A Study on Efficient Utilization of Word of Mouth in Recommender Systems
(口コミ情報と情報推薦の効果的な連携手法の研究)

by

Guanshen Fang

March, 2018

A dissertation submitted to the Faculty of the Information Engineering Course of the Graduate School of Engineering of Hiroshima University in partial fulfillment of the requirements for the degree of Doctor of Engineering.

Acknowledgments

Firstly, I would like to thank my supervisor Prof. Satoshi Fujita, who enthusiastically guides and supports me during my five years of research. He always gave me valuable suggestions and generously shared his knowledge in the discussion. With his help, I understood the methodology of research and developed logical thinking.

Besides my supervisor, I would like to thank Assoc. Prof. Sayaka Kamei, for her patience and encouragement. She gave many comments and payed great patience in the review of my manuscripts and this thesis. Furthermore, she supplied me necessary devices in research to help me to save time in experiments. Additionally, she also supported me in the internship and attendance of conferences.

I also would like to thank Assoc. Prof. Kenta Oku of Ryukoku University and Assoc. Prof. Tomoko Izumi of Osaka Institute of Technology, for their enthusiastic reception of my internship.

Secondly, I would like to thank Bahaa Alghazawy, who gave me advice in writing and submission of previous papers. The discussion with him is always significant and helpful.

I would like to acknowledge my colleagues and alumni of Distributed Systems Laboratory of Hiroshima University, especially, Tomohiko Kimura, Dai Kodama, Takanori Takeuchi and Shinjiro Fujita, for their friendly help.

Finally, I would like to especially thank my parents, family and my girlfriend for their endless support, encouragement and love during the preparation of the thesis.

Thanks again.

Abstract

Recommender systems are widely used in various domains including business, education, tourism and shopping, as a tool to effectively support the decision making of the users. "Items" recommended to the "users" are not limited to industrial products such as televisions and smartphones, but also logical entities such as news, trend, relationship with friends, and hot sightseeing spots. Recommender systems conduct a match making between a set of users to have specific preferences and a set of items to have specific characteristics. Preference of users is generally acquired through the history of past actions taken by the users (e.g., items bought by the users in the past) and personal attributes of the users such as age, gender, and occupation. In addition to them, *electronic word of mouth* (eWOM) information given by the users concerned with items has recently attracted considerable attentions as a way of extracting: 1) the preference of users and 2) the characteristic of items. The reader should note that it could not only include the textual reviews of users but also the user-generated contents regarding specific items issued in Social Networking Services such as Twitter and Facebook.

In this thesis, we discuss the effective utilization of such eWOM in two types of recommender systems. Firstly, we focus on content-based recommender systems in the tourism domain. Unlike industrial products, many sightseeing spots (spots, for short) such as park and mountain could have different features in different seasons. Thus tourism recommender systems should take into account such seasonal features in suggesting appropriate spots to the users, while very few existing systems realize it. Our solution to this issue is to generate a *seasonal feature vector* of spot for each season using the seasonal variance of trend words in Twitter. More concretely, we

simply filter the tweets to obtain the ones that concern with tourism, and extract the words from the tweets published in each season. The baseline vector of a spot is generated from the article concerned with the spot in Wikipedia and the weight of trend words is highlighted in seasonal vectors. The performance of the proposed method is experimentally evaluated. The result indicates that: 1) derived seasonal vectors reflect the similarity of spots for designated time period, and 2) by using seasonal vectors in proposed tourism recommender system, the recommended ranking of spots (containing more than 3 spots) obtains higher precision of user's actual choices of spots than only using baseline vectors instead.

Secondly, we focus on the utilization of textual review of items in collaborative filtering-based recommender systems. In many online services such as review sites and social media, the submitter of textual review is encouraged to couple it with a numeric rating as the explicit feedback concerned with purchased items. Based on the history of such explicit feedbacks, the recommender system predicts the rating of unpurchased items and discovers items that they might like and will buy in the future. Matrix Factorization (MF) is a typical technique to realize such a prediction of unknown ratings. Unfortunately however, most of conventional MF-based systems omit textual reviews, which is pointed out to be a main reason of the mediocre performance. In order to overcome this issue, we propose a method of predicting unknown ratings using textual reviews, which includes a new first-order gradient method for the MF named Topic Gradient Descent (TGD, for short). The proposed prediction method firstly derives latent topics from textual reviews through Latent Dirichlet Allocation. Each of the derived topics is characterized by the probability distribution of words and is assigned a latent factor. Secondly, it conducts the prediction of ratings using MF trained by the TGD method. In the training process, the latent factor of each topic is dynamically updated according to the stochastic proportion of the topic in the review. To evaluate the performance of the proposed method, we conducted experiments using YELP challenge dataset and per-category Amazon review datasets. The experimental results show that the proposed method improves the performance of traditional matrix factorization up to 12.23%.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As the rapid growth of the Internet, huge size of information is created by human. It is reported that the volume of information generated in a single year has exceeded 1 zettabyte (10^{21} bytes) in 2010 [60]. For the web users facing to the so-called information explosion phenomenon, it is difficult and time consuming to find the desired information. In order to support such overwhelmed users, several information filtering techniques are developed by researchers and applied in recent years [44].

An efficient tool to support the decision making of users is recommender systems, which are widely used in various domains including business, education, tourism and shopping [29]. Figure 1.1 shows the recommendation of products on Amazon[1] as an example. As the user has checked or purchased a product, the related other ones would be recommended afterwards. The "Items" recommended to the "users" are not limited to industrial products such as televisions and smartphones, but also logical entities such as news, trend, relationship with friends, and hot sightseeing spots. Recommender systems conducts a match making between a set of users to have specific preferences and a set of items to have specific characteristics. Therefore, there are mainly two tasks in recommender systems: 1) to define the characterization of the user called user profile, which can represent the preference, and 2) to estimate whether the user would like an unpurchased item. With step 2 going through all items, system can simply find out and recommend the ones that the user is most

[1] `www.amazon.com`

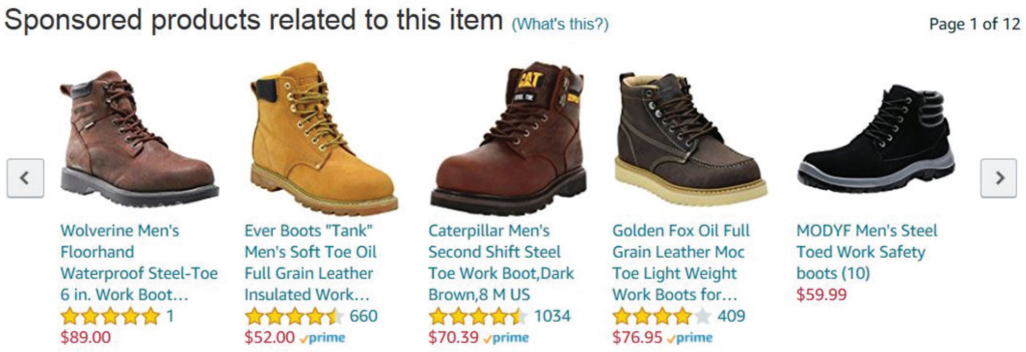


Figure 1.1: Recommendation of productions on Amazon.

probably like.

Preference of a user is generally acquired in two manners to further build their profiles [39, 87]: 1) explicitly through questionnaires or personal attributes such as age, gender, and occupation [18, 38, 51, 63] and 2) implicitly from history of past actions taken by the user (e.g., the items bought in the past) [42, 59, 7, 69, 55]. Reliability is the advantage of explicit method since it acquires the information directly from the user [50]. While it needs his/her cooperation (e.g. to ful-fill the basic information), which may become a additional burden placed to the user [39]. On the other hand, the implicit method does not require any mental efforts by the users. For example, the profile of an individual user is defined as a set of words which are included in his/her navigated websites.

For the second task stated before, an efficient way is to estimate the user's response to an unpurchased item based on his profile. For instance, how would the user rates the item if he/she buys it, or like/dislike it. According to the strategies to make such estimation, recommender systems are generally categorized into three types [9]:

- Content-based recommender system: to recommend items that share the common characteristic with the ones having been chosen by the user before;
- Collaborative filtering system: to recommend items having been chosen by other users who share common purchased items with the current user;

Figure 1.2: User's feedback of an item including a rating represented in stars and textual review.

- Hybrid recommender system: combination of content-based approach and collaborative filtering approach.

For each of them, we give the introduction in Section 1.2.

In this thesis, we focus on the effective utilization of *electronic word-of-mouth* (eWOM) information on recommender system, to enhance the accuracy of the estimation. eWOM is any positive or negative statement regarding specific items made by potential, actual or a former customer which is available to a multitude of people via the internet [47]. As a way of extracting the preferences of users and characteristic of items, eWOM have recently attracted considerable attentions [3, 32, 52, 77, 114]. Such user-generated contents could be the feedback of users issued on e-commerce site, or messages on social media such as Twitter and Facebook. The background of eWOM will be introduced in following subsection.

1.1 Background of Word of Mouth

Before the invention of Internet, word-of-mouth refers to an oral face-to-face communication of users about products and services [6]. As the development of Internet

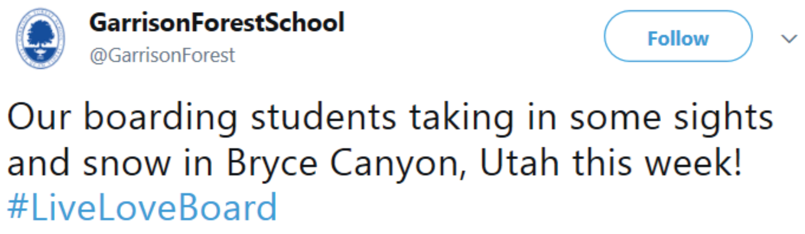


Figure 1.3: An example of tweets concern with tourism on Twitter.

and e-commerce sites as Amazon[2], more and more users use online shopping, and are encouraged to submit their feedback for items after their purchase. As one form of eWOM, the feedback are often automatically shared to others, and expected to draw the attention from the ones who potentially buy in future. Typically, as shown in Figure 1.2, feedback of a user against an item includes its author's name, the item's name, the date of submission, a numeric *rating* and a *textual review*. Although "review" can also be generally refered to the tuple of these five components, in order to avoid the ambiguation we especially call the short "document" itself as review in this thesis. The rating represents the overall evaluation of the user to the item and is assigned in a range (e.g. On Amazon, a rating is illustrated with up to 5 stars). For most of the users, medium ratings are assigned to the items which are neither too bad nor too good (3 of 5 stars), and full marks to the satisfied ones. Additionally, a textual review contains the opinion of the user about the item, such as personal in-use experience, which could be seen as detailed representation of the corresponding rating. Regardless of its rich information, traditional recommender systems focused on only the users' history of ratings. The recommendation made for an individual user are based on only his/her history of ratings. The utilization of reviews in recommender systems is discussed in this thesis, in order to further improve their performance.

On the other hand, social media has developed rapidly and provides a new platform for eWOM in the last decade. It describes a variety of new sources of online information that is created, initiated, circulated and used by consumers about products, brands, services, personalities, and issues [76, 111]. One of the most successful

[2] www.amazon.com

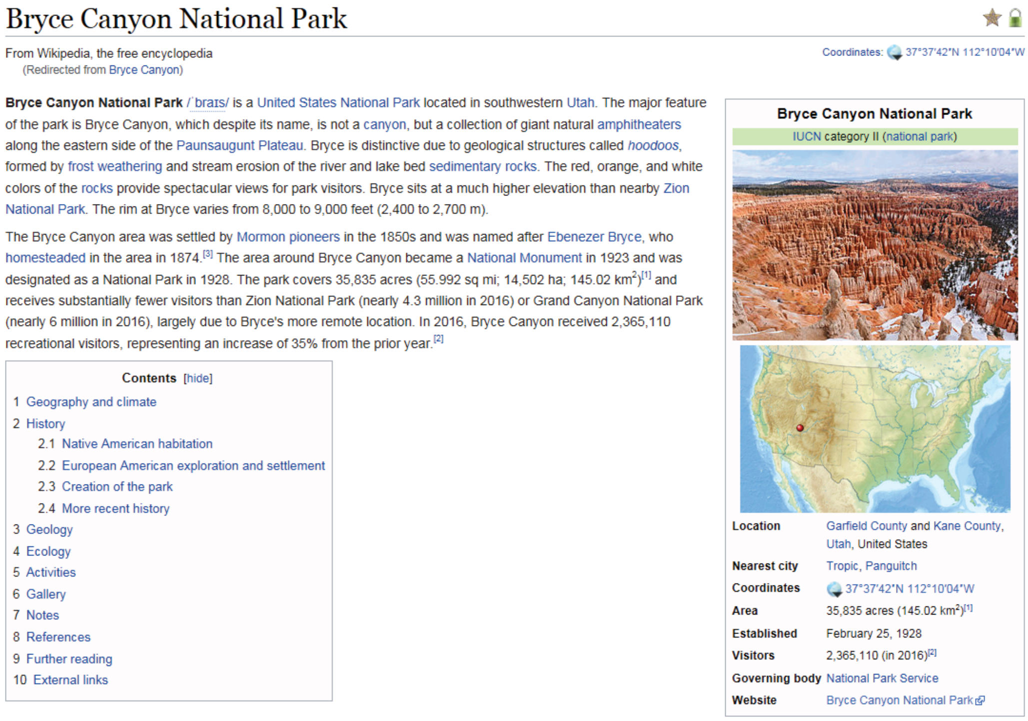


Figure 1.4: An example of Wikipedia article of Bryce Canyon National Park.

social media is Twitter [3], which has exceeding 271 millions of monthly active users in 2014. With mobile devices such as smartphones and tablets, its users can conveniently publish messages called *tweets* about the surroundings whenever and wherever they want. Such tweets are automatically shared to their online friends called followers to encourage the communication. For example, in the visit of a tourist spot, a user may share his/her feeling about an attraction once he/she arrives (Figure 1.3). When publishing a tweet, a textual message is always necessary with 140 words limit of its length. Therefore, one tweet is often assumed to be associated to one actual topic. In this thesis, we especially focus on a certain portion of tweets that are relevant to tourist spots. Since every tweet is tagged with a time stamp, they could effectively work as the source of seasonal information of spots.

[3] `www.twitter.com`

Different with Twitter, Wikipedia[4] is a free, open content online encyclopedia created through the collaborative effort of a community of users known as Wikipedians [17]. As one of the first social media, it is currently the world's largest knowledge resource, containing more than three million articles. Each of the articles concerns with a specific object or concept in real world, such as a country, person, or an event held in a certain area and period, etc.. As Figure 1.4 shown, an article often consists several parts: 1) a certain title with disambiguation, and 2) a basic information box in the right part and 3) a detailed content to describe the object. The textual content is usually seriously edited and contains rich interlinks to other related articles. Another significant property of the article is the categories, which are also tagged by Wikipedians manually. For example Bryce Canyon National Park and Arches National Park are both categoried into "National parks in Utah", which is also a sub-category of "United States National Parks". With such categories to be seen as nodes, they form a large structure of graph containing edges to represent their hierarchical relationship. Due to its good organization and accurate information, Wikipedia is widely used as a knowlegde resource in many researches [55, 23, 72, 75]. In this thesis, we use it as an external source to provide the basic description to the sightseeing spots of Japan.

