5-2017

Peeking into the other half of the glass : handling polarization in recommender systems.

Mahsa Badami
University of Louisville

PEEKING INTO THE OTHER HALF OF THE GLASS:
HANDLING POLARIZATION IN RECOMMENDER SYSTEMS

By

Mahsa Badami
M.Sc., Computer Science, University of Louisville, Louisville, 2017
M.Sc., Computer Engineering, Artificial Intelligence, Shiraz University of, Iran, 2012

A Dissertation
Submitted to the Faculty of the
J.B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy in Computer Science and Engineering

Department of Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky

May 2017

PEEKING INTO THE OTHER HALF OF THE GLASS:
HANDLING POLARIZATION IN RECOMMENDER SYSTEMS

By

Mahsa Badami
M.Sc., Computer Science, University of Louisville, Louisville, 2017
M.Sc., Computer Engineering, Artificial Intelligence, Shiraz University of, Iran, 2012

A Dissertation Approved On

April 26, 2017

by the following Dissertation Committee:

Olfa Nasraoui, Ph.D., Dissertation Director

Hichem Frigui, Ph.D.

Roman Yampolskiy, Ph.D.

Nihat Altiparmak, Ph.D.

Scott Sanders, Ph.D.

.

To those who fight for truth and justice ...

ACKNOWLEDGEMENTS

There are so many people to thank for helping me during my Ph.D journey. Foremost, I would like to express my deepest gratitude to my advisor Dr. Olfa Nasraoui for her patience, guidance, encouragement, and support. I am fully indebted to Dr. Nasraoui for always believing in me and I am honored for having her as my role model.

I would also like to extend my thanks to Dr Hichem Frigui, Dr. Roman Yampolskiy, Dr. Nihat Altiparmak and Dr. Scott Sanders for serving on my dissertation committee and providing valuable feedback. A very special gratitude goes to the Computer Science and Computer Engineering Department, Speed School of Engineering and School of Interdisciplinary and Graduate Studies, where the faculty and staff made my stay at the University of Louisville one I will always remember. I am also grateful to the following university staff: Dr. Micheal Harris, Dr. Gail DePuy, Dr. Beth Boehm, Dr. Paul DeMarco and Gabrielle Billings for their unfailing support and assistance.

Nevertheless, I wish to express my sincere thanks to my good friends, Elaheh Arabmakki and Faezeh Tafazzoli for supporting me through though times and always believing in me. I would also like to thank my fellow doctoral students for their feedback, cooperation and friendship.

Last but not the least, I would like to thank my family: my parents, Mr. Badami and Mrs. Soheilian, my brother, Ehsan, and sister in-law, Rana, who have provided me thorough moral and emotional support in my life. I wouldn't be this strong without them as my inspiration.

This work was partially supported by National Science Foundation: NSF INSPIRE (IIS)- Grant #1549981.

ABSTRACT

PEEKING INTO THE OTHER HALF OF THE GLASS:

HANDLING POLARIZATION IN RECOMMENDER SYSTEMS

Mahsa Badami

April 26, 2017

This dissertation is about filtering and discovering information online while using recommender systems. In the first part of our research, we study the phenomenon of polarization and its impact on filtering and discovering information. Polarization is a social phenomenon, with serious consequences, in real-life, particularly on social media. Thus it is important to understand how machine learning algorithms, especially recommender systems, behave in polarized environments. We study polarization within the context of the users' interactions with a space of items and how this affects recommender systems. We first formalize the concept of polarization based on item ratings and then relate it to the item reviews, when available. We then propose a domain independent data science pipeline to automatically detect polarization using the ratings rather than the properties, typically used to detect polarization, such as item's content or social network topology. We perform an extensive comparison of polarization measures on several benchmark data sets and show that our polarization detection framework can detect different degrees of polarization and outperforms existing measures in capturing an intuitive notion of polarization.

We also investigate and uncover certain peculiar patterns that are characteristic of environments where polarization emerges: A machine learning algorithm finds it easier to learn discriminating models in polarized environments: The models will quickly learn to

keep each user in the safety of their preferred viewpoint, essentially, giving rise to filter bubbles and making them easier to *learn*.

After quantifying the extent of polarization in current recommender system benchmark data, we propose new counter-polarization approaches for existing collaborative filtering recommender systems, focusing particularly on the state of the art models based on Matrix Factorization. Our work represents an essential step toward the new research area concerned with quantifying, detecting and counteracting polarization in human-generated data and machine learning algorithms.We also make a theoretical analysis of how polarization affects learning latent factor models, and how counter-polarization affects these models.

In the second part of our dissertation, we investigate the problem of discovering related information by recommendation of tags on social media micro-blogging platforms. Real-time micro-blogging services such as Twitter have recently witnessed exponential growth, with millions of active web users who generate billions of micro-posts to share information, opinions and personal viewpoints, daily. However, these posts are inherently noisy and unstructured because they could be in any format, hence making them difficult to organize for the purpose of retrieval of relevant information. One way to solve this problem is using hashtags, which are quickly becoming the standard approach for annotation of various information on social media, such that varied posts about the same or related topic are annotated with the same hashtag. However hashtags are not used in a consistent manner and most importantly, are completely optional to use. This makes them unreliable as the sole mechanism for searching for relevant information. We investigate mechanisms for consolidating the hashtag space using recommender systems. Our methods are general enough that they can be used for hashtag annotation in various social media services such as twitter, as well as for general item recommendations on systems that rely on implicit user interest data such as e-learning and news sites, or explicit user ratings, such as e-commerce and online entertainment sites. To conclude, we propose a methodology to extract stories based on two types of hashtag co-occurrence graphs. Our research in hashtag recommendation was able to exploit the textual content that is available as part of user messages or posts, and thus resulted in hybrid recommendation strategies. Using content within this context can bridge polarization boundaries. However, when content is not available, is missing, or is unreliable, as in the case of platforms that are rich in multimedia and multilingual posts, the content option becomes less powerful and pure collaborative filtering regains its important role, along with the challenges of polarization.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

ALGORITHM Page

CHAPTER 1

INTRODUCTION

Recommender systems (RS) appeared in early work in cognitive science, information retrieval and consumer behavior in marketing [1–3]. However, they became an important research area in the mid-1990s when the central goal of RS was to estimate the rating for the items to be selected and shown to a user who has not seen those items before [4, 5]. Using this estimation, the system is able to decide whether to recommend the items to the user or not; intuitively, a RS chooses the items with the highest rating [6]. Over the years, recommender systems witnessed fast development and wider applications [4,5], where they are used to learn user“s” preferences, likes and dislikes.

Over the last two decades, a wide variety of recommender systems (RSs) has been developed across several domains, where researchers have focused mainly on the performance of the RS algorithm, primarily meaning providing increasingly accurate rating predictions [5] [7, 8]. This is a reasonable goal since more accurate prediction will lead to a better user experience, in general. However, there is still a long way to go in terms of *all* aspects of the user's interaction experience, including accuracy of the algorithm, subjective aspects such as system satisfaction and user satisfaction [8–11]. Nowadays, users rely on recommender systems to overcome the information overload problem when navigating through a vast amount of options and content, on a variety of platforms ranging from traditional browsers to mobile applications. This reliance on algorithmic guidance is often not explicitly or intentionally chosen by the user, since many social media services (e.g.facebook) automatically filter and sort the order of items on a user's home feed, without an explicit request from the user to do so.

Figure 1.1 shows a closed feedback loop between a user and a Machine Learning

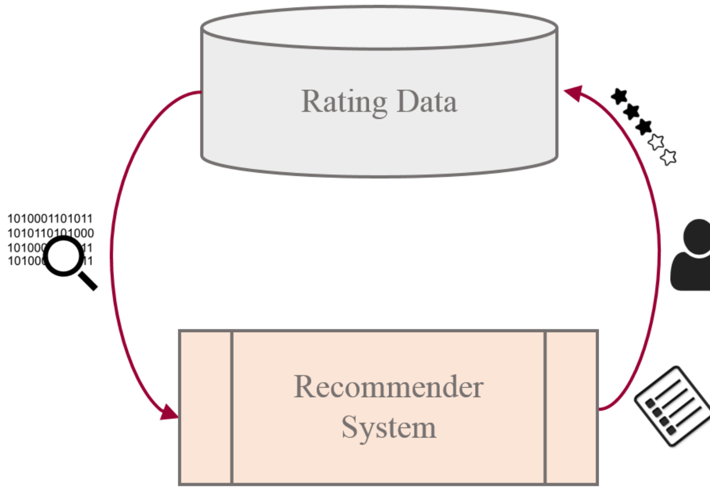


Figure 1.1: Recommender system feedback loop

algorithm, in our case a RS. One intuitive consequence of increased reliance on algorithmic guidance, combined with the user's consumption behavior, is the possibility of establishing a strong loop, called a "positive feedback loop" [12, 13]. This loop is due to the *persuasive* qualities of the recommender systems, where users mainly follow a recommender's advice [14] and after a while agree with it, unknowingly [15]. According to [16], this may be due to the human tendency to take the default setting rather than going through the trouble of developing one's own unique taste. This may be related to the pheromone, called *temporal discounting* [17], where we tend to choose guilty pleasures that provide instant satisfaction over a long value (e.g. watching a spectacular action movie over an educational experience). As a result, users will make themselves better "fit in" with the (recommender) system and simply consume whatever the system serves them, because it is "easy" to like or consume [18, 19]. Another reason behind the positive feedback loop is that common RSs usually employ limited historical data to recommend a product. These products are mostly similar to what users have already purchased or liked in the past, which causes the loop to narrow the diversity of the recommended items [20]. As advocated by various philosophers and researchers, the positive feedback loop then affects customer's choices by limiting her/his exposure to alternative views which disturbs the balance of her/his options [13, 21–23]. This phenomenon is called a "filter bubble [13], and is defined as: "**the invisible and personal**

universes of information that might trap users into a relevance paradox confining them to isolated information neighborhoods and restricting them from seeing or exploring the vast array of other possibilities" [13, 24].

Despite the numerous benefits of personalized recommender systems, narrow-minded over-specialization, resulting from positive feedback loops, can exacerbate "Filter bubbles" by not expanding, or even restricting, the users' exposure to more diverse and non-obvious options [25]. This in turn can lead to getting locked in by the RS where it creates "self-fulfilling identities": **Your identity shapes your recommendations, and your recommendations then shape what you believe, and what you care about** [13]. In addition, recommender systems suffer from the "Rich-get-richer effect where popular items draw more attention and hence it creates a bias toward more seen products which may hide better consumer-product matches from the users [26]. This is often inconsistent with users' preferences, needs, social welfare and business goals [27, 28].

Using *over-specialized* recommender systems can hence make the population more similar to each other over time, which in turn, leads to concentration biases, called "gravitational force", that lead to grouping individuals together and further shrinking filter bubbles [21]. *Narrow-minded* personalization techniques can have additional detrimental effects, such as deconstructing non-prevailing views, opinions, and behaviors [27]. Moreover, daily interaction between these systems and customers may even lead them to adapting more extreme attitudes [29] and mis-perceiving facts about events [30].

1.1 Recommender Systems and Polarization

The growing popularity of online services and social networks and the trend to integrate Recommender Systems (RS) within most e-commerce applications and social media platforms to help filter data to the users, has led to a dynamic interplay between the information that users can discover and the algorithms that filter such information. This has given rise to several side effects, such as algorithmic biases [31], filter bubbles [32] and polarization [33], and human-algorithm iterated bias [34]. Algorithmic bias has also been explored by Baeza-Yates [35] mainly in the information retrieval context. Polarization occurs when information happens to be related to controversial issues where a population is divided in groups with opposite opinions, fewer individuals with moderate opinions, and there is no clear majority in "ardent supporters" and "motivated adversaries" [33, 36, 37].

Polarization around controversial issues have arguably affected recommender systems (and vice-versa) due to the ill-fated consequences of this phenomenon [38, 39]. An effective and efficient recommender system needs to be able to apply the most suitable recommendation method even in the presence of a set of polarized items. Trust based recommender systems try to offer a solution to this problem by defining a trust network for each user [40]. These types of recommender systems leverage the user's trust network's opinion on an item to finally decide whether to recommend an item. However, when such issues emerge on social media, we often observe the creation of "echo chambers" or "Filter Bubbles", where there is greater interaction between like-minded people who reinforce each other's opinion [38]. These individuals do not get exposed to the views of the opposing side, and this in turn results in polarization [31]. Allowing users to discover different viewpoints could allow them to develop unique tastes and diverse perspectives that are not limited by filter bubbles [18]. In addition, it is important to fully understand the population before recommending any items. Very few recent studies investigated the effect of datasets on the recommendation process. For instance, the experiments in [41] have indicated that population characteristics have an influence on the behavior of the algorithm, with respect to different measures such as catalog coverage (What recommenders recommend). Hence, considering characteristics such as polarization would go unnoticed if accuracy measures were the only criteria considered for assessing the quality of recommendation algorithms, as is typically the case. Studying the emergence of polarization in human-algorithm interaction is related to studying the evolution and impact of bias in user-generated data that is consumed by recommender algorithms, which in turn makes these algorithms' predictions biased, as pointed to by Nasraoui and Shafto [42] who proposed new models, inspired by language evolution, for the bias resulting from iterated interactions between users and recommendation algorithms.

1.2 Objectives

The main objective of this research is to survey and understand the interplay between polarization and recommendations, and to contribute to building a new paradigm of recommendation systems that are aware of polarization, while still providing high quality personalized and counter-polarized recommendations, such that the users can control their degree of counter-polarization. Polarization awareness consists of being able to detect polarization in the first place, and to expand the domain of options recommended to the user in a second place. Expanding the domain of recommended options promises to expand the discovery potential for humans, and thus reduce their filter bubble, as they interact with filtering and recommendation algorithms. Whether humans prefer to discover more or less is beyond the scope of this research. However, our counter-polarization paradigm does provide a user-controlled discovery parameter to give the user the freedom to tune their bubbles as they wish. The second component of this research targets the domain of micro-blogging social networks for recommending hashtags.

1.3 Contributions

We summarize our major contributions in the following list:

- Multi-perspective survey of polarization.
- Automated polarization detection methodology.
- Counter-polarization approach to recommendation.
- Theoretical analysis of polarization's impact on learning RS models and polarization aware models' impact on polarization
- Hashtag recommendation and story discovery strategies in social media micro-blogging services

Polarization is an important phenomenon, with serious consequences, in real-life, particularly on social media. Thus it is important to understand how machine learning algorithms, especially recommender systems, behave in a polarized environment; and to this end it is important to quantify polarization in existing and future data sets. Our contribution is a significant and essential step toward quantifying and detecting polarization in ongoing ratings on generic platforms and in benchmark data sets. Hence it supports future research in the emerging topic of designing and studying the behavior of recommender systems in polarized environments. In the following, we summarize each one of the contributions.

Studying emergent bias and polarization phenomena in machine learning is by no means an easy task. In fact, recent work [43] has put in question the prevalent evaluation strategies and benchmark data sets, traditionally used in recommender system research, because benchmark data sets have, heretofore, no way of capturing or measuring the bias and polarization resulting from algorithmic guidance. This difficulty in capturing and testing the emergence of polarization in user-algorithm interaction, imposes a significant challenge to research in this emerging field; a challenge that we will address in this dissertation, first by presenting an original survey of polarization that draws on literature from multiple perspectives and disciplines; then proposing a new polarization detection methodology to measure polarization in generic, rating-based data sets; and finally formulating theoretically-grounded scenarios for polarization which will allow a simulation-based analysis of the emergence of polarization, as well as designing new counter-polarization strategies for recommender systems.

1.3.1 Multi-perspective Survey

Our first contribution is a spanning, multi-pronged survey, that draws from diverse disciplines to review various literature that relates to polarization from near and far, and with varying levels of formal treatments. Our survey showed that the field is rather not unified in how polarization is defined and what is done after recognizing it. Very little work has gone beyond the simple task of defining or at best trying to measure polarization.

Based on our survey, we found that polarization has been studied from three main perspectives (and occasionally hybrids thereof), leading us to define the following taxonomy:

1. social polarization: how people congregate with one another,

2. written polarization: how people write about topics,

3. rated and recommended polarization: how people behave, consume and express their preferences.

1.3.2 Detecting Polarization

Our second contribution is motivated by the fundamental, yet challenging task of detecting polarized/controversial items across diverse platforms, rather than studying the evolution of polarization. We thus design and develop an algorithm to identify polarization/controversy regarding items in any domain (e.g. political, economical, or cultural), and without prior domain-specific knowledge about the items. Having an algorithm with these properties allows us to deploy a system in-the-wild, and is valuable for building real world applications, such as recommender systems. Contrary to other models where polarization is based on either a social graph or item content, we propose a model that works with ratings in any domain and on a large scale. Ratings are more intuitive to work with since they directly capture the distribution of user opinions. In addition, an item itself is not polarized unless there are users with opposing opinion on the item, so the content may not be very reliable for polarization detection. In the absence of polarization, the distribution of opinions is either J-shaped or bell shaped. However, as polarization emerges, the resulting distribution shifts to a U-shaped distribution with two peaks emerging around the two dominant and confronted opinions at the extreme sides of the rating scale [33,44]. Different examples of such distributions are shown in figure 1.2.

Our new approach to quantify polarization is based on three stages: (i) building an item ratings' histogram from user-item rating data; (ii) extracting a set of features from the histograms; (iii) training a polarization classifier based on a sample of annotated cases; and (iv) measuring the item-level polarization score. To verify our approach, we apply a multi-pooling text classifier, which combines both labeled comments/reviews and a lexical (words) library, to understand the item's polarization and its possible relation with the item's reviews.

We design a systematic lightweight pipeline, with simple yet effective features, derived mainly from the item ratings, and hence they can be used in any domain, where explicit or implicit user interest or feedback can be captured. Due to the simplicity, generality, and speed of this methodology, it can be used with any recommender system, and on diverse platforms, such as news-reading and public-debate scenarios. We use human intuition to capture polarization, in other words, "what" would cause someone, trying to gauge the overall opinion about an item, to be undecided about the overall opinions on this item by looking at the ratings distribution.

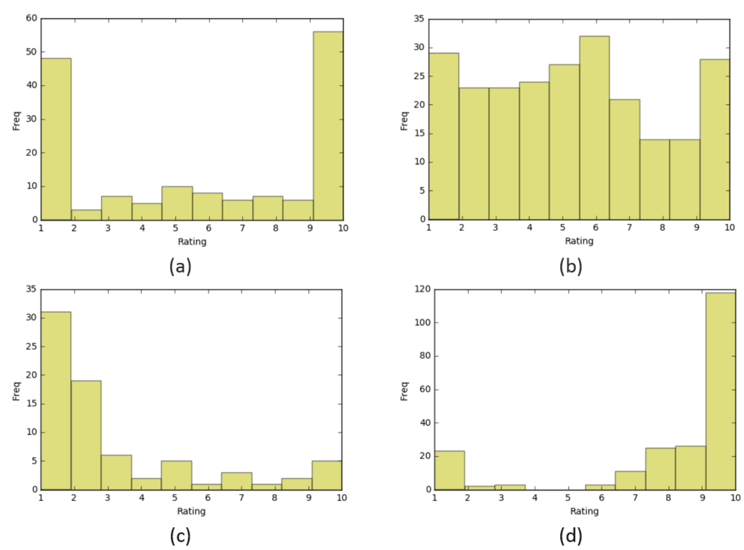


Figure 1.2: Different rating distributions from IMDb.com.

1.3.3 Polarization-Aware Recommender System

Aiming toward alleviating the important problems of over-specialization and concentration bias, especially in a polarized environment, and enhancing the usefulness of collaborative filtering RS, we propose a new approach to generating recommendation lists based on a modified Non-Negative Matrix Factorization approach. In particular, we estimate a user's interest on an item, while considering both (i) the items' polarization scores, and (ii) the user's expression of their willingness to discovery. Our proposed model aims to achieve a trade-off between accurate personalized recommendations and expanding the space of items that can be discovered, hence escaping a filter bubble.

In order to give the users a choice to see more items, we believe that a recommender system should have a systematic mechanism that enables users to discover novel items caused by the user's continuous engagement with the system. This is different from recommending a random item by trial and error, to increase arbitrary discovery and serendipity.

We propose two models for combating over-specialization in polarized environments. These two models are used in the positive feedback loop and are independent of the item domain and the type of recommender system. The first model works by altering the input ratings based on the automatically detected polarization and the user's tolerance for discovery, while the second model consists of a new recommendation model.

We also make a theoretical analysis of how polarization affects learning latent factor models, and how counter-polarization affects these models.

It is important to note that recommender systems that improve diversity and serendipity are not the same as polarization aware recommender systems. This is because the former generally require diversity in the actual description or nature of items, which in turn requires content data. Our research primarily focuses on items that can cross polarization boundaries, where polarization is based on how users interact with the items (ratings) and not their content.

1.3.4 Empowering Users of Recommender Systems

We investigated and uncovered certain peculiar patterns that are characteristic of environments where polarization emerges, for instance by monitoring traces of the objective function that is minimized in matrix factorization. We have found that environments with different polarization degrees engender different patterns. The ability to recognize such patterns, that arise during incremental optimization, can help quickly detect and quantify the evolution of polarization without the tedious analysis of rating data distributions. This objective function optimization pattern monitoring-based approach open a new direction of research in studying and handling polarization, not only in recommender systems, but also in other machine learning algorithms that also result in filtering information from humans.Our proposed counter-polarization methodology succeeds to cover items from the opposite view after a few iterations and can broaden the viewpoint spectrum even faster if the user is more interested in discovering items from different viewpoints. Endowing humans with the ability to interactively and adaptively control the breadth of viewpoints offered by an intelligent polarization-aware recommendation algorithm is an important capability. As a result, users who are nowadays increasingly and unconsciously entrapped in algorithmic filters over which they have had no control, can become empowered to break free from their algorithmic chains. To our knowledge, this feature which allows humans to remain in or regain control of algorithm-induced filter bubble traps, has heretofore not been allowed or engineered in existing information filtering systems, whether on social media, e-commerce or e-learning recommender systems.

1.3.5 Hashtag Recommendation

In the second part of our dissertation, we investigate the problem of discovering related information by recommendation of tags on social media micro-blogging platforms. Real-time micro-blogging services such as Twitter have recently witnessed exponential growth, with millions of active web users who generate billions of micro-posts to share information, opinions and personal viewpoints, daily. However, these posts are inherently noisy and unstructured because they could be in any format, hence making them difficult to organize for the purpose of retrieval of relevant information. One way to solve this problem is using hashtags, which are quickly becoming the standard approach for annotation of various information on social media, such that varied posts about the same or related topic are annotated with the same hashtag. However hashtags are not used in a consistent manner and most importantly, are completely optional to use. This makes them unreliable as the sole mechanism for searching for relevant information. We investigate mechanisms for consolidating the hashtag space using recommender systems. Our methods are general enough that they can be used for hashtag annotation in various social media services such as twitter, as well as for general item recommendations on systems that rely on implicit user interest data such as e-learning and news sites, or explicit user ratings, such as e-commerce and online entertainment sites. To conclude, we propose a methodology to extract stories based on two types of hashtag co-occurrence graphs. Our research in hashtag recommendation was able to exploit the textual content that is available as part of user messages or posts, and thus resulted in hybrid recommendation strategies. Using content within this context can bridge polarization boundaries. However, when content is not available, is missing, or is unreliable, as in the case of platforms that are rich in multimedia and multilingual posts, the content option becomes less powerful and pure collaborative filtering regains its important role, along with the challenges of polarization.

1.4 Dissertation Outline

The remainder of this dissertation is organized as follows. Chapter 2 provides a literature review of recommender systems and sentiment analysis. Chapter 3 presents a new survey and taxonomy of polarization. Chapter 4 introduces our proposed methods on handling polarization in recommender systems which is followed by the experiments in Chapter 5. We present our proposed Hashtag recommendation in Chapter 6 and Chapter 7 presents our experiments. Finally, we make our conclusions in chapter 8.

CHAPTER 2

BACKGROUND

In this chapter, we review research that is related to our work. As already mentioned, our research focuses on recommender systems. Hence, we will start with reviewing recommender systems methods, along with their effectiveness and capabilities. We describe polarization/controversy and outline literature related to this phoneme. Our research also builds on a existing method, Matrix Factorization (MF), which we will review in this chapter as well.

We will complete our review with an overview of Hashtag annotation work as well as polarization on social media.

2.1 Recommender Systems

Traditionally, recommender systems (RS) produce a list of recommendation of products for users in e-commerce e.g. Amazon Netflix, Pandora. A RS helps users to find the items which are more suited to their interest and they typically do this by filtering the vast amounts of information to suggest new items (e.g. movies, songs and books) to users. Recommender system can be built with many approaches which are reviewed below [45]:

2.1.1 Random prediction algorithm

This algorithm randomly chooses items from available items and then recommends the selected ones to the user. Basically, Random prediction algorithm is based on luck, which leads to recommending unrelated items. In another word, if there are more items in the set, there is less chance to select better items [5].

2.1.2 Frequent sequences

This algorithm is based on the frequency of rated items by the user. It uses frequent patterns to recommend items to the user. However, it requires a minimum purchase or viewing frequency from the users to be able to make the recommendation [45].

2.1.3 Content based algorithms

Content-based recommender systems exploit the items or users past history in order to create item or user profiles [45]. Using this profile, it can recommend the items that best match the items that the user has highly rated in the past. It can also use this history to find similar users and then recommend their interest to the current user. This similarity can be based on either items or users and users profiles [46]. A content-based user profile can indicate the user's preferences in terms of the keywords or attributes of their preferred items. These in turn can help the algorithm formulate a relevant query in order to find popular items using similar keywords. Therefore, content-based algorithms utilize item descriptions in order to identify items that are of particular interest to the user [6].

2.1.4 Collaborative Filtering algorithms (CF)

Most recommender systems have evolved toward collaborative filtering algorithms [5, 6, 45], which either need the users to express their interests, or infer the users interests based on their behavior. A CF algorithm then finds a group of users with similar interest and recommends new items that are part of these similar users interests to the active user. In other words, CF algorithms make prediction about the users interests by filtering information in order to collect preferences or taste information from many (so called collaborating) users [45].

Basically, there are two assumptions behind CF algorithms: First, users who had similar tastes in the past will have similar tastes in the future; second, user's preferences remain stable and consistent overtime [6]. In other words, if person A has the same opinion as person B on an issue, A is more likely to have B's opinion on a different issue x than to have the opinion on x of a person chosen randomly [47].

CF recommender systems filter available information in order to provide an intelligent mechanism for customers to be able to find items in which they are more interested. CF algorithms have several advantages over other, particularly content-based, approaches: They are able to filter any type of content e.g. text, artwork, music, mutual funds [47] without requiring that the items be describable with keywords or attributes. At the same time, they are able to model complex concepts such as taste and quality, which can be hard to express explicitly. For example, a collaborative filtering recommendation system can be used for prediction of television tastes. It means that a list is provided for the users based on their likes and dislikes. The interesting part of the CF methodology is that it provides recommendation for only the current (also called active) user; however it uses information gathered from many other users. Thus, the accuracy of Collaborative Filtering algorithms depends on the availability of information from many users and more data helps the algorithm to learn the preferences better.

Generally, Collaborative filtering techniques are divided into two main categories [48]: memory-based (user-based or item-based neighborhood methods), and model-based (the most prominent being latent factor models):

2.1.4.1 Memory-based Collaborative Filtering technique (neighborhood-based)

Memory-based collaborative filtering [5, 45] uses the entire collection of user-item rating data in order to make a prediction. It is also considered a heuristic method since it utilizes statistical techniques to find a group of users with similar history. Then, the preference of this neighborhood is combined in order to obtain the recommendations. One very well-known algorithm which is used with memory-based CF is the nearest neighbor algorithm [49].

In order to calculate the similarity between user ratings, purchase history, or interests, there are several measures that can be used such as Cosine similarity and Pearson correlation [4]. In addition, there are two different approaches for computing the similarity between users [46]:

a. Recommendation based on explicit information: in this case users are required to express their interests on items explicitly. These interests may have several forms such as rating, like/dislike buttons, reviews, etc.

b. Recommendation based on implicit information: in this case, the algorithm extracts the users preferences from the available information indirectly, such as whether the user spent more time looking at an item or repeatedly came back to check the item. Remember that, in this case, the explicit information may be also available.

CF algorithms can perform very well, but they tend to suffer from sparsity and scalability problems [45]. As mentioned before, a CF recommender systems accuracy depends on the availability of extensive amounts of information. However, most users only rate a small proportion of the items available, leading to sparse data sets. In addition, the latency (waiting time) of each recommendation will increase for large datasets [6], making scalability a difficult challenge.

2.1.4.2 Model-based Collaborative filtering

Model-based CF recommender systems use machine learning techniques in order to estimate a global model which fits the data. Then, the model is used to make predictions for either users or items. However, machine learning approaches require enough training data points in order to be able to have an accurate prediction. Providing enough training data points is a challenge for some application.

Different Model-based approaches have been employed in recommender systems, such as designing a CF system for estimating all missing ratings using Support Vector Machine (SVM) and heuristics [50]. An approach based on neural networks considers both user-based and item-based techniques [51]. [52] developed a probabilistic model for recommendation that uses a Bayesian network model and considers only integer ratings. Another efficient model-based CF algorithm is matrix factorization [53]. This algorithm, basically represent users and items in a low dimensional latent factors space and try to explain the preferences by characterizing both items and users on say, 20 to 100 factors inferred from the ratings patterns. Recent studies showed that the latent factor CF is one of the most successful techniques, especially after it was used for the Netflix prize, where it provided the best results [54]. Matrix factorization is reviewed in the next section.

2.1.4.2.1 Matrix Factorization Recently, due to the efficiency of dealing with large datasets, several low-dimensional matrix approximation methods have been proposed for collaborative filtering [55]. All of these methods focus on fitting the user-item rating matrix using low-rank approximations, and employ the matrix to make further predictions. The Low-rank matrix factorization methods are very efficient regarding the training since they assume that only a small number of factors influences the preferences in the user-item rating matrix, and that a users preference vector is determined by how each factor applies to that user. Low-rank matrix approximations, based on minimizing the sum of squared errors, can be easily solved using Singular Value Decomposition (SVD) [56].

An Expectation Maximization (EM) algorithm for solving weighted low-rank approximation was proposed by [57]. Salakhutdinov et al. presented a probabilistic linear model with Gaussian observation noise in [58]. In [59], the Gaussian-Wishart priors are placed on the user and item hyper-parameters.

Recently, latent factor CF algorithms have attracted a lot of attention due to their efficiency in dealing with large datasets [59–61]. The latent factor approach employs low-dimensional matrix approximation methods which yield better results in CF algorithms. In its basic form of low-rank matrix factorization methods, although they consider the user-item rating matrix, they assume that only a small number of factors influence the user/items interests. In other words, this method captures both items and user characteristics and represents them by vectors of factors inferred from available patterns. High correspondence between item and user factors leads to a recommendation [62]. The Matrix factorization CF recommender system is capable of combining good scalability with predictive accuracy in addition to its flexibility for modeling various real-life situations [62]. Another advantage of MF recommender systems is its incorporation of additional information. It is a very useful characteristic when there is no explicit feedback, since it infers user interests using implicit information including purchase history, browsing history, search patterns, or even mouse movements [54].

Matrix factorization models, map both users and items, to a joint latent factor space of dimensionality f, such that user-item interactions are modeled as inner products in that space [55].

2.1.4.2.2 Non-Negative Matrix Factorization (NMF) Mathematically speaking, the general problem of Non negative matrix factorization is to decompose a non-negative data matrix R with size $m \times n$ into two matrix factors P and Q with size of $m \times k$ and $k \times n$ respectively, such that:

$$R_{m \times n} = P_{m \times k} . Q_{k \times n} \tag{2.1}$$

where $k \leq \min(n\,m)$ is a positive integer representing the rank of matrices P and Q.

The NMF algorithm finds a basis that is able to represent high-dimensional data, using a compact set of vectors, effectively inducing a new low-dimensional representation (referred to as the latent space) of the original data. The vectors of the basis represent clusters of data objects. Since the basis generated by other algorithms, such as SVD, cannot be generally interpreted as cluster prototypes, NMF can be a good replacement for other dimensionality reduction or transformation techniques. This is because the vectors of the basis generated by NMF are not necessarily mutually orthogonal. Moreover, the documents projections on the basis vectors have only non-negative values corresponding to the membership values of a given document with respect to each cluster. These two properties of NMF make it superior to SVD when it comes to document clustering [63]. The third advantage of NMF is its computational efficiency, since the matrix multiplication based update equations [64] can be efficiently implemented for large and sparse. This is a critical point for mining web-scale datasets. Finally, the basis vectors can be used to support sophisticated tasks, in addition to dimensionality reduction or clustering, such as the task of automatically computing associated textual annotations for new unannotated images by matching, in the reduced dimensional latent space, the new images with annotated images and then picking the annotation words that appear the most frequently [65].

There are different ways to estimate P and Q [66], the most common one minimizes the Frobenius norm of the errors between the approximation P, Q and the data matrix R. The objective function is defined as follows:

$$J = \sum_{i,j \in R} (r_{ij} - p_i q_j^T)^2 + \frac{\beta}{2}(||p_i||^2 + ||q_j||^2) + \frac{\lambda}{2}(p_i - q_j)^2 W_{ij} \tag{2.2}$$

where R is the set of user-item pairs for which the ratings are available, $\frac{1}{2}(||p_i||^2 + ||q_j||^2)$ is an $L2$ regularization term weighted by the coefficient β,

where the constraint is $P, Q \leq 0$. This is an optimization problem where the cost function needs to be minimized. In order to find the stationary point of the optimization problem (P and Q), a gradient-descent technique is applied. Two approaches have been typically used: a) stochastic gradient descent and b) alternating least squares (ALS). Alternating least squares exploits the fact that although the NMF cost function is not simultaneously convex, it is convex in either one of the low rank matrices alone [67]. Hence, by fixing one matrix in the current iteration, it can use the Least Squares technique to derive the other matrix. Then it alternates this process until convergence. Although there is no guarantee to find the global optimum, a local optimal point should be a stationary point due to the optimization principle [68].

2.1.4.2.3 Hybrid Collaborative Filtering Techniques Hybrid CF methods combine collaborative filtering with other recommendation techniques, such as content-based RS. This combination helps the recommender system avoid the limitations of both content-based and collaborative filtering approaches [69, 70]. For example, one can apply collaborative and content-based separately and then merge their predictions. A hybrid model is typically more general and is strengthened with the characteristics of both CF and content-based recommender systems [6, 71].

2.2 A Special Recommendation Domain: Hashtag recommendation

Hashtag recommendation, like other item recommendations, can be computed in one of two ways - collaborative filtering or content-based filtering. The collaborative filtering approach builds a model based on finding similar users, i.e. with similar behavior, as well as the users past behavior (e.g. the past preference ratings, page views, or purchases by either the active user or by other similar users) [6, 7]. The content-based approach utilizes the items with the most similar characteristics in order to recommend additional items with the same properties [69, 72]. Moreover, there are hybrid approaches that consider both similar users and similar items to predict the users interests [73].

