Modeling Temporal Bias of Uplift Events in Recommender Systems

Thesis by
Basmah Altaf

In Partial Fulfillment of the Requirements

For the Degree of
Masters of Computer Science

King Abdullah University of Science and Technology, Thuwal, Kingdom of Saudi Arabia

May, 2013
The thesis of Your Full Name is approved by the examination committee

Committee Chairperson: Dr. Xiangliang Zhang
Committee Member: Dr. Xin Gao
Committee Member: Dr. Mikhail Moshkov
Modeling Temporal Bias of Uplift Events in Recommender Systems

Basmah Altaf

Today, commercial industry spends huge amount of resources in advertisement campaigns, new marketing strategies, and promotional deals to introduce their product to public and attract a large number of customers. These massive investments by a company are worthwhile because marketing tactics greatly influence the consumer behavior. Alternatively, these advertising campaigns have a discernible impact on recommendation systems which tend to promote popular items by ranking them at the top, resulting in biased and unfair decision making and loss of customers’ trust. The biasing impact of popularity of items on recommendations, however, is not fixed, and varies with time. Therefore, it is important to build a bias-aware recommendation system that can rank or predict items based on their true merit at given time frame.

This thesis proposes a framework that can model the temporal bias of individual items defined by their characteristic contents, and provides a simple process for bias correction. Bias correction is done either by cleaning the bias from historical training data that is used for building predictive model, or by ignoring the estimated bias from the predictions of a standard predictor. Evaluated on two real world datasets, NetFlix and MovieLens, our framework is shown to be able to estimate and remove
the bias as a result of adopted marketing techniques from the predicted popularity of items at a given time.
ACKNOWLEDGEMENTS

I am deeply indebted to my supervisor Dr. Xiangliang Zhang whose help, discussion, stimulating suggestions, and encouragement helped me in all the time of research for and writing of this thesis.

I would also like to thank all my professors at KAUST, in particular: Dr. Mikhail Mishkov, Dr. Xin Gao, Dr. David Keyes, Dr. Slim Alouni, and Dr. Basem Shihada.

I am particularly grateful for the assistance given by Dr. Faisal Kamiran for countless invaluable discussions, and practical support that helped me see this research to conclusion. I also wish to thank Dr. Indre Zliobaite for her contribution and valuable ideas in this research. My sincere gratitude and thanks to current members in MINE group at KAUST.

I am grateful to Marcin Rogowski, Imran Ansari and Shahid Hussain for teaching me how to use Latex.

I want to thank all my friends, especially Mohammed Farhan, Idris Aja, Daniel Binhem, Marcos Bracchitta, Amber Siddiqui, Iqra Mughal, Rishabh Dutta, Jagdish Vyas, Anas Daghistani, Basil Arkasosy, Hanan Syed, Konpal Ali, Amani AlOnazi, Enas Ahmed, Sally Ahmed, Tamour Javed, Usman Majeed and Agha Tahir for their help, love and support.

Last but not the least, I thank my parents Altaf Hussain and Raheela Bano, my brother Ahmed Hussain, and my sister Fatima Altaf for their constant encouragement, support and love in my academic and day to day life.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examination Committee Approval</td>
<td>2</td>
</tr>
<tr>
<td>Copyright</td>
<td>3</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>6</td>
</tr>
<tr>
<td>List of Abbreviations</td>
<td>9</td>
</tr>
<tr>
<td>List of Figures</td>
<td>10</td>
</tr>
<tr>
<td>List of Tables</td>
<td>11</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td></td>
</tr>
<tr>
<td>1.1 Background of Uplift Events and Recommender Systems</td>
<td>13</td>
</tr>
<tr>
<td>1.1.1 Uplift Events and Their Impact</td>
<td>13</td>
</tr>
<tr>
<td>1.1.2 Recommender Systems (RS)</td>
<td>14</td>
</tr>
<tr>
<td>1.1.3 Bias in Recommender Systems (RS) Introduced by Uplift Events</td>
<td>15</td>
</tr>
<tr>
<td>1.2 Motivation</td>
<td>17</td>
</tr>
<tr>
<td>1.3 Research Objective</td>
<td>21</td>
</tr>
<tr>
<td>1.4 Thesis Outline</td>
<td>22</td>
</tr>
<tr>
<td><strong>2 Related Work</strong></td>
<td></td>
</tr>
<tr>
<td>2.1 Bias Estimation - Discrimination-aware Data Mining</td>
<td>23</td>
</tr>
<tr>
<td>2.2 Main Types of Recommender Systems and Various Bias</td>
<td>26</td>
</tr>
<tr>
<td>2.3 Uplift Modeling in Marketing</td>
<td>29</td>
</tr>
<tr>
<td><strong>3 Bias Modeling and Recommendation</strong></td>
<td></td>
</tr>
<tr>
<td>3.1 Problem Setting</td>
<td>32</td>
</tr>
<tr>
<td>3.2 Bias Modeling</td>
<td>34</td>
</tr>
<tr>
<td>3.3 Bias-aware Recommender (BAR)</td>
<td>36</td>
</tr>
</tbody>
</table>
3.3.1 Offline Learning ........................................ 39
3.3.2 Online Learning ......................................... 39
3.3.3 Adaptive Learning ....................................... 40

4 Dataset Preparation ........................................... 44
  4.1 Features Collection ....................................... 45
    4.1.1 IMDB ................................................. 45
    4.1.2 TMDb API ............................................. 46
    4.1.3 MovieLens Dataset .................................. 46
    4.1.4 Netflix ............................................... 47
  4.2 Features Definition ..................................... 48
  4.3 Ready-to-use Data ....................................... 52

5 Experimental Evaluation ...................................... 55
  5.1 Bias Quantification ...................................... 56
  5.2 Temporal Aspect of the Bias .............................. 59
  5.3 Bias Correction ......................................... 62

6 Conclusions and Future Work .................................. 66
  6.1 Conclusions ............................................. 66
  6.2 Applications of Bias Aware RS ........................... 69
    6.3 Future Research Work .................................. 69
      6.3.1 Personalized RS .................................. 69
      6.3.2 Alternative Causes of Uplift Events .............. 71

References ..................................................... 74

Appendices ...................................................... 79
  A.1 MPAA Rating ............................................. 80
  B.1 My Contribution ......................................... 81
LIST OF ABBREVIATIONS

BAR Bias Aware Recommender
CF Collaborative Filtering
MAE Mean Absolute Error
RS Recommender Systems
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Self-re-enforcing cycles in RS due to uplift events in the life cycle of a</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>product</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>Oscar Bias: The Lord of the Rings</td>
<td>19</td>
</tr>
<tr>
<td>1.3</td>
<td>Popularity CDF of Oscar winners versus non-Oscar winners</td>
<td>20</td>
</tr>
<tr>
<td>2.1</td>
<td>Knowledge discovery and deployment in data mining</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>Discrimination-aware data mining</td>
<td>25</td>
</tr>
<tr>
<td>3.1</td>
<td>A model of biased popularity. Shaded circles denote observed variables,</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>transparent circles denote hidden variables</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Dataset preparation steps</td>
<td>45</td>
</tr>
<tr>
<td>5.1</td>
<td>Evaluation of Bias Aware Recommender (BAR) on NetFlix dataset</td>
<td>57</td>
</tr>
<tr>
<td>5.2</td>
<td>Evaluation of BAR on MovieLens dataset</td>
<td>58</td>
</tr>
<tr>
<td>5.3</td>
<td>Comparison of models with different bias estimation in offline, online</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>and adaptive learning framework</td>
<td></td>
</tr>
</tbody>
</table>
LIST OF TABLES

4.1 NetFlix subscribers growth ........................................... 51
4.2 Datasets selection ....................................................... 52
4.3 Statistics of NetFlix train and test sets ........................... 54
4.4 Statistics of MovieLens train and test sets ....................... 54

5.1 Bias correction measure (Mean Absolute Error (MAE)) by removing
the estimated bias from the training data in NetFlix dataset ........ 63
5.2 Mean and standard deviation of target values of test set instances . 64
5.3 Bias correction measure (MAE) by removing the estimated bias from
the training data in MovieLens dataset ............................... 65

6.1 User preferences for technology platform ........................... 70
Chapter 1

Introduction

Promotional deals, word of mouth and advertisements largely influence the sales of items. These activities are normally intended to increase the awareness of certain products and/or to attract people to buy them. With the advent of internet and advances in the field of information technology, the sharing of knowledge has become increasingly accessible in a number of different fields. This technological advancement has led to enormous data/information availability. As a result, users find it difficult to choose certain items due to a bewildering number of options. Nowadays many websites employ search engines and recommender systems to tackle the challenge of selecting the most relevant and beneficial information from the massive amount of online data. Search engines allow users to express their query in text form that is then processed and finally the most relevant items to the query are returned and displayed. Alternatively, a set of predicted items that best match the user’s interest are suggested by Recommender Systems (RS). This recommended list may include items that were previously unknown to users.

Recommender system (RS) predictions are impacted by several factors; genuineness of historical data, sufficiency of users’ profile data, and prediction algorithms. In this thesis, genuineness of data in recommender systems is studied. More specifically, popularity bias introduced by uplift events is modeled with a purpose to make bias-aware recommendation.
1.1 Background of Uplift Events and Recommender Systems

Popular items are usually ranked at a higher position in the recommended list of items to users. In this section, we introduce uplift events and how they boost the targeted item popularity in recommender systems.

1.1.1 Uplift Events and Their Impact

Uplift events are marketing events that are widely used to attract customers attention on certain products and create more demand in the consumer industry. Examples of such events are pre-release and post-release product advertisements, visual attractions, publicity on social networks, accolades such as Oscar awards for movies or songs, promotion deals and new marketing strategies. As the product gains more popularity due to an uplift event, it becomes more quoted, viewed and purchased, effectively increasing its customer base. The resulting effect is additional growth and higher sales.

Uplift events can be of two types: (a) on-purpose (promoting products via strategic marketing campaigns) or (b) consequential (accolades or award nominations). An on-purpose uplift event can be induced, for example, by publicizing a promotional offer for a product on social media, or by using visual attractions. As customers become aware of the promotional deals, they will be more interested in purchasing those products. For example, if the manufacturer of a shampoo announces that it will protect from hair loss (a common problem) and offers a complimentary hair conditioner with every purchase, this may attract more consumers to switch to this shampoo brand. In turn, this may decrease sales of other potentially better quality shampoos while uplifting the popularity and sales of the shampoo advertised with the promotional offer. Consequential uplift events can also be considered in the case
of a movie life cycle. If a movie gets nominated for or wins an Oscar award, there is an uplift in the number of views. This is because hundreds of millions of viewers are enticed by the merits of the awarding body, by the curiosity due to enhanced popularity, or by both. Naturally, this translates into a spike in box office ratings.

1.1.2 Recommender Systems (RS)

Online RS, an information technology service, have been employed in many e-commerce websites. They automate suggestion of items more likely to appeal and attract users. Users are often overwhelmed by the large number and diversity of available alternatives for the product they wish to buy. RS help the users to select easily and readily from the available choices by suggesting the most appealing items to them. For example, RS could suggest what clothes to buy, which places to visit for tourism, which movies to watch and what to eat in a restaurant.

There are two main types of RS: (i) personalized RS and (ii) non-personalized RS. Personalized RS recommend items by targeting on a specific user based on the user’s profile and his/her previous transactions. For every user, personalized RS present a tailored suggestion as all individuals have different tastes. On the other hand, non-personalized RS suggest items that are globally popular in the consumer market and may be interested by most of the users, based on total sales or total revenue of those products. Thus, in non-personalized RS, the same items are recommended to every user without considering any user profile attributes.

Recommender systems increase sales of items and have a positive influence on the growth of a consumer market. Major online industries, for instance, Facebook, Netflix, Amazon, Trip Advisor and Yahoo! Music deploy RS to increase their views/sales by compelling more customers. RS boost e-commerce sales in three ways: they (a) entice viewers of a website to find something relevant, hence turning a browsing customer into a buyer, (b) suggest items similar to the ones purchased
or in the shopping cart of the user, thus escalating cross-selling, and (c) create a
trust between a customer and the recommender system. This makes loyal customers
return and visit often to purchase other or similar items because of the trust in the
RS accumulated from their previous usage experience [3].

