1-1-2017

Probabilistic Personalized Recommendation Models For Heterogeneous Social Data

Vineeth Rakesh Mohan
Wayne State University,

PROBABILISTIC PERSONALIZED RECOMMENDATION MODELS FOR HETEROGENEOUS SOCIAL DATA

by

VINEETH RAKESH MOHAN

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2017

MAJOR: Computer Engineering

Approved By:

_____

Advisor Date

_____

_____

_____

_____

DEDICATION

This dissertation is dedicated to my grandmother Padmavathi Rajamiyer.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Dr. Chandan K. Reddy, a man with extensive knowledge and a remarkable dedication. Throughout my carrier as a Ph.D. student, he was a monumental source of inspiration. As the saying goes, “The only way to discover the limits of the possible is to go beyond them into the impossible”. I was able to challenge the impossible due to his constant guidance and motivations. I could not have imagined having a better advisor and mentor for my Ph.D. study. Besides my advisor, I also thank my co-advisor Dr. Harpreet Singh for his insightful comments, patience and encouragement.

This journey would not have been possible without the support of my parents and my sister. To me, my parents were more than a couple of people who provided moral support; they were tutors, guide, mentors and a beacon of optimism and hope. My father got his doctorate in Mathematics at the age of sixty and taught me the true meaning of perseverance, while my mother taught me the art of patience and humor.

I would like to thank my fellow doctoral students and colleagues Bhanukiran Vizhamuzhi, Ping Wang, Rajiur Rahman, Tian Shi and Yan Li for their feedback and cooperation. A special thanks goes to Bhanukiran Vizhamuzhi for providing several valuable suggestions and Rajiur Rahman for being a great friend and joining me in many photographic road trips. I am also indebted to my friends and students from other labs. I thank Andrey Kashlev, Chuan Li, Erfan Najmi, Khayyam Hashmi, Nariman Ammar, Safraz Rampersaud, Xiaohui Liu, and Yu Chen for sharing their knowledge and time. As my ex-colleague, I must say it was a lot of fun working alongside Xiaohui Liu during my days as a Masters student. I thank him for being an amazing person and a great friend.

Besides my family members, I am also grateful to have wonderful relatives who played a crucial role in my journey as a doctorate student. Especially, I thank my cousin Ramachandran Subramanyam for helping me understand several concepts on programming, my cousin Vidya Rajagopal for being a wonderful and helpful person, my aunt Saraswathi Rajagopal and uncle Dr. Rajagopal Ganapathy for setting an exemplary example. In fact, I lost count of the number of

graduate degrees my aunt holds. I also thank my aunt Sandya Ravi and uncle A.R Ravi for their love and unconditional support.

As rightly said by Amy Pholer, “Find a group of people who challenge and inspire you; spend a lot of time with them, and it will change your life”. I must say, I was extremely lucky to find such inspiring friends who created a positive impact on my life. For this, I thank Abhi Patel, Adeniyi Osisanya, Anusha Mannepalli, Andrew Jones, Jashwanth Reddy, Kadeem Sims, Avinash Phadatare, Nirav Patel, Rahul Gururaj, Ronald Wilson. Specifically, I thank Abhi Patel and Nirav Patel for creating a wonderful home were I was able to shrug-off the stress and work better. I also thank my brother Adeniyi Osisanya for showing the true meaning of dedication and sharing some great workout sessions at the gym. No matter how bad the situation was, I always found him working out at the gym without missing a single day. Thanks to such people, I was able to lead a healthy lifestyle amidst the pressure of research and publication.

Finally, I thank Dr. Abhilash Pandya, Dr. Feng Lin, and Dr. Kazhuhiko Shinki for serving on my committee and providing invaluable feedback.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Content recommendation has risen to a new dimension with the advent of social media platforms such as Twitter, Facebook, FriendFeed, Dailybooth, Instagram, etc. Although, this uproar of data provides us with a gold-mine of real-world information, the conventional recommendation models based on collaborative or content-based techniques are not capable of capturing the complex heterogeneous relationship provided by these data sources. The content-based recommendation techniques work on a non-relational setting where the association between the entities are ignored. In other words, it assumes that users do not influence each other. Contrary to this notion, collaborative filtering techniques model the relationship between the entities; however, it ignores the content-based features. Therefore, designing a model that is independently based on either collaborative or content-based systems is not sufficient to create a robust recommender framework. For example, consider a scenario of recommending an item to a user in Amazon. To understand the likes and dislikes of this user, one could use his purchase history to obtain features such as frequently purchased categories, spending pattern, wish lists etc. However, it is important to note that the behavior of this user is not restricted to a single platform (in this case Amazon); instead, the digital footprints of this user can manifest in the form of social network activities, blogs, web searches and even purchases made in other e-commerce websites. These activities provide us with a plethora of valuable information. For example, information from the user's Facebook profile can provide insights about the nature of his social circle and the type of pages the user likes and follows. Profile information from Twitter can provide insights on follower, friends and the content information about his Tweets and geo-location information. In addition to the user-based attributes, external domains also provide various product-based attributes such as promotional activities about the product, attractive product images from Instagram etc. With the availability of such a wide variety of information, it is essential to address the question "how to effectively utilize such high dimensional distribution of attributes?". The solution to this question lies in addressing the following challenges associated with recommendation systems and the heterogeneity of data.

(1) **Sparsity in item selection (explicit feedback or implicit feedback)**: most users buy few items from a repository containing several thousand items, therefore distribution of items per-user is extremely sparse. In the recommendation literature this problem is widely known as the *cold start problem*.

(2) **Difficulty in clustering similar users:** some users have unique tastes and varied buying patterns. Such users form a distribution of their own with very little similarity with other users. Therefore, forcefully merging these users with other users will degrade the performance of the recommendation model. This phenomenon is widely known as the *grey sheep problem*.

(3) **Overcoming the curse of dimensionality to create scalable recommendation framework:** having a huge feature space is not always profitable since it becomes extremely difficult to choose the right set of features that can capture the behavior of users. Additionally, the large feature dimension also affects the scalability of the recommendation models.

(4) **Optimizing for Bias and Variance:** with a large heterogeneous feature space it is very much likely for a model to under- or over-fit. Therefore, it is extremely important to create models that have a good balance between bias and variances.

In this thesis, we show various ways in which we can create hybrid recommendation systems. From simple recommendation models involving heuristic combinations of collaborative and content based techniques to more complex systems involving latent probabilistic frameworks that jointly models user's behavior for greater personalization.

1.1 Heterogeneity of Data

We begin by explaining the intuition behind data heterogeneity in Figure 1.1, which shows the different *layers* of features that affect a user's interest in a particular item. In this example, we have chosen the item to be the popular movie *Moana* by Disney pictures. This item can vary from physical products such as books, electronics or even digital recommendations such as push notifications in mobile phones and point of interest recommendation. In this Figure, we can see that the user's interest to pick a movie is influenced by the following attribute layers:

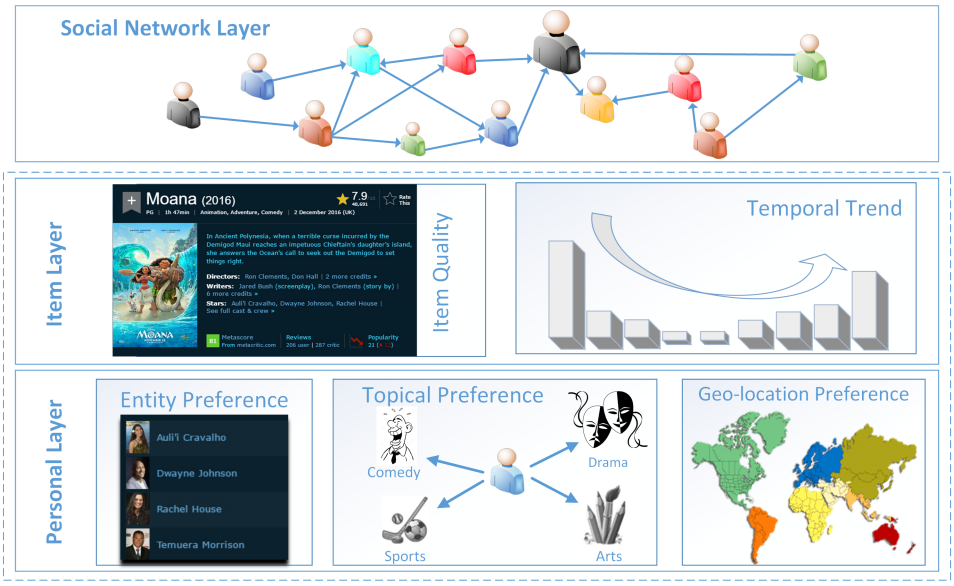


Figure 1.1: Impact of project, personal and social network based features on Kickstarter users

(1) **The social network layer:** users are socially linked through various social media sites, which in-turn influences their purchase behavior. For instance, people discuss about movies in Twitter, follow celebrities, retweet and like Twitter posts. Some companies also have dedicated Facebook pages to promote their product and keep users informed about the latest developments.

(2) **The item layer:** this layer comprises the content quality and the *temporal trend* of the movie. The temporal trend can be determined using popularity factors of the item such as box office collection, promotional activities and reviews.

(3) **The personal layer**: This layer is a combination of the topical preference of the user, the user's preference over the cast members of the movie, and the influence of geo-location. The geo-location of users have a strong influence over the selection of items. Nonetheless, this impact is not uniform for all types of items. For example, products such as high end DSLR cameras have a strong demand in USA, UK, France and Germany, while consumer grade cameras are preferred in Asian countries. Similarly, action-based movies such as the Transformers series are highly popular in China, while drama-based movies such as *La La land* and *Moonlight* have a mediocre response.

From the above explanation, we can see that in web data, features are not confined to a single platform. Therefore, to build a robust recommendation model for a specific domain, it is extremely important to incorporate the heterogeneous linkage between other external domains.

1.2 Contributions

The key contributions of our thesis are summarized in the following topics. In the upcoming sections, each of these topics will be explained in detail.

Recommending Twitter Lists: Lists in social networks have become popular tools to organize content. In paper [1], we propose a novel framework for recommending *lists* to users by combining several features that jointly capture their personal interests. Our contribution is of two-fold. First, we develop a `ListRec` model that leverages the dynamically varying tweet content, the network of *twitterers* and the popularity of lists to collectively model the users' preference towards social lists. Second, we use the topical interests of users, and the list network structure to develop a novel network-based model called the LIST-PAGERANK. We use this model to recommend auxiliary lists that are more popular than the lists that are currently subscribed by the users. We evaluate our `ListRec` model using the Twitter dataset consisting of 2988 direct list subscriptions. Using automatic evaluation technique, we compare the performance of the `ListRec` model with different baseline methods and other competing approaches and show that our model delivers better precision in terms of the prediction of the subscribed lists of the *twitterers*. Furthermore, we also demonstrate the importance of combining different weighting schemes and their effect on capturing users' interest towards Twitter lists. To evaluate the LIST-PAGERANK model, we employ a user-study based evaluation to show that the model is effective in recommending auxiliary lists that are more authoritative than the lists subscribed by the users.

Analysis of Kickstarter Crowdfunding Domain: Crowdfunding has gained widespread popularity in recent years. By funding entrepreneurs with creative minds, it is gradually taking over the role of venture capitalists who provide the much needed seed capital to jump start business ventures. Despite the huge success of the crowdfunding platforms, not every project is successful in reaching its funding goal. Therefore, in [2] we answer the following question "what set

of features determine a project's success?". We begin by studying the dynamics of Kickstarter, a popular reward-based crowdfunding platform, and the impact of social networks on this platform. Contrary to previous studies, our analysis is not restricted to project-based features alone; instead, we expand the features into four different categories: temporal traits, personal traits, geo-location traits, and network traits. Using a comprehensive dataset of 18K projects and 116K tweets, we provide several unique insights about these features and their effects on the success of Kickstarter projects. Based on these insights, we build a supervised learning framework to learn a model that can recommend a set of investors to Kickstarter projects. By utilizing features from the first three days of project duration alone, we show that our results are significantly better than the previous studies.

Group Recommendation Models for Crowdfunding: As an extension to the analysis on Crowdfunding platforms, our recent work [3] proposes a probabilistic recommendation model, called *CrowdRec*, that recommends Kickstarter projects to a group of investors. Being a highly heterogeneous platform, Kickstarter is fueled by a dynamic community of people who constantly interact with each other before investing in projects. Therefore, the decision to invest in a project depends not only on the preference of individuals, but also on the influence of groups that a person belongs and the on-going status of the projects. The proposed *CrowdRec* seamlessly incorporates the on-going status of projects, the personal preference of individual members, and the collective preference of the group . Using a comprehensive dataset of over 40K crowdfunding groups and 5K projects, we show that our model is effective in recommending projects to groups of Kickstarter users.

Trip Recommendation in Location-based Social Networks (LBSNS): The pervasive growth of location-based services such as Foursquare and Yelp has enabled researchers to incorporate better personalization into recommendation models by leveraging the geo-temporal breadcrumbs left by a plethora of travelers. In our research [4], we explore *Travel path recommendation*, which is one of the applications of intelligent urban navigation that aims in recommending *sequence of point of interest* (POIs) to tourists. Currently, travelers rely on a tedious and time-consuming process of

searching the web, browsing through websites such as Trip Advisor, and reading travel blogs to compile an itinerary. On the other hand, people who do not plan ahead of their trip find it extremely difficult to do this in real-time since there are no automated systems that can provide personalized itinerary for travelers. To tackle this problem, we propose a tour recommendation model that uses a probabilistic generative framework to incorporate user's categorical preference, influence from their social circle, the dynamic travel transitions (or patterns) and the popularity of venues to recommend sequence of POIs for tourists. Through comprehensive experiments over a rich dataset of travel patterns from Foursquare, we show that our model is capable of outperforming the state-of-the-art probabilistic tour recommendation model by providing contextual and meaningful recommendation for travelers.

1.3 Organization

The rest of this thesis is organized as follows. We begin by surveying the related work on recommendation algorithms in Chapter 2 and delineate our work on ranking tweet content and recommending Twitter List using regression and PageRank models. Chapter 4 will explain the characteristics of the Kickstarter domain and provide various interesting outcomes from the feature analysis of this domain. This chapter will also illustrate a simple hybrid recommendation model based on gradient boosted classifier to recommend backers to Kickstarter projects. The concept of group recommendation will be introduced to the readers in Chapter 5, where we will outline the challenges associated with this form of recommendation and propose our model called CrowdRec and its generative process. Finally, in Chapter 6 we will illustrate the notion of Travel recommendation by providing some statistical insights about the behavior of travelers and propose a *social sequential tour recommendation model* for travelers.

CHAPTER 2: BACKGROUND ON RECOMMENDATION MODELS

The main objective of all recommender systems is to obtain a utility function that estimates the preference of a user towards an item. Typically a recommendation system can also be viewed as a ranking engine, which presents users with a summary list of items in some order. The user then interacts with the ranked list to receive more details about the item. Essentially, recommender systems can be divided into two main categories: collaborative filtering methods and content-based methods. Collaborative filtering-based techniques utilize the action of users selecting items (i.e. rating, like etc) as input to obtain the similarity between users (or items). Content-based approaches on the other hand, utilize the content information of the items to provide recommendations. This information might include user's topical interest, item description, age, gender, etc. We begin this chapter by introducing the fundamentals of content and collaborative filtering (CF) techniques, where we explain various model- and memory-based methods. In the subsequent sections, we will demonstrate advanced recommender techniques using latent factors models involving matrix factorization and probabilistic generative models. In Section,3.3, we delineate our work on developing hybrid recommendation models using ridge-regression and topic-specific PageRank. More specialized models such as Group recommendation, and tour recommendation models will be described in the chapters dedicated for such models.

2.1 Content Based Recommendation

Content-based recommendation system learns the user behavior exclusively from the features of the objects rated by the user. These features can be in the form textual representation of the object or other meta-data information that exhibits the characteristics of the object. Irrespective of the characteristics of the object, the first step in building a recommender framework is to pre-process the data to extract these features. This completely depends on whether the data is structured or unstructured. For example, if we are building a recommendation model for suggesting News feeds, the feature extraction stage would involve extracting keywords, n-grams, concepts, sentiments etc. Once the information is extracted, we can represent these items by a vector of attributes. For text-based attributes, the feature space is usually represented as a vector of term weights, where each

weight indicates the degree of association between the item and the term. There are numerous ways to calculate these terms weights, but the most widely used techniques involve the following concepts: (a) *Term-Frequency*, which is based on the observation that multiple occurrences of a term in a document are not less relevant than single occurrences and (b) *Inverse Document Frequency*, which is based on the observation that frequently used terms are less relevant (or important) than rare terms. In paper [5], the authors propose a news recommendation model called *News@hand* that makes use of Semantic Web technologies to represent item features. The attributes of the news articles are represented as TF-IDF scores in the space of concepts defined in the ontologies. Textual content of items can also take the form of social tags which are keywords generated by the users. For instance, in paper [6], the authors represent the user profile as a tag vector, where the weights indicate the number of times a tag has been assigned to a document; [7] on the other hand, adopt a more sophisticated approach by matching the user-tag co-occurrence using *WORDNET* [8]. Besides these classic techniques, plethora of works exists on learning the term weights from the item's textual content [9–13]. In a recent work Gu et. al. [14] propose a unified method that can simultaneously learn the weights of multiple content matching signals and global term weights. In addition to content based features, other meta-data such as likes, ratings and comments can also be extracted to enhance the feature space. For example, the authors of paper [15] extract publisher, date, ISBN, price, etc., to recommend books for Amazon users using a Naive Bayes classification model. Papers [16, 17] target movie recommendation by learning the synopses extracted from the Internet Movie Database (IMDB). The synopsis of the movies are represented by feature vectors that contain weighting for word with noun tags and noun phrases, where the phrases are weighted according to its importance in the synopsis. Similar to this idea, paper [17] proposes a model called *Movies2GO* that learns user preferences from the synopsis of movies rated by the user; however, unlike conventional method of ranking items, this paper uses a novel voting scheme that allows multiple individuals with conflicting preferences to arrive at an acceptable compromise. Content-based systems also adopt the concept of relevance feedback to

refine the recommendation model to incrementally refine queries based on previous search results [18–20].

Recent works on content-based systems include the extracting of stylistic features such as lighting, color, and motion for video recommendations [21] and developing recommendation models for spoken documents[22]. In [22], the authors develop recommendations for internet audio (i.e. spoken document) by extracting content-based features to characterize non-linguistic aspects of the audio such as speaker, language and gender. Paper [23] on the other hand, combines search and recommendation system that refines the learners from the re-ranking results to enhance the recommendation performance and [24] study the impact of content uniqueness for group recommendation systems. For a comprehensive summary of collaborative filtering techniques, the readers are referred to survey articles [25, 26].

2.2 Recommendation using Collaborative Filtering

2.2.1 Memory-Based Techniques

Although content-based techniques are extremely popular and performs reasonably well even with the lack of user ratings, they are limited by the features that are explicitly associated with the objects that they recommend. For example, consider the scenario of recommending a movie I to two users U_1 and U_2. Let us assume that the movie is characterized by a set of attributes $\mathcal{A}$. These attributes can be the textual contents of the movies or it's meta-data information such as plot, description, names of the cast members etc. When recommending, the content-based system assumes that the attribute values of the user U_1 is independent from the attribute values of U_2, i.e. relationships of the form $U_1 \times U_2$ are completely ignored. Collaborative systems (in short CF) on the other hand, rely only on user ratings and can be used to recommend items without any descriptive data. The greatest strength of CF algorithms is that it captures the correlation between the users to make robust recommendations that are more accurate than content-based ones [27]. As explained by the survey [28], this allows the CF algorithms to perform out-of-the box recommendations. For instance, it is possible for a person who likes pop music to also enjoy music from other genres such as classical, jazz, country etc. A content-based recommender system

trained on the features pop albums will not be able to suggest jazz or classical albums since there will be very little overlap of features (i.e., performers, instruments, mood, tempo etc.) between two different genres of music.

In Collaborative filtering, the training dataset is represented as a $U \times I$ matrix with U representing the number of users and I representing the items. The values of this matrix can be the rating provided by the user to the items or implicit binary values indicating the binary actions such as likes and clicks. Memory-based CF algorithms creates a prediction function by utilizing the entire training sample of the user-item database. Usually, the prediction function measures the similarity between the users using some distance measure. The popular ones include neighborhood-based CF algorithm [29], which adopts the following steps to calculate the user-user similarity: (1) calculate similarity weight $w_{i,j}$ between two users (or items) i and j, which indicates the correlation between two users (2) find the k most similar users (i.e. neighbors) after computing the similarities and (3) aggregate the weighted similarity scores of the neighbors to rank the unseen items and recommend the top-N items to the user. One of the earliest and the most popular research on memory-based CF techniques can be seen from paper [30] that proposes a recommendation system for the retail bookseller Barnes and Noble and [31] that proposes a recommendation system for the e-commerce website *Amazon.com*.

2.2.2 Model-Based Techniques

Although simple to implement and interpret, memory-based CF techniques come with several limitations. First, similarity between two users are based directly on the commonly rated/selected items and therefore these methods become unreliable when the data is sparse. Second, memory-based techniques require the entire data to calculate the user similarity, which makes these algorithms extremely slow, thus hampering the scalability. To overcome this issue, researchers proposed models that work by factorizing the user-item matrix to a lower dimension called *latent-factors*. The preference of users towards items is then calculated based on this latent dimension to perform recommendation. These methods can also be viewed as dimensionality reduction techniques that utilizes algorithms such as *Singular Value Decomposition* [32], *Principal Component*

Analysis (PCA) [33]. Nonetheless, from a recommendation context, the most popular latent-factor models span from two areas of research namely: (a) Matrix-factorization (MF) [34, 35] and (b) Probabilistic latent factor (PMF) models. First, we briefly explain the rationale behind MF techniques and then provide details about PMF models, which is the main focus of this thesis.

Matrix Factorization for CF: the key assumption made by any form of latent-factor model is that a large user-item matrix $U \times I$ (usually $I >> U$) can be reduced to a low-dimensional latent space of two individual components $\theta_u \in \mathbb{R}^K$ and $\phi_i \in \mathbb{R}^K$ of users and items respectively, where where $K << I$. The rating of a user u towards an item i is then estimated using the following function:

$$\hat{r}_{ui} = \theta_u^T \phi_i \tag{2.1}$$

In the above equation $\hat{r}_{ui}$ is the estimated rating of an item i by the user u, this equation is also termed as the *hypothesis function*. The classic paper on MF for recommendation was proposed by Korean et. al. [34]. In this paper, the cost function between the estimated and the actual rating is formulated as follows:

$$C = \sum_{u,i \in \text{observed ratings}} (r_{ui} - \theta_u^T \phi_i) + \lambda (\sum_u \|\theta_u\|^2 + \sum_i \|\phi_i\|^2) \tag{2.2}$$

where r_{ui} is actual set of observed rating (i.e. ground truth), $\theta^T \cdot \phi$ is the estimated rating, and λ is the regularization parameter that penalizes the L2-norms $\|\theta_u\|^2$ $\|\phi_i\|^2$ in-order to avoid over-fitting. The objective is to minimize the cost function w.r.t to the parameters ϕ and θ, which is performed using a technique called *alternating least squares* (ALS). The steps to perform ALS is briefly described as follows:

- Fix the user parameter θ and solve the quadratic function for item parameter ϕ_i^j.
- Once item parameters are estimated, use this updated set of item factors to estimate user parameter θ_u^j.
- Repeat step 1 and 2 until convergence.

Although equation 2.2 can be solved using techniques such as stochastic gradient descent (SGD), ALS is a better algorithm due to the following reasons. First, by holding the user or

item parameter constant, equation 2.2 essentially becomes a convex function and hence the minimization should converge at some point. Second, unlike SGD, by holding θ fixed and estimating ϕ (or vise versa), ALS allows faster convergence since the algorithm finds the absolute minimum at each step and does not take small steps in the downward direction of the slope. The formulation of the cost function in equation 2.2 can also be extended to incorporate user and item biases to capture the uncertainty in the rating system. For example, consider two users A and B who like an item I. In a rating system that allows users to rate items on a scale of 1-5, let us assume that user A gives a rating of 5 for the item, while B provides a rating of 3. This does not mean, user B dislikes the item. It simply means user B is more critical than A. On the contrary, it could also mean that user A is more lenient in rating. Therefore, in-order to accommodate such user biases, we can reformulate the cost function as follows:

$$C = \sum_{u,i \in \text{observed ratings}} (r_{ui} - \theta_u^T \phi_i) + \lambda(\sum_u (\|\theta_u\|^2 + b_u^2) + \sum_i (\|\phi_i\|^2 + c_i^2)) \tag{2.3}$$

In addition to the above formulation, another popular variant of the MF-based recommender system modifies the hypothesis function for *implicit feedback data*. Not all recommender systems are rating-based; in fact, most of the web-data is associated with implicit feedback where users expresse their preference towards items using hidden actions such as clicks, page scrolls, likes and wish lists. In other words, a user *need not exclusively buy an item or rate an item to express his/her interest*. To model such implicit feedback Hu et al., [35] introduce a boolean factor $p_{ui} \sim \theta_u^T \phi_i$, where p_{ui} is a binary value that is determined based on the level of interaction between the user and the item. For instance, $p_{u,i}$ can be 1 if the user likes or adds an item to his wish list. The cost function for this implicit MF formulation is defined as follows:

$$C_{imp} = \sum_{u,i \in \text{observed ratings}} \delta_{ui}(r_{ui} - \theta_u^T \phi_i) + \lambda(\sum_u \|\theta_u\|^2 + \sum_i \|\phi_i\|^2) \tag{2.4}$$

where $\delta_{ui} := 1 + \lambda r_{ui}$ is the confidence in p_{ui}, which penalizes the cost function for incorrect prediction. So far, we provided an overview of the matrix factorization techniques for recommen-

dation. For a detailed understanding of MF algorithms, readers can refer the recent works such as [36–38] and [39].

Probabilistic Matrix Factorization: As explained earlier, the primary focus of this thesis is to tackle the challenges of recommendation systems from the *perspective of probabilistic latent models*. Therefore, we begin by introducing one of the earliest works on *generative latent factor models* for recommendation [40]. Although this model was originally projected as a variant of matrix factorization technique, the formulation is much closer to a class of models known as *probabilistic graphical models* (PGMs). Probabilistic graphical model is a vast area of research; nonetheless, our research is confined to a sub-class of PGMs known as *Generative models for Directed Acyclic Graphs*. For a detailed understanding of probabilistic graphical models the readers are suggested to survey the book by Koller et al. [41]. The generative model of PMF is shown in Figure 2.1, where the nodes represent random variables, V represents an item, U represents a user and the plate notations imply a set of M items and N users. Relationship between the random variables are represented as directed edges. The generative process is explained as follows:

- For each user u, draw a user latent factor: $\theta_u \sim \mathcal{N}(0, \lambda_\theta^{-1} I_K)$
- For each item i, draw item latent factor: $\beta_i \sim \mathcal{N}(0, \lambda_\phi^{-1}) I_K)$
- For each user-item pair (u, i), draw a rating (or feedback): $r_{ui} \sim \mathcal{N}(\theta_u^T \phi_i, c_{ui}^{-1})$

Parameters θ and ϕ are estimated using a *maximum a posteriori* (MAP) estimate. Similar to the above formulation, c_{ui} represents the confidence on the response r_{ui}. Our research work essentially focuses on such generative models, where *an action is explained using a set of random variables connected via directed edges*. More specifically, we focus on a class of generative models called the *topic models*. We begin by explaining a primitive topic model called *probabilistic latent semantic analysis* (PLSA) for recommendation. We then gradually add more complexity to this model to explain the popular document-topic model called *latent dirichlet allocation* (LDA) in the upcoming sections.

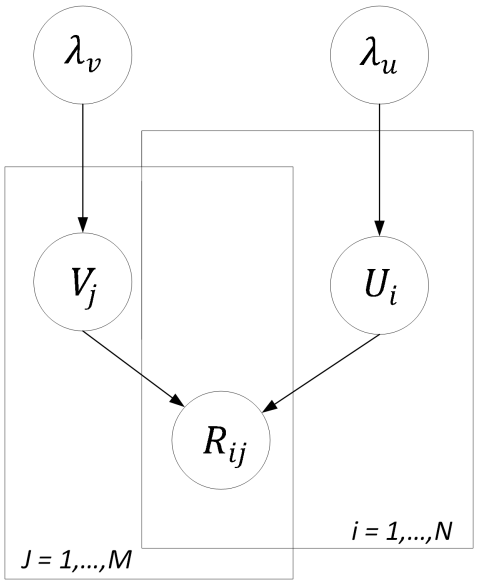


Figure 2.1: Graphical Structure of the Probabilistic Matrix Factorization

2.2.3 Probabilistic Latent Models

In this section, we will be discussion two important models (a) Probabilistic Latent Semantic Analysis and (b) Latent Dirichlet Allocation. These two models will serve as the foundation for our work in the later chapters.

Probabilistic Latent Semantic Analysis (PLSA) for Recommendation: The probabilistic latent semantic analysis is a generative model, which laid the foundation for the research on topic models. Although there are several variations of the PLSA algorithm, the very first model was proposed by Hoffman et al., [42] for document and word clustering. The authors extended the same framework for collaborative filtering in paper [43]. The PLSA framework is also called as the *aspect model* and the traditional way of estimating the parameters is to use the expectation maximization (EM) algorithm. The graphical structure of PLSA is shown in Figure 2.2, where U denotes the set of users $\{u_j\}_{j=1}^{U}$ and I denotes the set of items $\{i_m\}_{m=1}^{I}$. The variable Z is basically the latent factor of the model. With the graphical structure and the notations defined, the generative process of PLSA can be explained as follows:

- User u chooses a topic of interest z.

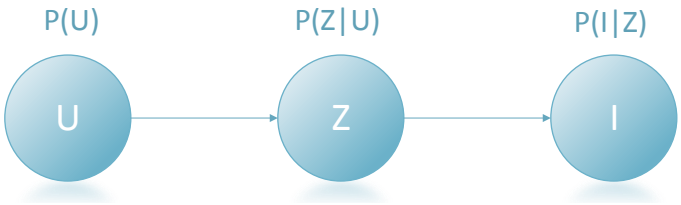


Figure 2.2: Graphical Model of PLSA: Variant1

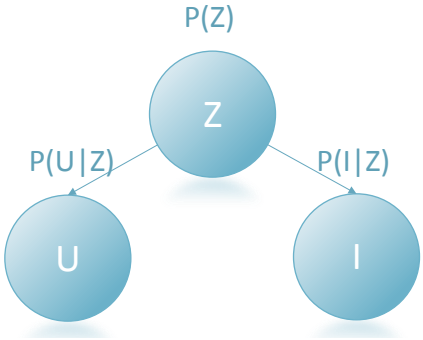


Figure 2.3: Graphical Model of PLSA: Variant2

- Based on this z, the user then selects an item i.

The central assumption of the model is that *users choose items based on some hidden interests* z and the goal is to infer these hidden interests. The generative model shown in Figure 2.3 is similar to 2.2; except, the graphical model is re-structured to fit the d-separation property which states that once the latent interest of the user z is known, the selection of item i is independent of the user u. The central inferential problem of PLSA is to estimate the posterior probability given by:

$$\begin{aligned} p(z|u,i) &= \frac{p(z,u,i)}{p(u,i)} \\ &= \frac{p(u)p(z|u)p(i|z)}{\sum_{Z} p(u)p(z|u)p(i|z)} \end{aligned} \tag{2.5}$$

We can ignore the normalization factor in the above equation and proceed by finding the log-likelihood of the numerator. The probability of a user u choosing an item i can be obtained by marginalizing over the variable z as follows:

$$p(u,i) = \sum_{Z} p(z,u,i)$$
$$= \sum_{Z} p(u)p(z|u)p(i|z) \tag{2.6}$$

Since we have U users and I items, the complete likelihood for the dataset is defined as follows:

$$D = \prod_{U}\prod_{I} p(u,i)^{n(u,i)} \tag{2.7}$$

where $n(u,i)$ is the number of times a user u picked an item i. Taking the log of the likelihood function we obtain the following equation:

$$\mathcal{L} = \sum_{U}\sum_{I} n(u,i) log(u,i) \tag{2.8a}$$
$$= \sum_{U}\sum_{I} n(u,i) \log [\sum_{Z} p(u)p(z|u)p(i|z)] \tag{2.8b}$$

Since, in our case, variable z is latent, explicitly finding the maximum likelihood estimates of the parameters is hard. Therefore, we maximize the expected value of the log-likelihood as follows:

$$\mathbb{E}(\mathcal{L}) = \sum_{U}\sum_{I} n(u,i) \sum_{Z} \log p(u)p(z|u)p(i|z)$$
$$= \sum_{U}\sum_{I} n(u,i) Q(z) \sum_{Z} \log p(u) + \log p(z|u) + \log p(i|z) \tag{2.9}$$

where the Q-function $Q(z)$ is generally set to the posterior $p(z|u,i)$. In the E-step we obtain this posterior probability and for the M-step we maximize the modified version of the log-likelihood that is obtained by introducing lagrange multipliers as follows:

$$\mathcal{H} = \mathbb{E}(\mathcal{L}) + \alpha[1 - \sum_{U} p(u)] + \beta \sum_{U}[1 - \sum_{Z} p(u|z)] + \gamma \sum_{I}[1 - \sum_{Z} p(i|z)] \tag{2.10}$$

The Lagrange Multipliers α, β, and γ can be obtained by taking the partial derivatives of the equation (2.10) w.r.t $p(u)$,$p(u|z)$, and $p(i|z)$. When simplified the parameters $p(u)$,$p(u|z)$, and

$p(i|z)$ equates to the following expressions:

$$p(u) = \frac{\sum_I \sum_Z n(u,i)p(z|i,u)}{\sum_U \sum_I \sum_Z n(u,i)p(z|i,u)} \tag{2.11a}$$

$$p(i|z) = \frac{\sum_U n(u,i)p(z|i,u)}{\sum_I \sum_U n(u,i)p(z|i,u)} \tag{2.11b}$$

$$p(z|u) = \frac{\sum_I n(u,i)p(z|i,u)}{\sum_Z \sum_I n(u,i)p(z|i,u)} \tag{2.11c}$$

Using the above set of equation, the EM algorithm is defined as follows:

1. **Initialization:** Begin by randomly initializing the parameters $p(u)$, $p(u|z)$, and $p(i|z)$.
2. **E step:** Estimate the posterior $p(z|u,i)$ using Equation 2.5.
3. **M step:** Use the newly calculated posterior from Step 2 to recalculate the parameters using expressions 2.11(a-c).
4. Repeat **Steps 2 and 3** until convergence.