Due to the usefulness of tag recommendation, many social media services have been employing user driven tagging systems, including image tagging (e.g., on Flickr, Instagram), video tagging (e.g., on YouTube), and web page tagging (e.g., using Del.ico.us), text tagging (e.g., on Twitter). There are many research efforts from different perspectives which considered online resources, e.g. images, videos, texts, links, etc. [74–78].

Here, we are mostly interested in recommending tags for real-time micro-blogging posts which are very challenging and sparse due to the twitter application specific limitations, such as the maximum number of characters (140) in a post and the inherent ambiguity in text.

There has been considerable work in recommending tags, which have focused on recommending general hashtags [79] while other work utilized user history, neighbors and other personal documents to predict personalized hashtags [80]. Khabiri [81] recently presented a content-based hashtag recommendation method based on the content of the tweets. The recommendation system measures the relevance of words and hashtags based on a co-occurrence graph. This approach can be used for recommending general hashtags. In another direction, Yang recently proposed a new approach which considers user-level hashtags in a way to see whether or not a hashtag would be used by a user [82]. The approach considers some basic properties of users and hashtags (e.g., the number of unique hashtags used by user u, and the number of tweets containing hashtag h as well as the relevance between users and hashtags measured by their cosine similarity. Finally they utilize a SVM classifier for prediction. This approach considers the same set of hashtags for prediction, since it discounts tweet content and other information about the tweets.

In [83], the authors proposed a variation of TF-IDF that considers hashtag relevancy and data sparseness. The proposed method relies on a hashtag frequency map which considers the frequency with which a hashtag appears with a given term. In addition, these approaches consider the hashtags associated with only a small subset of words and ignores the common hashtags. Compared to KNN and Nave Bayes, the approach achieves a high performance. Similarly, in a study by Godin, et al. the authors consider the content of tweets [84]. However, they suggest hashtags based on detecting the hidden topics of the tweets, which are estimated using a Latent Dirichlet Allocation (LDA) model.

Some approaches have considered both tweet content and user information. These approaches are mostly based on graph models [85–87] and tensor factorization [6,88], where the recommendation system considers ¡user, tweet, hashtag¿ triples. For instance, Kywe, et al. recommend hashtags based on similar tweets as well as similar users [89]. Their approach considers the similarities between users based on the cosine similarity between their hashtags and also their tweets, separately. However, this approach considers the hashtags from similar tweets only when a user has never used hashtags before.

[90] proposed a statistical model for personalized hashtag recommendation based on ¡user, tweet, hashtag¿ using the wisdom of the crowds. Their approach considers (1) tweet-related features including terms, links, and mentions; (2) user-related features including user IDs, locations, and social relations; and (3) hashtag-related features which describe temporal characteristics of hashtag adoption. These features incorporate both latent factors and explicit features, in order to be used in the ranking function. However using these approaches increases the complexity of the model which makes it challenging to use for online recommendation and for processing huge twitter data set.

In addition to the hashtag recommendation problem, some efforts have focused on studying patterns of hashtags [91], for instance [92] explained how the rise and fall in hashtag adoption happens while [91] investigated the patterns of the hashtags to predict if any of them are still common the next day.

2.3 Evaluation of Recommender Systems

In the past decade, researchers have mostly focused on developing new recommender system algorithms [45]. They hence typically compared their new methods with rivaling methods using some numerical evaluation metrics or scores. However, it is now widely agreed that accurate predictions are crucial but insufficient to deploy a good recommendation engine [5].

In many applications, people use a recommendation system for more than an exact anticipation of their tastes. Users are also interested in discovering new diverse items. A good RS should respond fast while being careful about the users privacy [93]. Therefore, there are a set of properties which affect the success of recommender systems in the context of a specific application. Generally, three different types of experiments have been performed to evaluate a recommender system: offline experiments, user studies, and online experiments [93]. Each type of experiment tries to capture user satisfaction which is the most important factor in the success of a recommender system [62].

The most common type of evaluation is to perform offline experiments using available data sets. During these experiments, the RS is able to model the users behavior and then estimate some predicted ratings of recommended items. Then some evaluation metrics are used to measure the performance of the prediction accuracy. The next step is to ask a set of users to work with the system and provide information by answering a set of questions. Finally, the recommender systems is tested on real users as part of online experiments. Because the users, who are unaware of the purpose of the system, can provide valuable information, these experiments are considered to be the most trustworthy. In addition, the recommender systems can be tested on e-commerce websites where they can be evaluated based on specialized success metrics, including customer satisfaction with the recommendations and how often they follow recommendations.

2.3.1 Offline Experiments

An offline experiment uses a pre-collected data set to simulate user behavior when the users interact with a recommender system. This step is crucial since it helps researchers make reliable decisions while comparing a wide range of candidate algorithms at a low cost. A typical example of this process is to filter out inappropriate approaches, leaving a relatively small set of candidate algorithms to be tested by more costly user studies or online experiments. However, it is important to use data sets with no bias in the distributions of users, items and ratings selected.

2.3.2 User Studies

A user study experiment is performed by recruiting a set of test subjects, and asking them to interact with the recommender system. User studies are able to answer more questions since they allow researchers to collect qualitative data that is often crucial for interpreting the quantitative results. However, user study experiments are expensive to conduct and are practical for only a few scenarios. In addition, the subjects must be chosen such that they represent the real users of the RS as much as possible.

2.3.3 Online Evaluation

The real effect of the recommendation systems can be tested on real recommendation applications. This evaluation aids the researchers to investigate the RS better with providing strongest evidence as the true value of the system. In addition, researchers are able to study the influence on the behavior of the users in more detail.

As already mentioned, it is important to choose the data in offline experiments, and the real subjects in a user study, to best simulate the online application. And as in any type of experiment, one needs to reduce the possibilities of statistical errors. for instance by performing significance testing on the results. Significance can be assessed by computing confidence and p-values.

Several desired criteria capture the quality of a recommender system [45]:

Accuracy: Good recommender systems recommend items which are as close as possible to a user's preference including his upcoming action. The accuracy can be measured by statistical accuracy and decision support accuracy. The statistical accuracy evaluates the recommender systems based on numerical recommendation scores, such as MAE (Mean absolute error) and MAP (Mean average precision). The decision support accuracy shows the effectiveness by using metrics such as Receiver Operating Characteristic (ROC).

Efficiency: A recommender system needs to predict items in a reasonable time. In order to be able to evaluate the efficiency of a RS we consider memory and computation time. However, there is a trade-off between accuracy and efficiency, because often, speeding up a RS may result in pruning a portion of the information.

Scalability: Because real-life recommender systems work in a streaming manner, they should be able to handle hundreds of thousands of users and items to be practical. As in the case of efficiency, there is an unavoidable trade-off between the prediction accuracy and scalability. Moreover, handling concept drifts in data and usage patterns is important in both modeling and evaluation contexts [94–97].

In the following, we review several criteria that ca be used to evaluate a recommender system:

User Preference: In this evaluation, we ask the participants to choose one of the systems [98].

Prediction Accuracy: Depending on the recommender system objective, the prediction accuracy can be measured by: Root Mean Squared error (RMSE) between the predicted and actual ratings, Mean Absolute Error (MAE), Precision, Recall, Receiver Operating

Characteristic or ROC curves, Normalized Distance-based Performance Measure (NDPM), Normalized Cumulative Discounted Gain (NCDG) [93].

Serendipity: This measure gauges how surprising the successful recommendations are. Serendipity typically hurts accuracy, so, it is important to balance serendipity with accuracy. One way to measure the serendipity of a recommender is to ask the users to mark the recommendations that they find unexpected. This metric is only applicable in user subject tests in online study experiments.

Diversity: This criterion is generally defined as the opposite of similarity (between recommended items). It is important for a user to be able to view various recommendations which will result in shorter search interactions. One way to measure the diversity of a set of recommended item is based on the sum, average, minimum, or maximum distance between item pairs [99].

As mentioned before, each recommendation application needs proper metric to evaluate the used RS. However, the proper way of gaining such evaluation without intensive online testing or deferring to the opinions of domain experts is still an open question [93].

2.4 Recommender systems challenges

Traditionally, common recommender systems aim to improve rating prediction accuracy [62]. However, recent work started paying attention to other perspectives such as combating narrow rating prediction focus [21,100]. Recent real-world and laboratory studies indicated that improving user rating prediction accuracy alone does not necessarily increase the level of user-perceived quality [McNee et al., 2006; Jannach and Hegelich, 2009; Jannach et al., 2013; Cremonesi et al., 2011]. Therefore, a new stream of research, that started focusing on improving other aspects of recommender systems, has been gaining significant attention. For example, the ability of providing explanations alongside recommendations is an essential component which can improve the transparency of the working mechanisms within the recommender system [101]. In addition to the above, other approaches to improve recommender systems recently tried to go beyond rating prediction accuracy, for instance by also considering the diversity of items in a recommendation list [102–104].

It is important to note that recommender systems that improve diversity and serendipity are not the same as polarization aware recommender systems. This is because the former criteria generally require diversity in the actual description or nature of items, which in turn requires content data. Our research primarily focuses on items that can cross polarization boundaries, where polarization is based on how users interact with the items (ratings) and not their content.

2.5 Sentiment Analysis

Sentiment analysis (SA) is studying peoples opinions, emotions and attitudes toward an entity. Entities are usually in the format of reviews, comments and blogs. The first step in the SA classification task is to extract and select text features. Current features include: Part of speech, Opinion words, and Negations [105].

Generally, sentiment analysis methods can be categorized into three main approaches: machine learning, lexicon based, and hybrids [106]. Previous work on sentiment analysis has at least partially relied on expert knowledge, with some of this work aiming to classify the semantic orientation of words or phrases based on predefined seed words [107, 108]. Some of the work has focused on categorizing an entire document by manual or semi-manual construction of discriminant word lexicons [109, 110]. The lexicon-based approach focuses on finding a predefined lexicon which is then used to analyze the text. The Dictionary-based approach is one of the main strategies based on a lexicon. First, a small quantity of lexicon words is collected, then the algorithm searches well known corpora, such as WordNet for synonyms and antonyms [111]. After the process is finished, manual inspection is required to detect possible errors. On the other hand, a corpora-based approach aims to find opinion words with certain specific context orientations. Typically, this relies on syntactic patterns that occur together with a list of seed words to determine the overall sentiment of an instance [112].

Pang, et al. proposed to consider the sentiment analysis as a classification problem from a machine learning perspective [113], which since then, became a widely used approach to solve sentiment analysis problems. The sentiment classification methods, using a machine learning approach, can be roughly divided into supervised and unsupervised learning [114]. The supervised approaches take advantage of a large number of labeled documents, while the unsupervised methods are used when it is difficult to find labeled documents. Among supervised sentiment classifiers, the probabilistic classifier uses a mixture model for classification. The model assumes that the mixture is made of multiple classes. Each class is a generative model which gives the probability of sampling a particular term [115]. One popular probabilistic model is the Nave Bayes (NB) classifier. Hanhoon, et.al applied an improved NB classifier which achieves higher accuracy [115]. Prem et al. proposed a pooling multinomial Nave Bayes classifier in which the posterior probability of each word given the document is weighted from two sources, labeled documents and labeled features [116, 117]. They successfully applied this method to analyze the text messages in the 'voice of youth Africa' project. Another probabilistic model is the Bayesian Network (BN) classifier which assumes that all features are dependent. Therefore, the model needs to calculate the joint probability distribution over all variables (features). This method was used to solve a real-world problem within a semi-supervised framework [118]. One of the challenges associated with BN classifiers is their prohibitively high computational cost [118].

Another family of supervised models are linear classifiers. Given the word vector of a document, and a vector of weights of each word, the linear model assumes the label of a document is a linear combination between words [114]. Support vector machines (SVM) is one of the typical techniques, in the linear model family, that aims to determine a linear separator between different classes [119]. Sindhwani et al. proposed a graph-based semi-supervised learning technique, which combined both label and unlabeled documents in the linear regression objective function [120].

the decision tree classifier is considered another important type of sentiment classifier which provides a hierarchical decomposition of labeled training instance based on the attributes. The division is continued untill the leaf nodes contain a small number of instances [121]. Decision tree classifiers, in the sentiment analysis task, follow standard packages such as C4.5. Hu and Li developed a topical term description model for sentiment classification, in which they extract topic terms for particular topics and then differentiate documents based on those terms [122].

2.5.1 Review Sentiment Classification

Movie reviews contain emotional expressions by users about movies. Understanding the sentiment of movie reviews is critical to get first-hand information about the movies. Many prior efforts in this field followed a knowledge-based approach, which simply used the sentiment-polarity labeled words, defined in a preliminary stage in lexicons, and classified the text according to its linguistic patterns. An alternative approach is based on training machine learning models on documents that have been labeled with positive/negative sentiment, to learn text classifiers that distinguish the sentiments in domain-specific documents [123]. This solved the problem of adaptation for different and changing domains, however manual annotation requires too much human labor. Prem et al. proposed an approach that combined lexical knowledge and a multinomial Naive Bayes classifier model [116] to build successful sentiment classifiers for complex practical problems [117]. Given a Naive Bayes classifier, the rule to combine two resources is:

$$P(w_i|c_j) = \sum_{k=1}^{K} \alpha_k P_k(w_i|c_j) \tag{2.3}$$

where, K is the number of experts; K is 2 for two experts are considered in the proposed model; $P_k(w_i|c_j))$ represents the probability assigned by expert k to word w_i occurring in a document of class c_j; and the weights α_k are weights for combining these distributions and are normalized to sum to one. The weights for individual experts are calculated based on their error on training examples using:

$$\alpha_k = \log\left(\frac{1-e_k}{e_k}\right) \tag{2.4}$$

where e_k is the error of expert k on the training set.

2.6 Chapter Summary

In this chapter, we reviewed the background on recommender systems and sentiment analysis because of their relevance to our research. In the next chapter, we will present a new survey of polarization, including a new taxonomy of polarization. We conclude with discussing how our research relates to existing work in related fields.

CHAPTER 3

POLARIZATION IN THE ONLINE DOMAIN

In this chapter, we conduct a new survey of the topic of polarization in online spaces. We start with the closely related but distinct problem of concentration bias in recommendations, then proceed to the subject of polarization. We conclude with a definition of the scope of our research, and discuss how our work on polarization relates to but is distinct from the most closely related previous research on concentration bias, such as novelty, diversity and unexpectedness.

3.1 Over-personalization and Concentration Bias of Recommendations

Recommender systems typically **recommend items that are very similar to what the users have already purchased or liked in the past [20].** Although this feature has been the original aim of personalized recommender systems, it may lead to a strong positive feedback loop which makes the recommender system too personalized. Reinforcing similar item recommendation results in a closed loop in which the user and the RS reinforce each other until the 'ideal' personalized algorithm gets stuck in always recommending very similar items to very similar users. This reinforcement feature is inconsistent with some studies that have shown that diversification improves user satisfaction [25,35,99,124]. Following the same direction, Nguyen [25] aimed to answer the following research question: *Do recommender systems expose users to narrow content over time, at least in the domain of movies?*. To answer this question, they employed a longitudinal dataset to simulate users' interactions with RSs and found that RSs indeed expose the users to narrowing sets of items over time.

[125] argued that there are significant differences between recommender system al-

gorithms in terms of aggregate diversity. They showed that the rating distributions strongly vary depending on the RS algorithm and that algorithms such as Popular Rank would lead to making popular items even more popular. This makes those items constantly appear in the recommendation lists of more users. Other recommendation approaches such as Item-based KNN and Matrix Factorization have less of this effect, meaning that the recommendation of popular items is amplified only slowly over time. Finally, Collaborative filtering and User-based KNN were found to have less focus on popular items initially, and thus popular item bias remains stable or decreases again after several iterations.

In another direction, that however, requires full access to the content description of each item, a common solution to diversification is injecting randomness in the content-based recommendation process [72]. Another approach is filtering out items which are too similar (based on content) to items the user has purchased or liked in the past [126]. Some recent studies tried to identify less ordinary neighborhoods to create a diverse recommendation. One example includes an inverted k-nearest neighbors, which recommends items that are disliked by the least similar users [127, 128].

3.1.0.1 Novelty, Serendipity, diversity and unexpectedness of Recommendations

Recommender systems generally recommend items that maximize prediction accuracy. However, recent research has stipulated that a good recommender system needs to recommend items that are: [8, 21, 93]:

- Novel - items that the user did not know about.

- Serendipitous - (novel) items that positively surprise users.

- Diverse - a variety of 'relevant' items.

- Unexpected - non-obvious items for potentially higher user satisfaction.

Novelty: Researchers have been working toward fulfilling these criteria. A common approach to recommend novel items is by asking users to identify the items they already know [98]. [129] enhances novelty by partitioning the user profile into several clusters of similar items. They use these clusters to recommend items that match well with each cluster rather than with the entire user profile. In another direction, [130] presented a taxonomy-based RS that detects hot topics using association rules to improve novelty and quality of recommendations

Serendipity: Serendipitous recommendations are, by definition, also novel, however, increasing serendipity is challenging. Although the user might autonomously discover novel items, s/he would not be likely to discover serendipitous items. Some studies aim at detecting these items by finding those that are semantically far from the users' profile [131], while others adopt an assumption that users follow earlier adopters who have demonstrated similar preferences [132].

Diversity: Researchers in the field of RS and Information Retrieval (IR) have studied the principle of diversity to improve user satisfaction, over a span of the last two decades.. A common approach, pioneered by Billsus and Pazzani, is to decrease similarity among the items in the recommendation list by removing obvious items as well as very similar ones within the context of content-based RS [126]. Following the same direction, Zhang et al. [133] presented a multi-objective function optimization based on similarity and diversity. They improved their approach by formulating a binary optimization problem to capture the trade-off between diversity and recommendation accuracy [134]. Said et al. [128] presented another interesting approach based on an inverted nearest neighbor model which recommends items disliked by the least similar users.

Combined Approaches and Other Perspectives to Diversity: There have been recent studies that tried to promote novelty, diversity, and serendipity, at a slight cost to accuracy [135–137]. For example, [138] presented several contextual pre-filtering, post-filtering, and contextual modeling methods to promote accuracy, diversity, and serendipity under different circumstances.

In another stream of studies, researchers studied the importance of personalization and users' perception of diversity from different perspectives [139]. For example, some researchers presented different recommendation designs and investigated how the users' perception improved their satisfaction [122]. **[140] argued that the diversification level in a recommendation list should be adapted to the target users' individual situations and needs, and proposed a framework to adaptively diversify recommendation results for individual users based on latent factor models**. To this end, [141] showed that the combination of personalization and diversification could achieve significant performance in terms of both diversity and accuracy, compared to pure personalization or pure diversification-oriented approaches.

In addition, other researchers proposed a different approach to measure aggregated diversity where the RS is able to recommend across all users as many different items as possible, without a significant loss of accuracy [102,125,142]. For example, [102] controlled the promotion of popular items towards the top of the recommendation list in order to suggest more diverse items. Finally, in order to identify useful sources to achieve a better recommendation, [143] presented a comparative study on social system generated information in the recommendation process.

Unexpectedness: The field of unexpectedness in knowledge discovery has been well researched since the 1990's [144–147]. Recently, research has also paid more attention towards unexpectedness in recommender systems, where the algorithm is desired to suggest unexpected and non-obvious items [148, 149]. [150] presented a general metric to measure unexpectedness. The metric involves only an unlikely combination of item features and does not depend on a user's historic record. In another direction, [21] presented a new formulation of unexpectedness in recommender systems, such as the non-obvious items' significant departure from users' expectations. The recommendation process is based on the utility of the user, the quality that the user will gain from using the item, and the utility of the unexpectedness of the recommended product.

3.2 Polarization

Polarization is a common phenomenon that emerges in public opinion about contentious subjects or topics, and it can be observed in many forms [151]. In one form, users will congregate into communities (based on friend or follower connections) that are divided along polarizing issues, as demonstrated by the PEW Report [152]. In another form, users will congregate into polarized groups based on the tweets and particularly the hashtags that they choose to use when posting or sharing messages about a certain topic. In yet another form which is mainly dependent on filtering algorithms, users discover a narrower proportion of the information on social media, as a result of recommendation system algorithms, that filter the vast amounts of information [153]. Based on our survey, we found that polarization has been studied from three main perspectives (and occasionally hybrids thereof), leading us to define the following taxonomy:

1. social polarization: how people congregate with one another,
2. written polarization: how people write about topics,
3. rated and recommended polarization: how people behave, consume and express their preferences.

The following subsections will dive into each one of these perspectives to understanding polarization, in order.

3.2.1 Social Polarization: Polarization in Social Media and Networks

Over the last few years, researchers have turned to online data to study polarization, due to the importance of online political opinion formation. [154] studied the linking patterns of political bloggers with the aim to uncover differences in the structure of the two communities, such as how often they referred to the opposite viewpoint, topics they discussed, and conversations across the communities. Specifically, this research considered blogs that spanned two months before the U.S. Presidential Election of 2004. [154, 155] found an interesting common pattern which shows that users are less likely to share posts from the opposite party. [155] showed this fact by employing a combination of network clustering algorithms on the Twitter graph. Recently, [156] presented an approach to capture the opinions of a large number of politically active individuals and employed a combination of the Random Forest classifier over a constructed dataset.

3.2.2 Written Polarization: Polarization in the Textual Domain

In another direction, some researchers have studied polarization/controversy from the sentiment of terms used in text, such as tweets, articles, and blogs [157–159]. For example [158] presented an approach based on text sentiment analysis where they computed the magnitude of positive and negative sentiments as well as the difference between the amounts of two different polarities. [160] presented three Controversy Rank (CR) models in order to identify controversial articles on Wikipedia. The models uses the articles edit histories as the level of controversy. Based on their model, *"an article is controversial when it has lots of disputes among less contributors and a contributor is controversial when they are engaged in lots of disputes in less articles"*. [157] studied disagreement about an entity in the Twitter data. To do so, they presented a polarization score by combining the current sentiment of the text and its historical controversy score over time.

Research in social psychology has shown that emotions have a significant effect on people's opinion, judgment and evaluation [161]. Within this theoretical framework, emotions includes two fundamental dimensions: 1) valence, which captures the feeling of pleasure or displeasure, and 2) arousal, which measures the strength of a feeling [162]. Researchers have studied the correlation between polarization of reviews and arousal/valence expressed in the reviews [104].

3.2.3 Rated and Recommended Polarization: Polarization in Recommender Systems

As we have seen above, polarization has been investigated from a network perspective mainly using the social network structure, as well as the content and sentiment of discussions, in order to compute a polarization score [163]. However, this type of information is expensive to extract or is not always available. Hence, another line of work has studied polarization based on the *ratings* provided by users on items, within the context of a recommender system. Research on polarization in recommender systems has emerged rapidly, in recent years, as an important interdisciplinary topic [31, 38, 164, 165], with some efforts trying to decrease online polarization, especially in recommender systems [32, 38, 163].

Although various models have been proposed from different perspectives, there is not yet a general agreement on how to define or quantify polarization, let alone how to handle it. A simple but naive way to detect and quantify a polarized item is to inspect the standard deviation of the item's ratings. However, this approach is unable to distinguish between a flat distribution and U-shaped rating distribution, which represent diversity and polarization, respectively.

To overcome this limitation, [40] considered the standard deviation of adjacent ratings with respect to the total number of received ratings. The intuition is that different ratings that are close to each other reflect less disagreement than different ratings that are on opposite ends of the scale. This study was mainly focused on general disagreement/controversy rather than polarization, where they defined controversial items as:

"*Items that receive a variety of high and low scores, reflecting disagreement.*"

They defined the level of disagreement for a system, with ratings on a scale from 1 to M, as $(\alpha@\Delta)$, where Δ controls the granularity of a window in which adjacent ratings are being considered. The level of disagreement for item i is computed as follows:

$$(\alpha@\Delta)(i) = 1 - \max_{a\in 1,\dots,M-\Delta+1} \left(\frac{\sum_{k=a}^{a+\Delta-1} f_i(k)}{\sum_{k=1}^{M} f_i(k)} \right) \tag{3.1}$$

where $f_i(k)$ is the number of items with rating k

The proposed measure was used for trust-based recommender systems where there is an disagreement in a user's trust network over an item. In order to evaluate their proposed trust-enhanced recommender systems, they used the Epinions.com website, where users can provide feedback on different products. The e-commerce website asks users to indicate which member they trust and then recommends items based on the users' personal 'web of trust' (WoT).

In another research direction, [104] presented a polarization measure based on the geometric mean of likes and dislikes' distributions to investigate the existence of a local and a global regime U-shaped histogram. They defined (opinion) polarizations as follows:

"*Opinion polarization is characterized by a division of the population into a small number of fractions with high internal consensus and sharp disagreement between them.*"

They treated the logarithms of likes $ln(L)$ and dislikes $ln(L)$ as centrally distributed around their means, since the distributions of likes and dislikes per item are approximately log-normal. They also evaluated the role of items' emotion in the creation of polarization and found evidence that emotions, in particular arousal, has more effect on the polarization of opinions [166]. This is in line with other research in psychology that shows arousal can lead to extreme responses. To verify, they applied sentiment analysis on the title/header of each item to extract sentiment scores. Then after identifying the likes and dislikes distribution, they calculated the polarization score using the following equation:

$$\begin{gathered} Z_L = \frac{ln(L) - ln(L)}{sd(ln(L))} \\ Z_D = \frac{ln(D) - ln(D)}{sd(ln(D))} \\ Pol = \sqrt[2]{Z_l \times Z_D} \end{gathered} \tag{3.2}$$

Finally, the authors tested the effect of emotions on the polarization score using logistic regression models.

Following a similar intuition, [33] used a general formula as a function of the dif-

ference in the two opposite population sizes and their distance. To do so, they first built a distribution of opinions from a social network, distinguishing two types of individuals, *listeners* and *elite* users. The *elite* users have a fixed opinion, while the opinions of *listeners* depend on their social interactions, particularly with their neighbors. Using this model, they measured the network polarization. They defined polarization as:
"*A population is perfectly polarized when divided in two groups of the same size and with opposite opinions*".

Assuming that there are two, positive and negative opinions, they first estimated the normalized distance between these two opinions as follow:

$$\Delta A = |A^{+} - A^{-}| = |P(X > 0) - P(X < 0)| \tag{3.3}$$

Then, they measured the normalized distance between the two positive and negative gravity centers using Eq. 3.4 as below:

$$d = \frac{|gc^{+} - gc^{-}|}{|X_{max} - X_{min}|} = \frac{|gc^{+} - gc^{-}|}{2} \tag{3.4}$$

where gc^{-} and gc^{+} are the gravity centers of the positive and negative opinions, respectively:

$$\begin{cases} gc^{-} = \frac{\int_{-1}^{0} p(X)X\mathrm{d}X}{\int_{-1}^{0} p(X)\mathrm{d}X} \\ gc^{+} = \frac{\int_{0}^{1} p(X)X\mathrm{d}X}{\int_{0}^{1} p(X)\mathrm{d}X} \end{cases} \tag{3.5}$$

Finally, they proposed using Eq. 3.6 in order to estimate the network polarization score, which is 1 when the network distribution is perfectly polarized.

$$\mu = (1 - \Delta A)d \tag{3.6}$$

They applied their methodology to a Twitter conversation about the late Venezuelan president, Hugo Chavez, to study the effect of polarization. They particularly found that *"a minority of elite users were able to influence the whole online social network, resulting in a highly politically polarized conversation."*

In another research, [44] studied the temporal evolution of text-based reviews. In particular, the authors investigated the *self-selection bias*, where only users that strongly disagree with the item's current average rating will make an effort to provide explicit feedback. This can explain the reason behind extreme reviews/ratings by users when they do not think that the average ratings reflect the true quality of an item. This can make the users try to maximize their influence on the average rating score, which in turn may lead to polarization in case there are two groups of users who hold opposite opinions about the item in question. They defined the polarization degree using the opinion spectrum as follows:

"*At times the opinion distribution takes the form of U-shape, where almost all the mass is divided between the two extreme ratings.*"

To measure polarization for a given item, they computed the variance(V_k), mean (μ_k), mean deviation (Δk), and kurtosis (K_k) of a given item's ratings, at time k based on the ratings $r_1, ..., r_k$:

$$\begin{aligned} \mu_k &= \frac{1}{k}\sum_{i=1}^{k} r_i \\ V_k &= \frac{1}{k}\sum_{i=1}^{k}(r_i - \mu_k)^2 \\ \Delta k &= \frac{1}{k}\sum_{i=1}^{k}|r_i - \mu_{i-1}| \\ K_k &= \frac{1}{k}\sum_{i=1}^{k}\frac{(r_i - \mu_k)^4}{\sigma^4} - 3 \end{aligned} \tag{3.7}$$

These metrics were computed for an item to study the average behavior of items over time. In particular, the authors studied the temporal evolution of the ratings by computing the average values of the different metrics and plotting their trends. The authors found that users adopt more extreme opinions when they disagree with the average opinion, and concluded that this may be one reason why polarization increases over time.

To conclude, we note that there is neither reported evaluation on the aforementioned measures nor a comparison with other polarization metrics.

3.2.4 The scope of our Research: Rated and Recommended Polarization

3.2.4.1 Distinction Between our Research and Existing Research on Polarization

Most of the current work on polarization has either been limited to simple artificial toy problems (two users) [31], or relied on textual content to detect sentiment and then polarization, or were confined to specific domains where contentious issues lead to polarization. To date, most content-based studies have been typically conducted within the context of political (or other controversial domain) news and blogs. We are more interested in this research in studying the emergence and aggravation of polarization as a result of recommender systems.

It is along these differentiating features, that our research carves a distinct niche. More specifically, we do not confine our research to content-based, such as text, nor to controversial domain-specific issues. We mainly focus on ratings, in other words, the traces of interactions between users who provide preference labels and machine learning algorithms that use these labels to learn preference models.

Even when comparing to existing research that focused on rating-based (or rated) polarization, our approach presents several differences: In our proposed polarization detection approach, instead of dividing the rating histogram into two sub-populations and fitting two log-normal distributions, we consider the rating histogram all together at once to extract valuable information.

In contrast to the existing social network(graph)-based approach, we aim to identify if an item is polarized, rather than to conclude whether the 'entire' population is polarized. Obviously, more polarized items make the population more polarized. Therefore, the population-based polarization can be easily inferred from the item-based polarization. It is also important to estimate the polarization score for each item individually, since items are the main component that affects the efficiency of a RS. In other words, we need to know the polarization score for each item in order to be able to choose the best strategy when deciding whether to recommend it to a user. Hence, a population-based polarization score is not enough for this type of decision.

We also propose a generic model to detect polarization rather than several individual scores. It is important to be able to finally decide whether an item is polarized. However, it is hard to define a cut point for the aforementioned scores. Moreover, this cut threshold needs to be tuned for different datasets and domains, individually, making it difficult to choose.

In addition to the different methods, we note that although the aforementioned recent efforts tried to detect polarization, they lack an algorithmic approach that works in a domain-independent manner, and they used metrics that are either too simple to fully capture an item's polarization or too expensive and complicated to be used as a component of other algorithms, such as recommender systems. In addition, existing efforts to compute a score require this score to be binarized based on a threshold to finally decide if an item is polarized or not. Finding this threshold is a challenging task, as it is domain specific.

To overcome these limitations, we propose a data-driven machine learning pipeline to learn an optimal polarization classifier using features that are engineered based on rating distributions. When these features are used to build and train a classifier, the result is a Polarization Detection Classifier (PDT) that works in a domain and language-independent manner to detect and quantify polarization in a variety of domains. The similarity between the above-mentioned approaches and ours is limited to the intuition of detecting a U-shaped ratings distribution. However, **our methodology is the first]individual item rating-based' polarization detection classifier which is different from those found in the literature, because our proposed data science pipeline-based approach is designed based on a *data-driven methodology.*** In order to handle the case where no individual ratings are available (e.g. IMDb), we resort to the preliminary extra step of mapping reviews to ratings to build a ratings' histogram. In addition, we extensively evaluate our polarization pipeline on diverse real world data, and demonstrate that it outperforms existing polarization scores.

3.2.4.2 How is Polarization in Recommender Systems Different from Novelty, Diversity, and Serendipity?

Within the scope of recommender systems, polarization is distinct from the seemingly related aspects of novelty, diversity and serendipity in recommendations. As an example let's consider two users who like either red or blue colors. User 1 likes red outfits more than blue ones; in other words, s/he gives higher ratings to the red items. On the contrary, User 2 prefers blue items, which makes him/her give higher ratings to these items. Now, let's assume that User A has been purchasing only red shirts. A novelty-based recommendation would be to recommend red pants. Although the user has not seen these red pants before (hence being novel), s/he is able to discover it on her/his own, probably by searching long enough, particularly with an explicit intention to find 'matching color' pants to go with the red shirts.

A serendipitous recommendation would for example be recommending a red mug to the user. It is similar to the user's taste, however it is a novel suggestion which the user could probably not discover on his/her own, in case s/he keeps exploring the same area (say, clothes or anything outside the cups and mugs category) over and over again.

Recommending different red items from outfits to books increases diversity significantly. However, none of these tricks would make the recommender system recommend a blue item to red-loving user A. What if a blue belt went perfectly with a red dress? (as a matter of fact, some designer brands, such as Tommy Hilfiger and Ralph Lauren, seem to have captured this style and capitalized on it very well!). This scenario gets even worse if all the rest of the users are divided in two groups, where one group only likes red items and another only likes blue items. A traditional recommender system would group the red-loving users even more tightly and drive them even further from the blue-loving users from the point of view of interest and rating feedback; which would make the chance of being recommended a blue item, smaller for red-loving users. Despite the very simple color-based example, the scenario that we have just described is exactly what constitutes a polarization phenomenon, in this case, it is an increasing rift between users in the red and blue groups.

In order to give the user more options to discover choices in a polarized environment, we propose a polarization-aware recommender system.

3.3 Chapter Summary

In this chapter, we reviewed the existing research on polarization and defined a new taxonomy to distinguish between different families within this nascent research topic. We further discussed the scope of our research, and pointed to how our work on polarization relates to the most closely related previous research on concentration bias, such as novelty, diversity and unexpectedness. In the next chapter, we present our new research in polarization-aware recommender systems and in asking new research questions about the impact of polarization on the recommendation process.

CHAPTER 4

PROPOSED METHODS:
POLARIZATION IN RECOMMENDER SYSTEMS

4.1 Introduction

In this chapter, we start by presenting a new data-driven methodology for polarization detection in a set of items rated by a population of users. The approach is based on a data science pipeline that starts with feature engineering, modeling, evaluation, and interpretation. We first extract a set features based on items' rating histograms. Then we train a 2-class classifier to predict the polarization score for each item. Next, although our approach does not require the review text to determine polarization scores, we were curious about how the reviews relate to polarization. For this purpose, we investigate the possible relationships existing between the polarization score and item reviews. To do so, we perform Sentiment Analysis on the reviews written by users for each movie.

Finally, we propose a novel approach for countering polarization in a recommender system. We start with a counter-polarization strategy that can be used regardless of the type of recommender system. Next, we propose a novel NMF-based recommender system that can suggest a list of relevant items, while also considering several polarization scenario factors, such as a user's willingness to discover more items from the opposite polarity, the items' polarization scores, as well as the population polarization.