However, not all RS are trustworthy. Collaborative filtering, a well-known tech-
nique of personalized RS matches user profile with other users who share similar
characteristic to find the best products that match a user’s interests. However, in
hedonistic industries, e.g., books or movies, the collections are so large that it is hard
to find similar user profiles who have rated exactly the same movies/books before.
Attackers could introduce fake user profiles that pretend to be similar to target user to
force the recommender system to produce results that are lucrative to them, thereby
making RS predictions untrustworthy [5, 6]. This, in turn, will diminish users trust
and confidence in RS predictions as they may perceive the results as unfair or biased.

Lack of trust in RS may impact their usage badly, specifically by bringing a
feeling of dissatisfaction in users when they discover violations of consumer rights or
anti-discrimination laws [7, 8]. For example, reports in [9, 10] state that Google does
breach users trust by promoting links that refer to its own services or products. It was
debated that customers should be aware of the product prioritization methodology
of a recommender system in order to build their trust in the system. Vice president
of the European Commission Joaquin Almunia stated on 21 May, 2012 that Google is
involved in four potential abuses of dominance (violation of European laws), the first
one being preferential treatment in the display of its own vertical search services [10].

1.1.3 Bias in RS Introduced by Uplift Events

Recommender Systems results may get influenced and biased by an uplift event of the
item as it induces popularity. There are two contradictory views about RS impact
on diversity of sales [3]. One belief is that RS increase sales diversity by helping
customers explore new products \[4\]. The contrary belief is that RS create a self-re-enforcing cycle of already popular items \[3\] as shown in Fig. 1.1. Concentration of sales at the aggregate level causes a decrease in sales diversity. It may satisfy consumer needs by suggesting the right and relevant items, but ones that tend to be globally popular already. As already discussed, there are two types of RS: (i) personalized, and (ii) non-personalized. In this thesis, non-personalized RS are the main focus because they rely more on the product popularity (high volume of sales or views) than personalized RS.

![Figure 1.1: Self-re-enforcing cycles in RS due to uplift events in the life cycle of a product](image)

Artificial popularity induced by uplift events in recommender systems is undesirable both to customers and to the producer industry in the long run, as it may
produce a rich-get-richer effect that will lead to biased predictions. Consequently, a narrow range of products will be offered to users by allowing only the promotion of already popular items. This will impact hugely on niche producers that are not so famous and produce a Long Tail effect \[11\]. Moreover, consumers may be underserved if there are non-recommended products that match their requirement better than the popular and hit ones \[3\]. In \[12\], Hosanagar and Fleder argued that because common recommenders recommend products on the basis of total sales and ratings by consumer, they cannot recommend products with limited historical data, even if they have favorable ratings. This is one of the reason that creates rich-get-richer phenomenon for popular products and vice-versa for unpopular ones, thus reducing diversity.

1.2 Motivation

From the available NetFlix\footnote{NetFlix system is a movie rental, online streaming and recommender system. It was founded in 1998 and followed an incremental growth in its number of subscribers and movies data set.} Prize data set \[13\], we analyzed the behavior of the growth of the ratings count for the movie series, The Lord of the Rings. The Lord of the Rings is a film trilogy consisting of three adventure/fantasy films with the following subtitles: The Fellowship of the Ring, The Two Towers and The Return of the King, released in years 2001, 2002 and 2003, respectively. Let us denote them by LOR-2001\footnote{An initial version of LOR-2001 was released on 17th Dec 2001. On 12 November, 2002, an extended version edition of LOR-2001 was released on VHS and DVD, with 30 minutes of new material, added special effects and music, plus 20 minutes of fan-club credits, totalling to 219. In NetFlix system, LOR-2001 has ratings data from Aug 2002. (Source: Wikepedia)} LOR-2002\footnote{LOR-2002 was initially released in Dec 2002. However, its VHS and DVD version were released on 26 August, 2003 in the United States. Also, an extended version of LOR-2002 was released on 19 November,2003. (Source: Wikepedia)} and LOR-2003\footnote{LOR-2003 was released in December 2003 and it’s theatrical version was released on 25 May, 2004. (Source: Wikepedia)} respectively. All three films of the series were nominated for best picture Oscar award in the 74th, 75th and 76th Academy awards, respectively. However, only the last movie of the series, The Lord of the
Rings: the Return of the King, successfully qualified for this award. If we observe the rating distribution graphs of years 2001 and 2002, it is evident that in the first few years after the release of LOR-2001 and LOR-2002, the movies suffered bias in the number of ratings and remained relatively unpopular until the Best Picture Oscar award was declared for LOR-2003, as shown in Fig. 1.2.

In Fig. 1.2 (a), and (b), the x-axis represents the date on which the ratings were captured, while the y-axis represents the number of ratings and the average rating, respectively. It is important to consider that, here, popularity is reckoned as a measure of the number of ratings for a movie received on the given date. It is clear from Fig. 1.2 (a) that the popularity of LOR-2001 was uplifted after its successor movie LOR-2003 won the Best Picture Oscar award in Feb 2004. LOR-2002 was released on VHS and DVD on 26 August, 2003 in the United States. Therefore, we see a sudden spike in Aug-Sep 2003 for LOR-2002. After the initial few months publicity, the number of ratings is reduced to around 200. However, LOR-2002 witnessed a gradual uplift in number of ratings after LOR-2003 won the Oscar award. Thus, the award brought an impulsive and radical shift by prompting an uplift in the number of ratings of the film trilogy. LOR-2001 took 3 years before it actually started earning high revenue.

Our research objective highlights such ranking discrimination, which promotes bias resulting from Oscar awards. A fair recommender system should be free from sudden drifts and should be capable of recommending good movies at the right time, i.e., soon after their release. Our claim is that if the movie gets a good average rating, which is a measure of its quality according to users, then it should be popular without waiting for any uplift event, e.g., Oscar award. To support this claim, in Fig. 1.2 (b) we note that the daily average ratings assigned to these movies by viewers continued to be 4-5 on average. It shows that people rated LOR-01 and LOR-02 reasonably high all the time, and there were no sudden rating changes that could be attributed to the Oscar of LOR-03.
Figure 1.2: Oscar Bias: The Lord of the Rings
In Fig. 1.3 the cumulative distribution function of the popularity of Oscar winning movies is presented. It is apparent that the Oscar winning movies (shown by red line) are always more popular than the non-Oscar movies, even the most popular non-winners (represented as blue line). Considering the fact that only a small fraction of movies get Oscar award; should we conclude that all movies that do not get the Oscar are bad movies and their content does not have enough merit? It is hard to believe and unfair to consider movies without the Oscar award automatically inferior content-wise to the movies that have got the award.

One may argue that Oscar winning movies are always the best movies released during that period and it is justified to recommend them more frequently than other movies. We are not denying the fact that Oscar winners deserve people’s attention. Our target is to model the bias introduced to the popularity of movies due to Oscar award, e.g., extra views due to curiosity and rich-get-richer effect. Quantification of such bias enables RS to make bias-aware recommendation.
1.3 Research Objective

The objective of this thesis is to study and measure the propagating impact of uplift events of products which, when input to RS, produce biased predictions. We consider the NetFlix Prize data set [13] and Movie Lens Data Set [14] as a case study for carrying out experiments, while considering an Oscar award as an uplift event that brings a sudden drift in the popularity of movies. We model and quantify the bias of products that observed an external uplift event. We then formulate a bias aware predictive model with the bias measured and constrained in order to neutralize the biased impact of uplift events on data and predict future popularity correctly. This bias aware model can be used as a building block in RS to produce predictions free of bias caused by uplift events. In this thesis, non-personalized RS are focused as they use the popularity of items as a measure of total sales or aggregate revenue in the overall market. In future study, this work can be extended to personalized RS.

Some movies’ popularity may be highly impacted by an Oscar award nomination, while other movies may observe only a very slight incremental change in demand. Also, the popularity of an item varies with time. Estimating the temporal and varying bias in different uplifted items’ popularity is a complex and novel problem hither to un-researched to a significant degree. Another challenge of this study is the data collection for study. Two principal sources of movie ratings, i.e., NetFlix Prize [13] and MovieLens [14] datasets contain limited information. Additional movie features like budget, runtime, release date are not readily available and had to be collected from different data sources in order to carry out the comprehensive study.

This thesis makes the following novel contributions:

1. Analyze and illustrate the bias problem in content based RS;

2. Study and formulate a new research problem for RS with the appropriate coinage: Bias Aware Recommenders (BAR);
3. Design an algorithm for estimating the instance level bias of an item in a given time period;

4. Present a methodology for constructing a bias aware predictive model that estimates the bias from the training data or for the learned model which can then be used as a building block for trustworthy and fair RS.

1.4 Thesis Outline

The rest of the thesis is organized as follows: Chapter 2 discusses related work and reviews the state of the art of bias estimation, RS, and uplift modeling in marketing. Chapter 3 formulates the problem setting and framework for modeling the bias, and then introduces an algorithmic approach to bias aware recommenders (BAR). Chapter 4 presents the data set features and preparation steps in detail. Chapter 5 reports experimental evaluations and makes discussion. Finally, Chapter 6 concludes the key contributions and novelty of our thesis problem, and points out challenges and expansions of our current work for future study.
Chapter 2

Related Work

The problem studied in this thesis is associated with three interesting research domains. First, it estimates bias in prediction models which is closely linked to discrimination aware data mining techniques. Second, from the application viewpoint, this study can be used as a building block in recommender systems. Last, from the problem setting and formulation viewpoint, it remotely links to specific problem settings in marketing such as uplift events. We will discuss the related work in these three domains in this chapter.

2.1 Bias Estimation - Discrimination-aware Data Mining

Data mining allows extraction and discovery of knowledge that is potentially useful but unknown, hidden in data sources \[15\]. The extracted knowledge is then deployed to make decisions, as shown in Fig. 2.1.
During knowledge deployment, one concern is the unfair use of the discovered knowledge in making discriminatory decisions about the items/people [16]. Discrimination refers to unfair treatment of people/items based on their belonging to a special group or category without considering individual merit [16]. If the data is intrinsically biased against some parameter(s), e.g., female gender, or foreign nationality, then the model built on such data will also exhibit discriminatory behavior [17].

Discrimination aware data mining is the study of discrimination present in historical data with the focus on obtaining results that are free from any discriminatory bias. This field has two potential dimensions to explore (shown in Fig. 2.2) which are: (i) detection of discrimination in datasets, to discover direct or indirect discriminatory decisions hidden in collected records, and (ii) discrimination prevention to eliminate the effect of bias due to discrimination by use of (a) pre-processings of datasets before building a model, (b) in-processing of model learning, and (c) post-processing after the model is built [17].

Our study conceptually relates to the second dimension of discrimination aware study, i.e., discrimination prevention, as our ultimate goal is to have predictive models that adhere to externally given constraints. The three types of discrimination prevention techniques are discussed in detail in the following passage.
Pre-processing converts source data into a form that cleans the discriminatory behavior from the original input dataset to ensure the impartiality of data. After the transformation of the original dataset, any standard algorithms in the field of data mining can be applied to obtain fair decision model from the unbiased dataset [17]. Main techniques of pre-processing discriminated data are discussed in [18], for example, distortion of data by eliminating attributes that are the source of bias, by changing the class labels for some of the biased instances to ensure the data is free from discrimination, by assigning weights to instances to neutralize the impact of biased features, or by sampling carefully to ensure the selected data is bias free.

In-processing techniques, change the data mining algorithms using an anti-discrimination procedure that results in models containing only unbiased rules [17]. For example, Calder and Verwer proposed to modify the probability of getting positive decisions to obtain fair recommendations using naive Bayes classifier [19].
Post-processing modifies the resulting biased model built from the original data and a standard data mining algorithm to remove bias effects to correct its decision criteria, instead of cleaning the original dataset or altering the standard data mining algorithms [17]. An example of postprocessing is discussed in [20], which proposes a confidence altering approach for classification rules inferred by the rule-based classifier CPAR (classification based on predictive association rules) algorithm.