1.2 Background of Recommender System

1.2.1 Content-based Recommendation

Content-based recommender systems are primarily proposed to filter and recommend textual documents, e.g. news and web pages [65]. They assume that the items preferred by the user in the past, will also be preferred in the future. Therefore, content-based systems tend to recommend items that are similar to the ones having been chosen. Figure 1.5 shows their structure, where three main process are included: 1) to characterize user with user profile as stated previously; 2) to define item profile which characterizes the properties of each item and 3) to match the profile of user

[4] `www.Wikipedia.com`

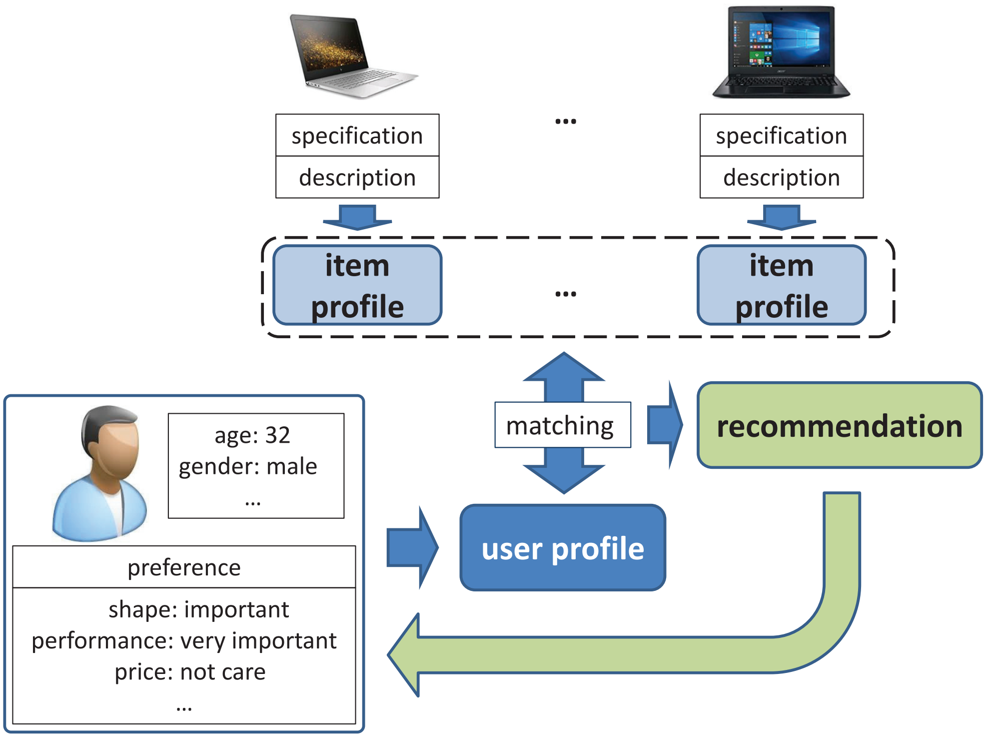


Figure 1.5: The structure of content-based recommender systems.

with items', to make the recommendation. In the construction of item profile, most of the systems generally focus on the features of the item (e.g. actors, directors and the plot of a movie). Efficient but costly approach to collect such features is manual and direct assignment [79, 19, 54]. For instance, building a database to store the manually assigned features of songs by experts. Other approaches build item profile automatically, by using the information collaboratively edited by the users(e.g. tags to the movies), or external source that concerns with the items [33, 99, 31]. A widely used method is to extract the words from each item's related textual document as features, and weight them by using techniques from information retrieval (e.g. TF-IDF [97]). Such words and their weights define an individual item's profile.

In the matching of the user and item profiles, the essential basis is to merge the two spaces of user profiles and item profiles into one, to make them comparable. For

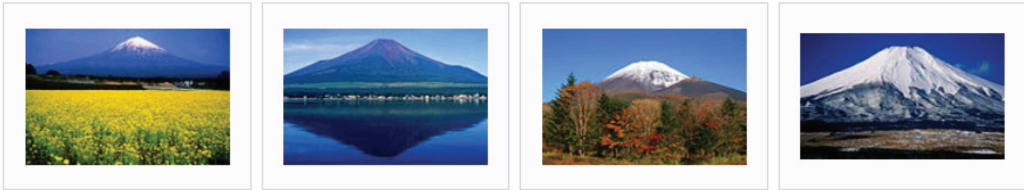


Figure 1.6: The scenes of Mt. Fuji in spring, summer, autumn and winter.

example, both the users and items are characterized with weighted words. With the common space defined, the measure of similarity is computed to judge whether a item's profile is similar with the one of current user. The higher the similarity the more the item fits him/her. Widely used measurements include consine similarity, Euclidean distance etc.. If the number of users and items are huge so that the computation of similarity among them is time consuming, clustering models could be integrated to previously segment them to improve system's scalability [68].

Although content-based recommendation has been applied successfully in many domains including tourism [21, 64, 5, 112], they seldom produce recommendation considering season. Unlike industrial products, many sightseeing spots (spots, for short) such as park and mountain could have different features in different seasons. Figure 1.6 shows the seasonal features of Mt. Fuji as examples: the cherry-blossom of spring, climbing of summer, red leaves of autumn and snow of winter. Thus tourism recommender systems should take into account such seasonal features in suggesting appropriate spots to the users, while very few existing systems realize it. In this thesis, we propose a seasonal tourism recommender system as a solution to such issue, which is described in Chapter 3.

1.2.2 Collaborative Filtering Recommendation

Different with content-based approaches, collaborative filtering (CF) recommender systems estimate the response of the user to a new item based on others' ratings. In this thesis, we consider such response to be the numeric rating of feedback. A significant assumption is that the users sharing similar ratings to their common items

Table 1.1: Similarity computation and rating prediction in CF.

	$item_1$	$item_2$	$item_3$	$item_4$	$item_5$	$item_6$	Pearson with $user_2$
$user_1$	7	6	7	4	5	4	0.956
$user_2$	?	3	3	1	1	?	1
$user_3$	1	2	2	3	3	4	0.789

in the past, are also likely to have similar ones for a certain item in the future [41]. The approaches in CF are classified into two categories [20, 2]: 1) neighborhood-based, also refers to memory-based CF and 2) model-based.

Although neighborhood-based approaches are first proposed in late 1990s, due to their simplicity and intuition, nowadays they are still with huge amount of popularity.

For given users and their history of ratings for a set of items, they define a user-item rating matrix with the ratings to be its elements, as Figure 3.7 shown. The methods to predict the unknown rating of $user_2$ to the $item_1$ can further be seperated into user-based and item-based, according to their perspective on user or item. User-based methods compute the similarities between $user_2$ and other two users, as shown in the last column. The focused unknown rating can be calculated using such similarities: $(6 \times 0.981 + 1 \times 0.789)/(0.981 + 0.789) = 3.77$. Instead, item-based methods compute the similarities among items based on their concerned ratings from all users. The prediction of an unknown rating for current user is based on his/her ratings to other similar items.

On the other hand, model-based approaches try to design models to recognize the underlying patterns in users' ratings, and then to make intelligent predictions for unknown ones. Since rating prediction essentially is a classification problem (e.g. classification of 5 classes for ratings in the range of $[1, 5]$), machine learning algorithms such as Bayesian belief net [86], neural networks [94, 13] have been applied. The objective of a model-based recommender system is to define and minimize a function of errors between the predictions with actual ratings, called loss function or objective function. In the optimization of the model, definite portion of a given dataset is used to train the parameters in loss function. With such learnt parameters, system generates the predictions for the unknown ratings. Performance of the model can be evaluated according to the such prediction errors with actual ratings in the remained testing set. Recent researches verified that model-based systems obtained better performance than neighborhood-based systems [103, 16, 101, 66]. Among model-based CF algorithms, latent factor-based matrix factorization (MF) is the most famous one [62, 96, 56], which is only based on users' history of ratings. As pointed out in some researches, the ignorance of the reviews is the main reason of its mediocre performance [108, 77, 10]. In this thesis, we focus on the analysis of textual reviews and its combination with MF, to further improve the performance.

As drawbacks, CF systems suffer the cold-start problem and decline of performance caused by data sparsity. Cold-start problem occurs when making the recommendation for new users/items. Since they have few ratings concerned, the computation of similarities becomes an issue. In the case of recommender systems, data sparsity refers to the phenomenon that few ratings are included in dataset. The similarities calculated solely using such ratings may be inaccurate, which further influence the accuracy of recommendation.

1.2.3 Hybrid Recommendation

Hybrid approaches combine content-based and CF approaches together. Their motivation is to overcome the presented drawbacks [80, 85], or to further improve systems' performance and the explanation of recommendation [25]. For instance, in some hybrid systems which sequentially combine content-based approaches with CF, the given users' profiles are constructed by content-based techniques. Neighborhood-based CF methods are then applied to estimate the unknown ratings, with similarities computed by such profiles instead of the ratings [22, 58]. The methods of combination in hybrid recommender systems can be classified into three categories [2]:

- Linear combination: to separately implement content-based and CF methods, and only aggregate their predictions;
- Sequential combination: to introduce content-based characteristics into CF approaches, or conversely CF characteristics into a content-based approaches;

- Mixed combination: to build one model which integrates both content-based and CF characteristics.

Since we do not focus on such type of systems in this thesis, in following chapters we will not further introduce their detail.

1.3 Contribution

In this thesis, we discuss the effective utilization of such eWOM information in two types of recommender systems. We firstly propose a content-based tourism recommender system using Twitter and Wikipedia, with considering the season of user's travel. The difficulty is how to characterize the seasonal features of individual spot to build its profile. Our solution is firstly to generate a seasonal feature vector of spot for each season using the seasonal variance of trend words in Twitter. More concretely, we simply filter the tweets to obtain the ones that concern with tourism, and extract the words from the tweets published in each season. The baseline vector of a spot is generated from the article concerned with each spot in Wikipedia and the weights of trend words is highlighted in seasonal vectors. After that, such seasonal vectors are used to be the profile of the spot, and matched with the profile of the current user to decide the recommendation for his/her designated travel period. The user profile is built in the manner that the proposed system directly asks the user for his/her most favorite spots in the past. The performance of the proposed method is experimentally evaluated. The results indicate that: 1) derived seasonal vectors reflect the similarity of spots for designated time period, and 2) by using seasonal vectors in proposed tourism recommender system, the recommended ranking of spots obtains higher precision of user's actual choices of spots than comparison, including the ones generated only using baseline vectors instead and without the awareness of season.

Secondly, we focus on the utilization of textual reviews of items in model-based collaborative filtering recommender systems. We propose a method of predicting unknown ratings using textual reviews, which includes a new first-order gradient method for the MF named Topic Gradient Descent (TGD, for short). The proposed prediction method firstly derives latent topics from textual reviews through Latent Dirichlet Allocation. Each of the derived topics is characterized by the probability distribution of words and is assigned a latent factor. After that, it conducts the prediction of ratings using MF trained by the TGD method. In the training process, the latent factor of each topic is dynamically updated according to the stochastic proportion of the topic in the review. To evaluate the performance of the proposed method, we conducted experiments using YELP challenge dataset and per-category Amazon review datasets. The experimental results show that the proposed method improves the performance of traditional matrix factorization up to 12.23%.

1.4 Thesis Organization

In Chapter 2, we go through the related work of content-based and collaborative filtering recommender systems. For the former, we especially concentrate our attention on tourism domain, to review relevent lectures and summarize techniques that have been applied. For the latter, we review some influential and state-of-the-art approaches, and presented their advantages and drawbacks.

In Chapter 3, we present the proposed content-based seasonal tourism recommender system in detail. The characteristic of Wikipedia articles and tweets are explained with examples. Not only system's structure and techniques having been applied, we also explain the several perspectives of evaluation and their methodologies and results. Especially, it includes a discussion of the ignorance to the noise from tweets.

In Chapter 4, a method to provide rating prediction to unknown ratings of users is presented. We firstly point out the drawbacks of existing methods with actual examples, and explain the main idea of our solution. The detail of proposed topic gradient descent method (TGD) is emphasized. Then we design and conduct complete evaluation to the performance of TGD as well as a discussion on its parameter assignments based on the results. Finally, with the perspective of entire rating prediction method, we analyze the results of evaluation to its prediction accuracy.

In Chapter 5, we summary the thesis and give its future work.

Chapter 2

Related Work

In this chapter, we review the prior related work which concerns with the topics discussed in this thesis, including content-based and CF-based recommender systems. We introduce and summarize not only the techniques having been applied, but also the methods in their evaluation additionally. Especially, since we propose a new content-based tourism recommender system in the next chapter we go deeper into the detail of this type's systems in the same field. At the end of this chapter, we investigate the existing methods to integrate eWOM in recommender systems.

2.1 Recommender System

Since we propose new content-based and collaborative filtering systems in following chapters, we focus on these types' approaches in this section. As stated previously, although hybrid recommender systems are recently proposed in many researches, we do not introduce them for their little relevance with this thesis.

2.1.1 Content-based Systems

The basis of content-based systems is that the choice of items made by a user is independent with others, and is only determined by his/her preferences and interests. Accordingly, the basic process performed by a content-based recommender consists in matching up the user profile which presents his/her preferences and interests, with the attributes of a content item, in order to recommend to the user new interesting items [92]. Three separate components consist in such process:

- Content analyzer: the component to characterize items using their conerned content from information source, which is often in the form of text. Such characterization can be called as item profiles, which represent a common information space for all items. The item profiles are the input of the next user profile learner and filtering component.

- User profile learner: the component collects information that represents preferences of users, and further generalizes them into user profiles. As stated in the last chapter, such collection can be implicit usage of the reactions of users to items(e.g. feedback), or explicitly ask users to define and provide their interested areas.

- Filtering component: the component conducts the matching of user profiles with item profiles, to suggest relevent items of users. The similarity metrics are often used as the measurement of relevance.

Since the information that content analyzer deals with is often text (e.g. description of items), it borrows the techniques from Natural Language Processing and Information Retrieval field including morphological analysis [83] and weighting schemes of words [8]. The textual information of items is always unstructured, the first step of analysis is to extract the words, and discard the stop words (meaningless content-free words) included. For the nouns, stemming is conducted to unify their phrases. With such pre-processing, structured representation of each item is produced, containing a set of words to be its features. In Vector Space Model [98], for each item one feature vector is generated to be its profile. More concretely, with $D = \{d_1, \ldots, d_N\}$ to be the textual content of documents of N items and $T = \{t_1, \ldots, t_n\}$ to be the dictionary of words in D, the task is to present each d_j with a n-demensional vector $\{w_{1,j}, \ldots, w_{n,j}\}$, with $w_{k,j}$ denoting the weight of word t_k in d_j. Widely used schemes of weighting words include Term Frequency-Inverse Document Frequency (TF-IDF)

[97], Okapi BM25 [93], etc.. Let $f_{k,j}$ denote the times that word t_k appears in d_j (frequency). TF-IDF weight of the word t_k in d_j is calculated as

$$TF-IDF(k,j) = \underbrace{\frac{f_{k,j}}{\sum_{k'} f_{k'j}}}_{\text{TF(k, j)}} \cdot \underbrace{log\frac{|D|}{m_k}}_{\text{IDF(k)}},$$

where m_k denotes the number of documents that include t_k in their content. Based on the Term Frequency (TF) and Inverse Document Frequency (IDF), Okapi BM25 weight of t_k in d_j is calculated as

$$BM25(k,j) = \frac{TF(k,j) \cdot IDF(k) \cdot (\mathcal{K}+1)}{TF(k,j) + \mathcal{K} \cdot (1 - b + b \cdot \frac{|D|}{dl})},$$

where dl denotes the average number of words in a document. For both TF-IDF and BM25, representative words of a document always have larger weights than the rest ones.

In user profile learner, based on item profiles, a user's profile can also be constructed as one or more vectors similarly. For example, in CRESDUP [26], a web site recommender system, a user is taken as a particular "document" to contain the sites he/she has navigated. For the words included in such "document", the TF-IDF weights are calculate among all users' "documents" to form a single vector as the user's profile. With such manner, both the profiles of users and items are represented into a common n-demensional word space. In filtering component, measurement such as consine similarity is computed between user profile and item profiles, to infer whether the items can fit the user. Let $u \in \mathcal{R}^n$ denotes the user's profile and the element u_k characterizes the preference on word-represented feature t_k. The cosine similarity with d_j is calculated as

$$SIM(u, d_i) = \frac{\sum_{i=1}^{n} u_i \cdot w_{i,j}}{\sqrt{\sum_{i=1}^{n} {u_i}^2} \cdot \sqrt{\sum_{i=1}^{n} w_{i,j}}} \tag{2.1}$$

Cosine similarity essentially depends on the angle between two vectors in the space. In other words, if the vectors of a user profile and a item profile have a small angle, they would have large similarity so that the item would be recommended to him/her.

Such keyword-based vector space model (VSM) has been used in many existing recommender systems in various fields, such as personal web search [82, 106, 105], recommendation of news [90, 89], etc.. Esparza *et al.* [35] propose a content-based product recommender system, which takes each item as a aggregated document to contain all its relevant textual reviews. In the profiling of item, it not only weights the words in reviews but also its tags using TF-IDF and BM25. Similarly, Bellogín *et al.* [12] proposes a method to recommend music, based on the tags collaboratively assigned by users. In the mobile application recommender system proposed by Wang *et al.* [109], for each application a vector representation is formed, which consists of the TF-IDF weights of the words extracted from its textual description. In their matching phase, the textual similarity of two applications is further calculated based on such vectors. In the domain of citation recommendation, Kannan *et al.* [24] use the concepts instead of the words included in the lecture. Each lecture's concept vector is given by Computing Classification System of ACM.