4.2 Polarization Detection Classifier

We propose a methodology to estimate the polarization of an item using a combination of features computed based on the item's rating distribution. Our methodology consists of two main steps: first, we construct a feature set from the histograms; and then,

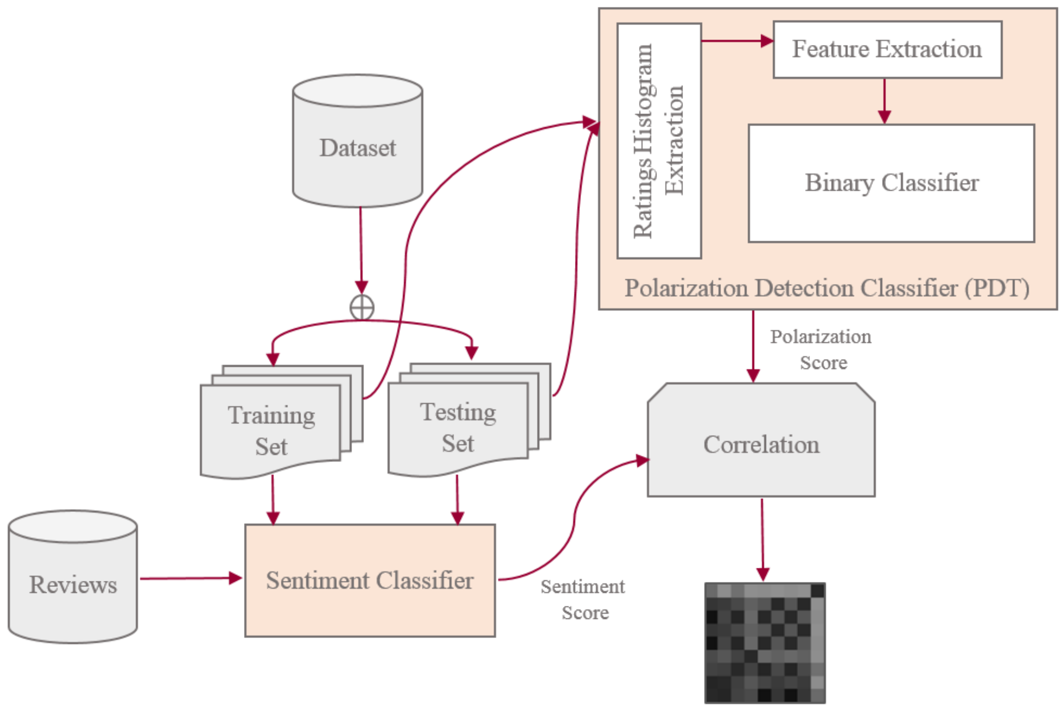


Figure 4.1: Proposed Polarization Detection Pipeline

we train a binary classifier model to estimate the polarization score for each item. Figure 4.1 shows the proposed polarization detection pipeline along with the sentiment classifier component.

4.2.1 Problem Definition

Definition 1 - Polarization: *Given an environment* $G = (U, I, R)$*, user* $u \in \mathbb{R}^{1\times n}$ *had rated item* $i \in \mathbb{R}^{m\times 1}$ *with rating* $r_{ui} \in \mathbb{R}^{m\times n}$ *on a scale of* x *to* y*. Item* i *'s polarization score* ϕ_i *is defined as the spread of its ratings* r_i*. We say the item is polarized if* $\phi_i \geq \delta$*.*

4.2.2 Feature Extraction

It is important to distinguish between "polarization" and "diversity", see figure 1.2a and 1.2b, respectively. Although both are social phenomena and can be sometimes related, they are not the same. A polarized item is only diverse in two opposite directions and this is sometimes considered a negative phenomenon in social studies. On the other

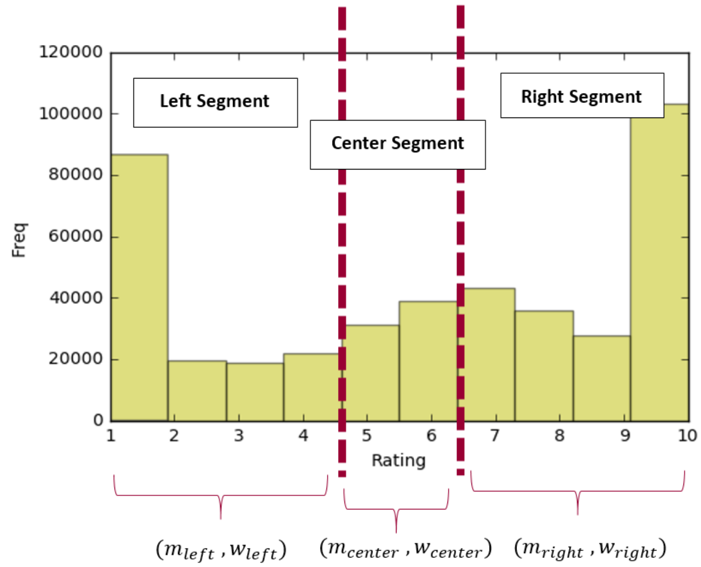


Figure 4.2: Proposed feature set 1, based on an item's rating histogram

hand, diversity is generally considered desirable as a social good since it represents different points of views or different varieties of choices. By focusing on the representation of the polarization phenomena within a distribution of user ratings, we introduce a set of features which capture polarization from the perspective of how the users react to the items, and not on the similarity or difference between the items' descriptions.

In general, there are three types of rating distributions, namely J-shaped, U-shaped and flat-shaped distributions, as shown in figure 1.2. As we can see, in a U-shaped distribution, the ratings are divided between the two extreme ratings. For this reason, we divide the distribution into *left*, *center* and *right* segments to identify these conceptual distinctions in the histograms of rating distributions. The segmentation can vary depending on the application and scale. For each segment, we compute the average rating and the count of ratings which is represented in Equation 4.1; we call it *Feature Set One*. The intuition behind taking into account the popularity of each segment is that polarization for an item with only a few ratings is probably not as significant as polarization for an item with many ratings. The six features that we define clearly capture the conceptual distinctions between polarized and unpolarized items based on the user ratings only, and hence are applicable to any domain with any rating scale. Figure 4.2 illustrates the extracted features using a movie's rating histogram.

$$F_1 = \{(w_{left}, m_{left})\} \cup \{(w_{center}, m_{center})\} \cup \{(w_{right}, m_{right})\}$$
$$\begin{cases} m_{left} = \sum_{j=1}^{4}(h_j) & w_{left} = \frac{\sum_{j=1}^{4}(h_j \times r_j)}{m_{left}} \\ m_{center} = \sum_{j=5}^{6}(h_j) & w_{center} = \frac{\sum_{j=5}^{6}(h_j \times r_j)}{m_{center}} \\ m_{right} = \sum_{j=7}^{10}(h_j) & w_{right} = \frac{\sum_{j=7}^{10}(h_j \times r_j)}{m_{right}} \end{cases} \tag{4.1}$$

where rating, r_j is on scale from 1 to 10 and

h_j = *number of items with ratings* $r_j, \forall j \in [1, 10]$

We extract another set of features, *Feature Set Two*, inspired by the literature [33, 36, 44, 104]. We start by estimating the best number of Gaussian distributions that can be fitted to the item's rating histogram. The next feature is based on an assumption that an extremely polarized item has two peaks in its rating histogram. As shown in figure 4.3, we first fit two Gaussian distributions, $\mathcal{N}(\mu_1, \sigma_1), \mathcal{N}(\mu_2, \sigma_2)$ and then compute the following features:

$$\Delta\mu = |\mu_1 - \mu_2|$$
$$\Delta z = |z_1, z_2|, \tag{4.2}$$

where z_1 is the peak of $\mathcal{N}(\mu_1, \sigma_1)$ and z_2 is the peak of $\mathcal{N}(\mu_1, \sigma_1)$

In addition, we calculate the similarity between the two fitted Gaussian distributions using the Chi-square measure [167].

4.2.3 Classification Model

After extracting the feature set for each item, we train a binary classifier, which needs ground truth labels. To do so, we asked multiple experts to manually categorize each item into the polarized or non-polarized class. To aid in annotation, a web application was created. Each item's ratings histogram was shown to at least 3 annotators, where they were given the definition of polarization and were instructed to identify the polarized items.

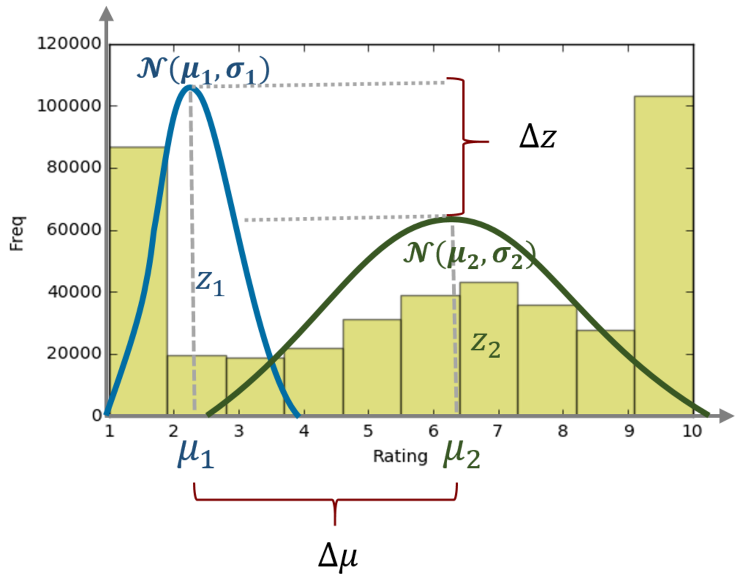


Figure 4.3: Proposed feature set 2, based on an item's rating histogram

In case of a tie, we use a majority voting scheme to decide the final category of the item. We rely on human intuition to detect polarization. A human is able to tell if an item is polarized or not by only looking at its rating histogram, often without needing to know the item's detailed information. Hence, for the sake of removing any bias, we hid all of the item's information, except for the rating histogram, from the annotators.

We trained different binary classifiers and finally chose the Random Forest Classifier [168] due to its high predictive accuracy and strength in handling imbalanced datasets. After the classification, the predicted probability of belonging to the 'polarized' class is considered as the polarization score for an item.

4.2.4 Item Review Sentiment Classification

In order to check the possible relationship between item polarization and reviews, we performed sentiment analysis on the reviews from each item, e.g. movie. The label for each review was based on the rating given by the user to the movie. We discretized the ratings into a binary scale, i.e. 1 if the rating is higher than 5, and 0 otherwise. From our observation, building the review sentiment classifier on the whole review data led to under-fitting, since our collected movies contain various genres, requiring distinct lexicons.

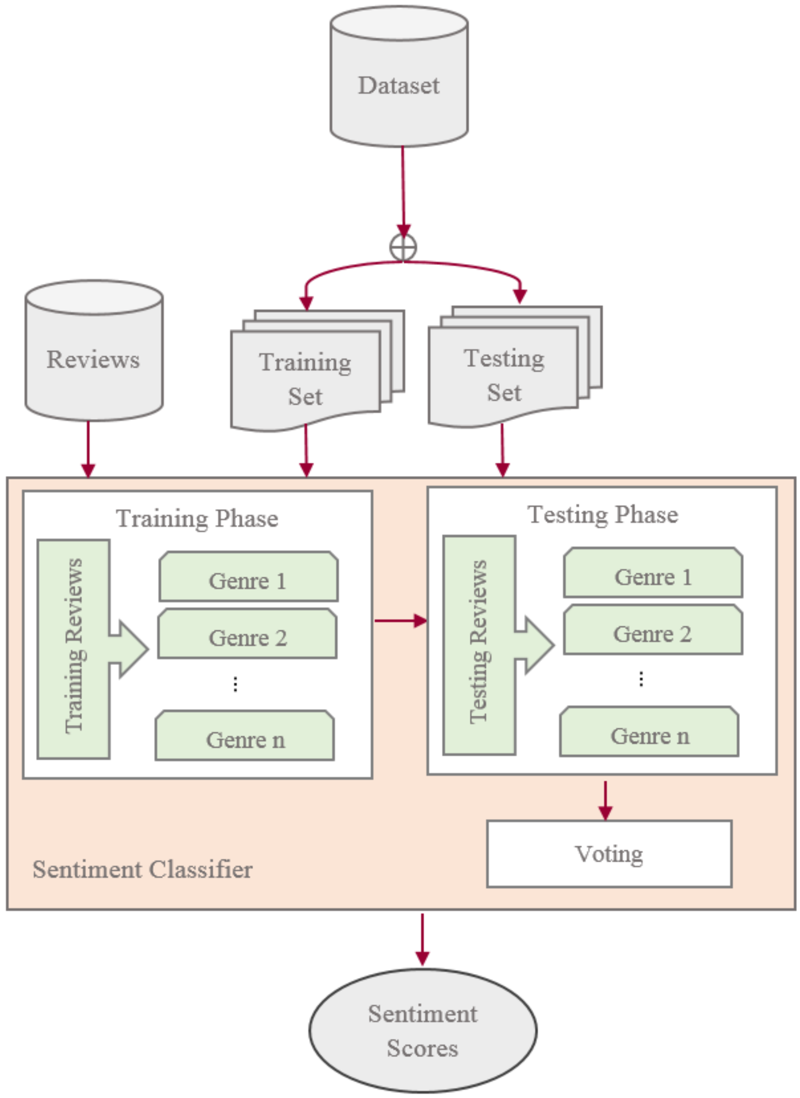


Figure 4.4: Proposed Polarization Detection Pipeline

On the other hand, building a sentiment classifier for each movie led to over-fitting, since it does not lead to good generalization for unseen movies. Therefore, we built a different sentiment classifier for each movie genre. When a movie has multiple genres, we decide its final sentiment label by voting based on the predicted labels from all genre classifiers. Figure 4.4 shows the steps of the presented sentiment approach.

We used a multi-pooling review sentiment classifier (see Chapter 2), which combines both lexical background knowledge and labeled reviews, due to its demonstrated efficiency and adaptability to diverse contexts [116].

To understand the correlation between polarization and arousal and valence, we adopted the approach in [104]. We used a lexicon of affective norms for valence (V) and arousal (A) of about 14,000 English words [169]. We averaged the V and A scores across all reviews for each movie, and assigned the score to that movie.

4.3 Handling Polarization in Recommender Systems

4.3.1 Problem Definition

We first define the problem of polarization-aware collaborative filtering (CF).

Definition 2 - polarization-aware collaborative filtering recommendation: Given a set of ratings $R \in \mathbb{R}^{m \times n}$ collected from a set of users $U \in \mathbb{R}^{1 \times n}$ for a set of items $I \in \mathbb{R}^{m \times 1}$, the problem of polarization-aware collaborative filtering recommendation (CF) can be modeled by the triplet (U, I, R), in a way that a recommender system should recommend a ranked item set $i_1, ..., i_t \in I$ according to 1) the relevance of the item to the user's interest, and 2) the item's polarization score. As a realization from definition 2, (U, I, R) can be denoted by (u, i, r) which means that user u rated item i with value r.

In this section, we will discuss possible strategies to deal with the undesirable effect of the polarization phenomenon and present novel algorithmic approaches that can be used to handle it without compromising too much on relevance-based (i.e. pure rating) predictive accuracy. Figure 1.1 shows a closed feedback loop between a user and a Machine Learning algorithm, in our case a Recommender System (RS).

We propose a novel recommendation system based on item-user ratings, item's polarization score and population polarization. In addition, we propose a strategy that can counteract population polarization, independent of a RS algorithm. This means that it later can be employed in a pre-filtering stage along with any recommender system algorithm. This approach mainly consists of two strategies. Figure 4.5 displays a flowchart of the different components.

In principle, several approaches are possible for counter-polarization of recommendations in the presence of polarization:

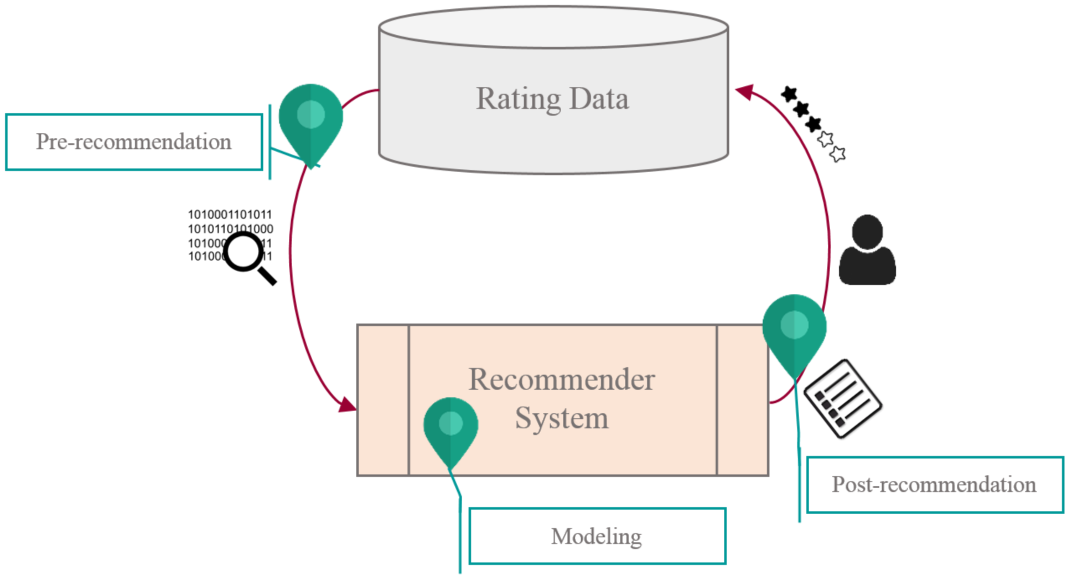


Figure 4.5: Proposed strategies for counteracting polarization in a recommender system feedback loop

1. Applying a pre-recommendation scheme to modify the input data before it is fed into a recommendation technique.

2. Optimizing a new objective function for a polarization-aware recommendation algorithm, such that the objective considers several factors when making a recommendation to a user.

3. Applying a post-recommendation scheme to modify the outcomes of a given recommendation framework and thus help mitigate the potentially undesirable effects of polarization.

One might also think of alternatives such as a straightforward counter-polarization approach, consisting of just including some randomly selected items from the opposite view. Temporarily, this would seem to solve the filter bubble problem and increase the diversity of the recommended list. However it would cause much information loss which leads to recommending irrelevant items and eventually risks reducing user satisfaction with the system. In addition, such a remedy is not able to solve the filter bubble problem for a long period.

In this research, we are more interested in the first and second approaches; and we will leave the third option to the future, and particularly to Human Interaction Computer (HCI) experts. We first propose a pre-recommendation approach to mitigate polarization from the population before learning any model. This is a useful strategy since most online system providers are using a RS as a black box; hence, it is difficult to look into the inner workings of the algorithm to modify it. We also present an approach in the context of the classical collaborative filtering (CF) Recommender system algorithm. We use latent factor models, specifically Non Negative Matrix algorithm (NMF), to characterize both items and users based on a set of factors inferred from user-item rating patterns. However, the proposed approach is not specific to NMF and can be easily extended to any RS method. The goal of our proposed recommender system is not to guide the user to the mostly popular items or the "long" tail of items but rather to include items that help users become aware of other items that s/he is not able to discover on her/his own.

4.3.2 Pre-recommendation: Countering Polarization

Changing the Recommender System algorithm is not always feasible due to many black box systems. Hence, modifying the learning model can be hard and time consuming. In addition, in real-life, recommender system platforms are very complicated and modifying the learning model relies on understanding the learning model, which can be challenging. Therefore, a systematic pre-recommendation strategy seems most flexible, since it does not require changing the algorithm or modeling approach. In this step, we aim to transform the source data in such a way that it mitigates extreme ratings that make an item polarized. By doing this, we still keep the user's relative preferences, yet make it more moderate so that no extreme recommendation can be generated from a standard recommender system algorithm. We perform a controlled distortion of the training data based on which a recommender system is trained to help the users receive more useful recommendations, in the presence of polarization. This transformation is based both on the user's willingness to discover more items and on the item's polarization score.

In addition, the proposed pre-recommendation approach is useful for other applications where a dataset should be either published by an online recommender system provider or used for other research.

The proposed solution to counteract polarization by making the training dataset less polarized, employs a stochastic mapping function as defined below:

$$f : (U, I, R) \rightarrow (U, I, R') \text{ with probability } p \tag{4.3}$$

The function transforms a user-item rating based on the rating itself, population average rating, item's polarization score and user's chosen discovery factor, as follows.

$$\begin{aligned} r'_{ui} &= r_{ui} - \lambda_u \times (\bar{r} + \frac{g_i}{g_{max}}) \times \Phi_i^{\lambda_u + r_{ui}} \quad \text{if } r_{ui} \geq \delta \\ r'_{ui} &= r_{ui} + \lambda_u \times (\bar{r} - \frac{g_i}{g_{max}}) \times \Phi_i^{\lambda_u + r_{ui}} \quad \text{if } r_{ui} < \delta \end{aligned} \tag{4.4}$$

where $\lambda_u \in [0, 1]$ is the user's selected discovery factor. At one extreme, It is 1 when the user indicates that s/he is interested in discovering more items, especially from the opposite view. At the opposite extreme, if the user sets $\lambda = 0$, the result reduces to using only the classical recommendation algorithm which aims to minimize the squared error on the set of raw ratings. Note that if a user expresses an interest in considering items from the opposite view, it does not necessary mean that s/he would definitely like or purchase those items. The goal here, is to simply give an option to the users to be able to burst out of their filter bubbles. $\Phi_i \in [0, 1]$ is the polarization score which is computed using the Polarization Detection Classifier. $g_i \in [0, 1]$ indicates the gap between the two rating extreme ranges for a polarized item, in other words it measures how polarized the user population's rating is for item i. We define the gap g_i as the difference between an item's minimum rating that a typical user provide as s/he liked the item and maximum rating that the user provides to show s/he did not like the item. In other words, the gap g_i captures the difference between extreme opinions regarding an item. g_{max} is simply the difference between the maximum and minimum rating that a typical user can provide for any item, using the system's rating scale. The more polarized a population gets, the higher g_i gets. δ indicates which ratings are considered as liked versus disliked. This is usually set to $\frac{r_{max}-r_{min}}{2}$, for example, for an arbitrary rating $1 \leq r_{ui} \leq 10$, if the rating scale is $(1, 10)$ then a user likes/dislikes an item if s/he rates an item more/less than 5. However, this threshold can depend on the application of the recommender system and the item domain. $\bar{r}$ is the average of ratings over the entire population. It helps to moderate more extreme ratings while we keep the exact value for those ratings close to average.

4.3.3 Polarization-aware Recommender Interactive System

Matrix Factorization (MF)-based recommender systems have shown good performance by combining good scalability with predictive accuracy [62]. However, different matrix factorization-based techniques have quite different behavior with respect to how many different items they recommend [41]. This is due to the fact that algorithms have different specific ways to learn their model and then generate predictions. Here, we are interested in the latent space that is induced using a Non-Negative Matrix Factorization (NMF) algorithm on a user-item matrix. Thus we investigate the recommended list of items from an NMF perspective.

In order to have a good recommender system, the specific application domain and recommendation goals should be taken into account when deciding on an algorithm. For example for a user, recommending popular items might provide valuable reminders, but could be of little value when it comes to discovering new items. Here, our goal is to design a recommendation system which not only recommends relevant items but also includes opposite views in case the user is interested to discover new items.

Recall that the general problem of non-negative matrix factorization (NMF) is to decompose a non-negative data matrix R with size $m \times n$ into two positive elements matrix factors P and Q, with size $m \times k$ and $k \times n$, respectively, such that $k \ll \min(n, m)$ is a positive integer representing the rank of matrices P and Q. A classical NMF predicts the overall rating $\hat{r_{ij}}$ by minimizing $||r_{ij} - p_i q_j||^2$. However, the item rating alone is not able to fully consider the polarization phenomenon. Hence, in order to estimate R from (U, I)

such that the system considers both relevance and polarization, we minimize the following objective function 4.5. The learned prediction function of the proposed polarization-aware recommender interactive system (PaRIS) has the following form:

$$\begin{aligned}
\min_{p_u,\, q_i} \quad \mathcal{J}_{PaRIS} &= \sum_{u \in U} \sum_{i \in I} \Big((1-\lambda_u) \times ||r_{ui} - p_u q_i||^2 + \lambda_u \times ||r'_{ui} - p_u q_i||^2 \Big) \\
r'_{ui} &= r_{ui} - \lambda_u \times (\bar{r} + \frac{g_i}{g_{max}}) \times \Phi_i^{\lambda_u + r_{ui}} \quad \text{if } r_{ui} \geq \delta \\
r'_{ui} &= r_{ui} + \lambda_u \times (\bar{r} - \frac{g_i}{g_{max}}) \times \Phi_i^{\lambda_u + r_{ui}} \quad \text{if } r_{ui} < \delta
\end{aligned} \tag{4.5}$$

where λ_u is the user discovery factor, Φ_i is the item's polarization score, computed using the Polarization Detection Classifier, g_i measures how extreme the different view points are, δ indicates which ratings are considered as liked versus disliked. The rest of the variables are the same as the ones used in Eq. 4.49.

The first part of the optimization equation is the classical NMF optimization criterion; while the second part is the counter-polarization component. The intuition behind the second part is to bring a user and an item closer in the latent space, if the user is interested in discovering more and the item is polarized. Recall that if we assume the user only sees items that are recommended, then the user is not able to discover a polarized item from the opposite view point on his/her own because a personalized recommender system filter such an item. The new incremental stochastic Gradient Descent update equations, after each new input, can be derived as follows:

$$\begin{aligned}
e_{ui} &= r_{ui} - p_u q_i \\
e'_{ui} &= r'_{ui} - p_u q_i
\end{aligned} \tag{4.6}$$

$$\begin{aligned}
\frac{\partial \mathcal{J}_{PaRIS}}{\partial p_u} &= 2(1-\lambda_u) \times e_{ui} \times (-q_i) + 2\lambda_u \times e'_{ui} \times (-q_i) \\
\frac{\partial \mathcal{J}_{PaRIS}}{\partial q_i} &= 2(1-\lambda_u) \times e_{ui} \times (-p_u) + 2\lambda_u \times e'_{ui} \times (-p_u)
\end{aligned} \tag{4.7}$$

$$\begin{aligned}
p_u^{t+1} &= p_u + \eta\big(2q_i((1-\lambda_u)e_{ui} + \lambda_u \times e'_{ui})\big) \\
q_j^{t+1} &= q_i + \eta\big(2p_u((1-\lambda_u)e_{ui} + \lambda_u \times e'_{ui})\big)
\end{aligned} \tag{4.8}$$

4.4 Simulating the Interactive Recommendation Process

Recommendation service providers are typically interested in the long-term effects of the recommender system on user satisfaction. Although conducting A/B tests is a common approach, measuring such effects is difficult [41]. Due to the discussed reason, a personalized recommender system can lead to decreased diversity of the recommendations [41, 104, 159].

In order to assess the effect of the proposed strategies on the diversity of recommendations in a polarized environment, we conduct an experiment to simulate the interaction between a user and a recommender system, as shown in algorithm 4.1 . The main idea of the proposed Polarization-aware Recommender Interactive System - (PaRIS) is to start with a given rating database and incrementally add new ratings to it as would happen on a real web platform. The assumption is that the system learns a model from the given rating database and then uses the model to recommend a set of new items to the user. We assume that the user chooses only one item randomly from the recommended list and rates it. Hence the system creates one additional rating per user for the chosen item. The new rating comes from the ground-truth rating database for that user and item. After adding the new rating to the dataset, **the assumption is that the selection of which items are actually selected to be rated by the user is to some extent determined by the deployed recommendation system.** We repeated this procedure over all users 100 times to simulate the evolution of the recommendation lists over time. In each iteration, we measure a) MSE and b) the number of items from a the opposite viewpoint/polarity that were recommended in this round. We see the proposed simulation approach as a complementary method to investigate the performance of a recommendation process in a polarized environment in an offline experimental setting.

4.5 Theoretical Analysis of Polarization's impact on Models and Counter-Polarization Methodologies

We start with studying the interaction between polarization and the update equation for Gradient Descent in NMF. We then show how the proposed counter polarization

Algorithm 4.1 Polarization-aware Recommender Interactive System - (PaRIS)

Input: initial user-item matrix (R_{train}), f_{PDC}(Polarization classifier function)
Output: final user-item matrix R

procedure Polarization Detection Classifier (PDC)
For each item $i \in I$:
    $h_i \leftarrow$ extract rating histogram
    $x_i \leftarrow$ extract features from h_i
    $\phi_i \leftarrow f_{PDC}(x_i)$
    $\Phi \leftarrow \Phi \cup \Phi_i$
    $g_i \leftarrow \frac{\max_u (r_{ui}) - \min_u (r_{ui})}{\max_u (R_u) - \min_u (R_u)}$
procedure Iterative Recommendation Process (IRP)
For each $u \in U$:
    Repeat While u rates unrated items:
        Update model P, Q using input data R by optimizing objective function 4.5
with the parameter set of λ_u, Φ_i, g_i
        $\bar{R} \leftarrow PQ$
        Find S_u which is the set of items sorted in descending order of predicted rating
        Select top k_t items from S and recommend them to u
        User picks an item i' randomly and gives rating $r_{ui'}$
        $R \leftarrow R \odot r_{ui'}$ for $i' \in S_u$
          where $\odot$ is the I/O operator,meaning that user u provides rating $r_{ui'}$ for item i'

solutions change the impact.

Recall the general problem of non-negative matrix factorization is to decompose a non-negative data matrix R with size $m \times n$ into two factor matrices P and Q with size $m \times k$ and $k \times n$ respectively, such that:

$$R_{m \times n} = P_{m \times k}.Q_{k \times n} \tag{4.9}$$

where $k \leq \min(n, m)$ is a positive integer representing the rank of matrices P and Q, and the constraint is $P, Q > 0$.

The most common way to estimate P and Q, is by minimizing the Frobenius norm of the errors between the approximation P, Q and the data matrix R. The objective function is defined as follows:

$$\mathcal{J} = \sum_{u \in U, i \in I} (r_{ui} - p_u q_i^T)^2 \tag{4.10}$$

In order to find the stationary point of the optimization problem (P and Q), an

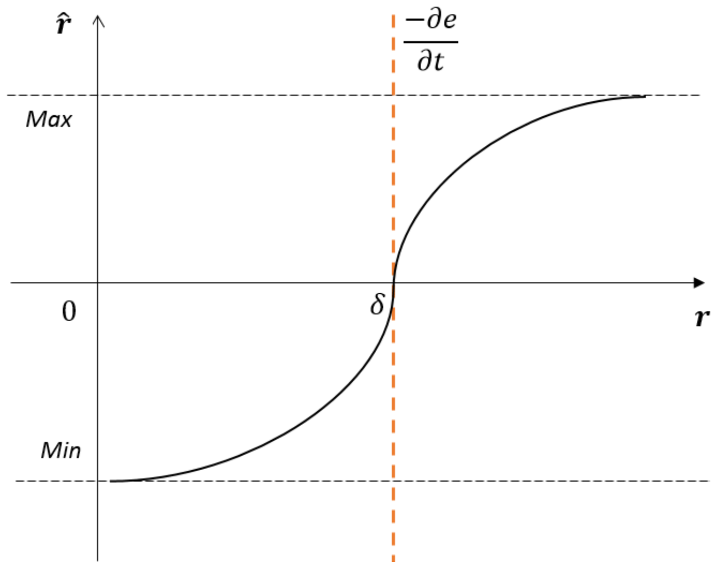


Figure 4.6: Relationship between $\hat{r}$ and r in a polarizing RS: ratings r are pushed toward the extreme by predicting polarized ratings that (1) exceed the true rating if the item is liked (rating higher than δ) and (2) are less than the true rating if the item is not liked (rating less than δ)

incremental gradient-descent technique is applied, as follows:

$$p_u^{(t+1)} = p_u^{(t)} - \eta(e_{ui}^{(t)} q_i^{(t)}) \tag{4.11}$$

$$q_i^{(t+1)} = q_i^{(t)} - \eta(e_{ui}^{(t)} p_u^{(t)}) \tag{4.12}$$

The rationale for polarization is given by the following definition:

1. **Definition 3 - Polarizing RS**: One way that NMF can be polarizing is if the model updates will push the ratings toward the extremes, hence further polarizing them, this could be formulated as follows, and is illustrated in 4.6:

$$IF \quad \hat{r}_{ui}^{(t)} > \delta + g \quad then \quad \hat{r}_{ui}^{(t+1)} > \hat{r}_{ui}^{(t)} \tag{4.13}$$

$$IF \quad \hat{r}_{ui}^{(t)} < \delta - g \quad then \quad \hat{r}_{ui}^{(t+1)} < \hat{r}_{ui}^{(t)} \tag{4.14}$$

2. **Definition 4 - counter-polarization strategies**:

- **Remedy 1** - *Pre-recommendation counter polarization*: avoid that initial rating $r > \delta + g$ or $r < \delta - g$, using the mapping function 4.3: $\Rightarrow f(r) \rightarrow r^{'}$ where $r^{'}$ is less polarized

- **Remedy 2** - *Polarization-aware recommender system*: Modify the inner working Recommender system to learn to reconstruct less polarized ratings $r^{'}$. This will ideally result in the following change:

$$\begin{cases} \text{IF} \quad \hat{r}_{ui}^{(t)} \geq \delta \quad \text{Then} \quad \hat{r}_{ui}^{(t+1)} \leq \hat{r}_{ui}^{(t)} \Rightarrow \frac{\partial \hat{r}}{\partial t} \leq 0 \\ \text{IF} \quad \hat{r}_{ui}^{(t)} \leq \delta \quad \text{Then} \quad \hat{r}_{ui}^{(t+1)} \geq \hat{r}_{ui}^{(t)} \Rightarrow \frac{\partial \hat{r}}{\partial t} \geq 0 \end{cases} \tag{4.15}$$

In the following we use the chain rule to rewrite the gradient of predicted ratings, where θ are the model parameters.

$$\frac{\partial \hat{r}}{\partial t} = \frac{\partial \hat{r}}{\partial \theta} \times \frac{\partial \theta}{\partial t} \tag{4.16}$$

For example, in case of Gradient Descent in NMF, where $\hat{r_{ui}} = p_u.q_i$ and $\theta = (p_u, q_i)$ we rewrite equation 4.16 as:

$$\frac{\partial \hat{r}}{\partial \theta} = \begin{cases} \frac{\partial \hat{r}}{\partial p_u} = q_i \\ \frac{\partial \hat{r}}{\partial q_i} = p_u^T \end{cases} \tag{4.17}$$

$$\frac{\partial \theta}{\partial t} = \begin{cases} \frac{\partial p_u}{\partial t} \\ \frac{\partial q_i}{\partial t} \end{cases} \tag{4.18}$$

The gradient, for small steps Δt, can be approximated using a linear approximation by Taylor Series first order expansion.