Of all the discrimination prevention techniques, this work weakly resembles the in-processing technique. It presents an algorithm that makes fair predictions using bias aware settings. In discrimination related work, it is assumed that bias is constant for all instances in each period. However, one of the contributions of this thesis is to frame the bias at instance level of an item in different time periods (as different items show different bias effects) and to design a framework for bias aware recommendation.

2.2 Main Types of Recommender Systems and Various Bias

From the application perspective, this work can be integrated easily in recommender systems. RS are used for suggesting movies, books, clothes, music, hotels, etc., and therefore, play an important role in ecommerce, advertisements, entertainment, news, business, travel and tourism, etc., by suggesting the right item and diverse options to users based on their taste [21, 4]. As discussed already in Chapter 1, RS can be broadly classified into (a) personalized RS, and (b) non-personalized RS. Although we are focusing on non-personalized RS in this thesis, the proposed algorithmic approach is applicable to build personalized content-based recommenders.

Generally RS employ two main types of techniques:

1. Content Analysis

Content Analysis offer choices to customers based on the description of items
as well as users preference profile \cite{22}. Items are recommended if the content information matches a user’s preference \cite{22}, which is usually profiled from: (a) explicit information that is provided by the user in website registration form or survey, and (b) implicit information that is gathered in an indirect way, e.g., number of clicks on an item, purchase history \cite{23}, mouse movements \cite{23}, purchase and return rates of products; with more purchase indicating interest, and high return rates indicating dislike \cite{22}.

Collaborative Filtering (Collaborative Filtering (CF)): The principle of CF is that similar users have similar taste. Thus, CF compares user preferences to other users by weighing all the users’ similarity with the active user and then selecting a subset of users who are highly correlated with the target user \cite{4}. Items that are attractive to and liked by the selected similar users are then recommended to the target user. It is a widely popular choice in RS as it considers user’s profile to filter information by using other users’ historical preferences \cite{24}.

In the problem setting of this thesis, content-based RS are considered since the popularity of an item depends on the item contents and the bias occurs due to special characteristics of an item (e.g., an Oscar award). In a CF approach, user similarity is studied to recommend similar items to a group of users. Removing bias or discrimination with respect to a user group (or characteristics) is an alternative scenario, which is different and beyond the scope of this thesis study.

It is undesirable to make decisions based on biased information. Providing neutral information is essential in recommendation. The enhancement of neutrality in recommender systems has been considered in a recent study \cite{25}. The research target is to improve prediction accuracy by allowing the user to define a viewpoint which he thinks is the source of adding bias in the given data. By defining statistical independence between the target viewpoint and the recommendation result, neutral recommender systems is built. For instance, if a user is interested in watching any
good movie irrespective of the release year, he can define release year as a view point.

The recommended movie options are then not biased to the information of release year, rather these movies are recommended based on other features.

The authors of [25] argue for a single target viewpoint, as RS cannot be built by neutralizing the effect of all attributes, since then recommendation would not be possible. However, in this research problem, more than one viewpoint is needed to eliminate the impact of a sudden change in the popularity of an item arising from an uplift event such as an Oscar award, which affects multiple temporal features of an item (e.g., average rating, viewers count). Unlike the neutralized RS which produces recommendations free from the bias of a single viewpoint, this thesis aims to estimate the bias itself and use it further to make bias free predictions. Moreover, we consider that the bias is different for each item and varies with time. To the best of our knowledge, it has not been studied before.

Temporal dynamics in Collaborative RS is studied by Koren in [26]. Their research objective is to improve prediction accuracy of user ratings in collaborative filtering RS by tracking change in customers’ interest in products over time. It deals with (1) user-biases’ ($b_u$) deviation over time; (2) item biases’ ($b_i$) deviation over time; and (3) user preferences’ ($p_u$) deviation over time. To study this problem, NetFlix movie dataset is divided into 30 bins spanning all days in the dataset by considering each bin to occupy roughly 10 consecutive weeks of data.

Our work also deals with temporal bias, but the research objective is different. Unlike the work in [26] which aims to improve prediction accuracy for user rating by including the temporal bias, our work assumes that bias occurs in RS due to uplift event in the life of certain items that causes an artificial popularity and will bias the rankings of items by RS. To study our problem, we also divide the data into week and month bins for NetFlix and MovieLens dataset. However, the problem settings of our work is different, as we aim to estimate true popularity of an item by estimating the
bias correctly using the proposed algorithm at different time instances.

2.3 Uplift Modeling in Marketing

As the aim of this thesis is to estimate the bias due to an uplift event, this work relates to modeling the effects of actions in marketing using online advertisements. Uplift modeling is a term coined to predict the true impact of an event, (e.g., marketing action) in user behavior, also known as incremental behavior. It aims to target those customers who will be triggered positively by a marketing campaign such as offering a discount in price or a complimentary gift with the product, or by contacting people to advertise products (i.e., direct marketing). These people are termed 'persuadable'. Another group of people are those who will purchase products any way and typically would not require any incentives (i.e., marketing campaigns) to buy. As a result, the need to spend resources incentivizing such group of people is nonexistent and sales would therefore not be incremental in this case [27].

Uplift modeling is done by comparing the two groups of people, the ones who are the target of marketing campaigns (treated group), and the others who were not subject to marketing action uplift (control group) [28]. In other words, uplift modeling actually simulates and measures the difference it makes by swaying the minds of people from the targeted group (i.e., treated group) positively [15], while leaving out the group of people who will buy anyway, and are either not influenced or negatively influenced by a marketing campaign. The resulting benefit is a dramatic improvement in return of investment, reduction in marketing cost, increase in cross-selling and relevancy along with an improvement in retention [29, 30] as uplift is focused on incremental responses of the selected individuals only [30].

The most basic and na"ıve approach to build an uplift model is to define a partition between the treated and the control group via setting of a partition field \( P \) (0 for
control, 1 for treated). Two separate models are then built on each of the groups. The uplift impact on the test dataset is then measured by subtracting the prediction score probabilities of the control model from the ones predicted by the treatment model as shown in Eq. 2.1.

\[
Uplift \text{ Measure} = \text{Probability (Purchase|treatment)} - \text{Probability (Purchase|no treatment)} \tag{2.1}
\]

The above basic approach offers the advantage of being simple but has some limitations. It does not address all the challenges in uplift modeling. For instance, the models are built on different datasets, and hence cannot measure variation in prediction due to uplift properly for the test dataset. More sophisticated techniques such as uplift decision trees are defined in [31], whereas [27] discusses different split criteria and pruning techniques in an uplift decision tree to make it more optimal. Modified Naïve Bayesian classifier and logistic regression are introduced in [32] to build tree based classifiers on real data and using real marketing data for application verification. Microsoft Research defines an approach of building a decision tree on the whole population, i.e., control and treated group, followed by forcing a treatment variable to be used as a split criterion on each node [33].

This thesis study borrows the concept of partitioning the target and control group from uplift modeling in our problem context. It partitions the dataset (movies dataset as example) into two groups. The first partition of the dataset contains items whose popularity is affected by the uplift event (Oscar award in case of movies, for example), and the other partition contains those items that did not observe an uplift event. To define a base case of experimental study, this work also builds two separate models on different datasets, and predict the accuracy of the test set by combining the accuracy scores of both models. Unlike the uplift modeling approach, one of the main
contribution of this thesis is that we measure the uplift for each individual item at different time moments of its life cycle and treat it simultaneously.
Chapter 3

Bias Modeling and Recommendation

In this chapter a formal description of the research problem is given and a model for quantifying the bias due to presence of an uplift event is proposed. A bias-aware recommender (BAR) algorithm to learn the bias model is then presented.

3.1 Problem Setting

Consider a set of items $I = \{i_1, i_2, i_3, ..., i_k, ..., i_m\}$ denoting $m$ items, where each item $i_k$ is rated or viewed in a given time $t$, where $t = \{t_1, t_2, t_3, ..., t_j, ..., t_n\}$, $1 \leq k \leq m$, and $1 \leq j \leq n$. The popularity of an item (i.e., views or sales count) varies with time, and attains an artificial increase after the occurrence of an uplift event $s$. Therefore, each item $i$ popularity deviance can be modeled by grouping the rating data into $n$ identically sized time bins, where each time bin represents an instance of the calendar, e.g., day, week, and month.

Without loss of generality, we consider an item $i$ is described by two kinds of features:

- Stationary Features, $X(i)$, are the attributes whose values stay fixed in each time bin for the item $i$. For instance, in the case of a movie, director, genre,
and actors of a movie stay fixed no matter in which week the movie’s popularity is modeled.

- Temporal Features, $Z^t(i)$, are the attributes whose values do not remain fixed in each time bin for the item $i$, but tend to vary with time. For example, average rating per week, count of ratings per week, and subscribers count. Thus, temporal features capture the drift in items’ popularity as a function of time.

The sensitive attribute $s$ is also a time varying attribute,

- $s^t(i) = 0$ when item $i$ did not observe an uplift event at $t$;
- $s^t(i) = 1$ when item $i$ observes the event at $t$.

We define the item instance $e_{it}$ to be an instance of an item $i$ in time bin $t$, and denote the popularity of this ‘instance’ in time bin $t$ by $y_{it}$.

In order to rank items in the recommendation list at time $t$, the popularity of each item is predicted. Items then are ranked by popularity $y_{it}$. A recommendation for the popularity of items is unbiased if

$$p(y_{it}|e_{it} = [X(i), Z^t(i), s^t(i) = 0]) = p(y_{it}|e_{it} = [X(i), Z^t(i), s^t(i) = 1])$$ (3.1)

Eq. 3.1 defines the research objective of this thesis. It states that a predictive model provides an unbiased recommendation if the probability for an item to be viewed at a given time remains the same value regardless a sensitive (uplift) event $s$ has happened or not.

In the study of recommender systems, this work assumes the unbiased popularity of an item to be the count of views a movie received without the uplift event. Unbiased models can be learned on data with no uplift event using regression techniques such as Multiple Linear Regression. In addition, a bias model is learned from the biased popularity due to an uplift event.
3.2 Bias Modeling

This section presents the proposed framework to calculate the observed popularity of an item as summation of its true merit (popularity) and an induced bias effect (due to an uplift event). Let $y$ be the observed popularity, $\mathcal{M}$ is the model learned on unbiased data that predicts true merit of a movie, $\mathcal{D}$ is the model that estimates the bias and $s$ represents the binary uplift attribute, such that

$$y = \mathcal{M} + s\mathcal{D}, \quad s \in \{0, 1\}, \quad (3.2)$$

The uplift attribute $s$ is active only for those items that observed an uplift event (for instance, Oscar award in the case of movies). For $s = 1$, the observed popularity is composed of true popularity $\mathcal{M}$ and uplift bias effect $\mathcal{D}$. On the other hand, for $s = 0$ (i.e., when uplift event is absent); the observed popularity is measured from the true merit model $\mathcal{M}$ only, as shown in Fig. 3.1.

Figure 3.1: A model of biased popularity. Shaded circles denote observed variables, transparent circles denote hidden variables.
Let us first discuss the proposed bias model in this section, and present the learning of $\mathcal{M}$ and $\mathcal{D}$ in the next section. Considering the bias in different scenarios, we have different models of $\mathcal{D}$:

1. $\mathcal{D}$ with constant bias, $\mathcal{D}_0$

2. $\mathcal{D}$ with item related bias, $\mathcal{D}_X$

3. $\mathcal{D}$ with item and time related bias, $\mathcal{D}_{X,Z}$

1. **Model $\mathcal{D}$ with constant bias:** Let us assume that the bias introduced by the uplift attribute $s$ is constant for each instance at all the time. It considers that an uplift event $s$ has a fixed impact on the popularity of items and can be generalized by a bias constant $\mathcal{D}_0$. An example for the constant bias effect can be considered for a movie which after winning the Oscar award may introduce a fixed amount of publicity. This would attract fixed view counts all the times in the future by viewers who may be curious to watch a winning movie irrespective of its contents. Eq. 3.2 then reduces to

   $$y = \mathcal{M} + s \cdot \mathcal{D}_0, \quad s \in \{0, 1\}$$  \hspace{1cm} (3.3)

   where $\mathcal{D}_0$ is a constant.