The advantage of content-based recommendation comes from its user independence. The recommendation of a item do not influenced by its popularity and only depends on its characterized features. Therefore, the new items which are not yet rated by any user can be covered normally. Additionally, content-based recommender systems can easily provide explanation of the recommendation by explicitly listing the features of items that match with the preference of user. The user may further decide whether to trust the recommendation and purchase. On the other hand, as drawback, content-based recommendation has the limitation on the number and type of features associated with items.

Content-based Tourism Recommender System

Content-based tourism recommendation systems try to recommend spots similar to those users have liked or visited in the past (*i.e.*, history)[70, 71]. From the history of visit, user's profile is built to represent his preference. On the other hand, the features of spots are characterized in order to match with user's profile to decide the recommendation. Some of existing researches aim to provide the user an appropriate tour plan to meet his/her constraints, such as time or cost [49, 1, 64]. The features of spots are given from experts of tourism, which simply includes available time, normal visiting time and geographical information, etc.. Therefore, the recommendation of the tour plan turns to an integer programming problem or traveling salesman problem to approximate a combination of spots with minimization of the travel path or time wasted in movement. Győrödi *et al.* [43] propose a spot recommendation with a mobile application. In order to determine user's interest and features of spot, they use tags such as food, music etc., which can be established by users and assigned to a specific spot. The recommendation is produced by matching such tags of the given user and spots. Herzog *et al.* [49] propose a travel recommender system, which provides a sequence of region (western Canada, etc.) instead of spots. There are three phases in their proposed system, firstly they filter all regions with linear programming to find out the candidates of regions that meet the time and cost limitation of user. After that, the feature of candidates are matched with the features of the user's query, to decide their ranking. Similarly, Abbaspour *et al.* [1] focus on the recommendation of tour plan. As the request of recommendation, the current user has to assign priorities to the spots that he/she wishes to visit, and the duration of travel as well. According to the user's limitation, the system generates a route including a list of spots which minimize the time wasted in user's movements and waiting. Carrasco *et al.* propose a content-based recommendation system called Sem-Fit [5], to provide hotels in the travel of users. More concretely, it applies the fuzzy logic to relate users and hotel characteristics, and continuously refines the fuzzy rules according to the feedback received from users.

In existing content-based researches much efforts are made to provide tourist spots or plan to meet user's needs. Such contents-based approaches would work well if we could generate accurate feature vectors of the users and items. However, it is difficult in actual tourism recommender systems since the features of POIs vary for each season. It still remains an issue that the travel season of the user which is an essential factor in decision of spots, is seldom taken into account.

2.1.2 Collaborative Filtering systems

The recommendation problem to which CF approaches generally face is the estimation to the respond of the user to his/her unseen items. The most common form of such evaluation is *ratings*. Other binary (e.g. like/dislike) and unary (e.g. purchase or access etc.) forms can be easily transformed into ratings by numerically labeling the possible values. Unlike content-based approaches, CF approaches focus on the relevance of users, and predict an unknown rating rely on the ones made by others [92]. Let $u_i (i \in \{1, \ldots, I\})$ denote a user and $v_j (j \in \{1, \ldots, J\})$ denote a item. The actual rating of u_i to v_j is denoted as $r_{i,j}$, and its prediction can be denoted as $\hat{r}_{i,j}$ correspondingly. The approaches of CF to infer $\hat{r}_{i,j}$ can be categoried into two types, namely memory (neighborhood)-based and model-based. In the following subsections, we introduce the detail and review the lectures for each of them.

Memory-based Approaches

As stated in Chapter 1, the process of memory-based CF usually begins with the construct of $I \times J$ rating matrix R, with the element $R_{i,j}$ in row i column j to store the rating $r_{i,j}$. If item v_j is new for u_i, the rating prediction $\hat{r}_{i,j}$ can be made using ratings given to v_j by other users (user-based), or the ones given to other items by user u_i (item-based). The first task of user-based method is to decide the other users whose ratings would be referenced in the prediction made for u_i. Such users are called nearest-neighbors, denoted by $N_u(i)$, and must have similar rating pattern with u_i. The similarity measurements play a double role: 1) they allow the selection of users whose ratings are used in the prediction, and 2) they provide the means to give more or less importance to such users in the rating prediction.

One of the measurements based on history of ratings is cosine similarity, which is introduced in the subsection of content-based approaches. For u_i and $u_{i'}$, their similarity is calculated by Equation 2.1, by taking place the word weights with their ratings stored in i and i' rows of R. Correspondingly the dictionary of words should be also replaced by the items that are both rated by u_i and $u_{i'}$, denoted as $\mathcal{I}_{i,i'}$.

Another popular measurement is Pearson correlation coefficient:

$$PC(u_i, u_{i'}) = \frac{\sum_{j \in \mathcal{I}_{i,i'}} (r_{i,j} - \bar{r_i}) \cdot (r_{i',j} - \bar{r_{i'}})}{\sqrt{\sum_{j \in \mathcal{I}_{i,i'}} (r_{i,j} - \bar{r_i})^2 \cdot \sum_{j \in \mathcal{I}_{i,i'}} (r_{i',j} - \bar{r_{i'}})^2}}, \tag{2.2}$$

where $\bar{r_i}$ and $\bar{r_{i'}}$ denote the average of ratings of u_i and $u_{i'}$ respectively. $PC(u_i, u_{i'})$ ranges in $[0, 1]$ and represents the strength of the their correlation in behavior of rating. Then the rating prediction $\hat{r}_{i,j}$ can be simply calculated as

$$\hat{r}_{i,j} = \frac{\sum_{i' \in N_u(i)} PC(u_i, u_i') \cdot r_{i',j}}{\sum_{i' \in N_u(i)} |PC(u_i, u_i')|}, \tag{2.3}$$

where $|\cdot|$ denotes the absolute value.

The process of item-based method is similar with user-based, only change its perspective to each individual item. In other words, to similar items, common pattern should underlies in the ratings made by the users. The similarity of items and rating prediction can be calculated by re-using Equation 2.2 and Equation 2.3, by using the neighborhoods of the item and the ratings concerned with them instead.

Due to their simplicity and efficiency, memory-based CF still has great popularity nowadays. Improvements are made in addressing the shortage that all items are treated the same in the computation of the similarity. Choi *et al.* [27] propose a new similarity measure for neighborhood CF. In the inference of a rating given by a user to a specific objective item, not only the similarities of the user and the neighborhoods, but also the similarities of such common items with the objective one is considered. In other words, for the user and his neighborhood, their ratings to the common items that are similar to the objective item is emphasised. Instead of rating patterns, Deshpande *et al.* [30] focus on the co-occurrence of purchase of items in the user's history information. They proposed a conditional probability-based similarity calculation of items. Bobadilla *et al.* [15] propose a new similarity measure for CF based recommendation, which takes the singularity of the ratings into account. Such singularity refers to the phenomenon that a few users' ratings to a specific item may totally different with others. They pointed out that the ratings with such property should be emphasised in the calculation of users' similarity. Additionally, in order to solve the failure of calculation of similarity when the users have few common items, Luo *et al.* [74] define global similarity of a user which depends on the ones of his/her neighborhoods to other users.

The performance of neighborhood-based methods depends on the entire numbers of users and items. When the number of users are far less than items', user-based neighborhood methods would obtain better performance than item-based ones [92]. The drawback is the famous *cold-start* problem, which refers to the failure of recommendation for a new user or item having few concerned ratings. As a solution, content-based recommendation technique is often integrated to compose a hybrid system.

Model-based Approaches

Model-based approaches have drawn huge amout of attention in recent years, which use the pure rating data to estimate or learn a model to make predictions [20]. The core of a model is a rule of rating's prediction, in which the parameters are learnt by machine learning algorithms to make such prediction to fit the actual ratings. For example, by assuming the ratings are integers and locate in $[0, L]$, in probabilistic approaches like Bayesian model [20] and clustering model [104], the unknown rating $r_{i,j}$ is expressed as

$$\hat{r}_{i,j} = \sum_{l=0}^{L} l \cdot Pr(r_{i,j} = l | R_{u_i}),$$

where R_{u_i} is the set of ratings that user u_i made in the past. In Bayesian model, in order to infer the probability, a Bayesian network is applied with its nodes correspond with items. The states of each node correspond to the possible rating values for each item and a "no rating" state for the missing ratings as well.

In recent years, among model-based CF algorithms, latent factor-based approaches are popular for their efficiencies in handling large scale datasource. The idea is to characterize both each user and each item with a vector of latent factors, which comprise a computerized alternative to the properties of items or preference of users. With the common latent factor space into which users and items are transformed, the latent factors of users and items are automatically inferred from the actual history of ratings, to further explain the unknown ratings. More concretely, the user-item rating matrix is approximated by singular value decomposition, which decompose into it two orthogonal matrices as the representation of latent factors of users and items respectively[62, 95]. In the training, the factors are often learnt via gradient descent method. For an unpurchased item of a given user, the rating prediction is made by calculating the inner product of their latent factors:

$$r_{i,j} \sim \hat{r}_{i,j} \quad = \quad \mu + b_i + b_j + U_i^T V_j, \tag{2.4}$$

where U_i and V_j is the vectors containing the latent factors of user u_i and v_j. b_i and b_j is the parameters represent the natural biases of ratings concerned with u_i and v_j respectively. For example, the user tends to rate highly to all the items he/she purchased. Such expression characterizes the interaction of user and item in the rating behavior.

Due to the excellent expansibility and efficiencies in handling large scale datasource, many researches focused on extent it to obtain better performance. Salakhutdinov *et al.* [95] propose Probabilistic Matrix Factorization (PMF) and introduced Gaussian priors as hyperparameters to present latent factors. They noted that maximizing the log-posterior of the ratings over users' and items' latent factors is equivalent to minimizing the squared errors of the rating prediction. Hu *et al.* [53] apply latent factor model to the recommendation for implicit feedback datasets, which includes the indirectly reflect opinions, such as the purchase history of products and browsing history of the web pages. In the optimization of the model, they let the differentiation of user's and item's latent factors be zero, and recalculate the expression for them. Koren *et al.* [61] propose a model for recommendation which integrates the latent factor model and traditional neighborhood model of CF. In their rating prediction, it directly sums up the predictions of these two models together. All these approaches ignore the reviews of users' feedbacks, which makes them scalability but mediocre accuracy in rating prediction. Karatzoglou *et al.* [57] apply latent factor models in the scheme of context-aware recommendation. They pointed out that the decision of users may be influence by their current situations, like location, time etc.. In their approach, they extent MF to 3-demensional tensor factorization in order to characterize such context with latent factors same as users and items. Since the ratings of users is distinguished by context, the rating matrix turns to a tensor, where the vertical slice facing to the front represents the ratings made under a specific context. Accordingly the rating tensor is decomposed into three matrices, including a matrix concern with the factors of context.

Drawbacks of latent factor models include the difficulty in description of recommendation. Its essential reason is that latent factors do not have definite correspondance with actual properties or genres of items, or interest of users. Another one is that latent factors are treated equally both in their learning and prediction of rating. It leads a gap with the actual scene of user's decision making - usually only a part of them is considered.

2.1.3 Evaluation metrics

In the evaluation of recommender systems, several metrics are used to verify its performance. Such metrics can be classified into three types, including predictive accuracy metrics, classification accuracy metrics and rank accuracy metrics [48, 101]. Which one should be used depends on the form of the recommendation provided (output of system).

Mean absolute error (MAE) and root mean squared error (RMSE) are typical predictive accuracy metrics and widely used in verifying the accuracy of rating prediction [10, 57, 74, 15]. As its name shown, for a testing set including users' actual ratings, MAE computes the average of the absolute errors between the actual ones and their predictions

$$MAE = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}|r_{i,j} - \hat{r}_{i,j}|}{m},$$

where m denotes the number of ratings in testing set. Since MAE is not sensitive enough for large prediction errors, many researches use RMSE or as a combination[10, 77, 95]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{I}\sum_{j=1}^{J}(r_{i,j} - \hat{r}_{i,j})^2}{m}}.$$

Both lower MAE and RMSE represent more accurate predictions.

Precision, recall and F1 measures are common classification accuracy metrics in verifying of a set of items to be the recommendation[110, 27, 5]. Let $|UC|$ denote the number of items in user's actual choice, and $|SC|$ in recommendation which contains $|TC|$ correct ones. Precision, Recall and F1 are defined as follows:

$$\begin{aligned} precision &= \frac{|TC|}{|SC|}, \\ recall &= \frac{|TC|}{|UC|}, \\ F1 &= \frac{2 * precision * recall}{precision + recall}. \end{aligned}$$

Additionally, if the recommended items have order (a ranking), based on the precision when each correct recommended item is obtained, average precision (AP) can be calculated to measure the accuracy of such ranking. Furthermore, the average of AP among all users is called mean average precision (MAP), which illustrates the accuracy of recommendation among the representatives of testing set [45, 4].

2.2 Utilization of eWOM in Recommender systems

2.2.1 Identification and Analysis of Tweets for Tourism

Recently, Twitter has been paid much attention as a source of data mining and characterization of spots for tourism informatics. In order to detect tourism related tweets which are posted at specific spots, Shimada *et al.* [100] apply a Support Vector Machine(SVM) to their gathered tweets. Their idea is that the target tweets are similar on their textual content. With the aid of geo-tag, Oku *et al.* [84] propose another SVM-based method of detection of tweets relevant to tourism, and extract temporal features of spots from them. They regard tweets issued within a week as a single document and obtain a temporal feature vector for each week by calculating the TF-IDF weights of keywords contained in such documents. However, it does not generate vectors to cover a sufficient number of spots because it is solely based on tweets. Similarly, Menchavez *et al.* [81] focus on the identification of tourism related tweets and Naive Bayes based sentiment analysis to mine the opinions when tourists visit spots in Philippines. Furthermore, such mined opinions are classified into positive and negative polarity and presented at the geographical map as references for tourists. Similar study has done by Claster *et al.* [28] for Thailand, in which the sentiment analysis of tweets is applied in time-series. Although the problem definitions of previously mentioned researches are different, the objectives are similar which aim to discover useful information of tourism via Twitter.

In such researches, much effort have made to reduce the noise in tweets. However, due to the irregularity of tweets, many meaningless words like punctuation or prefix are always extracted and significantly influence the accuracy of the analysis. In addition, those techniques based on machine learning are sometimes difficult to conduct for minor spots having few related tweets. Since in chapter 3 the proposed system uses Wikipedia as the corpus combined with Twitter, it is free of the influence from such noise. Furthermore, for the minor spots which are seldom tweeted Wikipedia can cover their features and avoid the failure of their recommendation.

2.2.2 Analysis of Wikipedia for Tourism

In recent researches of tourism recommendation, Wikipedia is integrated as an external source in identification of spots. Due to its information-rich articles and categories, it is effective in reducing the cost of manually construction or maintenance of spots' information. A common idea is to take advantage of their geographic information to filter users' geo-tagged photos (e.g., photos in Flickr) and extract their visiting trajectories of spots [11, 73, 67]. Techniques such as T-pattern tree [40] are exploited to mine the traveling patterns potentially contained in extracted trajectories. Lucchese *et al.* [73] propose a tourism recommendation using random walk technique. They identify the point-of-intests (POIs) that users have visited by the geographical information in Wikipedia articles. After that, such POIs are defined as the nodes of a graph, with the edge to represent the relationship of POIs in users touristic itinerary. Lim *et al.* [67] focus on building a system to provide the a travel plan. In the first step of their process, a list of POIs is extracted from Wikipedia, as well as their geographical information and categories. So that user's Flickr photos could be recognized as sequences of POIs.

On the other hand, benefitted by the detailed categories of articles, some researches focus on the construct concept graph to model users and items [88, 107, 37]. Bernadette *et al.* [107] propose a tourism recommender system which integrates DBpedia, which is a ontology based on the categories in Wikipedia. They retrieve the categories of each POI and use them as the characterization. Different with above systems, Fernández-Tobías *et al.* [37] propose a cross-domain recommendation of music and its artists considering the POIs that fit them. Such music tracks may be recommended and played during user's visit to specific POIs. They use the common categories tagged to them to bridge the two domains.

Although in many existed researches Wikipedia is used to combine with SNS to improve the performance of recommendations, few researches take advantage of the content of textual article in Wikipedia, even the detailed description of both permanent and seasonal features is contained.