Using 4.18, we have:

$$\begin{cases} \frac{\Delta p_u}{\Delta t} = \frac{p_u^{(t+1)} - p_u^{(t)}}{\Delta t} = \eta \times \frac{e^{(t)} \times q_i^{(t)}}{\Delta t} = \eta \times \frac{(r - \hat{r^t}) \times q_i^{(t)}}{\Delta t} \\ \frac{\Delta q_i}{\Delta t} = \frac{q_i^{(t+1)} - q_i^{(t)}}{\Delta t} = \eta \times \frac{e^{(t)} \times p_u^{(t)}}{\Delta t} = \eta \times \frac{(r - \hat{r^t}) \times q_i^{(t)}}{\Delta t} \end{cases} \tag{4.19}$$

we can write 4.19 as follows:

$$\begin{cases} \frac{\Delta p_u}{\Delta t} = \eta \, \frac{(\hat{r}-r)}{\Delta t} \times q_i^{(t)} \\ \frac{\Delta q_i}{\Delta t} = \eta \, \frac{(\hat{r}-r)}{\Delta t} \times p_u^{(t)} \end{cases} \tag{4.20}$$

From 4.18 and 4.19:

$$IF \, \frac{\partial e}{\partial t} > 0 \Rightarrow \frac{\partial \theta}{\partial t} < 0 \tag{4.21}$$

$$IF \, \frac{\partial e}{\partial t} < 0 \Rightarrow \frac{\partial \theta}{\partial t} > 0 \tag{4.22}$$

Here, we are only interested in the sign of each derivative:

$$Sgn\left(\frac{\Delta\theta}{\Delta t}\right) = Sgn\left(\frac{\hat{r}-r}{\Delta t}\right) \tag{4.23}$$

$$\tag{4.24}$$

What 4.23 says is that the difference between the predicted and true rating has the same sign as the gradient of the model parameter. In gradient descent, we know what the parameter gradient's sign is: If the gradient of the objective function is positive, then the parameter updates are such that gradient of the parameters have to be in the opposite direction (i.e. negative) to minimize the objective function. The opposite is true (if the objective's gradient is negative, then the parameter gradient will be positive).

In other words the sign of $\hat{r}-r$ will be the opposite of the objective function gradient

Hence:

$$Sgn\left(\frac{\partial\theta}{\partial t}\right) = Sgn\left(\frac{\hat{r}-r}{\Delta t}\right) \tag{4.25}$$

$$\tag{4.26}$$

Our aim is to show that the polarization rate after the proposed remedies is reduced. On the other hand, a steeper Slope in gradient descent leads to a faster polarization. Using pre-recommendation as a counter-polarization strategy, definition 4-(Remedy 1), we are interested to show that the magnitude of the gradients:

$$\left|\frac{\partial e^{'}}{\partial t}\right| < \left|\frac{\partial e}{\partial t}\right| \tag{4.27}$$

$$where \quad e^{'} = r^{'} - \hat{r} \quad \text{and} \quad e = r - \hat{r} \tag{4.28}$$

$$\tag{4.29}$$

$$\hat{r}^{'} - r^{'} = \left(\hat{r}^{'} - r\right) + \left(r - r^{'}\right) \tag{4.30}$$

$$= e - \left(r^{'} - r\right) \tag{4.31}$$

$$= e + sgn(r - \delta) \times \lambda_u(\hat{r} + sgn(r - \delta)\frac{g_i}{g_{max}} \times \phi_i^{\lambda_u + r_{ui}}) \tag{4.32}$$

$$\tag{4.33}$$

From 4.30:

$$e^{'}_{ui} = \begin{cases} e_{ui} + \lambda_u(\hat{r} + \frac{g_i}{g_{max}} \times \phi_i^{\lambda_u + r_{ui}}) & if\, r_{ui} > \delta \\ e_{ui} - \lambda_u(\hat{r} - \frac{g_i}{g_{max}} \times \phi_i^{\lambda_u + r_{ui}}) & if\, r_{ui} > \delta \end{cases} \Rightarrow \begin{cases} e^{'}_{ui} > e_{ui} & if\, r_{ui} > \delta \\ e^{'}_{ui} < e_{ui} & if\, r_{ui} < \delta \end{cases} \tag{4.34}$$

One way to reduce polarization is to control the change that is caused by each update step in gradient descent. For example, one could try to slow down updates or even allow updates that would increase the objective function, for instance by using Simulated Annealing. Also, methods that speed up convergence may be harmful in polarized scenarios, e.g. Steepest descent, Newton, etc.

4.6 Gap Effects in a Perfect Polarization Scenario

Let us suppose that we havea perfectly polarized population around an item. Users are either in Group A or Group B and the ratings of Group A are shifted by g_i compared to the ratings of users in Group B. This means that $r_{u_1 i} = r_{u'_1 i} + g_i$, where u_1 and u'_1 are users from the opposite poles (A and B, respectively) whose ratings differ by g_i, we assume for every user in Group B, there is a user in Group A with ratings shifted by g_i.

We study this simplified problems and show how the gradient changes with g_i. Note that our perfect polarization assumption means that every user in pole A has their reflection in the opposite pole B, such that the ratings differ by g_i.

We further assume a perfect polarized situation for all items, that is $\forall i \in I$, users are polarized.

Recall that the original NMF objective function, in a perfectly polarized scenario, can be divided into separate terms, as follows :

$$\mathcal{J} = \sum_{u \in U} \sum_{i \in I} (r_{ui} - p_u q_i)^2 \tag{4.35}$$

$$= \mathcal{J}_{\mathcal{A}} + \mathcal{J}_{\mathcal{B}} \tag{4.36}$$

$$= \sum_{u_1 \in U_A} \sum_{i \in I} (r_{u_1 i} - p_{u_1} q_i)^2 + \sum_{u_2 \in U_B} \sum_{i \in I} (r_{u_2 i} - p_{u_2} q_i)^2 \tag{4.37}$$

(4.38)

Here we assume that there are two groups of users, U_A and U_B, which are users from opposite poles with regard to their ratings on item i.

$$\mathcal{J} = \mathcal{J}_{\mathcal{A}} + \mathcal{J}_{\mathcal{B}} \tag{4.39}$$

$$= \sum_{u_1 \in U_A} \sum_{i \in I} (r_{u_1 i} - p_{u_1} q_i)^2 + \sum_{u_2 \in U_B} \sum_{i \in I} (r_{u_2 i} - p_{u_2} q_i)^2 \tag{4.40}$$

(4.41)

Perfect polarization implies that:

$$\forall u_2 \in U_B : \exists\, u'_1 \in U_A \;\Big|\; r_{u_2 i} = r_{u'_1 i} + g_i \tag{4.42}$$

where u'_i is the anti-pole of u_2, meaning the anti-pole of a user u is user u' whose rating is shifted relative to the rating of user u by the gap amount g.

We write 4.39 as follows:

$$\mathcal{J} = \sum_{u_1 \in U_A} \sum_{i \in I} (r_{u_1 i} - p_{u_1} q_i)^2 + \sum_{u'_1 \in U_A} \sum_{i \in I} (r_{u'_1 i} + g_i - p_{u'_1} q_i)^2 \tag{4.43}$$

$$= \sum_{u_1 \in U_A} \sum_{i \in I} (e_{u_1 i})^2 + \sum_{u'_1 \in U_A} \sum_{i \in I} (e_{u'_1 i})^2 \tag{4.44}$$

$$\frac{\partial \mathcal{J}}{\partial q_i} = \sum_{u_1 \in U_A} (e_{u_1 i} p_{u_1} + e_{u'_1 i} p_{u'_1}) \tag{4.45}$$

$$= \sum_{u_1 \in U_A} \Big(r_{u_1 i} p_{u_1} - p_{u_1}^2 q_i + (r_{u_1 i} + g_i) p_{u'_1} - p_{u'_1 i}^2 q_i \Big) \tag{4.46}$$

$$= \sum_{u_1 \in U_A} \Big(r_{u_1 i} (p_{u_1} - p_{u'_1}) - (p_{u_1}^2 - p_{u'_1}^2) q_i + g_i p_{u'_1} \Big) \tag{4.47}$$

$$\tag{4.48}$$

From 4.45, suppose $\frac{\partial \mathcal{J}}{\partial q_i} > 0$, this means that new gap (g'_i) will cause an increase in the step size in the gradient descent update equations. Hence counter-polarization, which decreases the gap to $(g'_i < g_i)$, will decrease the update step size in gradient q_i.

4.6.1 Pre-recommendation Counter-Polarization - (PrCP)

Recall pre-recommendation transformation function 4.49:

$$\begin{aligned} r'_{ui} &= r_{ui} - \lambda_u \times (\bar{r} + \frac{g_i}{g_{max}}) \times \Phi_i^{\lambda_u + r_{ui}} \quad \text{if } r_{ui} \geq \delta \\ r'_{ui} &= r_{ui} + \lambda_u \times (\bar{r} - \frac{g_i}{g_{max}}) \times \Phi_i^{\lambda_u + r_{ui}} \quad \text{if } r_{ui} < \delta \end{aligned} \tag{4.49}$$

$$r'_{ui} = r_{ui} - sgn(r - \delta).\lambda_u(\bar{r} + sgn(r - \delta)\frac{g_i}{g_{max}})\phi_i^{\lambda_u + r_{ui}} \tag{4.50}$$

$$\tag{4.51}$$

Again, we make the following assumptions:

1. Perfectly polarized scenario : every $r_{u_2 i} = r_{u_1 i} + g$

2. $\lambda_{u_1} = \lambda_{u_1} = \lambda_u$

We now compute the new gap:

$$g_i' = r_{u_2i}' - r_{u_1i}' \tag{4.52}$$
$$= (r_{u_2i} - r_{u_1i}) - \lambda_{u_1}(\bar{r} + \frac{g_i}{g_{max}})\phi_i^{\lambda_{u_1}+r_{u_1i}} + \lambda_{u_2}(\bar{r} - \frac{g_i}{g_{max}})\phi_i^{\lambda_{u_2}+r_{u_2i}} \tag{4.53}$$
$$= g_i - 2\lambda_u\left[\bar{r}\big(\phi_i^{\lambda_u+r_{u_1i}} + \phi_i^{\lambda_u+r_{u_1i}+g_i}\big) + \frac{g_i}{g_{max}}\big(\phi_i^{\lambda_u+r_{u_1i}} + \phi_i^{\lambda_u+r_{u_1i}+g_i}\big)\right] \tag{4.54}$$
$$\Rightarrow g_i' < g_i \tag{4.55}$$

This means that :

1. the new gap g_i' is smaller Compared to the previous gap g_i.
2. as ϕ_i increases g_i' decreases even further to compensate for higher polarization.

To conclude, a PrCP strategy makes the gap change, $\Delta g = g_i' - g_i$, negative, thus reducing polarization; furthermore, the gap reduction Δg is monotonically decreasing with item polarization levels ϕ_i and with the user's desire for discovery, λ_u.

4.7 Chapter Summary

In this chapter, we presented a new data-driven methodology for polarization detection in a set of items rated by a population of users. Next, we investigated the possible relationships existing between the polarization score and item reviews. To do so, we performed Sentiment Analysis on the reviews written by users for each movie.

Finally, we proposed a novel approach for countering polarization in a recommender system. We started with a counter-polarization strategy that can be used regardless of the type of recommender system. Next, we proposed a novel NMF-based recommender system that can suggest a list of relevant items, while also considering several polarization scenario factors, such as a user's willingness to discover more items from the opposite polarity, the items' polarization scores, as well as the population polarization.

CHAPTER 5

EXPERIMENTAL RESULTS:
POLARIZATION IN RECOMMENDER SYSTEMS

5.1 Polarization Detection

5.1.1 Datasets

MovieLens [170] :

The MovieLens datasets are a collection of movie ratings that were put together by the GroupLens research group at University of Minnesota over the past 20 years. There are different sizes of the dataset based on number of collected ratings: 1, 10 and 20 million records. The largest set, the 20M dataset, includes about 140,000 users and 27,000 movies. The ratings are on a scale from 1 to 5 and the dataset also contains movie information, such as genre and tags. The tags are provided by the user for each movie.

Jester [171] :

The Jester dataset was put together by Ken Goldberg and his group at the University of California, Berkeley. Jester is a collection of 6 million ratings for 150 jokes. The ratings are provided by users of the developed system on the internet. Jester uses continuous ratings from -10 to 10 and has the highest ratings density, 30%, among other available benchmarks. In other words, on average, a user has rated 30% of all the jokes.

Book-Crossings [99] :

The Book-Crossings dataset is a collection of book ratings, developed by Cai-Nicolas Ziegler based on bookcrossings.com. It contains about 1.1 million ratings of 270,000 books rated by 90,000 users. In addition to the ratings, which are on a scale of 1 to 10, the dataset contains implicit ratings as well.

Amazon Books [172] :

The Amazon Books dataset contains book reviews and metadata from Amazon, including over 8 million reviews spanning May 1996-July 2014. Here, we use 5-core data where users and items have at least 5 reviews each. A rating is an integer from 1 to 5.

Netflix prize [173] :

This dataset was originally constructed for the Netflix Prize Competition which aimed at accurately predict movie ratings. It contains over 100 million ratings from over 17,000 movies that are rated by 480,000 users. The data was collected from October 1998 till December 2005. A rating is an integer from 1 to 5.

IMDb Review Dataset [174] :

This dataset was originally used for a binary sentiment classification algorithm. It contains a set of 50,000 movie reviews from IMDb. The constructed dataset contains 15 positive and 15 negative reviews for each movie, and neutral reviews are not included. The polarity dataset contains substantially more data than previous benchmark datasets, including [175].

Epinions [40, 176]:

The Epinions datasets are collected from the e-commence site, Opinions.com, where users can write about products and give a rating. The users also can evaluate other users by including them in their 'Web of Trust'. The large dataset contains about 1.5 million reviews that received over 25 million ratings by 163,634 users. The rating for both reviews and items is on a scale from 1 to 5.

We evaluate our pipeline on the aforementioned benchmarks. The available datasets have a major limitation in that several have none or few polarized items. For example [174, 175] introduced polarity datasets that contain 2,000 and 50,000 movies, respectively. However, these datasets consider only reviews and ratings associated with these reviews. This is not enough for identifying polarization since not all users provide reviews; hence, by not considering those users, we lose informative ratings. Moreover, [174] hand-picked only 30 reviews, with an even number of positive and negative reviews, instead of considering all the reviews for each movie. This data set is therefore an artifact subset of the entire data.

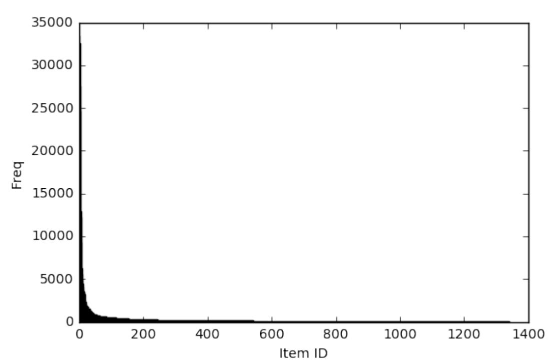
Due to the above limitations, we constructed a balanced collection of movie ratings from IMDb, by crawling polarized movies based on their histograms. After pruning and annotating the dataset, we ended up with 612 polarized movies, each with at least 50 ratings; then, we crawled all of the reviews for these movies. We also crawled an almost equal number of randomly selected non-polarized movies from different genres and years. Similar to other datasets, the movie popularity distribution follows a log-normal distribution. In the interest of providing a benchmark for future work in this area, we will release this dataset to the public.

5.1.2 Discussion

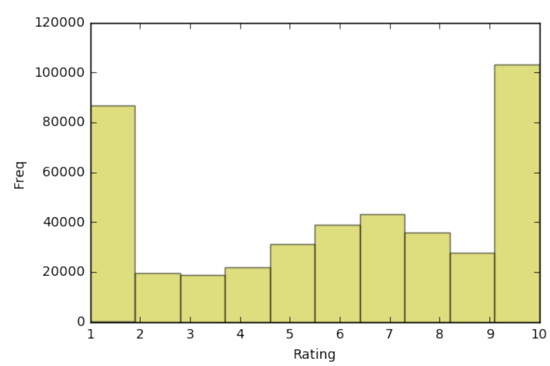
5.1.2.1 IMDb Dataset

The proposed IMDb dataset contains 1,340 movies and 427,074 ratings. The data was collected the last week of March 2017. Each movie has a rating on a scale from 1 to 10. Figure 5.1a and 5.1b show the movies' frequencies and rating frequencies, respectively.

We first build a rating histogram of each item in the datasets. Figure 5.5 shows a snapshot of a movie histogram along with 2 Gaussian distributions of our constructed dataset. Then, we label each item as polarized or non-polarized using our annotation methodology as shown in figure 5.2 (more description in Section 4.2.3). Next, we extract the proposed feature set (Sec. **??**) for the items. We report 5-fold cross validation results to make our results comparable with others in the literature. We then train our proposed Polarization Detection Classifier using the Random Forests algorithm.



(a) Item Frequency



(b) Rating Frequency

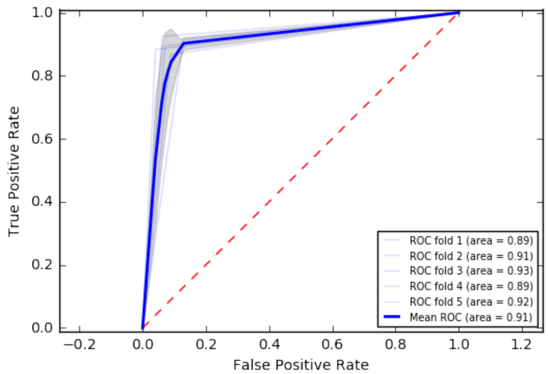
Figure 5.1: Frequency of movies in the dataset and total ratings histogram

Figure 5.2: Annotation Platform

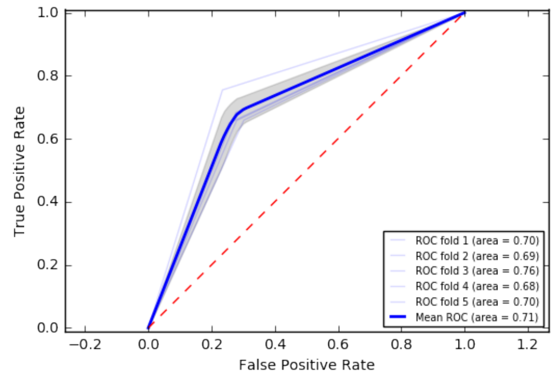
We compare our Polarization Detection Classifier (PDC) to several recent models [33, 36, 44, 104]. All of these methods use metrics that measure the polarization score for each item based on various criteria. In order to compare them with our classifier, we learned a threshold for each metric based on a validation set, and then, using this threshold, we decided whether an item was polarized or not. In addition, we tuned any additional parameters -e.g. for [40] to fully capture polarization. Figure 5.4 shows the Area Under Curve (AUC) of the Receiver Operating Characteristic (ROC) curve [177] for each method. The proposed polarization classifier achieves the highest AUC = 0.92.

We compared these two sets of features to decide which one to choose for further experiments. Figure 5.3a shows the results of the classifier using feature set 1, 2, and both of them, respectively. As we can see, considering all features (10 features) does not improve the accuracy significantly. Considering only feature set 2 decreases the accuracy by about 20%. These results verify that feature set 1 is simple, fast, and accurate for detecting polarized items, based only on how they have been rated by users.

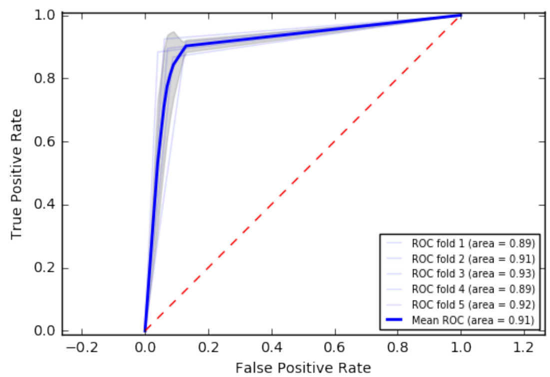
Figure 5.6 shows the Precision, Recall, and F-score [168] for the polarized and non-polarized class. PDC outperforms the other approaches significantly. To compare the total performance achieved by PDC, we also list the AUC for ROC curves in Table 5.2 which shows that PDC significantly outperforms the other methods in terms of AUC.



(a) ROC feature Set 1



(b) ROC feature Set 2



(c) ROC for both of the feature Sets

Figure 5.3: ROC comparison for the proposed feature sets

Table 5.2 shows the time taken by PDC compared to the other methods. PDC is significantly faster, even though we excluded the time needed for finding the best threshold for other polarization scores, and our approach has a training phase.

Our new approach is a valuable asset in the different domains where polarization matters, such as recommender systems. To illustrate this, we automatically quantify the polarization of several benchmark data sets for recommender systems, using our developed polarization detection pipeline to process these large data sets without the need to label them manually or to retrain the classifier.

5.1.2.2 Polarization versus Sentiment

For the review sentiment classifier, we built 13 review sentiment classifiers, one for each genre. We noticed that each genre has a different number of instances for training;

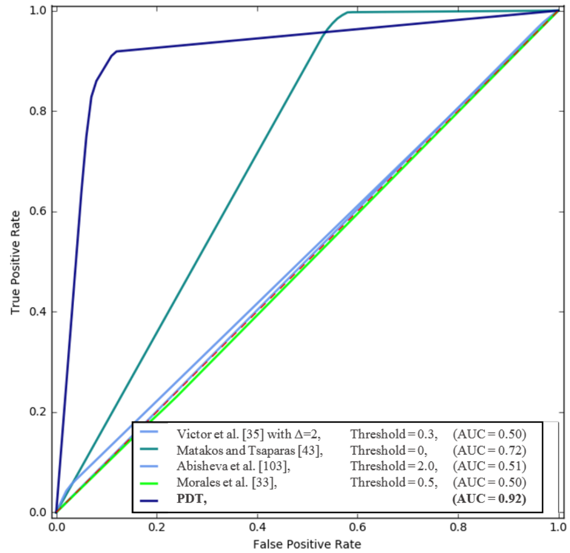


Figure 5.4: ROC comparison

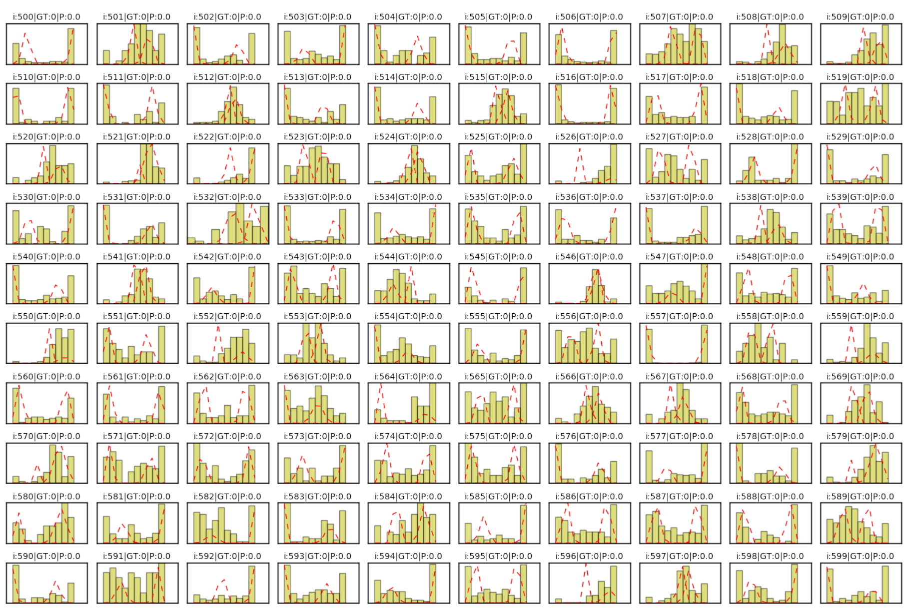


Figure 5.5: Histograms of 100 randomly selected movies from the crawled IMDb dataset

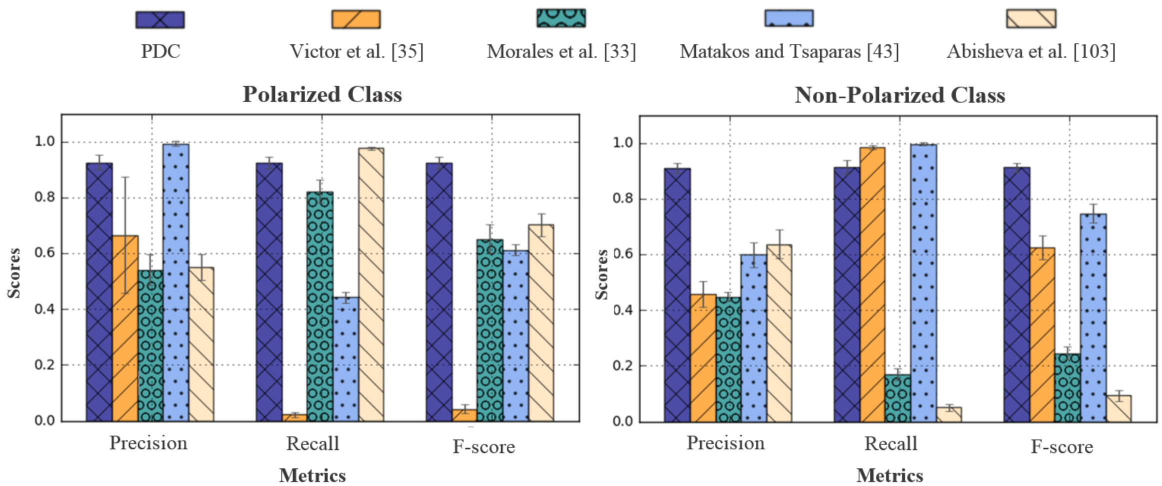


Figure 5.6: Performance of PDC compared to existing polarization metrics on polarized and non polarized data. PDC achieves a competitive tradeoff of precision and recall regardless of polarization

therefore, the accuracy varied depending on the genre. The accuracy is generally around 0.75 ± 0.15 with Area Under Curve (AUC) 0.7 ± 0.1 for all genres. However, some genres resulted in low accuracy, e.g. 'Horror' and 'Thriller'. One possible reason is that lexical knowledge gives high negative weight to words that are actually positive in horror/thriller movies. For example, "This movie is very scary!" expresses positive sentiment in horror movies, but the lexical knowledge gives a high negative sentiment to this review. As mentioned before, Valence and Arousal is highly related to the emotions of people as well. For example, 'This is a good movie' has lower arousal score than the sentence 'This is a very good movie!', meanwhile they have similar valence scores because they both express a positive feeling.

To understand the relationship between the ratings-based polarization score and review sentiments, we compute the Pearson correlation [178] between the polarization score and sentiment score of each movie. To do so, we first needed to aggregate review sentiments for each movie. We considered both average and standard deviation of review sentiments. The reason for considering standard deviation is that if a movie's sentiment has a larger standard deviation, this means that the reviews contain different sentiment scores. In other words, it shows that there is some disagreement between reviews that may indicate a higher

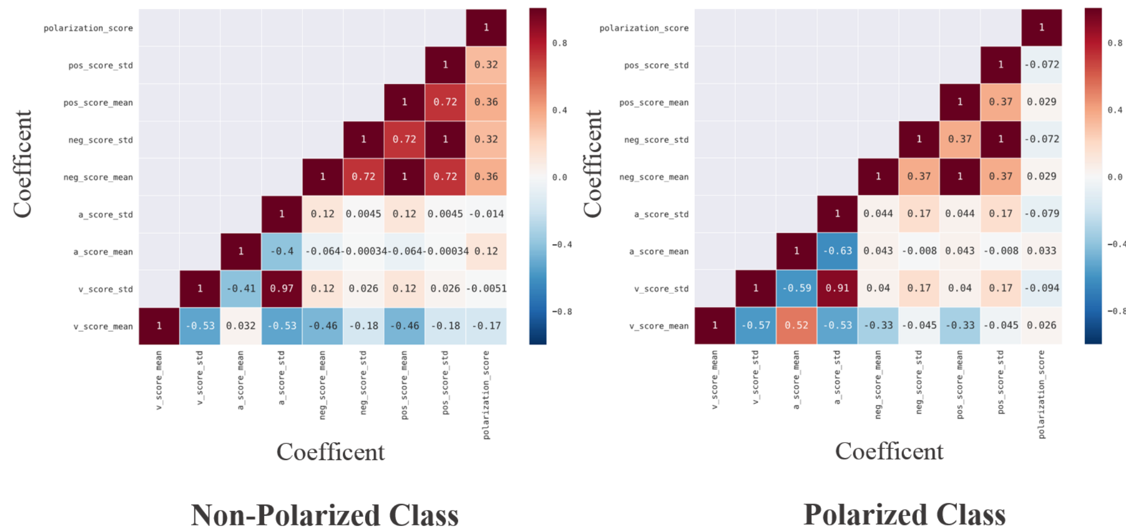


Figure 5.7: Heatmaps showing Pearson correlation coefficients between text reviews, polarization and emotion (v=Valence, a=Arousal) in polarized and non-polarized items.

polarization. However, the sentiments' disagreement does not always mean that extreme opinion (polarization) is present. It may simply show diverse sentiments regarding that movie. We investigate these assumptions in Figure 5.7, which shows different correlations for polarized and non-polarized movies. The correlation between the polarization score and review sentiments is close to zero, showing that reviews by themselves are not enough to automatically infer polarization status; and that furthermore, due to being domain-dependent, reviews are unable to fully infer polarization in generic domains. However, in agreement with the theory in [104], arousal sentiment is associated with higher levels of the polarization score, due to the hidden strong emotions in the reviews . In addition, there is a stronger correlation between the sentiments themselves and the polarization score in non-polarized movies. This is due to the fact that the non-polarized movies have many reviews (at least 100 for each movie) compared to the polarized movies. Finally, we can conclude based on our experiments and analysis, that although text-based item reviews are able to slightly point to the polarization, they are not sufficient to fully capture polarization.

5.1.2.3 Other Benchmark Data Sets

In this section, we apply our PDC pipeline on various recommender system benchmark data sets, including the Netflix Prize [62], IMDb-Polar [174], Book-Crossing [99], MovieLens (20M) [170] and Amazon books (for several random users) [172]. Table 5.1 shows the percentage of polarized items in each dataset after removing items with less than 50 ratings. As we can see, there are only a few polarized items for most datasets. This confirms the fact that the available benchmark datasets do not always fully capture the realistic distribution of polarized user-item ratings. Hence, they may not be suitable to study polarization's interplay with recommender systems. Polarization often emerges during a discussion about a controversial topic when there are two groups of individuals with extreme opinions. A typical example is the e-commerce site Opinions.com, in which users can evaluate other users by including them in their 'Web of Trust' [40]. The available benchmark on Opinions.com has been used for trust-based recommender systems mainly for recommending items that the trust network disagrees with. However, our large scale quantification of this benchmark data set shows that most of the items are diverse ratings (flat ratings histogram) rather than polarized ratings (U-shaped ratings histogram), as shown in table 5.1.

TABLE 5.1

Detecting Polarized Items in Different Domains

	Book Crossing	**Amazon books**	**Epinions**	**IMDb Polar**	**Movie Lens**	**Netflix**	**IMDb PDC**
Num. of items	1,340	5,912	103	3,581	19,344	10,524	1,340
Num. of ratings	49,317	25,230	1,561	25,000	138,493	19,847,947	427,074
Num. of polarized items	2	12	9	3,581	4	5	612

TABLE 5.2

A comparison of AUC and time complexity for different methods

	PDC	Victor et al. [40]	Morales et al. [33]	Matakos and Tsaparas [44]	Abisheva et al. [104]
AUC	**0.92**	0.5	0.72	0.5	0.51
Time (sec.)	**1.87**	63.93	3.92	3.21	5.27

5.2 Handling Polarization in Recommender Systems

As mentioned in chapter four, we first propose a pre-recommendation approach to mitigate polarization from the rating data before learning any recommendation model. This is a useful strategy since most online recommender system providers use a RS as a black box; hence, it is difficult to look into the inner workings of the algorithm to modify it. We also present an approach in the context of classical Collaborative Filtering (CF) Recommender System algorithms. We use latent factor models, specifically the Non-Negative Matrix (NMF) algorithm, to characterize both items and users on a set of factors inferred from user-item ratings patterns.

In the context of this work, our major interest is not in changing the user's rating, but rather how different recommendation algorithms and strategies affect which items a user would see (or discover) after working with a recommender system, especially in a polarized environment. We study the effect of polarization and propose several strategies to counteract the polarization effect.

In order to investigate the performance of the proposed approach, we need to know how many of the items in the opposite view are ever actually recommended to the user. This is different from catalog coverage, which considers how many of the recommended items belongs to the "long tail" of items. In this section we will take a deeper look at the view space coverage and effects of polarization on the algorithms.

5.2.1 pre-recommendation for Counter-Polarization

To empirically validate our proposed pre-recommendation schema, we first studied how factors λ_u, Φ_i, g_i would affect the mapping function from section 4.3. Figure 5.8 shows how the difference among extreme values affects the initial rating r_{ij} in a polarized environment if a user u has a high discovery factor λ. In figure 5.8(a), we assume that ratings are on a scale of 1 to 10 and that all items have the same polarization score, $\forall i \in I, \phi_i = 0.9$. *Gap* represents the difference between extreme opinions of an item. For example if $Gap_i = 2$, item i has received two diverging sets of ratings from users. Users who liked this item rated it 10,9,8,7, while those who did not enjoy the item as much had given ratings in the range of 1 to 4. So there is a *2*-gap between the given ratings; hence, the item ratings histogram looks like figure 5.9. Similarly, figure 5.8(c) indicates how the transformation affects the initial ratings for an arbitrary item i, where $g_i = 2$ and the user discover factor λ is 1. Finally, we study the effect of the user's chosen discovery factor on transforming the source data. Here, we assumed that $g_i = 2$ and that the item is polarized, with $\phi_i = 0.9$. As shown in the figures, we performed a controlled distortion of the training data from which a recommender system is learned to help the users receive more useful recommendations in the presence of polarization. By doing this, we still keep the users' preferences, yet make it more moderate so that no extreme recommendation can be generated when using a conventional recommender system algorithm.

5.2.2 Polarization-aware Recommender System

5.2.2.1 Experimental Settings

Although recommendation system providers are typically interested in the long-term effect of a proposed RS algorithm on user satisfaction, it is not always feasible. In our case, it is challenging to control parameters which may affect the polarization degree and hence user satisfaction. Conducting such experiments requires A/B testing for a couple of months to carefully study the counter-polarization effect and its consequences on the recommendation service, as well as user satisfaction [41].