2. **Model $\mathcal{D}$ learned on item stationary features:** Here assume that bias in the popularity of items after the occurrence of uplift event $s$ can be modeled as a function of movies’ stationary features only and does not vary with the temporal features. That is to say, temporal features have no relation with the bias. For instance, consider a movie of Science Fiction genre wins an Oscar award. In future times, Oscar effect on this movie may attract only those viewers who are fan of Science Fiction genres regardless of the amount of elapsed time since movie release. Therefore, time related factors like movie age might not affect the
viewers’ count and hence are ignored. $\mathcal{D}_X$ can be learned by using a regression algorithm, e.g., the Support Vector Regression.

$$\mathcal{D}_X = f(X \mid s = 1) \quad (3.4)$$

Eq. 3.4 models the target bias as a function of stationary features of those items’ instances whose popularity was impacted by an uplift event $s$.

3. **Model $\mathcal{D}$ learned on items’ stationary and temporal features:** The most realistic assumption is that the bias depends on the movies’ stationary features $X$ as well as the temporal features $Z^t$. For Oscar winner movies, additional viewers will depend on the elapsed time since the movie release, as well as on the contents of the movie. For instance, if the movie was released eleven months before the Oscar ceremony, maybe most of the people have already seen it or consider it to be an old movie and will not watch regardless of the Oscar. If, on the contrary, the movie was released two months before the Oscar ceremony, it may attract more new viewers after winning the Oscar because it is a new movie. Item and time related bias due to uplift event occurrence can be computed as follows:

$$\mathcal{D}_{X,Z} = f(X, Z^t \mid s = 1) \quad (3.5)$$

### 3.3 Bias-aware Recommender (BAR)

In this section, Bias aware Recommender (BAR) algorithm is presented. It is designed to learn the ‘true merit’ model $\mathcal{M}$ and the ‘bias estimation’ model $\mathcal{D}$, and to rank the recommended items by the predicted unbiased popularity. As introduced by Eq. 3.2 described in Section 3.2, the popularity of an item is split into two parts:
1. true popularity of an item $i$ according to the merit of its content, modeled by $\mathcal{M}$;

2. time varying bias in the popularity of an item $i$ induced by uplift event $s$, modeled by $\mathcal{D}$.

The dataset comprises of biased as well as unbiased data. In order to learn $\mathcal{M}$ and $\mathcal{D}$, we construct two datasets $A$ and $B$ by splitting the given data; dataset $A$ containing all the item instances for which uplift event $s$ is absent, while dataset $B$ contains all the item instances that are affected by uplift, and are characterized with the value of $s = 1$. Formally, for $i = 1, 2, \ldots, m$ and $t = 1, 2, \ldots, n$,

$$A = \{e_{it}, y_{it}\}$$

where $e_{it} = [X(i), Z^t(i), s^t(i) = 0]$ represents item instances, and $y_{it}$ is its corresponding popularity.

$$B = \{e_{it}, y_{it}\}$$

where $e_{it} = [X(i), Z^t(i), s^t(i) = 1]$ represents item instances, and $y_{it}$ is its corresponding popularity.

Each of these datasets $A$ and $B$ are further split into training and test set by considering the last few weeks data instances as test set, while the initial weeks’ data is part of the training set. The models $\mathcal{M}$ and $\mathcal{D}$ are then built using regression analysis techniques, e.g., Linear Regression, Support Vector Regression or Neural Network.

Model $\mathcal{M}$ is learned on item instances in training set $A$ (non-uplift items), to estimate the true merit of an item as follows:

$$\mathcal{M} : e_{it} \mapsto y_{it} \quad \{e_{it}, y_{it}\} \in A_{Train}$$

(3.6)

The resulting model $\mathcal{M}$ is an unbiased model and is capable of estimating only the true (unbiased) count of views for any set of items.
The learning of $\mathcal{D}$ needs the desired target of bias, which is implicit in the observed popularity. We define learning the target of bias to be

$$b = y_B - \hat{y}_B,$$

where $y_B$ is the observed popularity of the item instances in $B_{Train}$, and $\hat{y}_B$ is the predicted popularity when $\mathcal{M}$ is applied to the item instances in $B_{Train}$.

As mentioned in the previous section, we have 3 different bias models.

1. $\mathcal{D}$ with constant bias, $\mathcal{D}_0$: When learning $\mathcal{D}_0$, we consider the average bias in all item instances. $\mathcal{D}_0$ is estimated as

$$\mathcal{D}_0 = mean\{y_{it} - \hat{y}_{it}\} = mean\{y_{it} - \mathcal{M}(e_{it})\},$$

where $\mathcal{M}$ is the model learned in Eq. 3.6, and $e_{it}$ is the item instances of training set of $B$.

2. $\mathcal{D}$ with item related bias, $\mathcal{D}_X$: It is learned by applying regression model on a biased training set $\hat{B}_X$.

$$\hat{B}_X = \{X(i), b_{it}\},$$

where $b_{it} = y_{it} - \hat{y}_{it} = y_{it} - \mathcal{M}(e_{it})$, $X(i)$ is the stationary features of $e_{it}$ (item instances of training set of $B$), and $\mathcal{M}$ is the model learned in Eq. 3.6.

Then, $\mathcal{D}_X$ is learned as,

$$\mathcal{D}_X : X(i) \mapsto b_{it}.$$

3. $\mathcal{D}$ with item and time related bias, $\mathcal{D}_X,Z$: It is learned by applying regression model on a biased training set $\hat{B}_{X,Z}$.

$$\hat{B}_{X,Z} = \{e_{it}, b_{it}\},$$

where $b_{it} = y_{it} - \hat{y}_{it} = y_{it} - \mathcal{M}(e_{it})$, $e_{it}$ is the item instances of training set of $B$, and $\mathcal{M}$ is the model learned in Eq. 3.6.

Then $\mathcal{D}_{X,Z}$ is learned as,
$D_{X,Z} : e_{it} \rightarrow b_{it}$.

For different application perspectives, we design three different learning frameworks:

1. Offline Learning;
2. Online Learning;

### 3.3.1 Offline Learning

Offline learning provides a single model $M$ and $D$, learned from a fixed dataset. Once learned, the model is then used to predict popularity of new items without any changes. Offline learning is applicable in situations where the relation between the input and output data does not evolve with time and the same training model can be used in future for prediction on test set.

The training and testing data are decided by a time bin $t_{\text{split}}$, such that training data contains all movies instances before $t_{\text{split}}$ while test set contains all movies data after $t_{\text{split}}$. Then models $M$ and $D$ are built on training data using steps 5 – 11 as defined in Algorithm 1. Prediction and recommendation are evaluated on the test sets by following steps 12 – 15 of Algorithm 1.

### 3.3.2 Online Learning

Online learning is useful in scenario when initial training data is not enough for building an accurate model for future predictions and the relation between the input and the target variable is continuously evolving with time. The more the training data, the more robust the training model will be for future predictions.

In the initial settings, training data is the same as considered in offline setting (i.e. all the item instances before time bin $t_{\text{split}}$ are included in training dataset). However,
the training model is updated in an incremental manner, when new single time bin (e.g., week) data are available. To predict the popularity of items at time \( t_{split+1} \), all the instances of items until time bin \( t_{split} \) are considered in the training set. To predict the popularity of item instances at time \( t_{split+2} \), training set contains all data until \( t_{split+1} \) that are used to build new \( \mathcal{M} \) and \( \mathcal{D} \) models to be built again. Thus, models \( \mathcal{M} \) and \( \mathcal{D} \) are updated continuously with new available instances. Algorithm 2 defines the pseudo code for learning of models \( \mathcal{M} \) and \( \mathcal{D} \) in Online settings.

### 3.3.3 Adaptive Learning

Adaptive learning is a more advanced technique of online learning. Adaptive settings keep in the model only a window sized data defined by \( w \), such that each training model learns from data within \( w \) time bins, between the period \( split + k - w \) till \( split + k \). This learning manner is useful in scenarios where there is continuous and dramatic evolution in the relation between the input and target variables. Hence, it is important for the system to forget retrospective relation (as it may be misleading to use the whole historical data for building a model). Thus adaptive learning enables us to adapt the models according to new pattern learned from data. These most up-to-date data models enable to capture the dynamic, usually hidden but important factors affecting the popularity. Algorithm 3 defines the pseudo code for learning of models \( \mathcal{M} \) and \( \mathcal{D} \) in Adaptive settings.
Algorithm 1: Bias Aware Recommendation (BAR) algorithm for Offline Setting

1 \textbf{begin} Data Partitioning
   \begin{itemize}
     \item \textbf{Input}: Item instances $\mathcal{E} = \{e_{it}\}$, Target values $\mathcal{Y} = \{y_{it}\}$, Split criterion $t_{\text{split}}$ for partitioning data into train and test set
     \item \textbf{Output}: Dataset $A_{Train}$, Dataset $B_{Train}$, Dataset $A_{Test}$, Dataset $B_{Test}$
   \end{itemize}
   \begin{enumerate}
     \item Split data into two sets $A$ and $B$, where
       \begin{align*}
         A &= \{e_{it}, y_{it} | s_i = 0\}; \\
         B &= \{e_{it}, y_{it} | s_i = 1\};
       \end{align*}
     \item Split datasets $A$ and $B$ for Training and Testing,
       \begin{align*}
         TrainA &= \{A | t \leq t_{\text{split}}\} \\
         TestA &= \{A | t > t_{\text{split}}\} \\
         TrainB &= \{B | t \leq t_{\text{split}}\} \\
         TestB &= \{B | t > t_{\text{split}}\}
       \end{align*}
     \item END
   \end{enumerate}

2 \textbf{begin} Model learning
   \begin{itemize}
     \item \textbf{Input}: Dataset $A_{Train}$, Dataset $B_{Train}$
     \item \textbf{Output}: Unbiased predictive model $\mathcal{M}$, Bias prediction model $\mathcal{D}$
   \end{itemize}
   \begin{enumerate}
     \item Learn the unbiased model $\mathcal{M}$ on training dataset $A$,
       \begin{align*}
       \mathcal{M} : e_{it} \rightarrow y_{it}, \quad \{e_{it}, y_{it}\} \in A_{Train};
       \end{align*}
     \item Make predictions on training dataset $B$ using model $\mathcal{M}$,
       \begin{align*}
       \hat{y}_{it} = \mathcal{M}(e_{it}), \quad e_{it} \in B_{Train};
       \end{align*}
     \item Calculate the residuals from observed and under predicted popularity of training dataset $B$,
       \begin{align*}
       b_{it} = y_{it} - \hat{y}_{it};
       \end{align*}
     \item Construct a new training dataset $\hat{B}$ with $b$ as new targets;
     \item Learn the bias model $\mathcal{D}$ on set $\hat{B}$,
       \begin{align*}
       \mathcal{D} : e_{it} \rightarrow b_{it};
       \end{align*}
     \item END
   \end{enumerate}

3 \textbf{begin} Prediction and Recommendation
   \begin{itemize}
     \item \textbf{Input}: Dataset $A_{Test}$, Dataset $B_{Test}$, Unbiased predictive model $\mathcal{M}$, Bias prediction model $\mathcal{D}$
     \item \textbf{Output}: Predicted $y$, ranking of test data
   \end{itemize}
   \begin{enumerate}
     \item Apply the following on Test set $y = \mathcal{M} + s\mathcal{D}$;
     \item Rank testing data by $y$ in each time bin $t$;
     \item END
   \end{enumerate}
Algorithm 2: Bias Aware Recommendation (BAR) algorithm for Online Setting

begin Data Partitioning

Input: Item instances $E = \{e_{it}\}$, Target values $Y = \{y_{it}\}$, Split criterion $t_{\text{split}}$ for partitioning data into train and test set, Number of time bins $k$ in test set