Latent Dirichlet Allocation (LDA): A Recommendation Perspective: Proposed by Blei et al., [44], LDA is the most popular topic model for document and word clustering. Although the generative model of the document-topic generation is very similar to PLSA, LDA introduces Dirichlet priors that smooths the topic mixture in individual documents and the word distribution, thus alleviating the overfitting problem of PLSA. In this section, we provide the details of the LDA and derive the collapsed Gibbs Sampling process to infer the parameters. Instead of explaining LDA from a document-word perspective, we explain it from a recommendation perspective where documents are users and words are the items selected by these users. To explain the generative process, we provide a toy example in Figure 2.4, which illustrates the decision making process of a user u to watch a movie v. Since this is just an illustrative example, we have just four movies and genres (or topics); in reality, the topic and movie space can be extremely large. The following steps delineate the generative process of the model:

- Each user u has multiple topical interests (in our example, genres). Therefore, he can choose items (in our example, movies) v from various genres. In Figure 2.4, the user u is interested

in *comedy, horror, drama* and *action* genres. The user's interest over the set of genres follow a dirichlet distribution $\theta \sim Dirichlet(\alpha)$ with hyper-parameter α.

- From distribution of genres θ, the user then chooses a *single genre* z using a multinomial over the distribution θ_u.
- After selecting a genre z, u then picks a *movie* v from the genre-movie distribution matrix $\phi \sim Dirichlet(\eta)$ with hyper-parameter η. The movie is drawn using a multinomial distribution over the distribution ϕ.

However, in reality, we do not observe the user's distribution of interest θ, which implies we do not observe z, which in-turn means that we do not observe the movie-topic distribution either. The only observed variables are the users and the items. Therefore, the central inferential problem of LDA is to estimate θ and ϕ from the observed variables N (movies) and U (users).

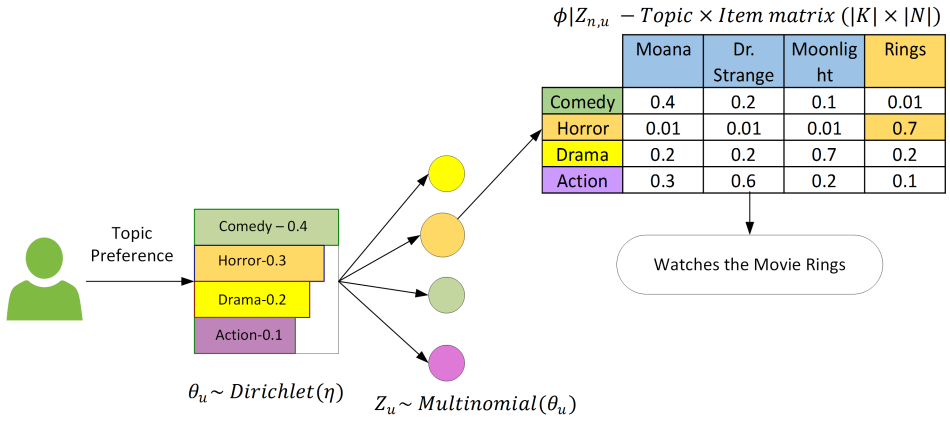


Figure 2.4: Plate notation of the LDA based recommender model

The formal representation of the LDA's generative framework is depicted in Figure 2.5. The posterior for the model is defined as follows:

$$p(\theta, \phi, z|w, \alpha, \beta) = \frac{p(\theta, \phi, z, w|\alpha, \beta)}{p(\theta, \phi, z, w|\alpha, \beta)} \tag{2.12a}$$

$$= \frac{p(\theta, \phi, z, w|\alpha, \beta)}{p(w|\alpha, \beta)} \tag{2.12b}$$

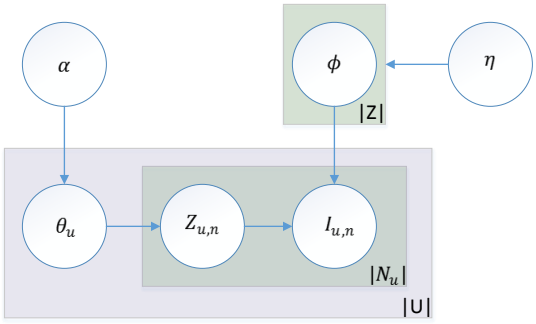


Figure 2.5: Plate notation of LDA based recommender model

Based on the generative model, the likelihood can be expanded into the following expression:

$$p(\theta, \phi, z, i|\alpha, \beta) = p(i|z, \phi)p(z|\theta)p(\theta|\alpha)p(\phi|\eta) \tag{2.13a}$$

More precisely, for a *single user* the likelihood is defined by:

$$p(\theta_u, \phi, z_u, v_u|\alpha, \beta) = \prod_{n=1}^{N_u} p(v|\phi_{z_{u,n}})p(z_{u,n}|\theta_u)p(\theta_u|\alpha)p(\phi|\eta) \tag{2.14a}$$

where N is the number of movies and N_u denotes the movies chosen by the user u. Unfortunately, the denominator (i.e. the marginalizing constant) of the equation 2.12a is intractable to compute. Therefore, we use an approximate inference technique called Gibbs sampling to obtain the posterior of LDA. The following paragraph describes derivation of Gibbs sampler for LDA.

Collapsed Gibbs Sampling: It is important to note that although we need to obtain the complete likelihood $p(\theta, \phi, z|w, \alpha, \beta)$ in the equation 2.12a, this expression can be collapsed by excluding θ and ϕ. This is because, once we infer z_i, which is the topic assignment for each item $\{v_i\}_{i=1}^{N}$, it is sufficient to calculate the user-topic distribution θ and item-topic distribution ϕ. Hence, the *collapsed posterior* of the Gibbs sampler is written as follows:

$$p(z|v, \alpha, \beta) = \frac{p(z, v|\alpha, \beta)}{p(w|\alpha, \beta)} \tag{2.15}$$

According to the Gibbs sampling procedure, when sampling a topic z for an item v, we assume that all other topic assignments for items other than v are known. Applying this rule to equation

2.15, we obtain the following posterior:

$$p(z_i|Z^{\neg i}, \alpha, \beta, v, N) = \frac{p(z_i, Z^{\neg i}, v, N|\alpha, \beta)}{p(z^{\neg i}, v, N|\alpha, \beta)} \tag{2.16}$$

$$\propto p(Z, N|\alpha, \beta) \tag{2.17}$$

where, $Z^{\neg i}$ indicates the set of topic assignments for all movies expect movie v_i. Now, using the graphical model, the numerator can be expanded as follows:

$$p(Z, N|\alpha, \beta) = \underbrace{p(N|Z, \beta)}_{(A)} \cdot \underbrace{p(Z|\alpha)}_{(B)} \tag{2.18}$$

Part (A) of the above expression is expanded into the following components:

$$p(V|Z, \beta) = \int p(V|Z, \phi) \cdot p(\phi|\beta) d\phi \tag{2.19}$$

In equation (2.19), $p(\phi|\beta)$ follows a dirichlet distribution and $p(V|Z, \phi)$ follows a multinomial distribution defined by:

$$p(\phi|\beta) = \prod_{k=1}^{K} \frac{1}{\Delta(\beta)} \prod_{v=1}^{N} \phi_{k,v}^{\beta_v - 1} \tag{2.20a}$$

$$p(V|Z, \phi) = \prod_{k=1}^{K} \prod_{v=1}^{N} \phi_{k,v}^{\psi_{k,v}} \tag{2.20b}$$

here ψ_{kv} is a count matrix indicating the number of times item v is assigned to the topic k, and $\Delta(\beta)$ is the gamma function defined as follows:

$$\Delta(\beta) = \frac{\prod_{i=1}^{|\beta|} \Gamma(\beta_i)}{\Gamma(\sum_{i=1}^{\beta_i} \beta_i)} \tag{2.21}$$

Substituting equations (2.20a) and (2.20b) in eqation (2.19) we obtain the following expression:

$$p(V|Z,\beta) = \int \prod_{k=1}^{K} \frac{1}{\Delta(\beta)} \prod_{v=1}^{N} \phi_{k,v}^{\beta_v - 1} \prod_{k=1}^{K} \prod_{v=1}^{N} \phi_{k,v}^{\psi_{k,v}} d\phi_k \tag{2.22a}$$

$$= \prod_{k=1}^{K} \frac{1}{\Delta(\beta)} \int \prod_{v=1}^{V} \phi_{k,v}^{\psi_{k,v} + \beta_v - 1} d\phi_k \tag{2.22b}$$

$$= \frac{\Delta(\psi_k + \beta)}{\Delta(\beta)} \tag{2.22c}$$

Where equation (2.22c) is obtained using the property $\int \prod_{v=1}^{|\alpha|} p_v^{x_v + \alpha_v - 1} dp = \Delta(x + \alpha)$. Deriving part (B) of equation (2.18) in the similar fashion yields the following expression:

$$p(Z|\alpha) = \frac{\Delta(\Omega_u + \alpha)}{\Delta(\alpha)} \tag{2.23}$$

where Ω_u is the count indicating the number of times a user u is assigned to topic z. Finally, we can obtain $p(Z, N|\alpha, \beta)$ using expressions (2.22c) and (2.23) as follows:

$$p(Z, N|\alpha, \beta) = \prod_{k=1}^{K} \frac{\Delta(\psi_k + \beta)}{\Delta(\beta)} \cdot \prod_{u=1}^{U} \frac{\Delta(\Omega_u + \alpha)}{\Delta(\alpha)} \tag{2.24}$$

As mentioned earlier, since the posterior cannot be estimated directly, we use the following Gibbs update rule:

$$p(z_i|N, Z^{\neg i}, \alpha, \beta) = \frac{p(Z, N|\alpha, \beta)}{p(Z^{\neg i}, N^{\neg i}|\alpha, \beta)} \tag{2.25}$$

The numerator of the above expression was derived in equation (2.24). The denominator of this expression is defined by:

$$p(Z^{\neg i}, N^{\neg i}|\alpha, \beta) = \prod_{k=1}^{K} \frac{\Delta(\psi_k^{\neg i} + \beta)}{\Delta(\beta)} \cdot \prod_{u=1}^{U} \frac{\Delta(\Omega_u^{\neg i} + \alpha)}{\Delta(\alpha)} \tag{2.26}$$

Substituting equations (2.23) and (2.26) we can obtain equation (2.25) as follows:

$$
\begin{aligned}
p(z_i = k|Z^{\neg i}, N, v, \alpha, \beta) &= \frac{\Delta(\psi_k + \beta)}{\Delta(\psi_k^{\neg i} + \beta)} \cdot \frac{\Delta(\Omega_u + \alpha)}{\Delta(\Omega_u^{\neg i} + \alpha)} \\
&= \frac{\frac{\Gamma(\psi_{k,v} + \beta_v)}{\Gamma(\sum_{v=1}^{N} \psi_{k,v} + \beta_v)} \cdot \frac{\Gamma(\Omega_{d,k} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} \Omega_{u,k} + \alpha_k)}}{\frac{\Gamma(\psi_{k,v}^{\neg i} + \beta_v)}{\Gamma(\sum_{v=1}^{N} \psi_{k,v}^{\neg i} + \beta_v)} \cdot \frac{\Gamma(\Omega_{u,k}^{\neg i} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} \Omega_{u,k}^{\neg i} + \alpha_k)}}
\end{aligned}
\tag{2.27a}
$$

Therefore, given a user u and a item i, the probability of assigning a topic z to the tuple (u, i) is then obtained by simplifying the Gamma functions in expression (2.27a). The final posterior expression is defined by:

$$
p(z_i = k|Z^{\neg i}, v, N^{\neg i}, \alpha, \beta) = (\Omega_{u,z}^{\neg i} + \alpha_z) \cdot \frac{\psi_{z,v}^{\neg i} + \beta_v}{\sum_{v=1}^{V} \psi_{z,v}^{\neg i} + \beta_v} \tag{2.28}
$$

It should be noted that, in this draft, we donot furnish all the nity grity details of the Gibbs sampling derivation of LDA. For complete details about the derivation, readers are suggested to survey the draft on *parameter estimation for text analysis* [45].

2.3 Summary

In this chapter, we provided a complete overview of the popular techniques in the field of Recommender systems. First, we furnished the research works on content-based recommendation which learns the user behavior exclusively from the features of the objects rated by the user. We then explained the flaws in this technique, and introduced the concept of collaborative filtering. The explanation about CF algorithms were split into (a) memory-based techniques and (b) model-based techniques. and the problems associated with memory-based techniques such as scalability and sparsity were delineated. Finally, we introduced probabilistic latent-factor models and explained the details of two important algorithms PLSA and LDA that will serve as the basics for understanding the upcoming chapters.

CHAPTER 3: LIST RECOMMENDATION IN TWITTER

3.1 Introduction

The increase in web contents in the form of social media websites, blogs, and news articles have resulted in the problem of information overload. Researchers have tackled the problem of information overload from different perspectives such as organizing trending topics in user's timeline, URL recommendations for *twitterers*, recommending followers and tweets [46–49]. A new direction of research that is proposed in this chapter is the development of personalized recommendation based on *social lists*. Lists serve a dual purpose in various social networks. First, they serve as a newsletter or a daily-digest for users who seek unified source of information. Second, they act as topical-hubs that unite users who share similar interests. Originally lists were introduced by Twitter in 2009; however, they have been adopted by various social networking websites in different forms under different names. For instance, Google+ terms lists as *social circles* and Facebook provides a feature called *community pages*. In general, every list has a curator who creates the list and makes it as private or public. Other users can freely subscribe to such public lists, while private lists are restricted to the owner's approval. Lists are one of the strongest indicators of topical homophily [50]. Consequently, they can be an excellent tool to smoothen the problem of information overload.

Recommending lists is a challenging task because most users create them for grouping friends or other users whom they find interesting. Such lists that are created for personal convenience do not gain the attention of people. This implies that most of them do not have any subscribers. Furthermore, list names are not unique; there can be thousands of lists with similar (or even same) names [51]. This further exacerbates the problem of finding genuine, authoritative and topically relevant set of lists. In this chapter, we propose two recommendation models that recommend lists for Twitter users based on their personalized interest. Our first model, called the `ListRec`, captures and models the users' interest based on a combination of content, network and trendiness based measures. For users with rich tweet history, we measure their interests using the topics derived from their tweets. Unlike the existing studies, we view the *twitterer's* interest as a temporally varying feature and exploit this variation using an exhaustive set of streaming tweets to dynami-

cally model the users' interest. For users with sparse tweet history, we project the user space into a followee space and utilize the followee's list subscriptions to indirectly measure the interest of the users. We also add a new trend based score that measures the popularity of lists in the Twitter domain. The final score is then modeled as a linear combination of these three individual scores (based on content, network, and popularity) to effectively measure the interests of the users and personalize list recommendation. The coefficients in this linear combination are estimated using a cyclic ridge regression estimation approach. Our experimental results show that the `ListRec` outperforms other competing state of the art methods. Our second model is the LIST-PAGERANK which will recommend lists that are popular and are more (topically) authoritative than the lists that are currently subscribed by the users. To the best of our knowledge, there are no studies that use Twitter lists for personalized recommendation. We summarize the major contributions of this chapter as follows:

a. We propose a recommendation framework called `ListRec` that recommends Twitter lists based on the personalized interest of *twitterers*. Unlike the existing studies that recommend external information like news articles and blogs, our work is purely domain-specific.

b. The interests of users are modeled using a combination of weighting schemes: (a) a content based scheme that models the users' interest based on temporally varying topics; (b) a network based scheme that uses the followee-network of the users to overcome the tweet sparsity; and (c) a trendiness based scheme that is based on the popularity of the lists.

c. We propose a LIST-PAGERANK based algorithm that leverages the network structure of Twitter lists to recommend authoritative lists that match the topical interest of the users.

The rest of this chapter is organized as follows. We begin by describing the modeling of `ListRec` in Section 3.3. Section 3.4 describes the creation of the list network and formulation of the LIST-PAGERANK. Section 6.6 will show the results of our experiments and explain the data collection methodology. Section 3.2 discusses the related work on this topic. Finally, the conclusions obtained through this study are presented in Section 3.6.

3.2 Related Work

Over the past few years, researchers have proposed various methods to overcome the problem of information overload in social networks. These studies can be classified into three main categories: (a) reorganization of user timeline in microblogs, (b) topic modelling, and (c) personalized recommendation.

Reorganization of user timeline: The research on timeline reorganization aims to re-rank the timeline of users in microblogging network like Twitter. Feng et al. [52] build a feature-aware factorization model that uses the graph containing nodes in the form of users, publishers and tweets. They build their model based on the notion that the tweet history reveals user's personal preference. Bernstein et al. [46] adopt a topic based technique for organizing twitter feeds. In their work, the tweets are transformed into queries for external search engine. The popular terms are then assigned as topics. Burgess et al. use Twitter lists to tackle the problem of timeline reorganization. Since lists implicitly denote the topical interests of *twitterers*, they propose a system called *Butterworth* that can automatically build twitter lists by leveraging user's social network and the content generated by friends. Our work is different from the ones mentioned above since it uses a novel *list based PageRank algorithm*. None of these works mention about the topic of list recommendation which forms a core part of our work.

Topic modeling The use of topic models in microblogging has been extensively studied by many authors. Ramage et al. [53] presents a scalable implementation of labeled LDA. Phan et al. [54] use the LDA topic model for building short and sparse text classifiers. [47] propose a URL recommendation system for Twitter users. According to the authors, the topics in Twitter are presented by different concepts that change over time. The concepts are built using a linguistic model that detects entities and mentions from users' tweets. Our topic modeling method uses the notion of dynamic temporal LDA which is not captured by the methods mentioned above.

User Recommendation Unlike timeline reorganization that restricts itself to the ranking of tweets, the user recommendation tackles the information overload problem by providing users with contents or users that are relevant to the user's interest. In [55], a URL recommendation system for

twitter users is proposed which aims to recommend URLs by constructing a vector-of-words from users' tweets to measure their interest. In our previous work [56], we developed a model that recommends geo-location based tweet summaries. Analyzing the tweet's content and social graph for recommending friends and followers have been studied by various researchers [57], [48], [49]. There are very few studies that exploit the list feature in Twitter [58, 59]. In their studies, [58] rank the tweets within the subscribed lists of users rather than recommending existing lists. In short, their work is similar to the ranking of user's timeline and hence it is quite different from our work. The models proposed in our work are similar to the ones described in [60] and [61]. However, unlike these works, our paper leverages on the Twitter *lists* and temporal interest of users to design a new recommendation system.

3.3 Recommending Twitter List using Regression

In this study, we classify the Twitter users into two categories: *the persistent twitterers*, and the *active consumers*. *Persistent twitterers* are users who tweet frequently and consistently. Therefore, they tend to have a rich tweet history. On the other hand, *active consumers* are characterized by a sparse tweet history, but they actively consume information from Twitter by following other users. Our aim is to develop a list recommender system that can be effective for both these categories of users. For this reason, we use a combination of users' tweet history (when available), and their network of followees to collectively measure their personalized interest.

List-preference based on varying topical interests: The topical interest of *twitterers* changes with time. For example, consider the following set of tweets tweeted by a *twitterer* over a period of 1 year.

1. Love my #iphone 4s and its retina screen simply colorful and vibrant. #iphoneRocks
   - March 2012
2. #Apple versus #Samsung this is interesting. I think #iphone has lost it's charm
   - December 2012
3. Finally sold my #iphone4s and got a #GalaxyS4 simply loving the big screen!. Can't wait to explore the new #Android - June 2013

We can clearly see the transition of the user's interest from iphone to Galaxy S4 mobile. This also means that recommending lists related to iphone might not be interesting to this user. Therefore, we model the interest of *twitterers* as a temporally varying factor by using the discrete dynamic topic model (dDTM) [62] to create a *temporal topic-preference matrix* that captures the inclination of the users towards a set of topics at different time frames. Unlike LDA [44], the dDTM sees the order of collection as an evolving set of topics. The dDTM uses a state space model on the natural parameters of the multinomial distributions that represent the topics. The alignment among topics across time steps is captured by a Kalman filter. The inferior performance of topic models over short text documents is a well known problem that has been widely studied in the literature [63]. To overcome this problem, we use tweet pooling technique [64] to collect all the tweets tweeted by these users, and use their history of tweets as input to the dDTM. For the set of users $\mathcal{U}$ in our database, we run the dDTM over their tweet history to obtain the set of topics $\mathcal{T}$ at different time frames t_f. We then use these topics as an intermediate plane to formulate a *content-list* matrix that maps the topical interests of the *twitterers* to the set of lists L. We explain this mapping using the following set of matrices:

- *User-topic matrix J:* The topical interest of *twitterers* J for a time frame t_f is $|\mathcal{U}| \times |\mathcal{T}|$ matrix, where the value $\mathcal{UT}_{ij}$ denotes the number of times a word in *twitterer* u's tweet has been assigned to the topic $\tau_j \epsilon \mathcal{T}$.
- *Topic-List matrix M:* The topic-List matrix defines a relation between the set of lists and the topics that are spanned by these lists. We create this matrix by collecting the set of tweets that emerge from every list $l \epsilon L$, and use the dDTM to generate a set of topics. The topic-List matrix is represented as $M = |\mathcal{T}| \times |L|$.

The interest of *twitterers* towards the lists is a $|\mathcal{U}| \times |L|$ matrix that is obtained as follows:

$$\Phi = J \cdot M \tag{3.1}$$

Network based List-preference: For users with low tweeting frequency (i.e. the active consumers), we use their followee network to indirectly measure the preference of user $u \epsilon \mathcal{U}$ to a

set of lists $\{l_1, ..., l_n\}$ in L. First, we obtain the set of followees F for users in $\mathcal{U}$ to create a *user-followee* matrix given by

$$E = |\mathcal{U}| \times |F| \tag{3.2}$$

Second, the user's interest towards his followees is measured based on the number of times a user u_i retweeted his followee f_j. The adjacency matrix E is defined as follows:

$$E_{ij} = \frac{RT(i,j)}{\sum\limits_{f \in F} RT(i,f)} \tag{3.3}$$

where $RT(i,j)$ is the number of times the user i retweeted his followee j, and $\sum RT(i,f)$ is the total number of retweets by the user i (normalization factor).

Third, the list subscriptions of followees in the set F is retrieved to create a *followee-list* matrix $|F| \times |L|$ given by

$$J_{ij} = \begin{cases} 1 \text{ if } i \text{ subsribes to list } j \\ 0 \text{ otherwise} \end{cases} \tag{3.4}$$

Finally, we obtain the *network-list* matrix $|\mathcal{U}| \times |L|$ as follows:

$$\Delta = E \cdot J \tag{3.5}$$

List-preference based on Trending List: A list can be considered *trending* in Twitter if the hashtags produced by this list are popular at a specific time t. Therefore, for every list in the set L, we retrieve the hashtags that emerge from their respective tweets to create a hashtag-List matrix given by

$$K = |H| \times |L| \tag{3.6}$$

We then determine the *trending lists* by estimating the popularity of hashtags in the set H at a specific time t in the ordered Twitter streams $\mathcal{D}_1, \ldots \mathcal{D}_n$. Each Twitter stream $\mathcal{D}_i$ is a set of ordered n-tuples represented as $\{< h_{i1}, t_{i1} >, \ldots, < h_{im}, t_{im} >\}$ where h_i is the hashtag and t_i is it's corresponding publishing time. Kwak et al. [65] showed that the topics in Twitter become popular for a certain period of time and gradually die. This encourages us to use a time-decay function to

estimate the trending hashtags [66]. The numeric weight of hashtags H in the Twitter stream D at any given time is a function of the elapsed time since the first occurrence of this hashtag. The common way to model such functions is using an exponential-decay. Mathematically, we denote the function as follows:

$$W(h) = \sum_{<h_i,t_i>\epsilon\mathcal{D}} \beta^{\frac{[t_{now} - t_i]}{T}} \tag{3.7}$$

where the parameter $\beta\epsilon(0, 1]$ controls the weight of the hashtags; t_{now} denotes the current time, and T sets the granularity of time-sensitivity. In this work, we give equal importance to every hashtag at the beginning by setting β to 1. The trendiness of hashtags $h\,\epsilon\,H$ is measured by estimating $\hat{W}(h)$ using first order derivative of their cumulative counts. The trending-List matrix is given by

$$\Omega = \mathcal{H} \cdot K \tag{3.8}$$

where $\mathcal{H}$ is a row matrix that contains the estimated weights for the hashtags H.

Recommendation score of Twitter List: The recommendation of Twitter lists L for a set of users $\{u_1, ..., u_n\}$ given their tweet history Z_u and followees F_u is represented as a linear combination of their topic based weightage Φ, their network based score Δ, and the score based on list trendiness Ω. Formally, we denote the preference score by

$$P(u, l) = \alpha\Phi + \beta\Delta + \gamma\Omega \tag{3.9}$$

Ridge regression for list recommendation: We now describe the algorithm to estimate α, β and γ for the preference score (3.9). The ridge regression algorithm is used as a solver for estimation at each step of Algorithm 1. The regularization function used here is the L_2 norm of the regression coefficient vector. We now explain the major steps involved in this estimation algorithm.

In the first step, β and γ are initialized using a fitting heuristic. In this fitting heuristic, we estimate β^{init} by selecting a randomly sampled subset of data, and fitting it to the response vector Δ. The size of this sample is set of 30% of the original data. Similarly γ^{init} is also estimated using this 30% fitting heuristic. The values of β^{init} and γ^{init} are used in Equation (3.9) to formulate the

Algorithm 1: Cyclic Approximate Ridge Regression for List Recommendation

Input: Binary response vector P, Topic List matrix F, Network List Matrix Δ, Trending List Matrix Ω

1: Initialize β=β^{init} and γ=γ^{init}
2: $P^{'} = P - \Delta\beta^{init} - \Omega\gamma^{init}$
3: Using $P^{'}$ and F estimate α^{final}
4: Set $P^{'} = P - F\alpha^{final} - \Delta\beta^{init}$
5: Using $P^{'}$ and Ω estimate γ^{final} as in Step 3
6: Set $P^{'} = P - F\alpha^{final} - \Omega\gamma^{final}$
7: Using $P^{'}$ and Δ estimate β^{final} as in Step 3
8: Output α^{final},β^{final} and γ^{final}.

regression problem to estimate α^{final}. This cycle of estimation is continued in the remaining steps to estimate the values of β^{final} and γ^{final} as explained in Algorithm 1. This final set of coefficient values are used for estimating the scores.

3.4 Recommending Twitter List using PageRank

In our previous work [1], we proposed a LIST-PAGERANK model that can recommend auxiliary set of lists that are authoritative and topically similar to the lists that are subscribed by the *twitterers*. We begin this section by explaining the construction of the list network. We define the set of Twitter lists as a tuple L_c <C,M,J,S>, where C denotes the curator of the list; M is the set of list members; J is the set of topical words, and S is the set of subscribers. A directed graph $D(V, E)$ is formed with lists as the vertex V of the network. Defining edges in Twitter lists can be tricky. This is because, unlike user-follower relationship in Twitter, an explicit relationship between lists does not exist. Therefore, in this section, we exploit the hidden structure of Twitter lists to define their linkage. We say that, an edge between two lists exists *if the member of a list is a subscriber of another list*. Figure 3.1 shows this notion using three list nodes. In this figure, the user C who is a member of list 3 subscribes to another list 2; thus, establishing a linkage between these lists. Similarly, user G, a member of list 2 subscribes to list 3.

With the list network defined, we now explain the meaning of authority in Twitter list. The definition of authority is based on the following observations:

- *Influential twitterers tend to be a member of many lists.*

- *Lists containing influential twitterers have the potential to attract many subscribers. The subscription count in turn makes the list authoritative.*

This notion is similar to the real-life event of news paper subscription. Top circulated news papers like *The Wall Street Journal* and *The New York Times* attract more subscribers because the content produced by them are relevant and exhaustive; more importantly, they are written by prominent reporters and journalists. The influence of a *twitterer* can be measured by his list membership count. For example, the user C in Figure 3.1 is a member of two lists: list 1 and list 3. Now, the authority score of the list 2 goes up due to the presence of this influential member.

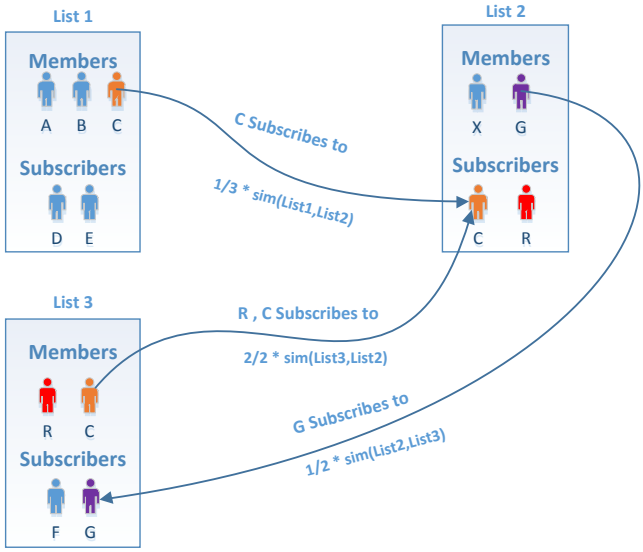


Figure 3.1: Representation of list-network using a subscriber-member relationship

We show this effect in Figure 3.1 by plotting the membership count of users against the subscriber count of their list subscription. We see that, as the membership count increases, the subscriber count also increases. The increase in subscriber count becomes more pronounced when the membership count goes beyond 80. This clearly shows that the subscription of users with high list membership results in attracting more subscribers; thereby, making the list more dominant. Our goal is to recommend auxiliary set of lists that are not only authoritative, but also topically similar to the lists that are subscribed by the *twitterers*. We now explain the creation of *list-topic* matrix J and the formulation of our LIST-PAGERANK.

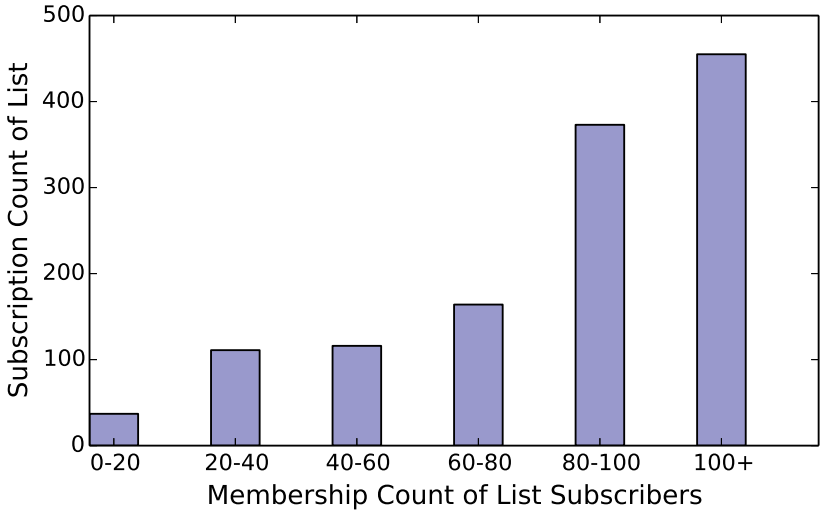


Figure 3.2: Influence of *twitterers'* membership count over their list subscription count

To create the *list-topic* matrix J, we obtain the topics from the auxiliary set of lists L_c by tokenizing the tweets from each list. We then construct a bag-of-words vector W, where $W =< tf(l, w_1), ..., tf(l, w_m) >$ and $tf(l, w)$ is the *term frequency* of the tweet word w in the list $l \, \epsilon \, L_c$. The matrix J is denoted by $|L_c| \times |W|$. We define the adjacency matrix for our list-network as follows:

$$LN_{i,j} = \begin{cases} \dfrac{|M(i) \cap S(j)|}{|M(i)|} \times Sim(i,j) & \textit{if link exists} \\ 0 & \textit{otherwise} \end{cases} \tag{3.10}$$

In the above equation, the *link exists* if atleast one member of list i is a subscriber of the list j, $M(i)$ is the set of all members of i, and $S(j)$ is the set of all subscribers of j. The numerator $|M(i) \cap S(j)|$ denotes the number of members of list i who are subscribers of list j, and the denominator $|M(i)|$ is the total number of members of the list i. An example of this formulation is shown in the Figure 3.1. In this example, there is just one user (user C), who is both a member of list 1 and a subscriber of list 2. Therefore, the edge weight between list 1 and list 2 is $1/3$. Finally, the similarity term $Sim(i, j)$ is calculated as the cosine similarity between the lists i and j given by

$$Sim(i,j) = \frac{J_i \cdot J_j}{|J_i| \times |J_j|} \tag{3.11}$$

where J_i and J_j are topic vectors obtained from the *list-topic* matrix J.

In our list-network, it is possible for lists to form loops. For example, in the Figure 3.1, a loop exists between list 3 and 2 since the member of list 3 is a subscriber of list 2 and vice versa. Such loops will accumulate high influence without distributing it. In the *random surfer* algorithm of PageRank, a link is added from every web page to all other web pages to overcome this problem. We adopt the same methodology in our list graph, by introducing the teleportation vector LW defined as:

$$LW = J_k \tag{3.12}$$

where J_k is the k-th column of the *list-topic* matrix J. In this manner, the teleportation probability is higher for a list which is more topically similar to the original list.

We finally represent the LIST-PAGERANK as a convex combination of the matrix $LN_{i,j}$ and the *teleportation* vector LW as follows:

$$L_{rank} = \alpha LN_{i,j} + (1 - \alpha) * LW \tag{3.13}$$

The addition of the teleportation vector LW enables the surfer visiting a list to jump to another random list with a probability $(1 - \alpha)$, where α is a parameter that controls the probability of teleportation that is set between 0 and 1.

3.5 Experimental Results

3.5.1 Dataset Description

In the earlier section, we categorized the users as *persistent twitterers* and *active consumers* based on their tweeting behaviour. In general, it is difficult to obtain users with such characteristics merely by querying the Twitter for random user Ids due to the API limitations. For this reason, we use our streaming database that was collected from January 2012 to August 2013 using Twitter's *firehose* API that provides 10% of every day's streaming tweets. Figure 3.3 shows the comparison of the tweet frequency plots between users who appear in over 60% of our database, and users who appear in less than 30% of our database. While both the plots follow a powerlaw distribution, the former shows a tweet count between 500-1000 for a majority of users, while the latter clearly

shows that most users have sparse number of tweets. We denote the set of users with high tweeting frequency by P, and those with low frequency by $\mathcal{A}$. We create our user dataset $\mathcal{U}$ as follows:

1. The set of *persistent users* is denoted by $P^* = \{p | p \epsilon P$ and p has atleast 3 list subscriptions$\}$.

2. Since the *active consumers* have a scarce set of tweets, we choose these users based on their followee count. Figure 3.4 shows that most users in the set $\mathcal{A}$ do not follow other users. Therefore, we impose a threshold on the followee count of the users. Formally, we denote the set of *active consumers* by $\mathcal{A}^*$, where $\mathcal{A}^* = \{a | a \epsilon \mathcal{A}$ and a has atleast 10 followees and 3 list subscriptions $\}$.