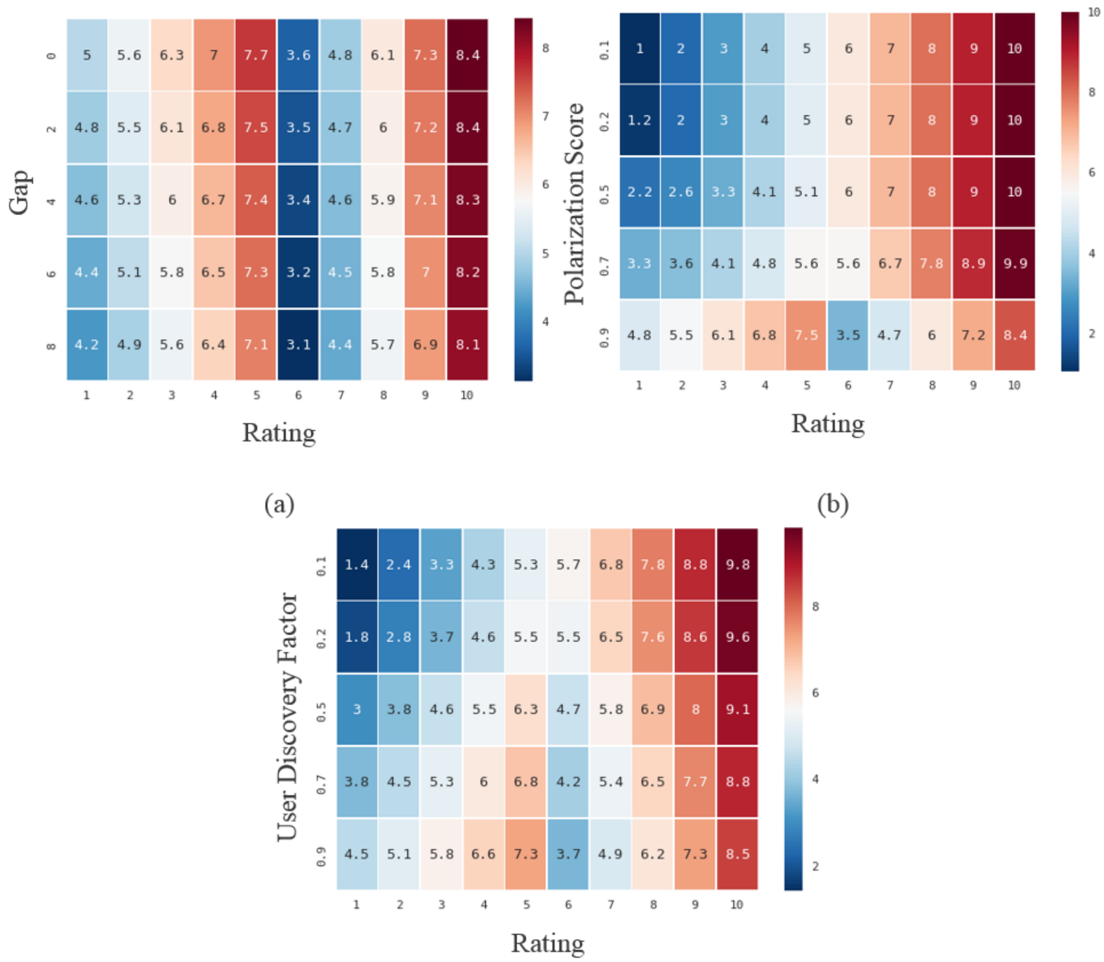


Figure 5.8: Correlation between Φ, g and rating r_{ui} in the pre-recommendation style counter-polarization approach

It is also important to consider the effect of datasets on the recommendation process. Recent experiments indicate that rating distribution characteristics have an influence on the behavior of the algorithm with respect to different measures, such as catalog coverage [41]. Hence, it is critical to fully understand a dataset before learning a model. However, information regarding collecting offline datasets is not always available. Especially in the case of interactive recommender systems, it is often hard to simulate the process of data collection [42] . Most data publishers provide information regarding the data collection process, yet there is often hidden biases which affect the recommendation process. For example, we have showed in table 5.1 that none of the available benchmarks have collected a sufficient number of polarized items, although these items exist.

For all these reasons, we study the effect of polarization on recommender systems on multiple users in a fixed environment, inspired by [31]. The fixed environment represents a source of information for recommending items to a user. We then conduct a large number of experiments on this environment to demonstrate the performance of our proposed approach compared to baseline methods.

5.2.2.2 Evaluation Metrics

We evaluate the performance of our approach in terms of rating prediction accuracy, using the Mean Squared Error (MSE) [62]. The MSE is the average of the square errors of the predicted ratings:

$$MSE = \frac{1}{m} \sum_{i=1}^{m} (r_i - r'_i) \tag{5.1}$$

where m is number of items, r_i is the true rating of item i and r'_i is the estimated rating.

As part of studying polarization, we also define a metric to evaluate different counter-polarization strategies. This metric helps us to verify whether an item from the opposite view is among the recommended items. Considering each user, if any of the items from the opposite view is included in the recommendation list, then a *hit* occurred. We define Opposite View Hit Rate (OVHR) ratio based on the ratio of the number of items from the opposite view to the total number of recommended items:

$$OVHR = \frac{Number\, of\, recommended\, items\, from\, the\, opposite\, view}{Total\, number\, of\, recommended\, items} \tag{5.2}$$

5.2.2.3 Multi-User Fixed Environment

We consider the following simple environment: Let $G = (U, I, R)$ be an environment where user $u \in U$ can rate item $i \in I$ with rating $r_{ui} \in R$ on a scale of x to y. The item could be a book, web page, news article, movie, etc. We define a recommender system algorithm as follows:

Definition 3 : *Let the number of users,* $|U| = n$ *and number of items,* $|I| = m$*. A recommender system algorithm takes environment* G *as input along with a user* $u \in U$*, and outputs a set of items* $i_1, ..., i_{k_t} \in I$*.*

Thus, given an environment G, representing which users have rated which items and a specific user u, a recommender system algorithm's output is a list of items to be recommended to u. We assume that u has to pick only one item from the recommendation list and that s/he then provides a rating r_{ui} for the selected item.

In order to analyze the polarization effect on the positive feedback loop, generated between a recommender system suggesting items for a user and the users providing ratings for the recommended items, we generate a rating environment with 50 users and 200 items. Items are evenly divided in two opposite viewpoint sets that we refer to as red items and blue items. Users are also divided into two groups based on whether they like red or blue items. These labels are purely for the purpose of analysis and they obviously do not affect the recommender system algorithms. Each user $u \in U$ rates half of the items of I, in such a way that the rating r_{ui} is greater than ζ if s/he likes item i, and less than ζ if s/he does not like it. This process forms environment G. We also assume that users are rational and are truly expressing their preferences with ratings on a scale of 1 to 10. For concreteness, we assumed $\zeta = 5$. In order to make the environment polarized, we assume that user $u_a \in GroupA$ likes red items more than blue ones, and hence all of his/her ratings for the red items are higher than all of his/her ratings for the blue ones. Similarly, we assume that user $u_b \in GroupB$ likes blue items more than red ones and hence all of his/her ratings for the blue items are higher than all of his/her ratings for the red ones. Finally, we generated environment G with different values of Gap and user's discovery factor.

We follow the same steps that were presented in Algorithm 1, for the classical NMF recommendation algorithm, to apply our proposed counter-polarization strategies. In order to understand the Interactive Recommender System (IRS), we start by showing some experiments that illustrate examples of how such a system works in environment G. In all of the examples, we set the number of factors in the latent space, k_f, to 5 and we compute the

Figure 5.9: Rating histograms of the items in environment G with polarization ratio 0.25

list of top $k_t = 5$ items to be recommended to each user. The user will give a rating for only one of the selected items and we take this rating value from the true source of ratings, i.e. the ground-truth data. We repeat this procedure 100 times (there are 100 unrated items for each user) to simulate an interactive recommender system scenario. In each iteration, we measure MSE from the training and testing phases. We also keep track of the items that a user decided to react by providing a rating.

Figure 5.10 shows traces from the interactive recommendation system for user $u \in GroupA$, which means s/he likes red items more. We generate environment G considering that gap is 2. This means that $7 \leq r_{ui} \leq 10$ if $u \in GroupA$ likes item i while $1 \leq r_{ui} \leq 4$ if $u \in GroupA$ does not like item i. Figure 5.9 shows the rating histogram of items and we can clearly see that the difference between the range of the two sub-populations of ratings given to an item is 2. Figure 5.10 shows that a classic state of the art recommender system, in our case NMF, is always going to recommend red items, to which the user had previously shown more interest. Although the red items are relevant, the user *Red* is trapped in a filter bubble

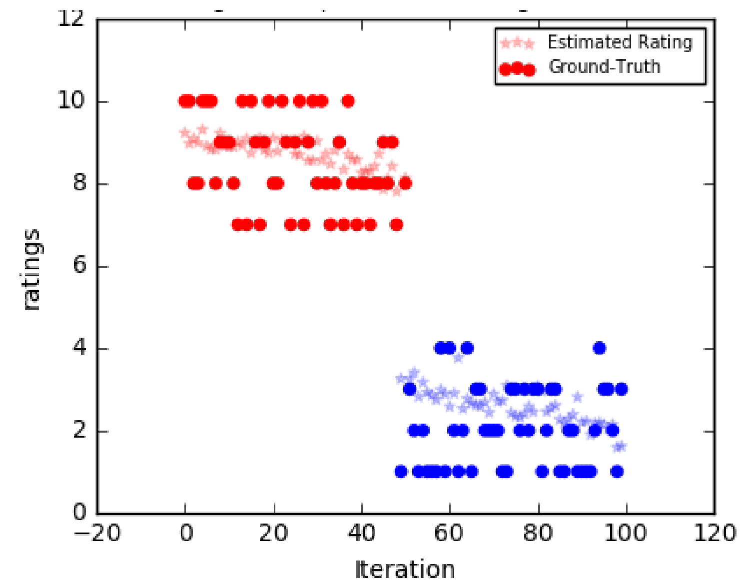


Figure 5.10: Traces of the interactive recommendation process for user u_a who had liked red items more than blue items

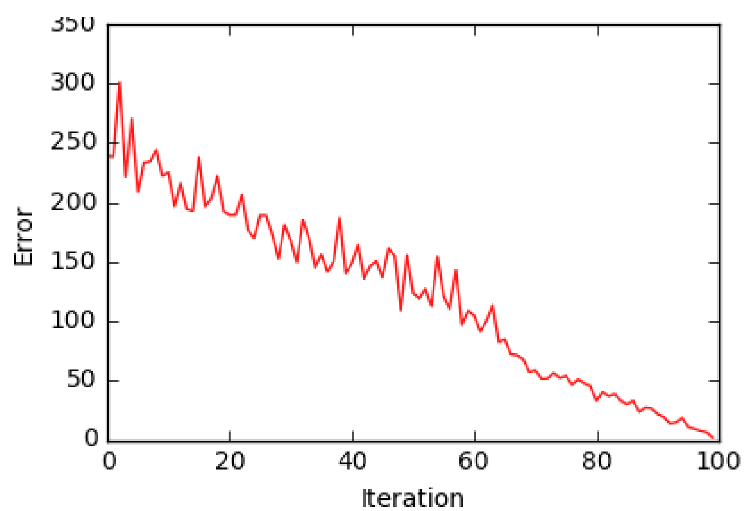


Figure 5.11: Traces of the interactive recommendation process for user u_a who had liked red items more than blue items

that does not allow him/her to explore any items from the opposite color/view, at least not before the user has seen all of the *Blue* items, the number of which may be enormous in a real life setting. This finding is in line with finding in most of the literature [179, 180] including what Dandekar et. al have proved mathematically for an over-simplified theoretical scenario with simpler CF strategies in [31]. Figure 5.11 shows the testing MSE error for user u_a. MSE decreases as the user provides ratings in each iteration; hence, there are fewer unrated items for the user.

Figure 5.12 shows a very interesting “pattern” in the NMF model’s convergence when using gradient descent optimization, where each line (or colored thread) represents the decrease of the objective function for an iteration of the interactive recommendation process. **One interesting question that arises is whether this peculiar pattern might be due to the fact that the environment G is polarized?** We will next investigate this possibility by learning NMF models in several environments with varying polarization rates. **The answer to this question is critical because it might indicate that the simple task of monitoring the convergence trends of the objective function of an interactive recommendation system, may hold the key to detect emerging polarization early on during the interaction between the user and algorithm.** Figure 5.13 provides an answer to this question, where we can see the contrast between the objective function evolution for varying polarization rates, compared to non polarization. There is a clear pattern (namely a bump followed by a plateau), in the decrease of the objective function, that is associated specifically with polarization. Furthermore the error converges to a smaller value for higher polarization. We interpret this algorithmically, by the fact that the higher the polarization in the ratings, the larger the gap (extreme likes and dislikes) between the items’ ratings in the opposite viewpoints. Hence increased polarization leads to increased separation between the opposite viewpoint ratings, which, very naturally makes learning the ratings an easier task from a machine learning perspective. This is analogous to improved class separability in learning class boundaries, which naturally make learning a model easier. **Although we have given a natural explanation for the model’s lower reconstruction error (hence better prediction) for higher polarization, this is still surprising (when taken in the context of learning in polarized environments), and disappointing (machine learning algorithms find it easier to learn discriminating models in polarized environments: The models will quickly learn to keep each user in the safety of their preferred viewpoint, essentially, giving rise to filter bubbles)**.

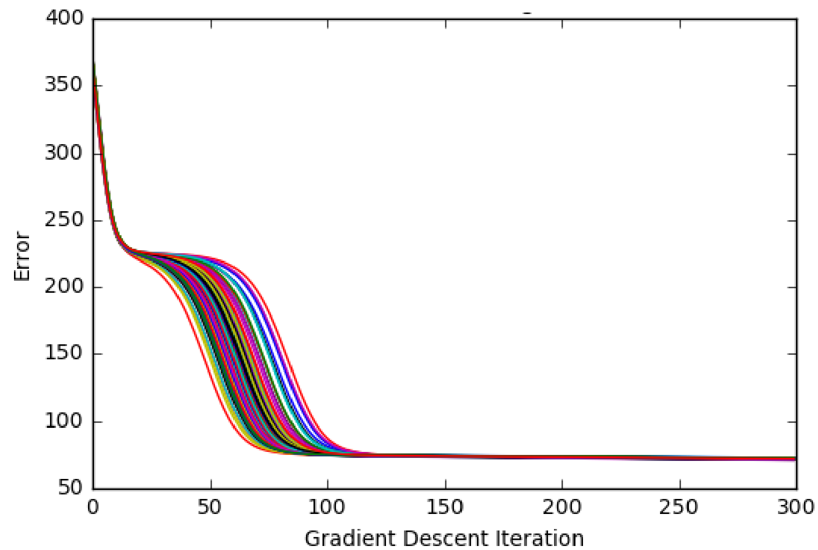


Figure 5.12: Traces of the interactive recommendation process for user u_a who had liked red items more than blue items

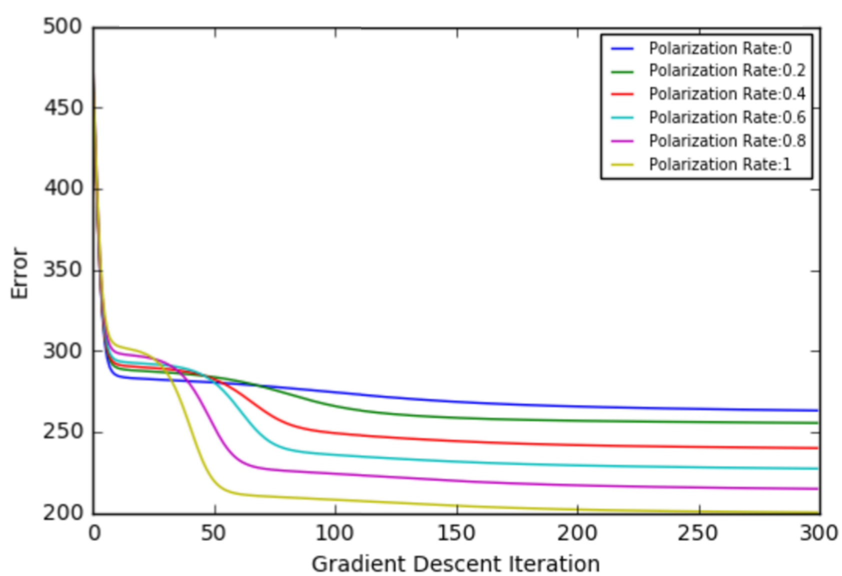
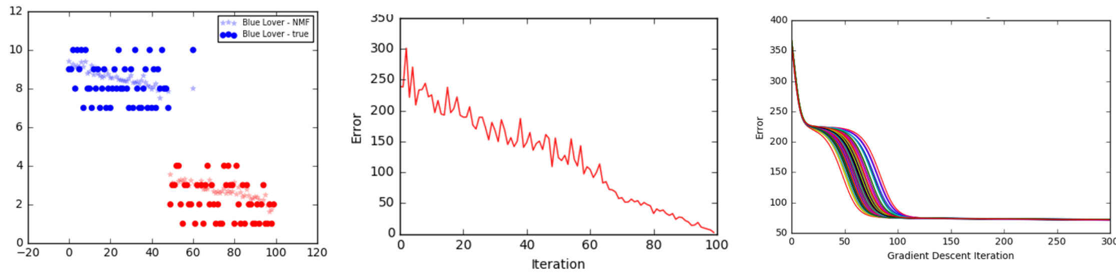
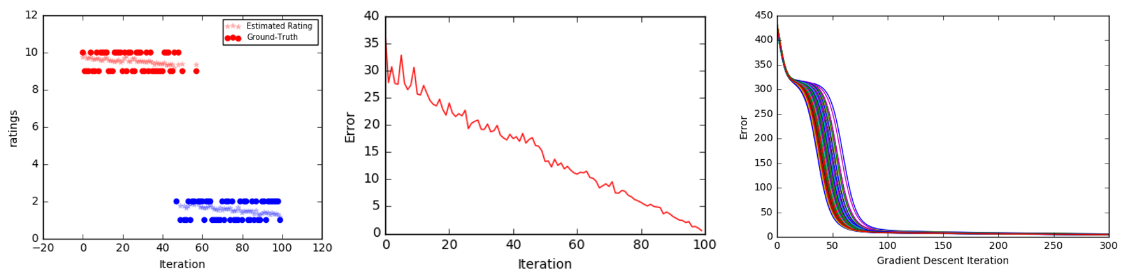


Figure 5.13: Traces of the NMF model's gradient descent convergence plot for several populations with different polarization degrees



(a) (b) (c)

Figure 5.14: Traces of the interactive recommender system for user u_b in environment G with polarization ratio $= 0.8$



(a) (b) (c)

Figure 5.15: Traces of the interactive recommender system for user u_a in environment G with polarization ratio $= 0.8$

We repeat the same experiment for user $u_b \in GroupB$ who likes blue items more than red items. Figure 5.14 shows the same pattern as user u_a which indicates that regardless of user groups and item types, the positive feedback loop in a polarized environment creates a filter bubble which prevents the users from exploring opposite views. In fact, the polarization causes severe consequences in increasingly polarized environments, as shown in figure 5.15 .

To make a more comprehensive evaluation of performance of the proposed counter-polarization approaches, we repeat the experiment with varying the parameter λ for the proposed counter-polarization methodology. We consider two scenarios:

(a) All users have the same λ, i.e. $\lambda_u = c \quad \forall u \in U$, where c is a constant $\in [0, 1]$.

(b) User u has his/her own unique $\lambda, \lambda_u = c_u$ for user u and $\lambda_u = 0 \quad \forall u \in U - u$, where c_u $\in [0, 1]$, is a user defined constant

Scenario (a) assumes that all users have same discovery factor which indicates a more unified population regarding exploring the item space. While Scenario (b) assumes only the

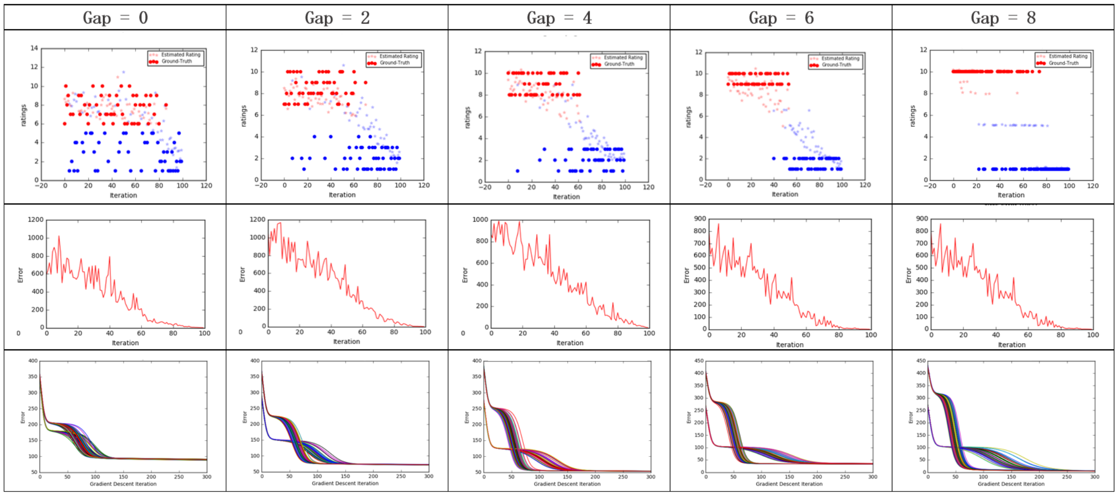


Figure 5.16: Traces of the Interactive Recommendation process when applying the pre-recommendation counter polarization (PrCP) strategy for user u_a

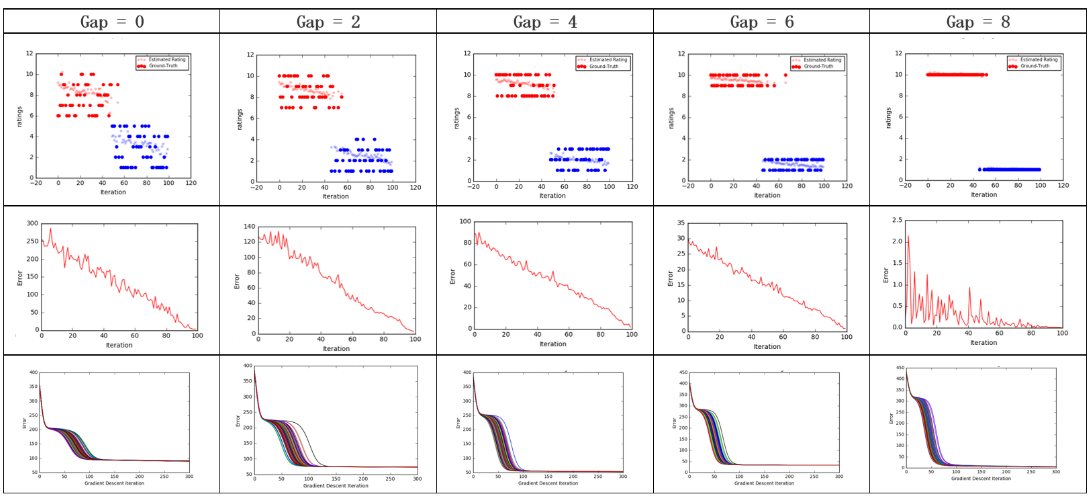


Figure 5.17: Traces of the Interactive Recommendation process with the classical NMF-based CF recommendation algorithm for user u_a

current user u can modify his/her user discovery coefficient λ_i and the rest of the users have no interest in discovering new items, which means their λ is set to 0. The intuition behind this experiment is to study the effect of a user population on recommending items to a single user and to all users.

We run the experiments for different $\lambda \in [0, 0.2, 0.5, 0.7, 1]$ in environment G with $gap = 2$. Then, we compute MSE_{test}, MSE_{train} and $OVHR$ for each user and then take an average over all 50 users. In order to have a comprehensive comparison, we compute

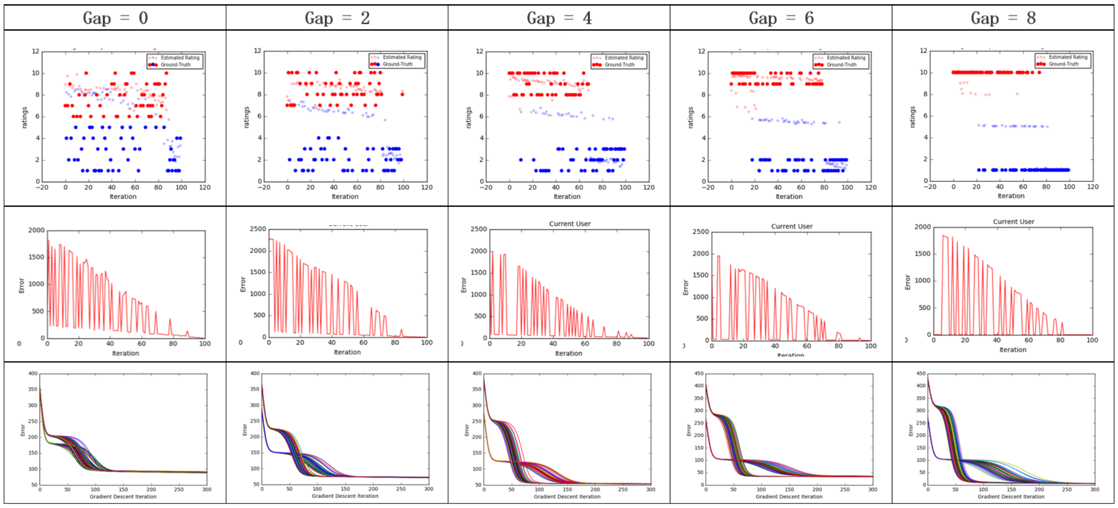


Figure 5.18: Interactive Recommendation process with the polarization-aware recommender system - (PaRIS) for user u_a

$OVHR$ in two ways:

(a) $OVHR_u$: Compute the ratio of number of items from the opposite view to what the user has picked from the recommendation list.

(b) $OVHR_{t_k}$: Compute the ratio of number of items recommended to the user from an opposite view.

Table 5.2 shows that the effects of the two metrics strongly vary depending on the chosen recommendation algorithm and strategy. Trends in varying d, show that the higher the user-defined parameter λ, the more she will be recommended items from the opposite view, as desired by the user. The ability of the user to tune the degree of discovery into the opposite viewpoint is an important feature in a polarization-aware recommender system because it allows the user to make decisions about hisher exploration space. This also adds to the transparency of a RS algorithm.

When comparing the traditional NMF-based RS with our polarization-aware RS, we see that the traditional NMF-based algorithm achieves good accuracy in rating prediction, yet it is not able to recommend any item from the opposite view. In contrast, both of our polarization-aware recommender system approaches are able to recommend a set of items which are a combination of traditional relevant single-view suggestions as well as items from the opposite view. Both the pre-recommendation scheme that can be added to the traditional NMF-based RS and the Polarization-Aware RS recommend significantly more items ($p \leq 0.05$) from the opposite view compared to the baseline approach, for all the degrees of user-defined discovery factors. These differences between different recommendation processes would go unnoticed if only accuracy measures were considered.

From a different perspective, the polarization-aware recommender system has higher $OVHR_u$ and $OVHR_{k_t}$ compared to the NMF-based RS with the pre-recommendation scheme. One reason behind this, is that we limit the probability of applying the pre-recommendation approach, p, by the maximum value of 0.5. This probability decreases as the users interact with the system. In other words, the probability that the user sees more items from the opposite view is high in the early stages of the interactive recommendation process; then it drops as the system receives more feedback from the user and updates its model. We assume that the user will pick a random item from the recommended list, regardless of the color. p is a flexible parameter that leads to more discovery if it is close to 1. Figure 5.19 shows the results if we define the probability of applying the pre-recommendation strategy, p, for user u as $p = \frac{|UI_t|}{|UI|}$, where UI is a set of items that have not been rated by user u and UI_t is the set of items that have received ratings from user u so far, during the interactive recommendation process. As we can see from this figure, the user sees more blue items. The fluctuation in the MSE plots is due to the fact that this strategy attempts to estimate the distortion rating rather than the true rating. This means that the error would be high for the extreme ratings since the pre-recommendation approach transforms ratings to less extreme ratings, while the error is lower for moderate ratings since this approach does not change those too much, so these ratings are closer to the true ratings.

One may assume that having the same user discovery factor for all individuals in the population would result in higher opposite view hit rates, since all users are interested in discovering new items and hence the entire population would be less extreme. However, our experiments reveal a different story. Table 5.3 shows that having the same user discovery

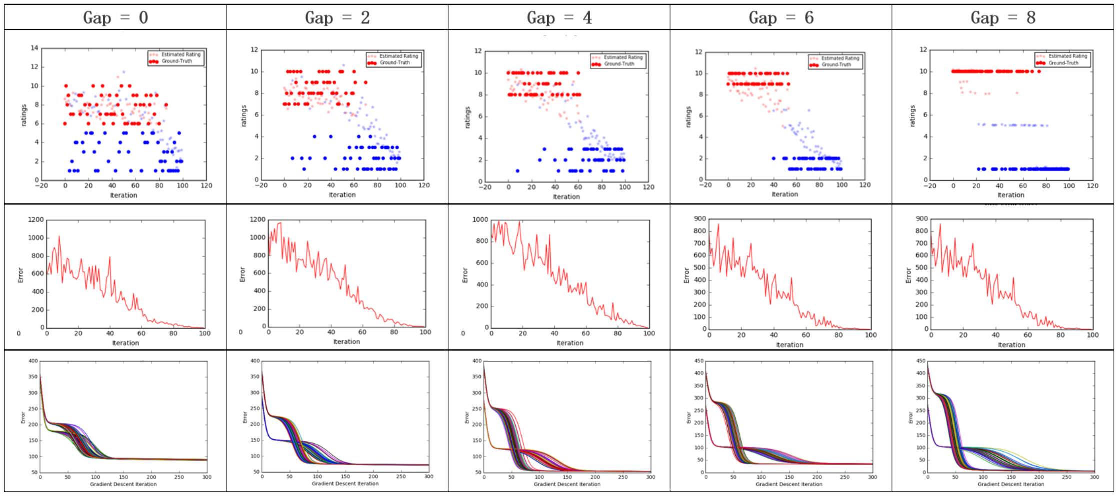


Figure 5.19: Traces of the Interactive Recommendation process when applying the pre-recommendation approach, without limiting p, for user u_a

factor for all users has less effect compared to increasing the user discovery factor for a specific user. It shows that having an enthusiastic population does not always result in counter-polarization. This effect is even more severe in the polarization-aware strategy where the users do not see any item from the opposite view even when the user population has $\lambda_u = 0.5 \quad \forall u \in U$.

Finally, by looking at the number of recommended items over time, we can see that the proposed counter-polarization methodology succeeds to cover items from the opposite view after a few iterations and broadens the viewpoint spectrum even faster if the user is more interested in discovering items from different viewpoints. Endowing the user with the ability to interactively and adaptively control the breadth of viewpoints offered by our polarization-aware recommendation algorithm is an important capability. **One consequence of this capability is empowering the users who are nowadays increasingly and unconsciously entrapped in algorithmic filters over which they have had no control. To our knowledge, this feature which allows humans to remain in or regain control of algorithm-induced filter bubble traps, has heretofore not been allowed or engineered in existing information filtering systems, whether on social media, e-commerce or e-learning recommender systems.**

TABLE 5.3

Comparison of the counter-polarization methodologies with the classical NMF-based Recommender system in terms of accuracy and opposite view ratio ($OVHR_u$,$OVHR_{t_k}$)

			Opposite View Ratio		MSE_{Train}	MSE_{Test}
			$OVHR_u$	$OVHR_{t_k}$		
			mean, std	mean, std	mean, std	mean, std
Classic NMF			0.0 ± 0.00		22.02 ± 5.27	138.96 ± 12.55
PrCP	Scenario (a)	$\lambda_i = 0.2$	$4.8\% \pm 0.06$	$25.0\% \pm 0.035$	126.57 ± 38.13	807.30 ± 70.51
		$\lambda_i = 0.5$	$4.8\% \pm 0.07$	$28.0\% \pm 0.41$	122.38 ± 37.16	805.33 ± 71.77
		$\lambda_i = 0.7$	$5.0\% \pm 0.06$	$2.9\% \pm 0.21$	120.14 ± 34.40	800.23 ± 64.91
		$\lambda_i = 1.0$	$5.0\% \pm 0.08$	$3.1\% \pm 0.048$	120.31 ± 34.96	789.89 ± 66.54
	Scenario (b)	$\lambda_i = 0.2$	$5.4\% \pm 0.073$	$4.9\% \pm 0.021$	123.92 ± 36.76	813.01 ± 36.76
		$\lambda_i = 0.5$	$6.2\% \pm 0.075$	$5.2\% \pm 0.042$	122.56 ± 39.081	804.01 ± 75.88
		$\lambda_i = 0.7$	$7.0\% \pm 0.075$	$5.4\% \pm 0.033$	120.97 ± 35.19	803.65 ± 64.65
		$\lambda_i = 1.0$	$6.8\% \pm 0.064$	$5.8\% \pm 0.03$	119.76 ± 34.93	801.86 ± 65.07
PaRIS	Scenario (a)	$\lambda_i = 0.2$	$0.0\% \pm 0.00$	$0.0\% \pm 0.00$	57.63 ± 14.35	223.31 ± 32.14
		$\lambda_i = 0.5$	$0.0\% \pm 0.00$	$0.0\% \pm 0.00$	146.51 ± 41.73	632.21 ± 57.73
		$\lambda_i = 0.7$	$5.2\% \pm 0.64$	$6.4\% \pm 0.24$	228.31 ± 66.45	1082.35 ± 79.95
		$\lambda_i = 1.0$	$48.0\% \pm 0.17$	$32.0\% \pm 0.12$	383.80 ± 108.52	2015.85 ± 167.46
	Scenario (b)	$\lambda_i = 0.2$	$5.4\% \pm 0.073$	$12.32\% \pm 0.31$	123.92 ± 36.76	813.01 ± 36.76
		$\lambda_i = 0.5$	$6.0\% \pm 0.08$	$18.1\% \pm 0.21$	124.46 ± 37.29	299.82 ± 76.01
		$\lambda_i = 0.7$	$61.0\% \pm 0.17$	$31.0\% \pm 0.167$	209.73 ± 59.53	967.103 ± 145.92
		$\lambda_i = 1.0$	$67.0\% \pm 0.24$	$68.0\% \pm 0.24$	361.77 ± 102.74	1883.50 ± 237.83

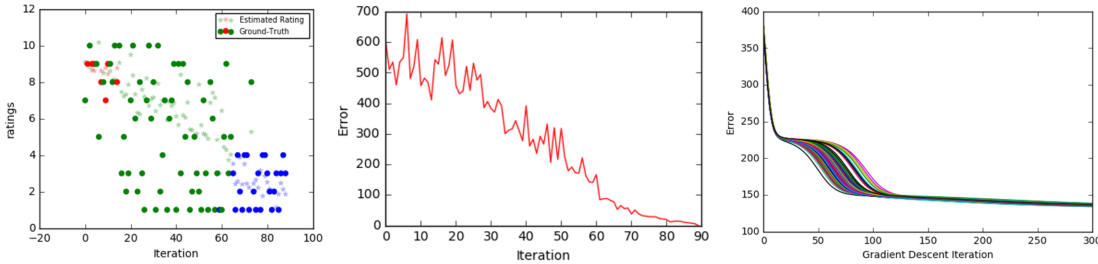
We repeated the polarization effect experiment in another environment G', where there are three types of items, *RED, GREEN and BLUE*. we define *Group A* and *Group B* as follows:

Figure 5.20: Rating histograms of the items in environment G'

$$
\begin{aligned}
&(1)\quad \text{if user } u \in \text{ Group A}: \begin{cases} 7 \le r_u i \le 10 & \text{for } i \in I^{RED}, \\ 1 \le r_u i \le 10 & \text{for } i \in I^{GREEN}, \\ 1 \le r_u i \le 4 & \text{for } i \in I^{BLUE} \end{cases} \\
&(2)\quad \text{if user } u \in \text{ Group B}: \begin{cases} 7 \le r_u i \le 10 & \text{for } i \in I^{BLUE} \\ 1 \le r_u i \le 10 & \text{for } i \in I^{GREEN} \\ 1 \le r_u i \le 4 & \text{for } i \in I^{RED} \end{cases}
\end{aligned}
\tag{5.3}
$$

Figure 5.20 shows the item rating histogram of environment G' where $|I^{RED}| = |I^{BLUE}| = 50$ and $|I^{GREEN}| = 50$.