Output: Ranking of test instances in each time bin

Split data into two sets $A$ and $B$, where

$A = \{e_{it}, y_{it} | s_i^t = 0\}$
$B = \{e_{it}, y_{it} | s_i^t = 1\}$;

for $i = 1 : k$ do

Split datasets $A$ and $B$ for Training and Testing,

$TrainA = \{A | t < t_{\text{split}+i}\}$
$TestA = \{A | t = t_{\text{split}+i}\}$
$TrainB = \{B | t < t_{\text{split}+i}\}$
$TestB = \{B | t = t_{\text{split}+i}\}$

end

begin Model learning

Input: Dataset $A_{Train}$, Dataset $B_{Train}$

Output: Unbiased predictive model $M$, Bias prediction model $D$

Learn the unbiased model $M$ on training dataset $A$,

$M : e_{it} \mapsto y_{it}$, \hspace{1cm} $\{e_{it}, y_{it}\} \in A_{Train}$;

Make predictions on training dataset $B$ using model $M$,

$\hat{y}_{it} = M(e_{it})$, \hspace{1cm} $e_{it} \in B_{Train}$;

Calculate the residuals from observed and under predicted popularity of training dataset $B$, 

$b_{it} = y_{it} - \hat{y}_{it}$;

Construct a new training dataset $\hat{B}$ with $b$ as new targets;

Learn the bias model $D$ on set $\hat{B}$, 

$D : e_{it} \mapsto b_{it}$;

end

begin Prediction and Recommendation

Input: Dataset $A_{Test}$, Dataset $B_{Test}$

Unbiased predictive model $M$, Bias prediction model $D$

Output: Predicted $y$, ranking of test data

Apply the following on Test set $y = M + sD$;

Rank testing data by $y$ in each time bin $t$;

end
Algorithm 3: Bias Aware Recommendation (BAR) algorithm for Adaptive Settings

begin Data Partitioning

**Input:** Item instances $\mathcal{E} = \{e_{it}\}$, Target values $\mathcal{Y} = \{y_{it}\}$,
Split criterion $t_{split}$ for partitioning data into train and test set,
Number $k$ of time bins in test set, Window size $w$

**Output:** Ranking of test data in each time bin

Split data into two sets $A$ and $B$, where
$A = \{e_{it}, y_{it}| s_i^t = 0\}$
$B = \{e_{it}, y_{it}| s_i^t = 1\}$;

for $i = 1 : k$ do

Split datasets $A$ and $B$ for Training and Testing,
$TrainA = \{A| t_{split+i-w} \leq t < t_{split+i}\}$
$TestA = \{A| t = t_{split+i}\}$
$TrainB = \{B| t_{split+i-w} \leq t < t_{split+i}\}$
$TestB = \{B| t = t_{split+i}\}$

begin Model learning

**Input:** Dataset $A_{Train}$, Dataset $B_{Train}$

**Output:** Unbiased predictive model $\mathcal{M}$,
Bias prediction model $\mathcal{D}$,

Learn the unbiased model $\mathcal{M}$ on training dataset $A$,
$\mathcal{M} : e_{it} \mapsto y_{it}, \quad \{e_{it}, y_{it}\} \in A_{Train}$;

Make predictions on training dataset $B$ using model $\mathcal{M}$,
$\hat{y}_{it} = \mathcal{M}(e_{it}), \quad e_{it} \in B_{Train}$;

Calculate the residuals from observed and under predicted popularity of training dataset $B$, $b_{it} = y_{it} - \hat{y}_{it}$;

Construct a new training dataset $\hat{B}$ with $b$ as new targets;
Learn the bias model $\mathcal{D}$ on set $\hat{B}$, $\mathcal{D} : e_{it} \mapsto b_{it}$;

end

begin Prediction and Recommendation

**Input:** Dataset $A_{Test}$, Dataset $B_{Test}$

Unbiased predictive model $\mathcal{M}$, Bias prediction model $\mathcal{D}$

**Output:** Predicted $y$, ranking of test data

Apply the following on Test set $y = \mathcal{M} + s\mathcal{D}$;
Rank testing data by $y$ in each time bin $t$;

end
Chapter 4

Dataset Preparation

Content information of items is needed for building a content based bias aware RS. NetFlix Prize dataset and MovieLens dataset contain only ratings information of the movies. We need to prepare data by collecting features from different sources and process the data to a form that can be used directly in our experiments. Therefore, data preparation is a very significant step towards the study of this thesis research objective.

This chapter discusses the preparations of dataset, including features collection, features definition and the statistics of ready-to-use data for experimental evaluation. Dataset preparation follows the steps below as shown in Fig. 4.1:

1. Extraction of movie content features from TMDb API, NetFlix API, and IMDB, followed by matching movie titles from NetFlix API.

2. Construction of integrated data base.

3. Retrieving ratings data from NetFlix Prize dataset (or MovieLens dataset).

4. Dataset Pre-processing, involving the following stages:

   (a) Removing outlier and extreme values from input features and target attribute y.

   (b) Calculate derived features to enrich the dataset.
(c) Normalize all the features in the range of 0-1.

\[ \text{Normalized value} = \frac{x - \min(X)}{\max(X) - \min(X)} \]

4.1 Features Collection

We extract movie content features from the following different sources:

4.1.1 IMDB

IMDB, the Internet Movie Database is a repository of large collection of movies, television programs, and video games [34]. It was initiated as a hobby project by movie fans belonging to an international group, but now it is currently owned by Amazon.com [35]. IMDB contains detailed information about each movie, i.e., writer, producer, actors, shooting location, and links to reviews. IMDB does not provide an API for developers to directly query any information. IMDB website was accessed manually for retrieving the missing information of some movies for the following features: budget, genre and count of languages in which movie is released.
4.1.2 TMDb API

TMDb (themoviedb.org) is a free movies database maintained by community users \[36\]. It is used by millions of users and provides access to information about whether the movie is translated or not, the audience of movie (i.e., kids or adults), languages it is released, number of votes it received, MPAA rating, release date, etc. For dataset preparation, following features are extracted from TMDB API: budget, run time, the overall average rating of a movie, and a boolean feature telling if it is adult movie only.

4.1.3 MovieLens Dataset

MovieLens is a collaborative filtering based movie recommender project, developed by the GroupLens (a research lab) in the department of Computer Science and Engineering at the University of Minnesota. It allows the users to rate movies, then groups the users with similar tastes and suggests movies to individuals that have not yet been watched by them. There are three datasets of different sizes available for download at GroupLens website \[14\] containing 100,000, 1 million and 10 million ratings respectively. For our experiments, the largest one, 10M MovieLens dataset was used. It includes 10000054 ratings on 1-5 scale by 71567 users on 10681 movies released during 1915-2008. Each user in the dataset rated at least 20 movies and was selected at random among other users. The data is partitioned into two files, i.e., movies.dat and ratings.dat. The format of movies.dat file is MovieID::Title (Release Year)::Genres whereas ratings.dat contains ratings in the following format UserID::MovieID::Rating::Timestamp. MovieLens dataset files contain ratings dated from Jan 1996-Dec 2008. In our experiments, only 68 movies released during 1995-2003 are selected because they have enough ratings (more than 10000 ratings).

Following features are extracted from MovieLens API: movie age (calculated as the number of weeks since the movie release date), average rating per week and count
of ratings received per week which is the target popularity of a movie in the given time instance.

4.1.4 NetFlix

NetFlix offers online streaming service to more than 33 million users in 40 different countries who can enjoy about one billion hours of movies and television shows monthly. It offers unlimited services at a very low subscription cost that can be accessed worldwide on any internet connected device. It was launched in 1997 and has observed remarkable growth since 2003.

NetFlix Prize Dataset

The NetFlix prize was announced in 2006 by NetFlix team to improve accuracy for prediction of movies according to users taste [13] as compared to their existing algorithm called Cine Match. NetFlix released a movies dataset of 17,700 movies containing 100,480,507 ratings on 1-5 scale by 480,189 users. Rating data files contain quadruplets of the form: <userID, movieID, rating, ratingDate>. Another data file contains triplets of the form: <movieID, movieName, releaseYear>. For experimental evaluation, only 75 movies released between years 2000-2004 are selected in total that have enough ratings data (more than 50000 ratings). The motive behind this selection of release years is to include real time ratings of the movie, as NetFlix Prize dataset contains ratings from Oct, 1998-Dec, 2005.

We extracted the following features from NetFlix API: movie age (calculated as the number of weeks since the movie release date), average rating per week, NetFlix subscribers count in each week, and count of ratings received per week representing the target popularity of a movie in the given time bin.

NetFlix Developers API

NetFlix released an internal API that handles billions of requests daily, provides support for more than 800 types of devices and is serving around 30M streaming
users. It allows 3rd party application development for streaming of TV shows and movies. Recently NetFlix announced that it would no longer dispatch new public API subscriptions through developer keys; however, it will continue providing support to the previous subscriptions. Following features are retrieved from NetFlix API: *release year*, *count of languages in which movie is released*, and *movie genre*.

### 4.2 Features Definition

With the collection of different features from aforementioned sources, our dataset contains the following features:

**Budget**: The amount in US Dollars allocated for the movie’s production and release.

**LanguageReleasedCount**: The number of languages in which the movie is released.

**IsAdult**: This feature determines the audience of the movie (i.e., if the movie can be watched by adults only or it can be watched by all ages). It is a derived feature and is calculated from MPAA Rating of movie. More about MPAA Rating is given in Appendix [A.1](#).

\[
\text{IsAdult} = 1, \text{ when MPAA Rating} = \text{R or NC-17}; \\
\text{IsAdult} = 0, \text{ Otherwise.}
\]

**RunTime**: It quantifies the length of movie run time in minutes.

**Genres Clustering**: NetFlix dataset contains 25 unique genres and MovieLens dataset contains 22 unique genres for the selected movies. To denote the presence of each genre in a given movie, a boolean variable can be used. However, inclusion of 25(22) boolean features for representing presence of genres in all the movies would result in a sparse matrix, as each movie contains only a max of 4 genres. To avoid the sparsity problem, we clustered the given movies’ genres using hierarchical clustering
and K-means clustering method. For clustering, each genre was represented by a boolean vector indicating its presence in all the movies. The number of clusters was determined by comparing the clustering results of hierarchial clustering and K-means clustering. Following 5 different clusters are formed for NetFlix and MovieLens dataset respectively.

- **NetFlix dataset:**
  
  GenreCluster 1: History, War, RoadMovie, Short, Sport, Drama  
  GenreCluster 2: Crime, Thriller, Horror, Mystery, Suspense  
  Genre Cluster 3: Adventure, Fantasy, SciFiction, Action, Disaster  
  Genre Cluster 4: Foreign, Romance, Holiday  
  Genre Cluster 5: Comedy, Documentary, Animation, Family, Indie, Music

- **MovieLens dataset:**
  
  GenreCluster 1: History, War, SportsFilm, Disaster  
  GenreCluster 2: Crime, Thriller, Horror, Mystery, Suspense  
  Genre Cluster 3: Adventure, Fantasy, ScienceFiction, Action  
  Genre Cluster 4: Foreign, Romance, Drama, Indie  
  Genre Cluster 5: Comedy, Animation, Family, Musical, Short

After clustering is done, we included five boolean features for each movie in our dataset as \([gc_1, gc_2, gc_3, gc_4, gc_5]\), where

\[
gc = 0, \text{ if all of the genres in that cluster is absent;}
\]

\[
gc = 1, \text{ if any of the genres in that cluster is present.}
\]

**YearOfRelease:** It represents the movies’ release year. In NetFlix dataset movies released in years 2000-2005 are considered only to capture real time rating, because

\(^1\) This vector has dimensionality of the total number of movies. Please note that each movie can have more than one genre. Therefore, the sum of all values in boolean genre vector is greater than the number of movies.
ratings are dated from Oct 1998-Dec, 2005 for all the movies in NetFlix prize dataset. However, for MovieLens dataset, movies released during years 1995-2003 are included (as the ratings are available from Jan, 1996 - Jan, 2009).

**MovieAge**: This determines the time spanned since the release date of a movie. For NetFlix dataset, this feature counts the number of weeks between the current time for popularity prediction and the movie’s release date. For MovieLens, it tallies the number of months since the release of the movie, as popularity in this data is measured as the number of ratings in month bin.

**CalendarWeekNumber (CalendarMonthNumber)**: This is an informative feature rather than an objective one. It is used for ordering data by time and for dividing the data into training and testing datasets.