Our final user dataset is given by $\mathcal{U} = P^* \cup \mathcal{A}^*$. For our experiments, we have $|P^*| = 529$, $|\mathcal{A}^*| = 221$ and $|\mathcal{U}| = 750$.

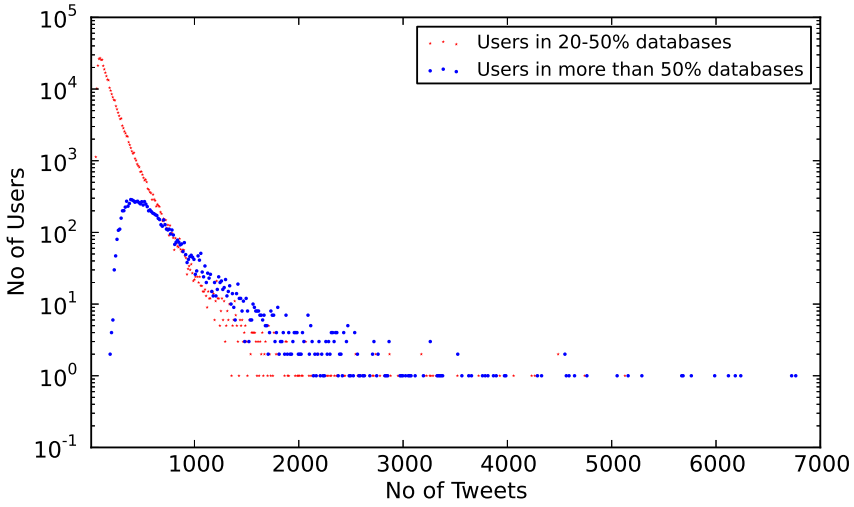


Figure 3.3: Tweeting frequency of frequent and infrequent *twitterers* from our streaming database

3.5.2 Automatic Evaluation

We evaluate the `ListRec` model based on the assumption that a user who subscribes to a list finds it interesting. Our test dataset is the set of all users in $\mathcal{U}$, and their list subscriptions L, $|L| = 2988$. Ideally, the correct recommendation for a user $u \epsilon \mathcal{U}$ should correspond to the lists from his own direct subscription.

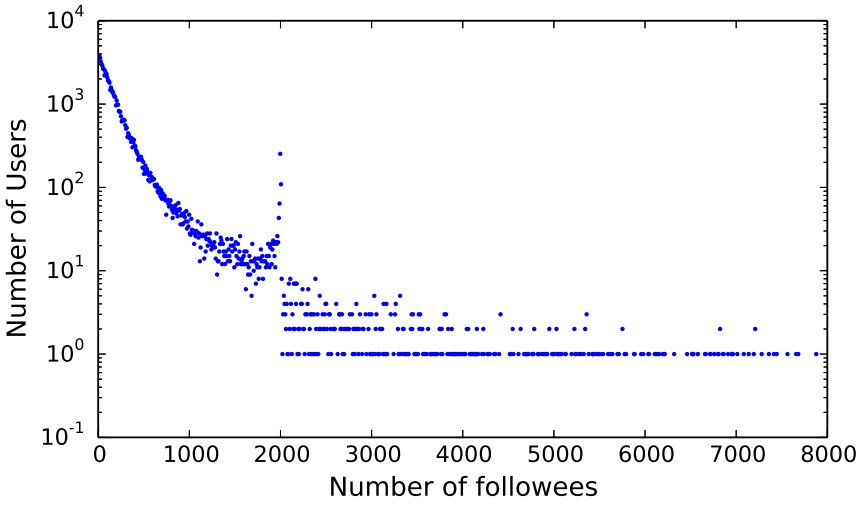


Figure 3.4: Number of followees per *active consumer*

Evaluation Metrics For evaluating our model we use the standard information retrieval measures. For every user we compute: (1) *precision at rank k* (P@k) for our task is defined as the fraction of rankings in which the subscribed lists is ranked in the *top-k* positions, (2) Our recommendation is correct when the user-subscribed list is present in the ranked set of lists. The *mean reciprocal rank* (MRR) is the inverse of the position of the first correct list in the ranked set of lists produced by our model, and (3) *success at rank k* (S@k) is the probability of finding at least one correct list in the *top-k* ranked ones. (4) The *discounted cumulative gain* (DCG) [67] is based on the simple idea that highly relevant lists are more important than marginally relevant lists. DCG computes the score for a list based on it's position in the ranked set of lists. It then calculates the cumulative gain by considering a linear summation of the relevance scores of the lists scaled by a logarithmic factor. The scaling helps in obtaining the discounted cumulative gain metric.

Method Comparison We compare the performance of our model to the following baselines

- `EntRank`: For every user, we collect the user entities mentioned in their tweets. The entity based ranking scheme ranks the lists based on the number of members who correspond to the entities mentioned in the user tweets.
- `Trendiness`: We set $\alpha = \beta = 0$ to rank the lists purely based on trendiness.

Table 3.1: Comparison of the past and current interest of users generated by dDTM, and the topics generated by LDA without taking the temporal shift of user interests

User	List Topics	Past Interests (dDTM)	Current Interests (dDTM)	LDA Topics
A	Influencial in tech {Android, #technology, microsoft }	food, volunteer, tech	Google, #tech, hackathon	Volunteer, fastfood latish, meal
B	Top 50 funny {#smile, Darwin , follow}	media, poll, breaking	Comdey, Science, actors	people, critics media, news
C	Astronauts in space {NASA, #ISS, Mars }	Game, #redsox, mars	#mars, NASA, astronauts	Redsox, win space, game
D	US Senators {politics, #syria, Obama }	Bills, business, venture	governor, syria, policy	startup, venture legislation, pay
E	Marketing Industry {adcampaign, business, media}	kobe, payments, ipad	eComm, Advertising, Basketball	ipad, payments play, game

Table 3.2: Performance comparison between different methods using MRR and Precision metrics

Algorithm	MRR	P@1	P@5	P@10
`EntRank`	0.08	0.04	0.0418	0.032
`Trendiness`	0.006	0.0	0.0017	0.001
`Content`	0.48	0.40	0.29	0.23
`UserNet`	0.51	0.31	0.34	0.21
`listRec`*	0.36	0.32	0.33	0.21
`listRec`	**0.54**	**0.44**	**0.39**	**0.35**

- `Content`: The content based weighting scheme ranks the lists purely based on the topical interest of the Twitter users. We set $\beta = \gamma = 0$ for this scheme.
- `UserNet`: This scheme purely based on user-network. We set $\alpha = \gamma = 0$ for this scheme.
- `listRec`*: Instead of using dDTM to measure the user interest, we use the LDA by ignoring the temporal variation of topical interest.

Table 4.4 shows the results of MRR and precision, while Table 4.5 reports the results of success at k and DCG. We clearly see that the proposed `listRec` is the best performing model for all measures. The `Content` based scheme closely follows our model, this clearly emphasizes the fact that topical homophily is one of the important features. The `UserNet` performs reasonably well when compared to `listRec` and `Content`. This shows that the followee network of a *twitterer* plays an important role in determining his list subscription. In other words, the probability of a user subscribing to a list increases if the list has already been subscribed by his followees. This

Table 3.3: Performance comparison between different methods using Success at k and DCG metrics

Algorithm	S@5	S@10	S@30	DCG
EntRank	0.14	0.19	0.198	0.27
Trendiness	0.024	0.027	0.0307	0.15
Content	0.43	0.471	0.52	17.54
UserNet	0.427	0.432	0.481	18.2
listRec*	0.324	0.342	0.348	16.49
listRec	**0.45**	**0.493**	**0.54**	**18.42**

clearly shows the impact of the social circle on users' interest. The poor performance of EntRank indicates that the entities mentioned in the users tweet need not be the members of a list.

Finally, it is important to note that the listRec* performs poorly when compared to our listRec model. As mentioned before, the topic component of listRec* ignores the temporal variation of user's interest while generating the topics. From the results, it is quite conclusive that the poor performance of EntRank is due to absence of this temporal variation. We provide further insights on the performance of listRec* by comparing the topics generated by dDTM and LDA over users' tweets in Table 3.1. The topics generated by dDTM are split into two columns denoting the past, and the current interests of the *twitterer*. We can clearly see that there is a significant shift between the *twitterer's* past and current interests. For example, user A's past interest was related to topics like fastfood and meal, while his current interest is more towards technology related topics like Google, hackathon etc. Similarly, user C's past interest was mostly centered around games, while his current interest is inclined towards space related topics like NASA, mars, etc. The last column in Table 3.1 shows the topical interest of users generated using LDA. We can see that the topics are a mixture of the users' past and current interests, with a majority of topics emerging from user's past time frame. On the other hand, the topics from the users' list subscription have a greater match with their current interests rather than their past. This is the most important reason for the superior performance of listRec over listRec*. Figure 3.5 shows the DCG measure for the top 20 ranks. We can see that the listRec is able to suggest more related lists when compared to all other performance measures.

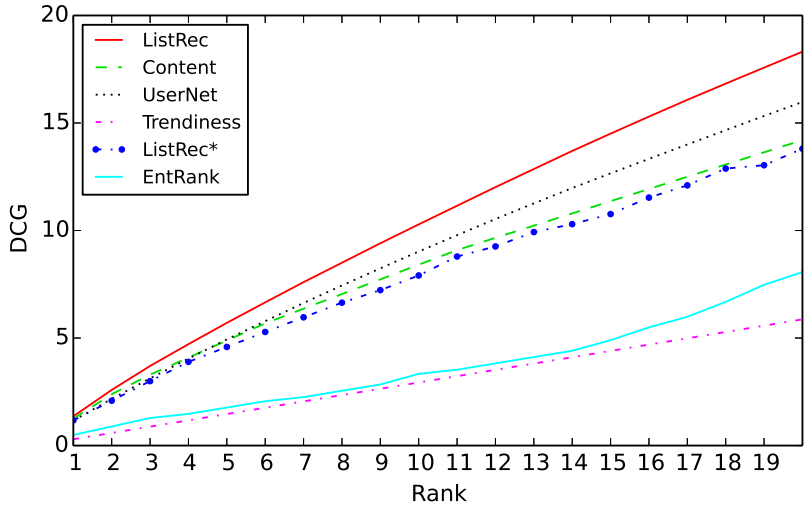


Figure 3.5: Average discounted cumulative gain of related lists for top 20 ranks for different algorithms

3.5.3 Empirical Evaluation

In this section, we compare the quality of lists that are subscribed by the users with the lists that are suggested by the LIST-PAGERANK. To run our experiments, we first create the list-network LN that was described in Section 3.4. Using the set of user-subscribed lists L as the seed set, we do the snow ball sampling using the following steps: (i) For every list $l \epsilon L$, we create a user-set by collecting all the members and the subscribers from l, and (ii) for every user in this user-set, we retrieve their list subscriptions. We iteratively perform these steps to create auxiliary set of lists L_c, $|L_c| = 10876$. The adjacency matrix LN is constructed using the set L_c.

Table 3.4 shows the comparison between the top 5 subscribed and recommended lists. We see that the recommended lists have extremely popular entities as members or subscribers. For instance, the list on *Technology News* have users like *stevenbjohnson* who is a popular media theorist; *timoreilly* is the CEO of O'Reilley Media. The list based on TV topics like #Glee, #NFL etc. have very famous actors like *davidschneider*, and movie rating website *RottenTomatoes* associated with them. Similarly, the lists on *Funny tweets* have very prominent entities associated with them and the lists on *Politics* have famous political news channels such as *foxnewspolitics* and *ABCPolitics*. The list related to aviation has *TheDEWLine* who is a prominent aerospace journalist and blogger

and *flightglobal* which is a news source for global aviation. These entities are not only famous in real-world, but are also active users in Twitter domain. All these users have a large number of followers and retweeters. Additionally, the recommended lists have a large number of followers and retweets when compared to the subscribed lists. We can clearly see that the presence of prominent Twitter personalities acts as a magnet for attracting more subscribers and retweeters for a list.

Characteristics of the recommended list So far, we showed the performance of our LIST-PAGERANK model using qualitative comparisons between the subscribed and the recommended lists. We now show the characteristics of these recommended list by choosing samples from user-subscribed lists L using various criteria. To achieve this, we use the quality measure proposed by Weng et al. [60] for their TwitterRank model. However, unlike the authors, we don't use their method to evaluate our model; instead, we simply use it to measure the characteristics of the recommended list and compare it with the classical PageRank algorithm and the indegree measure. This is mainly because unlike the TwitterRank, the user is not a part of the list network. The sampling procedure for measuring the list characteristics is shown in Algorithm 2.

Algorithm 2: Sampling procedure for analyzing the list characteristics

1 **Require:** *The user-subscribed set of lists* L
2 Choose a sub-set $|P|$ from the set L using different list-based features
3 **for** *each list* $l \epsilon P$ **do**
4 Crawl a set of 10 auxiliary lists, denote this set as Z
5 Create a new list-network with the set Z
6 Run the LIST-PAGERANK algorithm to rank the lists in this new network Z
7 Using equation (3.14) report the characteristics of the ranked lists
8 **end**

For all our experiments, we sample 20 lists from the user-subscribed list L. Therefore, we set $|P|$ as 20 in step 1 of Algorithm 2. The selection of the sample P is based on four different list-based features as described below:

Influence score of list members: Our first selection criteria is based on the influence score of the list members. We wanted to see whether there is any correlation between the authority of the lists that are calculated using individual authorities of the list member, and the authority of the list

Table 3.4: Comparison of quality between subscribed and recommended lists for top topics

list Number	list Description	list Topic	Subscribed list			Recommended list		
			S-Cnt	Rt-Cnt	Entities	S-Cnt	Rt-Cnt	Entities
1	Technology News	#Apple, Technology	37	14	@magpienikki, @Shadowrayven	14092	104	@stevenbjohnson, @timoreilly
2	TV	#Glee, #NFL	4	2	@TVGuide, @TheEllenShow	1875	42	@RottenTomatoes, @davidschneider
3	Funny Tweets	Fun, Comedy	4	17	@FakeAPStylebook, @badbanana	8802	69	@OMGFacts, @GuyCodes
4	Politics	#Obama, Law	113	12	@ezraklien, @SFist	1145	114	@ABCPolitics, @foxnewspolitics
5	Aviation	InFligtCalls, #privatejets	32	15	@DBaviation, @bizjetkev	256	31	@flightglobal, @TheDEWLine

calculated by our LIST-PAGERANK model. To measure the influence scores of the list members, we use the popular *klout score* [1] service that provides the influence score of twitter users using various inter and intra-domain based measures. For every list in L, we retrieve the members and calculate their klout scores. The klout score for a list is then calculated as the collective score of the individual members. To select the sample set P we rank the lists according to their klout scores and choose a set of lists P_{kh} (high klout score) from the 90th percentile and P_{kl} (low klout score) from the 10th percentile of the klout score counts, $P = P_{kh} \cup P_{kl}$.

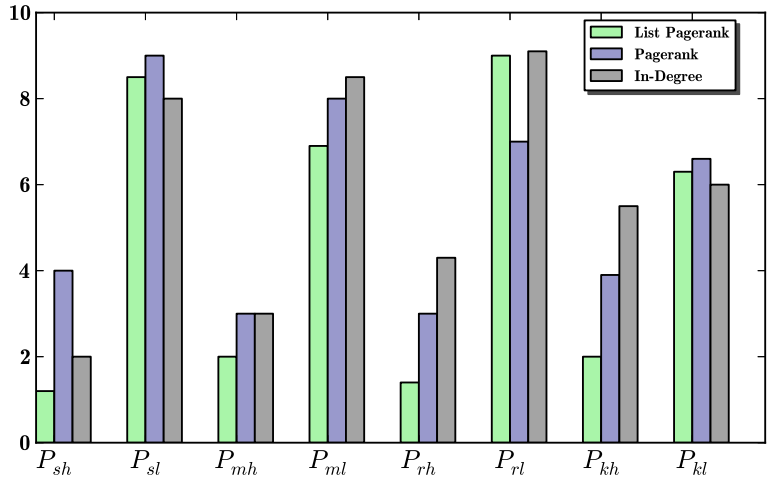


Figure 3.6: Characteristic score of ranked lists for different features

[1] http://klout.com/

Subscriber count of the list: In this criterion P is chosen based on the number of subscribers of the list. We calculate the subscriber count of each list in L, and rank them according to this count. We now choose the lists P_{sh} (high subscription count) from the 90th percentile, and the lists P_{sl} (low subscription) from the 10th percentile of the subscription count, $P = P_{sh} \cup P_{sl}$.

Retweet count of the list: Similar to the subscription count criterion, we rank the lists in L based on their retweet counts. We then choose lists P_{rh}, and P_{rl} from the 90th and 10th percentile of the retweet counts respectively.

Membership count of the list: The final criterion is based on the membership counts of the list. We rank the lists in L based on their membership counts. We then choose lists P_{mh} (high membership), and P_{ml} (low membership) from the 90th and 10th percentile of the membership counts respectively.

The characteristic score of the recommended list is measured using the following equation:

$$C(Z) = \{z_i | z_i \epsilon Z \text{ and } R(z_i) < R(z_p)\} \tag{3.14}$$

Where, z_p is the set of lists in Z which are directly subscribed by the users, and $R(z_i)$ denotes the rank of the list i. According to the equation (3.14), $C(Z)$ measures the number of auxiliary lists that have a higher recommendation score than the subscribed lists. A high score of $C(Z)$ implies that a major part of the recommended list is from the auxiliary list, while a low score implies most recommendation are from the user's direct list subscription.

We show the results of running our LIST-PAGERANK over the sample network Z in Figure 3.6. The x-axis denotes the different characteristic features that were used to choose the user-subscribed list P, and the y-axis denotes the characteristic score obtained using the equation (3.14). From this figure, we can infer three important characteristics of the recommended lists. When we choose the set P with high subscription count (P_{sh}), the lists recommended by our model is mostly from the subscribed set of lists rather than the auxiliary (crawled) set of lists. This trend is similar in both the PageRank and In-degree algorithms; nonetheless, the PageRank and In-degree tend to recommend more auxiliary lists when compared to our model. If the user-subscribed set P is chosen based on

the low subscription count criteria (P_{sl}), we can see that all three algorithms perform equally by recommending lists from the auxiliary set. We also see that the member count plays an important role in deciding the list authority. If P is chosen such that the lists contain a large member count, then most recommended lists are from the users' direct subscription. On contrary, if P is chosen with low membership count all the algorithms tend to recommend the auxiliary lists. It is important to note that both in-degree and PageRank closely follows our model.

Finally, when we choose P based on high klout score (P_{kh}), the LIST-PAGERANK model recommends a majority of lists from users' direct subscription; thus, indicating that the individual authority of list members collectively contribute to the total list authority. Similar to the subscription characteristic, both the in-degree and PageRank tend to recommend the direct list subscription rather than the auxiliary list. The in-degree however seems to recommend more auxiliary lists when compared to the other two. In case of low klout score (P_{kl}), all three algorithms perform in similar fashion.

3.6 Summary

As more and more users join social networking platforms like Twitter, facebook, foursquare etc., the data will be generated at an overwhelming pace, resulting in the problem of information overload. To overcome this problem, social networking sites have introduced the concept of *lists* that help users organize related information into a single bin. Despite being a powerful tool to organize related users and topics, it requires the laborious task of manually adding people who post about a similar topic. In this chapter, we outlined two major problems. First, we showed that majority of users have sparse list subscriptions. Second, we showed that most lists have extremely low number of subscribers, which in turn means that they are inferior in their topical content. To overcome the first problem we introduced the `ListRec` model. We formulated this model as a linear combination of content, network and trendiness based weighting schemes, and estimated the parameters using a cyclic ridge regression algorithm. Our results showed that the `ListRec` model outperformed other base line models in all the performance measures. Furthermore, we showed the importance of using temporal topic model by leveraging our rich repository of temporally

distributed streaming tweets. The results clearly showed that the user of dynamic temporal topic model (dDTM) over the conventional topic model (LDA) resulted in a superior recommendation.

To handle the second problem, we introduced a LIST-PAGERANK model that recommends auxiliary lists that are significantly better than the existing lists that are directly subscribed by the *twitterers*. To design this model, we introduced a new subscriber-member based relationship for the edges in the list network. Using empirical evaluation techniques, we showed that our model is efficient in recommending lists that contain members who are topically authoritative. We also showed that the recommended set of lists have high retweet and subscriber counts; thus, indicating it's topical dominance.

CHAPTER 4: RECOMMENDATION IN CROWDFUNDING

4.1 Introduction

For several years, entrepreneurs had to seek the help of banks, brokers, and other financial intermediaries to acquire the necessary funds for starting a business venture. Such financial constraints were a huge bottleneck to people with innovative ideas. However, this scenario has changed drastically with the emergence of *crowdfunding* platforms. Thanks to the widespread use of internet, entrepreneurs can effectively post their ideas on crowdfunding websites and gain the attention of people all over the world. The concept of crowdfunding is analogous to micro-financing or crowdsourcing [68], where the seed capital is collected by soliciting funds from a large group of people, rather than a single individual (venture capitalist).

Crowdfunding can be characterized into four different types: equity-based, lending-based, reward-based, and donation-based. In equity-based crowdfunding, the investors receive some form of stake from the company. The donation-based is similar to a charitable venture, while in lending-based, the investors are repaid for their investment. Finally, the most popular form of crowdfunding is the reward-based, where users receive some form of gift in return of their investment. Kickstarter, one of the popular crowdfunding sites, mainly adopts this reward-based crowdfunding mechanism while raising over 480 million dollars in pledged amount and 19,911 successfully funded projects in 2013. This domain follows the "all or nothing" policy, which means that the pledged money is collected only if the goal amount is reached; if not, the entire money is returned back to the investors. Kickstarter terms the investors as *backers*, and the founders of projects as *creators*. The creators post their ideas by providing a detailed description of their project which includes the scope of the project, video description, reward details, topical categories, location, updates, FAQs, etc. The backers then invest in the project based on its quality and their personal interests.

Despite being a valuable platform for crowdfunding ventures, statistics show that only 43% of the projects succeed in reaching their pledged goal [69]. Additionally, the margin by which successful projects exceed their pledged goal is extremely narrow [70]. Being a relatively new domain, very few studies have explored the crowdfunding domain from a data mining perspective [71–73]. Although innovative in their approach, these studies restrict themselves to the standard

set of features that are available readily from the Kickstarter domain. Therefore, our research on crowdfunding [2] explores the popular Kickstarter domain, by leveraging diverse information sources to construct a set of features that can play a significant role in determining the success of Kickstarter projects. In the first part of this chapter, we perform a comprehensive study on the set of features and their effect on Kickstarter projects. In the second part, we propose a supervised learning approach that effectively utilizes these features to tackle this unique recommendation problem. We formulate our recommendation problem as a binary classification/regression problem, where given a backer-project pair, the trained model computes the score that represents the likelihood of funding. Utilizing the proposed approaches together with a gradient boosting tree, a state-of-the-art learner model, we achieve a practically useful level of performance up to 0.89 AUC (area under the curve) value. Additionally, we perform an in-depth evaluation of our model using over 795K backer-project relations and a wide variety of other data sources like backer profiles, tweets and profile information of twitter users. Our analysis reveals various interesting knowledge about the behaviors of Kickstarter users with respect to their backing frequency, social network, geo-location, and other personality-based traits. The major contributions of this paper are summarized as follows:

1. We perform an exhaustive study of the crowdfunding domain from the project, backer, social network, and geo-location perspectives to provide several unique insights on inter- and intra-domain factors that affect the success of Kickstarter projects.
2. Our analysis is based on diverse data sources such as: (1) content information of projects, (2) profile information of backer and creators and (3) heterogeneous information from the Twitter network.
3. We build a robust predictive model for recommending backers in crowdfunding domain that achieves an AUC of 0.89, and a precision up to 0.8.

The rest of this chapter is organized as follows. We begin by presenting the related work on this topic in Section 4.2. We then explain the characteristics of the Kickstarter domain in Section

4.3. Section 4.4 describes the data collection methodology. Section 4.5 analyzes the features of Kickstarter. Finally, the conclusions obtained through this study are presented in Section 4.7.

4.2 Related Work

Crowdfunding and Kickstarter: Since crowdfunding is still an emerging platform, most studies on this domain are relatively new. One of the most comprehensive studies on Kickstarter can be seen in [70] and [74]. In [70], the authors examine the dynamics of kickstarter domain, and [74] explains various types of crowdfunding platforms. The authors of [75] and [76] perform a real-world analysis on crowdfunding platforms. Their study is based on a real-time survey that aims to learn the motivation behind users who create and invest crowdfunding projects. In [77], the authors use natural language processing techniques to analyze the textual content of Kickstarter projects, while [72] leverages the updates of projects to determine their success rate. There are very few papers that study the role of *twitter* for Kickstarter projects. In a recent study, Lu *et al.* [78] delineate the impact of social network on Kickstarter projects.

Studies on other Crowdfunding platforms: Apart from Kickstarter, there are many other crowdfunding platforms. In our previous work, we analyzed the micro-financial activities in *Kiva.org* [79, 80]. Few research works [81], [82] and [83] explored the effects of the internet on micro-financing, and peer-to-peer lending transactions. The paper closest to our research is characterized by a similar goal as that of ours [71]. In their paper, the authors adopt a hypothesis-driven approach to analyze features from Kickstarter. Despite a novel approach, their analysis is based on very basic set of features such as number of updates, comments, facebook friends, etc.; such features are readily available from the Kickstarter platform. To the best of our knowledge our work is the first to perform an extensive analysis of the Kickstarter domain by utilizing project-, user profile-, geo-location- and social network-based attributes.

4.3 Characteristics of Kickstarter Campaign

Before exploring the features of Kickstarter, we investigate the general characteristics of this crowdfunding domain. Figure 4.1(a) shows the overall trend of successful and failed projects. We observe that a majority of projects exceed their goal by a very marginal amount. Additionally,

projects that exceed their target goal by over 150% are extremely few. This suggests that people are not interested in supporting the projects once the project goal amount is received. Figure 4.1(b) shows the success ratio of top-10 categories of Kickstarter projects, where the top three project categories are dominated by film & video, music and games. Furthermore, the success ratio of these categories ranges roughly between 35%-65%, with Theater being the highest (about 65%) and technology being the lowest (about 35%).

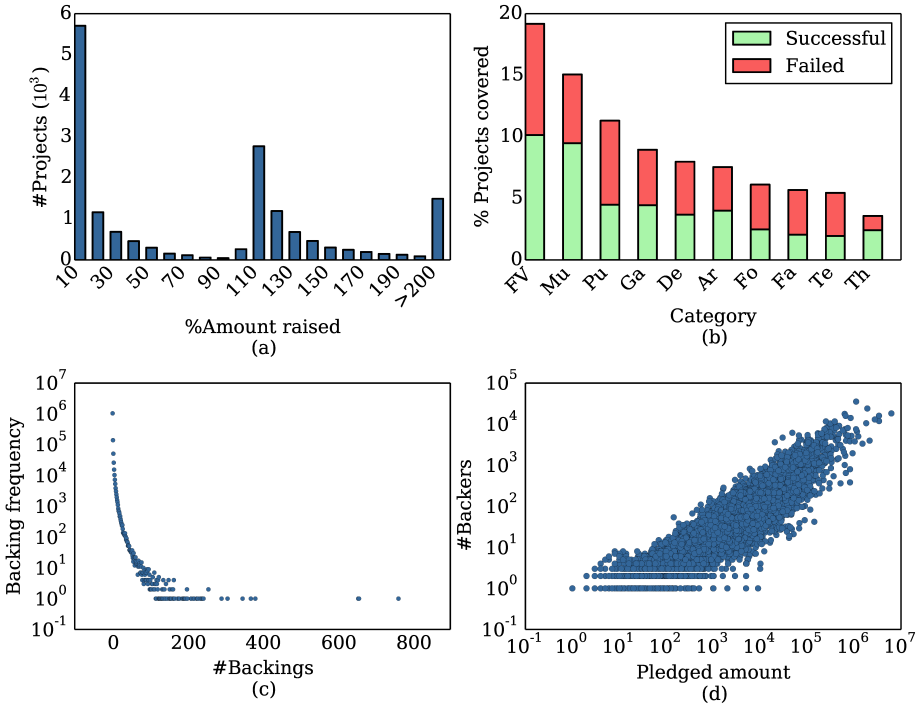


Figure 4.1: (a) the percentage of the total goal amount raised by Kickstarter projects; (b) shows the category-wise success ratio of projects: FV-film & video, Mu-music, Pu-publishing, Ga-games , De-design, Ar-art, Fo-food, Fa-fashion, Te-technology, and Th-theater. (c) Backing frequency of Kickstarter users and (d) shows the relationship between pledged amount and the number of backers.

The relationship between backers and the pledged amount is an essential component of Kickstarter. The backing pattern of Kickstarter users follows a power-law distribution, as depicted in Figure 4.1(c). We see that a large number of people tend to back just one or two projects; people who back more than 100 projects are extremely few. On the other hand, Figure 4.1(d) shows a strong correlation between the number of backers and the pledged amount. An earlier study on

Table 4.1: Kickstarter data statistics for 18,143 projects collected from Dec 2013 - Jun 2014.

Attribute	Mean	Min	Max	StdDev
Goal Amt	26,531.2	100	100,000,000	758,366.5
Pledged Amt	11,023.6	100	6,224,955	78,550.8
backers count	138	1	35,383	633.7
Duration(days)	31	1	60	10.05

Kickstarter reported a correlation of 0.9 [78]. However, we observe a lower but still strong correlation of 0.68. More importantly, we found that there exists many projects with a large pledged amount which had low number of backers. It implies that the kickstarter domain has gained more trust over the years, and users are ready to back a large amount of money in kickstarter projects. Nevertheless, in this paper we assume that backers impact the success of a project, and our goal is to investigate the effect of various features that impact the backing count.

4.4 Dataset Description

Kickstarter Database: For our experiments, we obtained six months of Kickstarter data from *kickspy*.[1] Our dataset spans from 12/15/13 to 06/15/14, which consists of 27,270 projects characterized by 30 project-based attributes. These attributes include a number of static features such as: project goal, duration, textual content, etc., and two dynamic features like: per-day increase in number of backers and pledged amount. To prepare our dataset, we removed projects that were canceled, suspended as well as those with less than one backer and $100 as pledged amount. In this manner, we obtained 18,143 projects and over 1 million backers. We denote our projects database as $\mathcal{K}$ and backers database as $\mathcal{B}$. The statistics of our database is given in Table 3.1.

Twitter Database: To build our tweet repository, we used the query API of *Topsy* [2] to search the titles of all projects in our Kickstarter database. By expanding the short URLs, we eliminated tweets that did not map to our database $\mathcal{K}$. Using this method, we obtained 106,738 unique tweets, which covered 55% of our projects. The remaining 45% were never promoted using Twitter. In

[1] www.kickspy.com
[2] www.topsy.com

addition to this, we also retrieved the complete profile information of the promoters who tweeted these tweets; we denote this database by $\mathcal{S}$.

4.5 Analyzing Kickstarter Traits

The success of an entrepreneurial venture is heavily dependent on its quality from the project contents. Therefore, in this section, we explore the set of factors that initiate a user to invest in Kickstarter ventures. We begin by categorizing the features into three main groups: (a) project-based traits, (b) personality-based traits and (c) network-based traits. The project-based traits are purely with respect to features derived from the qualities of Kickstarter projects. The personality-based trait is further divided into creator personality and backer personality. It represents the characteristics of creators who host the Kickstarter project, and the backers who invest in these projects. Lastly, the network-based traits are derived from the social media (Twitter) domain. In the following sections, we explain these features in detail.

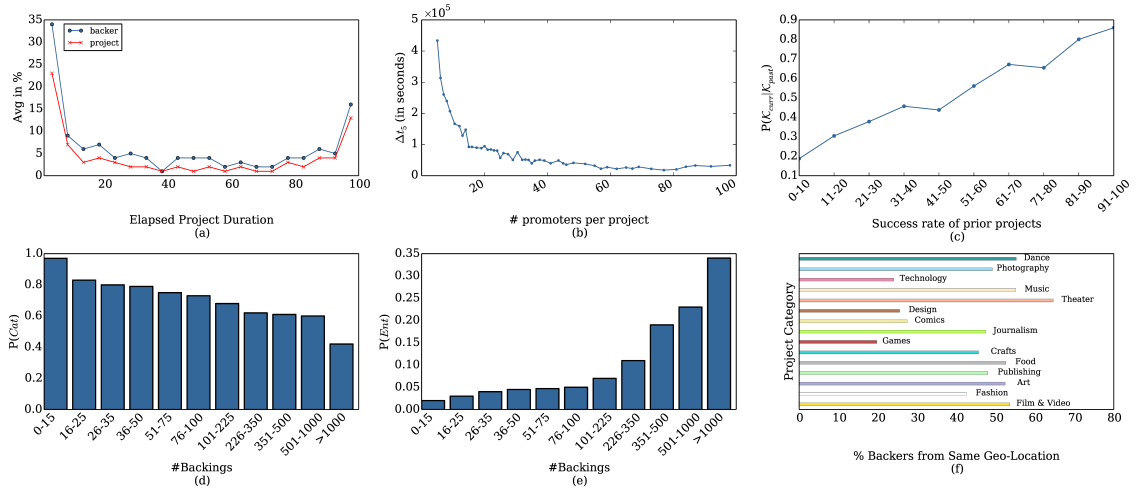


Figure 4.2: Analysis of features from Kickstarter domain. (a) Temporal progression of funds and backers in Kickstarter; (b) Adoption rate of project promotions in Twitter; (c) effect of *creator's* prior success rate on the success of his current kickstarter project; (d) topical preference of users towards projects; (e) trust relationship between the backers and creators, and (f) effect of geo-location on Kickstarter projects.

4.5.1 Project based Traits

Static features: We use 16 different features, which include generic features such as the duration of project, goal amount, number of facebook shares, main topical category, sub-categories, number

of backers, updates about the progress of project, pledged amount, comments, location, currency and rewards. The content based features include total number of words in the project description, risks and challenges, FAQs, number of images, and presence of videos.

