modeling and recommender systems. He gave me important suggestions during my thesis and it was always a pleasure to work with him.

I would also like to thank my lovely parents, Amel and Abdelhamid who supported me throughout.

“Ideas come and go, stories stay.”

Nassim Nicholas Taleb

The Black Swan: The Impact of the Highly Improbable

Contents

Sommario

La raccomandazione di articoli scientifici è un'attività che mira a migliorare l'utilizzo delle biblioteche digitali (Digital Libraries, DL) e aiuta i ricercatori a trovare articoli pertinenti da un ampio insieme di articoli. Tuttavia, devono essere fornite fonti affidabili per modellare gli interessi dei ricercatori in modo da avere raccomandazioni accurate. In questa tesi ci concentriamo sull'estrazione degli interessi degli utenti da articoli a cui l'utente è collegato (di cui è l'autore o che ha valutato), utilizzando anche la struttura sociale della rete accademica dell'utente (le relazioni tra ricercatori nello stesso dominio).

Abbiamo proposto un approccio di filtraggio basato sul contenuto (Content-Based Filtering, CBF) per la raccomandazione di articoli scientifici che si basa su topic modeling: il profilo di un ricercatore è modellato da una serie di argomenti ottenuti applicando il modello LDA (Latent Dirichlet Allocation) agli articoli scientifici dal ricercatore. Il profilo creato da questo modello è facilmente interpretabile e può spiegare i risultati della raccomandazione.

Gli utenti nei sistemi di raccomandazione di articoli scientifici hanno poche valutazioni che non consentono agli approcci di filtraggio basato sul contenuto di discriminare gli articoli che l'utente apprezza rispetto a quelli che non sono di suo interesse. A tale scopo, abbiamo proposto un modello di raccomandazione che misura la differenza tra due distribuzioni di probabilità tra il modello tematico (topic model) del corpus dei documenti del ricercatore e i modelli linguistici (language models) di nuovi documenti e ordina gli articoli minimizzando tale differenza.

Nei sistemi di raccomandazione di articoli scientifici, la matrice che contiene le valutazioni per le coppie utente-elemento (user-item) è molto sparsa e gli utenti sono relativamente pochi rispetto ai numerosi articoli disponibili. Per superare il problema della scarsità di dati negli approcci di filtraggio collaborativo (Collaborative Filtering, CF), abbiamo proposto un approccio per la raccomandazione di articoli scientifici che combina l'analisi del contenuto basato su tecniche LDA, analisi di reti sociali per la generazione di vicinati, e un modello di linguaggio (language model) basato sulla pertinenza per la raccomandazione tramite filtraggio collaborativo. Questo approccio utilizza gli argomenti nei articoli valutati del ricercatore per definire il profili utente, ignorando così i valori numerici delle valutazioni, e ap-

plicando una algoritmo per l'identificazione di comunità per raggruppare ricercatori simili in base ai loro argomenti correlati anzichè calcolando somiglianze basate su articoli co-valutati.

Abbiamo condotto studi sperimentali su DBLP, utilizzando metriche che dimostrano che i nostri approcci si comportano bene rispetto ai metodi proposti nello stato dell'arte.

Parole chiave: sistemi di raccomandazione, articoli scientifici, modellazione di argomenti

Abstract

Scientific paper recommendation is a task that aims to enhance the exploitation of Digital Libraries (DL) and helps researchers to find relevant papers from a large pool of papers. However, reliable sources to model the researcher interests must be provided to have accurate recommendations.

In this thesis, we focus on the extraction of the user topical interests from papers that the user is connected with (authored or rated) and also by using the social structure of the academic network of the user (relations among researchers in the same domain).

We proposed a fully Content-Based Filtering (CBF) approach for scientific paper recommendation that relies on topic modeling: the profile of a researcher is modeled by a set of topics obtained by applying Latent Dirichlet Allocation (LDA) to the papers written by the researcher. The profile built by this model is easily interpretable, and can explain the recommendation results. Users in recommender systems of scientific papers have few rated papers that do not allow CBF approaches to discriminate papers the user likes from others she/he does not like. For this purpose, we proposed a recommendation model that measures the difference between two probability distributions between the topic model of the researcher's corpus, and the language models of new papers and rank papers by minimizing the difference.

In recommender systems of scientific papers, the user-item rating matrix is very sparse and users are relatively few compared with the numerous available items. To overcome the issue of data sparsity in Collaborative Filtering (CF) approaches, we proposed a scientific paper recommendation approach that combines content analysis based on LDA, Social Networks (SN) techniques for neighborhood generation, and the relevance based language model to CF recommendation. This approach uses the topics in the researcher's rated papers to define the user profiles, thus ignoring the numeric values of ratings, and applying a community detection algorithm to group similar researchers according to their related topics instead of calculating similarities based on co-rated items.

We conducted experimental studies on DBLP, by using ranking-oriented metrics that demonstrate that our approaches are performing well compared to the state of-the-art methods.

Keywords: Recommender systems, scientific papers, topic modeling, content-based filtering, hybrid approaches.

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter a general overview of the thesis is presented. Motivations and open issues of this work are discussed. The addressed research questions and contributions, as well as the thesis structure and the list of related publications are presented.

1.1 Motivations and Problem Statement

The explosive growth of the world-wide web and the emergence of social web applications in Web 2.0 (blogs, wikis, content sharing sites, social networks, etc.) have caused an overwhelming amount of data [96]. Users are able to generate and share content on online platforms. A variety of items ranging from products offered in an online store to posts in social media can be rated by users [130]. For instance, in Netflix[1], a user is able to provide feedback with a simple click of mouse by using the five-star rating system that specifies her/his like or dislike of an item. In other websites like Amazon[2], editing details of a product may be viewed as an implicit positive rating for this product.

Such forms of information should be managed and processed to create personalized applications that tailor their functionality according to the user needs. Recommender systems have emerged in response to this issue. Based on each user's interest, previous ratings and behavior, they suggest items which users are likely to prefer from a huge collection of items [127].

Recommender systems have been successfully applied in several domains [16], suggesting products such as movies [153], music [141], TV programs [16] which can be bought and enjoyed.

In the scientific domain, one of the main tasks of researchers is to track what is going on in their research field [156]:

[1] www.netflix.com
[2] www.amazon.com

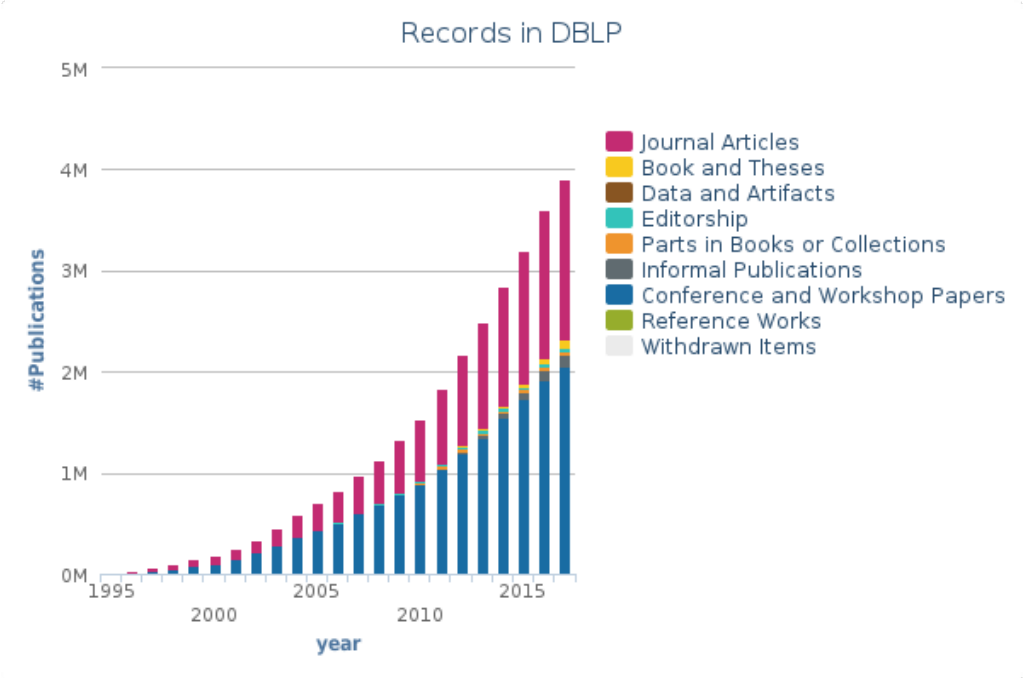


Figure 1.1: Total number of publications of the different publication types in DBLP (Figure taken from DBLP Computer Science Bibliography (September 2017))

- A PhD student needs to know what are the relevant papers matching her/his PhD subject.
- A postdoc may like to be up-to-date with which topics her/his colleagues are investigating.
- A professor might want to find funding opportunities relevant to the work done in her/his team.

In this context, the actions to discover relevant content can be difficult and time-consuming since there is a rapid development of academic publications: according the National Science Board the average annual growth of the indexes within the Web of Science is 2.5% [29].

The rate of scholarly data growth in DBLP is illustrated in figure 1.1. This has led to an increasing interest in and need for applying recommendation techniques to access relevant papers. Consequently, several scientific portals developed recommendation services such as Mendeley[3], CiteSeer[4], Citeulike[5] and Citations Google Scholar[6].

A recommender system can suggest automatically relevant papers based on the user preferences or/and other users with similar interests. Effective recom-

[3] www.mendeley.com
[4] citeseerx.ist.psu.edu
[5] www.citeulike.org
[6] scholar.google.fr/citations

mendation is largely dependent on the selection of an appropriate algorithm, and its adaptation to a given scenario or domain. It is also depending on the type of models that capture users' interests and preferences to create what is known as a "user model (profile)" [25]. For instance, when writing a paper, a researcher focuses on the topics related to her/his scientific domain. The topics addressed by a paper are also the first pieces of information a person is interested in when reading a scientific paper [70].

In this thesis we use topic modeling and language modeling to build user/item models for proposing innovative scientific paper recommendation approaches.

1.1.1 Representation of user/item models

The user/item models must be effective and efficient with respect to computational resources, and ideally they should be interpreted and easily understood by humans.

We explore the use of topic modeling in scientific paper recommendation to represent both user (researcher) and item (paper) models.

Topic models [30] have been employed as a machine learning technique to identify and annotate large text corpora with concepts, to track changes in topics over time, and to assess the similarity between documents. The purpose of topic modeling is the analysis of texts in natural language in order to discover the topics they contain and to represent them by means of probability distributions over words. They have been successfully applied to accomplish several tasks such as the analysis of scientific trends [70], query expansion in IR models [166] and scholarly publication search engines[7].

Topic models are based on the hypothesis that a person when writing a document has certain topics in mind. To write about a topic means to pick a word with a certain probability from the pool of words of that topic [70].

1.1.2 Recommendation of scientific papers

Most of Content-Based Filtering (CBF) approaches to scientific paper recommendation are mainly built upon simple retrieval models, such as keyword matching or the Vector Space Model (VSM) with basic TF-IDF weighting scheme, i.e. user and item models are represented by vectors of TF-IDF [115, 145, 86].

Some CBF approaches also used a set of concepts defined by a pre-existing ontology [44, 111] to represent both the user and the paper profiles. The recommender system matches the concepts in the user profile to each concept in the paper representation.

[7] Rexa.info

Existing Collaborative Filtering (CF) approaches to scientific paper recommendation recommend papers by predicting the rating of each paper. They are based on classic techniques of CF recommendation: model-based approaches such as the latent factor models used in [3, 161] and neighborhood-based approaches such as the extended item-based CF model presented in [61].

The neighborhood-based approaches to scientific paper recommendation apply similarity measures between users/ papers in an unary ratings space. For instance, in [61] the authors applied the item-based CF approach by building the ratings matrix from the citation network. The citations are weighted by their graph importance (PageRank) prior to item similarity computation.

The model-based CF approaches exploit the user-item rating matrix to learn a predictive model and then predict ratings of users for new papers. For instance, in [161] the authors proposed a matrix factorization model to recommend scientific papers by using topics extracted from the item descriptions and user metadata.

1.2 Open Issues

Keyword-based approaches rely on keywords match. For instance, if the user rated a paper which contains "text mining", keyword-based approaches will only recommend papers in which the words "text" and "mining" occur. Papers regarding Natural language processing techniques or Machine Learning applied to textual data will not appear in the set of recommendations [103].

On the other hand, in [90, 86] both user and new paper profiles are represented as trees of concepts from a pre-existing ontology; A limitation of these approaches is that the considered concepts are limited and too general to be able to well distinguish different topics.

Most of CF approaches to recommend scientific papers [161, 175] are designed to predict user ratings; this assumes that explicit ratings of users are available [164]. However, ratings in general are implicitly inferred; furthermore using predicted ratings as ranking scores may not accurately model the recommendation scenario as revealed in [57, 109]. The effectiveness of recommendation depends on accurately ranking of items rather than predicting ratings [164].

CF techniques use only the user-item rating matrix to provide recommendations. In recommender systems of scientific papers, the user-item rating matrix is usually very sparse. Consequently, the number of ratings is too low, as researchers rate only few papers, while CF techniques need many ratings to perform well.

1.3 Research Questions

With the above-stated open issues in mind, we have addressed the following research question:

How can we effectively apply topic modeling in the recommendation task to reflect the user interests for improved results?

Stemming from the above core research question are the following specific research questions:

1. Can topics in authored papers be utilized as reliable sources of knowledge to user and item modeling for recommendation?

   When writing a paper, a researcher focuses on the topics related to her/his scientific domain, by using a technical language. The topics in papers also are the essential information a researcher is interested in when reading a scientific paper. These core topics play an important role in the selection of new papers [70].

   In this thesis, we explore the use of topic modeling to represent the user model as a mixture of topics extracted from her/his past publications, and language modeling to represent new papers in a CBF recommendation mechanism.

2. How to identify researcher's community by using topics for scientific paper recommendation?

   In recommender systems of scientific papers, the number of ratings is very low, as researchers rate a small proportion of the available papers [28, 68], while user-based CF techniques need many ratings to perform well. Consequently, computing similarities based on co-rated items would fail to capture in an accurate way the preferences similarities between researchers. In this thesis, we explore the use of user's topics of interest to effectively identify her/his neighbors.

1.4 Contributions

The research that has been undertaken during the PhD has addressed the scientific problem of user and item modeling based on LDA for scientific paper recommendation. The main contributions of this thesis are related to the enhancement of the recommender systems of scientific papers.

In particular, this thesis brings the following contributions:

1. This thesis explores the use of topics as a source of user modeling for scientific paper recommendation. The users are modeled as a mixture of topics extracted by LDA from the researcher past publications (Chapter 4, [11]).

2. This thesis proposes a novel CBF approach for scientific papers recommendation by using the language modeling to IR model which compares the topics of interests of the researcher, as well as the technical language s/he uses to generate her/his papers. The experiment results demonstrate that it outperforms the state-of-art scientific paper recommendation approach. (Chapter 4, [11]).
3. It explores the use of topics to identify similar researchers instead of using their similar ratings for a relevance-based language modeling to CF recommendation approach [117] (Chapter 5, [10]).

1.5 Thesis Structure

The outline of this thesis is structured as follows:

- In this chapter, we introduced the research motivations, the open issues, the research questions and contributions.
- Chapter 2 presents background material to provide a description of information filtering, recommendation task along with an overview of scientific paper recommendation approaches.
- Chapter 3 presents a brief overview of topic modeling and language modeling.

Chapters 4 and 5 include our main contributions.

- Chapter 4 presents a CBF approach to scientific paper recommendation. The proposed approach is using the topics related to the researcher's scientific publications (authored papers) to formally define the author profile. In particular, we propose to employ topic modeling to formally represent the user profile, and language modeling (or topic modeling) to represent each new paper to recommend. The proposed recommendation approach is based on the language modeling to IR model to recommend new papers to the target user.
- Chapter 5 presents a recommendation approach for scientific papers which uses topics (content) in the researcher's rated papers ignoring the numeric values of ratings. We construct a collaborative researcher's graph based on the topics extracted by LDA from the researchers rated papers. We apply then a community detection algorithm to extract the neighborhood of each target researcher. The recommendation algorithm is the application of the relevance-based language modeling approach to the scientific paper recommendation.

- Finally, chapter 6 concludes the thesis, discussing the contributions provided and exploring the possibility of future work.

1.6 Publications

The research efforts presented in this dissertation are the summary of two international publications and a working journal paper:

International publications

- Amami, M., Pasi, G., Stella, F., and Faiz, R. (2016). An LDA-Based Approach to Scientific Paper Recommendation. In International Conference on Applications of Natural Language to Information Systems (pp. 200-210). Springer International Publishing.
- Amami, M., Faiz, R., Stella, F., and Pasi, G. (2017). A graph based approach to scientific paper recommendation. In Proceedings of the International Conference on Web Intelligence (pp. 777-782). ACM.

Working paper

- Amami, M., Pasi, G., Stella, F., and Faiz, R. LDA-based approaches for scientific paper recommendation.

Chapter 2

Background and Related Work

This chapter presents the background notions and related work of the scientific paper recommendation task. We start by giving a brief overview of the recommender systems: formulation, approaches and evaluation techniques. We also give an overview of the scientific paper recommendation approaches.

2.1 The Recommendation Task

The aim of Information Filtering (IF) is to expose to users only items that are relevant to them [74]. The term "collaborative filtering" was coined in 1992 by Goldberg when implementing a spam filtering system [67]. The basic idea was that "information filtering can be more effective when humans are involved in the filtering process".

Since the over abundance of product information which creates much inconvenience to users seeking products online, e-commerce sites like Amazon.com use collaborative filtering based on purchase history and customer ratings to make personalized recommendations to its customers. This category of softwares are called *recommender systems*.

Recommender systems are defined in [127] as software tools that suggest relevant items to a user, based on the user's preferences (tastes, interests, or priorities).

They have been applied successfully to different domains such as e-commerce, news, music, movies, etc [69]. For instance, on Amazon nearly 35% of what consumers purchase and 75% of what they watch on Netflix are recommended items[1].

Each domain has different characteristics that require different methods [127].

[1] www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers

Type of input data	ratings, content of items, user metadata, user context
Filtering method	content-based [103], collaborative [144], hybrid [39], social-based [148], context-aware [2]
Model chosen	memory-based [58], model-based [93], heuristic [42, 60]
Techniques	probabilistic [118, 79], machine learning [37] and fuzzy [169], matrix factorization [94]
Goal	rating prediction [153] and item-based top-n recommendation [88]
Quality of the results	diversity [176], novelty [43], serendipity [113], relevance

Table 2.1: Recommender system settings.

2.1.1 Formulation of the recommendation problem

A general formulation of the recommender problem has been defined in [1] as the following:

Let U be a set of users, and let I be a set of items. Let:

$$r^* : U \times I \rightarrow R$$

where R is a totally ordered set, be a real valued function defined on the product space of the users U and items I which predicts how a pair consisting of a user $u \in U$ and $i \in I$ is mapped to the evaluation $r^*(u, i)$ of the user u for the item i, namely the predicted "utility" or "predicted evaluation" of the item for the user [126] (Evaluation of an item by a user is called "rating" in a Collaborative Filtering (CF) recommender system). Then, items with the largest predicted evaluations are recommended to the user.

Depending on the required input data (e.g., ratings, context, logs) and the way in which the evaluation of item is predicted, three types of methods are commonly defined: Collaborative Filtering (CF) [47, 125, 164], Content-Based Filtering (CBF) [42, 60, 103] and hybrid [39, 40].

An implementer of a recommender system should consider the settings presented in table 2.1 [35].