2.2.3 Latent Factor Models Using Textual Reviews

In recent latent factor-based approaches, semantic analysis of textual reviews is also introduced to further improve the performance[108, 77, 10]. An idea is to take the reviews correlated with individual item as one summary, and construct a model to fit the words in reviews and ratings at the same time. Topic models are integrated in order to recognize the latent topics from reviews. Each topic is represented with a distributions of words so that its actual meaning can only be infered through manually observation. The entire number of topics contained in all reviews has to be previously estimated empirically, which are often set to equal with the number of latent factors. In the training process of a recommendation model, not only the error between prediction and actual ratings is minimized, but also the topic model is optimized at the same time. Wang *et al.* [108] propose a model named Collaborative Topic Regression (CTR), which combines PMF and LDA together. Although it is designed to provide recommendation of scientific articles, the textual reviews can also be handled, by aggregating the ones concerned with a specific item to a single "article". The latent topics of a given article are derived from its title and abstract. Their distribution and a set of parameters are together to form the latent factors of the article. Purushotham *et al.* [91] point out that CTR has poor performance in the case of few feedback contained in the dataset, the so-called sparsity of data. To solve the problem, they integrate follower relationship among users as assistant information. The social network structure is transformed into social matrix, and approximated by an additional MF model. On the other hand, Wang *et al.* [110] propose Collaborative Deep Learning (CDL), a latent factor model based on CTR to improve its performance. Instead of LDA, they use stacked denoising autoencoder to infer the distribution of latent topics for scientific article. Different with other CTR-based method, CDL directly uses the probabilistic distribution of topics to be the latent factors of the given article. The idea of these CTR-based method is to join the distribution of latent topics into users' or items' latent factors, or to replace them. Since the topics are not derived from the individual review, they cannot perceive the unequally treatment of factors in the user's evaluation to a specific item. In contrast, we learn the topics for each review by LDA independently, and use it to be the direction in updating of latent factors in their learning.

Another consideration to combine latent factor model and topic model is to define a transformation between the topic distribution of reviews and latent factor vector of the corresponding item. McAuley *et al.* [77] propose a latent factor based model named Hidden Factors of Topics (HFT) which integrates LDA and MF. The two models are combined with a softmax function of item's latent factors, which transforms them into topic distribution of correlated reviews. Based on HFT, Bao *et al.* [10] propose Topic-MF, in which they replace LDA with Non-negative Matrix Factorization model. Not only the items' latent factors, the users' factors are also introduced into the transformation of softmax function. Therefore, the topic distribution represents no longer the topics in the reviews that correlate with a specific item, but a single review in given feedback. Although it is demonstrated that the performance of HFT and TopicMF outperforms the traditional models such as MF, both of them suffer the drawback of the complicate transformation of latent factor and topic distribution.

Chapter 3

Utilization of SNS in Content-based Recommendation

3.1 Introduction

In this chapter, we focus on content-based seasonal recommendation. Although content-based recommendation has been applied successfully in the tourism domain[21, 64, 5, 112], they seldom produce recommendation considering season.

We propose a content-based seasonal tourism recommender system which fits the designated season of travel. For example, for a user who likes Matsushima island because of cherry-blossom viewing and wishes to travel in spring, the system can recommend other spots with the attraction of cherry-blossom viewing. However, such spots may not be the recommendations of another user who wishes to travel in autumn even if is fond of Matsushima's sea and coast. In order to characterize the dynamically changed seasonal features, the proposed system generates seasonal feature vector for each spot for a given season. Its generation consists three steps. Firstly, it identifies the vocabulary concerned through Wikipedia[1] document about the spot. The reason for using Wikipedia is that each tourist spot has its unique document which introduces its detail features (See Fig. 3.1 which shows the description about the spot "Ritsurin Garden" in Kagawa prefecture including seasonal features for each season). Secondly,

[1] http://ja.wikipedia.org

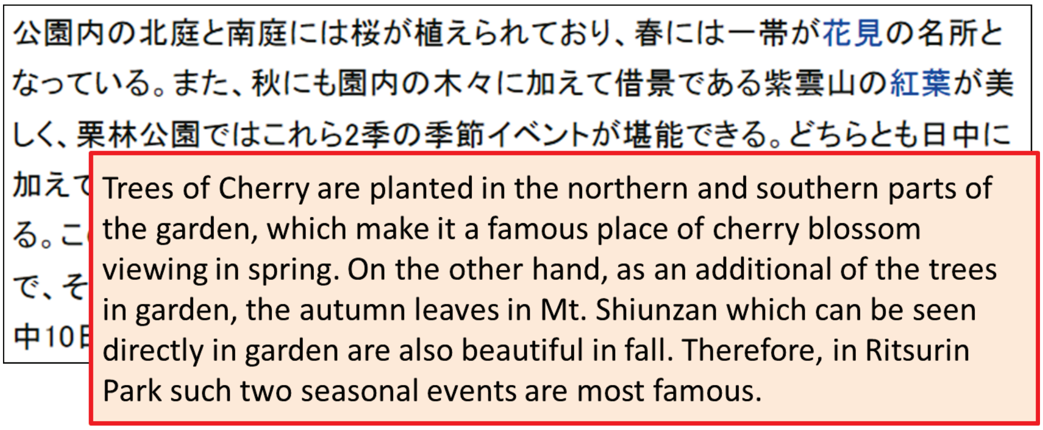


Figure 3.1: A part of the Wikipedia article about Ritsurin Park. A detailed description of the main features is included.

it identifies the trend (*i.e.*, seasonal variance of features for a designated period) over all spots in Japan through Twitter[2]. Finally, for the words (*i.e.*, the features) in the vocabulary, it highlights the ones contained in the identified trend which corresponds to the given season. The reason why we only focus on the defined vocabulary instead of all words in tweets is that it can solve the problem of irregularity and the noise of tweets. With these vectors of spots, the proposed system match them with user's profile which presents his/her preference to decide the recommendations for the travel period of user.

As the implementation, the proposed system gathered 6,057 Wikipedia documents to cover almost all sightseeing spots in Japan. On the other hand, it also collected more than 500 thousands tweets that are published during 7 months. In the experiments we conducted a series of experiments including both a computer simulation and a questionnaire evaluation. The results of experiments firstly indicated that the usage of Wikipedia documents of spots effectively reduces the noise included in tweets. Secondly, the generated season feature vectors certainly reflect the similarity of spots in designated season. Finally, the results of questionnaire shows that the compairing with the recommendation generated without using tweets or the awareness of season,

[2]https://www.twitter.com

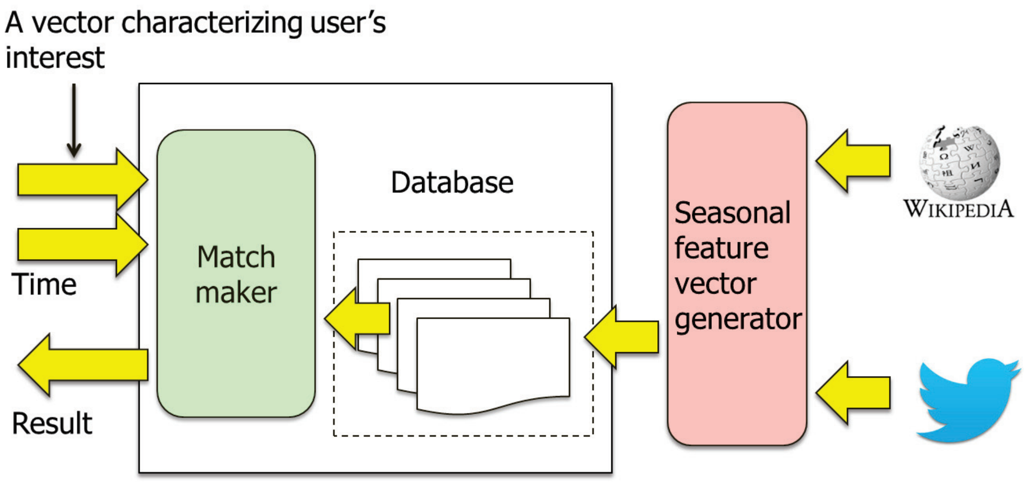


Figure 3.2: Components in proposed tourism recommender system.

the proposed system provides seasonal recommendation which has higher precision of user's actual choices.

The remainder of this chapter is organized as follows. Section 3.2 describes the detail of the proposed system including the generation of seasonal feature vectors and recommendation process as well. Section 3.3 represents the implementation of the prototype system. Section 3.4 represents the method of evaluation and shows its results. Finally, a conclusion is given in section 3.5.

3.2 Seasonal Recommendation of Tourist Spot

In this section, we represent the proposed recommendation system in detail. Fig. 4.2 shows the architecture, which consists of two processes: 1) to generate seasonal feature vectors for each spot; 2) to identify user's preference as profile, and match it with such vectors of spots to produce recommendation. Following subsections detail each part.

3.2.1 Generation of Seasonal Feature Vectors

Firstly, the time axis is assumed to be separated into several ranges so that the features of spots are regarded to be invariant, as in the year end season, the season of cherry-blossom viewing or the bathing season. Each range is called a **season**. The proposed system generates one seasonal feature vector (SFV, for short) of each spot for each season. SFV is calculated by extending the basic feature vector (BFV, for short), in such a way that it reflects the trend of words in each season. More concretely, BFV is a vector of TF-IDF weights (defined bellow) and SFV is its extension.

Let O be the set of spots and d_i be the Wikipedia document about spot $o_i \in O$. Generally d_i is a summarization of the entire information of o_i. Therefore, the reader should note that d_i is the union of statements on spot o_i relevant to various seasons. In other words, in order to generate SFV for each season, the system needs to distinguish word sets relevant to each season in document d_i. Let W_i be the set of words included in document d_i and $W = \bigcup_i W_i$ (*i.e.*, W is the set of words included in Wikipedia documents about O). Then, the term frequency (TF, for short) weight of word w_j in document d_i is defined as

$$TF_{i,w_j} = \frac{n_{i,w_j}}{\sum_{w \in W} n_{i,w}}$$

and the inverse document frequency (IDF, for short) weight of word w_j over $|O|$ documents is defined as

$$IDF_{w_j} = \log\left(\frac{|O|}{m_j}\right)$$

where n_{i,w_j} is the number of occurrences of w_j in d_i and $m_j (\leq |O|)$ is the number of documents containing w_j. With these notions, the BFV $\vec{v_i^b}$ of spot o_i is defined as

$$\vec{v_i^b} = \{(w_j, TF_{i,w_j} \times IDF_{w_j}) \mid w_j \in W_i\}. \tag{3.1}$$

The words which are frequently mentioned in d_i and seldom contained in other documents would have high weights in BFV.

For a given season, the key idea is to extend the definition of the TF weight in (3.1) by considering the trend of words. Let t_k be a collection of tweets issued in season s_k. By considering t_k as a single document, the TF weight of word w_j in season s_k is defined as follows:

$$TF'_{k,w_j} = \frac{n'_{k,w_j}}{\sum_{w \in W} n'_{k,w}} \tag{3.2}$$

where n'_{k,w_j} is the number of occurrences of word w_j in t_k. Because W is the set of words contained in Wikipedia documents about O, *the proposed system omits words in tweets which do not appear in any Wikipedia document*. With the above notions, SFV $\vec{v}^s_{i,k}$ of spot o_i for season s_k is defined as

$$\begin{aligned} \vec{v}^s_{i,k} = & \{(w_j, ((1-\alpha)TF_{i,w_j} + \alpha TF'_{k,w_j}) \times IDF_{w_j}) \\ & \mid w_j \in W_i\} \end{aligned} \tag{3.3}$$

where $0 \leq \alpha \leq 1$ is an appropriate parameter. Note that for o_i, only for the word $w_j \in W_i$ it has TF_{i,w_j} to be non-zero. If word $w_j \in W_i$ has not tweeted in specific season s_k, $TF'_{k,w_j} = 0$; otherwise $TF'_{k,w_j} > 0$, then we say that w_j is highlighted in s_k.

3.2.2 Identification of User's Preference and Recommendation Process

Although an analysis of user's history of tweets would help us to extract his/her preference on the features of sightseeing spots, it may fail for the users who even do not have Twitter accounts or seldom tweet about travel. In order to fit such users, the proposed system extracts user's preference in an explicit way that it directly asks the user for a history of travel. In other words, the user answers two easy questions when he/she begins to use the system: 1) the season that he/she wishes to travel; 2) the most favorite spot that he/she has visited during assigned season until now. Assume that user u chooses tourist spot $o_{i'}$ and the period of season $s_{k'}$. His/her profile which presents preference is defined by the SFV $\vec{v}^s_{i',k'}$ as $\vec{U_{k'}}$. Although the user profiling is simple, it effectively characterizes user's seasonal preference and is with various benefits: first, it does not suffer the cold start problem; second, such questions are easy to answer and time-saving.

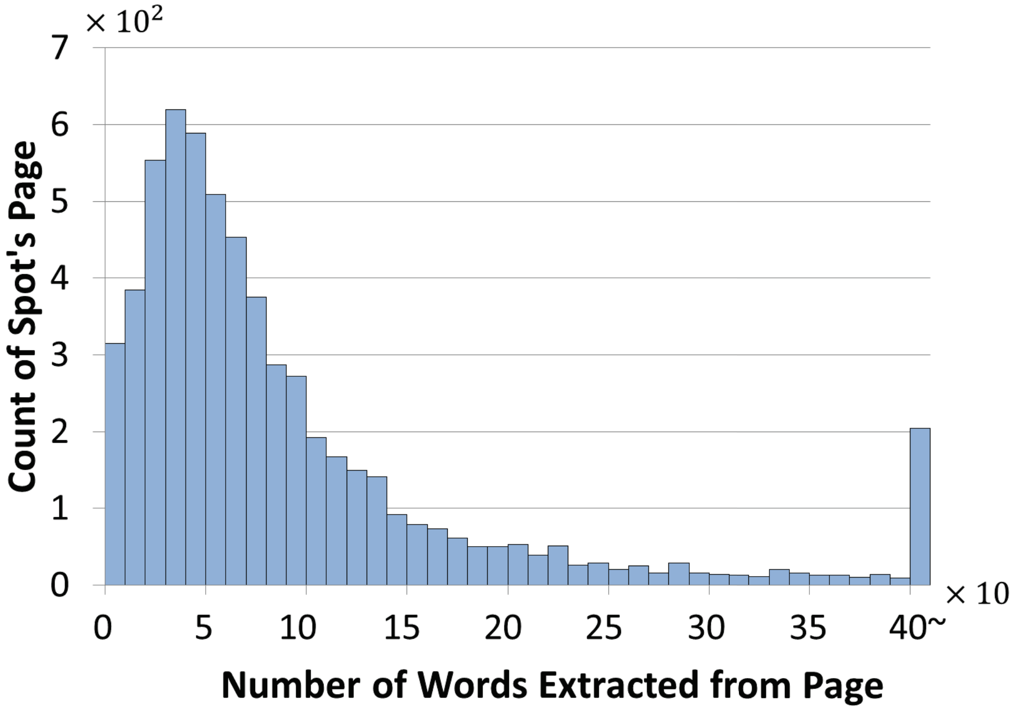


Figure 3.3: The counts of spots with various sizes of words extracted from their documents.

With the constructed user profile, the proposed system matches it with SFVs of spots to decide recommendations for season $s_{k'}$. To quantify the correspondence of $o_{i'}$ and a given spot $o_l(o_l \neq o_{i'})$ in $s_{k'}$, the system calculates their cosine similarity as follows:

$$Sim_{i',l} = \frac{\vec{U_{k'}} \cdot \vec{v}^s_{l,k'}}{|\vec{U_{k'}}||\vec{v}^s_{l,k'}|}, \quad for\ each\ o_l \neq o_{i'}, o_l \in O.$$

The spots with Top-t similarities are the recommendations to user u in $s_{k'}$ as a ranking. Note that the recommendations vary for different seasons designated.

3.3 Implementation of a prototype system

3.3.1 Datasets Description

Since the objective of recommendation is entire tourist spots in Japan, for this prototype system, we focus on 6,057 spots given in the category of "tourist spots in Japan" in Wikipedia, and download the Japanese document for each of them from the Wikipedia server. The prototype system uses only nouns as words in each document d_i. The set of words W_i in d_i is obtained by conducting the morphological analysis using MeCab [3] with the default IPA dictionary. From all collected Wikipedia documents, 608,390 words are extracted overall, in average 100.5 words for a document. The relationship of the count of spots and the number of words which are extracted from their documents is shown in Fig. 3.3. It represents that most spots are introduced in detail in their Wikipedia documents.