Temporal features: One of the most interesting aspects of Kickstarter campaign is the U-shaped distribution of fund progression over time. As shown in Figure 4.2(a), a large percentage of the pledged goal is accumulated in the first few days of the project duration. The progression then tapers out during the mid-phase, which can be considered as a dormant period in the project funding cycle. However, unlike the progression of news stories, the funding activity does not decay monotonically towards the end; instead, we suddenly see an increase in the pledged amount during the final phase of the project funding cycle (few days before the project end date). A popular term for this phenomenon is the *deadline effect* [84]. It is also important to note that the accumulation of backers follows closely with the pledged amount, where a majority of backing activity happens during the first and last weeks of the funding cycle. This classic behavior of Kickstarter data has also been shown some in recent studies [78, 85]. Another important temporal dynamics is related with the spread of Twitter promotions over time. If the first few tweets about a kickstarter project are tweeted within a short time frame, the number of Twitter users who adopt and promote these tweets are much higher. In social science, this phenomenon is widely known as the *Herding instinct* [86]. This effect is depicted in Figure 4.2(b), where Δt_5 denotes the average time delay between the first 5 consecutive tweets. From this figure, we can conclude that *early promotions are crucial to a project's success*.

4.5.2 Personal Traits

Backer personality: To begin with, we retrieve the backing history of all the users in $\mathcal{B}$, and obtain the list of categories and creators for every project in their backing history. The history of categories and creators are denoted by the set $\mathcal{H}(C)$ and $\mathcal{H}(E)$ respectively. The personality of backers are analyzed using these two sets.

Topical preference: Topical preference plays an important role in determining the interests of users [87]. In our setting, we define this as the tendency of users to continuously back projects from

the same topical category. We examine this by calculating the conditional probability of a user u to back a category c, given c is present in the backing history of this user. We represent this probability by $P_u(Cat) = P(c|c{\in}\mathcal{H}_u(C))$, where $\mathcal{H}_u(C)$ indicates the set of categories from user u's backing history. $P(c|c{\in}\mathcal{H}_u(C))$ is calculated by Bayes' theorem as follows:

$$P(c|c{\in}\mathcal{H}_u(C)) = \frac{P(c{\in}\mathcal{H}_u(C)|c)P(c)}{P(c{\in}\mathcal{H}_u(C))} \tag{4.1}$$

Figure 4.2(d) shows the outcome of this experiment. It can be seen that irrespective of the number of backings, the probability of users backing the same category ($P(Cat)$) is very high. Although this probability decreases with the increase in backing count, this reduction is significant only for users with very large backings (i.e. over 1000). This signifies that *backers have strong topical preference over Kickstarter projects*.

Mutual trust: It is shown that the investors do not just randomly choose projects for backing; instead, they look for a long-term connection to the creator [75]. We call this attribute as the *mutual trust*. To validate this claim, we calculate the conditional probability of a user u to back a creator e, given that e is in the backing history of this user. This probability is represented by $P_u(Ent) = P(c|c{\in}\mathcal{H}_u(E))$, where $H_u(E)$ indicates the set of creators from user u's backing history and $P(c|c{\in}\mathcal{H}_u(E))$ is calculated similar to Equation (5.12). Figure 4.2(e) shows the result of this analysis; here, we notice an increase in probability $P(Ent)$ as the backing count increases. In other words, when users start backing more and more projects in Kickstarter, they *tend to develop an inclination towards creators whom they have backed in the past*. This inclination leads to a stronger relationship with the creator thereby creating a mutual trust.

Creator personality: The personality of project creators is measured using three features: a) number of projects hosted by the creator, b) number of projects backed, and c) the expertise of the creator. The first two features are obtained from the profile information, while the third feature is analyzed as follows:

Creator expertise: We say that a creator having a high success ratio in his past projects is more likely to succeed in his current project. We evaluate this notion by calculating the probability

$p(k_{curr}|k_{past})$, where k_{curr} denotes success of current project, and k_{past} denotes the success ratio of prior projects. From Figure 4.2(e) we can distinctly see that this probability increases with the increase in the creator's success ratio. Hence, *experienced creators have a greater chance to succeed in the crowdfunding domain*.

4.5.3 Location based Traits

In this section, we try to understand the role of geo-location on Kickstarter projects. For every project in our database $\mathcal{K}$, we calculate the percentage of backers whose geo-location matches with the project's location. The result of this study is depicted in 4.2(f), which clearly shows that geo-location does impact the success of projects. Nonetheless, it is interesting to note that *the impact of geo-location is not uniform for all the categories of projects*; for instance, projects on games, comics and technology are not very much dependent on their geo-location, while projects on theater, food, and dance are highly dependent. A logical explanation for this trend can be attributed to the rewards that are provided by the projects. For example, the rewards offered by theatrical projects mostly include items such as movie tickets, tickets to the premier shows or personal interaction with the cast members. Such rewards are extremely dependent on the proximity to the project's geo-location since people from distant geo-locations might not travel to see the performances. Contrary to this, rewards offered by technical projects can be sent to people all over the world through mail.

4.5.4 Network based Traits

One of the main reasons for a project's failure is the lack of publicity [76]. Therefore, before we examine the social network features, we will see whether Twitter based promotions impact the success of Kickstarter projects. Table 4.2 shows that projects with promotions have 63% chance to succeed in their funding goal, while those without promotions have a mediocre success rate of 34%. This definitively proves that tweets play a dominant role in determining the success of projects. Hence, we divide our analysis into two parts: first, we examine the impact of various network measures over the success of Kickstarter projects; second, we build communities of Kick-

starter users from Twitter to examine the effect of these communities over the backing habits of individuals.

Table 4.2: Success rate of projects with promotional activities. w/o-promo: without-promotional activity; w-promo: with promotional activity.

# Projects w/o-promo	Success w/o-promo	# Projects w-promo	Success w-promo
8207	34%	9935	63%

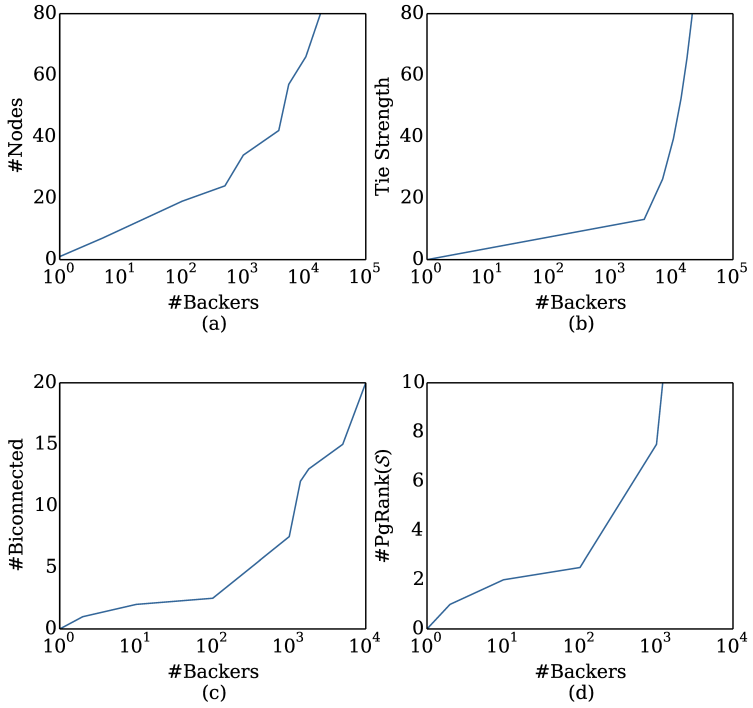


Figure 4.3: Impact of Twitter network on the backers of Kickstarter projects. (a) and (b) show the socio-centric analysis; (c) and (d) shows the ego-centric analysis.

Impact of Twitter network on Kickstarter projects: For this analysis, we construct a network using the Twitter database $\mathcal{S}$, which contains the set of users who tweeted about Kickstarter projects. Each user $s \in \mathcal{S}$ is a node, and a directed link exists between node A and node B based on the following conditions: 1) if A is a follower of B; 2) if A mentions B in his tweet. For the first case, we assign a link weight of 1, and for the second case, the link weight depends on the number of times A has mentioned B in his past tweets. By constructing this graph, we present the socio-

centric and the egocentric attributes of this network. Figure 4.3(a) shows that the number of backers increases with the number of nodes (i.e. promoters in Twitter). However, the accumulation of backers is not only based on the number of promoters, it also depends on the connectivity between these promoters. This notion is conveyed in Figures 4.3(b) and 4.3(c), where 4.3(b) shows that *stronger tie strength between the promoters results in greater accumulation of backers*, and 4.3(c) captures the same notion in terms of biconnectivity between the promoters. Lastly, Figure 4.3 (d) shows that *projects promoted by influential twitter users have the potential to attract many backers*; where the influence is determined by calculating the PageRank scores of the promoters.

Community influence on investors' backing habits: Many studies have shown that communities from social network play an important role in influencing the actions of individuals [88, 89]. Applying the same analogy, we say that *the backing habits of investors in Kickstarter are influenced by their social circle (or community)*. To validate this statement, we begin by explaining the procedure for creating Kickstarter communities from Twitter. Later, we show the method that was used to calculate the influence score between these communities and the individuals who back the projects.

To construct the communities, we use the promoters (i.e. twitter users) in $\mathcal{S}$ and create a bipartite graph of projects and users where each edge denotes the action of a user $s \in \mathcal{S}$ tweeting about a project $k \in \mathcal{K}$. These tweets can be simply promotions, or it can signify the action of backing. The bipartite graph is then projected into a unipartite graph resulting in a network that consists only of the users s. The edge weight between the users (s_1, s_2) is computed using Jaccard index, which is given by:

$$W = \frac{|\mathcal{K}(s_1) \cap \mathcal{K}(s_2)|}{|\mathcal{K}(s_1) \cup \mathcal{K}(s_2)|} \tag{4.2}$$

where $\mathcal{K}(s_1)$ and $\mathcal{K}(s_2)$ denote the set of projects that are tweeted by users s_1 and s_2. To form the communities from this network structure, we use the modularity metric, which is defined as follows:

$$M = \frac{1}{2W} \sum_{i,j} \left[W_{ij} - \frac{n(i)n(j)}{2W} \right] \delta(C_i, C_j) \tag{4.3}$$

where W_{ij} is the weight of edge between vertices i and j, W is the summation of all the edge weights. $n(i)$ and $n(j)$ are obtained by summing up all the edge weights of nodes i and j respectively. C_i denotes the community that i belongs to and δ is the Kronecker delta. Lastly, we use the Louvain method of community detection [90] over this unipartite network to obtain 160 communities. A snapshot of this procedure is shown in Figure 4.4.

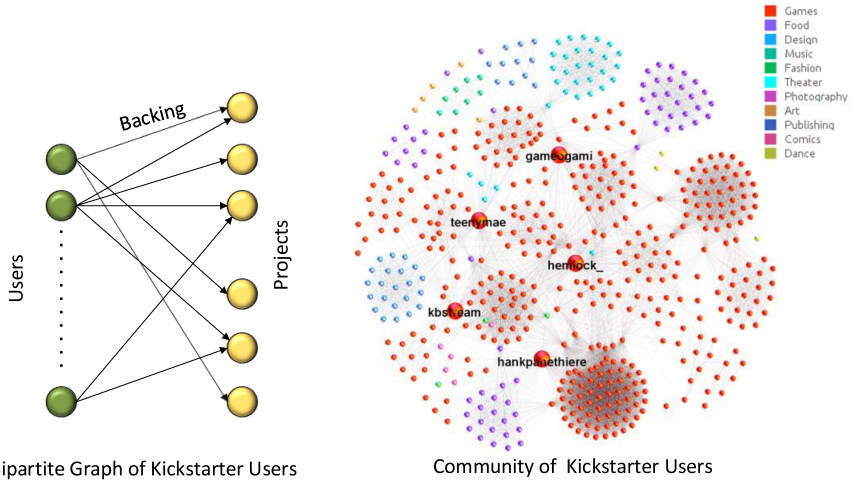


Figure 4.4: A Twitter-based community formed by the backers of Kickstarter (right) by projecting the bipartite graph of user and projects(left). The colors represent different topical categories of communities, and the names denote the top users of the community.

To calculate the influence of these communities over the backing habits of the users, we use our database $\mathcal{B}$, and retrieve a subset of backers who have their Twitter account information embedded in their Kickstarter profiles. We call this set as $\mathcal{B}_{tw}$, where $|\mathcal{B}_{tw}| = 9,266$. To measure the influence of the community $c \in C$ over the user $b_{tw} \in \mathcal{B}_{tw}$, we calculate their *Affinity* score as follows:

$$Affinity(b_{tw}, c) = |F(b_{tw}) \cap F(c)| \tag{4.4}$$

where $F(b_{tw})$ and $F(c)$ denote the set of all followers and followees of b_{tw} and c respectively; $|F(b_{tw}) \cap F(c)|$ indicates the number of mutual friends between this backer and the members of the community. Figure 4.6(a) shows the outcome of this analysis, where $P(b_{tw} = k|c = k)$ indicates the probability of the user $b_{tw} \in \mathcal{B}_{tw}$ backing the project k, given k is backed by the members of the community c. Figure 4.6(b) is simlar to 4.6(a) except, here, $\mathcal{T}$ denotes the project

category. From these figures, it is conclusive that the stronger the *Affinity* of a user towards a community, the greater the chance for this user to back the same project (or project category) that was backed by the community.

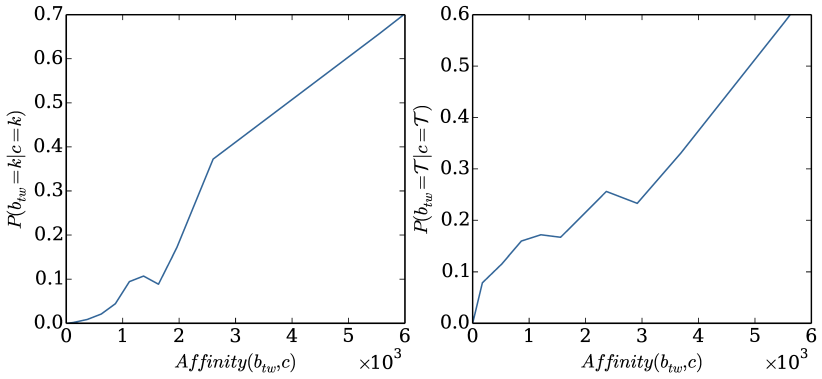


Figure 4.5: Influence of Twitter-based Kickstarter communities over the backing habits of users.

4.6 Recommending Backers

In the previous section, we performed a thorough analysis of the Kickstarter crowdfunding domain, where we explained various features that affect the success of projects. In this section, we propose a content based recommender framework for recommending backers to Kickstarter projects. We formulate our recommendation problem as a binary classification/regression problem, where given a backer-project pair, the trained model computes the score that represents the likelihood of funding. Considering the complexity and heterogeneity of our data and the problem, it is important to use the most suitable and powerful prediction model that are available. To this end, we have employed a gradient boosting tree (GBtree)[3] [91, 92]. A GBtree is an ensemble method where an individual learner is a decision tree [93].

The reason for choosing a GBtree for our problem is as follows: First of all, an ensemble method is known for its superior generalization capability for unseen data. Furthermore, a decision tree, our base learner, uses one variable at each node when it is trained/constructed as well as when it is applied to test data. This characteristic prevents us from worrying about how to properly consider heterogeneity in the features we generated. The drawback of using other learners, such as

[3]The GBtree implementation we used is available at https://sites.google.com/site/carlosbecker/resources/gradient-boosting-boosted-trees

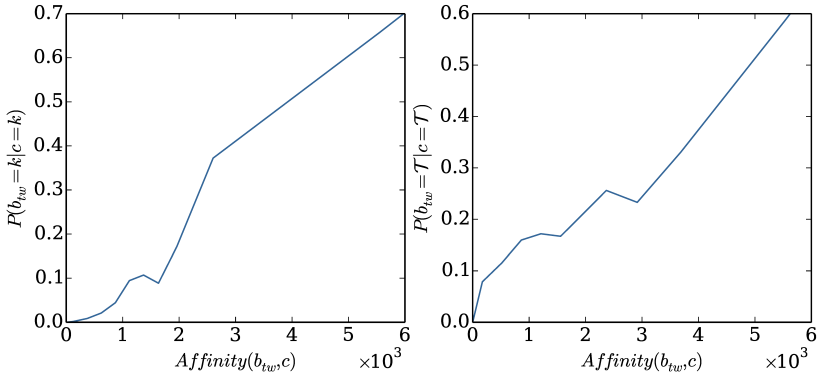


Figure 4.6: Influence of Twitter-based Kickstarter communities over the backing habits of users.

logistic regression and support vector machines, is that heterogeneous features have to be normalized via, say, standardization of their distributions by transforming each feature to have zero mean and unit variance. Such normalization does not always make sense for binary and integer features, and it also removes the nonnegativity of our feature representation that offers intuitive interpretation of them. It should be noted that the key contribution of this work is more about extracting the important features and understanding the domain by providing novel insights, but not necessarily about building a new predictive modeling algorithm.

4.6.1 Experimental Setup

We formulate the task of recommendation as a binary classification/regression problem. That is, every backer-project pair (b, k) is an individual data item, and given such a pair, our task is to predict how likely a user will back a project. Our aim is not only to show the superior prediction performance of our model, but also to conduct an in-depth analysis on the features discussed in the earlier sections. To achieve this, we create a dataset, $\mathcal{D}$ using a subset of backers from $\mathcal{B}$, defined as:

$$\mathcal{D} = \{(b, k)|k \in \mathcal{K}(\mathcal{T}), (b \in \mathcal{B} \cap prof(b) = 1)\}$$

where $\mathcal{K}(\mathcal{T})$ is the set of all the project in $\mathcal{K}$ that are covered by our tweet database $\mathcal{T}$, and $prof(b) = 1$ indicates the existence of the *complete* profile information of the backer b. In total, the cardinality of set $\mathcal{D}$ is 795,347. To train and test our model, *we only consider the features*

available within the first three days of project duration. This setting is much more realistic when compared to previous studies that use the complete set of features that are available only at the end of the project duration. This includes the features from project-, temporal-, personal-, geo-location-, and network-based traits that were discussed in the previous sections. However, *we eliminate comments, updates, and the number of Facebook shares* from the project-based feature since these features are generally not present in the initial stages of a project. It should be noted that $\mathcal{D}$ consists of only the positive samples, which indicates the action of user $b \in \mathcal{B}$ backing a project $k \in K$. Therefore, to create a balanced dataset, we augment $\mathcal{D}$ with $795,347$ randomly selected negative instances. Out of this entire dataset, we test the following cases by filtering out the data instances matching the conditions for each case.

Case 1: *Evaluating the influence of social network.* The influence from social network (Section 4.5.4) is much stronger on backers who have their Twitter profile. Additionally, the community-based influence is applicable only for backers who are connected to the Twitter network. Therefore, to evaluate this feature, we create a dataset $\mathcal{D}_{tw}$, which is defined below:

$$\mathcal{D}_{tw} = \{(b,k)|(b,k) \in \mathcal{D} \cap (b \in \mathcal{B}_{tw})\}$$

where $\mathcal{B}_{tw}$ is the set of backers who have their Twitter profiles.

Case 2: *Evaluating the impact of geo-location.* In Section 4.5.3, we showed that the geo-location does not affect every category to a similar extent. To further support this result, we use the dataset $\mathcal{D}$ and retrieve only those projects which have the following categories: 1. Theater, 2. Music, 3. Games, and 4. Technology. We chose these categories because theater and music strongly depend on their geo-locations, while games and technology have very weak geo-location dependency (Figure 4.2(f)). We term this dataset as $\mathcal{D}_g$.

The datasets $\mathcal{D}$, $\mathcal{D}_{tw}$, and $\mathcal{D}_g$ are used for our evaluation which was performed using the standard 10-fold cross validation strategy.

4.6.2 Performance Evaluation

Using our evaluation methodology, we want to understand how much each feature contributes towards the recommendation performance. To achieve this, we use the set of attributes from Section 4.5 and categorize them into the following groups:

(a) `prj` (13 dimensions): All the features from the project-based traits (Section 4.5.1) except for updates, comments, and the number of facebook shares.

(b) `crt-person` (3 dimensions): Features from creator personality, which includes number of projects created, projects backed by the creator, and success ratio of the creator.

(c) `bck-person` (4 dimensions): Features from the backer personality such as the number of backings, categories of backed projects, topical preference, and creator preference.

(d) `prjsoc` (4 dimensions): Social network features on Kickstarter projects such as the number of promoters, the tie strength, the bi-connected components, and the PageRank of promoters from the first three days of the project duration.

(e) `bcksoc` (1 dimension): The influence score of community over the backers.

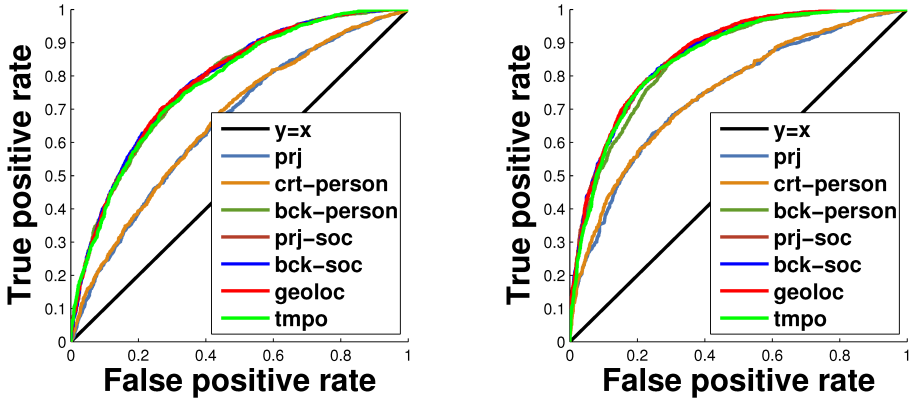
(f) `geoloc` (1 dimension): The influence score of geo-location over projects.

(g) `tmpo` (9 dimensions): The accumulation over the first three days in terms of the number of backers, the funding amount, and the number of tweet promotions.

Therefore, every object in our dataset (i.e., a backer-user pair) is represented by a 35-dimensional vector. In addition to these feature groups, we also split the Kickstarter users based on their backing frequency to study the performance of recommendation depending on various funding experiences. We begin by reporting the overall performance of our model, followed by the analysis of the variable importance for various feature groups. We conclude the evaluation by reporting the ranking performance of the recommendation model.

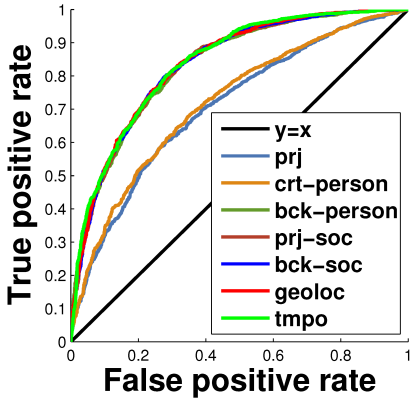
4.6.3 Predictive Performance

Overall performance: We test the performance of our model by gradually incorporating more features described in Section 4.6.2 to the experiments for different backer types. Figure 4.7 shows the performance results as the receiver operating characteristic (ROC) curve, and their AUC values

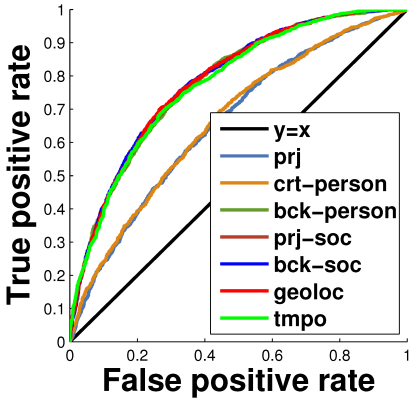


(a) BF = 2-10

(b) BF = 10-100



(c) BF = 101-500



(d) BF = 501-1000

Figure 4.7: The ROC curve results for different backing frequency (BF) values.

are summarized in Table 4.3.[4] One can see that more features generally lead to better performances, and the best AUC values ranges from 0.79 to 0.88 when using all the available features, indicating the efficacy of the features inspired by our in-depth analyses.

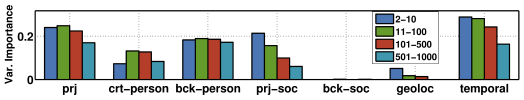
Analysis of feature groups: The analysis on the variable importance of each feature group is shown in Figure 4.8. We highlight the following insights about backing behaviors:

[4]The AUC value is computed using the trapezoidal approximation [94].

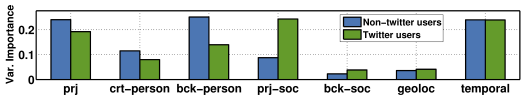
1. *The temporal progression of funds, backers, and tweet promotions have the strongest variable importance.* This claim is supported by high variable importance of the temporal features for all the different types of backers, as shown in Figure 4.8(a).
2. *Backers strongly depend on their personal preferences to fund a project*. This variable, which is denoted by `bck-person` includes the topical preference, and the mutual trust that were discussed in Section 4.5.2. In Figure 4.8(a), the inclusion of this feature has a significant effect over all the backer types.
3. *The impact of social network monotonically decreases with the increase in backing frequency.* This effect is shown by the `prjsoc` feature in Figure 4.8(a). From this trend, we infer that experienced investors do not solely rely on social network-based promotions, but instead they probably consider various other aspects of the projects for their backing decisions. Contrary to this, inexperienced investors are easily attracted to fund projects which have large promotional activity.
4. *Social network has stronger influence over backers who have their Twitter profile.* From Figure 4.8(b), we can see that the variable importance of `prjsoc` is distinctly higher for Twitter users (i.e. backers with Twitter profile) when compared to the non-Twitter users. This is because the non-Twitter users are not exposed to the activities in social media and therefore they seldom notice the tweets about Kickstarter projects. We also see that such users rely more on project and personal features. This trend is similar for community-based influence `bcksoc`. Nonetheless, it should be noted that, the very low variable importance of this feature is due to the fact that users with this feature are extremely fewer in number.
5. *The influence of geo-location strongly depends on the topical category of the project.* Figure 4.8(c) confirms our analysis in Section 4.5.3 where projects belonging to theater and music categories have a greater dependency on the `geoloc` feature when compared to games and technology.

Table 4.3: Cumulative AUC values obtained in the plots shown in Fig. 4.7.

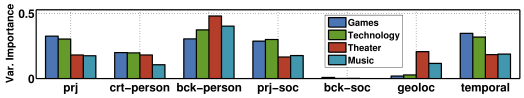
	Backing frequency			
Feature	2-10	11-100	101-500	501-1000
`prj`	0.736	0.744	0.707	0.659
`+crt-person`	0.744	0.748	0.719	0.664
`+bck-person`	0.882	0.849	0.830	0.780
`+prjsoc`	0.883	0.862	0.832	0.780
`+bcksoc`	0.882	0.862	0.834	0.784
`+geoloc`	0.886	0.864	0.836	0.783
`+tmpo`	**0.886**	**0.871**	**0.838**	**0.792**



(a) Dataset $\mathcal{D}$



(b) Dataset $\mathcal{D}_{tw}$ (**Case 1**)



(c) Dataset $\mathcal{D}_g$ (**Case 2**)

Figure 4.8: Variable importance of various Kickstarter features. (a)-(c) shows the AUC value improvements over 0.5 when using only a particular feature set.

4.6.4 Ranking the Backers

Although our experimental setting is a binary classification, the desired capability from learning the function $f(b, k)$ by a GBtree is to compute the likelihood of funding, which allows us to rank the most appropriate backer for a particular project. Therefore, to evaluate the performance of ranking, we use the standard information retrieval measures. For every project, we compute: 1) P@k: The *precision at rank k* for our task is defined as the fraction of rankings in which the true backers are ranked in the *top-k* positions, 2) MRR: The *mean reciprocal rank* is the inverse of the position of the first true backer in the ranked set of backers produced by our model, 3) S@k: The success at rank k is the probability of finding at least one true backer in the *top-k* ranked set, and 4) DCG: The discounted cumulative gain [67] is based on the simple idea that highly relevant backers are more important than marginally relevant ones.

Table 4.4: Performance comparison between different sets of features using MRR and Precision metrics.

Features	MRR	P@1	P@10	P@20
prj	0.5	0.314	0.322	0.321
+crt-person	0.505	0.324	0.329	0.323
+bck-person	0.816	0.707	0.684	0.623
+prjsoc	0.828	0.728	0.688	0.626
+bcksoc	0.834	0.722	0.71	0.62
+geoloc	0.89	0.818	0.706	0.618
+tmpo	**0.892**	**0.824**	**0.708**	**0.627**

Table 4.5: Performance comparison between different sets of features using Success at k and DCG metrics.

Features	S@1	S@10	DCG
prj	0.314	0.902	8.24
+crt-person	0.324	0.894	8.371
+bck-person	0.707	0.996	17.297
+prjsoc	0.728	0.998	17.424
+bcksoc	0.71	0.998	17.209
+geoloc	0.818	0.998	18.232
+tmpo	**0.824**	**0.998**	**18.388**

Table 4.4 shows the results of MRR and precision, while Table 4.5 reports the results of success at k and DCG. We clearly see that the addition of features results in a performance boost for all the measures. There is a clear increase in precision and MRR after the addition of backer personal traits, and this increase is further boosted with the addition of social network, geo-location, and temporal features. This trend is similar for all the other performance measures. It should be noted that, unlike the previous research [71], our recommendation is purely based on the features from the first three days of the project duration. Despite this fact, we can achieve a high precision value of 0.82.

4.7 Summary

In this chapter, we performed a rigorous analysis of the Kickstarter crowdfunding domain to reveal several unique insights about project-, social-, temporal-, and geo-location-based features that affect the success of its project campaigns. We showed that backers are strongly influenced by their topical preference and the trust relationship towards the creator of projects. In the analysis of

network-based features, we used the network of promoters from Twitter to show that the success of projects depends on the connectivity between the promoters. Additionally, we created Twitter-based communities of Kickstarter users to study its impact on the backing habits of individuals. Our analysis revealed that the backing habits are influenced by their social circle (or community). Lastly, we reported that the effect of geo-location is not uniform for all the projects; instead, it depends on their topical category. In the second part of this paper, we used the analyzed set of features to build a model that recommends a set of backers to Kickstarter projects. Using the gradient boosting tree, a state-of-the-art learner model, and the features from only the first three days of project duration, we were able to achieve an AUC of 0.89, and a precision up to 0.8.

CHAPTER 5: PROBABILISTIC GROUP RECOMMENDATION

5.1 Introduction

In the previous chapter, we performed a detailed analysis of the Kickstarter domain where we studied the impact of backer personality, geo-location, project quality, and social network on the success of projects. In addition to this, we developed a simple content-based recommendation framework using gradient boosting tree classifier to recommend backers to Kickstarter projects. In this chapter, we extend our study on recommendation in Crowdfunding domain by proposing a *Group recommendation* model called *CrowdRec* [3] that recommends projects to a group of investors by incorporating the on-going status of projects, the personal preference of individual members, and the collective preference of the group .

Recommendation challenges in crowdfunding. Recommendation in crowdfunding poses a number of challenges. Based on our preliminary study on analyzing Kickstarter user behaviors [2], we found that a diverse set of factors collectively influence the users' decision to back a project. Hence, recommendation cannot be based on some simple set of straightforward features that are directly available from the projects. Consider the decision making of a Kickstarter investor, as illustrated in Figure 5.1; we can see that the user is influenced by (1) his own personal interest, (2) the group (or community) that the user is associated with, and (3) the real-time status of the project. The personal interest of a backer is attributed to his preferences over the topic or the geo-location of the project. The group's influence on the other hand is amplified by the pervasive growth of social media such as Twitter and Facebook, where users' decision to back a project depends not only on their personal interests, but also on their relationship to a social-group of peer investors they communicate with. Therefore, when designing a recommender system for crowdfunding, it is important to incorporate the group's influence. Finally, due to the *transient* nature of crowdfunding projects, the real-time status of the project plays a critical role in determining the backing habits of a user. Notice that, in conventional recommendation such as movies or books, it is reasonable to apply collaborative filtering techniques since the recommended items usually can serve many users for several years. This is not the case in crowdfunding; in Kickstarter, once the project expires after its posting period, we cannot recommend the project to any user any more. For

instance, as shown in Figure 5.1, the project has a duration of 30 days. Recommending it after expiration makes no sense. This transient nature of projects gives rise to some interesting real-time properties such as (a) *the popularity of projects* and (b) *the availability of rewards*. In Kickstarter, the popularity of a project could be measured based on the percentage of funds collected at a specific time. In Figure 5.1, we see that, on Day 10 the project has already collected over 70% of it's goal amount, indicating high popularity. Therefore, there is a high chance for a user to fund such popular projects. Nonetheless, the popularity is not the sole deciding factor. In the same figure, we can also see that, on Day 20, although the project remains popular, most of the rewards are already sold out (denoted by the phrase "All Gone") due to the demand. This in-turn means that people might not be interested in backing such projects (with no availability of certain type of rewards), despite it's popularity.

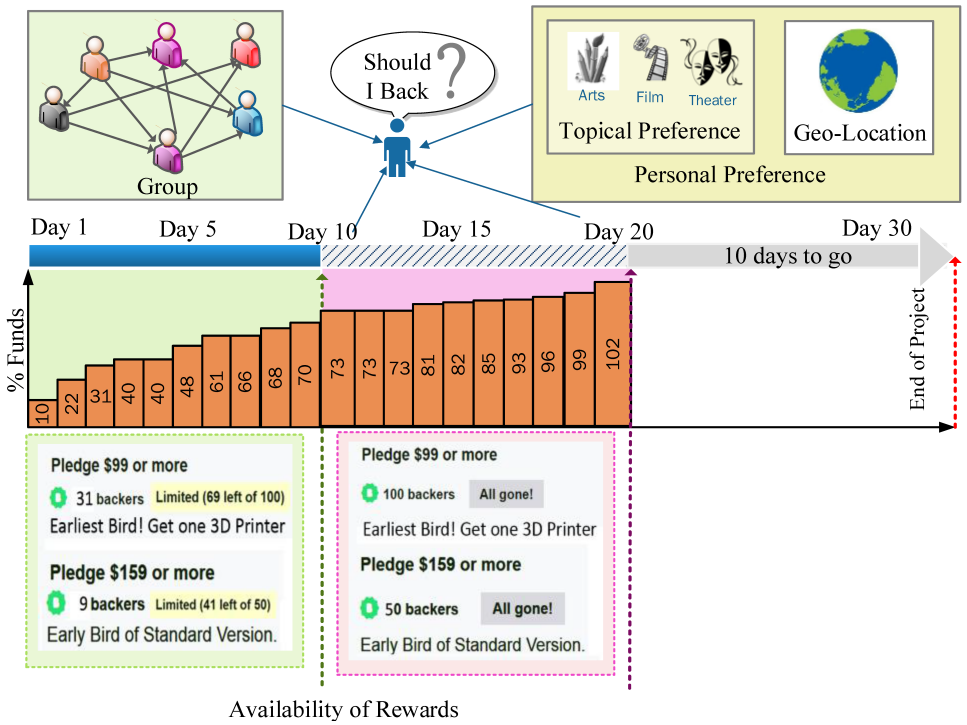


Figure 5.1: Impact of project, personal, and social network based features on Kickstarter users.