In NetFlix dataset, rating data spans 314 weeks (i.e., from Jan 2000 until Dec 2005) and each calendar instance is labeled numerically, e.g., the first week of Jan 2000 is labeled as 1, and the first week of Jan, 2003 is labeled as 158.

In MovieLens dataset, rating data is for 156 months period only (i.e., Jan 1996 until Dec 2008) and each month is denoted by a number, e.g., Jan 1996 is represented as 1, and Jan 2004 is represented in numeric form as 97.

**WeeklyAverageRating/MonthlyAverageRating**: It calculates the average rating of a movie per week (per month) for NetFlix (MovieLens) dataset.

**WeeklyPopularity**: It is the target feature \( y_{it} \) denoting the number of ratings received by a movie \( i \) in current week (month) in NetFlix (MovieLens) dataset respectively.

**LastWeekPopularity**: This feature is derived from the target value \( y_{it} \) of an item instance. It is an auto regressive feature that contains the count of ratings attained by a movie in the previous week.

\[
\text{LastWeekPopularity}_{i}(t) = y_{i(t-1)}
\]

**LastTwoWeeksAvgPopularity**: It is a derived feature from target value \( y_{it} \)
and is equal to the average count of ratings gained by a movie in the last 2 weeks.

\[ \text{LastTwoWeekAvgPopularity}_i(t) = \frac{y_i(t-1) + y_i(t-2)}{2} \]

**LastThreeWeeksAvgPopularity**: It is also a derived feature from target value \( y_{it} \). It is the average count of ratings attained by a movie in the last three weeks.

\[ \text{LastThreeWeeksAvgPopularity}_i(t) = \frac{y_i(t-1) + y_i(t-2) + y_i(t-3)}{3} \]

**TotalNumberOfRatingsInPreviousWeek**: It is another derived feature from target value \( y_{it} \). It is a summation of the number of ratings for all the movies in previous calendar week.

\[ \text{Total Ratings in Last Week or Month}_i(t) = \sum_{i=1}^{m} y_i(t-1) , \]

where \( y_{i(t-1)} \) is the popularity of \( i^{th} \) movie in the week before the current moment \( t \).

**UpliftEvent**: To model bias, the uplift event of a movie is denoted by a boolean variable \( s \), where

- \( s = 0 \), when the given movie won no Oscar award;
- \( s = 1 \), the movie won atleast one Oscar award.

**NetFlixSubscribersGrowth**: This feature is only present in NetFlix dataset. NetFlix system growth is modeled by allowing a constant incremental growth in the number of subscribers per week. This feature is considered because the popularity of a movie also relies on the number of subscribers in NetFlix at the given time period. Statistics of NetFlix subscribers during years 2000-2005 are given below in Table 4.1.

<table>
<thead>
<tr>
<th>As of End of Year i.e. December 31</th>
<th>Number of Subscribers (in Thousands)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>292</td>
</tr>
<tr>
<td>2001</td>
<td>456</td>
</tr>
<tr>
<td>2002</td>
<td>857</td>
</tr>
<tr>
<td>2003</td>
<td>1487</td>
</tr>
<tr>
<td>2004</td>
<td>2610</td>
</tr>
<tr>
<td>2005</td>
<td>4170</td>
</tr>
</tbody>
</table>
4.3 Ready-to-use Data

With the defined features, we have two ready-to-use datasets (a) NetFlix and (b) MovieLens. As shown in Table 4.2, NetFlix dataset includes 75 movies with sum of 74,32,008 ratings, released during the years 2000 – 2005 respectively. In contrast, MovieLens dataset contains 10,30,178 ratings of 68 movies in total, released during years 1995 – 2003.

<table>
<thead>
<tr>
<th></th>
<th>NetFlix dataset</th>
<th>MovieLens dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of movies</td>
<td>75</td>
<td>68</td>
</tr>
<tr>
<td>No. of ratings</td>
<td>74,32,008</td>
<td>10,30,178</td>
</tr>
<tr>
<td>No. of instances</td>
<td>12,554</td>
<td>9,007</td>
</tr>
</tbody>
</table>

As given in the problem setting of Section 3.1, each movie instance is described by 11 stationary attributes, and 7 temporal attributes respectively. $X(i)$ contains 5 numerical attributes and 6 binary attributes as shown below:

$$X(i) = \{ \text{budget, language released count, run time, year of release, overall average rating of a movie, isAdult, genreCluster}1, \text{genreCluster}2, \text{genreCluster}3, \text{genreCluster}4, \text{genreCluster}5 \}$$

$Z^t(i)$ includes 7 numerical attributes as:

$$Z^t(i) = \{ \text{movie age, weekly average rating, popularity of previous week, popularity of last two weeks, popularity of last three weeks, total number of views (of all movies in the previous week), Netflix subscribers growth} \}$$

All the attributes in $Z^t$ except the last two describe the dynamic popularity of a movie in different weeks. In addition, the uplift attribute $s$ is a boolean variable and denotes the Oscar award presence.

2 NetFlix system contains rating data from Oct 1998-Dec 2005, and MovieLens dataset contains ratings from Dec,1995-Dec,2008 respectively, therefore, we selected movies released in these periods to capture real time effect on movies’ popularity due to Academy awards win.
To model the bias and make unbiased recommendation, we need to split the available data into two datasets, such that Dataset $A$ contains non-Oscar movies instances, and Oscar movies instances before the occurrence of uplift event (i.e., $s = 0$), while dataset $B$ contains only those instances of Oscar-winner movies for which $s = 1$. From NetFlix prize dataset, 48 popular movies are selected as non-Oscar winners, and 27 as Oscar winners (at least one Oscar award in any category). MovieLens dataset contains 41 popular, non-Oscar winner movies and 27 Oscar winner movies. For a fair comparison, non-Oscar winner movies are the most popular ones, many of which were nominated for Oscar and other awards and have comparable number of viewers to Oscar winners (as shown in Fig. 1.3).

Each of the datasets $A$ and $B$ are then pre-processed\(^3\) to remove outliers and normalize attributes. After removing instances for which the recorded target values were deemed as outliers, we have 11,158 instances in NetFlix dataset, and 8,545 instances in MovieLens dataset respectively.

For evaluation of our proposed algorithm, we need to split datasets $A$ and $B$ into two sets: a training set and test set for each, such that these four subsets are disjoint. For NetFlix dataset, we partition data into train and test set on the basis of Calendar Week number. All the weeks instances from Jan, 2000-Jan, 2005 are part of train set, and the remaining week instances from Feb, 2005-Sep, 2005 are included in test set for both datasets $A$ and $B$. No. of ratings and instances count included in Train $A$, Test $A$, Train $B$ and Test $B$ after partition for NetFlix dataset are shown in Table 4.3.

\(^3\)We use Weka tool for data preprocessing. Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes http://www.cs.waikato.ac.nz/ml/weka/. We used Weka version 3.6.8. For outlier removal, filter method called Interquartile range is used, which filters outliers and extreme values after detection based on interquartile ranges. Also, the data is normalized using Weka tool unsupervised filter Normalize which normalizes all numeric values in the given dataset (apart from the class attribute, if set).
For MovieLens dataset, partition is done on the basis of Calendar Month number. All the months instances from Jan, 1996-Dec, 2005 are included in train set, and rest of the instances from Jan, 2006-Dec, 2008 form a test set for each of the datasets A and B. No. of ratings and instances count included in Train A, Test A, Train B and Test B after partition for MovieLens dataset are shown in Table 4.4.

Train A contains instances of all the non-Oscar movies, and Oscar movies before winning the Oscar award. Out of 6426 instances in Train A of NetFlix dataset, 577 instances belong to Oscar winning movies before the uplift event (i.e., Oscar award). For MovieLens dataset, Train A contains 147 Oscar winning movies’ instances out of total 3919 instances before the Oscar award ($s = 0$).

Table 4.3: Statistics of NetFlix train and test sets

<table>
<thead>
<tr>
<th>Partition</th>
<th>No of instances</th>
<th>No of ratings</th>
<th>Instances From</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train A</td>
<td>6426</td>
<td>2182176</td>
<td>Jan 2000 – Jan 2005</td>
</tr>
<tr>
<td>Train B</td>
<td>2624</td>
<td>1363893</td>
<td>Jan 2000 – Jan 2005</td>
</tr>
<tr>
<td>Test A</td>
<td>1378</td>
<td>1167913</td>
<td>Feb 2005 – Sep 2005</td>
</tr>
<tr>
<td>Test B</td>
<td>730</td>
<td>872587</td>
<td>Feb 2005 – Sep 2005</td>
</tr>
</tbody>
</table>

Table 4.4: Statistics of MovieLens train and test sets

<table>
<thead>
<tr>
<th>Partition</th>
<th>No of instances</th>
<th>No of ratings</th>
<th>Instances From</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train A</td>
<td>3919</td>
<td>285756</td>
<td>Jan 1996 – Dec 2005</td>
</tr>
<tr>
<td>Train B</td>
<td>2183</td>
<td>232006</td>
<td>Jan 1996 – Dec 2005</td>
</tr>
</tbody>
</table>
Chapter 5

Experimental Evaluation

This chapter presents the experiments conducted to evaluate the performance of our proposed approach BAR algorithm, and its different adaptations. First, bias estimation capacity of the proposed algorithm BAR is presented. Then we compare BAR performance using different temporal settings. Finally, we propose the bias correction method to clean the bias from historical training dataset.

All the experiments are carried out on Netflix and MovieLens datasets, respectively. Statistical information of datasets used in our experiments is provided in Section 4.3 Multiple Linear Regression (standard regression technique) is used to learn models $\mathcal{M}$ and $\mathcal{D}$ in BAR.

**Evaluation Measure:** In all the experiments, Mean Absolute Measure (MAE) is used as an evaluation measure. It is a measure of how close the predicted values are to the observed ones. In our dataset, we measure MAE as follows,

$$MAE = \frac{\sum_{j=1}^{n} |y_j - \hat{y}_j|}{n}$$  \hspace{1cm} (5.1)

where $y_j$ represents the popularity of item instances in $A_{\text{test}}$ or $B_{\text{test}}$, and $\hat{y}_j$ represents the predicted popularity of item instances in $A_{\text{test}}$ or $B_{\text{test}}$, respectively, and $n$ is the number of item instances.

$^1$ MATLAB tool (version R2012a) provides support for Multiple Linear Regression.
5.1 Bias Quantification

This section validates the bias estimation capacity of proposed algorithm BAR at an instance level of all the items. We apply BAR on test set for predicting the popularity of Oscar and non-Oscar winning instances, in the future unseen weeks. Fig. 5.1 and Fig. 5.2 shows instance level bias estimation on NetFlix dataset and MovieLens datasets, respectively. The X-axis displays all the test instances in ascending order of their popularity, while Y-axis shows the predicted and observed popularity for each instance.

Fig. 5.1 (a) shows the predicted popularity of test set $B$ shown by red stars when model is learned on Train $B$ using multiple linear regression classifier. Blue crosses denote the actual targets of test set $B$. This experimental setting can be considered as a special case when there is enough data in dataset $B$ to learn a specialized model for Oscar winning instances ($s = 1$). As expected, the model learnt on biased data (i.e., Oscar winning instances) performs the best on the future biased test set.

To calculate the true popularity of instances in test set $B$, predictions are carried out using BAR algorithm by turning off the uplift event effect, i.e., substituting $s = 0$ in $y = M + sD$. Fig. 5.1 (b) shows the estimated popularity of test set $B$ instances when model $M$ is learned using linear regression classifier. We observe that BAR mostly under-predicts the popularity of Oscar winning movies’ instances by ignoring the effect of Oscar award. This generates the true unbiased popularity scores of instances in test set $B$ which can then be ranked in ascending order to find the best and true popular movies for recommendation.

To evaluate the bias estimation capacity of BAR, the estimated bias is added to under-observed (true merit) predictions, i.e., by allowing $s = 1$ in $y = M + sD$ (see Eq. 3.2). In Fig. 5.1 (c), it can be observed that the proposed bias aware recommender (BAR) algorithm quantifies the bias very accurately, which when added to the true popularity (shown in Fig. 5.1 (b)) becomes approximately similar to the predictions
Figure 5.1: Evaluation of BAR on NetFlix dataset

of specialized but biased model learned over the biased dataset $B$ (refer Fig. 5.1 (a)).