2.1.2 Recommendation methods

Three types of recommender systems can be defined: CF recommender systems provide recommendations based on similarities between users' ratings. CBF recommender systems match the users' profiles to each items' content features to recommend new items. Hybrid recommender systems combine both of CBF and CF methods.

Other extended approaches are presented in the literature such as knowledge-based approaches and context-aware approaches.

2.1.2.1 CF methods

CF methods rely on the preferences of the user as well as those of other users in the system. The idea is that if two users are like-minded (agreed in the past on some preferences of items), relevant items to one user are recommended to the other user and vice versa [58]. They derive recommendations only from the user-item rating matrix [135]. CF methods are commonly classified into two main categories: memory-based and model-based [144].

Model-based CF Model-based CF builds offline a statistical model of user/item rating pairs based on the training set, and then it applies this model online to provide recommendations [85].

Several techniques belong to this category, such as probabilistic techniques [118], graph-based techniques [47] and the most popular ones are the latent factor models that perform a dimensionality reduction in order to uncover latent factors between users and items. Examples of latent factor models are Singular Value Decomposition (SVD) for matrix factorization [92], probabilistic Latent Semantic Analysis (pLSA), or LDA [33].

Memory-based CF Memory-based CF makes recommendations based on the entire user-item rating matrix. It computes user/item similarities based on distance or correlation measures [85].

Memory-based CF finds either like-minded users (neighbors) for the target user (user-based approach), or pairs of items that are rated by common users (item-based CF).

In user-based CF, similarities between users and the target user are computed to identify like-minded users, and preferences of neighbors are aggregated to generate recommendations.

The neighborhood selection is based on identifying most similar users to the target user according to a similarity measure.

	i_1	i_2	i_3	i_4	i_5
u_1	5	3	4	4	?
u_2	3	1	2	3	3
u_3	4	3	4	3	5
u_4	3	3	1	5	4
u_5	1	5	5	2	1

Table 2.2: user × item rating matrix.

	u_1
u_2	0.8528
u_3	0.7071
u_4	0
u_5	-0.7921

Table 2.3: Pearson correlation values between user u_1 and the other users.

The similarity between two users is typically computed by means of the Pearson's correlation coefficient between the vectors representing each user's preferences [1] which is calculated as the following:

$$sim_{Pearson}(u,v) = \frac{\sum_{i \in P}(r(u,i) - \bar{r}(u))(r(v,i) - \bar{r}(v))}{\sqrt{\sum_{i \in P}(r(u,i) - \bar{r}(u))^2}\sqrt{\sum_{i \in P}(r(v,i) - \bar{r}(v))^2}} \tag{2.1}$$

where $r(u,i)$ is the rating given by user u to the item i, $\bar{r}(u)$ is the average of all ratings given by user u and P is the set of items rated by both users u and v.

Table 2.2 shows a user-item rating matrix with ratings given by 5 users on 5 items.

We estimate in this example the rating that user u_1 will give to item i_5. Similarity values between user u_1 and other users are calculated in Table 2.3 by using the Pearson's correlation coefficient. The results show that user u_1 seems to be similar to user u_2 and user u_3, while she/he has opposite tastes with respect to user u_5.

To estimate the rating of a new item to the target user, the ratings from the neighborhood are aggregated to produce recommendations. In [125] the authors estimate the rating by considering rating deviations from the user's and neighbor's rating means.

$$\hat{r}(u,i) = \bar{r}(u) + \frac{\sum_{v \in N_u} sim_{Pearson}(u,v) * (r(v,i) - \bar{r}(m))}{\sum_{v \in N_u} sim_{Pearson}(u,v)} \tag{2.2}$$

where N_u is the set of user u's neighbors.

On the other hand, in [6] the authors predict the user ratings by computing

	i_1	i_2	i_3	i_4
i_5	0.9695	-0.4781	-0.4276	0.5817

Table 2.4: Adjusted cosine similarity values computed between i_5 and the other items.

the weighted sum of neighbors' ratings on item i. The rating is estimated as the following:

$$\hat{r}(u,i) = \frac{\sum_{v \in N_u} sim_{Pearson}(u,v) * r(v,i)}{\sum_{v \in N_u} sim_{Pearson}(u,v)} \tag{2.3}$$

Let $|N_u| = 2$ in our example, the estimated rating of item i_5 to the target user u_1 is computed according to formula 2.2 as the following:

$$\hat{r}(u_1,i_5) = 4 + \frac{0.8528 * 0.6 + 0.7071 * 1.2}{0 * 8528 + 0.7071} = 4.872 \tag{2.4}$$

Several extensions of user-based CF model in the literature have been proposed by modifying similarity measures [106], by using clustering algorithm to identify a user's neighbors [168] and by aggregating neighbor weights to generate item recommendations [92].

Item-based models consider similarities between items instead of users. It consists of building an item-item similarity matrix based on common ratings and a similarity measure. In [134] similarity between two items is defined by an adjusted cosine similarity measure that has been proved to obtain better performance than the Pearson's correlation coefficient. The adjusted cosine similarity considers the average rating of an item and it is calculated as the following:

$$sim_{cos}(i_1,i_2) = \frac{\sum_{u \in U}(r(u,i_1) - \bar{r}(u))(r(u,i_2) - \bar{r}(u))}{\sqrt{\sum_{u \in U}(r(u,i_1) - \bar{r}(u))^2}\sqrt{\sum_{u \in U}(r(u,i_2) - \bar{r}(u))^2}} \tag{2.5}$$

where U is the set of users who rated both items i_1 and i_2.

Table 2.4 shows the adjusted cosine similarity values between i_5 and the other items. i_5 seems to be very similar to i_1 and i_4. The estimated rating of a new item given the target user is computed as follows:

$$\hat{r}(u,i) = \frac{\sum_{j \in S_u} sim_{cos}(j,i) * r(u,j)}{\sum_{j \in S_u} sim_{cos}(j,i)} \tag{2.6}$$

where S_u is the set of items rated by user u. The prediction of item i_5 given the target user u_1 when the neighborhood size of an item is equal 2.

$$\hat{r}(u_1,i_5) = \frac{0.9695 * 5 + 0.5817 * 4}{0.9695 + 0.5817} = 4.625 \tag{2.7}$$

Ranking-based CF Both memory and model-based are rating-based CF models [91, 1, 78]. However, the goal of a recommender system is to rank a list of items relevant to the user rather than predicting ratings [165, 102].

In this context, several ranking-based CF models have been proposed: vector-space IR model [26, 165], binary independence retrieval model [164], statistical language models [163] and learning to rank models [167, 138, 147].

In [164] the authors found analogies between CF with implicit data and IR. They introduced the concept of binary relevance into CF by applying the probability ranking principle in IR [128] to CF.

In [26] the authors proposed to use IR models in the item-based CF framework. They applied an IR model by identifying an analogy between IR and item-based CF: in item-based CF, terms are seen as the items, while the term frequencies are the user ratings (in the query representation) or the item similarity (in the document representation). The experiments show better results than the classic item-based CF approach.

In [165] the authors introduced a recommendation framework for adapting the VSM to ranking-based CF. Users are considered as documents and pairwise relative preferences of items as terms. The terms are weighted by a degree-specialty weighting scheme similar to TF-IDF. A user-item based CF is applied by aggregating the partial rankings of the user's neighbors into a total ranking of items for recommendation.

In [50] the authors reformulated the recommendation problem and used algorithms from IR, namely a model based on Discrete Fourier Transform (DFT) and the VSM by modeling a user as a document, an item as a term, a target user as the query, and ratings as weighting scores of the original IR algorithm.

In [164] the authors presented a language modeling approach for the item ranking problem in CF. The new item to recommend is generated by using a linear smoothing technique defined by a combination of the item popularity and its co-occurrence with the items rated by the target user.

Advantages CF methods derive recommendations from the user-item rating matrix, thus avoiding the need of collecting extensive information about items and users.

Model-based CF can provide serendipitous recommendation with latent factor models i.e. recommending unexpected and interesting items. These models are able to characterize the preferences of a user with latent factors. For instance, in a book recommender system, a latent factor model can determine that a given user is a fan of books that are both fiction and romantic, without having to define the aspects "fiction" and "romantic".

Memory-based CF is simple to implement and do not require costly training

phases [58]. Furthermore, it can provide explanations which reveal the reasoning behind recommendations. For instance, the item-based recommender system can provide the list of similar items and ratings given by the user to these items [150].

Limitations CF methods suffer from a general issue called *cold start problem*, which may occurs in two cases: First, when new items (or old items which have very few ratings) are unlikely to be recommended (long tail). Second, when users made no or few ratings, the system is unable to find like-minded users and hence to provide recommendations.

The cold start problem is more prominent when the user-item matrix is very sparse: in many cases the number of common rated items between users is low, as users rate only a few items, while CF methods need many ratings to perform well.

In general, CF methods are less scalable and require an offline learning of the recommendation model [85]. In [125] the authors report that CF methods can be easily manipulated through fake users who promote their own items.

2.1.2.2 CBF methods

CBF methods analyze a set of features of items relevant to the user and learn a user profile based on these features. The filtering process basically consists in matching up the features of the user profile against the features of an item content (i.e., item profile) [103].

Differences between CBF and IR In [23] the authors made a comparison between CBF and IR by outlining their similarities and differences. IR has a common goal with CBF, which is to select items relevant to users. The differences between CBF and Information Retrieval (IR) systems are presented in table 2.5.

Formulation of CBF We define *content(i)* as the item profile which is a set of attributes characterizing the item i. It is usually computed by extracting features from item i and used to determine the appropriateness of the item for recommendation purposes. Then, we refer $content-based-profile(u)$ to be the profile of the target user u containing her/his preferences. Hence, the evaluation of an item (utility function) $g(u,i)$ is defined as:

$$g(u,i) = matching(content-based-profile(u), content(i)) \tag{2.8}$$

CBF system components A CBF system includes four basic components (see figure 2.2):

Characteristics	IR	CBF
Frequency of use	ad-hoc use of a one-time information need	long term users with long term information needs
Representation of user needs	queries	user profiles
Goal	selecting relevant documents from databases that match the query	omitting irrelevant from incoming streams of items or seeking relevant items for target users
Database	static	dynamic
Type of users	not known to the system	known to the system, a user model is saved in the system
Scope of system	only relevance of items	social issues like privacy and user profiling

Table 2.5: Differences between CBF and IR systems [74].

- The data analyzer component: The items are collected and represented in item models. They are the input of the similarity model component.
- The user model component: It constructs the user profile from explicit or implicit information that represents the user preferences. The user profile is also an input to the similarity component model.
- The similarity model component: It consists of matching the user profile with the represented items and calculating the similarity between them.
- The learning component: It detects the shifts of the user's interest in time from the user's feedback. Then, it updates the user model to fit the new preferences in order to improve the filtering process.

The user profile The user profile is a fundamental component in a CBF system. It is built by understanding and specifying the following aspects [120]:

- User characteristics: definition of a set of user properties such as age, gender, occupation, nationalities, spoken languages, etc.
- User's goals: identification of the user's goal. For instance, the Youtube recommender system presents different recommendation goals. It can suggest playists based on popular uploaded videos, on the musical style of the user, on an artist that the user frequently watched her/his videos, etc (see figure 2.1).

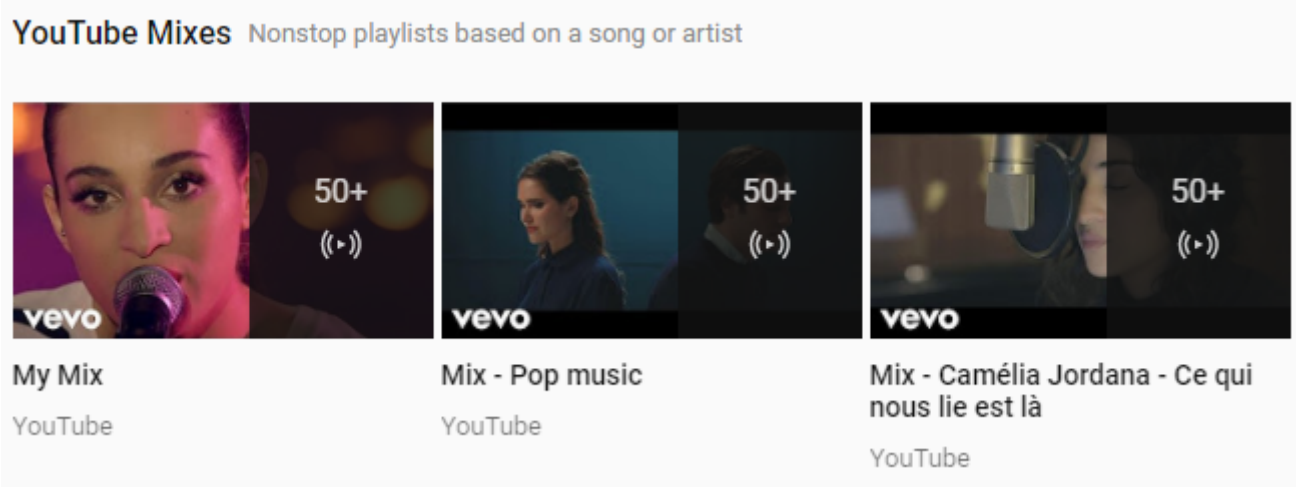


Figure 2.1: Personalized playists offered to a user on Youtube site

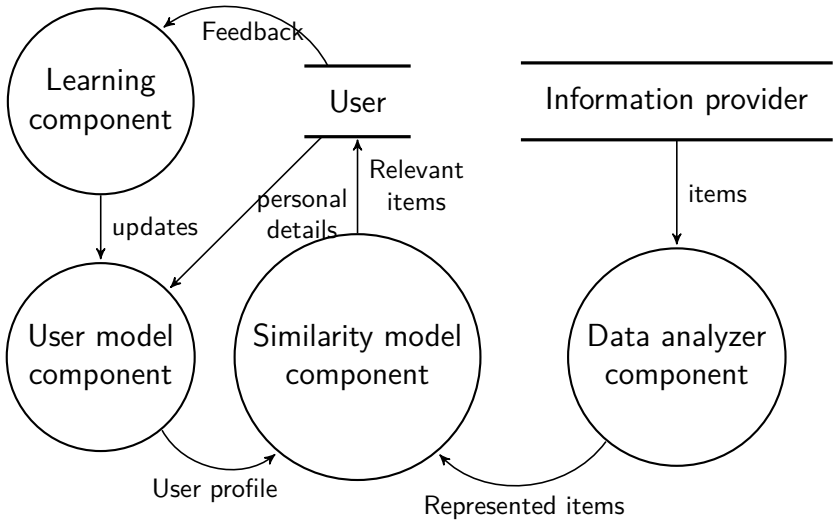


Figure 2.2: CBF system components

- User's context: definition of the user's device (e.g., mobile handset, PC desktop) and the situation of interaction (e.g., location, temporal information).

CBF methods CBF methods can be classified into two categories: *1)* models built on machine learning, such as naive Bayes model [103], decision trees and neural networks [119], or *2)* heuristic functions inspired from IR techniques [42, 60].

The ML methods try to classify new items in a system as relevant or irrelevant for each user. For instance, in [103], the authors used naive Bayes model to generate a probabilistic model that recommend items. The naive Bayes model estimates the probability of an item i is relevant or irrelevant (class c) $p(c|i)$,

based on the a priori probability for the class $p(c)$, the probability of observing the item $p(i)$, and the probability of observing the item given the class $p(i|c)$ (i.e., probabilities of items already rated by the user) as the following:

$$p(c|i) = \frac{p(c)p(i|c)}{p(i)} \tag{2.9}$$

Most of CBF methods are based on heuristic models which represent users and items as vectors of TF-IDF [142] (or BM25 [15]) in a VSM.

VSM is a spatial representation of text documents $D = \{d_1, d_2, \ldots, d_N\}$ where each document is represented by a vector in a n-dimensional space and each dimension corresponds to a term from the overall vocabulary of a given document collection $T = \{t_1, t_2, \ldots, t_n\}$. The document vector of d_j is a vector of term weights, where each weight indicates the degree of association between the document and the term, $d_j = \{w_{1j}, w_{2j}, \ldots, w_{nj}\}$, where w_{kj} is the weight for term t_k in document d_j [103].

The most popular used term weighting scheme is TF-IDF that considers terms which occur frequently in one document (TF =term-frequency), but rarely in the rest of the corpus (IDF = inverse-document-frequency).

The TF-IDF weight of a term is calculated as follows:

$$TF - IDF(t_k, d_j) = TF(t_k, d_j) \log \frac{N}{n_k} \tag{2.10}$$

where N denotes the number of documents in the corpus, n_k denotes the number of documents in the collection in which the term t_k occurs at least once, and $TF(t_k, d_j)$ which is calculated as follows:

$$TF(t_k, d_j) = \frac{freq(t_k, d_j)}{maxFreq(d_j)} \tag{2.11}$$

where $freq(t_k, d_j)$ is the frequency of the term t_k in document d_j and $maxFreq(d_j)$ is the maximum TF value computed in document d_j.

Since items and user profiles are represented as vectors of TF-IDF, items are recommended in decreasing order of similarity with the user (the user profiles are represented in the same form of the items). The most common measure is the cosine similarity which is calculates as the following [42]:

$$sim_{cos}(u, i) = \frac{\sum_k w_{ki} w_{ku}}{\sqrt{\sum_k w_{ki}^2} \sqrt{\sum_k w_{ku}^2}} \tag{2.12}$$

where w_{ki} is the weight assigned to the feature k in item i.

Advantages In contrast to CF, CBF allows user independence. Instead of using ratings from other users in order to find the "nearest neighbors", CBF exploits solely ratings provided by the target user to build her/his own profile.

CBF can provide content features or descriptions that caused an item to occur in the list of recommendations to provide explanations to the user [54].

Furthermore, CBF does not suffer from the long tail problem as they can recommend items not yet rated by any user [103].

Limitations CBF methods cannot provide recommendations if the content does not contain suitable features to discriminate items (Portfolio effect) [103]. Hence, a domain knowledge is needed to associate new types of features with the new items. For instance, for movie recommendations the system needs an external source like an ontology to know the actors, directors of the movie, etc.

Furthermore, CBF methods tend to recommend items that contain exactly the same information as the items the user already has rated. This issue is called the 'serendipity problem.

Also, CBF methods are not able to provide effective recommendations before collecting enough ratings to build the user profile (cold-start problem for users).

2.1.2.3 Combining recommender systems (Hybrid methods)

A hybrid recommender system combines CF and CBF techniques to improve their individual performance by for example alleviating the cold start problem and providing more diverse recommendations.

In [39] the authors presented three base designs:

- A single recommender system that incorporates diverse range of input data of several recommendation techniques in one algorithm implementation. For instance, in [18] the authors proposed to combine collaborative features such as user's ratings and content features of items.
- Parallelized hybrid recommender systems that operate independently and produce separate recommendations. Then, the output is combined into a final list of recommendations.
- Recommender systems are joined together in a pipeline architecture in which the output of a recommender is the input of a subsequent recommender.

In [17] the authors used both the last two strategies by running in parallel three recommendation techniques (a user-based CF, an item-based CF and a CBF algorithm) to generate rating predictions, which are then combined by a meta-learning algorithm that uses the rating predictions as meta-features.

2.1.2.4 Knowledge-based recommendation approaches (KRS)

For specific item domains, such as automobiles, tourism services, or expensive luxury goods, CBF and CF techniques are unable to provide recommendations. These items have a particular set of properties that make recommendation difficult based on few data [5].

KRS exploit deep knowledge about the item domain [154, 63, 38]. These systems are interactive, which allows the user to explore the complex feature space of the item and learn about the trade-offs available between various recommendation [5, 154].