Overview of the proposed approach. To create a personalized recommendation system for crowdfunding, we propose a *group recommendation* model called *C*rowdRec. Using a probabilistic generative framework, we incorporate various heterogeneous features related to the project, users,

and their social groups to precisely model the interests of investors. Notice that in the crowdfunding scenario, projects are like *living entities that survive for a finite duration, and are affected by real-time actions such as popularity and reward availability*. Therefore, we need to leverage the information about the dynamic on-going status of a project when recommending it to the users.

Research Contributions. The major contributions of this chapter are summarized as follows:

1. We propose a group-recommendation model for crowdfunding domains, which incorporates the dynamic-status of the on-going projects to recommend Kickstarter projects for a group of investors.
2. We use a diverse set of features about the projects. These features include (1) topical preference (2) geo-location preference (3) social-network links of backers and (4) various temporal information about the projects to incorporate rich prior information into our probabilistic model.
3. Using comprehensive evaluation techniques, we show that our model outperforms a number of baselines and a state-of-the-art group-recommendation model to provide effective and meaningful recommendations for backer groups in Kickstarter.

The rest of this chapter is organized as follows. We review the related work on crowdfunding and group recommendation models in Section 5.2. In Section 5.3, we discuss the notion of groups in Kickstarter and the challenges associated with group recommendation. The CrowdRec model and it's generative process are introduced in Section 5.4, followed by the derivation of the model parameters. In Section 5.5, we show the different ways of incorporating various prior information. In Section 6.6, we explain the data collection methodology and report the experimental results for performance evaluation. Finally, we conclude the chapter in Section 5.7.

5.2 Related Work

In this section, we review two lines of related research namely, crowdfunding and recommender systems.

Crowdfunding and Kickstarter. Since crowdfunding is still an emerging research domain, most works in this area are relatively new. The dynamics of Kickstarter are examined in a recent survey

[85], while various types of crowdfunding platforms are compared in [74]. In [75, 76], the authors analyze the crowdfunding platforms to learn the motivation behind the users who create and invest in crowdfunding projects. Moreover, the effect of frequent updates over the success rate of projects is explored in [72]; and the impact of social network on Kickstarter projects is delineated in [78].

Recommendation in Crowdfunding Platforms. So far, there are very few studies on developing recommendation models for crowdfunding platforms. In [79], a personalized loan recommendation system for a micro-financial platform called *Kiva.org* is proposed. Recently, an SVM classifier is trained using updates, comments, Facebook friends and other features from Kickstarter to recommend investors to Kickstarter projects [71]. In our previous work on Kickstarter [2], a variety of traits based on backer personality, geo-location, project quality, and social network have been analyzed and incorporated into a gradient boosting tree model to recommend backers to Kickstarter projects. Finally, in our recent work [95], we formulate a survival analysis problem to predict the success of Kickstarter projects.

Probabilistic Models for Recommendation. The original aspect model is proposed by Hofmann for Probabilistic Latent Semantic Analysis (PLSA) [42, 96]. Since then, more complex machine learning models have been proposed, including multinomial mixture models, latent dirichlet allocation (LDA), markov models and latent factor models [97–99]. The PLSA aspect model has been widely used in information retrieval and data mining applications [100]. For example, in [101], the aspect model is used for recommending communities, while in [102], an additional latent variable has been added to the aspect model to capture the influence of friends on a user's topical interest.

Group Recommendation. Different from conventional recommendation which typically recommends an item to a user, the group recommendation aims to either (a) recommend a group (or community) to a user or (b) recommend an item to a group of users. In type (a), one usually recommends a set of groups (i.e., communities) to a user based on various measures such as (1) topical similarity between the community and the user, (2) popularity of the community at a given time, (3) proximity of geo-location between the user and the members of the community, etc., [101, 103–106]. Our definition of the group recommendation belongs to type (b), which is new to the data

mining research and thus very few studies have explored this research direction [107–109]. In [102], a probabilistic aspect model is proposed to learn the interests of users using both their personal preference and the influence from their social network. Using score aggregation strategies such as average and least-misery techniques [110], recommendation of items is made for a targeted group of users. In [107], a unified group recommendation model is proposed and later extended by incorporating a group-topic distribution [109] to provide an improved recommendation. The group recommendation model proposed in this chapter is uniquely different from the aforementioned works as we incorporate the "live-status" of projects along with user preference and group influence in the model.

5.3 Groups in Crowdfunding

To elucidate the need for group recommendation in Kickstarter, we obtain backers who have their Twitter profiles and retrieve their friends and followers from Twitter search API. We then create communities of these backers to analyze *whether the backing habits of these investors are influenced by their relationship to a community*[1]. The process of community creation is detailed in our previous work [2]. To measure the influence of a group over the backer, we calculate their *Affinity* score as follows:

$$\textit{Affinity}(b, g) = |F(b) \cap F(g)| \tag{5.1}$$

where $F(b)$ and $F(g)$ denote the set of all the followers and followees of a backer b and a community g, respectively; $|F(b) \cap F(g)|$ indicates the number of mutual friends between this backer and the members of the community. Figure 5.2(a) shows the outcome of this analysis, where $p(M(b, v)|M(g, v))$ indicates the probability of a user b backing a project v, given that v is backed by the members of the group g. In this notation, $M(b, v)$ denotes the action of a backer investing in project v and $M(g, v)$ denotes the same by a community g. From this figure, one can see that the stronger *affinity* of a user towards a group leads to a greater chance for this user to back the same project that was backed by the group. Therefore, given a project, the goal of our CrowdRec recommendation model is to *identify a group of users who may potentially back this project*.

[1] we also refer these communities as *groups*, so these two terms are used interchangeably.

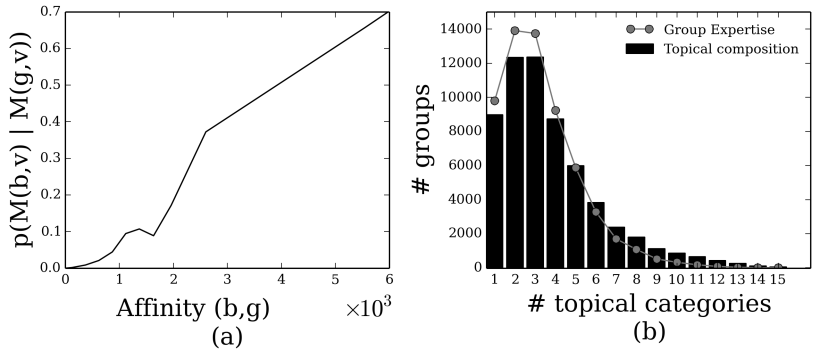


Figure 5.2: Characteristics of groups in Kickstarter. (a) shows the influence of communities (groups) over the backing habits of users and (b) shows the topical composition in groups.

Next, we analyze the preferences of individual members in a group. To proceed, for every group in our dataset, we calculate the number of unique topical categories of projects backed by the members of the group. As shown in Figure 5.2(b), although there are groups consisting of members who are interested in just one single topic, *a majority of them have diverse interests, i.e., the members have backed projects from multiple topical categories*. In addition, we analyze the expertise of individual group members by calculating the number of backers who are experts in a specific topical category of projects. We say a backer to be an expert in a topic if she has backed at least 3 projects from the same category and has over 15 backings in total. As depicted in Figure 5.2(b), a *majority of groups consist of backers who are experts in multiple topical categories*. We aim to exploit these observations in the proposed CrowdRec model.

5.4 The CROWDREC Model

In this section, we introduce CrowdRec, a probabilistic generative model for recommending crowdfunding projects to groups of users. The recommendation model aims to capture the following observations: (1) A crowdfunding group may support projects from multiple topical categories, (2) A user's backing decision is based not only on her personal preference but also on the collective preferences of her groups; (3) A group's collective preference to support a project is strongly correlated with the personal preferences of topically authoritative users (i.e., users exper-

tise) within the group, (4) The dynamic status of a project impacts both the individual investor's personal preferences and the group's collective preferences in backing crowdfunding projects.

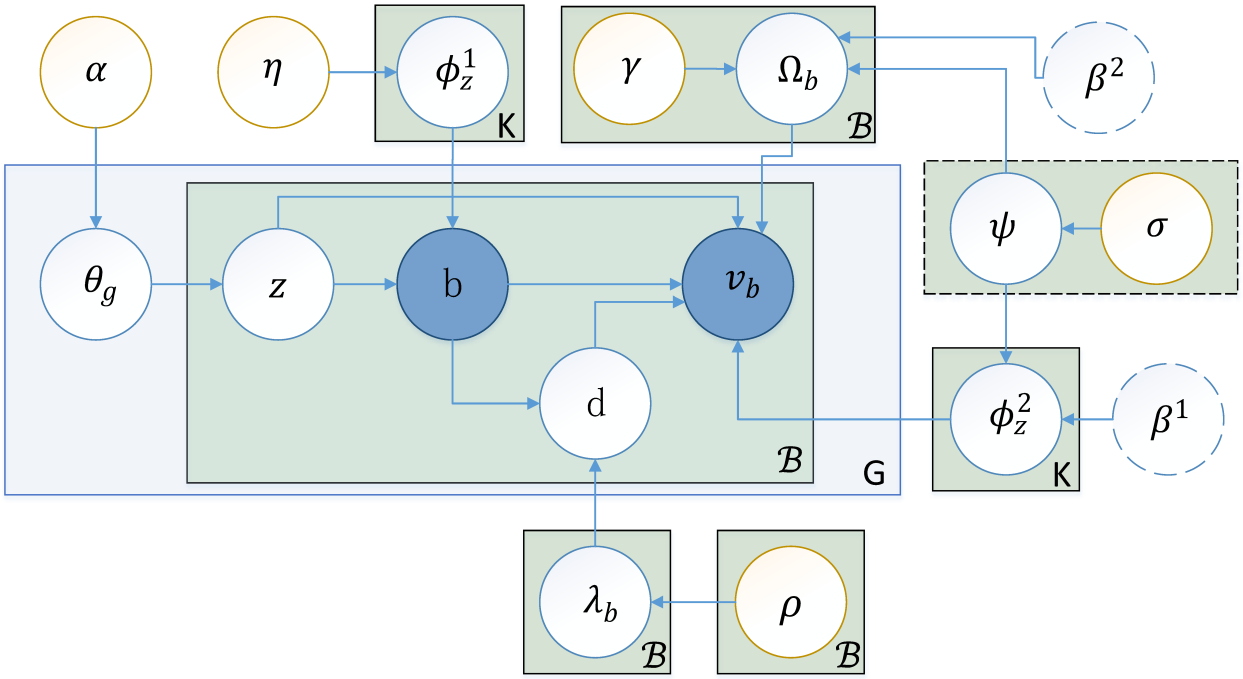


Figure 5.3: Graphical representation of the CrowdRec model.

In this section, we first formulate the modeling problem and then present the generative process captured in our CrowdRec model. Next, we show how the dynamic status of projects is incorporated into the model. Finally, we derive the parameters of the model to facilitate model learning via Gibbs sampling.

Problem Statement: Given a set of *projects* $V = \{v_1, v_2, ..., v_{|V|}\}$, a set of *backers* $B = \{b_1, b_2, ..., b_{|B|}\}$, and a set of *groups* $G = \{g_1, g_2, ..., g_{|G|}\}$, let $B^g \subset B$ denote a group of backers in group g, i.e., $B^g = \{b_1^g, b_2^g, ..., b_{|B^g|}^g\}$. The action of a group g backing a project v is denoted by $M(B^g, v) = \{(b, v)|b \in B^g\}$, where (b, v) refers to an individual b from a group g choosing to back a project v. The goal of our CrowdRec model is to recommend a ranked list of V projects to a target (or new) group $\widetilde{g}$.

5.4.1 Generative Process

The Graphical representation of the CrowdRec model is shown in Figure 5.3. We describe the generative process in our model as follows.

- Each group $g \in G$ is composed of members who are interested in certain particular project categories. Therefore, the collective preference of the group is captured by θ_g, which is represented as a distribution over a universal set of latent topics in projects. Here, θ_g follows a symmetric Dirichlet distribution, i.e., $\theta_g \sim Dirichlet(\alpha)$.
- Based on θ_g, the group g chooses a single topic z and nominates a user b to decide whether to back a project v. The distribution ϕ_z^1 captures the expertise of this backer over the topic z.
- To decide whether to back a project v, the nominated user relies on either her own personal interest or the collective preference of the group. This decision is governed by the random variable D, which takes the binary value 0 or 1. Thus, we model d using a binomial distribution with beta prior.
- If d is 0:
  - The user picks the project based on (1) *the group's influence* (i.e., collective preference), which is a multinomial over the distribution ϕ_z^2 and (2) *the current status of the project*. The current status of the project (in the dotted box) is denoted by a ψ, $\psi \sim Dirichlet(\sigma)$.
- If d is 1:
  - The user picks the project based on (1) *her own topical interest* (personal preference), which is a multinomial over the distribution Ω_b and (2) *the current status of the project*.

Incorporating Dynamic Status of Projects. Notice that the status of a project plays a key role in the generative process described above. It is an external factor determined by the progression of the project over time. In Figure 5.3, this is indicated by the *project-status* distribution ψ, which is constrained by a dirichlet prior σ. ψ in-turn affects the *project-topic* distribution ϕ^2 and *user-project* distribution Ω. When the user relies on the group's influence to back a project v, the project-topic distribution ϕ^2 (i.e. $V \times K$ matrix) is multiplied by the project-status distribution ψ

Algorithm 3: Generative process of `CrowdRec` model

for *each project* $v \in \mathcal{V}$ **do**
  Draw $\psi_v \sim Dirichlet(\sigma)$
end
for *each topic* $z_k, k \in \mathcal{K}$ **do**
  Draw $\phi_z^1 \sim Dirichlet(\eta)$
  Draw $\phi_z^2 \sim Dirichlet(\psi\beta^1)$
end
for *each backer* $b \in \mathcal{B}$ **do**
  Draw $\Omega_b \sim Dirichlet(\psi\beta^2\gamma)$
  Draw $\lambda \sim Beta(\rho)$
end
for *each group* $g \in G$ **do**
  Draw $\theta_g \sim Dirichlet(\alpha)$
  for *each backer,*b *in group* g **do**
    Draw $z \sim Multinomial(\theta_g)$
    Draw $b \sim Multinomial(\phi_z^1)$
    Draw the *decision* $d \sim Bernoulli(\lambda_b)$
    if *d = 0* **then**
      Draw $v \sim Multinomial(\phi_z^2)$
    end
    if *d = 1* **then**
      Draw $v \sim Multinomial(\Omega_b)$
    end
  end
end

(i.e. $1 \times V$ matrix). In other words, ψ becomes a prior for ϕ^2. Alternatively, if the user backs a project based on her own preference, Ω (i.e. $B \times V$ matrix) is multiplied by ψ.

In the CrowdRec model, β^1 and β^2 are concentration scalars that affects the extent to which a group (or a user) relies on project status to make the backing decision. When β^1 is high, the group strongly relies on the on-going status to back the project, which means ϕ^2 becomes similar to ψ. Alternatively, if β^1 is low, the group's decision to back a project is independent of the on-going status of the project. Same applies for the scalar β^2, which affects the variable Ω. In the literature, this type of formulation is known as the *hierarchical Polya-Urn model*, which has been used to model the global and local topic distributions in LDA [111–113]. Algorithm 4 summarizes the complete generative process and Table 6.1 provides the list of symbols used in this research.

Table 5.1: List of notations used in this chapter.

Symbol	Description
$\mathcal{V} = \{v_i\}$	project set, v_i indicates a single project
$G = \{g_j\}$	crowdfunding groups, g_j indicates a single group
D	a binary decision variable, representing d=1 or d=0
$Z = \{z_i\}$	latent topics assigned to projects in Z
K	number of topics specified as parameter
$i = (b, v)$	a tuple that indicates backer b picks project v
θ_g	topic distribution of a group g
ϕ_z^1	$B \times K$ latent matrix for a backer b
ϕ_z^2	$G \times K$ latent matrix for a group g
Ω_b	$B \times V$ matrix for a backer b
λ_b	prior for the binary decision variable D
ψ	dynamic status distribution of a project v
$\alpha, \eta, \gamma, \sigma, \rho$	parameters of $\theta, \phi^1, \Omega, \psi, \lambda$
β_1, β_2	concentration scalars for ϕ^2, Ω
$c_{k,g,i}$	# times i is assigned to topic k in group g
$c_{k,g,b}$	# times backer b is assigned to topic k in group g
$c_{k,g,v}$	# times project v is assigned to topic k in group g
$c_{b,g,v}$	# times project v is assigned to backer b in group g
$c_{b,g,d}$	# times choice d chosen by a backer b in group g

5.4.2 Parameter Estimation

To learn the parameters in the CrowdRec model, the eastimation of the posterior is given by:

$$p(z, d|b, v, \alpha, \eta, \gamma, \sigma, \rho) = \frac{p(z, d, b, v|.)}{p(b, v|.)} \tag{5.2}$$

The likelihood of the above equation is expanded as follows:

$$\begin{aligned} & p(z, d, b, v|.) \\ & = \underbrace{\int p(z|\theta)p(\theta|\alpha)d\theta}_{(A1)} . \underbrace{\int p(b|z, \phi^1)p(\phi^1|\eta)d\phi^1}_{(A2)} . \underbrace{\int p(d|\lambda)p(\lambda|\rho)d\lambda}_{(A3)} . \\ & \underbrace{\int\int\int p(v|b, d, z, \Omega, \phi^2)p(\Omega|\gamma, \psi, \beta^2).p(\phi^2|\psi, \beta^1)p(\psi|\sigma)d\Omega d\phi^2 d\psi}_{(A4)} \end{aligned} \tag{5.3}$$

To infer the parameters $\phi^1, \Omega, \psi, \phi^2$ and λ, we obtain samples from this high-dimensional distribution using collapsed Gibbs sampling-based approach. It is important to note that there are complex relationships between the latent-topic variable Z and the latent-decision variable D. To

overcome this problem, we adopt the two-step Gibbs sampling method proposed by Yuan et. al. [109] by decomposing the expression $(A4)$ of Equation (5.3) as follows:

$$\begin{aligned}
&\int\int\int p(v|b,d,z,\Omega,\phi^2).p(\Omega|\gamma,\psi,\beta^2).p(\phi^2|\psi,\beta^1).p(\psi|\sigma)d\Omega d\phi^2 d\psi \\
&= \underbrace{\int\int p(v^0|z,d^0,\phi^2)p(\phi^2|\psi,\beta^1)p(\psi|\sigma)d\phi^2 d\psi}_{(B1)} \\
&\quad \underbrace{\int\int p(v^1|b,d^1,\Omega)p(\Omega|\gamma,\psi,\beta^2)p(\psi|\sigma)d\Omega d\psi}_{(B2)}
\end{aligned} \tag{5.4}$$

where expression $(B1)$ corresponds to the decision variable $d = 0$; in other words, when a user chooses to back a project v^0 based on his group's interest and expression $(B2)$ corresponds to the decision $d = 1$ i.e., when a user chooses a project v^1 based on her own interest. Therefore, using Gibbs sampling, we sample the latent variable d for two different cases: (a) when $d = 0$ and (b) when $d = 1$. Similarly, we sample the latent variable z when (a) project v^0 is chosen and (b) when project v^1 is chosen.

The derivation of the collapsed Gibbs sampling equation for the topic-latent variable z and the decision-latent variable d is similar to [109]. The probability of a tuple $i = (b, v)$ belonging to a latent topic z is derived as follows:

$$p(z_{g,i} = k|Z^{-(g,i)}, v^0, b) \propto \frac{c^{-(g,i)}_{k,g,i^*} + \alpha_k}{c^{-(g,i)}_{k^*,g,i^*} + \alpha_{k^*}} \cdot \frac{c^{-(g,i)}_{k,g^*,b} + \eta_b}{c^{-(g,i)}_{k,g^*,b^*} + \eta_{b^*}} \cdot \left(\frac{c^{-(g,i)}_{k,g^*,v}}{c^{-(g,i)}_{k,g^*,v^*}} + \beta^1 \frac{c^{-(g,i)}_{k^*,g^*,v} + \sigma_v}{c^{-(g,i)}_{k^*,g^*,v^*} + \sigma_{v^*}} \right) \tag{5.5}$$

$$p(z_{g,i} = k|Z^{-(g,i)}, v^1, b) \propto \frac{c^{-(g,i)}_{k,g,i^*} + \alpha_k}{c^{-(g,i)}_{k^*,g,i^*} + \alpha_{k^*}} \cdot \frac{c^{-(g,i)}_{k,g^*,b} + \eta_b}{c^{-(g,i)}_{k,g^*,b^*} + \eta_{b^*}} \tag{5.6}$$

In the above, the variable of type $c_{x,y,z}$ indicates a count as described in Table 6.1, and the symbol $*$ over the subscript variables denotes the summation over the respective sub-script variables. For example, $c_{k,g,v}$ indicates the number of times the project v is assigned to topic k in group g and $c_{k,g*,v}$ is the same variable that is summed across all the groups $g \in G$. The superscript symbol

$-(g,i)$ means that we exclude the i^{th} tuple for group g when sampling. The probability of a tuple $i=(b,v)$ choosing a decision d is derived as follows:

$$p(d_{g,i}=0|D^{-(g,i)},Z,v,b) \propto$$
$$\frac{c^{-(g,i)}_{b,g^*,d_0}+\rho_0}{c^{-(g,i)}_{b,g^*,d_1}+c^{-(g,i)}_{b,g^*,d_0}+\rho_0+\rho_1}\cdot\left(\frac{c^{-(g,i)}_{k,g^*,v}}{c^{-(g,i)}_{k,g^*,v^*}}+\beta^1\frac{c^{-(g,i)}_{k^*,g^*,v}+\sigma_v}{c^{-(g,i)}_{k^*,g^*,v^*}+\sigma_{v^*}}\right) \tag{5.7}$$

$$p(d_{g,i}=1|D^{-(g,i)},Z,v,b) \propto$$
$$\frac{c^{-(g,i)}_{b,g^*,d_1}+\rho_1}{c^{-(g,i)}_{b,g^*,d_1}+c^{-(g,i)}_{b,g^*,d_0}+\rho_0+\rho_1}\cdot\left(\frac{c^{-(g,i)}_{g^*,b,v}+\gamma_v}{c^{-(g,i)}_{g^*,b,v^*}+\gamma_{v^*}}+\beta^2\frac{c^{-(g,i)}_{k^*,g^*,v}+\sigma_v}{c^{-(g,i)}_{k^*,g^*,v^*}+\sigma_{v^*}}\right) \tag{5.8}$$

After obtaining sufficient number of samples using the above Gibbs update rules, we can finally infer the parameters $\phi^1_z, \Omega_b, \psi, \phi^2_z$ and λ_b as follows:

$$\phi^1_{z,b}=\frac{c_{k,g^*,b}+\eta_b}{c_{k,g^*,b^*}+\eta_{b^*}} \tag{5.9a}$$

$$\Omega_{b,v}=\frac{c_{g^*,b,v}+\gamma_v}{c_{g^*,b,v^*}+\gamma_{v^*}}+\beta^2\frac{c_{k^*,g^*,v}+\sigma_v}{c_{k^*,g^*,v^*}+\sigma_{v^*}} \tag{5.9b}$$

$$\psi_v=\frac{c_{k^*,g^*,v}+\sigma_v}{c_{k^*,g^*,v^*}+\sigma_{v^*}} \tag{5.9c}$$

$$\phi^2_{z,v}=\frac{c_{k,g^*,v}}{c_{k,g^*,v^*}}+\beta^1\frac{c^{-(g,i)}_{k^*,g^*,v}+\sigma_v}{c^{-(g,i)}_{k^*,g^*,v^*}+\sigma_{v^*}} \tag{5.9d}$$

$$\lambda_b=\frac{c_{b,g^*,d_1}+\rho_1}{c_{b,g^*,d_1}+c_{b,g^*,d_0}+\rho_0+\rho_1} \tag{5.9e}$$

Recommending projects: To recommend a set of projects to a new group $\widetilde{g}$, we need to learn the group-topic distribution $\theta_{\widetilde{g}}$. This is done by estimating the posterior distribution of topics $\widetilde{z}$, given the backers $\widetilde{b}$ and the estimated backer-topic distribution ϕ^1_z and the hyperparameter α_k that was obtained from our CrowdRec model $\mathcal{M}$.

$$p(\widetilde{z}_{\widetilde{g},j}=k|\mathcal{B}_j=b_j,\widetilde{Z}^{-(\widetilde{g},b)},\mathcal{B}^{-(j)};\mathcal{M}) \propto \phi^1_{k,b}(c^{-(\widetilde{g},i)}_{k,\widetilde{g},i^*}) \tag{5.10}$$

Once we sample the topics for this new group $\widetilde{g}$, we recommend a new project based on the following equation.

$$p(v|\widetilde{g}) = \prod_{b \in \mathcal{B}^{\widetilde{g}}} \psi_v \sum_{z \in Z} \theta_{\widetilde{g},z}.(\lambda_b.\Omega_{b,v} + (1-\lambda_b).\phi^2_{z,v}) \tag{5.11}$$

The above equation captures the following components when recommending a project to a group: (1) the individual's personal preference over a project Ω, (2) the influence of the group over a backer is captured by ϕ^2, (3) the topical preference of the group and the users are captured by θ and Ω, and (4) finally, the dynamic status (or popularity) of the project ψ.

5.5 Prior Information

In this section, we discuss the approaches we adopt to estimate the priors incorporated in CrowdRec. Notice that we have priors for two different distributions: (1) a static prior γ for the distribution Ω_b, which is a $B \times V$ matrix indicating the preferences of backers towards the Kickstarter projects; (2) a dynamic prior σ for distribution ψ, which is a $1 \times V$ row matrix indicating the on-going status of the project.

For the first case, we estimate the static prior γ by exploiting the backing history of all the users $b \in B$. This backing history, denoted by $\mathcal{H}_b$, contains the details such as a project's topical category, the geo-location of project, and the person who created the project.[2] For most part, this information remains static. Since it is extracted from the backing history of users, we call γ as *user-specific* prior. On the contrary, in the second case, the prior information σ is not static due to the transient nature of the projects. As σ is changed at regular time intervals, we term it as *dynamic-status* prior. The calculation of these priors are detailed in the following sections.

User-Specific Priors: We incorporate three features from the users' backing history to create the user-specific prior γ, namely: (a) topical preference (b) creator preference, and (c) geo-location preference.

Topical Preference: In our previous work [2], we observed that Kickstarter users have a strong topical preference in their decisions to back a project. We assume users have a tendency to contin-

[2]In our experiments, we consider city and state as the geo-location of the projects.

uously back projects in the same topical category. This tendency can be modelled as a conditional probability of a user b to back a project in topic t, given t is present in the backing history of this user, denoted by $P((b,t)|t{\in}\mathcal{H}_b)$. Using Bayes$'$ theorem, it is derived as follows:

$$P((b,t)|t{\in}\mathcal{H}_b) = \frac{P(t{\in}\mathcal{H}_b|(b,t))P(b,t)}{P(t{\in}\mathcal{H}_b)} \tag{5.12}$$

Creator Preference: Users tend to develop an inclination towards creators whom they have backed in the past. We represent this as the conditional probability of a user b to back a creator e, given that e is in the backing history of this user, denoted by $P((b,e)|e{\in}\mathcal{H}_b)$. It is calculated in a similar way as that of Equation (5.12).

Geo-location Preference: Geo-location has a strong impact on the success of projects, and the level of impact depends on the topical category of the project [2]. For instance, we find that projects based on technology are relatively less dependent on their geo-location, while projects on theatrical arts are highly dependent. Therefore, we incorporate this information as prior by calculating the probability of a user b to back a project v, given b and v are from the same geo-location. This probability is represented by $p((b,v)|Loc(b,v) = \ell, \tau(v) = t)$, where $Loc(b,v)$ indicates the geo-location of the backer and project, $\tau(v)$ is the topic of the project v. This probability is calculated using Bayes$'$ theorem by expanding the likelihood using the chain rule of probability.

Finally, the interest of a user b towards a project v is obtained as a linear combination of topic-, creator-, and geo-location-based preferences as follows:

$$\gamma_{b,v} = p((b,t)|.) + p((b,e)|.) + p((b,v)|.) \tag{5.13}$$

Dynamic-Status Priors: We incorporate two factors in the prior σ for the project status ψ, namely, (1) the popularity of the project, and (2) the availability of popular rewards at specific time t.

Popularity of project: The popularity of a project at time t is derived as follows:

$$\mathcal{P}o_t(v) = \frac{\text{\# pledged amount of project } v \text{ at time } t}{\text{Goal amount of the project } v} \tag{5.14}$$

where $\mathcal{P}o_t(v)$ is the *popularity score* of project v at time t.

Availability of rewards: As explained earlier in Figure 5.1, the status of the project cannot be expressed purely based on it's popularity, but the availability of rewards should also be considered. To verify our claim, we extract the top 3 popular reward categories of projects and obtain the additional percentage of backers backing the project on the day when the rewards are sold out. The result of this analysis is given in Figure 5.4, which shows that the interest of users in backing a project decays with the depletion of popular reward categories. Therefore, we calculate this prior information as follows:

$$\mathcal{R}_t(v) = \frac{\text{\# rewards sold out at time } t}{\text{\# limited rewards}} \tag{5.15}$$

where $\mathcal{R}_t(v)$ is the *reward score* of the project v at time t.

Using Equations (5.14) and (5.15), we define the dynamic-status prior for a project v as follows:

$$\sigma_{(v,t)} = \mathcal{R}_t(v) \times \mathcal{P}o_t(v) \tag{5.16}$$

The score $\sigma_{(v,t)}$ is constantly updated at different time intervals $\{t1, t2..., tn\}$ until the project expires.

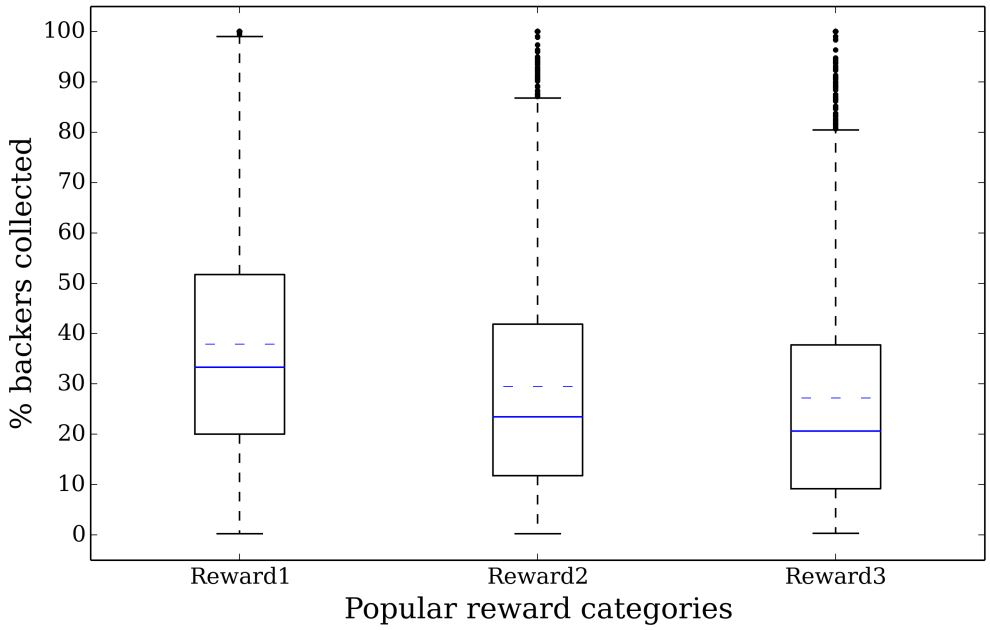


Figure 5.4: Decay of user interest with the depletion of popular reward categories.

5.6 Experiments

In this section, we conduct comprehensive experiments to evaluate the performance of the CrowdRec model in comparison with state-of-the-art models. In the following, we first describe the datasets used in our experiments and then report the experimental results.

5.6.1 Dataset Description

For our experiments, we obtain six months of Kickstarter data (12/15/13 to 06/15/14) from *kickspy*, which consists of 27,270 projects. We remove projects that were canceled or suspended, with less than one backer, or with less than $100 of pledged amount. Next, we build a web-crawler to fetch backers from this filtered set of projects. As such, we obtain over 1 million backers for the remaining 18,143 projects after the removal process.

In our previous study [2], we show that backers in Kickstarter are distinguished by two important features: (a) the presence of social network profile,[3] and (b) the backing frequency. Therefore, in this work, we leverage this information to categorize the backers $b \in \mathcal{B}$ into two types: (i) those who have linked their Kickstarter profiles to Twitter, and (ii) those without a Twitter profile. We denote these two types by Twt and Kck, respectively. In addition, we classify the backers into *occasional* backers who have backed 2-10 projects (denoted by *Occ*) and *experienced* backers who have backed over 10 projects (denoted by *Exp*). Thus, we have four different datasets: (1) **Twt-Occ** $= \{b|b \in \text{Twt} \cap (2 < Backings(b) < 10)\}$; (2) **Twt-Exp** $= \{b|b \in \text{Twt} \cap (Backings(b) > 10)\}$; (3) **Kck-Occ** $= \{b|b \in \text{Kck} \cap (2 < Backings(b) < 10)\}$ and (4) **Kck-Exp** $= \{b|b \in \text{Kck} \cap (Backings(b) > 10)\}$

Group Creation. A group consists of backers who have backed the same project. For the four datasets described above, we create groups of Kickstarter users using the methodology described in [110], i.e., in our group creation process, we calculate the inner group similarity between the group members using Pearson correlation co-efficient (PCC) and filter out groups which have PCC less than 0.2. The statistics of our backer-group datasets are shown in Table 5.2.

[3] we only consider Twitter profiles since Facebook data is not publicly available.

Table 5.2: Statistics of Kickstarter groups formed by frequent and occasional backers.