A set of tweets relevant to tourism is gained from Twitter using Twitter Streaming API. More concretely, 50 million Japanese tweets issued from September 2013 to March 2014 are acquired. For each of the tweet, its textual content is matched with the names of collected spots. As a result, about 500 thousands tweets containing at least one name of 6,057 spots are regarded as tweets relevant to the tourism and extracted as a part of dataset. Although it may contain tweets which are not relevant to tourism and may miss tweets relevant to the tourism, we didn't evaluate the precision of such a naive extraction since it is the out of scope of this thesis. Let T be the resulting set of tweets.

3.3.2 Parameter Assignments

Considering seasons always last more than one month with different periods of time, assume that one year is divided into 12 disjoint seasons of (almost) equal length in the way that the first season is from January 1st to January 31st, the second season is from February 1st to February 28th (or 29th), and so on. More precisely, seven documents, which represent the trend of each season, is derived from T because collected tweets

[3] http://mecab.sourceforge.net/

Table 3.1: The number of minor SFVs for each s_k when $\ell = 3$.

$\alpha < 1.0$	$\alpha = 1.0$						
	Sep.	Oct.	Nov.	Dec.	Jan.	Feb.	Mar.
61	220	199	205	175	274	601	191

are for seven months. For a given season and its corresponding document in T, the prototype system generates one SFV for each spot. As a result, in all spots' SFVs 170,978 words are highlighted by Tweets, 28.3 words for one spot's SFVs in average.

Finally, another task is to identify appropriate value for α in (3.3). The value α is assumed more than 0 without loss of generality. Recall that W_i is the word set contained in the Wikipedia document about spot o_i. Relatively, let W'_k be the word set contained in tweets in season s_k. When $\alpha < 1$, the L^0-norm of SFV $\vec{v}^s_{i,k}$ coincides with $|W_i|$ which is independent of α and k, but when $\alpha = 1$, it coincides with $|W_i \cap W'_k|$ which varies depending on k. It may cause failure of recommendation for some spots which are lack of vocabulary in their Wikipedia documents and have been seldom tweeted. Such spots are defined as **minor** if the L^0-norm of its SFV is smaller than or equal to ℓ. Table 3.1 shows the number of minor spots for given α and k. Although the number of minor ones is 61 when $\alpha < 1$, it exceeds 170 by increasing α to 1. Thus, in the following, the prototype system will restrict our attention to the case of $\alpha < 1$.

Next, consider the variance of SFVs in various α to decide its assignment. Let Ω be the vector space spanned by all SFVs (of all spots). In the following, each vector is normalized by the length in the L^2-norm to have an unit length in Ω. Therefore, each spot o_i is mapped to a point by SFV $\vec{v}^s_{i,k}$ in Ω for each season s_k. This implies that the "intensity" of the variance of SFVs is characterized by

$$\delta_i = \max_{k \neq k'} \{ |\vec{v}^s_{i,k} - \vec{v}^s_{i,k'}| \} \tag{3.4}$$

where $|\cdot|$ denotes the L^2-norm. See Fig. 4.3 for the illustration. δ_i is affected by α and called the **diameter** of o_i hereafter.

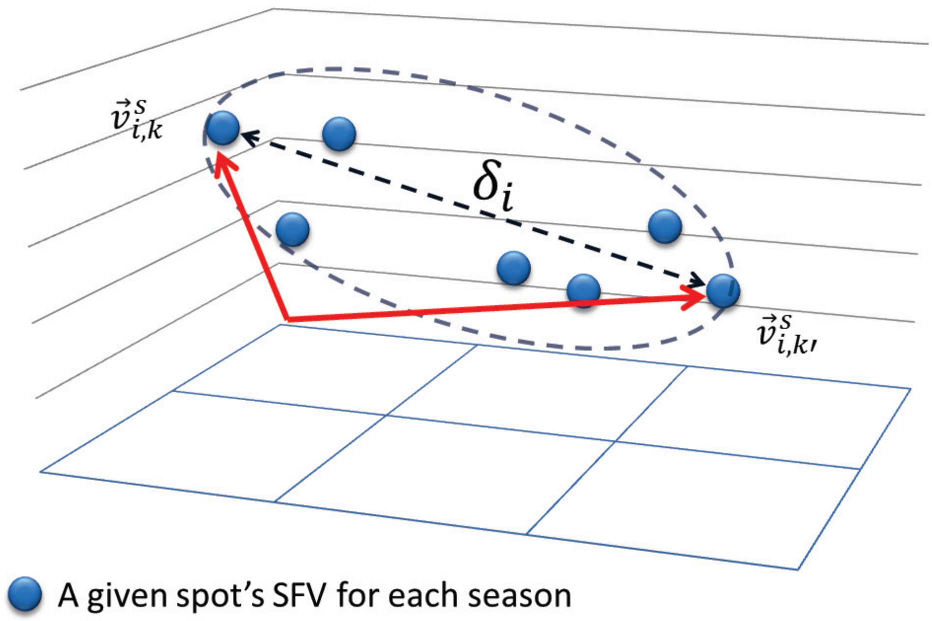


Figure 3.4: Diameter of spot in the vector space.

Fig. 3.5 illustrates the cumulative distributions of δ for $\alpha = 0.99$, 0.995 and 0.999, where the horizontal axis is the length of the diameter and the vertical axis is the accumulative size of the spots having diameter less or equal to a specific value. It indicates that the diameter follow Gaussian distribution and its mean and variance of diameter certainly increase as α increases. In general, a large δ implies that for corresponding spot its seasonal features are well highlighted in SFVs.

Additionally, for each of the word sets $W_i \setminus W'_k$ and $W_i \cap W'_k$, we observe its average weight of words in SFVs. When $\alpha = 0.995$, a comparison is made that the two averages are both at range of 2.3×10^{-4}. As α increased to 0.999 the words in $W_i \setminus W'_k$ are weighted as one-fifth as the ones in $W_i \cap W'_k$ overall. It implies that although in the latter case the seasonal features are well highlighted, static features which do not relate with the seasons are weakened significantly and with failures of characterization. On the other hand, it is observed that the average distance of a spot's BFV to its nearest neighbor is almost 1.2, which is nearly twice of the average

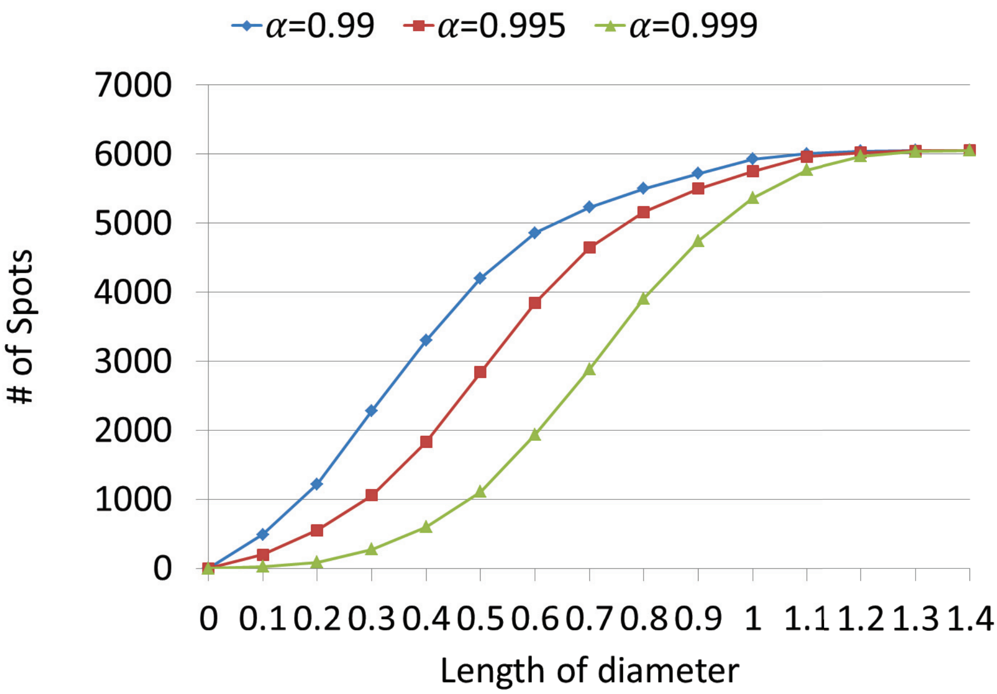


Figure 3.5: Distributions of the diameters of spots with $\alpha = 0.99$, 0.995 and 0.999.

of all spots' δ in the case of $\alpha = 0.995$(almost 0.55). Therefore, α is fixed to 0.995.

3.4 Evaluation

In this section, the effectiveness of the proposed system is evaluated with respect to the following two aspects: (1) whether the usage of Wikipedia documents reduces the noise included in tweets, (2) whether the proposed SFV certainly extracts and characterizes seasonal features from Wikipedia and Twitter, and (3) whether the proposed system effectively provides seasonal tourist spot's recommendation.

3.4.1 Elimination of Noise

Let W denotes the set of words contained in Wikipedia documents and W' denotes the set of words contained in collected tweets. Note that $W' \neq W$ and $W' \cap W \neq \emptyset$

hold in general. For word set W, we identify two word sets for season k as follows: 1) for each word $w \in W$, calculate the gap denoted as g_k for season k which is defined as $TF'_{k,w} - TF'_{k-1,w}$, where $TF'_{k,w}$ denotes the TF weight of word w in tweets for season k. Recall that it has $TF'_{k,w} > 0$ for all highlighted words. 2) select five words with largest g_k as Top-5 emphasized words, and 3) select five words with smallest g_k as Top-5 faded-out words. A similar calculation is conducted for word set W'.

Table 4.3 (resp. 4.3) summarizes the results for W (resp. W') for each season. The difference between them is that Table 4.3 shows representative words in tweets *filtered* by Wikipedia. In fact, Table 4.3 contains more meaningless words than the previous one, such as `follow` and `mutual`, although they commonly contain several seasonal features such as `red leaves`, `year-end party`, `illumination` and `cherry-blossom`. In other words, W contains less meaningless words than W', which implies that the combination of Wikipedia is effective to eliminate noise in tweets.

Table 3.2: Most varying words following the change of seasons in W.

The emphasized words in W with largest g_k for each s_{k-1} and s_k						
rank	Sep.-Oct.	Oct.-Nov.	Nov.-Dec.	Dec.-Jan.	Jan.-Feb.	Feb.-Mar.
1	cafe	saloon	bar	illumination	snow	Nihonbashi
2	bar	moon	cafe	nice	cherry-blossom	fellow
3	wine	red leaves	illumination	hatsumode	sensation	soccer
4	tide	job offer	year-end party	hot	Hotel Tokyo at Chinzanso	nail
5	nice	date	Christmas	girl	many times	snack
The fade-out words in W with smallest g_k for each s_{k-1} and s_k						
1	moon	cafe	red leaves	Christmas	illumination	sensation
2	festival	bar	weather	year-end party	cafe	many times
3	roast meat	wine	wind	estaminet	this year	Hotel Tokyo at Chinzanso
4	soil	Japanese food	temperature	festival	bar	(Kuma)mon
5	clear weather	typhoon	humidity	room	Japanese food	Kanazawa

Table 3.3: Most varying words following the change of seasons in W'.

The emphasized words in W' with largest g_k for each s_{k-1} and s_k						
rank	Sep.-Oct.	Oct.-Nov.	Nov.-Dec.	Dec.-Jan.	Jan.-Feb.	Feb.-Mar.
1	crotch	follow	Umeda	follow	cherry-blossom	follow
2	Ginza	shop	participation	Daiba	snow	mutual
3	cafe	other	card	mutual	lockout	support
4	bar	Mt.Fuji	Christmas	recommendation	Shinjuku	hope
5	card	mutual	Osaka	illumination	Tokyo	do (my) best
The fade-out words in W' with smallest g_k for each s_{k-1} and s_k						
1	Umeda	crotch	red leaves	Christmas	Daiba	cherry-blossom
2	Tokyo	Shinjuku	people	Ginza	Umeda	Tokyo
3	participation	cafe	Kamakura	participation	follow	Chinzansou
4	Asakusa	card	wind	Umeda	Fuji	sensation
5	Mt.Fuji	bar	other	year-end party	illumination	start

3.4.2 Variance of SFVs

Evaluation Methodology

In this section, rather than a direct observation of the difference of SFVs for a given spot, the evaluation of time transition of the similarity of spots is conducted. They are obtained by applying the K-means method [34] to SFVs of all spots. More concretely, if the spots contained in a cluster in season s_k are separated into several clusters in other seasons, those spots are given similar SFVs for s_k and the set of words characterizing the cluster should represent the feature of those spots for s_k. Considering the number of spots is over 6,000, the value of K is setted to 70 in the process of clustering. This evaluation examines the mean of each resulting clusters and focus on several typical ones for the convenience of presentation.

Result

From the resulting clusters, four typical clusters, say C_r, C_i, C_s and C_c, are identified. Their details are summarized in Table 3.4. Note that each of these four clusters is defined only for a specific season. Since `red leaves` and `cherry-blossom` have higher popularities than `illumination` and `snow` in Japan, the corresponding ones also have larger sizes than the others.

Results on the time transition of the similarity of spots are summarized in Fig. 3.6 and 3.7. The upper figure of the first line in Fig. 3.6 shows the result of cluster C_r and the lower one concern with clusters C_i. Results of C_s and C_c are shown in 3.7. In the figures, The bars with various colors represent different clusters and their lengths depend on the number of focused spots. For a given cluster and a season, the number of clusters into which the focused spots separate is also given in parentheses at the bottom. In November, all spots in C_r form a distinct cluster, but in other seasons, they separate into different ones. It indicates that the common features of those spots are *highlighted* in November, although they have various features in other seasons. Similar phenomenon can also be found in other clusters. On the other hand, several spots in C_c are also confirmed to remain in the same cluster through all seasons. It indicates that SFVs of those spots are close with each other in vector

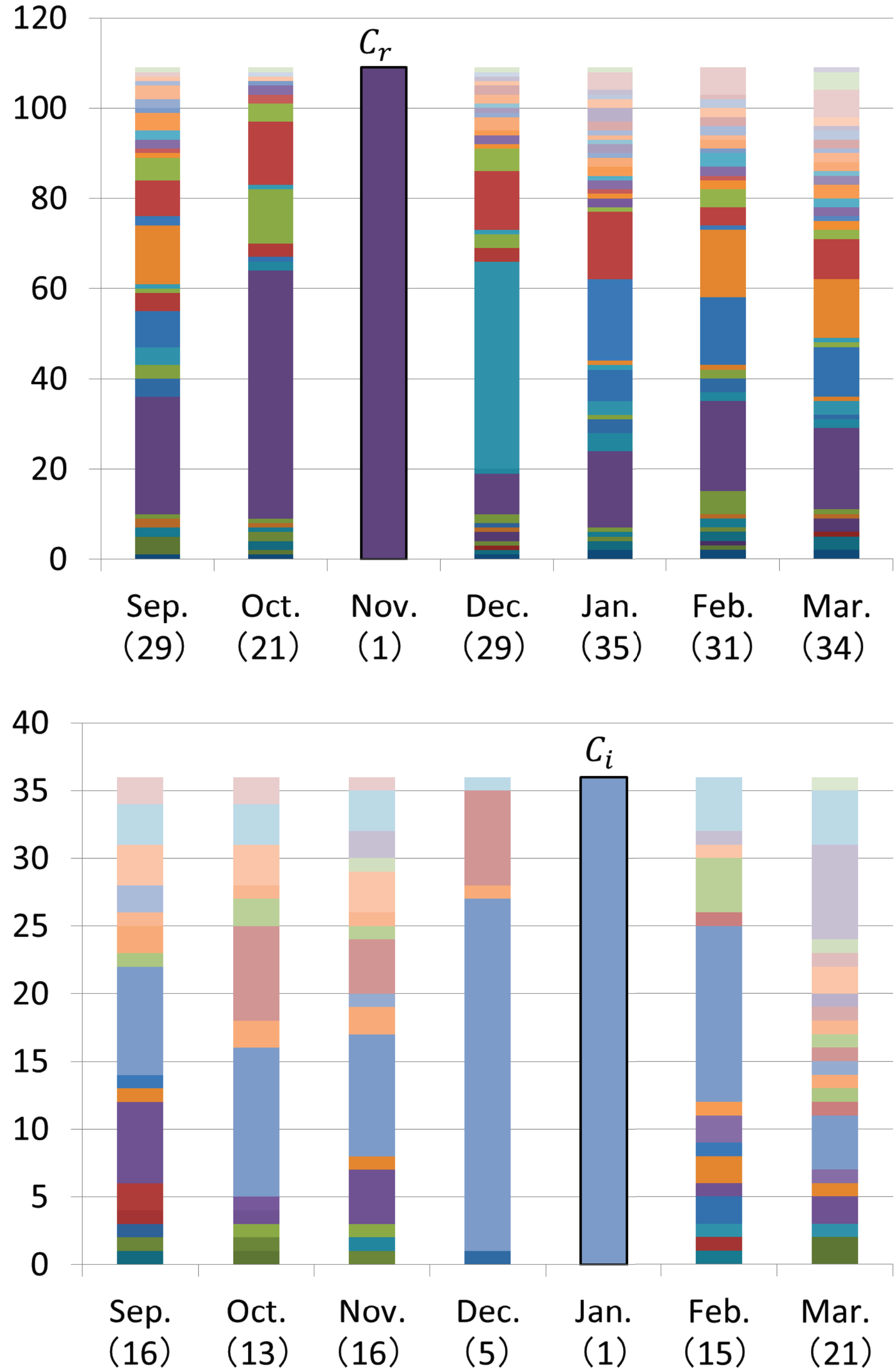


Figure 3.6: The distributions of clusters for the spots that are included in C_r and C_i.