The main benefit of BAR is that it enables to find the bias in the observed popularity of individual items at each time interval. Moreover, it is flexible and can be applied easily to both uplifted item instances (dataset $B$) and non-uplifted item instances (dataset $A$).

Fig. 5.1 (d) validates the performance of BAR over the unbiased test set $A$ such that the predictions of BAR correspond to the actual targets and are calculated using model $M$ which is learned from train set containing non-Oscar winning instances only.
As observed in Fig. 5.1 (d), regression technique captures the overall rating trend very well.

(a) Predictions of a biased model learned over train B (on Oscar Instances) $y_B = \mathcal{M}_B$

(b) Predictions of BAR (on Oscar Instances) $y_B = \mathcal{M}_A$

(c) Predictions of BAR including the estimated bias (on Oscar Instances) $y_B = \mathcal{M}_A + \mathcal{D}_B$

(d) Predictions of BAR (on non-Oscar Instances) $y_A = \mathcal{M}_A$

Figure 5.2: Evaluation of BAR on MovieLens dataset

The mean absolute error (MAE) for all the models is also labeled in the legend of Fig. 5.1. It is important to mention that movie instances data in test set is from the later period (than train set) when there were sufficient Netflix subscribers and movies in the system. Such a growth thus brings challenges in predicting the real popularity of movies. However, the prediction error of the proposed algorithm BAR is acceptable.

Fig. 5.2 shows the similar performance on MovieLens dataset. From the results of
Fig. 5.1 and Fig. 5.2, it can be concluded that BAR well quantifies and corrects the bias in the popularity at instance level of each item. The bias aware predictions done by the proposed algorithm (BAR) can be used to generate the bias aware list of most popular movies by ranking the movies according to their true (fair and unbiased) popularity scores.

5.2 Temporal Aspect of the Bias

It is really important to incorporate the temporal effect of bias while estimation because constant bias assumption (applied in the field of discrimination-aware data mining state of the art) for all the items at all the time instances does not work. In this section, the importance of considering the temporal aspect of the bias is studied by doing a comparative performance analysis of the following different models:

1. Standard Method: A regression model is build using full training dataset containing Oscar and non-Oscar winning instances, and then applied to whole test set (i.e., test A+ test B) to measure the predictive performance.

2. BAR with constant bias ($BAR + \mathcal{D}_0$): Separate models $\mathcal{M}$ and $\mathcal{D}$ are built as described in Algorithm 1 and target values for test set are predicted using $y = \mathcal{M} + s\mathcal{D}$, where $s = 1$, and $\mathcal{D}$ estimates a constant bias $\mathcal{D}_0$ for all the Oscar winning instances.

3. BAR with item related bias ($BAR + \mathcal{D}_X$): Predict the target values of test set using $y = \mathcal{M} + s\mathcal{D}$, such that $s = 1$, and $\mathcal{D}$ estimates the bias $\mathcal{D}_X$ (item static features bias only) with respect to the goodness of each Oscar winning movie content features $X$ only.

4. BAR with item and time related bias ($BAR + \mathcal{D}_{X,Z}$): Predict the target values of test set using $y = \mathcal{M} + s\mathcal{D}$ where $s = 1$, and $\mathcal{D}$ estimates the bias with respect to
goodness of each Oscar winning movie content features $X$ and temporal features $Z'$.

5. Two Models: Here an ideal scenario is considered when there is sufficient data to build two separate models; one on uplifted (Oscar winning) item instances, and the other on non-uplifted ($s = 0$) item instances. Both models are then used for predictions such that uplifted model calculates future predictions for test set of uplifted item instances, and non-uplifted model predicts the future popularity of non-uplifted item instances. Using this strategy, maximum accuracy can be observed to find the observed popularity in the real world precisely. However, it does not enable us to quantify nor correct the bias.

![Figure 5.3: Comparison of models with different bias estimation in offline, online and adaptive learning framework](image)

Fig. 5.3 shows the Mean Absolute Error (MAE) on Y-axis by calculating the prediction error on the complete test set (i.e., test set A and test set B are combined) for each of the bias estimation approaches which are represented on X-axis, respectively. From the observations, it is evident that BAR with different bias estimation parameters perform better than the standard method. A comparative performance is
observed for \texttt{BAR} with item and time level bias estimation (i.e., \texttt{BAR} + \texttt{D}_{X,Z}), and Two-models. Two-models used for prediction is not an ideal case because it does not quantify the bias, and in real world does not have practical application. It gives the best performance however, because it predicts the popularity of Oscar and non-Oscar test set movie instances correctly by using separate models. Alternatively, \texttt{BAR} with item and time related bias \texttt{BAR} + \texttt{D}_{X,Z} can effectively and accurately estimate the bias for each privileged item. This is an advantage of our proposed approach over its competitive approaches for bias estimation as none of them can estimate the temporal bias accurately.

In previous studies [18, 37, 38] constant bias was accounted for all the privileged items. However, from above experiments, it is observed that the temporal bias is well captured by \texttt{BAR} + \texttt{D}_{X,Z} (item and time related bias) model as compared to \texttt{BAR} + \texttt{D}_0 (constant bias) model, and is important to be accounted in the field of bias aware recommenders.

Comparison results of different kinds of bias estimation models using offline, online and adaptive model learning frameworks using linear regression bias estimator are presented in Fig. 5.3 (a) and (b) for NetFlix and MovieLens datasets, respectively. Online and Adaptive settings build most up-to-date data models that enable to capture the dynamic, usually hidden but important factors affecting the popularity. For given test set, online framework and adaptive framework perform slightly better than offline framework for model learning in our study. This is because online and adaptive learning are useful only when there is continuous and dramatic evolution in the relation between input features and target popularity over time in test set. The data used in test set (both NetFlix and MovieLens) does not evolve with sudden changes because popularity remains more or less stable in the period selected for testing. Therefore, online and adaptive settings show a slight improvement in results.
5.3 Bias Correction

The proposed method \textbf{BAR} can easily be used for bias correction in two ways. First, by switching off the bias estimator of \textbf{BAR}, i.e., $s = 0$ at prediction step (as shown in Fig. 5.1 (b) and 5.1 (d)). Second, correct the historical data used for training by subtracting the estimated temporal bias of \textbf{BAR} from the popularity labels of Oscar winning item instances for which $s = 1$. The treated training data can then be used as additional training data for learning an unbiased model.

Table. 5.1 presents the prediction error (MAE) on NetFlix dataset when models are learnt in offline, online and adaptive mode on the dataset of unbiased $A$ and biased $B$, unbiased $A$, and unbiased $A$ and treated (bias corrected) $B$.

- If the bias effect is neglected and bias cannot be quantified, instances of unbiased dataset $A$ and biased dataset $B$ will used without a difference. A model is then learned as follows:

$$\mathcal{M}: e_{it} \mapsto y_{it}, \quad \{e_{it}, y_{it}\} \in A, B$$

- For unbiased dataset, a single model $\mathcal{M}$ is build on the instances of training set of $A$, such that:

$$\mathcal{M}: e_{it} \mapsto y_{it}, \quad \{e_{it}, y_{it}\} \in A$$

- Suppose model $\mathcal{M}$ and $\mathcal{D}$ have been already built in Algorithm 1 and there is more biased data available. In order to learn an unbiased model on more unbiased data, we remove the bias in the new available data and call that treated data. The learned $\mathcal{D}$ is first used to predict the bias $D_{it}$ in the popularity of treated data. The unbiased popularity is then calculated for this dataset as $y_{it}^{treated} = y_{it} - D_{it}$, where $y_{it}$ is the observed popularity of item instances of that biased
Finally, a fair merit model $\mathcal{F}$ is constructed on the item instances of unbiased training set $A$ and treated instances of biased data.

$$
\text{Treated bias dataset } \hat{\mathcal{B}} = \{e_{it}, y_{it}^{\text{treated}}\}
$$

$$
\mathcal{F} : e_{it} \mapsto y_{it}, \quad \{e_{it}, y_{it}\} \in A, \hat{\mathcal{B}}
$$

We compare the models built on 3 different datasets to test their predictive performance on unbiased data, i.e., Test $A$. A naive approach for building a prediction system will be to build a model on all the data available, i.e., biased $A$ and unbiased $B$. The predictive performance of this system on unbiased item instances of Test $A$, however, will be less accurate, because the model will over predict the popularity of some unbiased item instances due to the presence of bias in the model. Alternately, the model learned on only unbiased data, makes more accurate predictions as compared to the naive approach, as the variation between target values of item instances in train set is not that high, and close to the target values of item instances in Test $A$.

The third model, that is learned on unbiased $A$ and treated (bias corrected) $B$ will outperform the other two models because the more the data available for learning a model, the better it is (model is less over fitted). Also, target values of treated (bias corrected) $B$ are close to the target popularity of unbiased $A$. Therefore, the resulting model best predicts the popularity of item instances of Test $A$.

Table 5.1: Bias correction measure (MAE) by removing the estimated bias from the training data in NetFlix dataset

<table>
<thead>
<tr>
<th>Training Data</th>
<th>Test Data</th>
<th>Offline</th>
<th>Online</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unbiased $A$ + Biased $B$</td>
<td>Test $A$</td>
<td>222.6076</td>
<td>133.4194</td>
<td>133.4748</td>
</tr>
<tr>
<td>Unbiased $A$</td>
<td>Test $A$</td>
<td>134.4122</td>
<td>131.7693</td>
<td>131.8184</td>
</tr>
<tr>
<td>Unbiased $A$ + Treated (bias corrected) $B$</td>
<td>Test $A$</td>
<td>136.8179</td>
<td>129.5928</td>
<td>129.6389</td>
</tr>
</tbody>
</table>
Table 5.1 reports the results of performance of model learned on these 3 different datasets. It is observed that BAR correctly estimates the bias. The model learned on unbiased A and treated (bias corrected) B results in more accurate predictions on unbiased set (test A) as expected. Also, the model learned on unbiased A and biased B is observed to be least accurate in predicting popularity of unbiased Test A.

Another important observation from Table. 5.1 is that online and adaptive approaches perform slightly better than offline settings because there is small sudden change in number of ratings or number of subscribers in the given test sets of NetFlix and MovieLens datasets. For the given data, online model learning framework performs slightly better compared to the other two settings. In adaptive model learning framework, window size can be adjusted to specify the number of recent weeks’ data taken as training set. In these experiments, window size for NetFlix dataset is adjusted to 250, as this setting provides maximum accuracy for prediction.

Table 5.2: Mean and standard deviation of target values of test set instances

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NetFlix</th>
<th>MovieLens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test A</td>
<td>847.5421</td>
<td>77.2811</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>454.5521</td>
<td>42.7503</td>
</tr>
</tbody>
</table>

It is important to mention that the target values of instances in test A have high standard deviation as shown in Table. 5.2. The mean and standard deviation of the target of the item instances in unbiased data for testing (i.e., Test A), are 847.5421 and 454.5521, respectively for NetFlix dataset. Since reported MAE values are very less than the actual target values, therefore, reported prediction error (MAE) is acceptable.

Table. 5.3 reports similar results on MovieLens dataset using MAE as estimation measures. Window size used for adaptive learning is 36. The mean and standard deviation of the target of the item instances in unbiased data for testing (i.e., Test A), are 77.2811 and 42.7503, respectively. Therefore, reported MAE is acceptable for
Table 5.3: Bias correction measure (MAE) by removing the estimated bias from the training data in MovieLens dataset

<table>
<thead>
<tr>
<th>Training Data</th>
<th>Test Data</th>
<th>Offline</th>
<th>Online</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnbiasedA + BiasedB</td>
<td>TestA</td>
<td>19.0823</td>
<td>16.3038</td>
<td>15.5877</td>
</tr>
<tr>
<td>UnbiasedA</td>
<td>TestA</td>
<td>17.2903</td>
<td>16.5957</td>
<td>16.3628</td>
</tr>
<tr>
<td>UnbiasedA + Treated (bias corrected) B</td>
<td>TestA</td>
<td>16.2031</td>
<td>16.3105</td>
<td>14.8808</td>
</tr>
</tbody>
</table>

MovieLens dataset also. The results conclude that the model learned on unbiased A and treated (bias corrected) B performs the best, as expected. However, surprisingly, the prediction performance of model learned only on unbiased A is less accurate as compared to the model learned on unbiased A and biased B when online and adaptive model learning frameworks are used. This anomaly in predictive behavior of unbiased A can be attributed to very few ratings in each bin for MovieLens dataset, and incremental learning of models may not suit such data nature.
Chapter 6

Conclusions and Future Work

In this chapter, the contribution of this thesis is summarized by outlining the main developments and discussing the results. Next, the application of bias aware recommenders is elaborated. Finally, some interesting future directions to extend this work are proposed.