2.1.2.5 Context-aware recommendation approaches (CARS)

CARS take into account contextual information such as time, place, etc. For instance, in music recommendation, the listener's mood and location may matter to provide better recommendations [2].

In [77] the authors asserted that contextual information about the user's task into the recommendation algorithm in certain applications can provide better recommendations.

CARS can be classified into two categories:

- The recommendation via context-driven querying and search which consists of using the current contextual information and user's ratings as queries to recommend the relevant items.
- The recommendation via contextual preference elicitation and estimation which learns the user model by incorporating the contextual information [48].

2.1.3 Recommendation with side information

The recommender systems collect the relations between users and items. The most popular form of relation is the rating which represents the relevance judgment (user's level of interest) given by the user to the item.

Ratings are collected explicitly or inferred by the system implicitly. They are represented in a variety of forms [135]:

- numerical: a user selects a number on a rating scale to reflect the relevance of an item such as the 1-5 stars provided in the book recommender system of Goodreads[2].
- ordinal: a user selects an ordinal rating that reflects her/his opinion

[2] goodreads.com

- binary: rating of 0 or 1 which can be explicit such as hitting thumbs-up/down buttons or implicit such as buying a product.
- unary: ratings are only positives which can be also explicit such as a "like" statement or implicit such as downloading a song.

Beyond ratings, there exists rich information such as user-generated text and social networks in recommender systems that can improve the recommendation results.

2.1.3.1 User-generated text

The content information generated by users such as tags, comments, reviews and in our case scientific text usually point out the reasons of ratings. For instance, travelers share their experiences on Tripadvisor[3] to support the travel provider in case of positive experiences and to aware people from bad services. Thus, reviews complements ratings and can be used to suggest ratings when these latter are missing. In [130] the authors extract sentiments and polarized latent features from textual reviews to identify useful predictors for the overall rating.

2.1.3.2 Social networks

With the rise of social networking, recommender systems acquire information about the *social influence* between users. In [75] the authors asserted that friends have a tendency to select the same items and give similar ratings. They proposed a recommendation model which make use of an online social network besides user's ratings and they did show that the use of these information in a recommender system not only improves the performance of recommendation but also remedies the data sparsity and cold start problems.

Indeed, in recommender systems of scientific papers, ratings are unary and implicitly inferred which means that there is no explicit evaluation of papers by users. Furthermore, the number of ratings is very low which results in the data sparsity and cold start problems.

Hence, the side information is considered in this thesis to build recommendation models for scientific papers which are typically the user generated content (topics in papers).

2.1.4 Evaluation of recommender systems

The evaluation process quantifies the quality of a recommender system. Typically, this quantification is performed via calculation of metrics which are directly associated with the relevance of items provided to the user.

[3] www.tripadvisor.com

2.1.4.1 Offline evaluation

Two main evaluation methods are presented in the literature to evaluate recommender systems: online and offline [72]. In this thesis, we use the offline evaluation.

The offline evaluation of a recommender system is based on methods used in ML evaluation. It consists of holding out from the system a part of ratings (testing data), leaving the rest (training data) as input to the algorithm, and requiring the system to predict such data.

If a specific number of ratings is required for training purposes, this method is called given-n: for every user the ratings are split into n training ratings and the remaining as testing ratings. Analogous to the given-n, the all-but-n in which the testing set can have a specific size: n testing ratings and the remaining as training ratings.

2.1.4.2 Metrics

In the classic formulation of recommendation, which is to predict ratings of items, the effectiveness of the recommender system has been evaluated by measuring the error between predicted and true ratings, using metrics such as the Mean Absolute Error (MAE), and the Root Mean Squared Error (RMSE).

However, since the recommendation task is considered in recent work as a relevance ranking problem, several authors [52, 24, 84] use evaluation metrics from IR such as precision, recall and coverage. We define in the following the evaluation metrics used throughout this thesis.

Recall@m The recall quantifies the fraction of relevant papers that are in the top-m of the ranking recommendation list sorted by their estimated relevance score (relevant recommended papers up to position m) from among all relevant papers in the test set.

For each target user u:

$$Recall@m = \frac{|N(m;u)|}{|N(u)|} \tag{2.13}$$

where $|.|$ denotes the cardinality of a set, $N(u)$ is the set of papers relevant to u (positively rated) in the test set and $N(m;u)$ is the subset of $N(u)$ contained in the top-m list of all papers sorted by their estimated relevance score. The recall for the entire system can be summarized by using the average recall from all researchers.

Coverage Two types of coverage can be defined: the user coverage which is the proportion of users for which the system can recommend at least one item; and the item coverage (catalog coverage) which is the proportion of items the system can recommend to users in the system [137]. We use the item coverage defined as the following [64]:

$$COV(L) = \frac{\bigcup_{j=1}^{N} I_L^j}{|I|} \tag{2.14}$$

where I is the number of items in the test set and $\bigcup_{j=1}^{N} I_L^j$ is the set of distinct items contained in all top-L recommendation lists.

A recommender system with a low coverage can recommend only a small number of distinct items which results in little diverse recommendations. On the contrary, systems with high coverage are more likely to provide diverse recommendations [104, 105].

2.2 The Scientific Paper Recommendation Task

In the last years, a big deal of research has addressed the issue of scientific papers recommendation. This problem has become more and more compelling due to the information overload phenomenon suffered by several categories of users, including the scientific community. Indeed, the increasing number of scientific papers published every day implies that a researcher spends a lot of time to find publications relevant to her/his research interests. In particular, recommender systems serve in this context the purpose of providing the researchers with a direct recommendation of contents that are likely to fit their needs. Several types of items can be recommended in scientific domain such as research papers, books [112], academic events [89], venues [171], citations [27, 76], academic datasets [140], new collaborators [177], research topics [151], etc.

2.2.1 Domain characteristics

Recommender systems are used in different application domains such as news, movies music, scientific papers, etc; each of which has different characteristics which require specialized methods for adapting the recommendation process [5].

In [41] the authors identified six important characteristics of the domain that an implementer of a recommender system should consider:

- Heterogeneity: It indicates that items have many features. For instance, in e-commerce applications the products to recommend are in different categories. In a homogeneous item space, the content knowledge is specific and

easier to acquire and to maintain.

- Risk: The risk in a recommender system consists of the degree of user's tolerance to false positives among recommendations and this is likely to occur in two cases: the user has very strong constraints on a relevant recommendation or the cost of accepting an item is very high.
- Churn: In a high churn domain, new items are introduced to the system continuously and they have a very short time span of relevance. Only few users can rate these items in that period and that incurs in a very sparse user-item rating matrix.
- Interaction style: In certain domains such as e-commerce, users (clients) are interacting with the system many times, often consuming similar items, and therefore explicit ratings are quite easy to obtain. In contrast, domains exist in which users make no special efforts to interact with the recommender system. The recommender systems extracts implicit ratings from user behavior.
- Preference stability: User preferences change over time due to evolving needs. Stable preference means that the user is interested in an item for a long period and she/he wishes to continue getting similar recommendations to it. For instance, a user purchasing tickets to Disneyland Paris would typically switch preferences of attractions in Paris once the purchase is complete, while a person likes Italian food may wish to continue getting recommendations of Italian restaurants for a long time.
- Scrutability: In some domains like medical applications users need explanations about why an item has been suggested and which features make that specific item more desirable than others.

Domain	Risk	Churn	Heterogeneous	Preferences	Interaction style	Scrutability	Techniques
News	Low	High	Low	Stable	Implicit	Not required	CBF / CF
E-commerce	Low	High	High	Stable	Explicit	Not required	CF
Movies	Low	Low	Low	Stable	Explicit	Not required	CF
Music	Low	Low	Low	Unstable	Implicit	Not required	CBF / CARS
Life insurance	High	Low	Low	Stable	Explicit	Required	KRS
Tourism	High	Low	Low	Unstable	Explicit	Required	CBF / CARS
Job search recruiting	High	Low	Low	Stable	Explicit	Required	CBF
Scientific paper	Low	Low	Low	Stable	**Implicit**	Not required	CBF / Hybrid

Table 2.6: Sample domains of recommendation [41].

Data type	unstructured
Quality of metadata	High quality: keywords-based annotation, few metadata, medium expressiveness through key-words annotation of content
Description based on standards (e.g ontologies) [111]	ACM Computing Classification System (CCS)[4]
Volume / diversity of items	huge number of papers on a large variety of topics
Distribution of items	long-tail
Stability and persistence of items	persistent, continuous stream
User ratings	implicit ratings: downloading a paper, citing it, adding it to one's library, editing paper details, consulting its bibliography

Table 2.7: Data model instantiation of a recommender system of scientific papers.

Table 2.6 shows the characteristics of eight different domains and the predominant recommendation approaches in each one [41]. CF techniques are applied in heterogeneous domains. High risk leads to use KRS techniques. In tourism services recommendation, unstable preferences can be handled by CARS [155].

In recommender systems of scientific papers, items are *homogeneous* i.e. even with different types of papers such as journal papers, conference papers, technical papers all have the same type of data. Furthermore, ratings are only associated with true positive items, and in general they are *implicitly* captured by means of the analysis of various kinds of interactions between researchers and papers, like downloading a paper, citing it, adding it to one's library, editing paper details, consulting its bibliography, etc.

Due to the high number of papers, new papers and the long tail (infrequently read papers) may contain the so-called *sleeping beauties*; papers containing extremely relevant topics, but remain unknown to most researchers for a very long time.

Table 2.7 shows the data model instantiation in a recommender system of scientific papers.

The user model represents the topics that the researcher is pursuing in the field (topical relevance). These topics may remain the same for a long period (for instance a PhD topic). Table 2.8 shows the user model instantiation in a recommender system of scientific papers.

Demographic information	Job as the only important demographic information: PhD student, postdoc, professor, etc.
Goal's existence and nature	pull mode: recommending scientific papers related to her/his topics of interest which are inferred from the implicit ratings
Level of expectation	medium: Some users are looking for papers that are exactly similar to their topics of interest; others would want to discover new papers that are different from their current topics
Change of expectation over time	yes: topics of interest of the researcher change over time
Importance of user situation	low: the user behavior cannot be affected by the context (location, temporal information)
Social environment	single user: looking for new scientific papers could be an individual activity or group of users in a research lab: collaborative work on a co-authored paper
Trust and privacy concerns	Not considered

Table 2.8: User model instantiation of a recommender system of scientific papers.

2.2.2 Scientific paper recommendation approaches

Content-Based Filtering (CBF) is the predominant technique for scientific paper recommendation because of the rich content of scientific papers and the few number of ratings (the user is not very active in a recommender system of scientific papers) [9]. Indeed, in [20] the authors reviewed 58 scientific paper recommendation approaches, 31 used CBF techniques (53%). Only 7 approaches applied Collaborative Filtering (CF). Furthermore, none of the reviewed CF approaches used explicit ratings.

When defining a CF approach to scientific paper recommendation, the *cold start problem* is particularly serious as the user-item rating matrix is very sparse. In applications like movie recommender systems, there are usually few items and several users. For instance, the MovieLens 1M4 dataset[5] contains 1.000.209 ratings from 6040 users and 3706 movies. Moreover, the users are clients who are very likely to interact with the system several times, often consuming similar items; therefore, ratings are quite easy to obtain. Hence, recommendation models can make accurate recommendations for most users in e-commerce domains. On the contrary, in recommender systems of scientific papers, users are relatively few compared with the numerous available items; the user-item rating matrix is usually very sparse (e.g., the sparsity of the implicit user-item rating matrix of the bibliographic portal Mendeley[6] is by three orders of magnitudes smaller than the Netflix[7] user-item rating matrix [158]).

As reviewed in [20], the ratings are in general implicitly inferred in most of scientific paper recommendation approaches. The key issue in inferring ratings is that there is no explicit quality evaluation any more by the target user. For instance, in [170] the authors made the assumption that the more pages a user read, the more she/he was considered to like the read paper. However, the user can spend a lot of time in reading the paper because she/he has some trouble to understand it.

2.2.2.1 CF approaches to scientific paper recommendation

The aim of scientific paper recommendation systems based on CF techniques is to exploit the users (researchers) /items (scientific papers) patterns identified within a research community to provide recommendations.

Many recommender systems of scientific papers were implemented based on extended item-based CF [152, 61].

In [61] the authors apply link analysis algorithms such as HITS and PageRank

[5] grouplens.org/datasets/movielens/
[6] www.mendeley.com
[7] www.netflix.com

to the citation network of the entire corpus (rated and unseen papers) to measure the importance of a paper in it. Then, they integrate the node (paper) weights to an item-based CF recommendation. This method is not able to provide recommendations for new papers or for papers with few ratings (cold-start problem for items).

In recommender systems of scientific papers, there are usually few users and several items. Hence, the item-based CF approaches do not scales well in contrast to the user-based nearest neighbors CF approaches which perform well in this domain [4].

In [80] the authors introduced the *Mendeley Suggest* recommender system which is a user-based nearest neighbors CF approach. They identified the 100 users most similar to each user by using the cosine similarity between users libraries. The relevance score of a candidate paper for a target user is the sum across the inverted neighborhood.

In [4] the authors assume that the accuracy of the nearest neighbor method to identify groups of users on sparse data is very low. Thus, they apply a subspace clustering algorithm to search similar users who share many common items in access logs.

More recently, memory-based approaches have been proposed to recommend items to users based on their ranking of shared items. In [170] the authors extracted the users access logs to model their preferences. Then, they computed similarities between any two users based on their rankings of the books they both read. Although this method alleviates the cold-start problem by using external resources (not related to items), these latter are generally noisy and not reliable, as pointed out in [136].

2.2.2.2 CBF approaches to scientific paper recommendation

Several approaches in scientific paper recommendation have adopted CBF [86, 44] as this method does not require particular assumptions over the size and the activity of the user (sparsity problem), nor penalize items that have not been rated or yet by many users if satisfactory meta-data are available (cold start problem of items).

In [145] the authors asserted that CBF techniques are more appropriate, as scientific papers are textual data and provide a reliable to base the user profile construction on. Indeed, several authors show that CBF methods in scientific paper recommendation outperform CF methods [19, 86].

CBF approaches for scientific paper recommendation extract features for user modeling from different sources.

In [44] the authors built user profile based on the user previously published

(co-authored) papers. In [90] the authors used the user previously viewed papers. The user profiles can be also modeled from external sources like social tags [87], mind maps [22, 21], etc.

Papers are pre-processed to extract content from titles [98], abstracts [61], keywords [83], introduction [82], paper's body text [115], citations [62, 110], citation context [76], discipline [80], etc.

In [80] the authors proposed to recommend items which look like the content of the most recently read paper. The recommended papers are then re-ranked by multiplying the similarity score by the log of the popularity of the paper.

In [44] both the user and papers are modeled as trees of concepts from the ACM's Computing Classification System (CCS); the recommender system matches the concepts in the user profile to each concept in the paper representation by means of a tree matching algorithm.

On the other hand, in [55] the authors modeled the user profile by a network structure called a context graph. For each document viewed by the user, a context graph is built by processing a weighted list of key-phrases.

In [145] and [146] the user profile is constructed by using the relations between each researcher's paper and its citation and reference papers. They also compute the feature vectors of the candidate papers to recommend by using relations between the candidate paper and its citations and references. The candidate papers are represented by TF weights of its terms and the terms of its context (citations and references). In this approach, the authors assumed the availability of the full text of the papers which is rarely the case.

In [115] the authors considered that existing recommender systems depend on the item provider system which is in general using only one source of items. For this purpose, they used only one single paper (title and abstract) as an input to represent the user preferences. Then, they generated queries by using terms in that paper to search for candidate papers of recommendation in search forms made available by Web information sources. Finally, they applied a TF-IDF approach to recommend the papers most related to the input paper. A limitation of this approach is that the considered input paper is not enough to identify the user preferences.

In [8] the authors built the researcher profile by using the multivariate linear regression problem to learn the relations between a given researcher and keywords from her/his previous publications (the importance of the keyword to the researcher). The used features are the keywords of her/his publications and their recency scores.

2.2.3 Scientific paper recommendation approaches based on topic modeling

In [161] the authors introduced the Collaborative Topic Regression (CTR) recommendation model, which combines matrix factorization and LDA into a single generative process, where item latent factors are obtained by adding an offset latent variable to the topic proportion vector.

In [175] the authors proposed a hybrid recommender system which is an extension of the CTR model called CAT (content + attributes recommender system). It incorporates content and descriptive attributes of items (e.g., author, venue, publication year) into an uniform model to improve the recommendation accuracy.

These above models suffer from several limitations. First, they are unable to make accurate predictions for researchers with only few ratings. The latent feature vectors for users with only few ratings are close to the prior mean and, consequently, the predicted ratings for the users are influenced by other users. Second, the latent factor models do not consider the relational structure among scientific papers. This relational structure information in academic networks provides useful directions for researchers to find interesting papers. Furthermore, they may not effectively support tasks that are specific to a certain research task such as recommending citations. For instance, a biological scientist who is interested in data mining applications to biological science might desire the recommender systems to support tasks such as recommending new papers on data mining techniques. While both CAT and CTR are using a rich source of metadata to generate recommendations, they are subject to certain limitations that affect their effectiveness in modeling citation patterns. The citation context, defined as a sequence of words that appear around a particular citation [146] is not highlighted in the learned models.

In [131] the authors proposed to use an alternative of LDA for assigning the user's interest and recommending papers based on the usage data (items saved in user's libraries). When applying LDA, they consider the documents as terms and users as a set of documents. The obtained probability distributions to the usage data are: $p(d|k)$ which is the probability distribution of the document d given the topic k and $p(k|u)$ which is the probability distribution of the topic k given a user u. Then, they assigned one topic k_u from the extracted topics to the target user by using two methods: the predominant topic in the user's library or the most recent one. Finally, the papers to recommend are sorted by decreasing values of $p(d|k_u)$.

2.3 Summary

There has been much done in recommendation research area over the past years that have used a broad range of statistics such as ML, IR and other techniques which advanced the state-of-the art of recommendation. The recommendation approaches can be classified based on the required input data and the used filtering techniques: CBF, CF and hybrid. The recommender systems have been applied in several domains by using different techniques depending upon the domain factors and the data model. We presented the scientific paper recommendation approaches.

Chapter 3

Topic Modeling and Language Modeling Overview

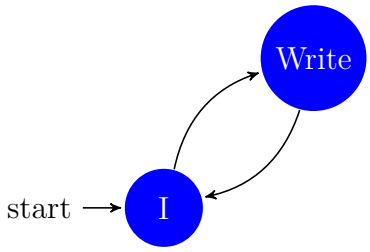
When the type of data is textual, to define a user model it is a necessary to process user related textual data from which to extract keywords, or topics. To this aim several techniques have been used such as probabilistic models, ontologies, etc. In this chapter, we focus on topic modeling and language modeling, which have been successfully in a multitude of web applications such as Information Filtering (IF), IR and text classification.

3.1 Language Modeling

Language models are used to represent textual data in many applications such as speech recognition, machine translation, and handwriting recognition. A language model is a probability distribution over words in an indexed vocabulary [53]. They can be used to "generate" new word sequences (documents) by sampling words according to the probability distribution: considering the language model as a pool of words, where the probabilities determine how many occurrences of a word are in the pool, then we can generate word sequences by reaching in (without looking), drawing out a word, writing it down, putting the word back in the pool, and drawing again [53]. This generative process is approximating the model of the topic that the author of the document had in mind when she/he was writing it.

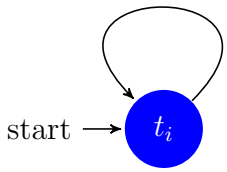
A traditional generative model of a language from formal language theory, can be used either to recognize patterns of strings or to generate strings (document). The generative model of a language is illustrated as a finite automaton that generates strings/ document as illustrated in figure 3.1.