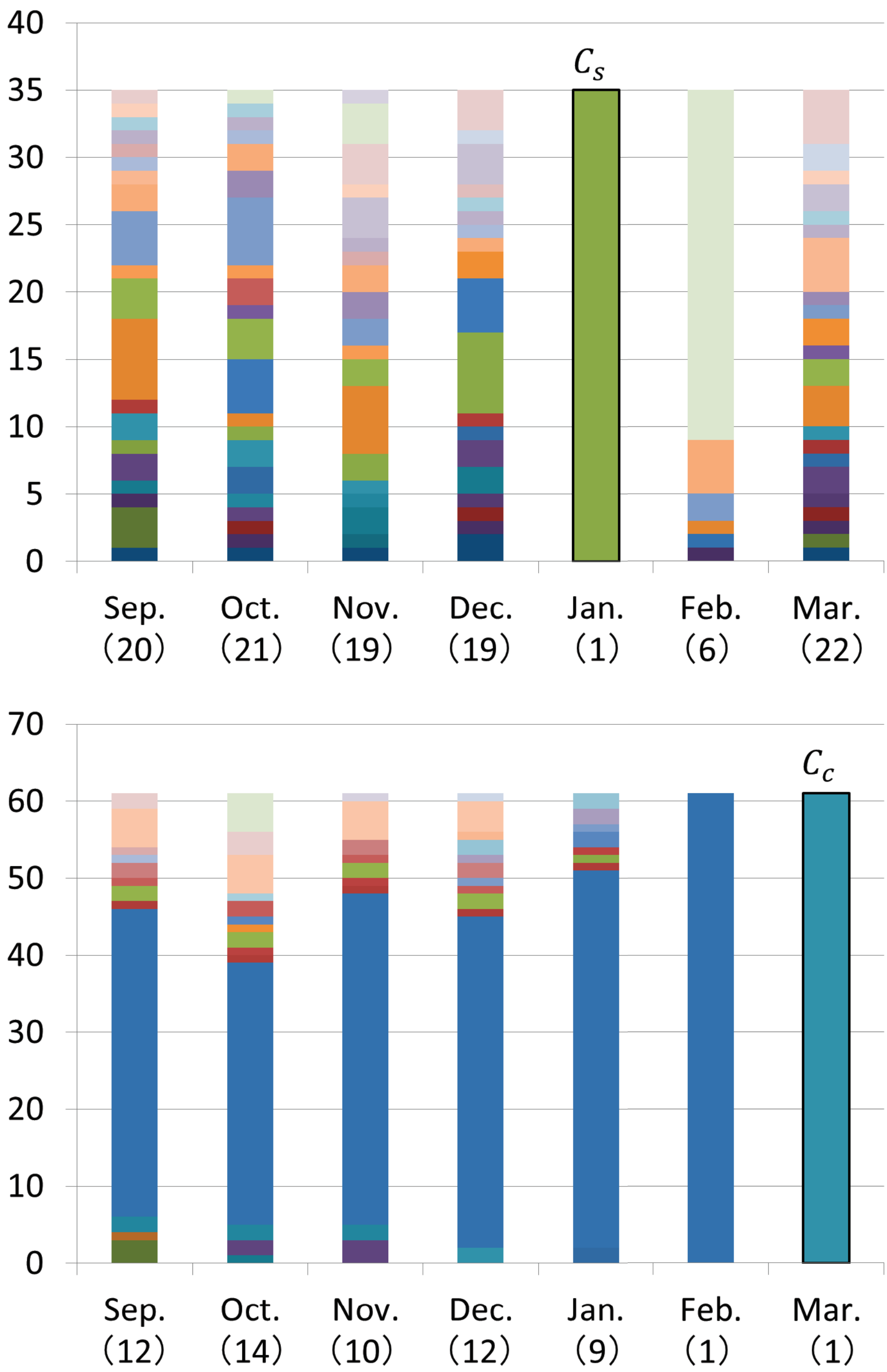


Figure 3.7: The distributions of clusters for the spots that are included in C_s and C_c.

Table 3.4: Details of four typical clusters.

	relevant features	season	# of spots
C_r	`red leaves, waterfall`	November	109
C_i	`illumination,cafe`	January	36
C_s	`snow,event`	January	35
C_c	`cherry-blossom,park`	March	61

space Ω regardless of the transition of seasons.

3.4.3 Impact on Recommendation

Evaluation Methodology

In this subsection, the performance of the proposed seasonal tourism recommender system is evaluated with simulated users. Although a questionnaire evaluation is also conducted in the next subsection, here we aim to compare and observe the difference between recommendations which are generated with considering season (*i.e.*, SFV) and without season (*i.e.*, BFV) in detail. The evaluation focuses on the aforementioned clusters C_r, C_i, C_s and C_c, and regards the mean of SFVs contained in each cluster as the preference of users[4]. In other words, there are four users who are fond of `red leaves`, `illumination`, `snow` and `cherry-blossom`, with the spots in clusters C_r, C_i, C_s and C_c as the answers respectively. The performance as the proposed system is evaluated by analyzing the Top-t spots' recommendation to the designated points for each season k. Such a subset of spots is denoted as Q_t^k hereafter. As comparison, according to the cosine similarity of the corresponding BFVs to the designated points, the Top-t spots (denoted as P_t) are also calculated.

For the mean of a given cluster C, the goodness of a subset X concerned is measured by $|C \cap X|$. Thus, the advantage of using SFV instead of ordinary BFV

[4]A typical scenario assumed in this chapter is that the user designates an interested spot as the preference with a hoped travel season and the system returns a set of spots as recommendation.

Table 3.5: The value of $\xi(30, k)$ with $k \in \{$ Sep, Oct, Nov, Dec, Jan, Feb, Mar $\}$.

	Sep.	Oct.	Nov.	Dec.	Jan.	Feb.	Mar.
C_r	5	5	**5**	5	5	4	4
C_i	−8	4	4	4	**4**	4	4
C_s	6	4	4	5	**8**	3	3
C_c	11	10	11	11	12	13	**13**

Table 3.6: The value of $\xi(t, k)$, where the value of t is fixed to be equal to the corresponding cluster size.

	Sep.	Oct.	Nov.	Dec.	Jan.	Feb.	Mar.
C_r	6	8	**7**	8	9	−4	2
C_i	−15	−5	0	0	**0**	−3	0
C_s	7	6	4	5	**9**	4	4
C_c	8	8	11	10	8	17	**17**

can be measured by calculating

$$\xi(t, k) \stackrel{\text{def}}{=} |Q_t^k \cap C| - |P_t \cap C|, \tag{3.5}$$

which depends on the value of parameter t and the selection of season k.

Result

Table 3.5 summarizes the results for $t = 30$, where the emphasized numbers designate the seasons in which the corresponding clusters are defined (e.g., cluster C_r is defined for November). The result implies that by using SFVs, the proposed system can recommend more spots to fit simulated users' preferences and the effect is maximized when the designated season coincides with the one defining the cluster.

Recall that the value of $\xi(t, k)$ depends on parameter t. Table 3.6 summarizes the results for each cluster, where the value of t is fixed to be equal to the cluster's size, e.g., let $t = 109$ for cluster C_r. Comparing with Table 3.5, in each row a larger gap of $\xi(t, k)$ is observed for each season. It indicated that there are various $Q^k_{|C|}$ for given cluster C. In other words, if the designated season is not relevant with given C, fewer spots that contained in C will be recommended.

3.4.4 Questionnaire Evaluation

Evaluation Methodology

Finally, a two-steps' seasonal tourism questionnaire is conducted to evaluate whether the proposed system can provide a seasonal recommendation of spots in an actual case. Recall that the proposed system extracts user's preference from his visited favorite tourist spot in Section 3.2.2. Therefore, as the first step of the questionnaire, participant selects the spot and the season (*i.e.* month) $s_{k'}$ that he/she wishes to travel. According to his/her selections, system generates seasonal recommendation $Q^{k'}_t$ for $s_{k'}$, where t denotes the number of spots that are included (the lengths of $Q^{k'}_t$). As comparison, we generate two recommendation lists including 1) P_t using BFVs instead of SFVs of spots similarly with the previous experiment, and 2) P'_t using the SFVs as the profiles of spots which are generated with only one season defined, *i.e.* without the separation of time axis into seasons. Therefore, both P_t and P'_t do not characterize the features for a specific season and are independent with $s_{k'}$. They are randomly combined with $Q^{k'}_t$ into one aggregated list of spots as the output to the participant. In the second step, from the list the participant chooses at most 5 spots that he/she wishes to visit in $s_{k'}$ as his answer. Since in following we focus on entire participants and their experimental results, the superscript k' in $Q'^{k'}_t$ is omitted for convenience.

As quantification, this evaluation calculates average precision and recall of all participants' choices from Q_t and comparison as follows:

$$precision = E(\frac{h_t}{t})$$

$$recall = E(\frac{h_t}{H})$$

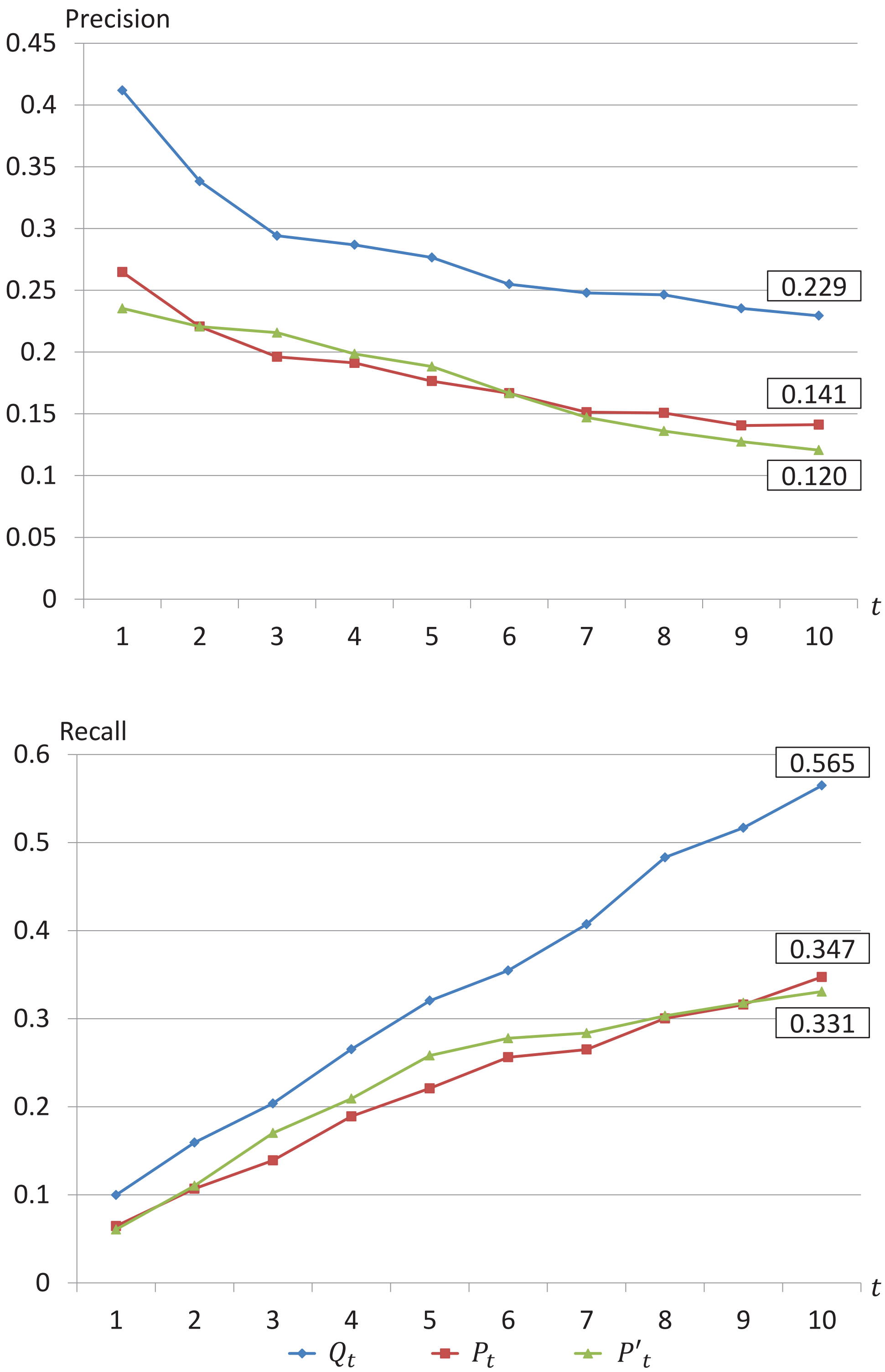


Figure 3.8: Precision and recall of Q_t, P_t and P'_t following t.

Table 3.7: The numbers of trials with each $s_{k'}$ having been chosen. For example, 2 questionnaires are submitted with $s_{k'} = Dec.$.

month	Jan.	Feb.	Mar.	Sep.	Oct.	Nov.	Dec.
# of trials	2	4	8	6	5	7	2

Table 3.8: The numbers of spots chosen from P_t,P'_t and Q_t in answers.

	P_t	P'_t	Q_t	$P_t \cap Q_t$	$P'_t \cap Q_t$	$P_t \cup P'_t \cup Q_t$
$t = 5$	30	32	47	12	10	87
$t = 10$	48	41	78	16	15	134

where h_t is the number of spots having been chosen from the aggregated list by a participant, and H is such number with $t = 10$.

In this evaluation, we have received cooperation from 18 college students major in information engineering. They have chosen 34 spots as their favorite spots overall, *i.e.* 34 trials by 18 participants. Table 3.7 summarizes their favorite spots in various $s_{k'}$.

Result

Table 3.8 shows the detail of participants' choice from Q_t, P_t and P'_t. In either case of t, the number of spots having been chosen from Q_t is the highest. It represents that participants prefer the seasonal recommended spots in Q_t than others. Also note that from the outputted aggregation $P_t \cup P'_t \cup Q_t$, 87 and 134 spots are chosen when $t = 5$ and $t = 10$ respectively (averagely 2.56 and 3.94 in one trial), which contain duplicated spots in Q_t and comparison. More concretely, when $t = 10$, 31 duplicated spots are chosen from $P_t \cap Q_t$ and $P'_t \cap Q_t$, almost a quarter of the entire choices of participants.

The results of precision and recall are given in Fig. 3.8. Both the precision and recall of Q_t are higher than the comparison. Correspondingly, F-measure of Q_t (0.326) is also much higher than P_t (0.201) and P'_t (0.177). It represents that the proposed seasonal recommendation fits users' demand better than the traditional one which solely uses Wikipedia and the one generated without the characterization of seasonal features of spots. More concretely, the variation of P'_t in $t \leq 5$ is similar with Q_t in $6 \leq t \leq 10$. When $t = 5$ its precision is higher than P_t but opposite when $t = 10$. The reason is that although the generation of P'_t uses both Wikipedia and tweets, without the defination of season the seasonal features of spots cannot be extracted and reflected into the recommendation successfully. Therefore, only a few of its top spots are included in lower ranks of Q_t.