6.1 Conclusions

We address the following in this thesis:

1. **Identifying the bias problem**
   
   Bias aware recommenders in the domain of discrimination aware data mining have been studied in previous works. All of these assume the bias to be a constant factor that either exists in an item or not when recommended by RS. Nevertheless, in this thesis, we present a framework that deals with temporal bias in RS resulting from uplift events in the life of an item. We say that popularity of an item is not a constant phenomenon; it changes as a matter of time. After the occurrence of an uplift event, e.g. Oscar awards, the popularity of an item is uplifted or sees a sudden positive drift, which may increase, decrease or remain constant in the future time. Thus, our goal in this thesis is to model the popularity of an item due to uplift events. We argue that
popularity of an item directly influences its recommendation. The more the popular items are, the more they are recommended at the top recommended items list. Studying the temporal bias due to uplift events and modeling its impact on RS predictions is a novel and difficult problem which to the best of our knowledge has not been considered before. In this thesis, we focused on non-personalized bias aware RS. However, in future, this work can be extended to personalized bias-aware RS as discussed in Section 6.3.

To study this problem we first collected evidence of uplift event bias in NetFlix dataset, as shown in Fig. 1.2 and Fig. 1.3. We then prepared a dataset by collecting movies’ content information from different sources, and integrating it with movie ratings collected from two real world datasets namely NetFlix Prize dataset [13] and MovieLens dataset [14] respectively.

2. Formulation of a framework to study different kinds of bias:

To study the bias problem, we formulate a framework which splits the item popularity into two parts: (a) popularity of an item due to its true merit; (b) additional increase in the popularity due to biasing impact of uplift events in the life of an item. We use Eq. 3.2 in all the experiments to evaluate the popularity of an item. Also, we study and introduce following different kinds of bias

(a) Static bias $\mathcal{D}_0$

(b) Item bias $\mathcal{D}_X$

(c) Item and Temporal bias $\mathcal{D}_{X,Z}$

Fig. 5.3 shows the comparison of BAR when bias is considered static, i.e., $\mathcal{D}_0$, bias is considered to be dependent on item content features only $\mathcal{D}_X$, and when bias is considered to occur with respect to item content and temporal features $\mathcal{D}_{X,Z}$. For our thesis problem statement, BAR accounting item and temporal bias performs the best by estimating the bias accurately, as shown in Fig. 5.3.
3. **Studying Temporal behavior of bias for each item at instance level**

The proposed algorithm **BAR** is capable of estimating bias for all the items at instance level, where each instance defines the popularity as a result of its content and temporal features in a given week.

(a) **Bias Estimation:**

As shown in Fig. 5.1 (c) and Fig. 5.2 (c), **BAR** estimates the bias at instance level accurately, which when added to true popularity of an item instance accounts for the real observed biased popularity for biased item instances.

(b) **Bias Correction:**

**BAR** enables us to correct the historical data by disregarding the bias effect, i.e., subtracting the estimated bias from unfair (biased) historical training data instances that is used for learning model. Table 5.1 and 5.3 present the accuracy of bias correction method using MAE as performance metric, on NetFlix and MovieLens dataset respectively.

4. **Proposing model for different application environments:**

In this thesis, we propose adaptation of **BAR** for following different application environments

(a) **Offline:** The model once learned remains fixed for all the future unseen instances.

(b) **Online:** The model is updated for each new instance.

(c) **Adaptive:** Adaptive settings are useful, when it is required to forget the past relation between the input features and target variable, and to adjust automatically with the new data. In this setting, a window size is defined to keep the most recent data in memory for learning a model, which is then used for testing instances at time $t_i$. 
We evaluated both the datasets, i.e., NetFlix and MovieLens using offline, online, and adaptive settings to evaluate bias aware recommender system (BAR). Fig. 5.3 present the results on comparative study of different kinds of biases using offline, online and adaptive model learning frameworks. For given dataset, online model performs comparatively slightly better than others. Online and adaptive settings are useful to capture dynamic, usually hidden but important factors affecting the popularity of items.

6.2 Applications of Bias Aware RS

1. BAR can be used to generate predictions for ranking and recommending items by realizing their true merit.

2. Bias estimated using RS algorithm can be used as an evidence of bias or discriminatory practices in the court of law and also to analyze the impact of certain on-purpose uplift events (e.g., new marketing strategies, advertisement campaigns).

6.3 Future Research Work

The work presented in this thesis can be extended in the following directions.

6.3.1 Personalized RS

Bias aware recommender algorithm can be extended for personalized RS by considering user profile. Some user features that will be significant to consider for extending this work are as follows:
Technology Preference

A very useful feature is the addition of user preference parameter in order to define user trends. In other words, a clear definition needs to be sought as to technology format user preferences. As can be seen from the statistics below in Table. 6.1 referenced from [39], differing platforms play a significant role in users’ viewing habits.

Table 6.1: User preferences for technology platform

<table>
<thead>
<tr>
<th>Platform</th>
<th>% share of film viewing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Live television</td>
<td>19.5</td>
</tr>
<tr>
<td>2 Recorded from television</td>
<td>14.5</td>
</tr>
<tr>
<td>3 Cinema</td>
<td>11.1</td>
</tr>
<tr>
<td>4 Subscription TV/Sky</td>
<td>10.9</td>
</tr>
<tr>
<td>5 Receive DVD gift</td>
<td>10.8</td>
</tr>
<tr>
<td>6 Buy DVD</td>
<td>9.0</td>
</tr>
<tr>
<td>7 Rent DVD</td>
<td>6.2</td>
</tr>
<tr>
<td>8 Catch-up TV/iPlayer</td>
<td>4.9</td>
</tr>
<tr>
<td>9 Piracy P2P</td>
<td>2.5</td>
</tr>
<tr>
<td>10 Buy Blu-ray disc (BD)</td>
<td>1.9</td>
</tr>
<tr>
<td>11 Online streaming</td>
<td>1.9</td>
</tr>
<tr>
<td>12 Pay-per-view TV</td>
<td>1.7</td>
</tr>
<tr>
<td>13 Receive Blu-ray disc (BD)</td>
<td>1.6</td>
</tr>
<tr>
<td>14 Rent Blu-ray disc (BD)</td>
<td>1.3</td>
</tr>
<tr>
<td>15 Piracy cyberblockers</td>
<td>1.1</td>
</tr>
<tr>
<td>16 Legal download</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Genre Preferences and Gender

Some users like to watch only specific genres (e.g., action, horror, comedy, historical, etc.) regardless of its publicity or award nominations or award wins. Movie watching habits are also reliant on audience gender. For instance, women have more interest in viewing movies that revolve around thrill, mystery, drama, or romance. Men, by contrast, prefer action films on the whole. However, comedy and adventure movies are preferred by both the sexes of all ages [40]. Therefore, consideration of genre preferences and gender in user profile will add significance to build personalized bias aware RS.
MPAA rating (refer A.1) of a movie signifies the age group appropriate for watching a movie. To make a personalized biais aware RS audience age will help in finding the relevant and right movie suggestion.

### 6.3.2 Alternative Causes of Uplift Events

Besides Academy awards, there are other important factors that influence the movie popularity and are noteworthy to study bias effect of uplift events in a movie’s life span. Some of them are briefly discussed below:

**Significant Awards**

In this thesis, awards are considered to be an example of an uplift event in the life span of movies. Awards indeed have an appealing effect on viewers. They are not only used as a branding tool, but also a source of generating free publicity [40]. It is believed that awards publicize the movie to general audiences. The most significant awards for movies would be Academy awards, Golden Globes, Golden Bears, Golden Lions, and Golden Palms [40]. Also, nominations play an important role by boosting the movie’s popularity, where the main attention and publicity is attracted by the ‘best film’, ‘best actor’, and ‘best actress’ awards. To lay down a solid foundation of a framework for bias estimation, we limited the study focus by considering the effect of Academy awards only. However, in future, this work can be extended by including the effect of all other major and significant awards.

**Technology**

Movies’ popularity is largely dependent on the technology used for production, distribution, and consumption [41]. For example, in today’s era, movies released in 3-D gain greater publicity and are viewed more when compared to movies of similar merit
released in other formats (e.g., Blu-ray, or DVD). A movie should be able to compete with other movies’ digital format \[42\]. This offers huge opportunities and challenges for the movie industry at the same time. A possible future extension of this work is to analyze how technology uplifts movies’ popularity by collecting enough related data that can be used to compare and understand its impact.

**Star Power (Actor, Actress, Director)**

Movie that cast popular actors, actresses or is directed by a famous director (who won an Oscar or other awards in the past) tend to attract large audience and hence may get high publicity and popularity in pre-release and post-release. The popularity variation in the longer run, however, depends on the quality of movie (i.e., script, and performance of acts). To determine true bias in recommender systems and for suggesting a fair list of movies, consideration of star power will add substantial value to the bias aware recommendation.

**Sequential Release**

Movies are distributed in different channels and released sequentially in different countries, and, therefore, can be considered to have multiple lives \[43\]. This is because, with each release of the same movie, new viewers are born in different geographical locations and this increases the revenue and popularity of the movie considerably high. Tracking multiple releases in dataset and algorithm settings, and predicting the movies popularity in different geographical locations would be a challenging task and can be considered as possible extension of this thesis study.

**Reviews**

“Will the film lead the media reviews?” is an important information noted when a marketing plan is designed for a movie \[44\]. As mentioned in \[43\], movies are
influenced at each stage of their life by many factors such as movie critics reviews, social media, etc. As some people rely on reviews rather than advertisements because they may be interested in knowing about the quality of a movie, therefore a good position of review is preferred [40]. Analyzing the impact or reviews on the popularity of items besides other uplift events will add significance to this problem study.
REFERENCES


APPENDICES
A General Notes

A.1 MPAA Rating

What does MPAA Rating means?

G General Audiences (All Ages Admitted). The motion picture contains nothing that would offend parents for viewing by their children.

PG Parental Guidance Suggested. Some Material May Not Be Suitable For Children Parents are urged to use Parental Guidance, as the motion picture may contain some material parents might not like for their younger children to view.

PG-13 Parents Strongly Cautioned. Some Material May Be Inappropriate For Children Under 13. Parents are urged to be cautious. Some material may be inappropriate for pre-teenagers.

R Restricted. Children Under 17 Require Accompanying Parent or Adult Guardian. Contains some adult material. Parents are urged to learn more about the motion picture before taking their younger children with them. Generally, it is not appropriate for parents to bring their young children with them to R-Rated Motion pictures.

NC-17 No One 17 and Under Admitted. Patently adult. Children are not admitted.
B Papers Submitted and Under Preparation


B.1 My Contribution:

1. Verification of problem existence in NetFlix dataset.

2. Preparation of dataset by extraction of features and ratings, and forming an integrated database.


4. Analysis of experimental results in detail.

5. Defining future work.

This thesis is an extension of the submitted work. Additional work done in this thesis is as follows:

• Problem setting for data partitioning - Instead of partitioning data into Oscar and non-Oscar movies (as done in submitted paper), here we partition the data into Oscar and non-Oscar instances.
• Simulation of experiments on the new data sets, and analyzing the effect of bias in detail.

• Discussion and Experimental analysis of MovieLens datasets results in detail besides NetFlix dataset.

• Analysis of different model learning frameworks, i.e., Offline, Online and Adaptive settings in more detail.

• Defining future extensions of this work.