Figure 5.21 shows the results of the interactive recommender system framework using the traditional NMF-based algorithm. Figure 5.21 demonstrates that although there are some items that are not polarized, the RS algorithm is not able to recommend items from the opposite view. However, the algorithm does recommend relevant items including



(a) (b) (c)

Figure 5.21: Traces of the interactive recommender system for user u_a in environment G'

the unpolarized (green) items which shows the superiority of the NMF algorithm in rating prediction.

CHAPTER 6

PROPOSED METHODS : HASHTAG RECOMMENDATION

6.1 Introduction

In this chapter, we start with a proposed methodology for hashtag recommendation. In order to recommend hashtags, we propose a general cross-domain hybrid recommendation model. To do so, we utilize a combination of the tweets textual content in a latent factor space, learned by MF, and a hashtag similarity graph.

We finish this chapter by discussing research that is centered on active learning and polarization. We dissect the most common active learning strategies assumptions and point to potential weaknesses that result from polarization. We conclude with proposed research questions for several leading assumptions.

6.2 HashTag Recommendation

Based on our review of hashtag annotation literature in section 2, we can observe that existing approaches work in the original space. However, our review also concluded that recommendation methods that exploit a latent factor space lead to more accurate recommendation prediction. In the latent factor approach, a user or item can be represented by a set of latent factors using matrix factorization (MF) techniques [62].

Due to this and also in order to handle the problem of sparsity, poor semantics and high dimensionality in the original bag of word spaces of tweets, we aim to design a recommender system that goes beyond raw user and item level information and represents them in the latent factor space. In this work, we propose a novel approach which computes tweet similarity within a common latent factor space that is learned using both tweet content and hashtags. Furthermore, we improve the recommendation system using a hashtag similarity graph in order to support better ranking.

6.3 Problem Statement

Given a tweet $t \in T$, a hashtag recommender system should recommend a ranked hashtag set $\{h_1, \cdots, h_f\}$ where $h_i \in H$ according to 1) the relevance to the content of tweets, and 2) the relevance of the hashtags themselves. H is a set of existing hashtags extracted from the past (training) data, hence this algorithm does not aim to recommend new non-existing hashtags.

We propose a novel hashtag recommendation system based on the content of each tweet. This approach mainly consists of two steps: 1) Candidate Tweet selection: retrieve the most similar tweets featuring hashtags from the data set. 2) Hashtag Ranking: extract hashtags contained in the top-n similar tweets and select the top-k hashtags based on a ranking method. Figure 6.1 displays a flow chart of the proposed approach.

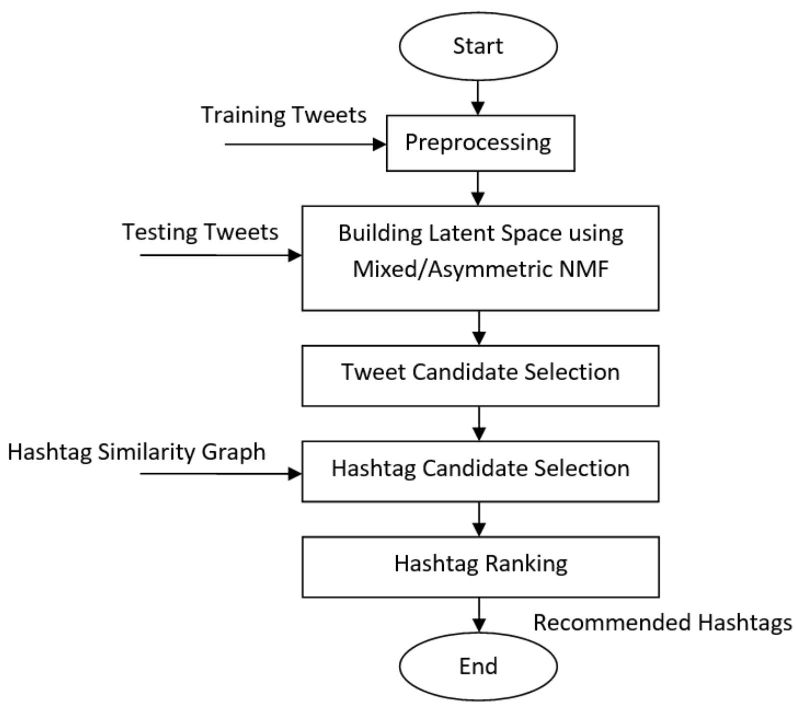


Figure 6.1: Flowchart of proposed hashtag recommendation algorithm.

6.4 Candidate Tweet Selection

In order to recommend a set of hashtags to the current unseen tweet t, first a set of similar tweets to the current one is selected. This selection is performed based on the similarity between tweet t and the rest of the training tweets in the latent space. Different text similarity measures can be used, such as Cosine similarity, Dice Coefficient, Jaccard coefficient and Levenshtein distance [181, 182]. For this step, we use the cosine similarity based on TF-IDF weighted vectors. In our context, v_i is the vector representing a test tweet and v_j is a vector representing a tweet from the past considered as the training set:

$$\cos(v_i, v_j) = \frac{(v_i, v_j)}{||v_i||||v_j||} \tag{6.1}$$

We compute the vectors using the term frequency-inverse document frequency (TF-IDF) weighting scheme as given by 6.3. TF-IDF first computes the relevance (IDF) of each term compared to the whole tweet corpus. 6.2 defines this relevancy where t_i is the given term, $|D|$ is the total number of tweets and $n(t_i)$ is the number of tweets containing t_i:

$$IDF(t_i) = log\frac{(|D|)}{n(t_i)} \tag{6.2}$$

$$TF - IDF(t_i, d) = TF(t_i, d).IDF(t_i) \tag{6.3}$$

In this work, we build a semantic text representation using Matrix Factorization (MF). In order to capture both words and hashtag domains, we apply both Mixed NMF and Asymmetric NMF [65]. The first model considers all data domains simultaneously while the latter tries to capture each domain at each step and attempts to transfer information from one domain to another. Asymmetric NMF has been recently used in different information retrieval and automated annotation systems [65].

6.4.1 Mixed NMF Multimodal Hashtag Recommendation

This approach constructs a multimodal training matrix consisting of both words and hashtags of the training tweets [65]. Mixed NMF considers both domain under $V = [V_w V_h]$

with $V_{(w\times n)}$ and $V_{(h\times n)}$ containing words and hashtag terms, respectively. NMF factors the matrix as follows:

$$V_{(w+h\times n)} = W_{(w+h\times k)}.H_(k \times n) \tag{6.4}$$

Where W is the basis of the latent space in which each multimodal object is represented by a linear combination of the k columns of W.

The next step for the hashtag recommendation system is to manage tweets with partial information, i.e. tweets without any associated hashtags. Matrix $Y_(w \times n')$ represents the n tweets without any hashtags. Using the same basis, these tweets are mapped to the latent space constructed before. To do so, we need to find $H_{(k\times n')} > 0$ which satisfies the following equation:

$$Y_{(w\times n')} = W_{(h\times k)}.H_{(k\times n')} \tag{6.5}$$

As mentioned before, to compute the new basis factors, we use the same basis from the last step. In order to find the factor matrix H, we apply a modified version of the NMF algorithm to update only H. The NMF algorithm finds a basis which is able to represent clusters of either multimodal or unimodal objects. As a result, it enables us to map both of them to the same latent space. The coefficients in a column of H express the contribution and value of membership of each word to the cluster. These clusters in the basis matrix can be used to find associated hashtags that best describe the original matrix containing the tweets. This can be done by matching the tweets without hashtags (test set) with tweets containing hashtags (training set). To do so, we first compute the similarity matrix between the test and training tweets as follows.

$$S_{n'\times n} = H_w.H \tag{6.6}$$

where H represents the training tweets in the latent space and is computed by solving 6.5. The similarity is computed based on the cosine similarity of the TF-IDF bag of words representations of the tweets, explained earlier in this section.

6.4.2 Asymmetric NMF Multimodal Hashtag Recommendation

Mixed NMF assumes an equal importance for both hashtags and words in order to decompose the multimodal information. However, based on [65], an asymmetric approach (Asym NMF) can also be used to construct the latent space which transfers information from the richer domain to the other. Inspired by this approach, we first use hashtag domain of the training tweets as TF-IDF terms. The hashtags, which are the ground truth, are extracted from the tweets. The reason behind choosing the hashtags for the first step is that a hashtag generally carries more reliable information about a tweet compared to the raw text, when it is present, and this makes the hashtag domain richer and more informative. Using NMF, the hashtag matrix $V_{(h \times n)}$ is decomposed as follows:

$$V_{h(h \times n)} = W_{h(h \times k)}.H_{h(k \times n)} \tag{6.7}$$

In our context, the vectors of the basis matrix W_h are correlated in k latent semantic topics or clusters. After this step, the semantic representation for all words in the data set is codified in the matrix H_h. To do so, we need to find $W_{w(w \times k)}$ based on the modified version of the NMF algorithm, which updates W while H remains fixed:

$$V_{w(h \times n)} = W_{w(w \times k)}.H_{w(k \times n)} \tag{6.8}$$

Then similar to the mixed NMF approach, the test tweets are projected on the semantic space using the multimodal basis that was learned before, as follow:

$$Y_(w \times n') = W_{w(w \times k)}.H_{h(k \times n')} \tag{6.9}$$

Where, Y contains the test tweets containing only words, $W_{w(w \times k)}$ is the output matrix obtained by solving 6.8 and $H_{h(k \times n')}$ is the corresponding tweets in the latent space. Finally, the cosine similarity is computed between the test and training tweets in the latent space based on $H_{h(k \times n)}$ and $H_{h(k \times n')}$, where the matrices are based on training and test sets, respectively. We compute the cosine similarity matrix using 6.1, which can also be done in matrix form for all tweet pairs as follows:

$$S_{(n' \times n)} = H_{h(k \times n')^T}.H_{h(k \times n)} \tag{6.10}$$

Once the similarity matrix, S, is ready, we can select the candidate tweet set. To do so, first the similarity matrix, S, is sorted for each row in descending order, and then the first K_n columns are selected. By doing this, we select the top K_n most similar training tweets to the test tweets. After performing the candidate tweet selection, we need to find relevant hashtags based on a ranking method which is described in the next section.

6.5 Hashtag Selection

So far, a set of candidate tweets has been selected based on cosine similarity in the semantic space. The selection of the hashtag recommendation candidates is a crucial part as only K_h hashtags are selected to be assigned to a tweet or shown to the user who is creating a new tweet. As shown in [183, 184] a set of $5-10$ suggestions or options is the most appropriate number for K_h considering the fact that short-term memory is limited, making it harder for a user to choose from more than this range. Therefore we select the top 5 hashtags for recommendations for a tweet.

In order to find these 5 hashtags for each tweet, we propose a method for ranking and selecting the recommendation of hashtags. The proposed approach for this part consists of two main steps:

1. Find the relevant hashtags based on the hashtag graph capturing related hashtags.
2. Rank the hashtags and select the top 5 for each tweet.

6.5.1 Step 1: Graph-based relevant hashtag selection

In order to obtain a set of relevant hashtags for each tweet, we build a hashtag graph. The core idea is to capture relevance among the hashtags in the data. The basic assumption behind this is that hashtags that are used together frequently are related and close in meaning. We create this relevancy graph based on either similarity or co-relevance of the hashtags. The first graph captures how similar the hashtags are while the latter captures the relationships between the hashtags based on their co-occurrences.

In order to build the similarity graph, the cosine similarity between tweets which use the same hashtags is computed. In other words, the weight of the edge between two hashtags is the average cosine similarity using TF-IDF weighted vectors of tweets which contain the mentioned hashtags. On the other hand, in the co-occurrence graph, an edge between two hashtags is weighted in proportion to the count of tweets which contain both of these hashtags at the same time.

To be more illustrative, Figure 6.2 shows a graph built from a tweet dataset[1] (described more in 7) in which a term like #obama is linked to #election. Due to the use of both terms in many tweets, the weight of the edge is 0.89. As we can see, the nodes which are connected share more relevant tweets, thus, they are more likely to occur together in the same tweet. We will use the hashtag network to improve the hashtag selection step.

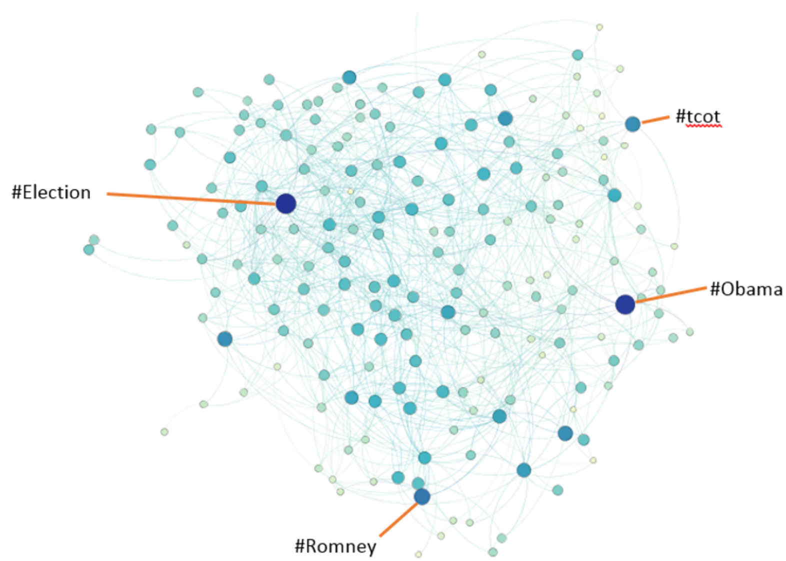


Figure 6.2: Cosine similarity-based hashtag graph.

After finding the candidate tweet set, in addition to the hashtags used in the current candidate tweets, we find the similar hashtags based on either the cosine similarity graph or co-occurrence graph. Basically, to build the recommendation candidate hashtag set, for each hashtag used in S, all the connected hashtags in the graph are also considered. In this case, the candidate hashtag set is expected to be enriched with more relevant hashtags.

[1] We used Gephi: https://gephi.org/

6.5.2 Step 2: Hashtag Ranking

After generating the candidate hashtag set, a ranking approach is performed on the set. We used Recommendation Popularity Rank [185] which ranks the hashtags based on their counts of occurrence within the recommendation candidate hashtag set selected in Step 1 above. This method is based on the fact that similar tweets tend to contain the same hashtags. Therefore a hashtag with higher occurrence frequency is more likely to be used for the current tweet. Finally, the hashtag recommendation system recommends the top highly ranked hashtags for each coming tweet.

Algorithm 6.1 and Figure 6.1 illustrate the proposed approach.

Algorithm 6.1 Hashtag Recommender System.

Input: training tweet set V, test tweet set Y, NMF algorithm f (NMF,Mixed NMF or Asymmetric NMF), hashtag graph G (either similarity or co-occurrence based)
Output: recommend hashtags R

Factor V into W and H using NMF algorithm f
Use W to factor Y into W and H' using NMF algorithm f
Compute cosine similarity matrix (S) using H and H'
Sort S in descending order
Select first K_n column of S as the candidate tweets set (T)
Update $T = T \cup \{y\, if\, there\, is\, a\, link\, between\, x\, and\, y\, in\, G, \forall x \in T\}$
Rank T based on frequency in descending order
Choose R = top K_n terms from T
return R

6.6 Chapter Summary

In this chapter, we propose a general cross-domain hybrid recommendation model. This model uses the actual words in the latent factors space in combination with a graph-based representation of how they are related. The model is extensible to new features in other domains as well. We use a hashtag similarity graph which helps take into account prevalent hashtags while also rewarding hashtags that are tightly associated with a narrow subset of words.

CHAPTER 7

EXPERIMENTAL RESULTS: HASHTAG RECOMMENDATION

In this chapter, we present preliminary experiments related to the proposed hashtag recommendation methodology, presented in Chapter 6. We first describe the dataset and the preprocessing steps. Then, we introduce the metrics used in evaluation, in addition to several hashtag recommendation baseline methods. Finally, we present and discuss the results.

7.1 Data Sets

We collected the tweets coinciding with the run-up to the presidential elections of 2012 using Twitters API[2]. Due to the massive amounts of data, we crawled 7,829,280 tweets posted during 24 hours starting Tue, 06 Nov 2012 23:30:00 GMT. The basis of our search queries is a dictionary containing election related words that enable us to capture polarization as well. In order to limit the tweets, we used Obama, Romney and Election as the filter words[3].

7.2 Data Preprocessing

First, all messages were transformed to lower case, then we removed non-english tweets and characters, as well as punctuation and numbers. As shown in Table 7.1, only 16% of tweets contain hashtag, and this is a compelling motivation for the need for a good approach to either recommend hashtags to the user while typing the message or

[2]http://search.twitter.com/search

[3]Note that we are cognizant of the fact that Twitter, and more so its public API, offers only a limited view into the vast variety of social discussions and communication.

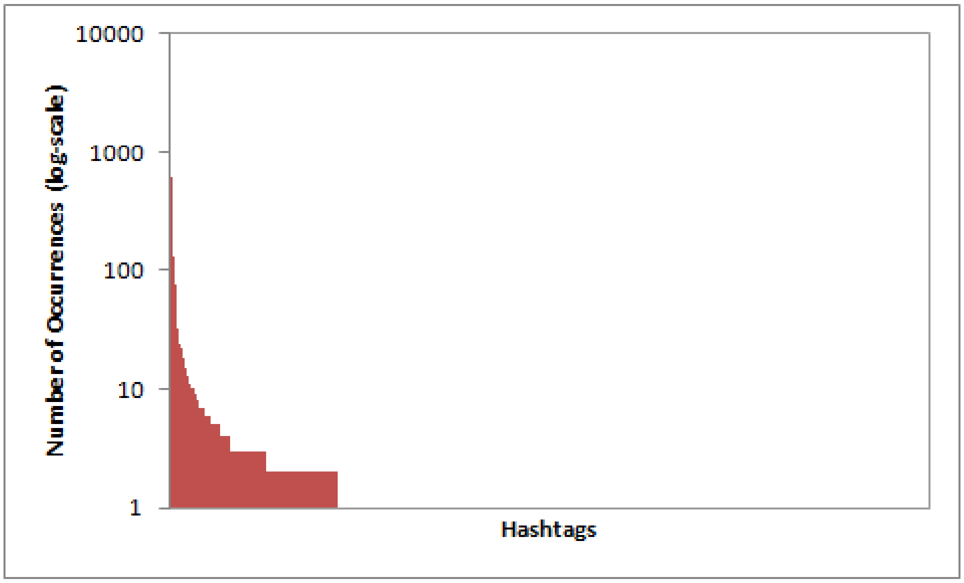


Figure 7.1: Long tail distribution of hashtag frequency.

predict hashtags offline for search, organization and knowledge discovery purposes. We also removed mentions and URLS, since the recommender system is based on tweets words only. More importantly, in order to have a fair evaluation, we also removed retweets which constitute about 38% of the tweets. Considering the retweets boosts the performance and gives biased results for hashtags in those tweets that are retweeted more than others. Moreover, 4% of the data consists of tweets with only hashtags or URLs, we discarded them due to lack of textual information. In order to remove spam and auto-generated tweets, we also removed tweets with 2 words or less (regardless of URLS and hashtags) which are about 11% of the dataset based on a heuristic based on observing the data.

Figure 7.1 and 7.2 depict the long tail distribution of hashtag frequencies and the makeup of the hashtags and their usage, respectively. These figures clearly indicate that only a small fraction of the hashtags is used with high frequency. This fraction contains the following hashtags, also shown in figure 7.2: #Obama, #Election, #RomneyRyan #Romney, #TCOT, #Vote. On the other hand, 98% of the hashtags were used less than 20 times in all the messages, of which the majority of hashtags occurred only once in the entire data. The fact that stands out about these figures is that the hashtag set is highly heterogeneous.

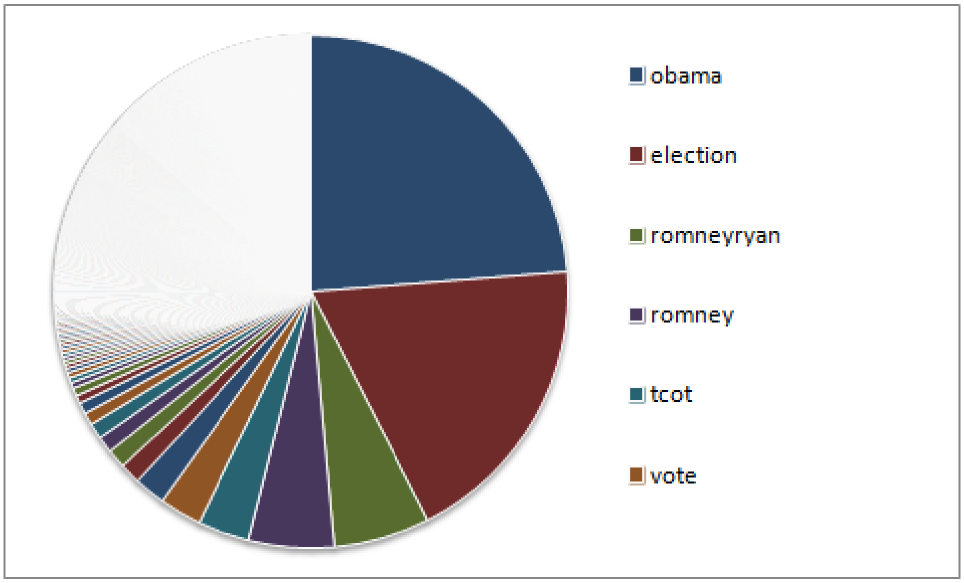


Figure 7.2: Distribution of hashtags.

TABLE 7.1

Overview of the crawled dataset and its characteristics.

Title	Number	Percentage of total
Number of tweets total	7,829,280	100%
Number of tweets with hashtags	1,311,840	16.8%
Number of retweets	2,979,360	38%
Number of tweets with only hashtags and URLs	343,440	4.4%
Number of tweets with only 2 or less words (not considering hashtags and URLs)	904,320	11.6%

After preprocessing the tweets data, we ended up with 208,160 tweets that were converted into a bag of words matrix. We then applied the same approach to extract the bag of words from the hashtags in each tweet. We thus ended up with two bag of words data matrices corresponding to two domains, text and hashtags, respectively. The details of the final data set, including the number of hashtags and words are shown in Table 7.2.

To prepare for the evaluation step, we split the data set into training and test sets to evaluate the generalization ability of the proposed hashtag recommender system.

Number of Tweets	Number of Distinct Hashtags	#Number of Words
208,160	68,187	136,383

TABLE 7.2

Details of Twitter dataset.

7.3 Evaluation Metrics

In order to evaluate the performance of our recommendation approach, two metrics are computed for each experiment. The Mean Average Precision (MAP) is the sum of inverse ranks divided by the minimum of relevant words and ranking length:

$$MAP = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{n_i}\sum_{k=1}^{n_i} Precision\,(R_{ik}) \times Relevance\,(i,k) \tag{7.1}$$

where Q is the set of queries, and $|Q|$ is the total number of queries submitted to the retrieval system. In our case, it is equal to the number of test tweets used for the evaluation. R is the matrix of ranked similar results for the current tweet. It is a $|Q| \times n$ matrix, with n being the number of tweets compared to, which is the number of test tweets in our case. $Precision(R_ik)$ is the precision value for hashtag i, of the results at recall level k (when k results are returned). $Relevance(i,k)$ is a binary function that indicates whether the result i is relevant to the query i or not. n_i is the number of results returned. MAP captures the relevance of the recommended tags compared to the real one.

The second metric is the Hit Rate, another important metric used to verify whether the recommended hashtags are among the ground-truth or not. Considering each tweet, if any of the recommended hashtags are among the ground-truth then a hit occurred. We compute the Hit Rate metric based on the ratio of the number of hits to the total number of test tweets.

$$Hit\,Rate = \frac{Number\,of\,hits}{Total\,number\,of\,hashtags} \tag{7.2}$$

A hit occurs when the ground-truth includes at least one of the predicted hashtags.

7.4 Experimental Results

7.4.1 Parameter Estimation

Before delving into the performance of the compared approaches, we will explain how we set the parameters used in the experiments. There are three important parameters which need to be tuned. The first parameter is k_f which indicates the number of latent factors in applying Non Negative Matrix Factorization, either Mixed NMF or Asymmetric NMF. The second important parameter is k_n which represents the size of the neighborhood and is necessary in order to determine neighboring tweets that are similar to a test tweet. The third parameter is the number of recommended hashtags denoted as k_t.

In this section, we investigate the impact of the parameters as well as finding the best values for further experiments. As discussed before, we recommend five hashtags for each test tweet. We used a grid search to tune all the parameters. In order to find the best k_f, we performed Mixed NMF using several numbers of latent factors in the set 10, 40, 70, 100, 200. Different k_f values lead to different latent spaces on which the data is projected. Figure 7.3 and 7.4 present the MAP and Hit Rate, respectively, for Mixed NMF. For each figure, we varied the neighborhood size k_n to find the best k_f. To remove bias, the experiments are based on 10-fold cross validation. Furthermore, to mitigate the effect of random initialization, we ran Mixed NMF 10 times and report the average of MAP and Hit Rate. In addition, the number of iterations for Mixed NMF is set to 1000. In Figure 7.3 and 7.4, we can see the trend for different neighborhood sizes k_n, allowing us to determine the best kf for further experiments. We note that $k_f = 10$ results in the highest MAP and Hit Rate, and therefore we set $k_f = 10$ for the remaining experiments.

It is also obvious that a higher k_f leads to higher MAP and Hit Rates. However, an important strength of using NMF is to estimate the original space with very few latent factors compared to the input features, so that it can be applied on large data sets. It is thanks to these savings that step 1 generates the tweets candidate set using the inner product using considerably smaller matrices compared to the original data, in terms of

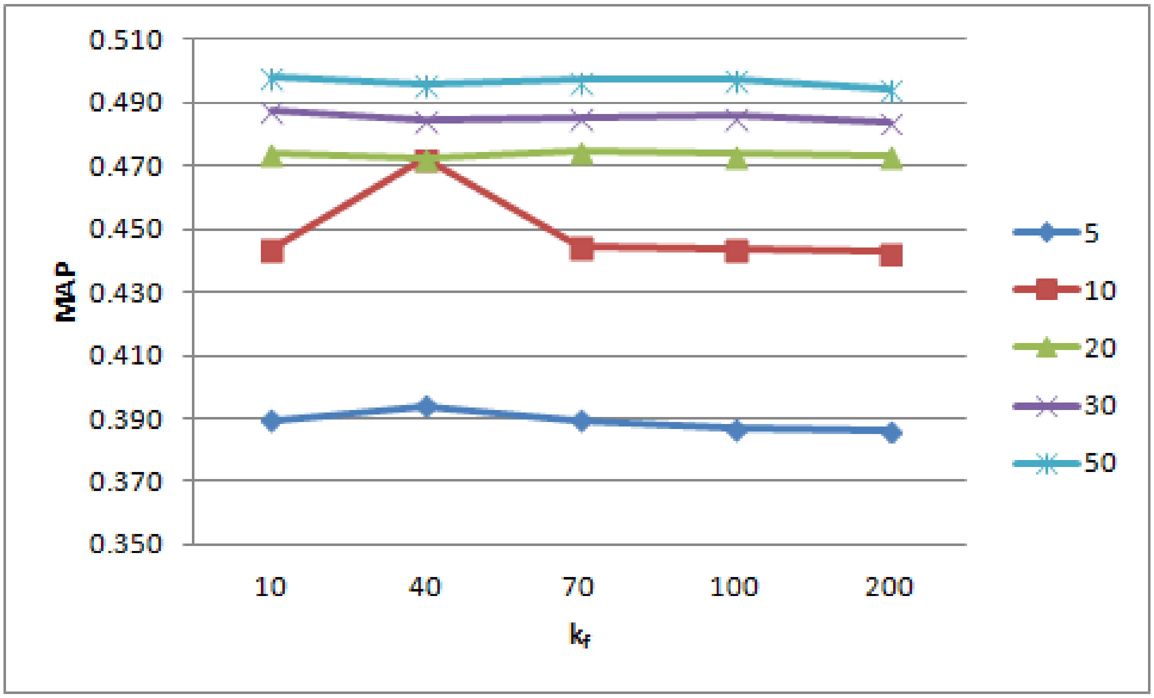


Figure 7.3: MAP for different values of k_n and k_f.

dimensionality.

As can also be seen in Figure 7.3 and 7.4, both metrics increase as a result of an increase in the neighborhood size. It is reasonable that considering more similar tweets leads to a larger tweet set and as a result of this, more related hashtags are retrieved. However, increasing the size of the neighborhood also requires more computations. At the same time, the proportion of increase of MAP and Hit Rate drops as k_n increases. Therefore, in order to select the tweet candidate set, it is better to use a sufficiently large neighborhood size, k_n, to capture similar tweets, while keeping it small enough to avoid any unnecessary complexity. This led us to set $k_n = 20$ neighbors.

7.4.2 Comparison of Recommendation Methods

In this section, we compare the performance of the proposed methods to several baseline methods. As mentioned before, the crawled tweets are related to the presidential elections of 2012. Since the tweets were limited to only one day, one may expect most of the hashtags to be very similar, essentially reducing to the most popular hashtags, thus

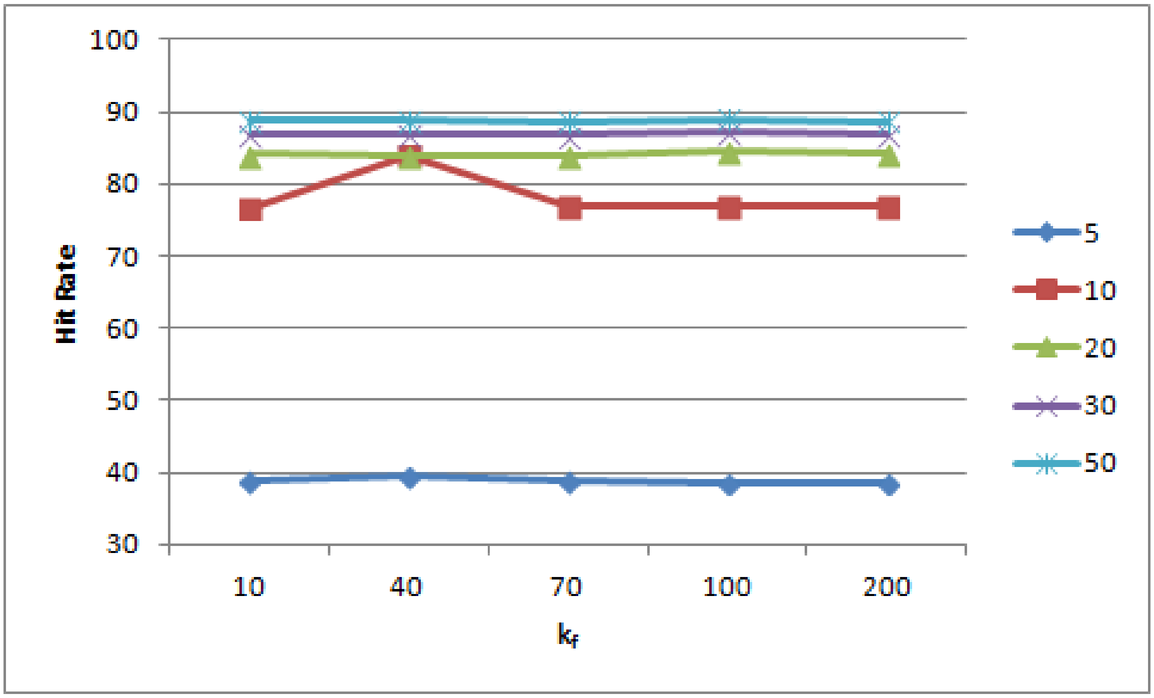


Figure 7.4: Hit Rate for different values of k_n and k_f.

eliminating the need for using a hashtag recommendation system. By using the Overall popularity (OP) method, we will investigate this assumption. This baseline approach assign the most frequent hashtags to all of the test tweets, we call it $OP@1$. To make it even more challenging, we also compared with a baseline where we recommended the 5 most frequent hashtags ($OP@5$) to further challenge the assumption mentioned above. As another baseline, we also use the K-nearest neighbor approach (KNN) with cosine similarity to find the tweet candidate set. KNN is a simple yet powerful machine learning approach that has been widely used for collaborative filtering recommender systems. In order to have a fair comparison, we use the same k_n and hashtag selection techniques for all NMF and KNN methods. Table 7.3 lists the parameter values for the rest of the experiments.

One important issue is to compute MAP at different levels. Figure 7.5 presents the MAP values for the proposed approaches and for KNN. As mentioned before, since we recommended 5 hashtags, we start with computing MAP at level 5 ($MAP@5$). By increasing the level of computing the MAP, the value also increases. This increase is due to considering a larger set of recommended hashtags to compute MAP. As mentioned before,

TABLE 7.3

Notations, descriptions and values of the parameters.

Notation	Description	Value
k_f	Number of NMF factors	10
k_n	Number of a tweet neighbors	20
k_t	Number of recommended hashtags	5
NMF_{ite}	Number of NMF iterations	1000
k-fold	Number of folds	10
NMF_{run}	Number of NMF runs	10

TABLE 7.4

for comparison of MAP@5 metric at different level for three different approaches.

Tested Hypothesis	KNN ¿ Mixed	ASYM ¿ Mixed	ASYM ¿ KNN
p-value	0.001	0.001	0.004

p-values

we recommend the top 5 relevant hashtags, so we report MAP at level 5. In addition, we can see that the Asymmetric NMF approach outperforms the other two approaches. The reason behind this is likely due to the ability to transfer knowledge from a richer content space to another content space, while allowing each to have its own latent space, this leads to a better recommender system. In addition, finding similar tweets in a different space rather than the original space enables Asymmetric NMF to capture more meaningful information. Table 7.4 shows the $p-values$ for each pair of methods. The p-values are obtained based on 10 different runs of the algorithm with 10 different NMF initializations. To obtain p-values we used paired t tests [186]; however in order to hold the normality assumption, we applied a log normal transformation [186] on the values. As we can see, all the p-values are less than 0.01, leading us to reject the null hypothesis that the differences between two approaches are not significant, with 99% confidence interval. The conclusion is that the differences are significant.