Dataset	#Bkrs/Grp	#Grps	Avg. #Prjs/Grp	#Prjs
Kck-Occ	10	30,100	2	1,609
Kck-Exp	10	20,675	4	1,397
Twt-Exp	5	3,373	3	959
Twt-Occ	5	3,513	2	1,104

5.6.2 Performance Evaluation

In the following, we first discuss the performance metrics employed in our evaluation and a number of recommendation models examined for comparison. Our evaluation is performed over all the four datasets *Twt-Occ*, *Twt-Exp*, *Kck-Exp*, and *Kck-Occ* by randomly holding-off 20% of the ground truth for testing. For most of our experiments we set the parameters β_1 and β_2 to be 0.5, the topic parameter K to 200 and the dynamic-status prior $\sigma_{(p,t)}$ is calculated using the 50% of the total project duration. The CrowdRec model and all other baselines are implemented using python's numpy numerical module, and scikits machine learning module.[4] The codes of our model are publicly hosted in the Github page.[5]

Evaluation Metrics

To evaluate the performance of ranking, we use the standard information retrieval measures. For every project, we compute: 1) P@N: *precision at rank N* is defined as the fraction of rankings in which the true backers are ranked in the *top-N* positions, 2) MRR: The *mean reciprocal rank* is the inverse of the position of the first true backer in the ranked set of backers produced by a recommendation model, 3) S@N: The *success at rank N* is the probability of finding at least one true backer in the *top-n* ranked set, and 4) DCG: The *discounted cumulative gain* [67] is based on the fact that highly relevant backers are more important than marginally relevant ones.

[4]http://scikit-learn.org

[5]https://github.com/magnetpest2k5/Crec

Baseline Methods for Comparison

We compare the performance of our model with a simple collaborative filtering-based approach that uses various aggregation strategies for group recommendation and other state-of-the-art group recommendation models as described below:

(1) **Collaborative Filtering with averaging (*CFA*)**: First we learn user-project preference using user-based collaborative filtering. We then take the average of the recommended scores for a group and rank the preference scores to recommend a project to the group.

(2) **Collaborative Filtering with least-misery strategy (*CFL*)**: First we learn the user-project preference using user-based collaborative filtering. We then take the least scores as the recommended scores for a group and rank the preference scores to recommend a project to the group.

(3) **Collaborative Filtering with relevance disagreement (*CFR*)**: First we learn user-project preference using user-based collaborative filtering, and then takes the relevance score as CFA and the disagreement is calculated as the difference between the preference scores of individuals within a group.

(4) **COM Model:** The state-of-the-art group recommendation model that does not include the dynamic-status component [109].

Experimental Results

Overall Performance: Figure 5.5 shows that the performance of CrowdRec and COM are distinctly better than the collaborative filtering-based group recommendation techniques. Although, the collaborative filtering with averaging technique (CFA) produces better results than CFL and CFR, the approaches that heuristically aggregate the individual scores of backers to determine the groups' preference towards projects do not produce reasonable results.[6] It is also clear that the CrowdRec performs better than COM model in terms of both precision and recall. This shows that users strongly rely on the on-going status information of projects to make backing decisions.

[6] In [109] the authors report similar observations.

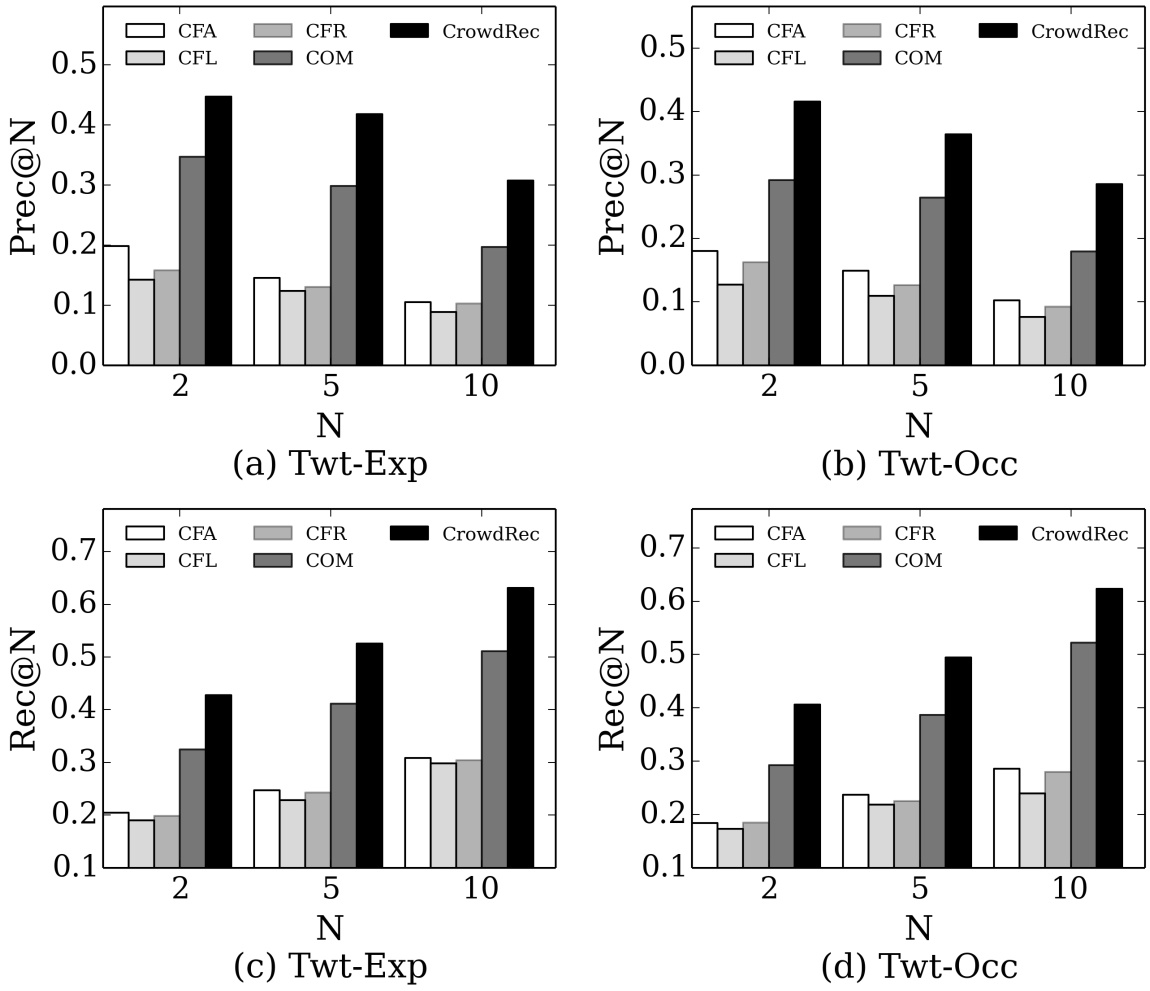


Figure 5.5: The precision and recall performance over experienced and occasional backers with Twitter profiles.

We also observe that the performance of CrowdRec over experienced backers (Figures 5.5(a) and 5.5(c)) is better than occasional backers (Figures 5.5(b) and 5.5(d)), because the former have a higher backing count, which provides a richer set of prior information about the backer's preference over topics, creators and geo-location compared to the occasional backers. Figure 5.6 shows similar set of results for the Kck dataset. In comparison to results shown in Figure 5.5, we see that the performance of our model slightly decreases in the dataset Kck. This is because the backers with Twitter profiles (i.e. dataset Twt) constantly receive Tweets about Kickstarter projects from their friends and followees, which leads to a better communication with their group members. Since one of the key components of our model is to effectively incorporate the groups' influence,

it has a stronger impact over these type of users. Finally, in Table 5.3, we show the superior performance of CrowdRec over all other models by averaging the Success@N measure over all four datasets.

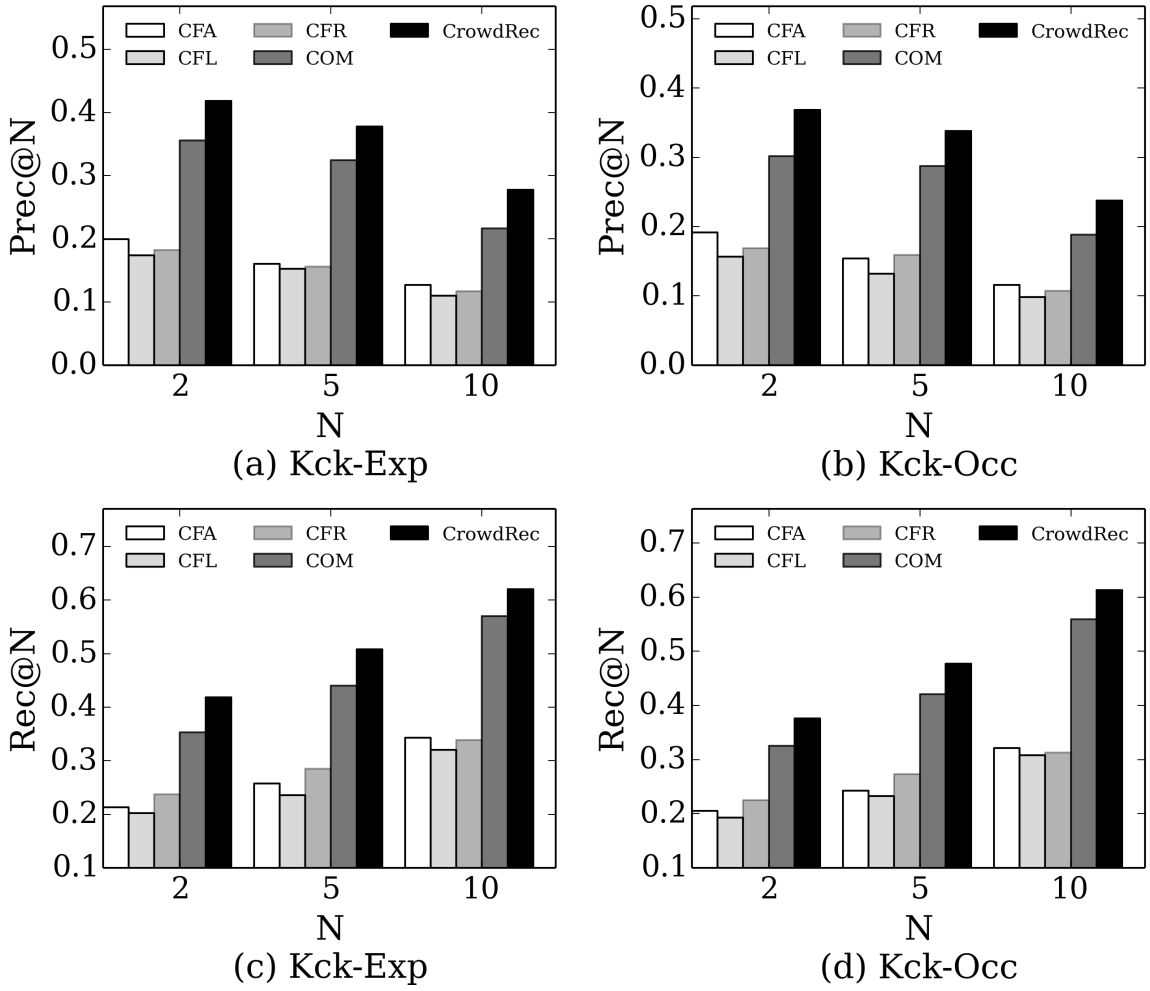


Figure 5.6: The precision and recall performance over experienced and occasional backers without Twitter profiles.

Table 5.3: The Average Performance over all datasets using Success @ N.

Model	Success@2	Success@5	Success@10
CFA	0.2638	0.3014	0.3352
CFR	0.2518	0.2857	0.3017
CFL	0.2698	0.3138	0.3369
COM	0.6347	0.7143	0.7584
CrowdRec	**0.6926**	**0.7436**	**0.7832**

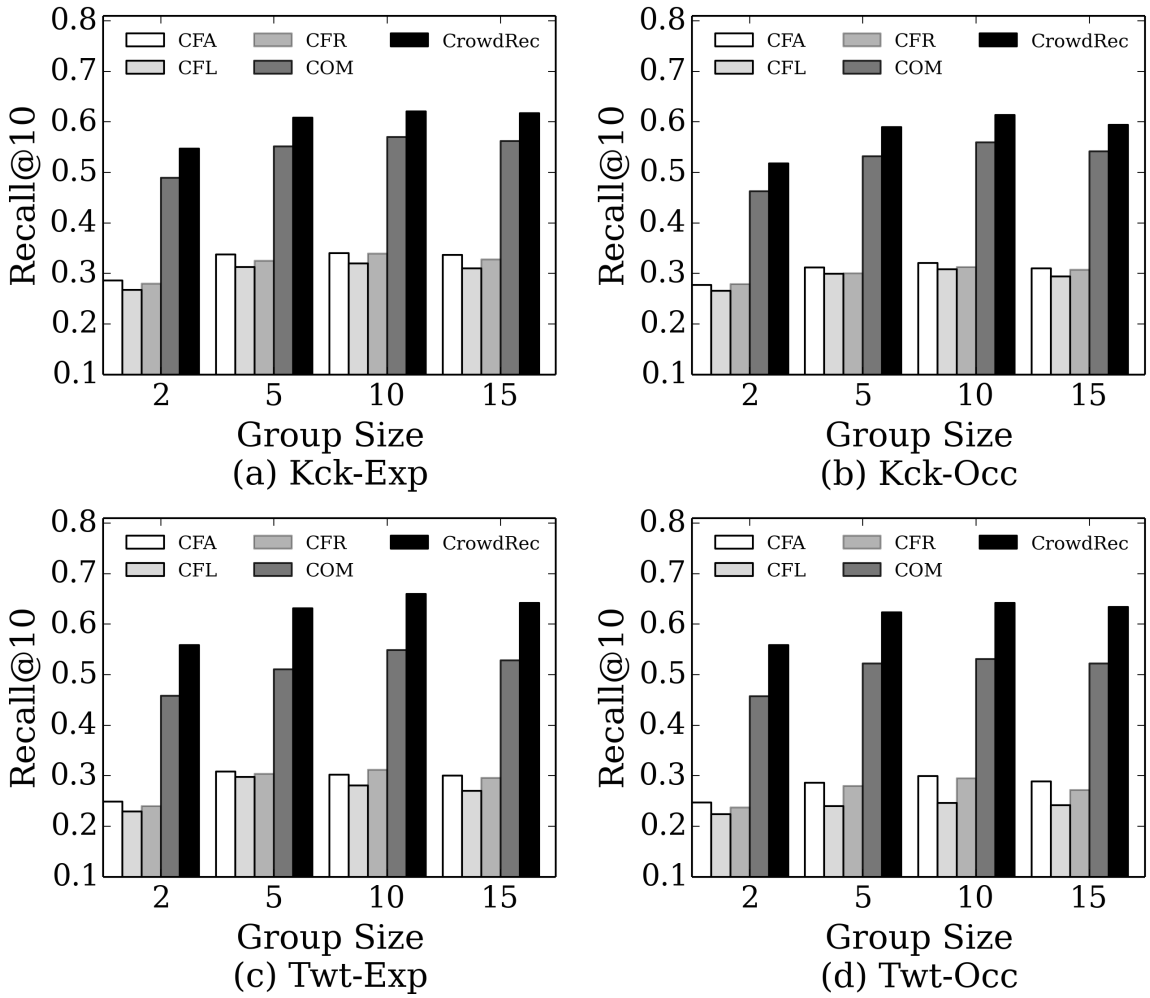


Figure 5.7: Effect of group sizes of Kickstarter users over the recall performance for all datasets.

Impact of Group Size: Depending on the number of backers in a group, a group can become more diverse or conservative in terms of their topical-preference. Therefore, we show the impact of group size on the performance of our model in Figure 5.7. Due to space constraints, we only show the recall performance for top 10 recommended projects. We again observe that CrowdRec performs better than COM and all other models irrespective of the group sizes. The performance of CrowdRec increases as we move from group size 2 to group size 5. However, this improvement becomes insignificant as the group size further increases. In fact, we observe that the performance slightly reduces for group sizes of 15 and above, mainly due to the sparsity of the data.[7]

[7]Groups greater than 15 members are extremely few in number.

Effect of Topic Size: To study the effect of topic size K over the performance of our model, in Figure 5.8, we plot the DCG scores of the top 10 recommended projects by varying K from 25 to 300. Similar to our prior observations, we see that the CrowdRec outperforms COM and all other models. Additionally, both CrowdRec and COM performs better on the backers with Twitter profiles (i.e. Twt-Exp and Twt-Occ) when compared to backers without Twitter profile (i.e. Kck-Exp and Kck-Occ). It is also important to note that the DCG values for experienced backers (Figure 5.8(a)) are higher than the occasional backers (Figure 5.8(b)), mainly because experienced backers have a much higher backing count (and thus richer prior information) than the occasional backers. Finally, we observe that the increase in the number of topics does not necessarily translate to a better performance. Although the DCG scores improves with the increase in topic count, the performance becomes static at about 200 topics. In fact, there is a slight decrease in the DCG scores when the topic count goes past 200.

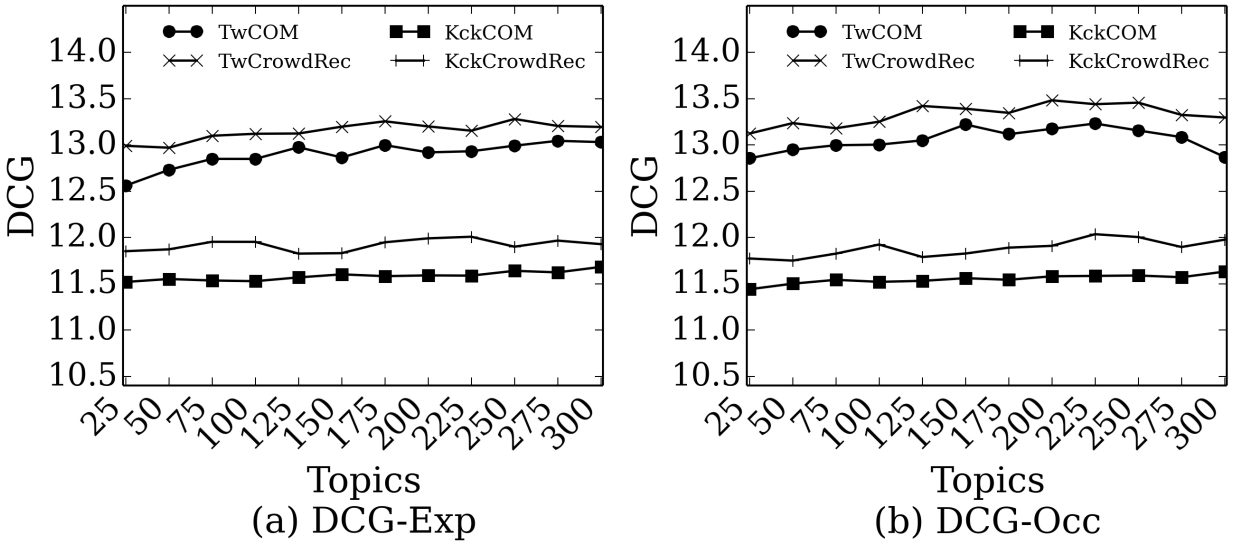


Figure 5.8: Effect of topics on the DCG measure.

Effect of Prior Information: Figure 5.9 shows the effect of various user- and project-based features that were used as priors in our model. In this figure, the y-axis denotes the priors and the x-axis indicates the MRR scores of the top 10 recommended projects. We begin with a symmetric prior (*s-prior*) and gradually add other features to the prior distribution, which is indicated by the +

symbol. For instance, Topic$^+$ implies that we are including the topical-preference of users, which was calculated in Equation (5.12); similarly, Cr$^+$ implies we add creator preference into the prior information. We observe that simply by including the topic prior provides a significant boost to the MRR scores indicating that backers strongly depend on their topical interest to fund a project. Although, the addition of creator (*Cr*) and geo-location (*Geo*) preferences of backer improves the performance of the model, though it is not as significant as the topical-preference. Finally, the inclusion of the popularity prior (*Pop*) provides a significant boost yet again, which shows the importance of the information about the on-going status of the Kickstarter projects.

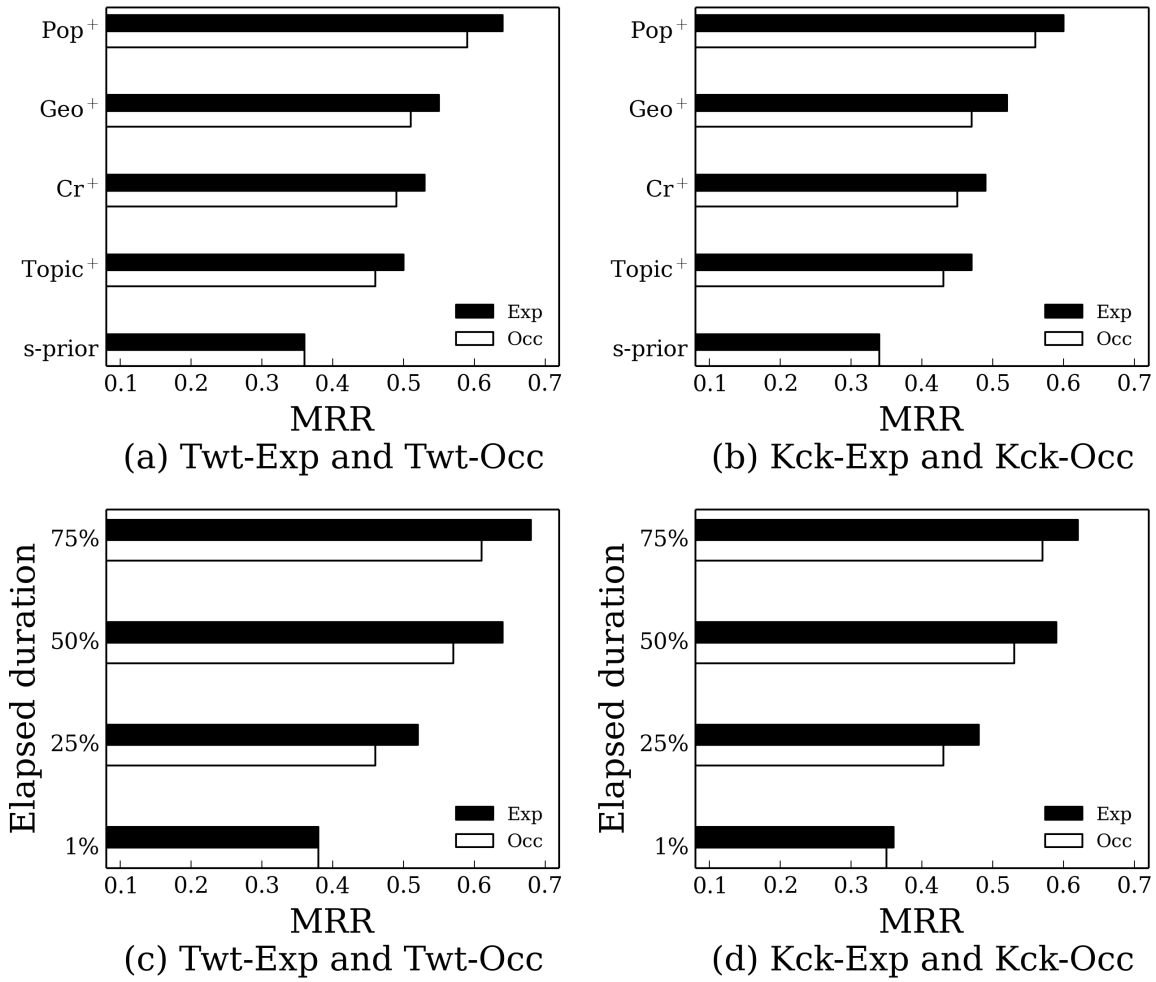


Figure 5.9: Effect of prior information ((a) and (b)) and project duration ((c) and (d)) on the recommendation performance. The terms *Topic*, *Cr*, *Geo* and *Pop* indicates the topical, creator, geolocation and popularity based priors, respectively.

Effect of Dynamic Status: Lastly, we show the effect of dynamically varying prior information in Figures 5.9(c) and 5.9(d). We calculate the status prior $\mathcal{R}_t$ using the Equation (5.14) at various intervals of the projects' duration ranging from 1% to 75% of the total project duration, which is indicated by the y-axis of this figure. It can be seen that, in general, the recommendation performance increases with the progression of the project. This is because, as the project progresses, we can obtain a much accurate estimation about it's status both in-terms of popularity and the availability of rewards.

5.7 Summary

In this chapter, we introduced a recommendation framework for a popular crowdfunding platform, i.e., Kickstarter. We point out the challenges arising in Kickstarter, where the backing habits of its users depend on a diverse set of features, including topical, geo-location, temporal, and social traits. By exploiting the notion of groups, we proposed a recommendation model that effectively incorporates all these features when recommending projects to groups of Kickstarter users. Using a real dataset, we conducted a comprehensive evaluation to show that our model outperforms other state-of-the-art group-recommendation models in terms of a variety of performance metrics. Finally, we also studied the impact of various prior information and show that the on-going status (or popularity) of the projects plays an important role in improving the recommendation performance.

CHAPTER 6: LOCATION RECOMMMENDATION FOR TRAVELERS

6.1 Introduction

Tour recommendation has become a new trend in the field of intelligent urban navigation. The dramatic increase in the amount of publicly available check-in data has generated substantial interest among different research communities to work on this problem. Different from conventional way of recommending independent venues, the objective of tour recommendation is to suggest a sequence of points of interest (POIs) that will serve as travel itineraries to users. Tour recommendation is more challenging than the conventional one due to two main reasons. First, since most users are not native to their tour destination (i.e. users are tourists), the check-in information of these users is extremely sparse. Therefore, using simple collaborative filtering based techniques will yield poor recommendation results. To overcome this problem, it is *crucial to learn the topical preference of users* from their historical check-ins and incorporate them as content-based features to create a hybrid recommendation model. Second, many researchers have shown that human mobility exhibits a strong temporal pattern [114, 115]. Unlike conventional recommendation, where POI suggestions are made in a disjoint manner, these *temporal features play a critical role in determining the next check-in spot* when suggesting a sequence of POIs for travelers.

In addition to the above-mentioned traits of tour recommendation, LBSNs such as Foursquare and Yelp enable travelers to communicate with other fellow travelers, interact with the residents from their tour destination and add other users to their social network of friends to make an informed choice about the travel destination. Therefore, it is *essential to utilize the social network of travelers as implicit meta-data information* to create a robust recommendation framework. Finally, travelers have limited duration of stay and they tend to prefer venues that are popular and well rated. Consequently, it is important to factor-in the POI-specific characteristics such as *geographical distance* between venues and their *popularity*. We illustrate the above-mentioned traits using a toy example in Figure 6.1 that depicts a tourist who begins his journey at New York's JFK airport. Let us assume that the tourist plans to stay for just 1 day and his topical interests are nature and history museums. Based on the time constraints and topical preference, one logical sequence would be to start with Prospect park, which is geo-graphically closer to his current location,

and suggest venues such as the Old Stone house, Botanical garden and Brooklyn bridge. When compared to the POIs in Manhattan, although these locations are not extremely popular, they are well-rated and more importantly they match the topical interests of the user. Another option is to recommend the set of POIs in route 2. Contrary to the sequence in route 1, the venues in route 2 are extremely popular, but they do not exactly match with the user's topical interest. This is just one scenario; as we can see, there are multiple travel routes. Nonetheless, the POIs suggested by a good recommendation system should be a blend of geographical distance, personal choice of the user, social preference of the user's community and popularity of the venues. To achieve this, in

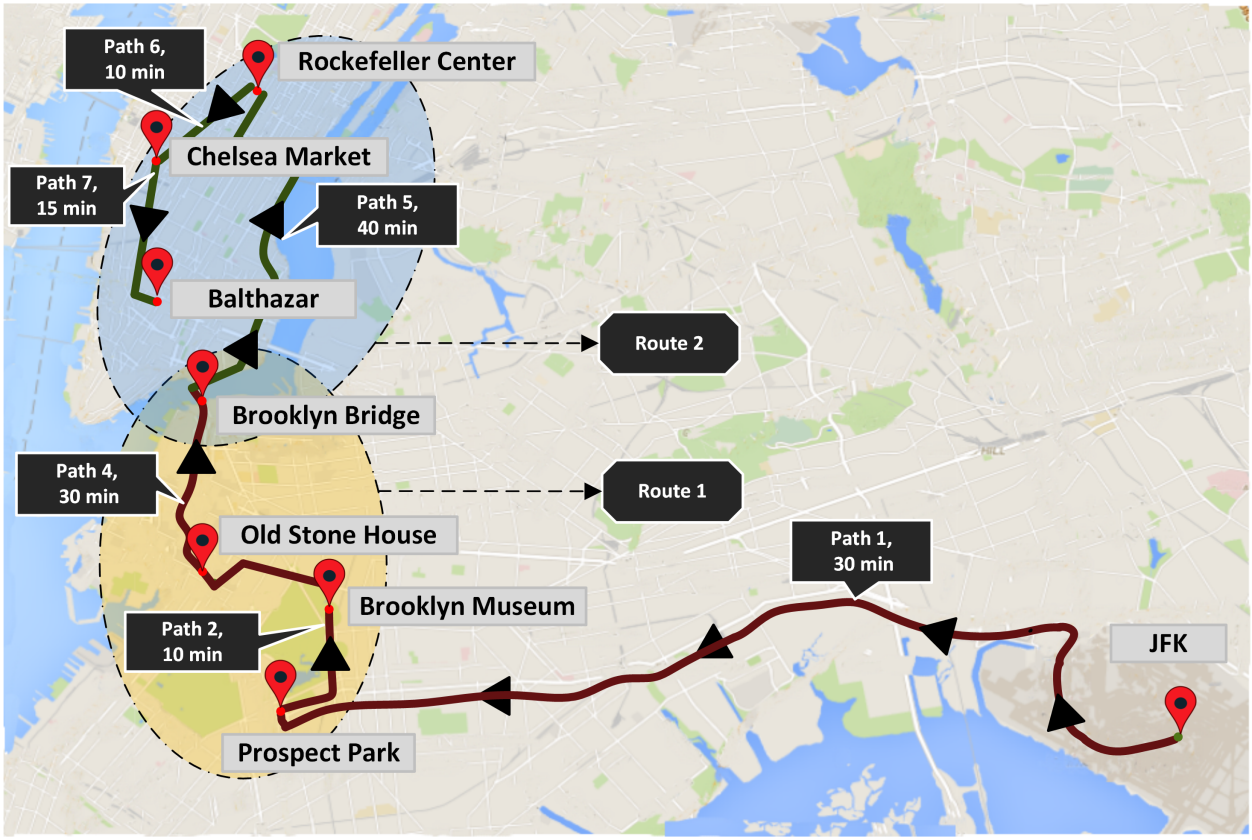


Figure 6.1: Travel pattern of a tourist who is interested in historical sites and nature.

this chapter, we propose a social sequential tour recommendation model, abbreviated as **SSTREC**, that aims at providing personalized POI recommendations for travelers. Inspired by several state-of-the-art generative models [3, 116, 117], we create a probabilistic framework that incorporates a multitude of features such as (a) the topical preference of users from their historical check-ins,

(b) their friend's choice from the social network of travelers, (c) the sequential visiting patterns of travelers, and (d) the trending popularity of venues, into a unified supervised topic model to make effective suggestions of POI sequences.

The rest of this chapter is structured as follows. We begin by reviewing the related works on location recommendation in Section 6.2 and provide some statistical insights about the behavior of travelers in Section 6.3. In Section 6.4, we introduce the proposed `SSTREC` model and the generative process. The description of the model is followed by the details of Gibbs sampling and the derivation of parameters in Section 6.4.2. We also explain the algorithm for creating POI sequences in this section. The data collection methodology and the results of our experiments are discussed in Section 6.6. Finally, we conclude this chapter in Section 6.7.

6.2 Related Work

Research on location recommendation can be broadly classified into the following three categories: (a) simple POI recommendation that aims at suggesting *individual* and *independent* landmarks, (b) travel package recommendation and (c) tour recommendation.

Recommending Independent POIs: There are numerous works on recommending independent venues. Matrix factorization techniques to recommend POIs in LSBNs are proposed in [118–120], while [118] incorporates temporal properties into these models. In [121], the authors propose a power-law probabilistic model, [122] formulates the probability of a user's check-in as a Multi-center Gaussian model, and [123] integrates user preference and location into a Bayesian learning model. The authors of [124] and [125] incorporate contextual information into a topic modeling based framework, while [126] proposed a hybrid matrix factorization model to incorporate sentiments. A more recent work on location recommendation addresses the cold start problem by viewing non-visited locations as non-negative samples and proposes a content-aware collaborative filtering [127]. Lian *et al*. [128] addresses the same problem by viewing mobility records as implicit feedback and leverages them as weighted matrix factorization. The authors of [129–131] adopt a different methodology of suggesting location by segmenting geographical areas into sub-regions based on the characteristics of POIs. Topic models incorporating geographical and social informa-

tion have also been shown to be effective for other tasks, such as opinion mining [132] and social media information retrieval [133–135].

Travel package recommendation: utilizes the geo-location information of travelers to recommend vacation packages such as a combined package of rental car plus hotel stay or flight travel hotels and local transportation [136, 137]. Nonetheless, this body of work is different from ours, since their goal is focused on creating combination of attractive packages that might draw the attention of travelers. On the other hand, our goal is to recommend a combination (more specifically a sequence) of points of interest for travelers. Although [138] and [139] consider the sequential pattern of POI visits, they do not factor in other important features, such as popularity of locations and social networks of users. Additionally, these methods recommend only single POI rather than their sequence.

Travel Route Recommendation: Unlike the above mentioned body of works, travel route recommendation is an emerging area, where most published papers are relatively new [140–144]. In [142], the authors adopt a collaborative retrieval model that incorporates pairwise weighted approximate rank function, while [143] proposes a pairwise tensor factorization-based framework that models user-POI, POI-time, and POI-POI interactions for successive POI recommendation. The authors of [145] model the interests of travelers using the popular HITS algorithm. By utilizing GPS logs from mobile devices various travel sequences are suggested for the users. Goinis et al. [146] adopted a time-aware tour recommendation framework that optimizes travel routes based on the best visiting times of POIs and [147] proposed algorithms that incorporate various constraints, such as variety of venues, budget constraints of users and the satisfaction provided by the POIs for recommendation. In a recent work, Wen et al. [148] incorporate the semantics of keywords from user queries in a skyline travel route framework for creating sequential POIs.

Despite being novel, the recommendation frameworks proposed in these works provide a very low degree of personalization. Some consider time-dependent factors, but ignore the topical preference of users; some capture user-level features, but do not incorporate temporal or sequential visiting patterns. In summary, our research is uniquely different from the above mentioned works

because of the following reasons: (1) we propose a novel topic-model based approach that incorporates the temporal quality of sequential visits, influence from social network of the user, the topical preference of users and the popularity of POIs. (2) using the proposed generative model, we recommend a series of travel sequences that will interest tourists. The paper that is closest to our work [149] uses a combination of a topic model and a Markov model to recommend sequences of venues. Therefore, in this work, we treat this model as the state-of-the-art tour recommendation system and compare it with the proposed SSTREC recommendation framework.