The "unigram" language model associates to each string in the vocabulary a probability of occurrence. For instance, if the documents in a collection contain



I Write
I Write I Write
I Write I Write I Write

Figure 3.1: A finite automaton and strings it generates.



t_1 0.2
t_2 0.1
t_3 0.01
t_4 0.02
... ...

$p(stop|t_i) = 0.2$
$p(t_1 t_2 t_3 t_4) = (0.2 * 0.1 * 0.01 * 0.02) * (0.8 * 0.8 * 0.8 * 0.2)$

Figure 3.2: A "unigram" language model that is illustrated by a one-state finite automaton [46].

just five different words, a possible language model for that collection might be (0.2, 0.1, 0.35, 0.25, 0.1). In this case, the language model is a probability distribution over vocabulary of the collection, that is:

$$\sum_{w \in V} p(w) = 1 \tag{3.1}$$

where V denotes the vocabulary.

The generative model of a "unigram" language model is illustrated by a probabilistic finite automaton with a single node and a probability distributions (see figure 3.2). After a word in the vocabulary is generated, the model can be looped to produce another word or stopped. It means that the model also includes a probability of stopping generating words (final state). To calculate the probability of a sequence of words, the probabilities of words are multiplied with the probabilities of continuing or stopping after generating each word.

3.2 Topic Modeling

Topic modeling has been employed as a technique to identify and annotate large text corpora with concepts, to track changes in topics over time, and to assess the similarity between documents. The purpose of this algorithm is the analysis

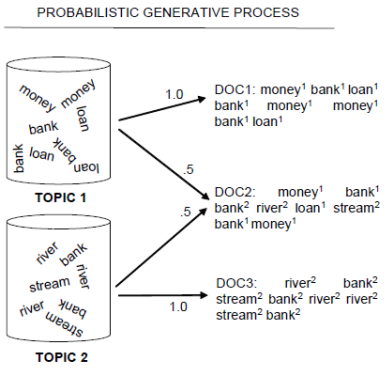


Figure 3.3: The generative process underlying topic models [143].

of texts in natural language in order to discover the topics they contain and to represent them by means of probability distributions over words. In the real world tasks, topic modeling has been successfully applied to address several tasks (e.g., analysis of scientific trends [99], IR [172, 166, 81], and scholarly publication search engines [1]).

Topic models are a range of generative models for language that specify procedures by which documents are built [30]. The basic algorithm of topic modeling is Latent Dirichlet Allocation (LDA) which describes a generative model for topics and documents [33]. The generative process of LDA is defined as: first, choose a set of topics; then for each word in the document choose a topic from that set and select a word from the topic (see figure 3.3). Thus, LDA exhibits multiple topics in a document and that each word in document can be associated with one of these topics.

3.2.1 LDA algorithm

We denote by D the number of documents, K as the number of topics and N_d the number of word in document d. We define the following variables:

- $\phi_{1:K}$: topic distributions over the vocabulary, where ϕ_k is the distribution for topic k.
- $\theta_{1:D}$: document distributions over topics, where θ_d is the distribution for document d.
- $z_{1:D,1:N_d}$: topic assignments for each document, where $z_{d,n}$ is the topic assignment for a word in position n of a document d.
- $w_{1:D,1:N_d}$: word occurrences for each document, where w_{dn} is the word that occurs in position n of document d.

[1] Rexa.info

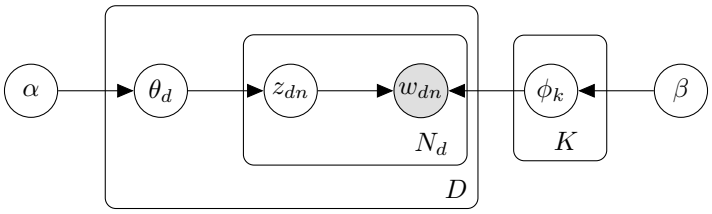


Figure 3.4: Plate notation of the LDA model.

The generative process of a generic document d consists of the following steps:

- a topic distribution θ_d is randomly generated.
- for each word position in d
  - Randomly choose a topic k from θ_d.
  - Randomly choose a word w from ϕ_k.

The generative process of LDA corresponds to the following joint distribution of the hidden and the observed variables [30]:

$$p(\phi_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{k=1}^{K} p(\phi_k) \prod_{d=1}^{D} p(\theta_d) \prod_{n=1}^{N_d} p(z_{d,n}|\theta_d) p(w_{d,n}|\phi_{1:K}, z_{d,n}) \quad (3.2)$$

The joint distribution is used to compute the conditional probability of hidden variables given the observed variables, called the posterior probability distribution.

$$p(\phi_{1:K}, \theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{p(\phi_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})} \quad (3.3)$$

The exact estimation can be computed by summing the joint distribution over every possible instantiation of the hidden structure (i.e., assigning each observed word to every possible topic), which is computationally unfeasible.

Topic model uses two different algorithms to approximate the formula 3.3 by adapting an alternative distribution over the latent topic structure to be close to the true posterior: sampling algorithms and variational algorithms.

Sampling algorithms are attempting to collect samples from the posterior to approximate it with an empirical distribution. Gibbs sampling is the most common sampling algorithm [65]. It consists of a definition of a Markov chain on the hidden topic variables for a corpus. The process is iterated multiple times to collect samples from the posterior and then approximate the distribution with the collected samples. In [70] the authors estimate the topic distributions over the vocabulary ϕ and document distributions over topics θ as follows:

$$\widehat{\phi}_{kw} = \frac{C^{\phi}_{kw} + \beta}{\sum_w C^{\phi}_{kw} + W\beta} \quad (3.4)$$

$$\widehat{\theta}_{dk} = \frac{C^{\theta}_{dk} + \alpha}{\sum_k C^{\theta}_{dk} + K\alpha} \tag{3.5}$$

where C^{ϕ}_{kw} maintains a count of all topic word assignments C^{θ}_{dk} counts the document topic assignments, and α and β are the hyper-parameters for the Dirichlet priors, serving as smoothing parameters for the counts.

Variational methods are a deterministic alternative to sampling algorithms that approximate the probability of the posterior through optimization [33]. They posit a parameterized family of distributions over the hidden variables and then find the member of that family that is closest to the posterior.

3.2.2 LDA extensions

The generative model LDA has been extended to discover more complex structure in big text corpora. To model topic changes over times and the evolution of topics, dynamic topic models have been introduced [32].

In [159] the authors have developed an extension of LDA to remove the bag of words assumption by assuming that the topics generate words conditional on the previous word.

Supervised model of LDA have been also proposed to predict labels to documents besides modeling the textual data [108, 123].

In [149] the authors proposed an extension of LDA where the number of topics is determined automatically (not fixed) during posterior inference.

In [31] the topic structure is assumed to be hierarchical, and the number of sub-trees and the depth of each sub-tree is determined automatically. This model (hierarchical topic model) discovers relations of specialization and generalization between topics. The Pachinko allocation model [101] is an extension of LDA that finds arbitrary relations between topics and terms.

There are other extensions that are incorporating document meta-data. For instance, the author topic model [129] includes the authorship information. This model can discover a multinomial distribution over topics for each author and a multinomial distribution over words for each topic.

Despite their success, the extended topic models are often slow in inference due to a very large parameter space.

3.2.3 Use case: Language model and LDA example

To illustrate the process of generating a topic model (LDA) for a particular document we look at a paper from CiteUlike corpus[2] "The metabolic world of Es-

[2] www.citeulike.org

cherichia coli is not small" shown in figure 3.5. The extracted textual data of the paper example is shown in figure 3.6.

After removing common stop words, we can optionally apply stemming or lemmatization. In our example, we only removed common stop words. Since we are interested in unigram language model, the context of each word or the correct ordering is neglected. The results of this step are shown in figure 3.7. This Bag-of-Words (BOWs) representation of the documents serves as the input for the LDA process to generate the topic model.

The language model of our paper example can be seen in figure 3.8. In our example we use a corpus consisting of 100 papers. This corpus information is needed for smoothing methods that could be applied to the language model. For instance, a term not appearing in our paper but in all other papers in the corpus might get a higher probability in the "metabolic world of Escherichia" language model than a term appearing in only one other paper.

When applying LDA to generate latent topics, we take the BOWs representation of all documents in the corpus. In our example, we chosen to extract 50 topics. The document distribution of these topics for our paper example can be seen in figure 3.9. Only the top 10 most likely topics are displayed. In figures 3.10 and 3.11 the document distributions of terms for topics 35 and 10 are shown. They cover different aspects about "metabolism of living organisms". Topic 35 clearly relates to protein interactions, topic 10 contains terms about flexibility and change.

3.2.4 Topic model and recommender systems

LDA was applied to the recommendation task to suggest textual items.

Various extensions of the basic LDA have been applied in recommender systems [162, 161]. In [162] the authors proposed the Latent Aspect Rating Analysis (LARA) which is a probabilistic rating regression model applied to textual reviews on hotels to estimate aspect ratings (e.g., ratings on cleanliness or sleep quality). The generative model assumes that the reviewer forms a rating on an aspect based on the sentiments words she/he used to discuss that aspect. In [122] the authors proposed a LDA-based behavior-topic model (B-LDA) in Twitter which jointly models user's topics of interest in the tweets content and her/his behavioral patterns (interaction with tweets such as post, retweet, mention and reply) and generates topics which are identified with a dominant behavior. This model was applied to recommend followees for the target user. In [73] the authors proposed a modified LDA probabilistic generative model for collaborative tagging, which clusters tags and users simultaneously, to generate user and community interest information from the LDA model, and employed that informa-

The metabolic world of *Escherichia coli* is not small

Masanori Arita*

Department of Computational Biology, Graduate School of Frontier Sciences, University of Tokyo, Precursory Research for Embryonic Science and Technology, Japan Science and Technology Agency, and Computational Biology Research Center, National Institute of Advanced Industrial Science and Technology, 5-1-5 Kashiwanoha, Kashiwa 277-8561, Japan

Edited by Michael S. Waterman, University of Southern California, Los Angeles, CA, and approved December 9, 2003 (received for review October 7, 2003)

To elucidate the organizational and evolutionary principles of the metabolism of living organisms, recent studies have addressed the graph-theoretic analysis of large biochemical networks responsible for the synthesis and degradation of cellular building blocks [Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N. & Barabási, A. L. (2000) *Nature* 407, 651–654; Wagner, A. & Fell, D. A. (2001) *Proc. R. Soc. London Ser. B* 268, 1803–1810; and Ma, H.-W. & Zeng, A.-P. (2003) *Bioinformatics* 19, 270–277]. In such studies, the global properties of the network are computed by considering enzymatic reactions as links between metabolites. However, the pathways computed in this manner do not conserve their structural moieties and therefore do not correspond to biochemical pathways on the traditional metabolic map. In this work, we reassessed earlier results by digitizing carbon atomic traces in metabolic reactions annotated for *Escherichia coli*. Our analysis revealed that the average path length of its metabolism is much longer than previously thought and that the metabolic world of this organism is not small in terms of biosynthesis and degradation.

bioinformatics | metabolism | small-world network

According to the formal definition, in a small-world network, (*i*) most nodes (metabolites in our case) have a low connection degree, and the degree distribution follows a power law also referred to as scale-freeness; (*ii*) high-degree nodes, called hubs, dominate the network, and most nodes are clustered around hubs; and (*iii*) the average path length (AL; i.e., the average of the shortest path length over all pairs of nodes in the network) remains the theoretical minimum, that of a random graph (1–3). Because of its topology with few hubs, a small-world network may be resistant to random failures: any peripheral node is likely to have a low connection degree and is therefore expendable. In biological networks, the hubs are thought to be functionally important and phylogenetically oldest (4–6).

Although several groups confirmed the small-world property of small-molecule metabolisms in multiple data sources, the details of their results differ depending on the purpose of the analysis and its data-preparation scheme (4–9). Notable differences are attributable to the reversibility of enzymatic reactions and to the treatment of metabolically ubiquitous compounds referred to as coenzymes or inorganics. Table 1 summarizes differences in the major analyses and compares the AL and hub metabolites they identified.

All of these studies used the same algorithmic procedure, and discrepancies are ascribable to the different aims of their network analyses. Jeong *et al.* (7) computed the proximity of metabolites by regarding all substrates and products in the same reaction as adjacent (Fig. 1; see also Fig. 7 and *Supporting Text*, which are published as supporting information on the PNAS web site). Wagner and Fell (5) computed stoichiometric relationships to estimate the transmission degree of perturbations in the metabolic network. They used the metrics with and without coenzymes such as ATP and NAD in both substrate- and reaction-based networks to compare their differences. Ma and Zeng (8) manually specified links in each reaction, aiming to delineate only physical relationships responsible for biosynthesis and degradation. To reproduce biochemical pathways in the traditional metabolic map, however, metabolites to be linked cannot be defined *per se* by compounds or reactions. The biochemical link between metabolites is context-sensitive; it depends on the conserved structural moieties in the adjacent reactions. To accurately compute the reaction connectivity as in the traditional metabolic map, we used digitally compiled atomic mappings, i.e., atomic position pairs between substrates and products corresponding to the substructural moieties conserved in each reaction (Figs. 1 and 2) (10). With this information, we reassessed the global properties of metabolic networks with special emphasis on the small-world hypothesis.

Methods

Definition of Metabolic Pathways. In this work, a metabolic pathway (pathway for short) from metabolite X to Y is defined as a sequence of biochemical reactions through which at least one carbon atom in X reaches Y. Only carbon atoms are considered throughout this article. A metabolite Y is called reachable from X if there is a pathway from X to Y.

Preparation of Reaction Data. The reaction formulas annotated for *Escherichia coli* were originally collected from the Kyoto Encyclopedia of Genes and Genomes (www.genome.ad.jp/kegg), the Encyclopedia of *Escherichia coli* K12 Genes and Metabolism (http://ecocyc.org), and BRENDA (www.brenda.uni-koeln.de) databases and Enzyme Nomenclature (www.chem.qmw.ac.uk/iubmb/enzyme) (11–14). For a gene annotation with a specific EC number of the enzyme hierarchy, the corresponding reaction formulas were collected. If additional metabolites (other than those in the registered reaction formulas) were described in the comment section of the EC definition, the corresponding reaction formula was extrapolated and also included in our data set. For a gene annotation with an incomplete EC number, such as EC 1.1.1.- (the hyphen is a "do not care" symbol), the corresponding reaction formulas were collected from the pathway maps of the aforementioned databases. When the formula in the pathway maps coincided with a specific EC number (e.g., EC 1.1.1.100), the incomplete number was overwritten with the specific number. In the curating process, we attempted to match as many *E. coli* genes with specific reaction formulas as possible. The reactions remaining with incomplete EC numbers were labeled with our original numbers, starting from 999, to distinguish annotated genes with the same (incomplete) EC numbers from each other. Some spontaneous reactions also were included in our data set. The direction of reactions was made to conform to the direction of the arrow in the Roche Applied Science Biochemical Pathways chart (15). All reactions underwent the following process to detect the atomic correspondents between substrates and products on either side: (*i*) resolution of synonyms for molecule names in the data set, (*ii*) substitution of generic molecule names with concrete ones (e.g., ethanol or methanol for alcohol), (*iii*) balancing the number of atoms on either side (hydrogen atoms were not considered), and (*iv*) rearranging

This paper was submitted directly (Track II) to the PNAS office.

Abbreviation: AL, average path length.

*E-mail: arita@k.u-tokyo.ac.jp.

The metabolic world of Escherichia coli is not small

To elucidate the organizational and evolutionary principles of the metabolism of living organisms, recent studies have addressed the graph-theoretic analysis of large biochemical networks responsible for the synthesis and degradation of cellular building blocks.

In such studies, the global properties of the network are computed by considering enzymatic reactions as links between metabolites. However, the pathways computed in this manner do not conserve their structural moieties and therefore do not correspond to biochemical pathways on the traditional metabolic map.

In this work, we reassessed earlier results by digitizing carbon atomic traces in metabolic reactions annotated for Escherichia coli. Our analysis revealed that the average path length of its metabolism is much longer than previously thought and that the metabolic world of this organism is not small in terms of bio-synthesis and degradation.

Figure 3.6: The title and abstract of the paper.

metabolic world escherichia coli small
elucidate organizational evolutionary principles metabolism living organisms recent studies addressed graph theoretic analysis large biochemical networks responsible synthesis degradation cellular building blocks
studies global properties network computed considering enzymatic reactions links metabolites pathways computed manner conserve structural moieties correspond biochemical pathways traditional metabolic map reassessed earlier results digitizing carbon atomic traces metabolic reactions annotated escherichia coli analysis revealed average path length metabolism longer previously thought metabolic world organism small terms bio synthesis degradation

Figure 3.7: Bag-of-Words of the paper.

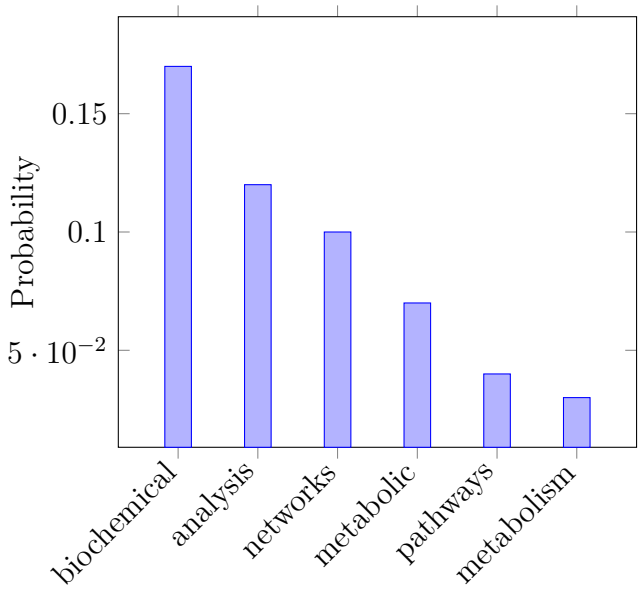


Figure 3.8: Language model for example paper (top terms)

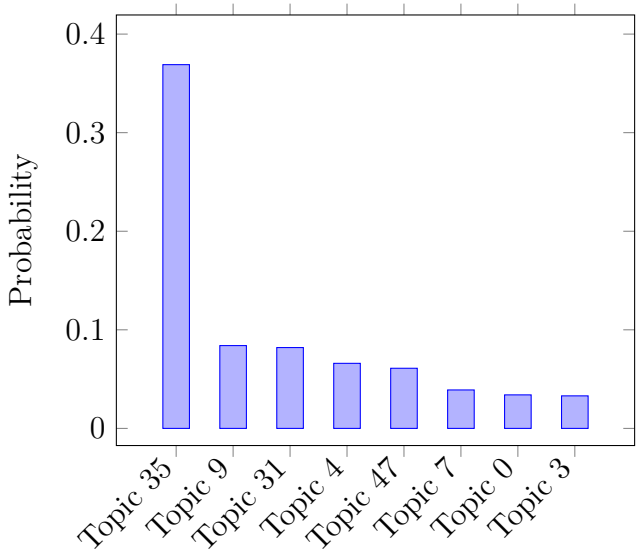


Figure 3.9: Topic model for paper (top topics)

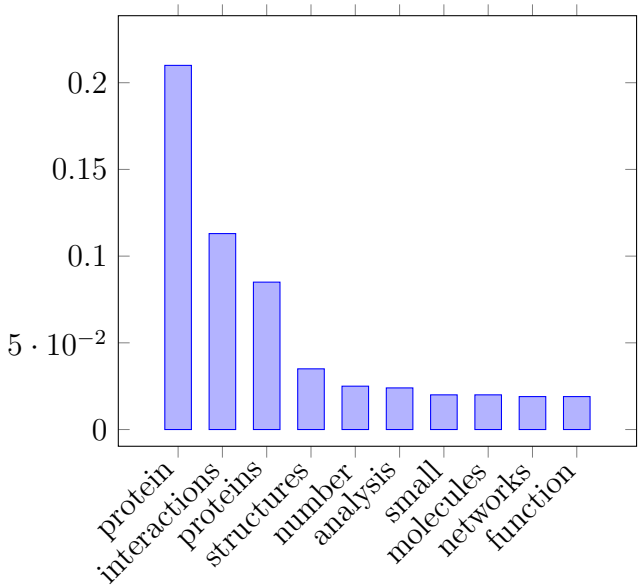


Figure 3.10: Topic 35 for CiteUlike dataset (top terms)

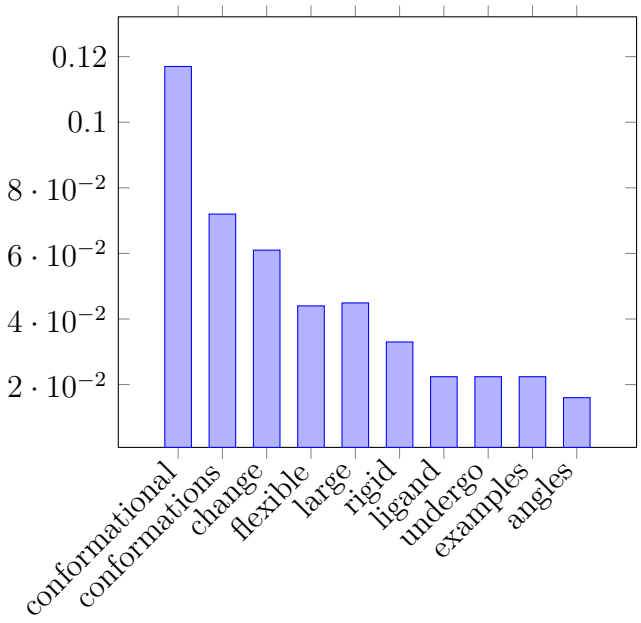


Figure 3.11: Topic 10 for CiteUlike dataset (top terms)

tion to recommend items to users. In [73] the authors proposed a modified LDA probabilistic generative model for collaborative tagging, which clusters tags and users simultaneously, to generate user and community interest information from the LDA model, and employed that information to recommend items to users. In [139] the authors proposed a social recommender system of items, users and groups based on the tagging vocabulary of the target user by applying LDA to her/his tags. This model builds a VSM [133] that represents users and items by using the topic distributions over tags within social networks. Then, it compares the user feature vectors with items to obtain the ranked list of recommendations.