Figure 7.6 shows a comparison of all the NMF approaches as well as the other two baseline methods, OP@1 and OP@5, based on the MAP@5 and Hit Rate metrics. As can be seen, OP@5 has the highest Hit Rate. This is due to the fact that it considers the

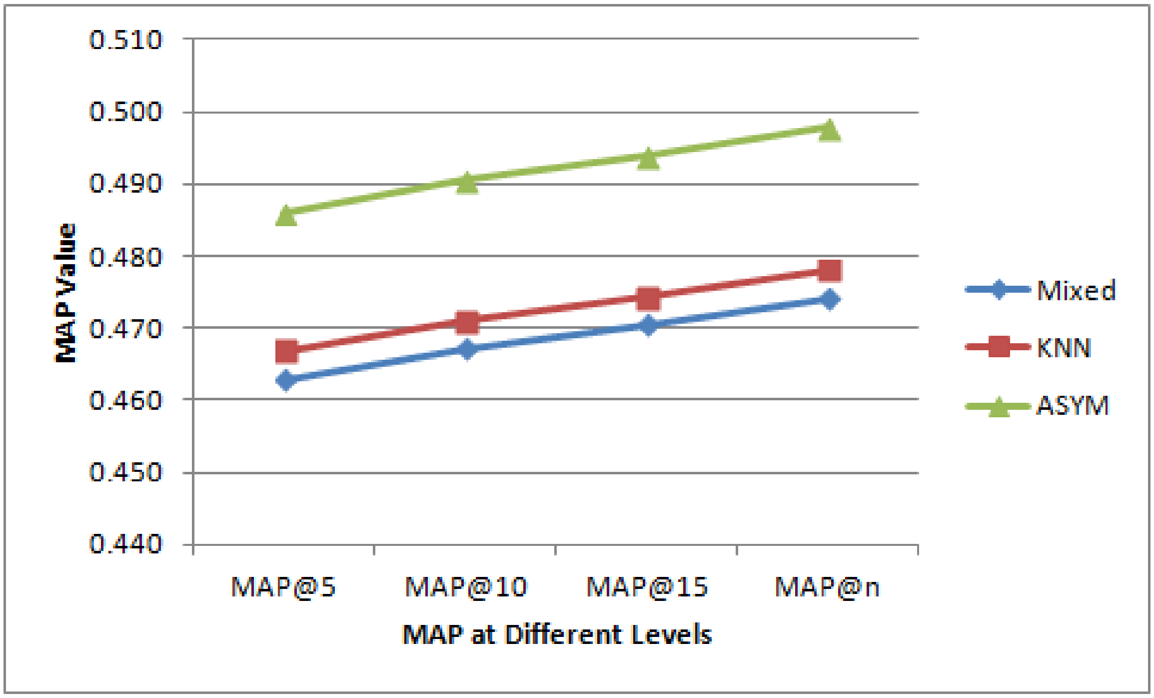


Figure 7.5: Comparison of MAP at different level for three different approaches.

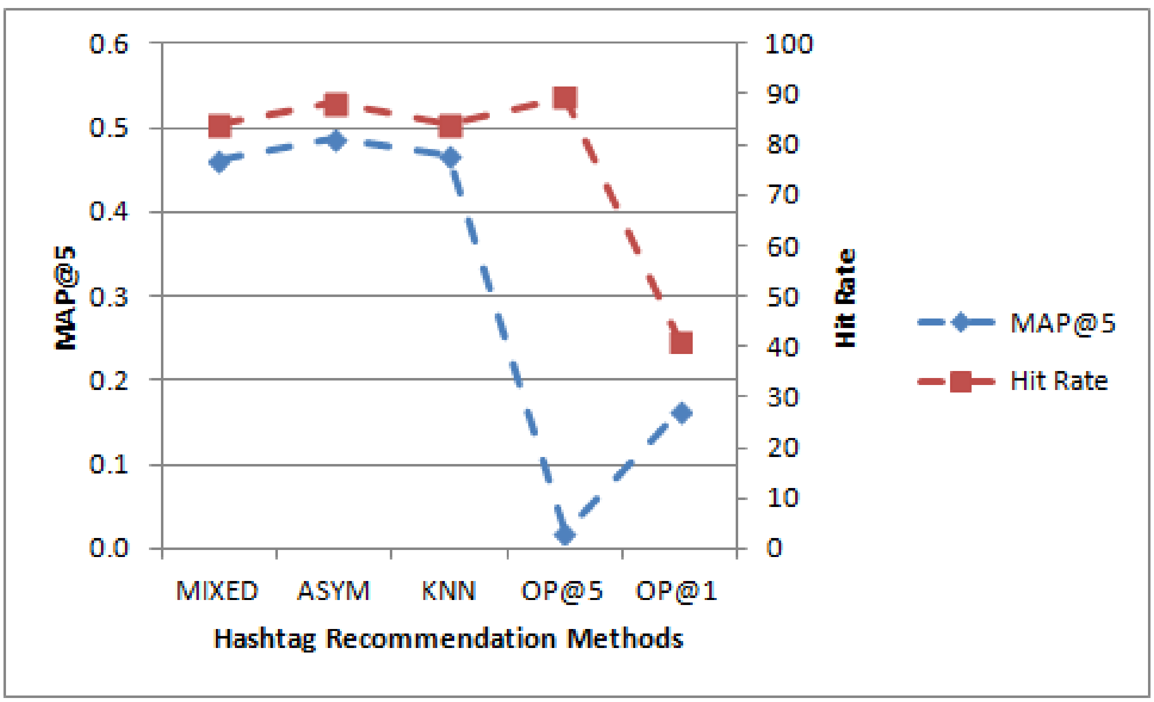


Figure 7.6: MAP@5 and Hit Rate for different hashtag recommendation methods.

TABLE 7.5

Comparison of MAP@5 and Hit Rate metrics for Asym and over popularity based approaches.

Method	ASYM vs OP@5	ASYM vs OP@1
p-value for MAP	0.0021	0.001
p-value for Hit Rate	0.0092	0.002

p-value

top 5 frequent hashtags for all tweets. However it has very low MAP@5, which shows how inefficient it is to always consider the same set of popular hashtags for recommendations. Moreover, Asymmetric NMF outperforms other rival approaches in terms of both Hit Rate and MAP@5. Table 7.5 shows the p-values for the comparison of ASYM, OP@5 and OP@1. The results confirm that Asymmetric NMF is superior in terms of both MAP@5 and Hit Rate metrics.

7.4.2.1 Graph-based analysis for story revelation

Recommending hashtags is very challenging due to the presence of very similar hashtags in terms of meaning and usage. To overcome this problem, we proposed a new approach for Phase two of this study, which is dedicated to hashtag selection. As mentioned in chapter 4, we propose a graph-based method which enables the recommender system to find different groups of hashtags which either have been used together or share some similarities in the previously used tweets. The graph shows the network of hashtags, in which each hashtag is a node. Two nodes are connected if there is sufficient similarity between them. The similarity is defined based on either one of two criteria, which are respectively, content-based or co-occurrence based, as follows: 1) average cosine similarity between the tweets they have appeared in, and 2) ratio of co-occurrence of the two hashtags in the data set.

Figure 6.2 shows the cosine similarity-based hashtag network. In order to visualize the graph, we randomly select a subset of 1000 tweets to illustrate the next steps. Fig. **??** shows the hashtags with cosine similarity value exceeding 0.15. Darker and larger nodes indicate hashtags with higher degree.

Several interesting facts can be discerned from Figure 6.2. For example, some cliques can be considered as communities which may contain valuable information. For instance, Table **??** shows some of these cliques and their descriptions. We can see a very strong relationship among the nodes of each community. We therefore list the hashtags used in each community as well as the related news that we could find by querying Google using the terms in the clique. In other words, each clique conveys a different story that was being discussed or shared on twitter. The first row in table 7.4 is about an attempted attack in Colombia on the day of the presidential elections. Although these tweets were not related to the US presidential election, they were found in the data set, due to the use of words such as President, Election, Vote, etc. It is interesting to note how the graph can capture different parallel events, co-occurring with the main topic. Another graph community, shown in figure 7.7 , corresponds to the campaign sign vandalism reported in Prince William, Manassas, where a large Romney sign was stolen and a number of signs were damaged.

7.4.2.2 Graph-based analysis for improved candidate selection in the hashtag recommender system

The clique-based communities in the hashtag graph illustrate how the hashtag connections are informative and may boost the recommendation system. We therefore leverage the hashtag network to find relevant hashtags for recommendation. Unrelated to recommendations, we can also use the terms in the hashtag cliques to query a search engine for verifiable stories.

Although many hashtags share similar tweets, finding the hashtags which tend to co-occur more often is very helpful. Figure **??** shows the hashtag network graph based on co-occurrence ratio. As shown in the graph, the center points are the most frequent hashtags with many other hashtags being connected to them. Similarly to the similarity graph, this graph also has cliques that reveal stories, some of which are similar to the stories found using the similarity graph. For example, Table 7.6 shows one of the cliques and its

Story one	**Hashtags**	#antioquia, #elecciones, #electoral, #farc, #nacional, #nueva, #octubre, #presidente, #trendnews
	News	Elections **Alleged homemade FARC bomb found on election day rural Colombia** May 25, 2014 posted by Philip Acuña http://goo.gl/YBgpzi
Story two	**Hashtags**	#blacks, #dalecity, #manassas, #mitromney, #woodbridge
	News	**Campaign Sign Vandalism Reported in Prince William, Manassas** By POTOMAC LOCAL - October 25, 2012 9:46 pm Leave a Comment  http://goo.gl/dxbpxi

Figure 7.7: Cliques found in the cosine similarity-based hashtag graph and corresponding news stories found by searching using the query consisting of the terms in the cliques.

hashtags which are related to the FARC story mentioned above.

Due to the rich knowledge that is contained in the hashtag network, we used it to improve the hashtag selection step. After finding the candidate tweet set, in addition to the hashtags used in the current candidate tweets, we find other similar tags based on either the cosine similarity graph or co-occurrence graph. This tends to augment the candidate hashtag set with more relevant hashtags. Figure 7.9 shows the comparison of KNN- Mixed and Asym NMF with and without using the hashtag graphs to augment the candidate sets. We can observe that using the hashtag graph significantly improves all of the methods.

TABLE 7.6

A Clique found in the co-occurrence-based hashtag graph.

Hashtags	#antioquia, #electoral, #octubre, #trendnews, #estados, #farc, #nueva, #presidente, #elecciones, #nacional

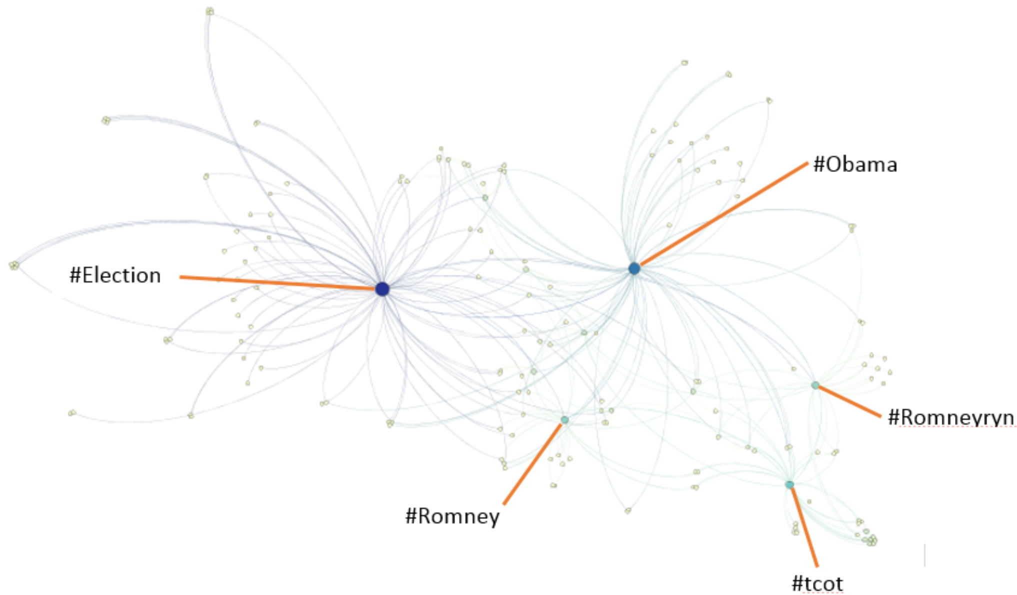


Figure 7.8: Co-occurrence-based hashtag graph.

This is due to the strong relationship between the words and tweets which improves all the recommender systems. Using the story carrying hashtag cliques allows the recommender system to discover and thus recommend more relevant hashtags. This can be seen as a specialized kind of recommender system: it recommends hashtags that potentially lead to annotating and most importantly connecting together tweets that are part of the same evolving story. This is very close to annotating tweets with stories and not just hashtags, which is another potential future application of the proposed recommendation methodology. Moreover, we notice that both the cosine similarity and co-occurrence networks have almost the same effect. To assess the significance of the differences between the two graphs, we computed the p-value considering 10 experiments to test the null hypothesis that there

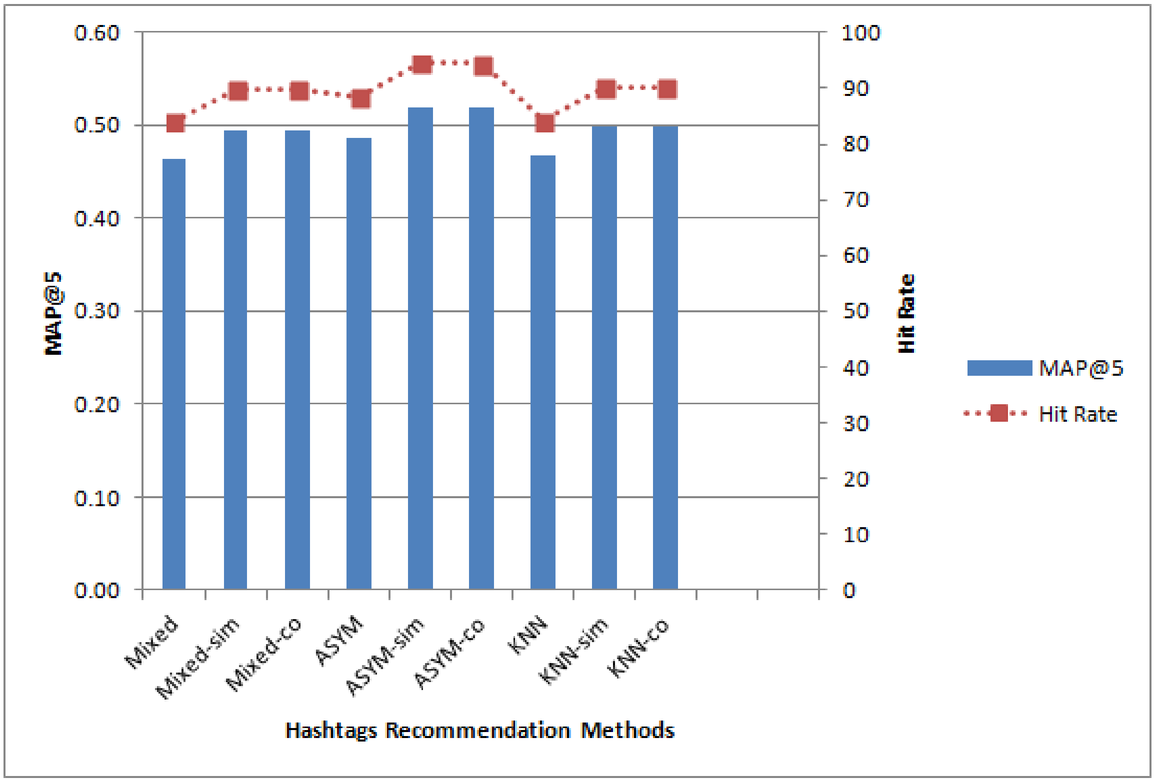


Figure 7.9: Comparison of KNN- Mixed and Asym NMF with and without using the hashtag graphs to augment the candidate sets.

are no significant differences between these two graph methods. The p-value was found to be 0.091, which leads us to accept the null hypothesis. In another words, there is no significant difference between NMF+cosine similarity graph and NMF+co-occurrence graph approaches. One reason behind this may be that both graphs capture the same information of hashtags used in the tweets.

TABLE 7.7

of comparison of KNN- Mixed and Asym NMF with and without using the hashtag graphs to augment the candidate sets.

p-values

	ASYM vs ASYM Sim	ASYM vs ASYM co	ASYM Sim vs ASYM co
p-value for MAP@5	0.001	0.009	0.091
p-value for Hit Rate	0.001	0.012	0.01

In addition, Fig 7.5 shows the superiority of Asymmetric NMF which experiences an even better improvement by using the hashtag similarity graphs. The Asym NMF+cosine similarity graph approach reaches 0.49 and 88.3% for MAP@5 and Hit Rate, respectively, corresponding to about 10% improvement compared to the highest performing baseline method, KNN.

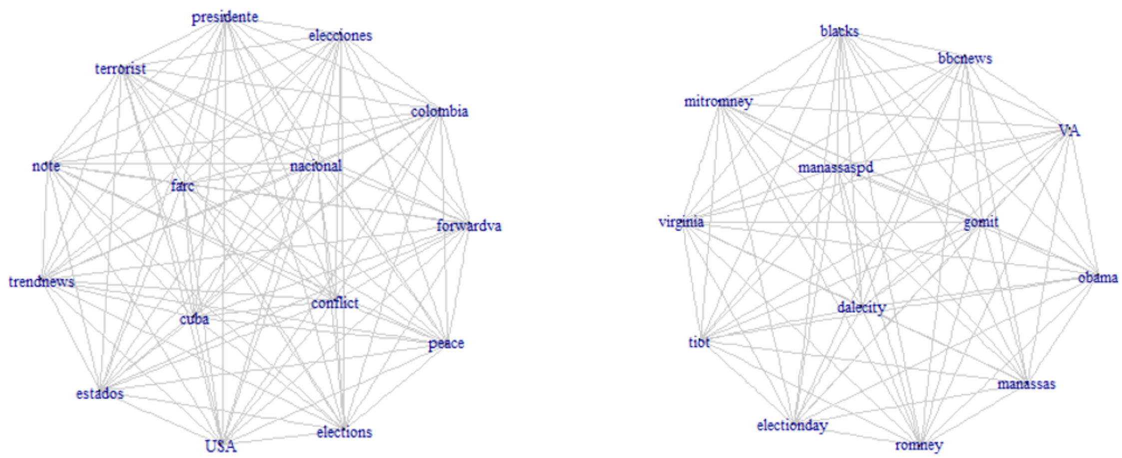


Figure 7.10: Story 1 and Story 2 cliques based on ground-truth hashtag network using cosine similarity.

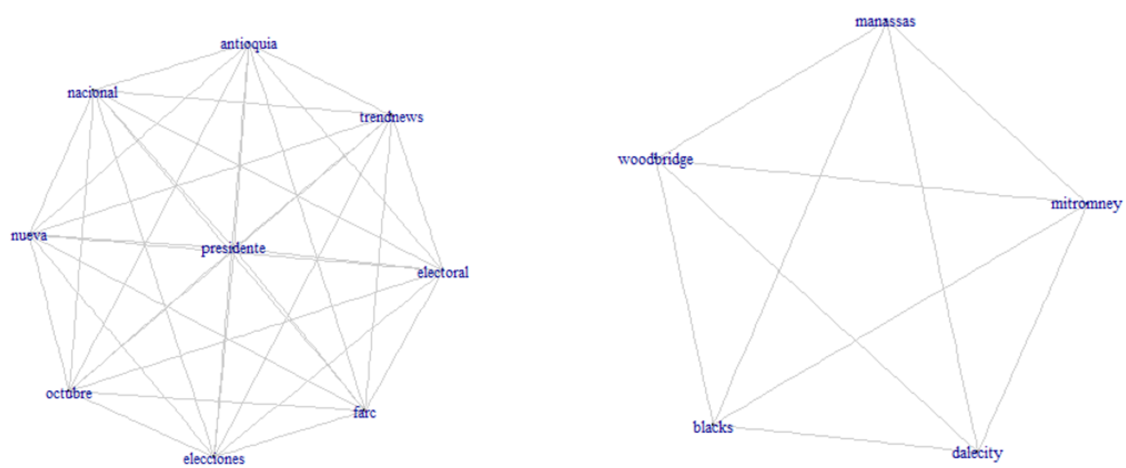


Figure 7.11: Story 1 and Story 2 cliques based on recommended hashtag network using cosine similarity

To verify the proposed method, we examined the cosine similarity hashtag network for both ground-truth and recommended hashtags. To do so, we consider the same 1000 tweets used before and partition them into training and test sets randomly. However, in

Ground-truth hashtag network

	$[C^-, C^+]$	φ_h
#ANTIOQUIA	[0, 1]	0
#ELECCIONES	[61, 532]	0.12
#ELECTORAL	[7, 109]	0.064
#FARC	[0, 5]	0
#NACIONAL	[0, 5]	0
#NUEVA	[0, 2]	0
#OCTUBRE	[1, 2]	0.5
#PRESIDENTE	[0, 38]	0
#TRENDNEWS	[0, 6]	0
#TERRORIST	[1, 4]	0.25
AVERAGE	**[7, 70.5]**	**0.093**

Recommended hashtag network

	$[C^-, C^+]$	φ_h
#ANTIOQUIA	[0, 1]	0
#ELECCIONES	[61, 532]	0.11
#ELECTORAL	[7, 109]	0.064
#FARC	[0, 5]	0
#NACIONAL	[0, 5]	0
#NUEVA	[0, 2]	0
#OCTUBRE	[1, 2]	0.5
#PRESIDENTE	[0, 38]	0
#TRENDNEWS	[0, 6]	0
#TERRORIST	[1, 4]	0.25
#COLOMBIA	[1, 11]	0.09
#CUBA	[3, 23]	0.13
#ESTADOS	[0, 7]	0
#FORWARDVA	[3, 50]	0.06
#NOTE	[1, 1]	1
#PEACE	[9, 94]	0.096
#USA	[475, 6999]	0.068
#ELECTION	[62628, 271509]	0.23
AVERAGE	**[3510.5, 15522.5]**	**0.145**

Figure 7.12: Polarization score for Story 1 clique based on ground-truth and recommended hashtag network using cosine similarity

order to be able to verify the same story that we described above, we consider tweets that are related to Story 1 and Story 2 as part of both training and test sets. Finally, the test set ended up with 200 tweets that will be used to evaluate the proposed approach. Figure 7.11 shows the cliques corresponding to Story 1 and Story 2 in the test set, we detect theses cliques based on ground-truth hashtags in the test set. In the next step, we hide the ground-truth hashtags and try to recommend new hashtags by building a model on the training set. Using a greedy clique detection algorithm [187], we found 41 cliques in the ground truth hashtag graph. After recommending new hashtags to the unseen test tweets, we built a cosine similarity hashtag network again. Then we used the same greedy clique detection algorithm and this time, found 53 cliques, showing that the recommended hashtags were able to connect more related communities together. However, it is important to see how the hashtags in a clique are related. In order to do so, we found the most relevant hashtag cliques to Story 1 and Story 2 above. As shown in Figure 7.10, these cliques are now denser compared to the cliques in Figure 7.10, while the recommended hashtags remain almost the same. These results imply that the recommendation system is able to recommend many more related hashtags for the test tweets using the graph-based candidate set augmentation.

Ground-truth hashtag network

	$[C^-, C^+]$	φ_h
#MANASSAS	[2, 1]	0.5
#BLACKS	[36, 13]	0.3611111111
#DALECITY	[2, 0]	0
#MITTROMNEY	[292, 1150]	0.2539130435
#WOODBRIDGE	[2, 1]	0.5
AVERAGE	**[67.2, 233]**	**0.28**

Recommended hashtag network

	$[C^-, C^+]$	φ_h
#MANASSAS	[2, 1]	0.5
#BLACKS	[36, 13]	0.3611111111
#DALECITY	[2, 0]	0
#MITTROMNEY	[292, 1150]	0.2539130435
#WOODBRIDGE	[2, 1]	0.5
#BBCNEWS	[22, 91]	0.2417582418
#ELECTIONDAY	[1293, 7331]	0.1763743009
#GOMITT	[1, 53]	0.01886792453
#OBAMA	[69416, 327069]	0.2122365617
#ROMNEY	[11981, 50340]	0.2380015892
#TIOT	[58, 191]	0.3036649215
#VA	[85, 597]	0.1423785595
#VIRGINIA	[109, 1321]	0.08251324754
AVERAGE	**[83198.4, 386938.6]**	**0.23**

Figure 7.13: Polarization score for the Story 2 clique based on ground-truth and recommended hashtag network using cosine similarity

7.5 Polarization in the Hashtag Community

In order to measure the polarization of hashtags in tweets, we first applied unsupervised lexicon-based sentiment analysis on each tweet and labeled it as positive/negative sentiment [188] using a sentiment analysis tool optimized to annotate short messages, like tweets, that contain abbreviations, slang, and other vernacular. It also employs linguistic rules for negations, amplifications, booster words, emoticons, spelling corrections, etc [188]. After obtaining the sentiment of each tweet, we check all the hashtags appearing in all tweets, and count the sentiment frequency of those hashtags based on the tweet label. We then compute an intuitive measure of polarization using the ratio of the number of tweets in the positive/negative sentiments, with the least frequent tweets in the numerator, so that the ratio is always below 1. Figure 7.12 and figure 7.13 show the polarization scores for Story 1 and Story 2 cliques based on the ground-truth and recommended hashtag network using the cosine similarity. For each connected word, we count the number of positive and negative tweets, which are shown as C^+ and C^- respectively. We then computed the average polarization score (ϕ_h) over all hashtags of each clique. We noticed that proposed hashtag recommendation did not increase the average polarization score, due to using the content of the tweets.

To conclude, our research in hashtag recommendation was able to exploit the textual content that is available as part of user messages or posts, and thus resulted in hybrid recommendation strategies. Using content within this context can bridge polarization boundaries. However, when content is not available, is missing, or is unreliable, as in the case of platforms that are rich in multimedia and multilingual posts, the content option becomes less powerful and pure collaborative filtering regains its important role, along with the challenges of polarization.

CHAPTER 8

CONCLUSIONS

In this dissertation, we investigated the mechanism of filtering and discovering information online while using recommender systems. In the first part of our research, we studied the phenomenon of *polarization* and its impact on filtering and discovering information. Polarization is an important phenomenon, with serious consequences, in real-life, particularly on social media. Thus it is important to understand how machine learning algorithms, especially recommender systems, behave in a polarized environment; and to this end it is important to quantify polarization in existing and new data sets. We presented a domain-independent data science pipeline to automatically detect polarization using the ratings given by users to items. Our polarization detection framework was shown to detect different degrees of polarization and to outperform existing measures in capturing an intuitive notion of polarization. Our work is an essential step toward quantifying and detecting polarization in ongoing ratings and in benchmark data sets, and to this end, we used our developed polarization detection pipeline to compute the polarization prevalence of several benchmark data sets. It is our hope that this work will contribute to supporting future research in the emerging topic of designing and studying the behavior of recommender systems in polarized environments.

We have also investigated and uncovered certain peculiar patterns that are characteristic of environments where polarization emerges, for instance by monitoring traces of the objective function that is minimized in matrix factorization. We have found that environments with different polarization degrees engender different patterns. The ability to recognize such patterns, that arise during incremental optimization, can help quickly detect and quantify the evolution of polarization without the tedious analysis of rating data distributions. This objective function optimization pattern monitoring-based approach open a new direction of research in studying and handling polarization, not only in recommender systems, but also in other machine learning algorithms that also result in filtering information from humans.

Most importantly, although we have given a natural explanation for the model's lower reconstruction error (hence better prediction) for higher polarization, this is still surprising (when taken in the context of learning in polarized environments), and disappointing (machine learning algorithms find it easier to learn discriminating models in polarized environments: The models will quickly learn to keep each user in the safety of their preferred viewpoint, essentially, giving rise to filter bubbles).

Our proposed counter-polarization methodology succeeds to cover items from the opposite view after a few iterations and can broaden the viewpoint spectrum even faster if the user is more interested in discovering items from different viewpoints. Endowing humans with the ability to interactively and adaptively control the breadth of viewpoints offered by an intelligent polarization-aware recommendation algorithm is an important capability. As a result, users who are nowadays increasingly and unconsciously entrapped in algorithmic filters over which they have had no control, can become empowered to break free from their algorithmic chains. To our knowledge, this feature which allows humans to remain in or regain control of algorithm-induced filter bubble traps, has heretofore not been allowed or engineered in existing information filtering systems, whether on social media, e-commerce or e-learning recommender systems. We also make a theoretical analysis of how polarization affects learning latent factor models, and how counter-polarization affects these models.

In the second part of our dissertation, we investigate the problem of discovering related information by recommendation of tags on social media micro-blogging platforms. Real-time micro-blogging services such as Twitter have recently witnessed exponential growth, with millions of active web users who generate billions of micro-posts to share information, opinions and personal viewpoints, daily. However, these posts are inherently noisy and unstructured because they could be in any format, hence making them difficult to organize for the purpose of retrieval of relevant information. One way to solve this problem is using hashtags, which are quickly becoming the standard approach for annotation of various information on social media, such that varied posts about the same or related topic are annotated with the same hashtag. However hashtags are not used in a consistent manner and most importantly, are completely optional to use. This makes them unreliable as the sole mechanism for searching for relevant information. We investigate mechanisms for consolidating the hashtag space using recommender systems. Our methods are general enough that they can be used for hashtag annotation in various social media services such as twitter, as well as for general item recommendations on systems that rely on implicit user interest data such as e-learning and news sites, or explicit user ratings, such as e-commerce and online entertainment sites. To conclude, we propose a methodology to extract stories based on two types of hashtag co-occurrence graphs. Our research in hashtag recommendation was able to exploit the textual content that is available as part of user messages or posts, and thus resulted in hybrid recommendation strategies. Using content within this context can bridge polarization boundaries. However, when content is not available, is missing, or is unreliable, as in the case of platforms that are rich in multimedia and multilingual posts, the content option becomes less powerful and pure collaborative filtering regains its important role, along with the challenges of polarization.

REFERENCES

[1] Elaine Rich, "User modeling via stereotypes," *Cognitive science*, vol. 3, no. 4, pp. 329–354, 1979. [Cited on page 1.]

[2] Gerard Salton, "Automatic text processing: The transformation, analysis, and retrieval of," *Reading: Addison-Wesley*, 1989. [Cited on page 1.]

[3] Gary L Lilien, Philip Kotler, and K Sridhar Moorthy, *Marketing models*, Prentice Hall, 1992. [Cited on page 1.]

[4] Prem Melville and Vikas Sindhwani, "Recommender systems," in *Encyclopedia of machine learning*, pp. 829–838. Springer, 2011. [Cited on pages 1 and 14.]

[5] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez, "Recommender systems survey," *Knowledge-based systems*, vol. 46, pp. 109–132, 2013. [Cited on pages 1, 12, 13, 14, and 21.]

[6] 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*, vol. 17, no. 6, pp. 734–749, 2005. [Cited on pages 1, 13, 15, 19, and 20.]

[7] 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*. ACM, 1994, pp. 175–186. [Cited on pages 1 and 19.]

[8] Joseph A Konstan and John Riedl, "Recommender systems: from algorithms to user experience," *User modeling and user-adapted interaction*, vol. 22, no. 1, pp. 101–123, 2012. [Cited on pages 1 and 30.]

[9] Bart P Knijnenburg and Martijn C Willemsen, "Evaluating recommender systems with user experiments," in *Recommender Systems Handbook*, pp. 309–352. Springer, 2015. [Cited on page 1.]

[10] Sean M McNee, John Riedl, and Joseph A Konstan, "Being accurate is not enough: how accuracy metrics have hurt recommender systems," in *CHI'06 extended abstracts on Human factors in computing systems*. ACM, 2006, pp. 1097–1101. [Cited on page 1.]

[11] Sean M McNee, John Riedl, and Joseph A Konstan, "Making recommendations better: an analytic model for human-recommender interaction," in *CHI'06 extended abstracts on Human factors in computing systems*. ACM, 2006, pp. 1103–1108. [Cited on page 1.]

[12] Eytan Bakshy, Solomon Messing, and Lada A Adamic, "Exposure to ideologically diverse news and opinion on facebook," *Science*, vol. 348, no. 6239, pp. 1130–1132, 2015. [Cited on page 2.]

[13] Eli Pariser, *The filter bubble: What the Internet is hiding from you*, Penguin UK, 2011. [Cited on pages 2 and 3.]

[14] Ulrike Gretzel and Daniel R Fesenmaier, "Persuasion in recommender systems," *International Journal of Electronic Commerce*, vol. 11, no. 2, pp. 81–100, 2006. [Cited on page 2.]

[15] 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*. ACM, 2003, pp. 585–592. [Cited on page 2.]

[16] H Thaler Richard and R Sunstein Cass, "Nudge: Improving decisions about health, wealth, and happiness," 2008. [Cited on page 2.]

[17] John R Doyle, "Survey of time preference, delay discounting models," 2012. [Cited on page 2.]

[18] Bart P Knijnenburg, Saadhika Sivakumar, and Daricia Wilkinson, "Recommender systems for self-actualization," in *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016, pp. 11–14. [Cited on pages 2 and 4.]

[19] Danah Boyd, "Streams of content, limited attention: The flow of information through social media.," *Educause Review*, vol. 45, no. 5, pp. 26, 2010. [Cited on page 2.]

[20] Zeinab Abbassi, Sihem Amer-Yahia, Laks VS Lakshmanan, Sergei Vassilvitskii, and Cong Yu, "Getting recommender systems to think outside the box," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 285–288. [Cited on pages 2 and 29.]

[21] Panagiotis Adamopoulos and Alexander Tuzhilin, "On unexpectedness in recommender systems: Or how to better expect the unexpected," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 4, pp. 54, 2015. [Cited on pages 2, 3, 24, 30, and 32.]

[22] Upendra Shardanand and Pattie Maes, "Social information filtering: algorithms for automating word of mouth," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 1995, pp. 210–217. [Cited on page 2.]

[23] Clive Thompson, "Can you count on voting machines?," *New York Times*, p. 40ff, 2008. [Cited on page 2.]

[24] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann, "Support vector machines for multiple-instance learning," in *Advances in neural information processing systems*, 2002, pp. 561–568. [Cited on page 3.]

[25] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan, "Exploring the filter bubble: the effect of using recommender systems on content diversity," in *Proceedings of the 23rd international conference on World wide web*. ACM, 2014, pp. 677–686. [Cited on pages 3 and 29.]

[26] Daniel Fleder and Kartik Hosanagar, "Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity," *Management science*, vol. 55, no. 5, pp. 697–712, 2009. [Cited on page 3.]

[27] James A Evans, "Electronic publication and the narrowing of science and scholarship," *science*, vol. 321, no. 5887, pp. 395–399, 2008. [Cited on page 3.]

[28] Rashmi R Sinha and Kirsten Swearingen, "Comparing recommendations made by online systems and friends.," in *DELOS workshop: personalisation and recommender systems in digital libraries*, 2001, vol. 106. [Cited on page 3.]

[29] Natalie Jomini Stroud, "Media use and political predispositions: Revisiting the concept of selective exposure," *Political Behavior*, vol. 30, no. 3, pp. 341–366, 2008. [Cited on page 3.]

[30] Steven Kull, Clay Ramsay, and Evan Lewis, "Misperceptions, the media, and the iraq war," *Political Science Quarterly*, vol. 118, no. 4, pp. 569–598, 2003. [Cited on page 3.]