6.3 Learning Traveler Behavior

Before designing a recommendation model for travelers, it is important to understand their check-in behavior. In other words, we try to answer the question "What motivates travelers to visit a POI?" from four different perspectives namely: (a) sequential travel patterns, (b) topical interest of users, (c) impact of users' social circle and (d) popularity of POIs.

Impact of Travel Sequence: One of the main goals of this research is to incorporate Markovian relationships between POIs into our recommendation model. Therefore, the first step is to understand the nature of decision making of travelers. In particular, we are interested in determining whether they follow sequential patterns when traveling or visit POIs randomly. To answer this question, we employ hypothesis testing and perform the following steps: (1) obtain the *global travel pattern* by calculating conditional probabilities of traveling from the source venue X to target venue Y for all POI combinations; (2) for each user, obtain the top 10 ranked global travel patterns that correspond to his POI visits. If the user has pursued at least 50% of the POI sequences from this global pattern, categorize him as *followed*, if not, categorize as *not followed*; (3) randomly sample 100 travelers for 1,000 iterations and count the number of users who followed and those who did not. The result of this experiment is shown in Figure 6.2(a), where we notice that the median number of users who adhere to a travel pattern are higher than those who do not. To test the significance of this result, we set our null hypothesis H_0 as: "the average number of users who follow the sequence is same as those who don't". By applying a two-sample t-test,

the null hypothesis was rejected with a significance value of 2.2×10^{-16} thereby concluding that the *majority of travelers visit POIs sequentially.*

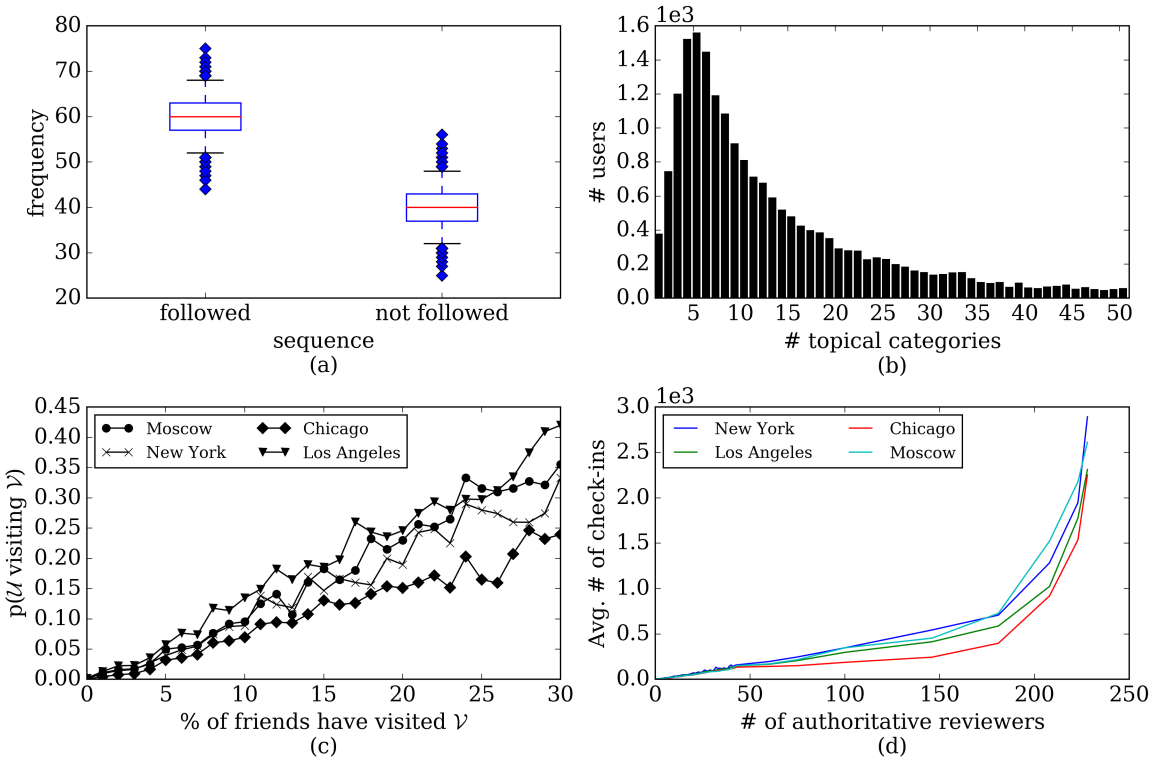


Figure 6.2: Behavior of travelers: (a) Variation in the distribution of follower versus non-followers; (b) Topical composition of tourists; (c) Impact of social circle on check-in behavior of tourists; (d) Influence of prominent reviewers over user check-ins.

Topical Composition: The decision to visit a landmark certainly depends on the topical interest of travelers. Nonetheless, to understand the topical variability of user interests, we extract the POI categories from their check-in history and plot the histogram of the topical compositions in Figure 6.2(b), where the x-axis is the number of unique topical categories liked by the travelers. In general, we see that the travelers are interested in multiple topical categories of POIs. However, a majority of them are restricted to about 5-10 categories and people who are interested in over 30 topical categories are extremely few in number.

Impact from the Social Circle: For every traveler $u \in U$ in our dataset, we obtain their list of friends F_u using the Foursquare API[1]. We then calculate the percentage of friends who have visited a

[1]https://developer.foursquare.com/

location $v \in V$, and the probability of this user *u* to visit v. The outcome of this analysis is depicted in Figure 6.2(c), which shows that the probability of a user visiting a POI increases as the cardinality of set F_u get larger. In other words, the *social circle of a user plays an important role in influencing the check-in habits of travelers.*

Presence of Prominent Reviewers: Reviewers play a critical role in attracting tourists; therefore, we also investigate whether check-ins are influenced by POI reviews by authoritative reviewers. Consequently, for every POI, we calculate the number of authoritative users based on the upvotes of their tips (or reviews) and plot it against the number of check-ins in Figure 6.2(d). This plot indicates that POIs having large number of such authoritative users have the potential to attract many tourists.

6.4 The SSTREC Model

In this section, we introduce SSTREC, a probabilistic generative model for recommending POIs for travelers. Our model is designed to capture the following behavioral traits of tourists: (1) traveling habits of users exhibit a strong sequential pattern, where the selection of a POI is dependent on the previously visited POI. (2) topical interests of users are strongly dependent on their level of relationship to their friend's circle. (3) the interest of users is confined to a limited set of topical categories, which can be obtained from their history of POI visits. (4) the choice of POIs are heavily dependent on their popularity.

Problem Statement: Given a set of *POIs* $V = \{v_1, v_2, ..., v_{|V|}\}$, a set of *travelers* $U = \{u_1, u_2, ..., u_{|U|}\}$, the goal of the proposed SSTREC model is to recommend a ranked list of V *sequential* POIs to a target (or new) traveler $\widetilde{u}$.

6.4.1 Generative Process

The behavior of a traveler is presented as a graphical model (SSTREC) in Figure 6.3. We describe the generative process of our model as follows:

- A traveler u can decide to visit a venue v based on his own decision or based on the decision of his friends F_u. This is determined by the distribution of social correlations ϕ^{UF} between

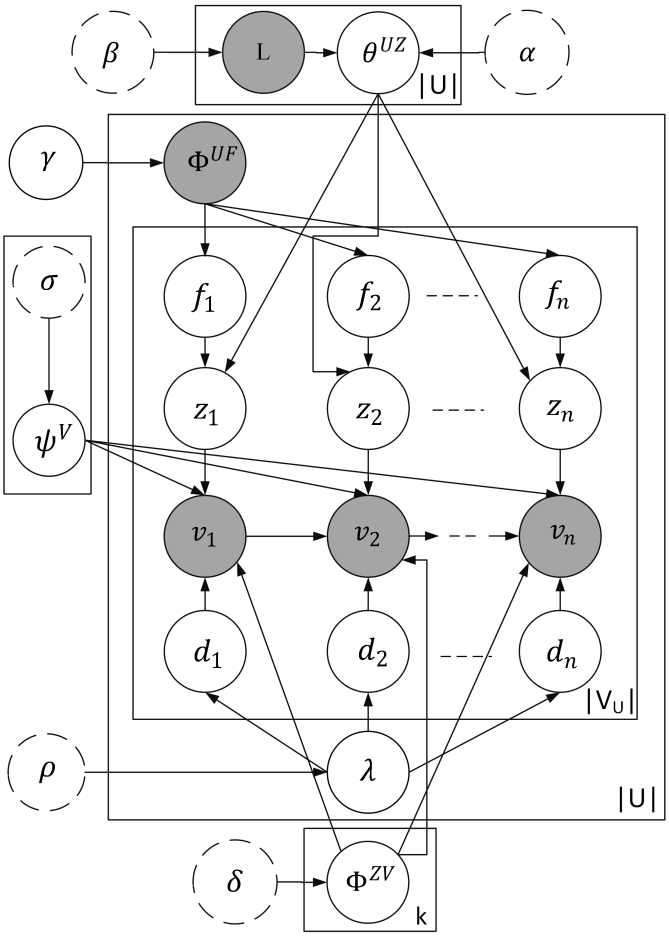


Figure 6.3: Plate diagram for generative process of SSTREC.

this user and his friends. Here, ϕ^{UF} is a multinomial distribution with symmetric Dirichlet prior, i.e., $\phi^{UF} \sim Dirichlet(\gamma)$.

- Based on the distribution of social correlations, the user chooses a friend $f_{j,i}$ and selects a POI i by first choosing its category (or topic) z_i. If the social correlation between the user and his friends is high, z_i is drawn from the topical distribution of his friends θ_f^{UZ}; if the correlation is low, z_i selected from his own distribution θ_u^{UZ}.
- The topical preference of the travelers (obtained from their history of POI visits), is incorporated in the form of supervised labels L, which acts as a constraint over the topic distribution θ^{UZ}. The label L^u of a user u follows a Bernoulli distribution with beta prior, i.e., $L^u \sim Bernoulli(\beta)$.

- A traveler can select a POI in two different ways: (a) based on his (or friend's) topical interest or (b) by simply choosing a POI purely based on it's popularity. In our model, this decision is governed by the variable d_i that takes a binary value 0 or 1.

- If d is 0:
  - The user chooses the venue based on the popularity distribution $\psi^V \sim Dirichlet(\sigma)$.
- If d is 1:
  - The user chooses the venue v_i based on (1.) the distribution over the previous POI v_{i-1} and (2.) the topical distribution of POIs, which is a multinomial ϕ^{ZV}.

Algorithm 4: Generative process of SSTREC

for *each POI* $v \in \mathcal{V}$ **do**
  Draw $\psi^v \sim Dirichlet(\sigma)$
end
for *each topic* $z_k, k \in \mathcal{K}$ **do**
  Draw $\phi^{ZV} \sim Dirichlet(\delta)$
end
for *each tourist* $u \in \mathcal{U}$ **do**
  for *each topic* $z_k, k \in \mathcal{K}$ **do**
    Draw $L^{(u)} \in \{0, 1\} \sim Bernoulli(\beta)$
  end
  Initialize $\alpha^{(u)} \leftarrow L \times \alpha$
  Draw $\lambda \sim Beta(\rho)$
  Draw $\theta^{(u)} \sim Dirichlet(\alpha)$
  for *each position* i *of POI* v*, in sequence* V^u **do**
    Draw $f \sim Multinomial(\phi^{UF})$
    Draw switch $d \sim Bernoulli(\lambda_u)$
    if *d = 0* **then**
      Draw $v \sim Multinomial(\psi^V)$
    end
    if *d = 1* **then**
      Draw $z \sim Multinomial(\theta^{(uf)})$
      Draw $v \sim Multinomial(\phi^{v_{i-1}z_i})$
    end
  end
end

6.4.2 Parameter Estimation

We adopt a collapsed Gibbs sampling for posterior inference of SSTREC parameters ($\phi^{UF}, \theta^{UZ}, \lambda, \phi^{ZV}$ and ψ^{V}). The posterior probability of our model is:

$$p(z, f, d|v, \beta, \alpha, \sigma, \delta, \rho, \lambda) = \frac{p(z, f, d, v|.)}{p(v|.)} \tag{6.1}$$

Direct multinomial relationship between the variables f and z creates complex inter-dependencies of two latent variables. To overcome this problem, we first estimate the social correlations ϕ^{UF} by using the traditional LDA topic model [44], where the observed words are the set of friends in our data, and the documents are the users. After this step, we treat ϕ^{UF} as an observed variable (denoted by the shaded circle in Figure 6.3).

Using the generative process, the total likelihood can be expanded as follows:

$$\begin{aligned} & p(f, z, d, v|.) \\ & = \int p(f|\phi^{UF}) p(\phi^{UF}|\gamma) d\phi^{UF} \cdot \int p(d|\lambda) p(\lambda|\rho) d\lambda \\ & \cdot \int p(z|f, \theta^{UZ}) p(\theta^{UZ}|\alpha, \mathcal{L}) d\theta^{UZ} \\ & \cdot \int \int p(v_i|v_{i-1}, d, z, \psi^{V}, \phi^{ZV}) p(\psi^{V}|\sigma) p(\phi^{ZV}|\delta) d\psi^{V} \phi^{ZV} \end{aligned} \tag{6.2}$$

According to Algorithm 4, the selection of POI by a user is based on two distinct choices, namely, popularity and personal choice. This allows to decompose the last multiplier in Equation (6.2) into a product of two independent components:

$$\underbrace{\int p(v_i^{(0)}|d, \psi^{V}) p(\psi^{V}|\sigma) d\sigma}_{\text{popularity}} \cdot \underbrace{\int p(v_i^{(1)}|v_{i-1}, d, z, \phi^{ZV}) p(\phi^{ZV}|\delta) d\delta}_{\text{personal choice}} \tag{6.3}$$

In the above equation, v_i^0 indicates the POI that is selected when $d = 0$ and v_i^1 corresponds to the POI chosen when $d = 1$. There are two important notes about Equation 6.2: first, the labeling prior β does not have a direct impact over the model; hence, it is not included in the equation. This is because, once the labels $\mathcal{L}$ are observed, β becomes d-separated [117]. Second, since we infer the

ϕ^{UF} using an independent LDA model, the distribution of friendship correlation is now observed, which makes the γ d-separated as well.

Since f is a multinomial that directly affects z, we first sample z for all combinations of f. Based on Equations (6.2) and (6.3) the posterior is given by:

$$
\begin{aligned}
& p(z_i = k, f_i = \mathfrak{f} | v^{(1)}, z_{\neg i}, f_{\neg i}) \qquad (6.4)\\
&= \frac{\int p(f|\phi^{UF}) p(\phi^{UF}|\gamma) d\gamma}{\int p(f_{\neg j}|\phi^{UF}) p(\phi^{UF}|\gamma) d\gamma} \cdot \frac{\int p(z|f, \theta^{UZ}) p(\theta^{UZ}|\alpha) d\theta^{UZ}}{\int p(z_{\neg i}|f, \theta^{UZ}) p(\theta^{UZ}|\alpha) d\theta^{UZ}} \\
&\cdot \frac{\int p(v_i^1|v_{i-1}, d, z, \phi^{ZV}) p(\phi^{ZV}|\beta) d\phi^{ZV}}{\int p(v_i^1|v_{i-1}, d, z_{\neg i}, \phi^{ZV}) p(\phi^{ZV}|\beta) d\phi^{ZV}} \\
&\propto \frac{n^{UF}_{u,f,\neg i} + \gamma_{fu}}{\sum_{f'}^{F} (n^{UF}_{u,f',\neg i} + \gamma_{f'u})} \cdot \frac{n^{ZF}_{z,f,\neg i} + \alpha_z}{\sum_{z=k}^{K} (n^{ZF}_{z,f,\neg i} + \alpha_z) - 1} \cdot \frac{n^{ZV}_{z_i, v_i|v_{i-1}, \neg i} + \delta_{v_i|v_{i-1}}}{\sum_{v'}^{V} (n^{ZV}_{z_i, v', \neg i} + \delta_{v'}) - 1}
\end{aligned}
$$

Once we sample the topics z for all combinations of friends, we then sample for the decision variable $d = 1$ and $d = 0$ as follows:

$$
\begin{aligned}
& p(d_i = 1 | d_{\neg i}, z, v, f) = \qquad (6.5)\\
& \frac{\int p(d|\lambda) p(\lambda|\rho) d\lambda}{\int p(d_{\neg i}) p(\lambda|\rho) d\lambda} \cdot \frac{\int p(v_i^{(1)}|v_{i-1}, d, z, \phi^{ZV}) p(\phi^{ZV}|\delta) d\phi^{ZV}}{\int p(v_i^{(1)}|v_{i-1}, d_{\neg i}, z, \phi^{ZV}) p(\phi^{ZV}|\delta) d\phi^{ZV}} \\
& \propto \frac{n^{UD}_{u,(1),\neg i} + \rho}{n^{UD}_{u,(1),\neg i} + n^{UD}_{u,(0),\neg i} + 2\rho - 1} \cdot \frac{n^{ZV}_{z_i, v_i|v_{i-1}, \neg i} + \delta_v}{\sum_{v'}^{V} (n^{ZV}_{z_i, v', \neg i} + \delta_{v'}) - 1}
\end{aligned}
$$

$$
\begin{aligned}
& p(d_i = 0 | d_{\neg i}, z, v, f) \qquad (6.6)\\
& \propto \frac{n^{UD}_{u,(0),\neg i} + \rho}{n^{UD}_{u,(0),\neg i} + n^{UD}_{u,(1),\neg i} + 2\rho - 1} \cdot \frac{n^{V}_{v,\neg i} + \sigma_v}{\sum_{v'}^{V} (n^{V}_{v',\neg i} + \sigma_{v'}) - 1}
\end{aligned}
$$

Once the topics z and decision variables d are sampled, the estimated parameters of the model $\hat{\theta}^{UZ}$, $\hat{\lambda}$, $\hat{\phi}^{ZV}$, and $\hat{\psi}^{V}$ can be derived by normalizing the counts n^{NF}, n^{UD}, n^{ZV}, and n^{V}, respectively.

Recommending POIs: Given a traveler $\widetilde{u}$, and his current venue v_{i-1} the recommendation score of an unseen (or next) POI $\widetilde{v}_i$ is calculated as:

$$p(\widetilde{v'}_i|v_{i-1},\widetilde{u}) = \sum_{z\in K} \hat{\theta}^{UZ}_{\widetilde{u},z} \cdot \hat{\phi}^{ZV}_{z,\widetilde{v'}_i|v_{i-1}} \cdot \hat{\lambda}_{\widetilde{u}} + (1-\hat{\lambda}_{\widetilde{u}}) \cdot \hat{\psi}^{V}_{v} \tag{6.7}$$

One can observe that Equation (6.7) supports the set of all behavioral traits of travelers that were outlined in Section 6.3. First, parameter $\theta^{\hat{UZ}}$ captures the favorite categories of travelers, $\phi^{\hat{ZV}}{}_{v_i|v_{i-1}}$ captures the sequential relationship between the POIs, and $\hat{\psi}^{V}$ captures the popularity-based POI preference. It is important to note that the distribution $\theta^{\hat{UZ}}$ also encompasses the social component of our model, since according to our generative model, the topical space of users is constrained by the selection of friends.

Table 6.1: List of notations used in this chapter.

Symbol	Description
$\mathcal{V} = \{v_i\}$	set of POIs, v_i indicates a single POI
$U = \{u_i\}$	set of travelers, u_i indicates a single traveler
$F = \{f_j\}$	set of friends of users, where $F \subset U$
D	binary decision variable, representing d=1 or d=0
$Z = \{z_i\}$	set of latent topics
K	number of topics
L	set of observed categories from the history of travelers
θ^{UZ}	topic distribution of travelers
ϕ^{UF}	distribution of social correlation between U and F
ϕ^{ZV}	topical distribution of POIs
λ	$U \times D$ social circle-popularity preference matrix
ψ^{V}	popularity distribution of POIs
α, β, γ	hyper-parameters of Dirichlet priors for θ^{UZ} , ϕ^{UF}
ρ, δ, σ	hyper-parameters of Dirichlet priors for $\lambda, \phi^{ZV}, \psi^{V}$
$n^{UF}_{u,f}$	# times user u preferred friend f's choice
$n^{ZU}_{z,f}$	# times a user (or his friend) preferred topic z
$n^{ZV}_{z_i,v_i\|v_{i-1}}$	# times POI v_i is assigned to topic z_i given v_{i-1}
$n^{UD}_{u,d}$	# times decision d is picked by user u
n^{V}_{v}	# times POI v is picked only based on popularity

6.4.3 Incorporating Prior Information

Our proposed model is a semi-supervised generative framework, which allows us to incorporate traveler and POI-specific features as priors.

Prior Information for Travelers: the traveler-based features are two-fold: (a) POI categories from the historical visits of users, which form the supervised topical labels $\mathcal{L}$ and (b) strength of friendship, which correspond to the prior information for the distribution ϕ^{UF}. The prior β for the label $\mathcal{L}$ is defined as follows:

$$\beta(u, c) = \frac{\text{\# times user } u \text{ visits POI with category } c}{\text{total POI visits by } u}$$

The priors for social relationship γ is determined using three different features, namely (1) the presence of a friendship link between users, (2) the number of overlapping topical categories (obtained from historical POI visits) and (3) the number of *lists* that are commonly followed by the users. In Foursquare, lists are like folders which enable the users to organize POIs that share similar characteristics in the form of geographical proximity or topical categories. This feature is extremely useful since it expresses the explicit action of users' interest over a collection of POIs. Consequently, the strength of friendship between users a and b is formulated as a linear combination of the above features as follows:

$$\gamma(a, b) = I(a, b)w + Cat(a, b) + \ell(a, b)$$

In the above equation, w is a weight factor, I is the indicator variable which denotes the presence of link between users a and b, Cat and ℓ denote the categorical and lists features respectively.

Prior Information for POIs: Popularity based features are introduced as prior σ into our model. We formulate the popularity score of a POI v as a function of the two features namely, the number of prominent reviewers for a POI $\mathcal{R}$ (explained in Section 6.3) and the total number of check-ins $\mathcal{M}$. Formally, this prior is calculated as $\sigma(v) = \mathcal{R}(v) + \mathcal{M}(v)$.

6.5 Creating POI Sequences

The ultimate goal of the proposed method is to recommend a series of POIs that can serve as travel itineraries to the users. Inspired by the algorithm proposed in [149], we assume that a

traveler will provide the following inputs to our model: (1) the current location, (2) the arrival time, (3) the number of route options, and (4) the *spare time*, which indicates the total time a traveler is willing to spend during his current trip. The procedure for route generation is shown in Algorithm 5, where the spare time is indicated by B, $\flat$ denotes buffer time, and K denotes the number of route options. The algorithm starts by inserting the start location of the user v_u^{start} to the priority queue Q as the very first sequence (a single POI is a special case of a sequence). It then generates K travel routes by performing the following set of operations. First, it pops the sequence with highest weight (i.e. the first POI v_u^{start} in this case) from the priority queue and checks if it meets our distance criteria (line 5). This distance d^r should be greater than the total travel time, which is the spare time plus the buffer time. If yes, then the algorithm acknowledges this as a route for recommendation (line 6); if not, it looks for alternative sequences of routes in lines 9-15, where it calculates the posterior according to the `SSTREC` model and the total distance for each new route in lines 11 and 12 and adds this new route along with other metadata to the priority queue. In this algorithm, $rv+$ denotes a new POI v being added to route r, $r[v_l]$ is the last visited venue in the route, p^{rv+} and d^{rv+} indicates the updated probability and distance for the sequence $rv+$ respectively. To reduce the number of POI combinations in our Gibbs sampling algorithm, we only consider POIs that meet the distance threshold.

6.6 Experimental Results

In this section, we report the results of comprehensive experimental evaluation of the proposed **SSTREC** model and compare it with other baselines and the state-of-the-art probabilistic model. We begin by discussing the details of our data collection methodology, which is then followed by the explanation of the evaluation metrics and the results of our experiments.

6.6.1 Dataset Description

For our experiments, we obtained the tweets of foursquare check-ins from the authors of [131], which spans from March-July 2014. We then augmented this raw data with a variety of POI and user-profile information by querying the Foursquare API. Information about POI includes textual description, rating, etc. Information about user includes friends, number of check-ins, and lists.

Algorithm 5: T-Route: Recommending POI sequences

Input: B,♭,K
Output: R

Initialize: k←0, A←array[ ], Q←*PriorityQueue*( )
Assign: Q← v_u^{start}
while $k \leq K$ **do**
  r←get sequence with highest probability from Q
  if $B - \flat \leq d^r \leq B + \flat$ **then**
    Insert r into R
    $k \leftarrow k + 1$
  end
  else if $d^r \leq (B + \flat)$ **then**
    for $v \in V_u$ **do**
      Set $p^{rv+} \leftarrow p(v|u, r[v_l])$ using Equation (6.7)
      Set $d^{rv^+} \leftarrow d_r +$ TravelTime$(r[v_l], v)$
      Insert tuple $< rv^+, p^{rv+}, d^{rv^+} >$ in Q
    end
  end
end

In total, we obtained 1,247,847 check-ins by 108,341 unique users and 170,472 venues (POIs) distributed over 20 cities. In this section, we perform our experiments over a subset of this data, which pertains to the top four cities based on the frequency of check-ins. The statistics of our dataset is shown in Table 6.2, where *Mn* denotes mean, *Md* the median, *Chk* indicates check-ins, *Frns* denotes friends and *Cats* corresponds to the categories (or topics) of POIs. Using this dataset as the base, we mimic a real-world scenario of travelers by taking every location, and creating three new cases based on the following conditions: (1) **Tourist dataset (D1):** Every user should have checked-in for at least 2 consecutive days and at most 6 days. The home location of the user should be different from the target city. For instance, if we are recommending POIs in Chicago, the user location should not be from Chicago. (2) **Localè dataset (D2):** We relax the constraint on consecutive check-ins; meaning, we do not care about whether a person has checked-in on consecutive days. However, for this case, the home location of the user should be the same as the target location. (3) **Social dataset (D3):** We remove constraints that was set for D1 and D2. Nonetheless, for this case, every user should have atleast 4 friends (social connection) with every

member who is a part of this dataset. It should be noted that the statistics shown in Table 6.2 vary for each of the above cases (i.e. D1-D3). Researchers can download the raw datasets used in our experiments from our public Github repository[2].

Table 6.2: Statistics of our Foursquare dataset.

City	#Users	#POI	Mn Chk	Md Chk	Mn Frns	Mn Cats
NY	1521	2076	10.75	8	10.04	17.14
Moscow	1132	1574	13.05	7	5.80	9.36
LA	794	976	8.51	6	9.37	13.45
Chicago	822	1079	8.9	6	7.4	13.36

6.6.2 Evaluation Metrics

The evaluation methodology for our model consists of two types; the first type is called the *uni-step* recommendation and the second type is termed as the *multi-step* recommendation. In the uni-step evaluation, we use the first n-1 visited POIs in the sequence for training and the last visited POI for testing. In multi-step evaluation, instead of removing only the last visited POI, we use the last 3 contiguous POI sequences for testing. We make sure that the testing POIs fall within the spare time plus the buffer time that was mentioned in Section 6.5. Our evaluation is performed over all cases of the dataset, namely, Tourist (D1), Locale (D2) and Social (D3). The SSTREC model and all other baselines are implemented using Python's numpy numerical module, and Scikit-Learn machine learning module[3].

Evaluation Metrics

To evaluate the performance of ranking, we use the standard information retrieval measures. For every traveler, we compute: (1) P@N: *precision at rank N* is the fraction of POIs that were actually visited by users in the top-N ranked POI instances, (2) R@N: *recall at rank N* is the fraction of the visited POIs that were retrieved at every top-N ranked POI instances, (3) S@N: The *success at rank N* is the probability of finding at least one truly visited POI in the *top-N*

[2]https://github.com/magnetpest2k5/WSDM17
[3]http://scikit-learn.org

ranked set, (4) DCG: The *discounted cumulative gain* is based on the fact that highly relevant POIs are more important than marginally relevant ones and (5) Edit distance: The minimum number of operations that are required to transform the recommended sequence of POIs to the sequence of POIs in the actual ground-truth (i.e., test set) of the user.

Baseline Methods for Comparison: We compare the performance of our proposed model with the following baselines:

- **Random Multinomial Choice (*RC*)** recommends a location by naively drawing a location v from a multinomial distribution of global location weights $v_i \sim Multi\{v_1, ..., v_n\}$. These weights are not user-specific and simply based on the number of check-ins for each location.
- **Markov Model (*MM*)** predicts the next visiting venue of the user using a Markov model that calculates the probability $p(v_t|v_{t-1})$, where v_t is the next (to be visited) landmark, v_{t-1} is the previously visited landmark. This model completely ignores the topical interests of users.
- **Photographer Behavior Model (*PBM*)** is a state-of-the-art topic model that uses a combination of Markov model and PLSA topic model [100] to recommend sequence of POIs [149].

6.6.3 Recommending POI Sequences

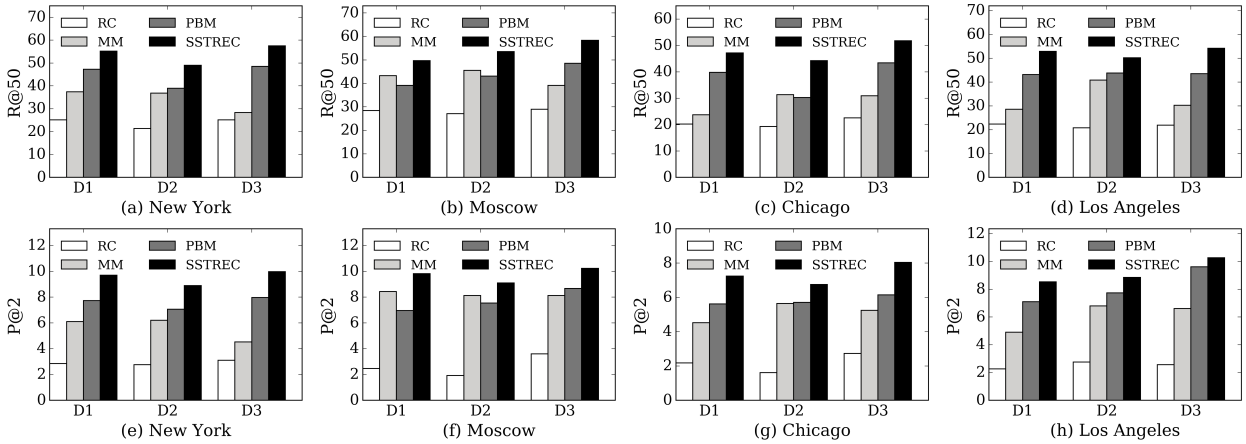


Figure 6.4: Performance comparison of SSTREC model: (a)-(d) shows the recall scores and (e)-(h) show the precision scores.

Single POI Recommendation We begin this section, by presenting the performance of the model in terms of recall in Figure 6.4. In general, the proposed SSTREC model outperforms other models on all dataset types, while the random multinomial choice (RC) has the worst performance. Al-

though PBM performs better than the Markov model (MM), it is important to note that the results are not consistent. For instance, in Figure 6.4(b), the performance of PBM is lower than MM on D1 and D2 datasets. On the other hand, the results of the proposed model are consistent throughout all the scenarios. A possible explanation for such inconsistency is that PBM relies on a naive combination of Markovian and the topic probability; this makes the topic space disjoint from the POI transition probability. Contrary to this, SSTREC learns the preference of POIs using a unified generative framework where the topic space of travelers and POIs are constrained on Markovian transition probabilities, popularity of POIs and categorical choice of users. The best performance of SSTREC (and all other models) is observed over the social dataset D3. This is mainly due to the nature of this dataset, where the presence of friendship links between the users results in many commonly visited POIs. Additionally, our model is able to leverage this social linkage to overcome sparsity and yield better results. The precision performance of the models shown in Figures 6.4(e)-(h) are similar to their recall counterparts with SSTREC outperforming other models by achieving a precision up to 10%. The PBM closely follows our model, but not for all scenarios; its poor consistency is yet again revealed in Figures 6.4(f) and 6.4(g), where MM outperforms all other models except SSTREC.

Quality of Ranked POIs: DCG is a classic performance measure that is used widely in evaluating information retrieval systems. In our setting, we use this measure to penalize incorrectly ranked POIs based on their positions. Unlike information retrieval, where documents are assigned different relevance levels, our data is binary (i.e. 1 if user visits a POI and 0 otherwise); consequently, we set a constant relevance score of 3 for all POIs. The comparison of DCG scores between SSTREC and PBM for top 50 ranked POIs is shown in Figures 6.5(a)-(d), where the x-axis denotes the topic size. We see that SSTREC performs better than PBM over both the data set types D1 and D3. It should be noted that SSTREC is a supervised model where the topic space corresponds to the number of unique POI categories. Therefore, the topic count for this model corresponds to the number of unique POI categories. The DCG scores of PBM reaches a saturation point at about 20 topics. Contrary to this, SSTREC performs better with more topics since this essentially translates

into more supervised information. In our experiments, we were able to see a steady improvement in DCG even beyond 100 topics, although the improvement was marginal. Originally, we have 320 unique categories of POIs from Foursquare.

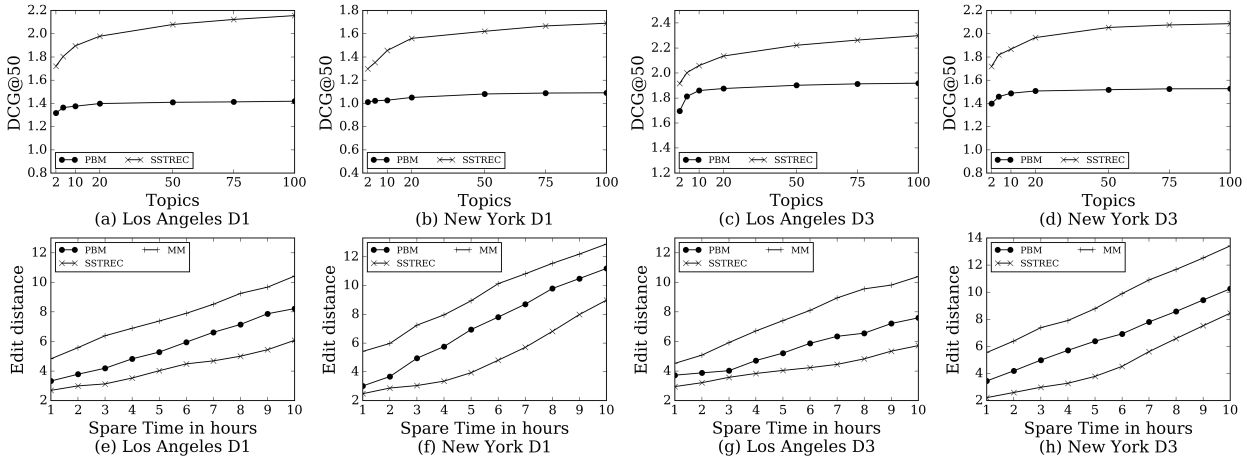


Figure 6.5: (a)-(d) shows the performance comparison of SSTREC model in terms of DCG. (e)-(f) shows the variation in the recommended POI sequences for travelers.