On the other hand, the authors in [130] proposed to apply the classic LDA on user reviews and exploit the extracted topics to define user and item models. The rating scores are simply computed by the sum of the products of item and user profiles for each topic. In [95] the authors proposed a LDA-based approach for collective tags recommendation. LDA is applied to extract topics from tags previously assigned by users to resources. Then, tags are recommended to new resources from these topics. A similar method is applied to tweets in [66] where the authors proposed a twitter hashtag recommender system. Given a tweet, LDA is applied to generate its topic distribution, and then top keywords are recommended from the dominant topics to this tweet as hash-tags. In [59] the authors proposed a LDA-based model for music recommendation. This method uses LDA to describe a listening session as a topic distribution and then compares the topic distributions of sessions by using the Kullback-Leibler (KL) divergence [97, 51]. The KL divergence scores are used in the user-based CF model (Formula

2.2 considering sessions as users) to determine how likely a song will fit in the current active session.

3.2.5 Topic model validation

The quality and efficiency of the topic models must be evaluated [124]. Topic validation approaches have been developed to compare the quality of different topic models [160, 124]. The first approach was proposed to evaluate topic models is based on the perplexity measure which calculates how well the topics extracted by topic model using the training documents allows to predict the occurrence of words belonging to validation papers [160].

The perplexity measures how well the topics extracted by LDA using the training papers (in-sample papers, i.e., a portion of D), allows to predict the occurrence of words belonging to validation papers (out-of-sample papers, i.e., the papers belonging to D which are not used by LDA to extract topics). Perplexity is defined as follows [160]:

$$Perplexity(D^{out-of-sample}) = exp\{-\frac{\log p(D^{out-of-sample}|p(w|1:K))}{|D^{out-of-sample}|}\} \quad (3.6)$$

where $D^{out-of-sample}$ is the set of out-of-sample papers belonging to D. A lower perplexity value means that the model explains the natural language text in a good way.

Other approaches focus on the semantic coherence of topics. For instance, in [45] the authors introduced human validation of topical coherence via intrusion tests. The judges had to find the intruder in the evaluated topics and if the intruder was easy to detect it means that the other words had a strong thematic correlation. However, the process requires manual validation of every built model.

Automatic approaches have been proposed in [116] by using the Point-wise Mutual Information (PMI) to calculate the co-occurrence in a sliding window of 10-words over Wikipedia data[3] / Google search results for all given word pairs in the topic. This approach achieves similar results as human judgments.

In [7] the authors introduced heuristic measures of topical significance. They identified three definitions of junk and insignificant topics. Then, they quantified the difference between a learned topic and an insignificant topic distribution.

[3] en.wikipedia.org/wiki/Main_Page

Notations

In chapters 4, 5 and appendix A, we adopt the following notations:

- $A = \{A_1, A_2, \ldots, A_n\}$ is the set of target researchers.
- n is the number of target researchers.
- A_i is a generic target researcher (the researcher to whom we want to provide paper recommendations). We use the terms researcher, user and author interchangeably.
- Q is the set consisting of all papers rated / written by at least one target researcher in A, i.e., $Q = \cup_{i=1}^{i=n} Q_i$.
- Q_i is the set N_i of papers rated / written by a researcher A_i.
- $D_i = \{d_1, d_2, \ldots, d_M\}$ the set (consisting of M papers) that contains the papers not rated by the target researcher A_i.
- $D = \cup_{i=0}^{n} D_i$
- W_i be the vocabulary employed in $D_i \cup Q_i$.
- K is the number of topics.
- K_i is the number of topics of a researcher A_i.
- M_d is the language model of a paper d.

It is worthwhile to mention that Q_i and D_i are disjoint sets, i.e., $Q_i \cap D_i = \emptyset$. Furthermore, in recommender systems it is well known that we cannot ensure that the researcher A_i is unaware of the papers in D_i, i.e., the papers she/he did not rated. This means that it could be the case the researcher A_i read some papers in D_i but she/he made the decision not to rate them.

Chapter 4

A CBF Approach to Scientific Paper Recommendation

In this chapter, a CBF approach to scientific paper recommendation is proposed. The core idea of this approaches is to exploit the topics related to the researcher's scientific production (authored papers) to formally define her/his profile; in particular we propose to employ topic modeling to represent the user profile. We present a preliminary evaluation of our approach on the DBLP.

4.1 Introduction and Motivations

As mentioned in section 2.1.2.2 (Chapter 2), CBF methods recommend items based on both the items content and a profile that formally represents the user interests [103]. They exploit the items metadata and content to provide recommendations based on users preferences represented in the user profile. However, CBF methods need reliable sources of the user preferences; these preferences must be captured either by means of an explicit user involvement or by means of the analysis of various kinds of interactions of the user with the system (implicit feedback).

In e-commerce applications, the user generated content in comments includes explicit preferences of users regarding their opinions on items. For instance, Amazon users rate and comment about the product after purchasing it, and subsequently other users may consider those ratings and comments before selecting the product for their purchases.

However, in recommender systems of scientific papers, the user generated content (i.e., scientific papers) do not imply opinions, only conveying topics. In her/his task of writing papers, a researcher focuses on a set of topics related to her/his scientific investigations, and s/he uses a technical language related to those topics. These core topics play an important role in the selection of papers

related to her/his preferences.

The rationale behind the approach we propose is to make use of the researcher's scientific corpus to formally define her/his profile: in this way the user profile will exploit the core concepts contained in the papers authored by the researcher.

In this chapter, we make the assumption that the user scientific corpus (publications co-authored by the researcher) is a reliable source for the user modeling.

Most of CBF approaches to recommend scientific papers formally represent the user model as keywords, represented by a vector [115, 145, 86]. For instance, in [86] both researchers and new papers to recommend are represented as TF-IDF vectors. In [44] the authors proposed a concept-based user profile, based on the concepts of the ACM's Computing Classification System (CCS).

In this chapter, the CBF approach we propose relies on topic modeling: the profile of a researcher is a topic model obtained by applying LDA to the papers written by the researcher. Then, the topics generated by topic modeling from the researcher's collection and the language models of the new papers are used to assess their similarity by using a language modeling to IR model [53], which is adapted to the CBF recommendation mechanism.

The aim of this chapter is to apply the topic modeling to the researcher's publications to represent the user model. The proposed CBF approach is presented in section 4.2. The evaluation is presented in section 4.3, with a description of the employed dataset and a discussion of the obtained results. In section 4.4 we draw some conclusions.

4.2 The Proposed Approach

In this section we introduce our proposed CBF approach to scientific paper recommendation to address the issues pointed out in section 2.2.2 (Chapter 2).

The rationale behind our approach is that the generation of the researcher profile should rely on the content generated by the researcher her/himself, as it exposes the topics of interests of the researcher, as well as the technical language s/he uses to generate her/his publications. The researcher profile is then conceived as a mixture of topics extracted by the LDA algorithm from the researcher past publications.

To estimate if a new paper could be of interests to the researcher, a formal representation of the new paper by a language model is provided, which is then compared with each topic characterizing the researcher profile (we remind that a topic is formally represented as a probability distribution over the considered vocabulary, as also a language model is).

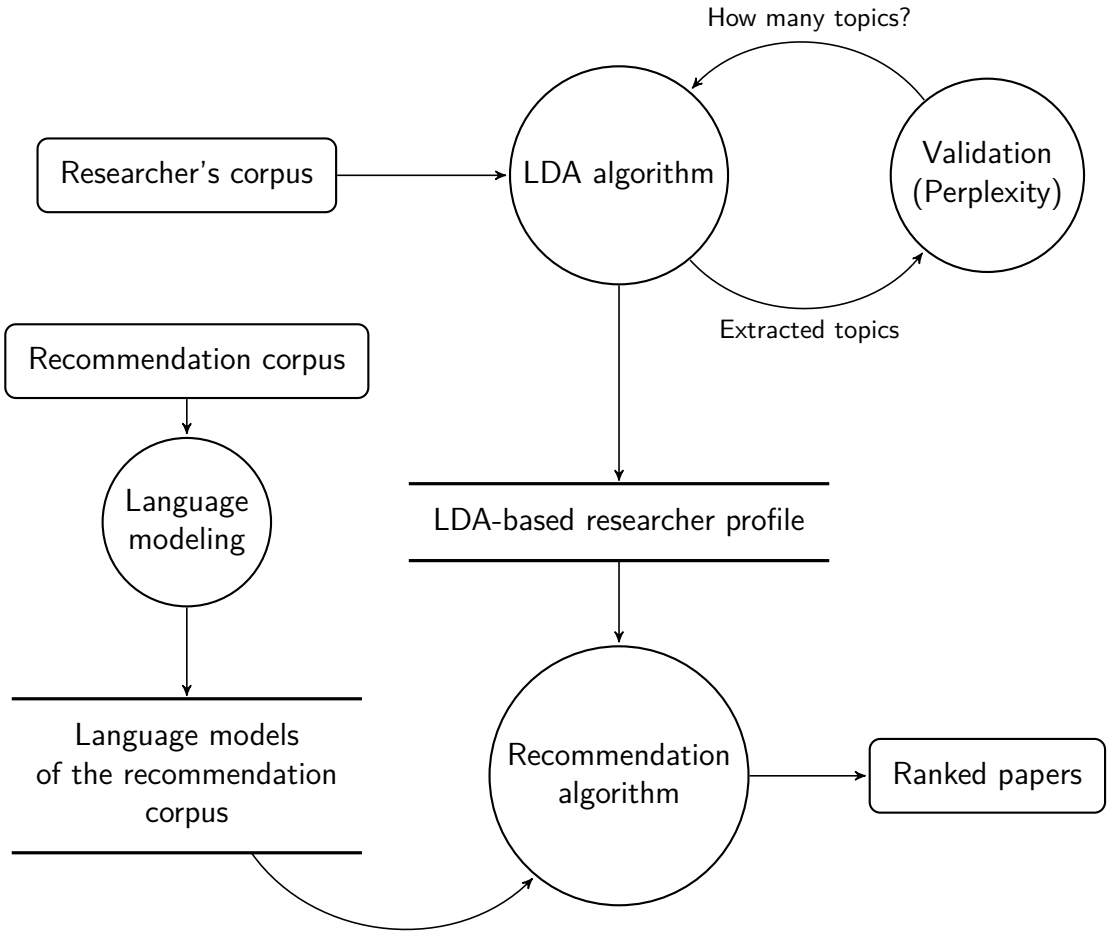


Figure 4.1: Process Flow of our Recommendation Model

The objective of the CBF algorithm is to recommend to researcher A_i the top m papers from the recommendation corpus D_i based on her/his user model. We address this task by means of the following sub-tasks (see figure 4.2); *i)* to define the topic model representing the researcher's interests: as previously outlined this is done by applying the LDA algorithm to the researcher's corpus Q_i, *ii)* to validate the topics extracted by the LDA algorithm form the researcher's corpus, *iii)* to evaluate if a given paper d_j from corpus D_i has to be recommended to researcher A_i; to this purpose the distance between the validated topics and the language model of the paper to be recommended is computed, and *iv)* to rank papers belonging to D_i in descending order of similarity and recommend, to researcher A_i, the first m papers in the provided ranking. In the following sections the above sub-tasks will be detailed.

4.2.1 Generation of the researcher profile and topic validation

Sub-tasks *i)* and *ii)* are performed by applying LDA to the texts extracted from the researcher's corpus Q_i for each researcher A_i. In particular, to select in a distinct way the optimal number of topics K_i for each researcher A_i the researcher's corpus Q_i is cross validated. More specifically, the optimal number of topics K_i for a given researcher A_i is selected by optimizing the cross validated perplexity, where perplexity [160] measures the uncertainty in predicting the occurrence of a word when using a given model.

In topic modeling, the perplexity measures how well the topics extracted by LDA using the training papers (in-sample papers, i.e., a portion of Q_i), allows to predict the occurrence of words belonging to validation papers (out-of-sample papers, i.e., the papers belonging to Q_i which are not used by LDA to extract topics). Perplexity is defined as follows [160]:

$$Perplexity(Q_i^{out-of-sample}) = exp\{-\frac{\log p(Q_i^{out-of-sample}|p_i(w|1:K_i))}{|Q_i^{out-of-sample}|}\} \quad (4.1)$$

where $Q_i^{out-of-sample}$ is the set of out-of-sample papers belonging to Q_i.

4.2.2 The recommendation algorithm

Step iii) of the proposed procedure consists of computing, for each researcher, the similarity between her/his K_i validated topics and the language model computed for each paper in the recommendation corpus D_i.

Formally, we propose to define the similarity between the profile of the researcher A_i and the paper $d_j \in D_i$ as the maximum value among the K_i similarity values between the language model of paper d_j and the K_i topics associated with the profile of author A_i. As each topic is represented as a probability distribution over words (as produced by the LDA algorithm), the similarity between a topic in the LDA-based researcher profile and the language model representing the paper to be recommended is defined by exploiting the *Symmetrized Kullback Leibler divergence* between the above probability distributions. The language model associated with the new paper $d_j \in D_i$ is computed as follows:

$$p(w|d_j) = \frac{nocc(w, d_j) + \frac{\mu nocc(w,Q_i)}{\sum_w nocc(w,Q_i)}}{\sum_w nocc(w, d_j) + \mu} \tag{4.2}$$

where $nocc(w, Q_i)$ is the number of occurrences of word w in Q_i, and μ is the hyperparameter of the Dirichlet distribution. Indeed, formula 4.2, which is known as *Bayesian smoothing using Dirichlet priors*, does not incur the black swan paradox; i.e., a word w not occurring in the researcher's corpus Q_i is assigned a null probability value.

Given the topic distribution $p_i(w|k)$, i.e., the probability distribution over words w associated with topic k, extracted by LDA using the corpus of papers (co-)authored by researcher A_i, and the language model $p(w|d_j)$ associated with paper $d_j \in D_i$, we compute the Symmetrized Kullback Leibler divergence between the topic k and the paper d_j as follows:

$$SKL(k, j) = \frac{1}{2} \sum_{w \in W_i} p(w|d_j) \log \frac{p(w|d_j)}{p(w|k)} + \frac{1}{2} \sum_{w \in W_i} p(w|k) \log \frac{p(w|k)}{p(w|d_j)} \tag{4.3}$$

Then, for each researcher A_i and paper $d_j \in D_i$, we find the topic k^* which minimizes the Symmetrized Kullback Leibler divergence 4.3 across all the K_i validated topics associated with researcher A_i. Then, the similarity between researcher A_i and paper d_j is defined as follows:

$$Similarity(i, j) = \frac{1}{SKL(k^*, j)} \tag{4.4}$$

where we assume $SKL(k^*, j) \neq 0$ for each paper $d_j \in D_i$. Formula 4.4 corresponds to an optimistic computation of the similarity between researcher A_i and paper $d_j \in D_i$.

Indeed, we are assuming that each researcher is summarized by a single topic k^*, i.e., the topic which is the most similar to the language model associated with the considered paper $d_j \in D_i$.

Once we have computed for each paper $d_j \in D_i$ the similarity 4.4, we rank

papers of the recommendation corpus D_i in descending order of similarity and use the ranking to recommend papers to researcher A_i. We then apply the same procedure to all researchers to implement step *iv)* of the proposed procedure.

4.3 Evaluation Experiments

In this section we describe the experimental evaluations that we have conducted to verify the effectiveness of our approach. We first present the dataset and pre-processing step followed by details of the experimental procedure.

4.3.1 Dataset

We used the dataset of ArnetMiner[1], which contains 1.5 million papers from DBLP and 700 thousand authors. We have preprocessed this dataset to select only papers with complete titles and abstracts [175]; we denote this reduced set by $\mathcal{L}$ with $|\mathcal{L}| = 236,012$. To the purpose of our evaluations we have randomly selected 1,600 authors. We denote the set of the considered authors as $A = \{A_1, \ldots, A_{1,600}\}$ with $|A| = 1,600$ and we denote by $Q_{A_i} = \{q_1, \ldots, q_{N_A}\}$ the set of papers written by author A_i with $N_{A_i} \geq 10$.

We build the profile for each author based on her/his scientific production, namely the papers s/he (co-)authored by using the MALLET topic model API[2]. We have applied the following pre-processing steps to the titles and abstracts of the author's scientific production. First, we eliminated any words occurring in a standard stop list. Then, we converted the abstracts to a sequence of unigrams. To the purpose of defining a feasible test set, we have assumed that citations in papers written by a researcher A_i represent her/his preferences. We denote the test set by C, and its cardinality is defined as:

$$|C| = \sum_{A \in U} |C_A| = 24,000$$

where $C = \{d_j \in \mathcal{L} | \exists q_i \in Q, (q_i \rightarrow d_j)$ and $d_j \notin Q\}$ where $q_i \rightarrow d_j$ means q_i cites d_j.

4.3.2 Metrics

Two possible metrics to quantitatively assess the effectiveness are precision and recall. However, as the unrated papers (false positives) in the test set are unla-

[1] aminer.org/citation
[2] mallet.cs.umass.edu/index.php

beled. It is not possible to establish if they are known by the user. This makes it difficult to accurately compute precision.