3.5 Concluding Remarks

This chapter proposes a seasonal tourism recommender system using Wikipedia and Twitter to provide a list of tourist spots as seasonal recommendation. The effectiveness of the proposed system is experimentally evaluated by detailed observation of seasonal feature vector of spot and questionnaires of users' actual choices of spots. The results of evaluations indicate that SFVs certainly characterize the variable seasonal features of the spots. More concretely, the variance of SFVs follows Gaussian distribution and the similarity of SFVs reflects the similarity of the features of the corresponding spots in a designated season. Further more, the result of questionnaire verifies that the proposed system provides seasonal spots' recommendation which fit user's demand better than comparison in tourism.

A future work is to extend the proposed recommender system to extract and characterize *spatial-temporal* features of the spot. Another issue is to integrate user modeling techniques into proposed recommender system, in order to improve the accuracy of recommendations. On the other hand, we also consider that in some hybrid recommender systems like [36, 102], our proposed method can be used as a component to improve them to achieve a seasonal recommendations. In future, we wish to combine the proposed method with such approaches and evaluate the performance of recommendations.

Chapter 4

Textual Review Enhanced Collaborative Filtering

4.1 Introduction

In this chapter, we focus on matrix factorization (MF)[62, 96, 56], which is is the most famous latent factor-based to provide rating prediction for the user's unpurchased item. It characterizes both items and users by vectors of latent factors, which comprise computerized alternatives to the human created genres. The rating of a user to a specific item is modeled by the inner product of their factors. Using machine learning algorithm, the latent factors can be learnt based on the ratings in the history of feedback.

However, recent researches pointed out that the ignorance of the texual reviews is the major shortcoming of MF and brings it mediocre performance [108, 77, 10]. Figure 4.1 shows two users and their feedback to three products from Amazon. Please note that although the combination of a rating and its related short textual "document" could also be called as a review, we clearly define review to such "document" to avoid ambiguation. In the reviews, each user mentioned many topics in which he is interested. For different products each user mentions different parts of interested topics in the reviews; for a specific product, different users focus on different associated topics. For product B (a music player), user A gives a low rating and points out the bad qual-

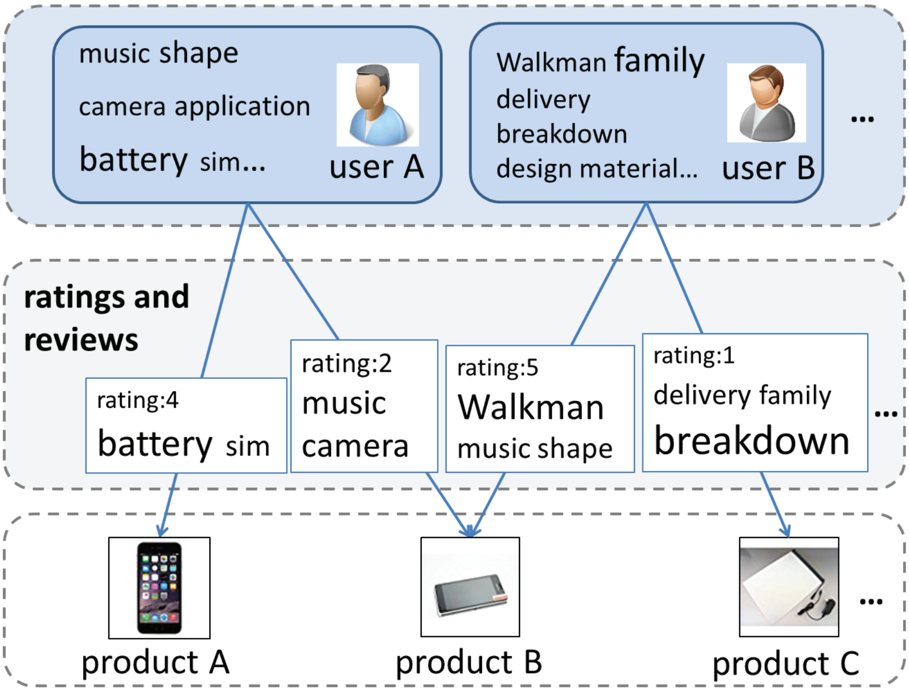


Figure 4.1: A graph that characterizes the actual topics in reviews of two users to three products on Amazon.

ities of music's playing and camera; conversely, user B rates highly and his principal reason is that he is a fan of the player's maker. It represents the fact that although a user often has his own overall opinion (i.e. like or dislike) to obvious properties of product, he/she may focus only on a part of them in the evaluation. While the description of the properties are contained in the textual review, MF cannot realize such unequally treatment since the correspondence of them to latent factors are not defined. To bridge this gap, existing works [77, 10] model the latent topics of reviews with distributions of words, and transform them to latent factors. Unfortunately, the transformation is complicated and makes their methods time consuming in dealing with large-scale data.

In this chapter, in order to solve the issues mentioned above, we propose a new method to predict the rating of user's unpurchased item for recommendation. In order to model the latent topics in the reviews, we train a Latent Dirichlet Allocation (LDA) [14] independently. Each of the topics is assigned to a latent factor. Our idea is to present a new first-order gradient descent method, called Topic Gradient Descent (TGD), which binds the latent topics to latent factors via the training process of MF instead of the transformation. Since a more mentioned topic in a review is considered more importantly when the user rates, its proportion to all topics represent the degree of importance. When iteratively updating its corresponding latent factor in the training, its updating step is dynamically fixed based on such importance. In other words, the importance of topics points out the direction to update the latent factor vectors of users and items. With these trained latent factor vectors, we use the biased MF to predict the ratings.

In our evaluation, we conducted a series of experiments using 11 datasets, including YELP challenge dataset and per-category reviews from Amazon. It evaluates not only the entire performance in the problem of missing rating's prediction, but also the convergence of the squared error of rating prediction in the training. For proposed TGD, the results of the experiment demonstrate that it converges the objective function of MF in training. As an entire rating prediction method, the predicted ratings derives an improvement up to 9.03% in term of MAE, 12.23% in RMSE against simplex MF [62]. It even outperforms state-of-the-art model [10] for recommendation in most of the datasets. Additionally, the proposed method of rating prediction is also demonstrated to have higher accuracy than simplex MF in the prediction of high-scored ratings.

The remainder of this chapter is organized as follows. Section 4.2 describes the problem that we focus and briefly reviews MF and LDA. Section 4.3 describes the detail of proposed method. Section 4.4 represents the method of evaluation and shows its results. Finally, Section 4.5 concludes the chapter with future work.

4.2 Preliminaries

4.2.1 Problem Definition

The problem we study is to accurately predict the ratings of unpurchased items for users based on their history of feedback. Such feedback for an item include a numerical rating in scale of $[a, b]$ (e.g. ratings of one to five stars on Amazon) and a textual correlated review. Suppose in the feedback we have I users and J items overall. The rating made by user u_i ($i \in \{1, \ldots, I\}$) to item v_j ($j \in \{1, \ldots, J\}$) is denoted as $r_{i,j}$. If $r_{i,j}$ exists, it must have a correlated review $d_{i,j}$ written by u_i. Therefore, feedback of a user to an item is a 4-tuple $< u_i, v_j, r_{i,j}, d_{i,j} >$. Note that for a given user u_i and his unpurchased item $v_{j'}$ ($j' \in \{1, \ldots, J\}$), we only predict the unknown rating as $\hat{r}_{i,j'}$ without the unknown review $d_{i,j'}$.

4.2.2 Latent Dirichlet Allocation (LDA)

Topic model is the algorithm used to discover topics in a large set of documents, i.e., corpus. Such topic is a probability distribution over all words. The words associate with a single theme make their corresonding elements bias in the distribution of a topic. Therefore, topic model provides an interpretable dimension reduction of corpus.

LDA [14] is a generative probabilistic topic model of a set of semantic documents. Its basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. Assume there are K topics in corpus, and one topic t_k ($k \in [1, \ldots, K]$) is a probability distribution of all words. For each document $\mathbf{w}_j$ in corpus, the generative process is as follows:

- Choose topic proportion θ_j of $\mathbf{w}_j \sim \mathrm{Dir}(\alpha)$
- For each of the words w_n:
  - Choose a topic assignment $z_{j,n} \sim \mathrm{Multinomial}(\theta_j)$
  - Choose a word $w_n \sim \mathrm{Multinomial}(\beta_{z_{j,n}})$

The process represents the assumption how a document is generated. Overall topic size K is assumed to be known and t_k is shared by corpus. For each document $\mathbf{w}_j$, a specific topic proportion θ_j is provide as its representation.

The objective is to estimate the maximun likelihood of β_k and α to generate the documents of corpus. We approximate them by using an variational EM algorithm[14] which maximizes their lower bound. Further, the parameters θ_j and $z_{j,n}$ can be updated via Gibbs sampling[46] iteratively.

4.2.3 Matrix Factorization (MF)

Biased matrix factorization is an influential approach which maps users and items into a joint latent factor space with arbitrary predefined K dimensions. Accordingly, each user u_i is associated with a vector $U_i \in \mathbb{R}^K$, whose elements measure his/her extent of interest to such factors. On the other hand, vector $V_j \in \mathbb{R}^K$ is associated with a given item v_j, and presents the positive or negative extent of those factors that v_j possesses. The inner product of U_i and V_j represents the interaction of u_i and v_j, and approximates the corresponding rating $r_{i,j}$:

$$r_{i,j} \sim \hat{r}_{i,j} \quad = \quad U_i^T V_j + \mu + b_i + b_j, \tag{4.1}$$

where μ is the average of ratings over all users and items, b_i and b_j denote the observed biases of user u_i and item v_j respectively. Normally, a bias of a given user or item is calculated as the result of subtraction of μ from the average of correlated ratings.

The objective is to learn U_i and V_j by given training set including observed ratings, by minimizing the function of regularized squared error:

$$\begin{aligned}\mathcal{L} = \frac{1}{2}\sum_{i,j}&[c_{i,j}(r_{i,j} - \hat{r}_{i,j})^2 \\ &+\lambda(\|U_i\|^2 + \|V_j\|^2 + {b_i}^2 + {b_j}^2)],\end{aligned} \tag{4.2}$$

where λ is the parameter to control the regularization to avoid over-fitting in learning, and $\|\cdot\|^2$ denotes the L^2 norm. $c_{i,j}$ is the confidence parameter of rating $r_{i,j}$, which indicates how much we trust it. A large $c_{i,j}$ should be assigned for some deliberate ratings, and a small $c_{i,j}$ for the ones that do not deserve seriously treatment such as advertisings and fakes.

A typical way to minimize the objective function (4.2) is to use gradient descent algorithm [62, 95]. It calculates the gradients of U_i and V_j for every given rating $r_{i,j}$ as

$$\begin{aligned} g_{U_i} &= -c_{i,j}(r_{i,j} - \hat{r}_{i,j})V_j + \lambda \cdot U_i \\ g_{V_j} &= -c_{i,j}(r_{i,j} - \hat{r}_{i,j})U_i + \lambda \cdot V_j, \end{aligned} \tag{4.3}$$

and updates them to the inverse direction of gradients iteratively. The updating step is often unique and controlled by a constant learning rate. Since a big learning rate causes divergence of the objective function and a small one may result in slow learning, it is crucial to find a proper learning rate [113].

4.3 Proposed Method

In this section, we present our proposed method, whose structure is shown in Figure 4.2. With given history of tuples of feedback, the first task for us is to derive the topics from each review. As the pre-processing, we use LDA [14], which is a probabilistic generative latent topic model of a set of semantic documents called corpus. Its idea is that each latent topic is characterized by a distribution over words, and a document is a random mixture over such topics. We take each review as a single document, and all reviews as the corpus D. Assume that there are K topics overall in D, which are shared by all documents. A topic is denoted by t_k with $k \in \{1, \ldots, K\}$. For a review $d_{i,j} \in D$, its topic distribution is denoted by $\theta_{i,j}$, which is a K-dimensional stochastic vector. Therefore, each of the elements $\theta^k_{i,j}$ represents the proportion of corresponding topic t_k having been mentioned in $d_{i,j}$. Following the method presented in [46], we independently train the LDA model for D and infer $\theta_{i,j}$ for each review $d_{i,j}$ by Gibbs Sampling.

Next, we use MF to model the ratings and further to predict the missing ones for users. The difficulty comes from the link of the topic distributions of reviews and latent factors without a complicated transformation between them. We propose

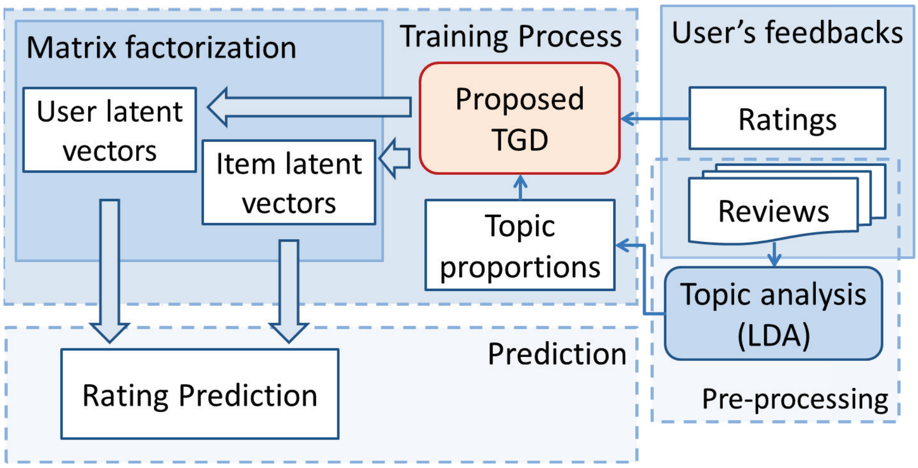


Figure 4.2: The constructure of proposed method.

a new first-order gradient descent method named Topic Gradient Descent (TGD), to correlate them through the training process of MF. Since the reviews provide an efficient tool for the users to explain their ratings, important topics are often mentioned much in the reviews. Therefore, the topic distribution $\theta_{i,j}$ represents the importance of degree of topics in the evaluation of user u_i to item v_j, rather than his/her preference on v_j. In other words, when $\theta_{i,j}^k = 0$, t_k is not worth to mention for u_i and have no impact on the evaluation of v_j. Assume that the number of latent factors is equal to the number of topics, and topic t_k corresponds and interprets the elements U_i^k and V_j^k of U_i and V_j. The key idea is to use $\theta_{i,j}^k$ to affect the learning of U_i^k and V_j^k in the training process of MF. To be more specific, a given error of the rating prediction $r_{i,j} - \hat{r}_{i,j}$ is a linear combination of $\theta_{i,j}$, U_i and V_j. With the denotation of gradients g_{U_i} and g_{V_j} in (4.3), we write the updating equation for U_i and V_j as

$$\begin{aligned} U_i &\leftarrow U_i - \gamma H_{i,j} \cdot g_{U_i} \\ V_j &\leftarrow V_j - \gamma H_{i,j} \cdot g_{V_j}, \end{aligned} \tag{4.4}$$

where γ is a pre-defined constant, and $H_{i,j}$ is a $K \times K$ diagonal matrix with $\theta_{i,j}$ as the diagonal elements. $H_{i,j}$ is together with γ to be the learning rate, which assigns various updating steps for each latent factor. For the topics which have high importance and generate much error, their corresponding latent factors are updated with large steps. In contrast, factors of unimportant topics are updated with small steps in every epoch of training. When U_i and V_j are intialized with vectors of extremely small constant, such factors will remain the initial values and further have little impact on the rating prediction.

Algorithm 1 Topic Gradient Descent

Require: $\theta_{i,j}$ for $d_{i,j} \in D$
Initialize U_i and V_j with vectors of unique value and α with constant, set $s = 1$.
while The objective function (4.2) has not converged **do**
  Calculate $\gamma \Leftarrow \alpha \cdot s^{-1/2}$
  for $d_{i,j} \in D$ **do**
    Compute gradients g_{U_i} and g_{V_j}
    Apply update $U_i^{s+1} \Leftarrow U_i^s + \gamma H_{i,j} \cdot g_{U_i}$
    Apply update $V_j^{s+1} \Leftarrow V_j^s + \gamma H_{i,j} \cdot g_{V_j}$
  end for
  $t \Leftarrow t + 1$
end while

Although we have correlated latent factors with topics and realized their unequal treatment, an issue remained to be solved is that the convergence of the objective function (4.2) may be slow. Since the average of $\theta_{i,j}^t$ is $1/K$, the average of updating step reduces to $1/K$ of the traditional gradient descent method [1] . Let $s \in [1, +\infty]$ be the timestamp that represents the epochs in training. Following the idea of previous effort [115], we introduce the timestamp into the learning rate. Instead of a constant, γ is re-defined with a function of the timestamp s:

$$\gamma = \frac{\alpha}{\sqrt{s}} \tag{4.5}$$

where α is an arbitrary predefined constant. γ is inverse to s so that it reduces following the growth of s. Therefore, U_i and V_j are updated with large steps at the beginning of training, and slightly adjusted to find the most proper values at last.