[31] Pranav Dandekar, Ashish Goel, and David T Lee, "Biased assimilation, homophily, and the dynamics of polarization," *Proceedings of the National Academy of Sciences*, vol. 110, no. 15, pp. 5791–5796, 2013. [Cited on pages 3, 4, 35, 39, 76, and 79.]

[32] Q Vera Liao and Wai-Tat Fu, "Can you hear me now?: mitigating the echo chamber effect by source position indicators," in *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 2014, pp. 184–196. [Cited on pages 3 and 35.]

[33] AJ Morales, Javier Borondo, Juan Carlos Losada, and Rosa M Benito, "Measuring political polarization: Twitter shows the two sides of venezuela," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 3, pp. 033114, 2015. [Cited on pages 3, 4, 7, 36, 46, 67, and 73.]

[34] Patrick Shafto and Olfa Nasraoui, "Human-recommender systems: From benchmark data to benchmark cognitive models," in *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016, pp. 127–130. [Cited on page 3.]

[35] Ricardo Baeza-Yates, "Data and algorithmic bias in the web," in *Proceedings of the 8th ACM Conference on Web Science*. ACM, 2016, pp. 1–1. [Cited on page 4.]

[36] Patricia Victor, Chris Cornelis, Martine De Cock, and Ankur Teredesai, "Trust-and distrust-based recommendations for controversial reviews," in *Web Science Conference (WebSci'09: Society On-Line)*, 2009, number 161. [Cited on pages 4, 46, and 67.]

[37] Cass R Sunstein, "The law of group polarization," *Journal of political philosophy*, vol. 10, no. 2, pp. 175–195, 2002. [Cited on page 4.]

[38] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis, "Balancing opposing views to reduce controversy," *arXiv preprint arXiv:1611.00172*, 2016. [Cited on pages 4 and 35.]

[39] Daniel J Isenberg, "Group polarization: A critical review and meta-analysis.," *Journal of personality and social psychology*, vol. 50, no. 6, pp. 1141, 1986. [Cited on page 4.]

[40] Patricia Victor, Chris Cornelis, Martine De Cock, and Ankur Teredesai, "A comparative analysis of trust-enhanced recommenders for controversial items.," in *ICWSM*, 2009. [Cited on pages 4, 35, 65, 67, 72, and 73.]

[41] Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac, "What recommenders recommend: An analysis of recommendation biases and possible countermeasures," *User Modeling and User-Adapted Interaction*, vol. 25, no. 5, pp. 427–491, 2015. [Cited on pages 4, 53, 55, 74, and 75.]

[42] Olfa Nasraoui and Patrick Shafto, "Human-algorithm interaction biases in the big data cycle: A markov chain iterated learning framework," *CoRR*, vol. abs/1608.07895, 2016. [Cited on pages 4 and 75.]

[43] Shilad Sen, Werner Geyer, Jill Freyne, and Pablo Castells, Eds., *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*. ACM, 2016. [Cited on page 6.]

[44] Antonis Matakos and Panayiotis Tsaparas, "Temporal mechanisms of polarization in online reviews," in *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. IEEE, 2016, pp. 529–532. [Cited on pages 7, 38, 46, 67, and 73.]

[45] Dhoha Almazro, Ghadeer Shahatah, Lamia Albdulkarim, Mona Kherees, Romy Martinez, and William Nzoukou, "A survey paper on recommender systems," *arXiv preprint arXiv:1006.5278*, 2010. [Cited on pages 12, 13, 14, 15, 21, and 23.]

[46] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender systems handbook*, pp. 73–105. Springer, 2011. [Cited on pages 13 and 15.]

[47] Francesco Ricci, Lior Rokach, and Bracha Shapira, *Introduction to recommender systems handbook*, Springer, 2011. [Cited on page 14.]

[48] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004. [Cited on page 14.]

[49] Xiaoyuan Su and Taghi M Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, pp. 4, 2009. [Cited on page 14.]

[50] Zhonghang Xia, Yulin Dong, and Guangming Xing, "Support vector machines for collaborative filtering," in *Proceedings of the 44th annual Southeast regional conference*. ACM, 2006, pp. 169–174. [Cited on page 15.]

[51] Myung Won Kim, Eun Ju Kim, and Joung Woo Ryu, "A collaborative recommendation based on neural networks," in *International Conference on Database Systems for Advanced Applications*. Springer, 2004, pp. 425–430. [Cited on page 15.]

[52] 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*. Morgan Kaufmann Publishers Inc., 1998, pp. 43–52. [Cited on page 15.]

[53] Jingu Kim and Haesun Park, "Fast nonnegative matrix factorization: An active-set-like method and comparisons," *SIAM Journal on Scientific Computing*, vol. 33, no. 6, pp. 3261–3281, 2011. [Cited on page 15.]

[54] James Bennett, Stan Lanning, et al., "The netflix prize," in *Proceedings of KDD cup and workshop*. New York, NY, USA, 2007, vol. 2007, p. 35. [Cited on pages 16 and 17.]

[55] Chris Ding, Xiaofeng He, and Horst D Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proceedings of the 2005 SIAM International Conference on Data Mining*. SIAM, 2005, pp. 606–610. [Cited on pages 16 and 17.]

[56] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Incremental singular value decomposition algorithms for highly scalable recommender systems," in *Fifth International Conference on Computer and Information Science*. Citeseer, 2002, pp. 27–28. [Cited on page 16.]

[57] Nathan Srebro, Tommi Jaakkola, et al., "Weighted low-rank approximations," in *Icml*, 2003, vol. 3, pp. 720–727. [Cited on page 16.]

[58] Ruslan Salakhutdinov and Andriy Mnih, "Probabilistic matrix factorization.," in *Nips*, 2007, vol. 1, pp. 2–1. [Cited on page 16.]

[59] Ruslan Salakhutdinov and Andriy Mnih, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 880–887. [Cited on page 16.]

[60] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King, "Recommender systems with social regularization," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 287–296. [Cited on page 16.]

[61] Jasson DM Rennie and Nathan Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 713–719. [Cited on page 16.]

[62] Yehuda Koren, Robert Bell, and Chris Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009. [Cited on pages 16, 21, 24, 53, 72, 76, and 90.]

[63] Fabio A González, Juan C Caicedo, Olfa Nasraoui, and Jaafar Ben-Abdallah, "Nmf-based multimodal image indexing for querying by visual example," in *Proceedings of the ACM International Conference on Image and Video Retrieval*. ACM, 2010, pp. 366–373. [Cited on page 17.]

[64] Inderjit S Dhillon and Suvrit Sra, "Generalized nonnegative matrix approximations with bregman divergences," in *NIPS*, 2005, vol. 18. [Cited on page 17.]

[65] Jaafar BenAbdallah, Juan C Caicedo, Fabio A Gonzalez, and Olfa Nasraoui, "Multimodal image annotation using non-negative matrix factorization," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*. IEEE, 2010, vol. 1, pp. 128–135. [Cited on pages 18, 92, and 94.]

[66] Daniel D Lee and H Sebastian Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562. [Cited on page 18.]

[67] Joel A Tropp, "Literature survey: Nonnegative matrix factorization," *University of Texas at Asutin*, 2003. [Cited on page 18.]

[68] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2004. [Cited on page 18.]

[69] Michael J Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial intelligence review*, vol. 13, no. 5-6, pp. 393–408, 1999. [Cited on pages 18 and 19.]

[70] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2002, pp. 253–260. [Cited on page 18.]

[71] Lyle H Ungar and Dean P Foster, "Clustering methods for collaborative filtering," in *AAAI workshop on recommendation systems*, 1998, vol. 1, pp. 114–129. [Cited on page 19.]

[72] Marko Balabanović and Yoav Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997. [Cited on pages 19 and 30.]

[73] Miles Efron, "Hashtag retrieval in a microblogging environment," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 787–788. [Cited on page 19.]

[74] Paul Heymann, Daniel Ramage, and Hector Garcia-Molina, "Social tag prediction," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2008, pp. 531–538. [Cited on page 19.]

[75] Ralf Krestel and Peter Fankhauser, "Tag recommendation using probabilistic topic models," *ECML PKDD Discovery Challenge*, vol. 2009, pp. 131, 2009. [Cited on page 19.]

[76] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme, "Learning optimal ranking with tensor factorization for tag recommendation," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 727–736. [Cited on page 19.]

[77] Steffen Rendle and Lars Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 81–90. [Cited on page 19.]

[78] Zhuoye Ding, Xipeng Qiu, Qi Zhang, and Xuanjing Huang, "Learning topical translation model for microblog hashtag suggestion.," in *IJCAI*, 2013. [Cited on page 19.]

[79] Jiliang Tang, Xia Hu, and Huan Liu, "Social recommendation: a review," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1113–1133, 2013. [Cited on page 19.]

[80] Sogol Naseri, Arash Bahrehmand, Chen Ding, and Chi-Hung Chi, "Enhancing tag-based collaborative filtering via integrated social networking information," in *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*. IEEE, 2013, pp. 761–765. [Cited on page 19.]

[81] Elham Khabiri, James Caverlee, and Krishna Y Kamath, "Predicting semantic annotations on the real-time web," in *Proceedings of the 23rd ACM conference on Hypertext and social media*. ACM, 2012, pp. 219–228. [Cited on page 19.]

[82] Lei Yang, Tao Sun, Ming Zhang, and Qiaozhu Mei, "We know what@ you# tag: does the dual role affect hashtag adoption?," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 261–270. [Cited on page 19.]

[83] Eriko Otsuka, Scott A Wallace, and David Chiu, "Design and evaluation of a twitter hashtag recommendation system," in *Proceedings of the 18th International Database Engineering & Applications Symposium*. ACM, 2014, pp. 330–333. [Cited on page 20.]

[84] 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*. ACM, 2013, pp. 593–596. [Cited on page 20.]

[85] Jaewon Yang and Jure Leskovec, "Modeling information diffusion in implicit networks," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 599–608. [Cited on page 20.]

[86] Yung-Ming Li and Ya-Lin Shiu, "A diffusion mechanism for social advertising over microblogs," *Decision Support Systems*, vol. 54, no. 1, pp. 9–22, 2012. [Cited on page 20.]

[87] Andrew L Brooks and Coye Cheshire, "Ad-itudes: twitter users & advertising," in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion*. ACM, 2012, pp. 63–66. [Cited on page 20.]

[88] Elizabeth Wright, Nile M Khanfar, Catherine Harrington, and Lee E Kizer, "The lasting effects of social media trends on advertising," *Journal of Business & Economics Research*, vol. 8, no. 11, pp. 73, 2010. [Cited on page 20.]

[89] Su Mon Kywe, Tuan-Anh Hoang, Ee-Peng Lim, and Feida Zhu, "On recommending hashtags in twitter networks," in *International Conference on Social Informatics*. Springer, 2012, pp. 337–350. [Cited on page 20.]

[90] Wei Feng and Jianyong Wang, "We can learn your# hashtags: Connecting tweets to explicit topics," in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*. IEEE, 2014, pp. 856–867. [Cited on page 20.]

[91] Janette Lehmann, Bruno Gonçalves, José J Ramasco, and Ciro Cattuto, "Dynamical classes of collective attention in twitter," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 251–260. [Cited on page 21.]

[92] Yasuko Matsubara, Yasushi Sakurai, B Aditya Prakash, Lei Li, and Christos Faloutsos, "Rise and fall patterns of information diffusion: model and implications," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 6–14. [Cited on page 21.]

[93] Guy Shani and Asela Gunawardana, "Evaluating recommendation systems," in *Recommender systems handbook*, pp. 257–297. Springer, 2011. [Cited on pages 21, 24, and 30.]

[94] Olfa Nasraoui, Jeff Cerwinske, Carlos Rojas, and Fabio Gonzalez, "Performance of recommendation systems in dynamic streaming environments," in *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM, 2007, pp. 569–574. [Cited on page 23.]

[95] Olfa Nasraoui, Osmar R Zaïane, Myra Spiliopoulou, Bamshad Mobasher, Brij Masand, and Philip S Yu, "Webkdd 2005: web mining and web usage analysis post-workshop report," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 139–142, 2005. [Cited on page 23.]

[96] Olfa Nasraoui, Jeff Cerwinske, Carlos Rojas, and Fabio Gonzalez, "Collaborative filtering in dynamic streaming environments," in *CIKM*, 2006, pp. 794–795. [Cited on page 23.]

[97] Olfa Nasraoui, Dipankar Dasgupta, and Fabio Gonzalez, "The promise and challenges of artificial immune system based web usage mining: Preliminary results," in *workshop on Web Analytics at Second SIAM International Conference on Data Mining (SDM), Arlington, VA*, 2002. [Cited on page 23.]

[98] Yoshinori Hijikata, Takuya Shimizu, and Shogo Nishida, "Discovery-oriented collaborative filtering for improving user satisfaction," in *Proceedings of the 14th international conference on Intelligent user interfaces*. ACM, 2009, pp. 67–76. [Cited on pages 23 and 30.]

[99] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th international conference on World Wide Web*. ACM, 2005, pp. 22–32. [Cited on pages 24, 29, 64, and 72.]

[100] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich, *Recommender systems: an introduction*, Cambridge University Press, 2010. [Cited on page 24.]

[101] Behnoush Abdollahi and Olfa Nasraoui, "Explainable matrix factorization for collaborative filtering," in *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 5–6. [Cited on page 24.]

[102] Gediminas Adomavicius and YoungOk Kwon, "Improving aggregate recommendation diversity using ranking-based techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 896–911, 2012. [Cited on pages 25 and 32.]

[103] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4511–4515, 2010. [Cited on page 25.]

[104] Adiya Abisheva, David Garcia, and Frank Schweitzer, "When the filter bubble bursts: collective evaluation dynamics in online communities," in *Proceedings of the 8th ACM Conference on Web Science*. ACM, 2016, pp. 307–308. [Cited on pages 25, 34, 36, 46, 49, 55, 67, 71, and 73.]

[105] Charu C Aggarwal and ChengXiang Zhai, "A survey of text classification algorithms," in *Mining text data*, pp. 163–222. Springer, 2012. [Cited on page 25.]

[106] Diana Maynard and Adam Funk, "Automatic detection of political opinions in tweets," in *Extended Semantic Web Conference*. Springer, 2011, pp. 88–99. [Cited on page 25.]

[107] Vasileios Hatzivassiloglou and Janyce M Wiebe, "Effects of adjective orientation and gradability on sentence subjectivity," in *Proceedings of the 18th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 2000, pp. 299–305. [Cited on page 25.]

[108] Peter D Turney and Michael L Littman, "Unsupervised learning of semantic orientation from a hundred-billion-word corpus," *arXiv preprint cs/0212012*, 2002. [Cited on page 25.]

[109] Alison Huettner and Pero Subasic, "Fuzzy typing for document management," *ACL 2000 Companion Volume: Tutorial Abstracts and Demonstration Notes*, pp. 26–27, 2000. [Cited on page 25.]

[110] Richard M Tong, "An operational system for detecting and tracking opinions in on-line discussion," in *Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification*, 2001, vol. 1, p. 6. [Cited on page 25.]

[111] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller, "Introduction to wordnet: An on-line lexical database," *International journal of lexicography*, vol. 3, no. 4, pp. 235–244, 1990. [Cited on page 25.]

[112] Vasileios Hatzivassiloglou and Kathleen R McKeown, "Predicting the semantic orientation of adjectives," in *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 1997, pp. 174–181. [Cited on page 25.]

[113] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86. [Cited on page 26.]

[114] Walaa Medhat, Ahmed Hassan, and Hoda Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014. [Cited on page 26.]

[115] Hanhoon Kang, Seong Joon Yoo, and Dongil Han, "Senti-lexicon and improved naïve bayes algorithms for sentiment analysis of restaurant reviews," *Expert Systems with Applications*, vol. 39, no. 5, pp. 6000–6010, 2012. [Cited on page 26.]

[116] Prem Melville, Wojciech Gryc, and Richard D Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 1275–1284. [Cited on pages 26, 27, and 48.]

[117] Prem Melville, Vijil Chenthamarakshan, Richard D Lawrence, James Powell, Moses Mugisha, Sharad Sapra, Rajesh Anandan, and Solomon Assefa, "Amplifying the voice of youth in africa via text analytics," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1204–1212. [Cited on pages 26 and 27.]

[118] Jonathan Ortigosa-Hernández, Juan Diego Rodríguez, Leandro Alzate, Manuel Lucania, Iñaki Inza, and Jose A Lozano, "Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers," *Neurocomputing*, vol. 92, pp. 98–115, 2012. [Cited on page 26.]

[119] Yung-Ming Li and Tsung-Ying Li, "Deriving market intelligence from microblogs," *Decision Support Systems*, vol. 55, no. 1, pp. 206–217, 2013. [Cited on page 26.]

[120] Vikas Sindhwani and Prem Melville, "Document-word co-regularization for semi-supervised sentiment analysis," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 2008, pp. 1025–1030. [Cited on page 26.]

[121] J. Ross Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986. [Cited on page 27.]

[122] Rong Hu and Pearl Pu, "Helping users perceive recommendation diversity.," in *DiveRS@ RecSys*, 2011, pp. 43–50. [Cited on pages 27 and 32.]

[123] Olfa Nasraoui and Bing Liu Brij Masand, "Advances in web mining and web usage analysis," in *8th international workshop on the Web, WebKDD*. Springer, 2006. [Cited on page 27.]

[124] Joydeep Ghosh, Yong Jae Lee, and Kristen Grauman, "Discovering important people and objects for egocentric video summarization," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1346–1353. [Cited on page 29.]

[125] Dietmar Jannach, Lukas Lerche, Fatih Gedikli, and Geoffray Bonnin, "What recommenders recommend–an analysis of accuracy, popularity, and sales diversity effects," in *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 2013, pp. 25–37. [Cited on pages 29 and 32.]

[126] Daniel Billsus and Michael J Pazzani, "User modeling for adaptive news access," *User modeling and user-adapted interaction*, vol. 10, no. 2-3, pp. 147–180, 2000. [Cited on pages 30 and 31.]

[127] Alan Said, Ben Fields, Brijnesh J Jain, and Sahin Albayrak, "User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm," in *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 2013, pp. 1399–1408. [Cited on page 30.]

[128] Alan Said, Brijnesh J Jain, and Sahin Albayrak, "Analyzing weighting schemes in collaborative filtering: cold start, post cold start and power users," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. ACM, 2012, pp. 2035–2040. [Cited on pages 30 and 31.]

[129] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell, "Part-based r-cnns for fine-grained category detection," in *Computer Vision–ECCV 2014*, pp. 834–849. Springer, 2014. [Cited on page 30.]

[130] Li-Tung Weng, Yue Xu, Yuefeng Li, and Richi Nayak, "Improving recommendation novelty based on topic taxonomy," in *Web Intelligence and Intelligent Agent Technology Workshops, 2007 IEEE/WIC/ACM International Conferences on*. IEEE, 2007, pp. 115–118. [Cited on page 31.]

[131] Leo Iaquinta, Marco De Gemmis, Pasquale Lops, Giovanni Semeraro, Michele Filannino, and Piero Molino, "Introducing serendipity in a content-based recommender system," in *Hybrid Intelligent Systems, 2008. HIS'08. Eighth International Conference on*. IEEE, 2008, pp. 168–173. [Cited on page 31.]

[132] Noriaki Kawamae, "Serendipitous recommendations via innovators," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 218–225. [Cited on page 31.]

[133] Chengcui Zhang, Xin Chen, and Wei-Bang Chen, "An online multiple instance learning system for semantic image retrieval," in *Multimedia Workshops, 2007. ISMW'07. Ninth IEEE International Symposium on*. IEEE, 2007, pp. 83–84. [Cited on page 31.]

[134] Neil Hurley and Mi Zhang, "Novelty and diversity in top-n recommendation–analysis and evaluation," *ACM Transactions on Internet Technology (TOIT)*, vol. 10, no. 4, pp. 14, 2011. [Cited on page 31.]

[135] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor, "Auralist: introducing serendipity into music recommendation," in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 13–22. [Cited on page 31.]

[136] Kai Zhou, Karthik Mahesh Varadarajan, Markus Vincze, and Fuqiang Liu, "Hybridization of appearance and symmetry for vehicle-logo localization," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE, 2012, pp. 1396–1401. [Cited on page 31.]

[137] Saúl Vargas and Pablo Castells, "Rank and relevance in novelty and diversity metrics for recommender systems," in *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 2011, pp. 109–116. [Cited on page 31.]

[138] Umberto Panniello, Alexander Tuzhilin, Michele Gorgoglione, Cosimo Palmisano, and Anto Pedone, "Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 265–268. [Cited on page 31.]

[139] Li-Chih Chen, Jun-Wei Hsieh, Yilin Yan, and Bo-Yuan Wong, "Real-time vehicle make and model recognition from roads," in *2013 12th Conference on Information Technology and Applications in Outlying Islands*, 2013, pp. 1033–1040. [Cited on page 32.]

[140] Yue Shi, Xiaoxue Zhao, Jun Wang, Martha Larson, and Alan Hanjalic, "Adaptive diversification of recommendation results via latent factor portfolio," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 175–184. [Cited on page 32.]

[141] Saúl Vargas, Pablo Castells, and David Vallet, "Explicit relevance models in intent-oriented information retrieval diversification," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 75–84. [Cited on page 32.]

[142] Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain, "Temporal diversity in recommender systems," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 210–217. [Cited on page 32.]

[143] Alejandro Bellogín, Javier Parapar, and Pablo Castells, "Probabilistic collaborative filtering with negative cross entropy," in *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 387–390. [Cited on page 32.]

[144] Abraham Silberschatz and Alexander Tuzhilin, "What makes patterns interesting in knowledge discovery systems," *IEEE Transactions on Knowledge and data engineering*, vol. 8, no. 6, pp. 970–974, 1996. [Cited on page 32.]

[145] Gideon Berger and Alexander Tuzhilin, "Discovering unexpected patterns in temporal data using temporal logic," *Temporal Databases: research and practice*, pp. 281–309, 1998. [Cited on page 32.]

[146] Balaji Padmanabhan and Alexander Tuzhilin, "A belief-driven method for discovering unexpected patterns.," in *KDD*, 1998, vol. 1998, pp. 94–100. [Cited on page 32.]

[147] Kleanthis-Nikolaos Kontonasios, Eirini Spyropoulou, and Tijl De Bie, "Knowledge discovery interestingness measures based on unexpectedness," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 5, pp. 386–399, 2012. [Cited on page 32.]

[148] 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*. Springer, 2007, pp. 40–46. [Cited on page 32.]

[149] 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*. ACM, 2010, pp. 257–260. [Cited on page 32.]

[150] Takayuki Akiyama, Kiyohiro Obara, and Masaaki Tanizaki, "Proposal and evaluation of serendipitous recommendation method using general unexpectedness.," in *PRSAT@ RecSys*, 2010, pp. 3–10. [Cited on page 32.]

[151] Andreas Flache and Michael W Macy, "Small worlds and cultural polarization," *The Journal of Mathematical Sociology*, vol. 35, no. 1-3, pp. 146–176, 2011. [Cited on page 33.]

[152] Marc A Smith, Lee Rainie, Ben Shneiderman, and Itai Himelboim, "Mapping twitter topic networks: From polarized crowds to community clusters," *Pew Research Center*, vol. 20, 2014. [Cited on page 33.]

[153] Dimitar Nikolov, Diego FM Oliveira, Alessandro Flammini, and Filippo Menczer, "Measuring online social bubbles," *PeerJ Computer Science*, vol. 1, pp. e38, 2015. [Cited on page 33.]

[154] Lada A Adamic and Natalie Glance, "The political blogosphere and the 2004 us election: divided they blog," in *Proceedings of the 3rd international workshop on Link discovery*. ACM, 2005, pp. 36–43. [Cited on page 33.]

[155] Michael Conover, Jacob Ratkiewicz, Matthew R Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini, "Political polarization on twitter.," *ICWSM*, vol. 133, pp. 89–96, 2011. [Cited on pages 33 and 34.]

[156] Alireza Hajibagheri and Gita Sukthankar, "Political polarization over global warming: analyzing twitter data on climate change," *ASE@ 360 Open Scientific Digital Library*, 2014. [Cited on page 34.]

[157] Marco Pennacchiotti and Ana-Maria Popescu, "Detecting controversies in twitter: a first study," in *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*. Association for Computational Linguistics, 2010, pp. 31–32. [Cited on page 34.]

[158] Yoonjung Choi, Yuchul Jung, and Sung-Hyon Myaeng, "Identifying controversial issues and their sub-topics in news articles," in *Pacific-Asia Workshop on Intelligence and Security Informatics*. Springer, 2010, pp. 140–153. [Cited on page 34.]

[159] M Anu Sree, P Athmigha, and L Sowmya Danalakshmi, "Controversy trend detection in social media," *Biometrics and Bioinformatics*, vol. 7, no. 2, pp. 55–60, 2015. [Cited on pages 34 and 55.]

[160] Ba-Quy Vuong, Ee-Peng Lim, Aixin Sun, Minh-Tam Le, Hady Wirawan Lauw, and Kuiyu Chang, "On ranking controversies in wikipedia: models and evaluation," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM, 2008, pp. 171–182. [Cited on page 34.]

[161] Rinaldo Kühne et al., "Political news, emotions, and opinion formation: Toward a model of emotional framing effects," in *Annual Conference of the International Communication Association (ICA), Phoenix, AZ*, 2012. [Cited on page 34.]

[162] James A Russell and Lisa Feldman Barrett, "Core affect, prototypical emotional episodes, and other things called emotion: dissecting the elephant.," *Journal of personality and social psychology*, vol. 76, no. 5, pp. 805, 1999. [Cited on page 34.]

[163] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis, "Quantifying controversy in social media," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 2016, pp. 33–42. [Cited on page 35.]

[164] Sean A Munson and Paul Resnick, "Presenting diverse political opinions: how and how much," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2010, pp. 1457–1466. [Cited on page 35.]

[165] Yelena Mejova, Amy X Zhang, Nicholas Diakopoulos, and Carlos Castillo, "Controversy and sentiment in online news," *arXiv preprint arXiv:1409.8152*, 2014. [Cited on page 35.]

[166] Rainer Reisenzein, "The schachter theory of emotion: two decades later.," *Psychological bulletin*, vol. 94, no. 2, pp. 239, 1983. [Cited on page 36.]

[167] Yu Ma, Xiaodong Gu, and Yuanyuan Wang, "Histogram similarity measure using variable bin size distance," *Computer Vision and Image Understanding*, vol. 114, no. 8, pp. 981–989, 2010. [Cited on page 46.]

[168] Tin Kam Ho, "Random decision forests," in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*. IEEE, 1995, vol. 1, pp. 278–282. [Cited on pages 47 and 67.]

[169] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert, "Norms of valence, arousal, and dominance for 13,915 english lemmas," *Behavior research methods*, vol. 45, no. 4, pp. 1191–1207, 2013. [Cited on page 49.]

[170] F Maxwell Harper and Joseph A Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 4, pp. 19, 2016. [Cited on pages 64 and 72.]

[171] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *information retrieval*, vol. 4, no. 2, pp. 133–151, 2001. [Cited on page 64.]

[172] Jure Leskovec and Andrej Krevl, "SNAP Datasets: Stanford large network dataset collection," `http://snap.stanford.edu/data`, June 2014. [Cited on pages 64 and 72.]

[173] Netflix, "Netflix prize data set," 2009. [Cited on page 65.]

[174] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 142–150. [Cited on pages 65 and 72.]

[175] Bo Pang and Lillian Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, p. 271. [Cited on page 65.]

[176] Ramanthan Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins, "Propagation of trust and distrust," in *Proceedings of the 13th international conference on World Wide Web*. ACM, 2004, pp. 403–412. [Cited on page 65.]

[177] David Martin Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011. [Cited on page 67.]

[178] Steven T Garren, "Maximum likelihood estimation of the correlation coefficient in a bivariate normal model with missing data," *Statistics & probability letters*, vol. 38, no. 3, pp. 281–288, 1998. [Cited on page 70.]

[179] Charles G Lord, Lee Ross, and Mark R Lepper, "Biased assimilation and attitude polarization: The effects of prior theories on subsequently considered evidence.," *Journal of personality and social psychology*, vol. 37, no. 11, pp. 2098, 1979. [Cited on page 79.]

[180] Seth Flaxman, Sharad Goel, and Justin Rao, "Filter bubbles, echo chambers, and online news consumption," *Public Opinion Quarterly*, p. nfw006, 2016. [Cited on page 79.]

[181] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al., *Introduction to information retrieval*, vol. 1, Cambridge university press Cambridge, 2008. [Cited on page 92.]

[182] Karen Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972. [Cited on page 92.]

[183] George A Miller, "The magical number seven, plus or minus two: some limits on our capacity for processing information.," *Psychological review*, vol. 63, no. 2, pp. 81, 1956. [Cited on page 95.]

[184] Dirk Bollen, Bart P Knijnenburg, Martijn C Willemsen, and Mark Graus, "Understanding choice overload in recommender systems," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 63–70. [Cited on page 95.]

[185] Eva Zangerle, Wolfgang Gassler, and Gunther Specht, "Recommending#-tags in twitter," in *Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings*, 2011, vol. 730, pp. 67–78. [Cited on page 97.]

[186] Henry Hsu and Peter A Lachenbruch, "Paired t test," *Wiley Encyclopedia of Clinical Trials*, 2008. [Cited on page 105.]

[187] Immanuel M Bomze, Marco Budinich, Panos M Pardalos, and Marcello Pelillo, "The maximum clique problem," in *Handbook of combinatorial optimization*, pp. 1–74. Springer, 1999. [Cited on page 113.]

[188] Clayton J Hutto and Eric Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Eighth International AAAI Conference on Weblogs and Social Media*, 2014. [Cited on page 114.]

CURRICULUM VITAE

NAME: Mahsa Badami

ADDRESS: Computer Engineering and Computer Science Department
Speed School of Engineering, University of Louisville
Louisville, KY 40292

EDUCATION:

Ph.D., Computer Science & Engineering, January 2013-April 2017
University of Louisville, *Louisville, Kentucky*

M.Sc., Computer Science & Engineering, January 2017-May 2017
University of Louisville, *Louisville, Kentucky*

M.Sc., Computer Engineering, Artificial Intelligence, 2009-2012
Shiraz University, *Iran*

B.Sc., Computer Engineering, 2004-2008
Islamic Azad University of Hamedan, *Iran*

PUBLICATIONS:

1. **M. Badami**, W. Sun, O. Nasraoui, *"Ratings, Reviews and Polarization"*, 2017 (Under Revision).

2. **M. Badami**, W. Sun, O. Nasraoui, P. Shafto, *"Detecting Polarization in Ratings: An Automated Pipeline and a Preliminary Quantification on Several Benchmark Data Sets"*, 2017 (Under Revision).

3. **M. Badami** and O. Nasraoui, *" Cross-Domain Hashtag Recommendation and Story Revelation in Social Media Posts"*, 2017 (Under Revision).

4. **M. Badami**, F. Tafazzoli, O. Nasraoui, *"A Data Science Case Study for Event Recommendation Using Data Storytelling and Fuzzy Approximate Reasoning"*, 2017 (Under Revision).

5. B. Abdollahi, **M. Badami**, G. C. Nutakki, O. Nasraoui, *"A Two Step Ranking Solution for Twitter User Engagement"*, Recommender Systems Challenge associated with ACM, 2014 Recsys14 Competition.

6. G. C. Nutakki, O. Nasraoui, B. Abdollahi, **M. Badami**, W. Sun, *"n, Distributed LDA based Topic Modeling and Topic Agglomeration in a Latent Space"*, 23rd International World Wide Web Conference, 7-11 April 2014- SNOW14 Competition.

7. **M. Badami**, N. Mozafari, A. Hamzeh, S. Hashemi, *"An Ensemble Game Theoretic Approach for MultiObjective Optimization"*, The European Journal of Artificial Intelligence, 2014

8. **M. Badami**, F. Tafazzoli, *"Event Recommendation Engine Challenge"*, KY-triwic (ACM_W), February 2014.

9. **M. Badami**, A. Hamzeh, S. Hashemi, *"An Enriched Game-Theoretic framework for Multi-Objective Clustering"*, International Journal of Applied Soft Computing, 2013.

10. **M. Badami**, A. Hamzeh, S. Hashemi, *"A novel Strategy Selection Method for Multi-Objective Clustering Algorithms using Game Theory"*, IJCSI Volume 9, Issue 3, May 2012.

11. **M. Badami**, A. Hamzeh, S. Hashemi, *"A Probabilistic Approach for Multi-Objective Clustering Using Game Theory"*, 16th Symposium on Artificial Intelligence and Signal Processing, 2012.

12. Q. Mirsharif, **M. Badami**, B. Salehi, Z. Azimifar, *"Recognition of Farsi Handwritten Digits using a small Feature Set"*, Proc. Of Int. Conf. on 3rd International Conference on Signal Acquisition and Processing, International Journal of Procedia Engineering, 2011.

WORK EXPERIENCE:

1. **Nationwide Insurance**, Columbus, OH, *May 2016-December 2016*
   Data scientist Intern.

2. **Knowledge Discovery and Web Mining laboratory**, Louisville, KY, *January 2013-Present*
   Research assistant.

HONORS AND AWARDS:

1. Guy Stevenson Award for excellent scholarship, leadership and academic performance by University of Louisville, 2017

2. Doctoral Award by CECS department, University of Louisville

3. Deans Citation Award for superior accomplishment by University of Louisville

4. 1st place at Bosch Device Detection Competition, KDD16

5. Arthur Riehl Award for excellent academic performance by University of Louisville, 2016

6. Tapia Doctoral Consortium Scholarship by NSF, ACM Richard Tapia Celebration of Diversity in Computing

7. SpeedUp competition award (cash prize) by Speed School of Engineering, UofL

8. First place Graduate Research Award 101st Kentucky Academy of Science

9. Top 3 selected app at National Day of Civic Hacking event, 2015

10. First place of poster competition in UofL Graduate Research Symposium
11. Best Poster winner of ACM_W at KY-triwic
12. Tuition Match Award of University of Louisville
13. Grad Cohort Celebration Scholarship, 2014
14. Grace Hopper Celebration Scholarship by (AT&T, LinkedIn and ABI), 2013-2015
15. Social Media Mentions Winner at National Day of Civic Hacking event, 2013
16. Semi-finalist of National Inform Mathematics & Computer Olympiads, Iran, 2003
17. Selected student in Iranian Best Students Congress in Tehran University, Tehran, Iran, 2002-2003
18. Ranked 1st in National Scientific Contest, Hamadan, Iran, 1996-1997

EXTRACURRICULAR ACTIVITIES:

1. Chair of Speed School of Engineering in Student Government Association (SGA), University of Louisville, 2015-2016
2. Director of Finance of Graduate Student Council (GSC), University of Louisville
3. Co-chair of Appropriation Committee in SGA, University of Louisville
4. President of Iranian Student Organization, University of Louisville, 2014-2015
5. Ranked 1st in Shiraz University inline speeds championship, Iran
6. Ranked 3rd in Shiraz University squash championship, Iran, 2010
7. Ranked 1st in ballet championship in west of Iran, 2006