Hence, the measure we have used to assess the effectiveness of the proposed algorithm is recall. In particular, we have performed a comparative study to evaluate our approach with respect to the CTR and LDA-based recommendation models.

4.3.3 Baselines

We compare our CBF approach to the LDA-based approach presented in [86] and to the CTR approach [161] reported in section 2.2.3 (Chapter 2).

4.3.3.1 LDA-based recommendation model

This approach requires a target user to provide a paper q to receive recommendations i.e., it neglects the user modeling step (55% of scientific paper recommendation approaches [114, 157, 61, 115] reviewed in [20] required users to explicitly provide text snippets, single papers or keywords to receive recommendations). LDA is applied to the entire corpus i.e., $q \in D_i$. Both paper q and the candidate paper to recommend d_j are represented by document distributions over topics, $p(k|q)$ and $p(k|d)$. The cosine similarity between the paper q and each paper in the candidate list $d \in D_i$ is calculated and the papers are ranked in decreasing order of similarity.

4.3.3.2 CTR model

CTR model [161] combines LDA and matrix factorization into a single method, where item latent factors are obtained adding an offset latent variable to the item topic distribution. The latent variable is optimized with an Expectation Maximization (EM) algorithm, together with LDA and Matrix Factorization (MF) parameters.

4.3.4 Parameters

To the purpose of our experiments, we have selected the optimal number of topics K_i for each researcher by optimizing the cross validated perplexity as described in section 4.2.1 with 5-cross validations. We used the left-to-right method defined in [160] to compute the perplexity.

The value of μ in the language model presented in section 4.2.2 is a value determined empirically, and it is set to $\mu = 0.000001$.

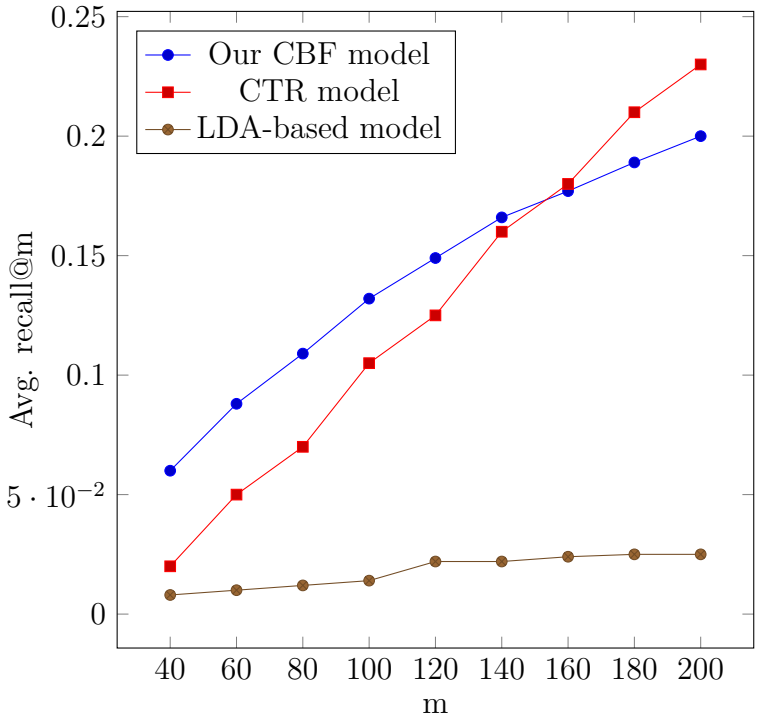


Figure 4.2: Average recall@m results of different models for 1,600 researchers.

4.3.5 Results

We compare the average recall@m results for 1,600 researchers produced by our approach to the results produced by the CTR model. We report the averaged results of 5 repeated experiments to measure the performance of the different methods. Figure 4.2 shows the comparison of the average recall@m values for 1,600 researchers with $m \leq 200$.

Our approach achieves better recall@m values than CTR and LDA-based systems. For $m = 100$, our approach performs better with a 25.7% improvement over CTR.

We also report the item coverage results (presented in section 2.1.4.2, chapter 2) in figure 4.3. The CTR model achieves better coverage results over the proposed CBF model and LDA-based model. CTR is a dimensionality-reduction-based CF which uses for recommendation latent features of users and papers instead of observed ratings [93]. Thus, new relations can be identified between users and papers which consequently increase the item coverage.

As explained in section 2.2.3 (Chapter 2), the CTR system is not able to make accurate recommendations to researchers who use few metadata (rates) to create the user model.

The LDA-based recommendation model proposed in [86] neglects the user modeling (identical to a classic search system) which led to low recall and coverage

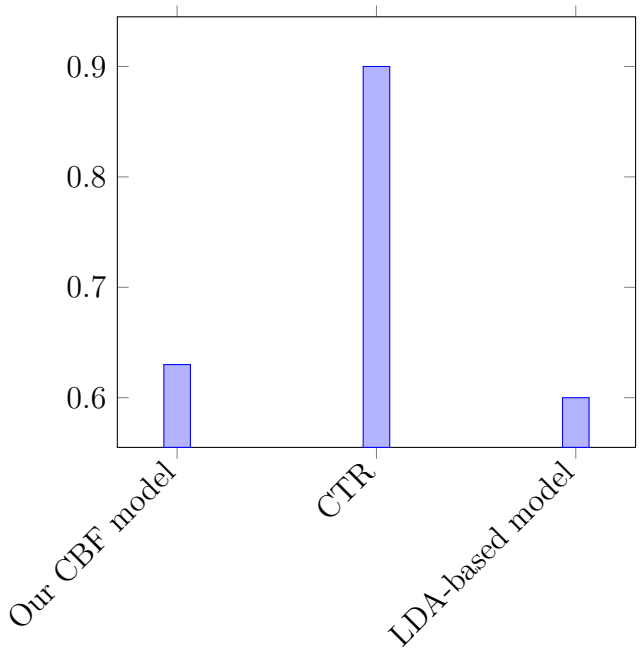


Figure 4.3: Coverage results of different models for 1,600 researchers.

results.

The advantage of our approach does not only consist in being as accurate as or better than other recommendation approaches, but in employing a researcher profile that is only based on past publications and therefore using few metadata (only content item) and alleviate the cold start problem for new items. Furthermore, our approach offers ways to better explain researchers why a specific paper is recommended.

4.4 Summary

In this chapter we have proposed CBF approach to scientific papers recommendation based on the researchers past publications. The researcher profile is built upon the topics generated by the LDA algorithm on the researchers publications corpus. The profile built by this technique is easily interpretable, and it can explain the recommendation results.

Our preliminary experiments show that our approach is performing well compared to the state-of-the art model CTR, which uses a huge number of ratings to provide recommendations.

Chapter 5

A Graph-based Approach to Scientific Paper Recommendation

In this chapter, a neighbors selection approach based on the topics of researchers is presented. In an attempt to address the second question raised in section 1.3 (Chapter 1), a graph-based approach to scientific paper recommendation is presented, which leverages the topics of the researchers and the relations among researchers in the same community.

5.1 Introduction and Motivations

CF approaches suggest to the target user items that similar users liked. The similarity between users is calculated based on the similarity of ratings [127]. Indeed (as mentioned in section 2.1.2.1, Chapter 2), CF approaches only use the user-item rating matrix to suggest items. However, in scientific domain there are usually less users than papers which results in the data sparsity problem. Consequently, the number of ratings is low, further there are few users who select the same papers. Thus, finding similar users only based on explicit ratings of papers is a difficult task. The similarities between users based on similar ratings of papers would fail to capture in an accurate way the preferences similarity between researchers.

We illustrate the raised issue with the following example:

Let $\{q_1, q_2, q_3\}$ be three papers:

- q_1 is "an overview on big data technologies"
- q_2 is "an overview on question answering"
- q_3 is "a large-scale evaluation of question answering systems"

Let u be a researcher who has rated q_3 (q_1 and q_2 are unseen) and v a researcher who has rated q_1, q_2 and q_3. While the topics contained in q_1 and q_2 can be mapped

on the topics contained in q_3, u and v have rated only q_3 in common, leading to a small similarity between u and v.

Furthermore, in [25] the authors show that producing accurate recommendations depends not only on the choice of the recommendation algorithm, but also on the quality of the information about users. The noise injected in a user profile affects the accuracy of the produced recommendations. For instance, in [49] the authors studied the reliability of user ratings in recommender systems. They report that users being consistent only 60% of the time when conducting a rate-rerate procedure (asking users to rate items that they have already rated in the past).

In [71] the authors reveal that using relational information from Social Network (SN) relationships for neighbors selection improves the effectiveness of the CF recommendation algorithm. We propose to build a researchers' graph by using topics extracted from the researchers' rated papers. Then, a community detection algorithm is applied to select the neighborhood of each target researcher.

In summary, we propose a hybrid scientific paper recommendation approach, which combines content analysis based on topic modeling for user profiling, SN techniques for neighbors selection, and relevance-based language modeling to CF recommendation [117].

More specifically, the researcher profiles are built based on topics extracted by LDA [33] from the papers they have rated, and a community detection algorithm is applied to group similar researchers according to their related topics. Then, to compute the relevance of a new paper, the researcher profiles are matched against the topics extracted from the papers of similar researcher.

In this chapter, we explore the use of topic modeling to identify the users' neighbors. Section 5.2 presents relevance-based language modeling to CF recommendation. The proposed researcher's graph-based approach to scientific paper recommendation is presented in details in section 5.3. We present and discuss the results of numerical experiments performed by using the DBLP dataset in section 5.4. In section 5.5 we draw some conclusions.

5.2 Relevance-based language modeling to CF recommendation

In [117] the authors proposed an adaptation of the Pseudo-Relevance Feedback framework of IR to the CF recommendation task.

Pseudo-relevance feedback is a technique for expanding the user's query with new relevant terms to improve the retrieval performance [100, 132]. The terms are extracted from the top documents of an initial retrieval result set (this set

of documents is referred to as the pseudo-relevant set). A second retrieval is performed by using a new query that has been obtained by expanding the previous one, based on the contents of the selected top documents it produced; both the expanded query and its results are presented to the user.

In [100] Relevance Models (RM) under the language modeling framework were presented as an effective method for pseudo relevance feedback. In RM the original user search query is seen as a short sample of words obtained from an underlying *relevance model* (probabilities of terms in the relevant documents [100]). To add more terms from the relevance model to the query (query expansion) then it is reasonable to choose those terms with the highest estimated probability given a sample of observed terms generated by the relevance model for the query.

The relevance-based language modeling is adapted to CF recommendation as follows [117]: a user profile (user model) acts as a representation of the user needs, which is equivalent to a query in the IR task. The items previously scored by the user (in the context of CF) act as the query words in the IR pseudo-relevance feedback model. The expansion of queries with new terms in pseudo-relevance feedback becomes in the CF context a kind of user profile expansion problem, where the objective is to expand the user's need representation with further items related to her/his interests. In this way, the problem of recommending items to users can be assimilated to the task of expanding users where the items to be recommended play the role of the candidate expansion terms. On the other hand, the neighborhood of the user (similar users) is modeled as the pseudo-relevant set.

5.3 The Proposed Approach

The differences between the proposed approach and those described in section 2.2.2.1 (Chapter 2), which make use of researcher-paper rating matrix for both researcher modeling and user neighborhood selection, is that in our case topic modeling is applied to both define the researchers' profiles and find their neighbors with similar topics by using SNA techniques. A comparison of the proposed model against other related approaches in terms of their main characteristics (i.e., used input data, model type, alleviating sparsity and cold-start problems and the use of an IR technique) is described in table 5.1.

Ref.	CF memory-based	CF model-based	Rating-based user profile	Topic-based user profile	SN	Using model	IR	Alleviating sparsity problem	Alleviating cold start problem
[161]	No	Yes	Yes	No	No	No	No		Yes
[61, 134]	Yes	No	Yes	No	Yes	Yes	No		No
[117]	No	Yes	Yes	No	No	Yes	No		No
Our model	No	Yes	No	Yes	Yes	Yes	Yes		Yes

Table 5.1: Comparison of the proposed model against related work approaches.

To evaluate if a new paper could be of interest to a researcher, the researcher's relevance model based on the topics of her/his profile and similar researchers' profiles is estimated, then the probability of the paper under this relevance model is calculated.

We propose to achieve this goal by means of the following steps: *i)* Application of LDA to the researcher's corpus Q to generate topics and to define the researcher profiles based on these topics. *ii)* Construction of the researchers' graph where researchers are nodes and similarities between profiles are weights of edges. *iii)* Partition of the researchers' graph into communities consisting of similar researchers to identify the neighborhood V_i of the target researcher A_i. *iv)* Application of LDA to the papers rated by the community of the target researcher A_i and definition of the profiles of neighbors in V_i based on the extracted topics. *v)* Prediction of the relevance of $d \in D_i$ to the target researcher A_i; to this purpose we calculate the probability of paper d given the relevance model R_{A_i} of the target researcher A_i, $p(d|R_{A_i})$ by making use of the profiles of neighbors in V_i. *vi)* Ranking of the papers belonging to D_i in descending order of relevance, and recommendation to researcher A_i of the top m ranked papers.

In sections below, these steps will be detailed.

5.3.1 Construction of the researchers' graph

The construction of the researchers' graph is performed by applying LDA to the researchers corpus Q (*sub-task i)*).

Perplexity is used to assess how well the topics learned based on the LDA model on the training papers (in-sample papers, i.e., a subset $Q_{in} \subset Q$), allows to predict the validation papers (out-of-sample papers, i.e., the papers belonging to $Q_{out} = Q \setminus Q_{in}$, i.e., those papers not used when applying the LDA model to learn topics) [160].

The profile of the target researcher A_i is constructed by aggregating the distributions over topics of all papers she/he has rated. Formally, the researcher profile RP_{A_i} for the researcher A_i is represented by a vector where each component is associated with a topic k (ranging from 1 to K).

Each component $RP_{A_i}(k)$ of RP_{A_i}, contains the probability of the topic k given the target researcher A_i according to [130]:

$$RP_{A_i}(k) = \frac{\sum_{q \in Q_i} p(k|q)}{N_i} \tag{5.1}$$

Sub-task *ii)* consists of building a researchers' graph. In particular, the researcher's graph undirected graph where nodes are the target researchers; the fact that two researchers A_i and A_j are interested to the same topics is represented

by an undirected link l_i j between nodes associated to them. Links are weighted according to the cosine similarity between the distributions over topics (the number of topics is the same for all researchers) of the researchers and it is computed as follows:

$$\begin{aligned} weight(l_{ij}) &= cos\_sim(RP_{A_i}, RP_{A_j}) \\ &= \frac{\sum_{k=1}^{K} RP_{A_i}(k)\ RP_{A_j}(k)}{\sqrt{\sum_{k=1}^{K} RP_{A_i}^2(k)}\sqrt{\sum_{k=1}^{K} RP_{A_j}^2(k)}} \end{aligned} \tag{5.2}$$

5.3.2 Community profiling

Sub-task *iii)* consists of partitioning the researchers' graph into communities of similar researchers.

Due to its computational efficiency, we adopt the community detection algorithm that has been introduced in [34] for weighted graphs. It is a fast-greedy algorithm that is based on the optimization of the modularity measure: a function that evaluates the goodness of partitions of a graph into communities.

Because of researcher's graph partitioning, a target researcher A_i belongs to a community of similar researchers with V_i.

Sub-task *iv)* consists of applying LDA to the papers previously rated by all researchers in the community of the target researcher A_i to generate topics of the community. We calculate then each neighbor's profile RP_v ($v \in V_i$) according to the formula A.7.

5.3.3 The recommendation algorithm

Sub-task *v)* of the proposed procedure consists of estimating the probability of a new paper d in D_i under the relevance model R_{A_i} for a target researcher A_i, $p(d|R_{A_i})$.

In [117] two alternative relevance models were proposed for recommendation: *1)* the RM1-based recommendation model that computes $p(d|R_{A_i})$ under the assumption that the items in the user's profile and the items rated by the user's neighbors are sampled identically and independently from a unigram distribution. *2)* the RM2-based recommendation model where $p(d|R_{A_i})$ is calculated assuming that the items in the user's profile are independent from each other but dependent on the items present in the profiles of the user's neighbors.

We focus on RM2, since it was reported in [117] that it provides the best results. RM2 estimates a relevance model R_{A_i} underlying the target researcher A_i and her/his neighbors V_i ($v \in V_i$) and then it calculates the relevance for each paper d to be recommended given this relevance model R_{A_i}, $p(d|R_{A_i})$ as the

following [117]:

$$p(d|R_{A_i}) \propto p(d) \prod_{q \in Q_i} \sum_{v \in V_i} \frac{p(d|v)p(v)}{p(d)} p(q|v) \tag{5.3}$$

where the priors $p(d)$ and $p(v)$ are assumed to be uniform distributions.

However, in the original paper [117], the probability of an item d given a user's neighbor v, $p(d|v)$ is computed by using the ratings assigned by the user's neighbors to the items.

In the proposed approach the dependency of a paper to the user's neighbor, $p(d|v)$, is calculated by using similar topics in user's neighbor profiles and papers rated by target users.

$p(d|v)$ is calculated by using the KL divergence between the user's neighbor profile, RP_v and the distribution over topics θ_d of the new paper d as the following ($p(q|v)$ is calculated by using the same formula):

$$p(d|v) = 1 - KL(RP_v||\theta_d) \tag{5.4}$$

where

$$KL(RP_v||\theta_d) = \sum_{k=1}^{K} RP_v(k) \log \frac{RP_v(k)}{p(k|d)} \tag{5.5}$$

5.4 Experimental Evaluations

In this section we describe the performed experiments. We first present the employed dataset and the preprocessing steps we applied, and then we present and compare the results of our model to the results produced by the relevance-based language modeling for recommender system [117] and by the PageRank-weighted CF [61].

5.4.1 Dataset and tools

We used the dataset of ArnetMiner defined in chapter 4. To define a feasible dataset for evaluation, we have assumed that citations in papers written by a researcher A_i represent her/his ratings. The final dataset consists of 1,000 users and 59,340 items with 125,578 user-item ratings.

We apply LDA to the papers rated by the researchers (i.e., in training) using the MALLET topic model API[1]. We create the researchers' graph by using Gephi[2]. The obtained researchers' graph is a dense graph. We removed edges

[1] mallet.cs.umass.edu/topics.php
[2] gephi.org

with $weight(l_{ij}) \leq 0.5$ (This value is determined empirically). Then, we apply the community partition algorithm to obtain clusters of similar researchers.

5.4.2 Baselines

We compare our approach to the relevance-based language modeling to CF recommendation reported in section 5.2, and to the the PageRank-weighted CF model presented in [61].

5.4.2.1 Relevance-based language modeling to CF

This approach estimates the conditional probabilities in Equation 5.3, $p(d|v)$ and $p(q|v)$ by smoothing the Maximum Likelihood Estimate (MLE) with the probability of the item in the collection. The MLE is calculated as follows:

$$p_{ml}(d|v) = \frac{r(v,d)}{\sum_{q \in Q_v} r(v,q)} \tag{5.6}$$

where $r(v,d)$ represents the rating assigned by researcher's neighbor v to paper d.

$r(v,d) = 1$ means that the user's neighbor likes (cites) the paper d and $r(v,d) = 0$ means that is unknown in our dataset that the user's neighbor likes or dislikes the paper d. We used the traditional neighborhood selection technique for user-based CF that is based on the k-nearest neighbors (k-NN) algorithm to identify similar researchers V_i.

5.4.2.2 PageRank-weighted CF model

This is the second baseline method [61], which is an extension of the traditional item-based CF model [88]. This model consists of computing the PageRank score of each paper in the citation graph to build the item-item similarity matrix.