Multiple POI Recommendation

In this experiment, we focus on evaluating *the exact order of POI sequences* generated by Algorithm 5; i.e., a recommended sequence is deemed as a true positive instance, only if it matches with the exact order of the POIs sequentially visited by a user in the test data. The outcome of this experiment is depicted as edit distance in Figure 6.5, which is based on the following edit operations: insert into a sequence, delete from a sequence, and replace one POI with another. From the results, we observe that the proposed SSTREC model has the lowest edit distance among all models. As the spare time increases, the number of recommended venues increases as well, which in-turn increases the number of mismatches in the recommended sequences. Unlike the precision and recall scores, we did not find any major performance increase between datasets D1 and D3.

6.6.4 Visual Interface for Travelers

In this section, we present the qualitative results of our model using a visual interface shown in Figure 6.6. Due to space limitations, we restrict our example to just one user and one city. In this interface, the user provides the input city as New York, a spare time of 4 hours and the start location

as Union square (denoted by red star) and the number of route options as 3. Based on the topical interest of the user (not shown in the Figure, since it is background information) and the travel time between POIs, the interface shows 3 recommended sequences of POIs with varying travel times. The topics associated with the sequences are presented above them. For instance, route 1 consists of Chelsea market, 9/11 Memorial, Rockwood Music Hall that are associated with Historical sites, Music and Shopping. It can be seen that all routes that are recommended have historical site and music as common topical categories. Although there is some mismatch between topics and POI sequences, for most part they are coherent. In addition to the recommended route sequences, the figure also shows the topics and POIs from the user's social network, which is provided by leveraging the distribution of social correlations. It should be noted that the travel times indicated on the routes *are not the exact travel times*; instead, they correspond to a combination of travel and visiting time. For example, the travel time of 1.5 hours between POI 5 and 6 indicates the time taken to reach POI 6 from 5, plus the time to tour

POI 6.

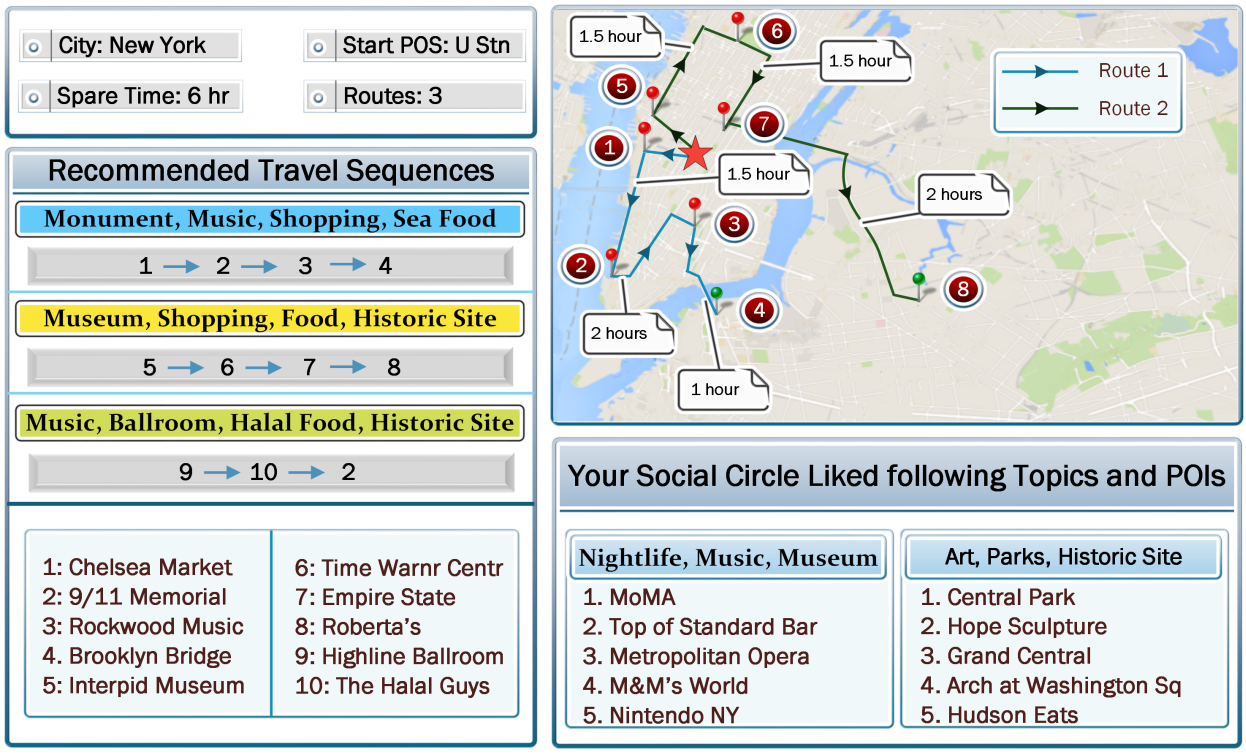


Figure 6.6: A visual example of travel routes recommended by the SSTREC model.

6.7 Summary

In this chapter, we developed a social sequential tour recommendation SSTREC model, which provides personalized POI recommendations for travelers. Using a novel generative approach, the proposed model utilizes diverse features, such as temporal sequences, social relations, topical preferences and popularity of POIs, to provide quality POI recommendations to travelers. The model was then extended using a best first search algorithm to recommend a sequence of POIs that could serve as travel itineraries. Using extensive set of experiments, and a rich dataset of Foursquare check-ins, we showed that our model outperforms a state-of-the-art probabilistic model in almost all scenarios.

CHAPTER 7: CONCLUSIONS AND FUTURE WORK

In this chapter, we summarize the major contributions of our research work. We started this Thesis with the review of the memory and the model based recommender systems in chapter 2. We then proposed a hybrid recommendation model called `ListRec` that leveraged the dynamically varying tweet content, the network of twitterers and the popularity of lists to collectively model the users preference towards social lists. We categorized the users into two types (a) persistent twitterers who tweet frequently and consistently and (b) active consumers who are characterized by a sparse tweet history, but they actively consume information from Twitter by following other users. For persistent twitterers, we obtained the user-topic vector and topic-list vector by running the dynamic topic model over the tweet content and measured the interest of users as a scalar product of these two vectors. For active consumers, we projected the user space into a followee space and utilized the followees list subscriptions to indirectly measure the interest of the users. We also added a trend based score that measures the popularity of lists in the Twitter domain. The final score was calculated as a linear combination of these three individual scores (based on content, network, and popularity). The coefficients in this linear combination was estimated using a cyclic ridge regression estimation approach. In Section 3.4 of chapter 2, we proposed the LIST-PAGERANK model to recommend auxiliary set of lists that are authoritative and topically similar to the lists that are subscribed by the twitterers. The main novelty of this model was the formulation of the list graph structure, where lists were considered as nodes and the edge between two lists exists if the member of a list is a subscriber of another list. Finally we used a variant of topic-specific PAGRERANK called the LIST-PAGERANK that leveraged the network structure of Twitter lists to recommend authoritative lists that match the topical interest of the users.

In Chapter 4, we explored the popular reward-based crowdfuding platform Kickstarter to understand "what set of features determine a project's success?". To answer this question, we scraped about 6 months of data consisting of 27K projects and 1 million users. The data included a variety of static and temporal features such as the duration of project, the goal amount, the number of Facebook shares, the number of backers, the number of updates about the project progress, progression

of rewards and funds, the geo-location, etc. By utilizing this exhaustive dataset, we were able to find several interesting aspects about the Kickstarter crowdfunding domain. We summarize our finding in the following points. (1) *Deadline Effect*: where we showed that campaigns in Kickstarter follow a U-shaped distribution of fund progression. (2) *Herding instinct*: which showed that the average time delay between the first 5 consecutive tweet determines the spread of promotions. (3) *Mutual Trust:* which explained that investors do not just randomly choose projects for backing; instead, they look for a long-term connection to the creator. (4) *Tie Strength and Structural Cohesion*: which explained that accumulation of backers is not only based on the number of promoters in Twitter, but also on the connectivity between these promoters and (5) *Influence of Prominent Promoters*: which revealed that projects promoted by influential twitter users have the potential to attract many backers. In the second part of this chapter, we formulated a binary classification/regression problem, where given a backer-project pair, the trained model computed the score for the likelihood of funding. Utilizing the gradient boosting tree, a state-of-the-art learner model, we achieved a practically useful level of performance up to 0.89 AUC (area under the curve) value. The extension to this work was discussed in chapter 5 in the form of Group Recommendation model for crowdfunding domain, which unlike the conventional recommendation, recommends projects to a group of investors. We proposed a recommendation model called CROWDREC that integrated the personal interest of users, the social group (or the community), and the real-time status of the project into a unified generative process to provide meaningful contextual recommendation for Kickstarter users. Our model was built on four key observations: (a) a crowdfunding group may support projects from multiple topical categories. (b) users backing decision is based not only on his personal preference but also on the collective preferences of his groups. (c) groups collective preference to support a project is strongly correlated with the personal preferences of topically authoritative users (i.e., users expertise) within the group. (d) the dynamic status of a project impacts both the individual investors personal preferences and the groups collective preferences in backing crowdfunding projects. To learn the parameters of the model, we adopted a two-step gibbs sampling procedure and the final recommendation was based on the weighted score of the groups

influence, the individual's preference and real-time trend of the project. The experimental results of our model revealed the following conclusions: 1. Temporal progression of funds, backers, and tweet promotions have the strongest variable importance. 2. Backers strongly depend on their topical preferences (obtained from the backing history of the users) to fund a project. 3. The impact of social network monotonically decreases with the increase in backing frequency. 4. The influence of geo-location strongly depends on the topical category of the project. For instance, projects on games, comics, and technology are relatively less dependent on their geo-location, while projects on theater, food, and dance are highly dependent.

In Chapter 6, we focused on a new form of recommendation called the Tour Recommendation, where the objective is to suggest a sequence of point of interests (POIs) that will serve as travel itineraries to tourists. We explained various challenges associated with Tour Recommendation. First, most users are not native to their tour destination (i.e. users are tourists), which makes the check-in information of these users extremely sparse. Second, unlike conventional recommendation where POI suggestions are made in a disjoint manner, these temporal features play a critical role in determining the next check-in spot when suggesting sequence of POIs for travelers. To overcome the above challenges, we proposed a probabilistic social sequential model called SSTREC that incorporated a multitude of features such as (a) the topical preference of users from their historical check-ins, (b) friends choice from the social network of travelers, (c) the sequential visiting patterns of travelers and (d) the trending popularity of venues into a unified supervised topic model to make effective suggestions of POI sequences. Using extensive evaluation techniques, we showed that the model achieves an impressive recall performance over a wide array of datasets by seamlessly integrating collaborative and content-based recommendation. We also demonstrated a mock-up interface that provided different route options consisting of POI sequences that were not only optimized for geographical distance, but also for the user's topical interest, time constraints provided by the user and the real-time popularity of the POIs.

APPENDIX: LIST OF PUBLICATIONS

1. V. Rakesh, N. Jhadhav, A. Kotov, and C. K. Reddy. Probabilistic social sequential model for tour recommendation. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, 2017.
2. V. Rakesh, W.-C. Lee, and C. K. Reddy. Probabilistic group recommendation model for crowdfunding domains. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, pages 257-266. ACM, 2016.
3. Y. Li, V. Rakesh, and C. K. Reddy. Project success prediction in crowdfunding environments. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, pages 247-256. ACM, 2016.
4. V. Rakesh, J. Choo, and C. K. Reddy. Project recommendation using heterogeneous traits in crowdfunding. In Ninth International AAAI Conference on Web and Social Media, 2015.
5. A. Kotov, V. Rakesh, E. Agichtein, and C. K. Reddy. Geographical latent variable models for microblog retrieval. In European Conference on Information Retrieval, pages 635-647. Springer, 2015.
6. V. Rakesh, D. Singh, B. Vinzamuri, and C. K. Reddy. Personalized recommendation of twitter lists using content and network information.In Eighth International AAAI Conference on Web and Social Media, 2014.
7. V. Rakesh, C. K. Reddy, D. Singh, and M. Ramachandran. Location-specific tweet detection and topic summarization in twitter. In Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on, pages 1441-1444. IEEE, 2013.

REFERENCES

[1] Vineeth Rakesh, Dilpreet Singh, Bhanukiran Vinzamuri, and Chandan K Reddy. Personalized recommendation of twitter lists using content and network information. 2014.

[2] Vineeth Rakesh, Jaegul Choo, and Chandan K Reddy. Project recommendation using heterogeneous traits in crowdfunding. In *Ninth International AAAI Conference on Web and Social Media*, 2015.

[3] Vineeth Rakesh, Wang-Chien Lee, and Chandan K Reddy. Probabilistic group recommendation model for crowdfunding domains. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 257–266. ACM, 2016.

[4] Vineeth Rakesh, Niranjan Jadhav, Alexander Kotov, and Chandan K Reddy. Probabilistic social sequential model for tour recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 631–640. ACM, 2017.

[5] Iván Cantador, Alejandro Bellogín, and Pablo Castells. News@ hand: A semantic web approach to recommending news. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 279–283. Springer, 2008.

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

[7] Elke Michlmayr and Steve Cayzer. Learning user profiles from tagging data and leveraging them for personal (ized) information access. 2007.

[8] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.

[9] Franca Debole and Fabrizio Sebastiani. Supervised term weighting for automated text categorization. In *Text mining and its applications*, pages 81–97. Springer, 2004.

[10] Zhi-Hong Deng, Shi-Wei Tang, Dong-Qing Yang, Ming Zhang Li-Yu Li, and Kun-Qing Xie. A comparative study on feature weight in text categorization. In *Asia-Pacific Web Conference*, pages 588–597. Springer, 2004.

[11] Man Lan, Chew Lim Tan, and Hwee-Boon Low. Proposing a new term weighting scheme for text categorization. In *AAAI*, volume 6, pages 763–768, 2006.

[12] Man Lan, Chew Lim Tan, Jian Su, and Yue Lu. Supervised and traditional term weighting methods for automatic text categorization. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):721–735, 2009.

[13] Zhi-Hong Deng, Kun-Hu Luo, and Hong-Liang Yu. A study of supervised term weighting scheme for sentiment analysis. *Expert Systems with Applications*, 41(7):3506–3513, 2014.

[14] Yupeng Gu, Bo Zhao, David Hardtke, and Yizhou Sun. Learning global term weights for content-based recommender systems. In *Proceedings of the 25th International Conference on World Wide Web*, pages 391–400. International World Wide Web Conferences Steering Committee, 2016.

[15] Raymond J Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204. ACM, 2000.

[16] Harry Mak, Irena Koprinska, and Josiah Poon. Intimate: A web-based movie recommender using text categorization. In *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on*, pages 602–605. IEEE, 2003.

[17] Rajatish Mukherjee, Gerdur Jonsdottir, Sandip Sen, and Partha Sarathi. Movies2go: An online voting based movie recommender system. In *Proceedings of the fifth international conference on Autonomous agents*, pages 114–115. ACM, 2001.

[18] Beerud Sheth and Pattie Maes. Evolving agents for personalized information filtering. In *Artificial Intelligence for Applications, 1993. Proceedings., Ninth Conference on*, pages 345–352. IEEE, 1993.

[19] Jae-wook Ahn, Peter Brusilovsky, Jonathan Grady, Daqing He, and Sue Yeon Syn. Open user profiles for adaptive news systems: help or harm? In *Proceedings of the 16th international conference on World Wide Web*, pages 11–20. ACM, 2007.

[20] Marko Balabanović and Yoav Shohom. Content-based, callaborative recommendation. *Communications of the ACM*, 40(3), 1997.

[21] Yashar Deldjoo, Mehdi Elahi, Paolo Cremonesi, Franca Garzotto, Pietro Piazzolla, and Massimo Quadrana. Content-based video recommendation system based on stylistic visual features. *Journal on Data Semantics*, 5(2):99–113, 2016.

[22] Jonathan Wintrode, Gregory Sell, Aren Jansen, Michelle Fox, Daniel Garcia-Romero, and Alan McCree. Content-based recommender systems for spoken documents. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5201–5205. IEEE, 2015.

[23] Bo-Wen Zhang, Xu-Cheng Yin, Xiao-Ping Cui, Jiao Qu, Bin Geng, Fang Zhou, Li Song, and Hong-Wei Hao. Social book search reranking with generalized content-based filtering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 361–370. ACM, 2014.

[24] Ludovico Boratto and Salvatore Carta. Impact of content novelty on the accuracy of a group recommender system. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 159–170. Springer, 2014.

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

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

[27] Joshua Alspector, Aleksander Koicz, and Nachimuthu Karunanithi. Feature-based and clique-based user models for movie selection: A comparative study. *User Modeling and User-Adapted Interaction*, 7(4):279–304, 1997.

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

[29] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[30] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[31] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

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

[33] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

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

[35] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. Ieee, 2008.

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

[37] Saikishore Kalloori, Francesco Ricci, and Marko Tkalcic. Pairwise preferences based matrix factorization and nearest neighbor recommendation techniques. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 143–146. ACM, 2016.

[38] Robin Devooght, Nicolas Kourtellis, and Amin Mantrach. Dynamic matrix factorization with priors on unknown values. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 189–198. ACM, 2015.

[39] Maksims Volkovs and Guang Wei Yu. Effective latent models for binary feedback in recommender systems. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 313–322. ACM, 2015.

[40] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Nips*, volume 1, pages 2–1, 2007.

[41] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[42] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[43] Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. In *IJCAI*, volume 99, pages 688–693, 1999.

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

[45] Gregor Heinrich. Parameter estimation for text analysis. *University of Leipzig, Tech. Rep*, 2008.

[46] Michael S Bernstein, Bongwon Suh, Lichan Hong, Jilin Chen, Sanjay Kairam, and Ed H Chi. Eddi: interactive topic-based browsing of social status streams. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 303–312. ACM, 2010.

[47] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web. In *Proceedings of the 3rd International Web Science Conference*, page 2. ACM, 2011.

[48] Marcelo G Armentano, Daniela Godoy, and Analía Amandi. Topology-based recommendation of users in micro-blogging communities. *Journal of Computer Science and Technology*, 27(3):624–634, 2012.

[49] John Hannon, Mike Bennett, and Barry Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 199–206. ACM, 2010.

[50] Jeon Hyung Kang and Kristina Lerman. Using lists to measure homophily on twitter. In *AAAI workshop on Intelligent techniques for web personalization and recommendation*, 2012.

[51] Dongwoo Kim, Yohan Jo, and Il-Chul Moon. Analysis of twitter lists as a potential source for discovering latent characteristics of users. In *CHI 2010 Workshop on Microblogging: What and How Can We Learn From It?* Citeseer, 2010.

[52] Wei Feng and Jianyong Wang. Retweet or not?: personalized tweet re-ranking. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 577–586. ACM, 2013.

[53] Daniel Ramage, Susan T Dumais, and Daniel J Liebling. Characterizing microblogs with topic models. In *ICWSM*, pages 130–137, 2010.

[54] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, pages 91–100. ACM, 2008.

[55] Jilin Chen, Rowan Nairn, Les Nelson, Michael Bernstein, and Ed Chi. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1185–1194. ACM, 2010.

[56] Vineeth Rakesh, Chandan K Reddy, and Dilpreet Singh. Location-specific tweet detection and topic summarization in twitter. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1441–1444. ACM, 2013.

[57] John Hannon, Kevin McCarthy, and Barry Smyth. Finding useful users on twitter: twittomender the followee recommender. In *Advances in Information Retrieval*, pages 784–787. Springer, 2011.

[58] Matthew Burgess, Alessandra Mazzia, Eytan Adar, and Michael Cafarella. Leveraging noisy lists for social feed ranking. In *Seventh International AAAI Conference on Weblogs and Social Media*. ICWSM, 2013.

[59] Yuto Yamaguchi, Toshiyuki Amagasa, and Hiroyuki Kitagawa. Tag-based user topic discovery using twitter lists. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 13–20. IEEE, 2011.

[60] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270. ACM, 2010.

[61] Gianmarco De Francisci Morales, Aristides Gionis, and Claudio Lucchese. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 153–162. ACM, 2012.

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

[63] Liangjie Hong and Brian D Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88. ACM, 2010.

[64] Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 889–892. ACM, 2013.

[65] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[66] Amit Manjhi, Vladislav Shkapenyuk, Kedar Dhamdhere, and Christopher Olston. Finding (recently) frequent items in distributed data streams. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 767–778. IEEE, 2005.

[67] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[68] Jonathan Morduch. The microfinance promise. *Journal of economic literature*, pages 1569–1614, 1999.

[69] KickStarterStats. Kickstarter stats. *Kickstarter* https://www.kickstarter.com/help/stats. accessed 07-23-2014.

[70] Venkat Kuppuswamy and Barry L Bayus. Crowdfunding creative ideas: The dynamics of project backers in kickstarter. *UNC Kenan-Flagler Research Paper*, (2013-15), 2014.

[71] Jisun An, Daniele Quercia, and Jon Crowcroft. Recommending investors for crowdfunding projects. In *Proceedings of the 23rd international conference on World wide web*, pages 261–270. International World Wide Web Conferences Steering Committee, 2014.

[72] Anbang Xu, Xiao Yang, Huaming Rao, Wai-Tat Fu, Shih-Wen Huang, and Brian P Bailey. Show me the money!: an analysis of project updates during crowdfunding campaigns. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 591–600. ACM, 2014.

[73] Vincent Etter, Matthias Grossglauser, and Patrick Thiran. Launch hard or go home!: predicting the success of kickstarter campaigns. In *Proceedings of the first ACM conference on Online social networks*, pages 177–182. ACM, 2013.

[74] Ethan Mollick. The dynamics of crowdfunding: An exploratory study. *Journal of Business Venturing*, 29(1):1–16, 2014.

[75] Elizabeth M Gerber, Julie S Hui, and Pei-Yi Kuo. Crowdfunding: Why people are motivated to post and fund projects on crowdfunding platforms. In *CSCW Workshop*, 2012.

[76] Julie S Hui, Michael D Greenberg, and Elizabeth M Gerber. Understanding the role of community in crowdfunding work. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 62–74. ACM, 2014.

[77] Tanushree Mitra and Eric Gilbert. The language that gets people to give: Phrases that predict success on kickstarter. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 49–61. ACM, 2014.

[78] Chun-Ta Lu, Sihong Xie, Xiangnan Kong, and Philip S Yu. Inferring the impacts of social media on crowdfunding. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 573–582. ACM, 2014.

[79] Jaegul Choo, Changhyun Lee, Daniel Lee, Hongyuan Zha, and Haesun Park. Understanding and promoting micro-finance activities in kiva. org. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 583–592. ACM, 2014.

[80] Jaegul Choo, Daniel Lee, Bistra Dilkina, Hongyuan Zha, and Haesun Park. To gather together for a better world: understanding and leveraging communities in micro-lending recommendation. In *Proceedings of the 23rd international conference on World wide web*, pages 249–260, 2014.

[81] Tillman Bruett. Cows, kiva, and prosper. com: How disintermediation and the internet are changing microfinance. *Community Development Investment Review*, 3(2):44–50, 2007.

[82] James Andreoni. Impure altruism and donations to public goods: a theory of warm-glow giving. *The economic journal*, pages 464–477, 1990.

[83] Arvind Ashta and Djamchid Assadi. Do social cause and social technology meet? impact of web 2.0 technologies on peer-to-peer lending transactions. *Cahiers du CEREN*, 29:177–192, 2009.

[84] Eugene Webb and Karl E Weick. Unobtrusive measures in organizational theory: A reminder. *Administrative Science Quarterly*, pages 650–659, 1979.

[85] Venkat Kuppuswamy and Barry L Bayus. Crowdfunding creative ideas: The dynamics of project backers in kickstarter. *SSRN Electronic Journal*, 2013.

[86] Richard Nadeau, Edouard Cloutier, and J-H Guay. New evidence about the existence of a bandwagon effect in the opinion formation process. *International Political Science Review*, 14(2):203–213, 1993.

[87] Michael J Welch, Uri Schonfeld, Dan He, and Junghoo Cho. Topical semantics of twitter links. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 327–336. ACM, 2011.

[88] Sitaram Asur, Bernardo A Huberman, Gabor Szabo, and Chunyan Wang. Trends in social media: persistence and decay. In *ICWSM*, 2011.

[89] Eytan Bakshy, Jake M Hofman, Winter A Mason, and Duncan J Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 65–74. ACM, 2011.

[90] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[91] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[92] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[93] Leo Breiman. *Classification and regression trees*. CRC press, 1993.

[94] Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[95] Yan Li, Vineeth Rakesh, and Chandan K Reddy. Project success prediction in crowdfunding environments. In *In Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, 2016.

[96] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2):177–196, 2001.

[97] Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu. Collaborative personalized tweet recommendation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 661–670. ACM, 2012.

[98] Benjamin Marlin. *Collaborative filtering: A machine learning perspective*. PhD thesis, University of Toronto, 2004.

[99] Adam Sadilek, Henry Kautz, and Jeffrey P Bigham. Finding your friends and following them to where you are. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 723–732. ACM, 2012.

[100] Alexandrin Popescul, David M Pennock, and Steve Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 437–444. Morgan Kaufmann Publishers Inc., 2001.

[101] Wen-Yen Chen, Dong Zhang, and Edward Y Chang. Combinational collaborative filtering for personalized community recommendation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 115–123. ACM, 2008.

[102] Mao Ye, Xingjie Liu, and Wang-Chien Lee. Exploring social influence for recommendation: a generative model approach. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 671–680. ACM, 2012.

[103] Wen-Yen Chen, Jon-Chyuan Chu, Junyi Luan, Hongjie Bai, Yi Wang, and Edward Y Chang. Collaborative filtering for orkut communities: discovery of user latent behavior. In *Proceedings of the 18th international conference on World wide web*, pages 681–690. ACM, 2009.

[104] Vishvas Vasuki, Nagarajan Natarajan, Zhengdong Lu, and Inderjit S Dhillon. Affiliation recommendation using auxiliary networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 103–110. ACM, 2010.

[105] Jingdong Wang, Zhe Zhao, Jiazhen Zhou, Hao Wang, Bin Cui, and Guojun Qi. Recommending flickr groups with social topic model. *Information retrieval*, 15(3-4):278–295, 2012.

[106] Wei Zhang, Jianyong Wang, and Wei Feng. Combining latent factor model with location features for event-based group recommendation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 910–918. ACM, 2013.

[107] Xingjie Liu, Yuan Tian, Mao Ye, and Wang-Chien Lee. Exploring personal impact for group recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 674–683. ACM, 2012.

[108] Joseph F McCarthy. Pocket restaurantfinder: A situated recommender system for groups. In *Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems*, 2002.

[109] Quan Yuan, Gao Cong, and Chin-Yew Lin. Com: a generative model for group recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 163–172. ACM, 2014.

[110] Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 119–126, 2010.

[111] David Blackwell and James B MacQueen. Ferguson distributions via pólya urn schemes. *The annals of statistics*, pages 353–355, 1973.

[112] Amr Ahmed, Yucheng Low, Mohamed Aly, Vanja Josifovski, and Alexander J Smola. Scalable distributed inference of dynamic user interests for behavioral targeting. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 114–122, 2011.

[113] Youngchul Cha, Bin Bi, Chu-Cheng Hsieh, and Junghoo Cho. Incorporating popularity in topic models for social network analysis. In *Proceedings of the 36th international ACM*

SIGIR conference on Research and development in information retrieval, pages 223–232. ACM, 2013.

[114] Mao Ye, Krzysztof Janowicz, Christoph Mülligann, and Wang-Chien Lee. What you are is when you are: the temporal dimension of feature types in location-based social networks. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 102–111. ACM, 2011.

[115] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z Sui. Exploring millions of footprints in location sharing services. *ICWSM*, 2011:81–88, 2011.

[116] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 977–984. ACM, 2006.

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

[118] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 93–100. ACM, 2013.

[119] Betim Berjani and Thorsten Strufe. A recommendation system for spots in location-based online social networks. In *Proceedings of the 4th Workshop on Social Network Systems*. ACM, 2011.

[120] Vincent W Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th International Conference on World Wide Web*, pages 1029–1038. ACM, 2010.

[121] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 325–334. ACM, 2011.

[122] Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[123] Moon-Hee Park, Jin-Hyuk Hong, and Sung-Bae Cho. Location-based recommendation system using bayesian user's preference model in mobile devices. In *Ubiquitous Intelligence and Computing*, pages 1130–1139. Springer, 2007.

[124] Bin Liu and Hui Xiong. Point-of-interest recommendation in location based social networks with topic and location awareness. In *SDM*, volume 13, pages 396–404. SIAM, 2013.

[125] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. Lcars: a location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 221–229. ACM, 2013.

[126] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pages 119–128. ACM, 2013.

[127] Defu Lian, Yong Ge, Fuzheng Zhang, Nicholas Jing Yuan, Xing Xie, Tao Zhou, and Yong Rui. Content-aware collaborative filtering for location recommendation based on human mobility data. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 261–270. IEEE, 2015.

[128] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 831–840. ACM, 2014.

[129] Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. Geographical topic discovery and comparison. In *Proceedings of the 20th international conference on World wide web*, pages 247–256. ACM, 2011.

[130] Bo Hu and Martin Ester. Spatial topic modeling in online social media for location recommendation. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 25–32. ACM, 2013.

[131] Géraud Le Falher, Aristides Gionis, and Michael Mathioudakis. Where is the soho of rome? measures and algorithms for finding similar neighborhoods in cities. In *AAAI Conference on Web and Social Media*, 2015.

[132] Zaihan Yang, Alexander Kotov, Aravind Mohan, and Shiyong Lu. Parametric and non-parametric user-aware sentiment topic models. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 413–422. ACM, 2015.

[133] Alexander Kotov and Eugene Agichtein. The importance of being socially-savvy: quantifying the influence of social networks on microblog retrieval. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pages 1905–1908. ACM, 2013.

[134] Alexander Kotov, Yu Wang, and Eugene Agichtein. Leveraging geographical metadata to improve search over social media. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 151–152. ACM, 2013.

[135] Alexander Kotov, Vineeth Rakesh, Eugene Agichtein, and Chandan K Reddy. Geographical latent variable models for microblog retrieval. In *Proceedings of the 37th European Conference on Information Retrieval*, pages 635–647. ACM, 2015.

[136] Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. Personalized travel package recommendation. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 407–416. IEEE, 2011.

[137] Yong Ge, Qi Liu, Hui Xiong, Alexander Tuzhilin, and Jian Chen. Cost-aware travel tour recommendation. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 983–991. ACM, 2011.

[138] Xin Liu, Yong Liu, Karl Aberer, and Chunyan Miao. Personalized point-of-interest recommendation by mining users' preference transition. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pages 733–738. ACM, 2013.

[139] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. Where you like to go next: successive point-of-interest recommendation. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2605–2611. AAAI Press, 2013.

[140] Senjuti Basu Roy, Gautam Das, Sihem Amer-Yahia, and Cong Yu. Interactive itinerary planning. In *IEEE 27th International Conference on Data Engineering*, pages 15–26. IEEE, 2011.

[141] Chenyi Zhang, Hongwei Liang, Ke Wang, and Jianling Sun. Personalized trip recommendation with poi availability and uncertain traveling time. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 911–920. ACM, 2015.

[142] Wei Zhang and Jianyong Wang. Location and time aware social collaborative retrieval for new successive point-of-interest recommendation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1221–1230. ACM, 2015.

[143] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R Lyu, and Irwin King. Stellar: Spatial-temporal latent ranking for successive point-of-interest recommendation. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[144] Weiqing Wang, Hongzhi Yin, Shazia Sadiq, Ling Chen, Min Xie, and Xiaofang Zhou. Spore: A sequential personalized spatial item recommender system. In *IEEE 32nd International Conference on Data Engineering, ICDE,*, 2016.

[145] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM, 2009.

[146] Aristides Gionis, Theodoros Lappas, Konstantinos Pelechrinis, and Evimaria Terzi. Customized tour recommendations in urban areas. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 313–322. ACM, 2014.

[147] Hsun-Ping Hsieh and Cheng-Te Li. Mining and planning time-aware routes from check-in data. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 481–490. ACM, 2014.

[148] Yu-Ting Wen, Kae-Jer Cho, Wen-Chih Peng, Jinyoung Yeo, and Seung-won Hwang. Kstr: Keyword-aware skyline travel route recommendation. In *Data Mining (ICDM), IEEE International Conference on*, pages 449–458. IEEE, 2015.

[149] Takeshi Kurashima, Tomoharu Iwata, Go Irie, and Ko Fujimura. Travel route recommendation using geotags in photo sharing sites. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 579–588. ACM, 2010.

ABSTRACT

PROBABILISTIC PERSONALIZED RECOMMENDATION MODELS FOR HETEROGENEOUS SOCIAL DATA

by

VINEETH RAKESH MOHAN

August 2017

Advisor: Dr. Chandan K. Reddy and Dr. Harpreet Singh

Major: Computer Engineering

Degree: Doctor of Philosophy

Content recommendation has risen to a new dimension with the advent of platforms like Twitter, Facebook, FriendFeed, Dailybooth, and Instagram. Although this uproar of data has provided us with a goldmine of real-world information, the problem of information overload has become a major barrier in developing predictive models. Therefore, the objective of this Thesis is to propose various recommendation, prediction and information retrieval models that are capable of leveraging such vast heterogeneous content. More specifically, this Thesis focuses on proposing models based on probabilistic generative frameworks for the following tasks: (a) recommending backers and projects in Kickstarter crowdfunding domain and (b) point of interest recommendation in Foursquare. Through comprehensive set of experiments over a variety of datasets, we show that our models are capable of providing practically useful results for recommendation and information retrieval tasks.

AUTOBIOGRAPHICAL STATEMENT

Vineeth Rakesh Mohan completed his Bachelors in Electronics and Communication Engineering from Jerusalem College of Engineering, Chennai, India in 2009 and obtained his Masters in Computer Science from Wayne State University in 2013. He enrolled in the PhD program under the department of Computer Engineering at Wayne State University in August 2011. His primary research interests include data mining, machine learning, topic models, graphical models, recommender systems, text mining, information retrieval systems, social networks and big data analysis. He has completed internships at Comcast Labs and Technicolor Labs and published papers at top-tier conferences such as IEEE, ACM, WSDM, AAAI, ICWSM, and ECIR. He has also served as a co-reviewer for more than 50 conference and journal papers in Computer Science.