5.4.3 Evaluation setup

We performed 5-fold cross-validation. In each fold, 80% of ratings was randomly selected as the training set and the remaining 20% as the testing set. For each researcher, a ranking is generated by estimating a relevance score for every paper in the test set, ignoring the already seen papers.

In order to provide a fair comparison to CF recommendation baselines, we perform an in-matrix prediction where each user has a set of papers that she has not rated, but that at least one other user has rated.

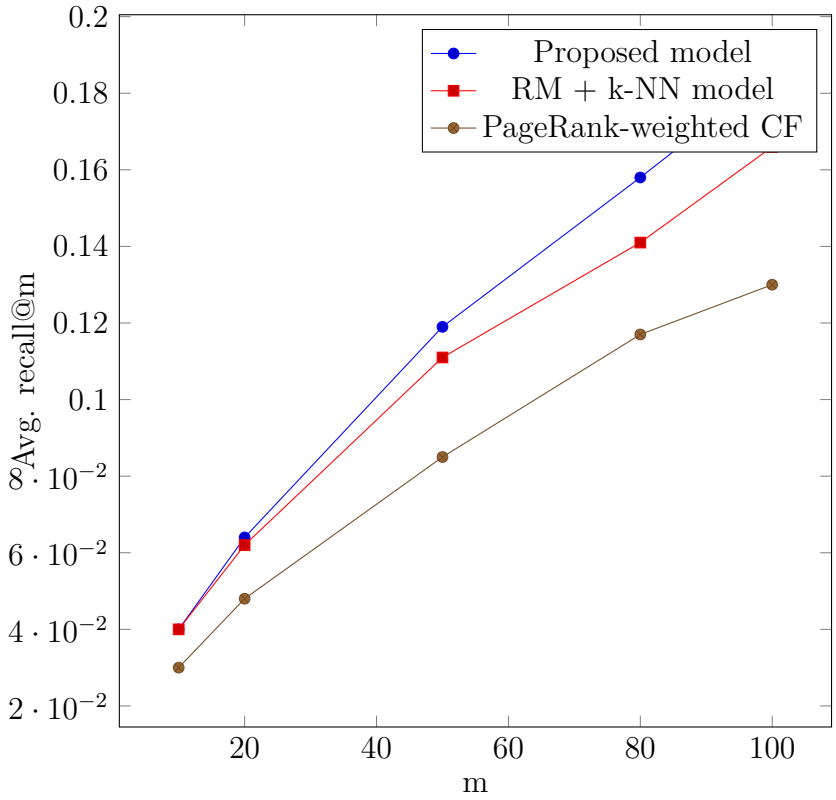


Figure 5.1: Average recall@m results of different models for 1,000 researchers

We apply the recall metric which has been widely used to evaluate recommender systems.

5.4.4 Results

The preliminary results reported in figure 5.1 show that our approach (which combines topic models and relevance modeling over the scientific paper recommendation process) achieves better average Recall@m values than RM + k-NN model and PageRank-weighted CF model. For instance, for $m = 100$, our approach performs better with a 11.4% and 28.5% improvement over RM + k-NN model and PageRank-weighted CF model.

Our approach also alleviates the cold start problem of users by exploiting paper's topics instead of ratings. For instance, a novice researcher can select only a set of topics instead of forcing her to rate papers to get recommendations.

In addition, relevance-based language modeling to CF and PageRank-weighted CF cannot deal with the cold start problem of new items (i.e., when performing an out-of-matrix prediction where new items are introduced to the system and no one has rated them), but our model can alleviate it by considering paper's topics and user's collaborative network.

5.5 Summary

The chapter's contribution is threefold; first, a new neighbor selection model is proposed that includes a novel way of user modeling based on topics in her/his rated papers that discloses her/his preferences. In addition, neighbors are modeled based on topics that indicate their preferences. Second, in order to obtain better neighbors for the user-based recommendation, we proposed the application of a community detection algorithm. This proposal achieves effectiveness improvements over traditional neighbor-based approaches. Third, the relevance-based language modeling is adapted to scientific paper recommendation. The experiments show an improvement in terms of effectiveness (measured by recall) against different related baselines.

Chapter 6

Conclusion

This chapter provides a broad summary of the proposed work in this thesis. We begin by summarizing the main contributions to the research field in section 6.1. We then review our findings with respect to our research questions in section 6.2 followed by summarizing the significance of research outcomes of this thesis in section 6.3. Finally, in section 6.4 we discuss future directions for the research conducted in this thesis.

6.1 Summary of Contributions

Recommender systems have been applied in different applicative domains to suggest items to users by capturing their interests and needs based on their feedback. However, the recommendation of scientific papers is not like other recommendation tasks such as recommending movies or products: the user-item rating matrix is very sparse i.e., the number of co-rated items between users is low; furthermore, ratings are implicitly inferred. A researcher is mainly focusing on the topics of papers that are related to her/his domain.

This thesis offers a contribution to exploit the usage of topics in recommender systems of scientific papers. Specifically, we have contributed to the issue of making use of the topics related to researchers as an effective means to improve both user modeling and user's neighbor selection.

In the following we present a summary of the contributions of this thesis:

1. We proposed a fully CBF recommendation approach for scientific papers based on the researchers past publications. The researcher profile is built upon the topics generated by the LDA algorithm on the researchers publications corpus. The profile built by this technique is easily interpretable, and it can explain the recommendation results. The proposed approach relies on the assessment of the closeness of the language and topics used in

the researchers papers and the one employed in the unseen papers; we have shown that this approach achieves better results than the state-of-the-art recommendation approach CTR.

2. We proposed a relevance-based language modeling for a recommender system of scientific papers. The researchers profiles are built based on the topics extracted by LDA and a neighbor selection method is applied to group similar researchers according to these topics.

6.2 Answers to Research Questions

1. Can topics in authored papers be utilized as reliable sources of knowledge to user and item modeling for recommendation?

   This question has been answered based on the results of the CBF recommendation approach presented in chapter 4. We explored the possibility of using topics in authored papers (researcher corpus) for user and item modeling and tuning these topical-based models in a CBF recommendation process.

   The proposed CBF model achieves better results than the CF model as it is using only a user profile based on authored papers and therefore using few metadata (only content item) and alleviates the cold start problem for new items.

2. How to identify researcher's neighbors by using topics for scientific paper recommendation?

   This question has been answered by the researcher's graph construction step presented in chapter 5 whereby we proposed to generate the user's neighbors based on her/his topics of interest instead of using similar ratings of items. This method addresses the problem of sparsity by calculating similarities between users by using topics of researchers.

6.3 Significance of Research Outcome

The benefits of using topical information for scientific paper recommendation are the following:

- It may be a reliable source for user modeling to deal with the noise injected in user profiles.
- It helps to provide explainable recommendations when characterizing users as a set of topics.

- It deals with the data sparsity problem observed in scientific paper recommendation, in particular in CF approaches.

6.4 Future Directions

As future directions, we aim to extend the proposed recommendation models to include the following proposals:

- ***Topic model validation:*** The semantic quality of learned topics models must be evaluated as mentioned in section 3.2.5. In [116] the authors used topic scoring models based on both the Wikipedia corpus and google results to calculate the coherence of terms of the same set of topics. A future direction could be the use of the citations corpus to validate the learned topics from the authored papers.
- ***Incorporating domain knowledge into user modeling:*** The user model is conceived as mixture of topics that are represented by probability distributions over terms. In [13] the authors assume that a user may prefer additional knowledge about the composition of terms that have high probabilities in various topics. A future direction could be to use concepts of existing ontologies associated with these terms (such as the Gene Ontology [14]) for a richer user profile representation.
- ***Integrating papers metadata:*** Another interesting direction could be in incorporating various attributes of papers such as recency, the venue impact factor, the citation index, the conference field to improve the results of the proposed recommendation models.
- ***Scalability of the recommender systems:*** An interesting aspect with regards of dealing with millions of papers could be the use of scalable technologies and adapting the proposed recommendation algorithms to operate in parallel computing environments [12] such as MapReduce [56] and Spark [173] frameworks.
- ***Group recommender systems of scientific papers:*** When colleagues in a research Lab collaborate to co-author a paper, they need a group recommender system that aggregates models of individual users into a group model to produce group recommendations [36, 107]. In [36] the authors consider the domains as crucial for this type of recommender systems. For instance, we may stress upon the diversity of results in a group recommender systems of scientific papers as papers to recommend should cover topics in each of the individual user model.

Appendix A

The Remaining Proposed CBF Approaches

In this appendix, we present the proposed CBF approaches which did not achieve good results compared to the baseline recommender systems in chapter 4. In section A.1, we present briefly the original language modeling to IR models. In section A.2 we present the proposed recommendation models based on language modeling in detail. Finally, the results are shown in section A.3.

A.1 Language modeling to IR models

In [121] the authors assume that the language model is a representation of the topic (a probability distribution over words) that a user had in mind when she/he was writing a query and it is sampled from an "ideal" document that satisfies her/his information need. Three IR models based on language models are presented in [121, 53].

A.1.1 Language Modeling to IR Models

The query likelihood model ranks documents by the probability that the query text could be generated by the document language model M_d. In other words, the query q would be observed as a random sample from the document model M_d. The IR model specifies the ranking of documents by $p(q|M_d)$, which is calculated by using the unigram language model for the document M_d as follows:

$$p(q|M_d) = \prod_{i=1}^{n} p(q_i|M_d) \tag{A.1}$$

where q_i is a query word and n is the number of words in the query.

A.1.2 Document likelihood model

The document likelihood model consists of calculating the probability of a query language model M_q (relevance model) generating the document. This model assumes that given some examples of relevant documents for a query, the relevance model is estimated and then used to calculate the scores of new generated documents [46].

A.1.3 Comparison model

It consists of comparing both the query and the document language models to calculate how different they are and rank documents by using a general risk minimization approach for document retrieval [174]. Formally, the risk of returning a document d as relevant to a query q is calculated by using the Kullback Leibler divergence between their language models as the following:

$$KL(M_d||M_q) = \sum_{w \in V} p(w|M_q) log \frac{p(w|M_q)}{p(w|M_d)} \tag{A.2}$$

where V denotes the vocabulary.

A.2 The Proposed CBF Approaches

We implemented the three CBF recommender systems based on language modeling framework: **a)** the researcher likelihood model that is based on the probability of generating the user profile from a paper language model, **b)** the paper likelihood model that is based on generating the paper text from the user profile, **c)** comparing the topic models representing the user profile and new papers to recommend.

The three recommendation algorithms are based on the user profile presented in section 4.2.1 (Chapter 4). We present in the following the three CBF algorithms in detail (Sub-task *iii* of the CBF mechanism presented in section 4.2.2).

A.2.1 Researcher likelihood model (Model a)

We consider as a relevant paper to the target researcher A_i the one that it best represents one of her/his topics of interest (topical relevance). For each researcher A_i and new paper to recommend d_j, the probabilities of the topics associated to the researcher's corpus Q_i generated by the paper language model M_{d_j} are calculated: First, for each topic k, given the topic distribution $p(w|k)$, the language model M_{d_j} associated with the new paper to recommend $d_j \in D_i$ is calculated as

follows:

$$p(M_{d_j}|k) = \prod_{w \in d_j} p(w|k)^{\frac{nocc(w,d_j)}{\sum_w nocc(w_{d_j})}} \tag{A.3}$$

where $nocc(w, d_j)$ represents the number of occurrences of word w in paper d_j. Second, the topic k which maximizes $p(M_{d_j}|k)$ (Formula A.3) across all the K_i validated topics associated to the researcher A_i is defined to calculate the ranking scores $p(A_i|M_{d_j})$ as the following:

$$p(A_i|M_{d_j}) = \arg\max_k p(M_{d_j}|k) \tag{A.4}$$

The recommendation for each researcher A_i is the list of papers D_i ranked by decreasing values of $p(A_i|M_{d_j})$.

A.2.2 Paper likelihood model (Model b)

The paper's likelihood model consists of calculating the probability of the new paper d_j generated by the language model of the researcher A_i (i.e., relevance model M_{A_i}).

$$p(d_j|M_{A_i}) = \prod_{w \in d_j} p(w|M_{A_i})^{\frac{nocc(w,d_j)}{\sum_w nocc(w,d_j)}} \tag{A.5}$$

where

$$p(w|M_{A_i}) = \sum_{k=1}^{K} p(w|k)p(k) \tag{A.6}$$

and

$$p(k) = \frac{\sum_{q \in Q_i} p(k|q)}{N_i} \tag{A.7}$$

A.2.3 Comparison between topic models of the researcher profile and candidate papers (Model c)

The comparison model calculates the similarity between the topics of each paper in the researcher's corpus and the topics of the new paper M_{d_j} by using the SKL divergence.

Given the topic distribution $p(k|q)$, i.e., the probability distribution over topic k associated with paper $q \in Q_i$, extracted by LDA by using the corpus of papers (co-)authored by researcher A_i, and the topic model $p(k|d_j)$ associated with paper $d_j \in D_i$, the KL divergence is calculated for each candidate paper d_j as the

following:

$$SKL(M_q||M_{d_j}) = \frac{1}{2}\sum_{k=1}^{K} p(k|d_j)\log\frac{p(k|d_j)}{p(k|q)} + \frac{1}{2}\sum_{k=1}^{K} p(k|q)\log\frac{p(k|q)}{p(k|d_j)} \tag{A.8}$$

For each researcher A_i and paper d_j, the paper q^* that minimizes the SKL divergence across all the papers in the researcher's corpus Q is determined. The similarity between the researcher A_i and paper d_j is defined as the following:

$$Similarity(A_i, d_j) = \frac{1}{SKL(M_{q^*}||M_{d_j})} \tag{A.9}$$

where $SKL(M_{q^*}||M_{d_j}) \neq 0$ for each paper $d_j \in D$.

Herein, we are assuming that each researcher is summarized by a single paper q^* i.e., the paper that has the most similar topics to the topics associated with the considered paper d_j. For each paper $d_j \in D_i$, the similarity function is defined by the formula A.9).

A.3 Experimental Results

We present the average recall@m results of the proposed recommendation models in figure A.1 for 1,600 researchers with $m \leq 200$. The CBF model presented in chapter 4 achieves better recall@m values than the other models.

In models a and b, the new papers are represented by a topic distribution $p_i(w|k)$ (i.e., the probability distribution over words w associated with topic k). However, an unseen paper is very likely to contain words that did not appear in the researcher's corpus Q_i. The LDA model assigns zero probability to such words, and thus it models a and b to give zero probability to unseen papers. A smoothing method must be applied to represent the new paper in these models.

The coverage results of the proposed models are presented in figure A.2 compared to the CBF model introduced in chapter 4.

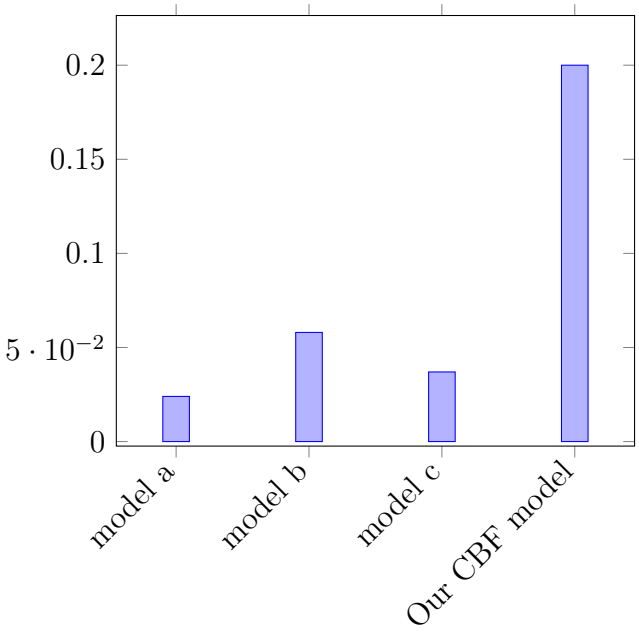


Figure A.1: Average recall@200 results of different models for 1,600 researchers.

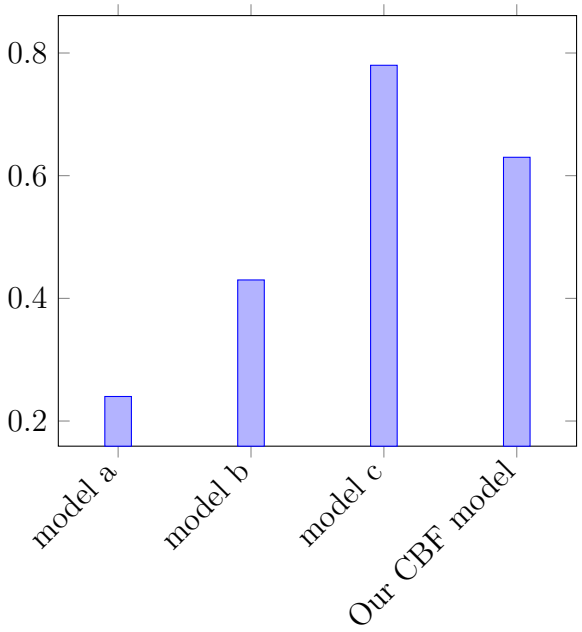


Figure A.2: Coverage results of different models for 1,600 researchers.

Bibliography

[1] Gediminas Adomavicius and Alexander 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.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 191–226. Springer, 2015.

[3] Deepak Agarwal and Bee-Chung Chen. flda: matrix factorization through latent dirichlet allocation. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 91–100. ACM, 2010.

[4] Nitin Agarwal, Ehtesham Haque, Huan Liu, and Lance Parsons. Research paper recommender systems: A subspace clustering approach. In *WAIM*, pages 475–491. Springer, 2005.

[5] Charu C Aggarwal. *Recommender systems*. Springer, 2016.

[6] Charu C Aggarwal, Joel L Wolf, Kun-Lung Wu, and Philip S Yu. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 201–212. ACM, 1999.

[7] Loulwah AlSumait, Daniel Barbará, James Gentle, and Carlotta Domeniconi. Topic significance ranking of lda generative models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 67–82. Springer, 2009.

[8] Anas Alzoghbi, Victor Anthony Arrascue Ayala, Peter M Fischer, and Georg Lausen. Pubrec: Recommending publications based on publicly available meta-data. In *LWA*, pages 11–18, 2015.

[9] Anas Alzoghbi, Victor Anthony Arrascue Ayala, Peter M Fischer, and Georg Lausen. Learning-to-rank in research paper cbf recommendation: Leveraging irrelevant papers. In *CBRecSys RecSys*, pages 43–46, 2016.

[10] Maha Amami, Rim Faiz, Fabio Stella, and Gabriella Pasi. A graph based approach to scientific paper recommendation. In *Proceedings of the International Conference on Web Intelligence*, pages 777–782. ACM, 2017.

[11] Maha Amami, Gabriella Pasi, Fabio Stella, and Rim Faiz. An lda-based approach to scientific paper recommendation. In *International Conference on Applications of Natural Language to Information Systems*, pages 200–210. Springer, 2016.

[12] David C Anastasiu, Evangelia Christakopoulou, Shaden Smith, Mohit Sharma, and George Karypis. Big data and recommender systems. 2016.

[13] David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 25–32. ACM, 2009.

[14] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[15] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[16] Mariem Bambia, Mohand Boughanem, and Rim Faiz. Exploring current viewing context for tv contents recommendation. In *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on*, pages 272–279. IEEE, 2016.

[17] Xinlong Bao, Lawrence Bergman, and Rich Thompson. Stacking recommendation engines with additional meta-features. In *Proceedings of the third ACM conference on Recommender systems*, pages 109–116. ACM, 2009.

[18] Chumki Basu, Haym Hirsh, William Cohen, et al. Recommendation as classification: Using social and content-based information in recommendation. In *Aaai/iaai*, pages 714–720, 1998.

[19] Chumki Basu, Haym Hirsh, William W Cohen, and Craig G Nevill-Manning. Technical paper recommendation: A study in combining multiple information sources. *J. Artif. Intell. Res.(JAIR)*, 14:231–252, 2001.

[20] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4):305–338, 2016.

[21] Joeran Beel, Stefan Langer, Marcel Genzmehr, and Andreas Nürnberger. Introducing docear's research paper recommender system. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pages 459–460. ACM, 2013.

[22] Joeran Beel, Stefan Langer, Georgia Kapitsaki, Corinna Breitinger, and Bela Gipp. Exploring the potential of user modeling based on mind maps. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 3–17. Springer, 2015.

[23] Nicholas J Belkin and W Bruce Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

[24] Alejandro Bellogin, Pablo Castells, and Ivan Cantador. Precision-oriented evaluation of recommender systems: an algorithmic comparison. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 333–336. ACM, 2011.

[25] Alejandro Bellogín, Alan Said, and Arjen P de Vries. The magic barrier of recommender systems–no magic, just ratings. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 25–36. Springer, 2014.

[26] Alejandro Bellogın, Jun Wang, and Pablo Castells. Text retrieval methods for item ranking in collaborative filtering. In *Proceedings of the 33rd European conference on Advances in information retrieval, ECIR*, volume 11, pages 301–306. Springer, 2011.

[27] Steven Bethard and Dan Jurafsky. Who should i cite: learning literature search models from citation behavior. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 609–618. ACM, 2010.

[28] Daniel Billsus and Michael J Pazzani. Learning collaborative information filters. In *Icml*, volume 98, pages 46–54, 1998.

[29] Bo-Christer Björk, Annikki Roos, and Mari Lauri. Global annual volume of peer reviewed scholarly articles and the share available via different open access options. In *ELPUB2008*, 2008.

[30] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[31] David M Blei, Thomas L Griffiths, and Michael I Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):7, 2010.

[32] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[33] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[34] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[35] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013.

[36] Ludovico Boratto. Group recommender systems. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 427–428. ACM, 2016.

[37] 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*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[38] Derek Bridge, Mehmet H Göker, Lorraine McGinty, and Barry Smyth. Case-based recommender systems. *The Knowledge Engineering Review*, 20(03):315–320, 2005.

[39] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[40] Robin Burke. Hybrid systems for personalized recommendations. *Intelligent Techniques for Web Personalization*, pages 133–152, 2005.

[41] Robin Burke and Maryam Ramezani. Matching recommendation technologies and domains. In *Recommender systems handbook*, pages 367–386. Springer, 2011.

[42] Iván Cantador, Alejandro Bellogín, and David Vallet. Content-based recommendation in social tagging systems. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 237–240. ACM, 2010.

[43] Pablo Castells, Saúl Vargas, and Jun Wang. Novelty and diversity metrics for recommender systems: choice, discovery and relevance. 2011.

[44] Kannan Chandrasekaran, Susan Gauch, Praveen Lakkaraju, and Hiep Phuc Luong. Concept-based document recommendations for citeseer authors. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 83–92. Springer, 2008.

[45] Jonathan Chang, Jordan L Boyd-Graber, Sean Gerrish, Chong Wang, and David M Blei. Reading tea leaves: How humans interpret topic models. In *Nips*, volume 31, pages 1–9, 2009.

[46] D Manning Christopher, Raghavan Prabhakar, and SCHÜTZE Hinrich. Introduction to information retrieval. *An Introduction To Information Retrieval*, 151:177, 2008.

[47] Maarten Clements, Arjen P de Vries, and Marcel JT Reinders. Exploiting positive and negative graded relevance assessments for content recommendation. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 155–166. Springer, 2009.

[48] Victor Codina and Luigi Ceccaroni. Extending recommendation systems with semantics and context-awareness: Pre-filtering algorithms. In *CCIA*, pages 81–90, 2011.

[49] Dan Cosley, Shyong K Lam, Istvan Albert, Joseph A Konstan, and John Riedl. Is seeing believing?: how recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 585–592. ACM, 2003.

[50] Alberto Costa and Fabio Roda. Recommender systems by means of information retrieval. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, page 57. ACM, 2011.

[51] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[52] Paolo Cremonesi, Franca Garzotto, Sara Negro, Alessandro Papadopoulos, and Roberto Turrin. Comparative evaluation of recommender system quality. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 1927–1932. ACM, 2011.

[53] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*, volume 283. Addison-Wesley Reading, 2010.

[54] Dario De Nart, Felice Ferrara, and Carlo Tasso. Personalized access to scientific publications: from recommendation to explanation. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 296–301. Springer, 2013.

[55] Dario De Nart and Carlo Tasso. A personalized concept-driven recommender system for scientific libraries. *Procedia Computer Science*, 38:84–91, 2014.

[56] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[57] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[58] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. *Recommender systems handbook*, pages 107–144, 2011.

[59] Ricardo Dias and Manuel J Fonseca. Improving music recommendation in session-based collaborative filtering by using temporal context. In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pages 783–788. IEEE, 2013.

[60] Jörg Diederich and Tereza Iofciu. Finding communities of practice from user profiles based on folksonomies. In *Innovative Approaches for Learning and Knowledge Sharing, EC-TEL Workshop Proc*, pages 288–297, 2006.

[61] Michael D Ekstrand, Praveen Kannan, James A Stemper, John T Butler, Joseph A Konstan, and John T Riedl. Automatically building research reading lists. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 159–166. ACM, 2010.

[62] Elena Erosheva, Stephen Fienberg, and John Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5220–5227, 2004.

[63] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. An integrated environment for the development of knowledge-based recommender applications. *International Journal of Electronic Commerce*, 11(2):11–34, 2006.

[64] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In

Proceedings of the fourth ACM conference on Recommender systems, pages 257–260. ACM, 2010.

[65] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[66] Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 593–596. ACM, 2013.

[67] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[68] Nathaniel Good, J Ben Schafer, Joseph A Konstan, Al Borchers, Badrul Sarwar, Jon Herlocker, John Riedl, et al. Combining collaborative filtering with personal agents for better recommendations. In *AAAI/IAAI*, pages 439–446, 1999.

[69] J Grau. Personalized product recommendations: Predicting shoppers' needs, 2009.

[70] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[71] Georg Groh and Christian Ehmig. Recommendations in taste related domains: collaborative filtering vs. social filtering. In *Proceedings of the 2007 international ACM conference on Supporting group work*, pages 127–136. ACM, 2007.

[72] Asela Gunawardana and Guy Shani. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, 10(Dec):2935–2962, 2009.

[73] Yanhui Guo and James BD Joshi. Topic-based personalized recommendation for collaborative tagging system. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, pages 61–66. ACM, 2010.

[74] Uri Hanani, Bracha Shapira, and Peretz Shoval. Information filtering: Overview of issues, research and systems. *User modeling and user-adapted interaction*, 11(3):203–259, 2001.

[75] Jianming He and Wesley W Chu. A social network-based recommender system (snrs). In *Data Mining for Social Network Data*, pages 47–74. Springer, 2010.

[76] Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, and Lee Giles. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 421–430. ACM, 2010.

[77] Jonathan L. Herlocker and Joseph A. Konstan. Content-independent task-focused recommendation. *IEEE Internet Computing*, 5(6):40–47, 2001.

[78] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM, 1999.

[79] Thomas Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 259–266. ACM, 2003.

[80] Maya Hristakeva, Daniel Kershaw, Marco Rossetti, Petr Knoth, Benjamin Pettit, Saúl Vargas, and Kris Jack. Building recommender systems for scholarly information. In *Proceedings of the 1st Workshop on Scholarly Web Mining*, pages 25–32. ACM, 2017.

[81] Shu Huang, Qiankun Zhao, Prasenjit Mitra, and C Lee Giles. Hierarchical location and topic based query expansion. In *AAAI*, pages 1150–1155, 2008.

[82] Zan Huang, Wingyan Chung, Thian-Huat Ong, and Hsinchun Chen. A graph-based recommender system for digital library. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 65–73. ACM, 2002.

[83] K Jack. Mendeley: recommendation systems for academic literature. *Presentation at Technical University of Graz (TUG)*, 2012.

[84] Tamas Jambor and Jun Wang. Goal-driven collaborative filtering–a directional error based approach. In *European Conference on Information Retrieval*, pages 407–419. Springer, 2010.

[85] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[86] Yichen Jiang, Aixia Jia, Yansong Feng, and Dongyan Zhao. Recommending academic papers via users' reading purposes. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 241–244. ACM, 2012.

[87] Pijitra Jomsri, Siripun Sanguansintukul, and Worasit Choochaiwattana. A framework for tag-based research paper recommender system: an ir approach. In *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on*, pages 103–108. IEEE, 2010.

[88] George Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 247–254. ACM, 2001.

[89] Ralf Klamma, Pham Manh Cuong, and Yiwei Cao. You never walk alone: Recommending academic events based on social network analysis. *Complex Sciences*, pages 657–670, 2009.

[90] Ajith Kodakateri Pudhiyaveetil, Susan Gauch, Hiep Luong, and Josh Eno. Conceptual recommender system for citeseerx. In *Proceedings of the third ACM conference on Recommender systems*, pages 241–244. ACM, 2009.

[91] Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[92] 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*, pages 426–434. ACM, 2008.

[93] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender systems handbook*, pages 77–118. Springer, 2015.

[94] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[95] Ralf Krestel and Peter Fankhauser. Tag recommendation using probabilistic topic models. *ECML PKDD Discovery Challenge*, 2009:131, 2009.

[96] Ralf Krestel and Peter Fankhauser. Personalized topic-based tag recommendation. *Neurocomputing*, 76(1):61–70, 2012.

[97] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[98] Ni Lao and William W Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67, 2010.

[99] Jey Han Lau, Nigel Collier, and Timothy Baldwin. On-line trend analysis with topic models:\# twitter trends detection topic model online. In *COLING*, pages 1519–1534, 2012.

[100] Victor Lavrenko and W Bruce Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM, 2001.

[101] Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*, pages 577–584. ACM, 2006.

[102] Nathan N Liu and Qiang Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 83–90. ACM, 2008.

[103] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[104] Linyuan Lü and Weiping Liu. Information filtering via preferential diffusion. *Physical Review E*, 83(6):066119, 2011.

[105] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics Reports*, 519(1):1–49, 2012.

[106] Hao Ma, Irwin King, and Michael R Lyu. Effective missing data prediction for collaborative filtering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 39–46. ACM, 2007.

[107] Judith Masthoff. Group recommender systems: Combining individual models. *Recommender systems handbook*, pages 677–702, 2011.

[108] Jon D Mcauliffe and David M Blei. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128, 2008.

[109] Matthew R McLaughlin and Jonathan L Herlocker. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 329–336. ACM, 2004.

[110] Sean M McNee, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K Lam, Al Mamunur Rashid, Joseph A Konstan, and John Riedl. On the recommending of citations for research papers. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 116–125. ACM, 2002.

[111] Stuart E Middleton, Nigel R Shadbolt, and David C De Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):54–88, 2004.

[112] 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*, pages 195–204. ACM, 2000.

[113] Tomoko Murakami, Koichiro Mori, and Ryohei Orihara. Metrics for evaluating the serendipity of recommendation lists. In *Annual Conference of the Japanese Society for Artificial Intelligence*, pages 40–46. Springer, 2007.

[114] Amine Naak, Hicham Hage, and Esma Aimeur. A multi-criteria collaborative filtering approach for research paper recommendation in papyres. *E-technologies: Innovation in an Open World*, pages 25–39, 2009.

[115] Cristiano Nascimento, Alberto HF Laender, Altigran S da Silva, and Marcos André Gonçalves. A source independent framework for research paper recommendation. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 297–306. ACM, 2011.

[116] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108. Association for Computational Linguistics, 2010.

[117] Javier Parapar, Alejandro Bellogín, Pablo Castells, and Álvaro Barreiro. Relevance-based language modelling for recommender systems. *Information Processing & Management*, 49(4):966–980, 2013.

[118] Dmitry Pavlov, Eren Manavoglu, C Lee Giles, and David M Pennock. Collaborative filtering with maximum entropy. *IEEE Intelligent Systems*, 19(6):40–47, 2004.

[119] Michael Pazzani and Daniel Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, 27(3):313–331, 1997.

[120] Jérome Picault, Myriam Ribiere, David Bonnefoy, and Kevin Mercer. How to get the recommender out of the lab? In *Recommender Systems Handbook*, pages 333–365. Springer, 2011.

[121] Jay M Ponte and W Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM, 1998.

[122] Minghui Qiu, Feida Zhu, and Jing Jiang. It is not just what we say, but how we say them: Lda-based behavior-topic model. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 794–802. SIAM, 2013.

[123] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

[124] Eduardo H Ramirez, Ramon Brena, Davide Magatti, and Fabio Stella. Topic model validation. *Neurocomputing*, 76(1):125–133, 2012.

[125] 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*, pages 175–186. ACM, 1994.

[126] Francesco Ricci. Recommender systems: Models and techniques. In *Encyclopedia of Social Network Analysis and Mining*, pages 1511–1522. Springer, 2014.

[127] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

[128] Stephen E Robertson. The probability ranking principle in ir. *Readings in information retrieval*, pages 281–286, 1997.

[129] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.

[130] Marco Rossetti, Fabio Stella, and Markus Zanker. Analyzing user reviews in tourism with topic models. *Information Technology & Tourism*, 16(1):5–21, 2016.

[131] Marco Rossetti, Saúl Vargas, Davide Magatti, Benjamin Pettit, Daniel Kershaw, Maya Hristakeva, and Kris Jack. Effectively identifying users' research interests for scholarly reference management and discovery. In *Proceedings of the 1st Workshop on Scholarly Web Mining*, pages 17–24. ACM, 2017.

[132] Ian Ruthven and Mounia Lalmas. A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review*, 18(2):95–145, 2003.

[133] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[134] 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*, pages 285–295. ACM, 2001.

[135] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[136] Cyrus Shahabi and Yi-Shin Chen. An adaptive recommendation system without explicit acquisition of user relevance feedback. *Distributed and Parallel Databases*, 14(2):173–192, 2003.

[137] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. *Recommender systems handbook*, pages 257–297, 2011.

[138] Yue Shi, Martha Larson, and Alan Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 269–272. ACM, 2010.

[139] Stefan Siersdorfer and Sergej Sizov. Social recommender systems for web 2.0 folksonomies. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 261–270. ACM, 2009.

[140] Ayush Singhal, Ravindra Kasturi, Vidyashankar Sivakumar, and Jaideep Srivastava. Leveraging web intelligence for finding interesting research datasets. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 321–328. IEEE, 2013.

[141] Yading Song, Simon Dixon, and Marcus Pearce. A survey of music recommendation systems and future perspectives. In *9th International Symposium on Computer Music Modeling and Retrieval*, volume 4, 2012.

[142] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[143] Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

[144] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[145] Kazunari Sugiyama and Min-Yen Kan. Scholarly paper recommendation via user's recent research interests. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 29–38. ACM, 2010.

[146] Kazunari Sugiyama and Min-Yen Kan. Exploiting potential citation papers in scholarly paper recommendation. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pages 153–162. ACM, 2013.

[147] Jiankai Sun, Shuaiqiang Wang, Byron J Gao, and Jun Ma. Learning to rank for hybrid recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2239–2242. ACM, 2012.

[148] Zhoubao Sun, Lixin Han, Wenliang Huang, Xueting Wang, Xiaoqin Zeng, Min Wang, and Hong Yan. Recommender systems based on social networks. *Journal of Systems and Software*, 99:109–119, 2015.

[149] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. In *NIPS*, pages 1385–1392, 2004.

[150] Nava Tintarev and Judith Masthoff. Designing and evaluating explanations for recommender systems. *Recommender Systems Handbook*, pages 479–510, 2011.

[151] Peyman Toreini, Mohamed Amine Chatti, Hendrik Thüs, and Ulrik Schroeder. Interest-based recommendation in academic networks using social network analysis. In *DeLFI*, pages 23–34, 2016.

[152] Roberto Torres, Sean M McNee, Mara Abel, Joseph A Konstan, and John Riedl. Enhancing digital libraries with techlens. In *Digital Libraries, 2004. Proceedings of the 2004 Joint ACM/IEEE Conference on*, pages 228–236. IEEE, 2004.

[153] Chiraz Trabelsi and Gabriella Pasi. Mrra: A new approach for movie rating recommendation. In *International Conference on Flexible Query Answering Systems*, pages 84–95. Springer, 2017.

[154] Shari Trewin. Knowledge-based recommender systems. *Encyclopedia of library and information science*, 69(Supplement 32):180, 2000.

[155] Mark Van Setten, Stanislav Pokraev, and Johan Koolwaaij. Context-aware recommendations in the mobile tourist application compass. In *Adaptive hypermedia and adaptive web-based systems*, pages 515–548. Springer, 2004.

[156] Saúl Vargas, Maya Hristakeva, and Kris Jack. Mendeley: Recommendations for researchers. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 365–365. ACM, 2016.

[157] André Vellino. A comparison between usage-based and citation-based methods for recommending scholarly research articles. *Proceedings of the Association for Information Science and Technology*, 47(1):1–2, 2010.

[158] André Vellino. Usage-based vs. citation-based methods for recommending scholarly research articles. *arXiv preprint arXiv:1303.7149*, 2013.

[159] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM, 2006.

[160] Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning*, pages 1105–1112. ACM, 2009.

[161] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011.

[162] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 783–792. ACM, 2010.

[163] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. A user-item relevance model for log-based collaborative filtering. In *ECIR*, volume 3936, pages 37–48. Springer, 2006.

[164] Jun Wang, Stephen Robertson, Arjen P de Vries, and Marcel JT Reinders. Probabilistic relevance ranking for collaborative filtering. *Information Retrieval*, 11(6):477–497, 2008.

[165] Shuaiqiang Wang, Jiankai Sun, Byron J Gao, and Jun Ma. Vsrank: A novel framework for ranking-based collaborative filtering. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):51, 2014.

[166] Xing Wei and W Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2006.

[167] Markus Weimer, Alexandros Karatzoglou, Quoc V Le, and Alex J Smola. Cofi rank-maximum margin matrix factorization for collaborative ranking. In *Advances in neural information processing systems*, pages 1593–1600, 2008.

[168] Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu, and Zheng Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 114–121. ACM, 2005.

[169] Ronald R Yager. Fuzzy logic methods in recommender systems. *Fuzzy Sets and Systems*, 136(2):133–149, 2003.

[170] Chenxing Yang, Baogang Wei, Jiangqin Wu, Yin Zhang, and Liang Zhang. Cares: a ranking-oriented cadal recommender system. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 203–212. ACM, 2009.

[171] Zaihan Yang and Brian D Davison. Venue recommendation: Submitting your paper with style. In *Machine learning and applications (ICMLA), 2012 11th international conference on*, volume 1, pages 681–686. IEEE, 2012.

[172] Xing Yi and James Allan. A comparative study of utilizing topic models for information retrieval. In *European Conference on Information Retrieval*, pages 29–41. Springer, 2009.

[173] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. *HotCloud*, 10(10-10):95, 2010.

[174] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342. ACM, 2001.

[175] Chenyi Zhang, Xueyi Zhao, Ke Wang, and Jianling Sun. Content+ attributes: A latent factor model for recommending scientific papers in heterogeneous academic networks. In *European Conference on Information Retrieval*, pages 39–50. Springer, 2014.

[176] Mi Zhang. Enhancing diversity in top-n recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pages 397–400. ACM, 2009.

[177] Xing Zhou, Lixin Ding, Zhaokui Li, and Runze Wan. Collaborator recommendation in heterogeneous bibliographic networks using random walks. *Information Retrieval Journal*, 20(4):317–337, 2017.