[1]Brandyn Webb. Netflix update: Try this at home. `http://sifter.org/simon/journal/20061211.html`, 2006

We present TGD method in Algorithm 1, where U_i^s and V_i^s denote the values of U_i and V_j in epoch s. Note that although the form of updating is similar to second-order Newton's method, we only use first-order information of U_i and V_j. Let $|D|$ denotes the number of reviews in corpus D. In each epoch, the time complexities for the calculation of gradients and update of U_i and V_j are $\mathcal{O}(|D| \cdot K)$. Also assume that in the epoch T the objective function converges. Therefore, the time complexity of TGD remains $\mathcal{O}(T \cdot |D| \cdot K)$, the same as existing first-order method.

With the MF model trained by TGD, for a given user u_i and an unpurchased item $v_{j'}$, we calculate the rating prediction $\hat{r}_{i,j'}$ following (4.1).

Table 4.1: The description of datasets used in experiments.

dataset	# tuples	avg.rating	# users	# items	sparsity	avg.words	since
YELP	1933	4.059	341	286	0.019	66	Jul. 2014
Movies and Videos	1400	3.954	172	145	0.056	76	Sep. 2012
Tools and Home Improvement	1152	4.375	147	125	0.063	46	Sep. 2012
Baby	1096	4.187	130	112	0.075	39	Feb. 2013
Toys and Games	385	3.977	36	36	0.297	34	Oct. 2013
Cell Phones and Accessories	1084	4.565	103	94	0.112	45	Feb. 2014
Beauty	913	4.772	103	44	0.201	36	May. 2014
Video Games	949	4.080	137	78	0.089	71	May. 2013
Sports and Outdoors	1304	4.364	178	138	0.053	38	Jun. 2012
Grocery and Gourmet Food	647	4.136	87	66	0.113	36	Oct. 2013
Digital Music	1078	4.079	113	116	0.082	67	Oct. 2008

4.4 Evaluation

In this section, we conduct the evaluation with three perspectives: 1) whether the proposed TGD method makes the objective function (4.2) rapidly converge; 2) how the parameters impact the performance of the proposed method; 3) how is the performance of proposed method comparing with MF and the state-of-the-art model for recommendation.

4.4.1 Datasets and Implementation

In evaluation, we use several datasets have been driven from YELP [2] and Amazon [78]. They are filtered by the following constraints to have the tuples of feedback that: 1) their reviews have at least 10 words; 2) each of the users has reviewed at least 5 items; 3) each of the items has purchased by at least 5 users. Additionally, since in the following comparison with existing method [62, 10] large datasets make the experiments time consuming, we cut each of them by the publishing date of the feedback. For *YELP challenge dataset*, we only utilize the feedback from State of Arizona and Nevada for the sparsity of data. Discard of the stop words and stemming are also conducted for each review. With these processes, table 4.3 shows their statistic including the number of users, items and tuples of feedback contained. The third and seventh columns show the average of rating and number of words in a review in the datasets respectively. The sparsity of a dataset is calculated as #tuples of feedback /(# users $\times$ # items).

For each dataset, we randomly take its 80% as training set, and the rest as testing set to conduct the experiments.

4.4.2 Convergence of Topic Gradient Descent

For each of the training sets, we train the proposed method to observe the sum of squared error of rating prediction in each epoch. Considering the total number of the reviews in datasets, parameters K and λ are fixed to 20 and 0.01 respectively. The

[2] `http://www.yelp.com/dataset_challenge`

latent factors in U_i and V_j are initialized to be unique values of 0.001. As comparison, we also train MF by the method presented in [62], with its K and λ fixed with the same values as our proposed method. Different with the proposed method, factors in U_i and V_j are initialized by randomly generated values following the zero mean gaussian distribution of $\mathcal{N}(0, \lambda^2)$. In order to guarantee the fairness, we set the confidence parameter $c_{i,j}$ to 1 if $r_{i,j}$ exists, and 0 otherwise for both the proposed method and MF.

For the their typical results, we show the results in the first 500 epochs for dataset *Video Games*, and 150 epochs for *Movies and Videos* in Figure 4.3. The parameter α in (4.5) is fixed from 1.0 to 1.3, and the learning rate of MF is set to a general value of 0.03. For both of the datasets, MF reaches lower levels of squared error than the proposed method. For *Video Games*, the proposed method reduces the squared error more slowly than MF, which is opposite to the result of *Movies and Videos*. Especially in *Movies and Videos*, $\alpha = 1.3$ is not a proper assignment since the squared error divergences early. Considering that for a given tuple, the updating steps of latent factors depend on the topic distribution of its review, we calculated and observed the standard deviation (SD) of topic distribution for each review. As a consequence, the average SD of *Movies and Videos* (0.067) is much higher than *Video Games* (0.029). It indicates that the speed of convergence depends on the dispersion of the topics' proportions in the reviews.

4.4.3 Impact of Parameters

Since the problem is to predict the ratings of the users to their unpurchased items, the performance of the proposed method is evaluated by observing the accuracy of predictions. For given feedback from the testing set, we compare the rating prediction $\hat{r}_{i,j}$ with its actual rating $r_{i,j}$. As quantification, we use mean absolute error (MAE) and root mean square error (RMSE) which are calculated as follows:

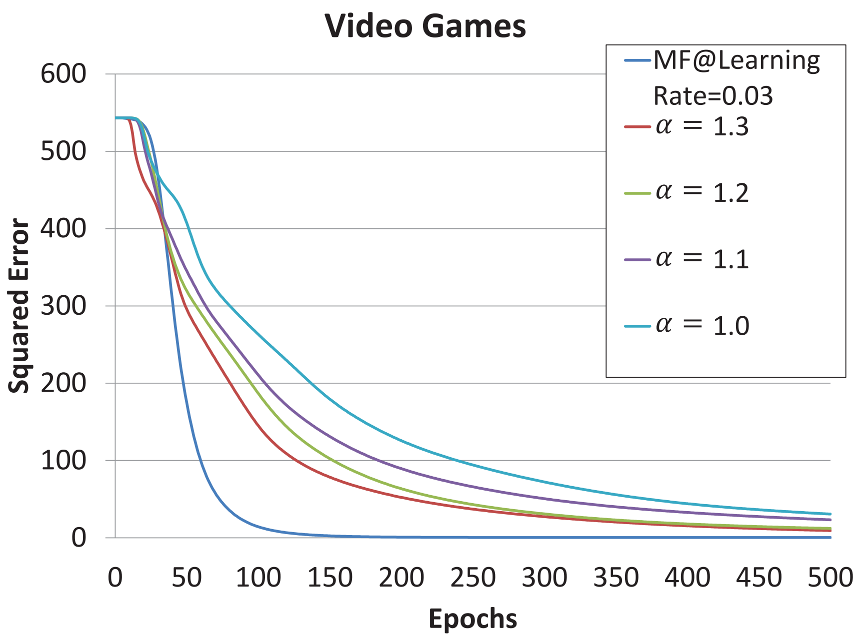
$$MAE = \frac{1}{N} \sum_{i,j}(|r_{i,j} - \hat{r}_{i,j}|),$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i,j}(r_{i,j} - \hat{r}_{i,j})^2},$$

where N denotes the number of data tuples in the testing set, and $|\cdot|$ denotes the absolute value. In general, RMSE is more sensitive than MAE for large error of prediction. The assignment of parameters and initialization follows the previous experiment in Section 4.4.2. For each of the training sets, the proposed method is trained until the objective function converges.

Figure 4.4 shows the performance of the proposed method with α changed from 1.0 to 1.3. For *Video Games* RMSE is stable for all cases of α. On the other hand, RMSE of *Movies and Videos* is over 0.6 when $\alpha = 1.3$, and reduces to roughly 0.3 when $\alpha \in \{1.2, 1.1, 1.0\}$. Combining the results of the previous experiment, it is indicated that the divergence of objective function in learning further affects the performance. In other words, the performance of the proposed method is stable to small enough α. In order to avoid such affection, we fix α to 1.2 to conduct following experiments.

Figure 4.5 shows the performance with K changed from 10 to 50. Recall that K denotes the number of overall topics, also the dimension of U_i and V_j. For *Video Games*, MAE and RMSE vary in parallel following K. When $K = 20$ the proposed method has the best performance and when $K \geq 40$ the performance declines. In the case of *Movies and Videos*, although MAE is stable, a trough of RMSE is observed when $K = 25$. Therefore, in order to achieve the best performance, K should be fixed into a proper range which depends on the dataset. An assignment of too small or too large values makes the performance declines.



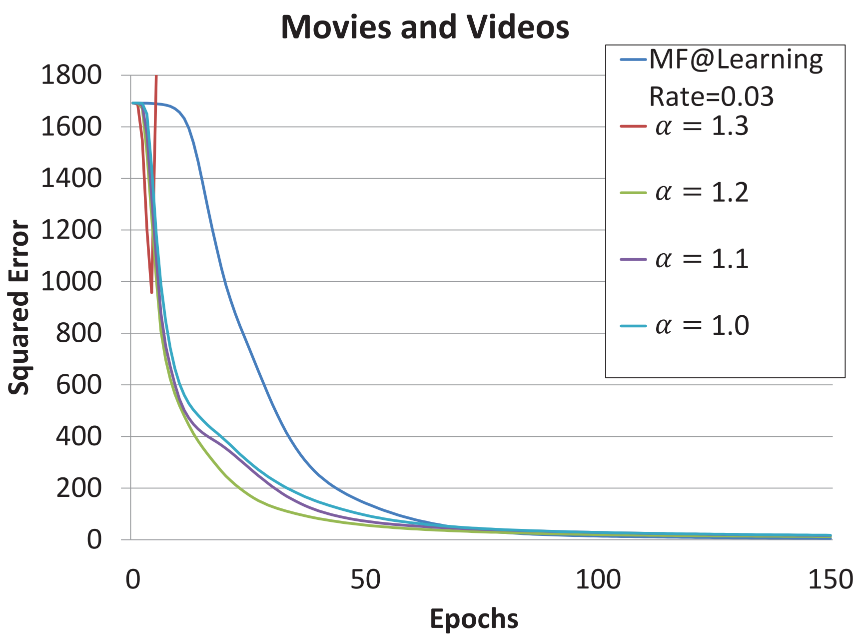


Figure 4.3: The squared error of rating prediction in the training of MF with using proposed TGD and existing method.

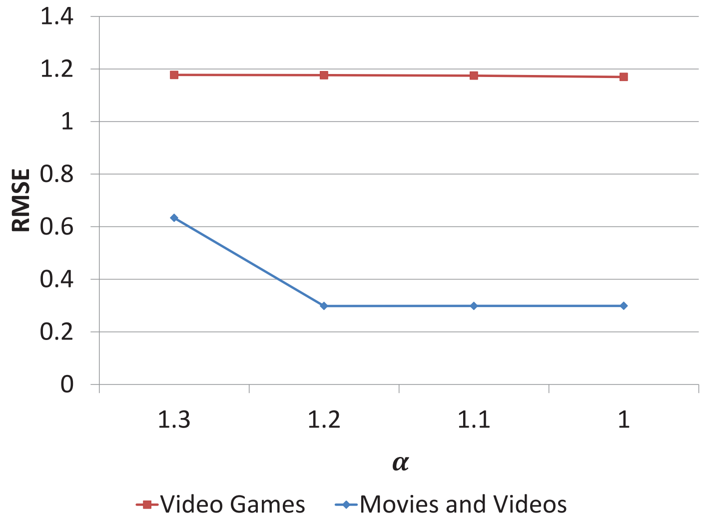


Figure 4.4: The performance of *Video Games* and *Movies and Videos* in term of RMSE with α is fixed to 1.3, 1.2, 1.1 and 1.0.

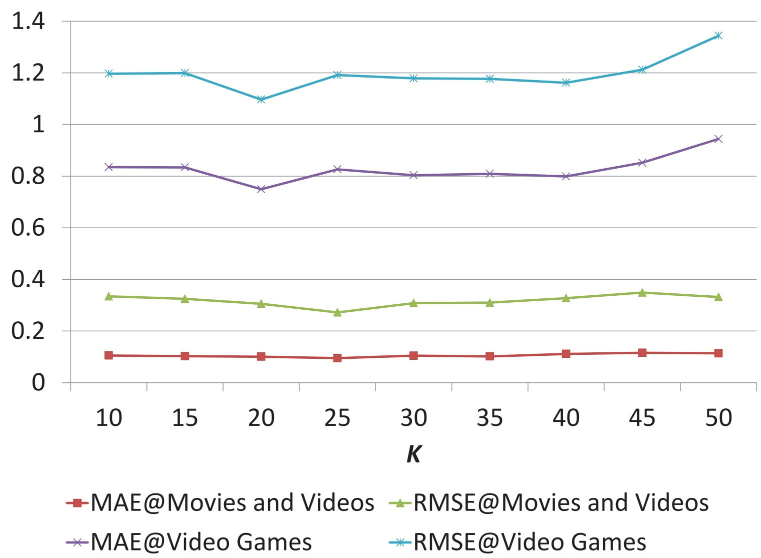


Figure 4.5: The performance of *Video Games* and *Movies and Videos* with K fixed from 10 to 50.

Table 4.2: The performance of MF, TopicMF and the proposed method,in term of MAE with $K = 20$ on all datasets.

	MAE			Improvement in MAE	
Dataset	MF	TopicMF	Proposed Method	vs MF	vs TopicMF
YELP	0.772	0.771	**0.723**	6.40%	6.25%
Movies and Videos	0.097	0.182	<u>**0.088**</u>	8.86%	51.45%
Tools and Home Improvement	0.648	0.648	<u>**0.622**</u>	3.96%	3.90%
Baby	0.693	0.703	<u>**0.660**</u>	4.72%	6.11%
Toys and Games	0.587	0.586	<u>**0.568**</u>	3.18%	3.00%
Cell Phones and Accessories	0.472	0.453	<u>**0.444**</u>	5.98%	2.03%
Beauty	0.238	0.225	**0.221**	7.48%	1.85%
Video Games	0.817	0.791	<u>**0.760**</u>	7.01%	4.01%
Sports and Outdoors	**0.607**	0.633	0.643	-5.87%	-1.50%
Grocery and Gourmet Food	**0.708**	0.738	0.739	-4.33%	-0.15%
Digital Music	0.765	0.749	**0.696**	9.03%	7.14%
Average	0.582	0.589	<u>**0.560**</u>	3.77%	4.87%

Table 4.3: The performance of MF, TopicMF and the proposed method, in term of RMSE with $K = 20$ on all datasets.

	RMSE			Improvement in RMSE	
Dataset	MF	TopicMF	Proposed Method	vs MF	vs TopicMF
YELP	1.095	1.001	<u>**0.969**</u>	11.50%	3.19%
Movies and Videos	0.321	0.421	<u>**0.306**</u>	4.60%	27.18%
Tools and Home Improvement	0.934	0.925	<u>**0.891**</u>	4.66%	3.72%
Baby	0.876	0.892	<u>**0.850**</u>	2.95%	4.65%
Toys and Games	0.839	0.814	<u>**0.758**</u>	9.70%	6.93%
Cell Phones and Accessories	0.685	0.674	<u>**0.638**</u>	6.82%	5.32%
Beauty	0.401	0.385	<u>**0.380**</u>	5.28%	1.19%
Video Games	1.173	1.115	<u>**1.061**</u>	9.48%	4.77%
Sports and Outdoors	**0.844**	0.867	0.902	-6.86%	-3.99%
Grocery and Gourmet Food	**0.959**	0.986	0.970	-1.12%	1.64%
Digital Music	1.107	0.980	**0.971**	12.23%	0.83%
Average	0.839	0.824	<u>**0.791**</u>	5.82%	3.99%

Table 4.4: The performance in term of MAE of MF, TopicMF and the proposed method with $K = 40$ on all datasets.

Dataset	MAE			Improvement of MAE	
	MF	TopicMF	Proposed Method	vs MF	vs TopicMF
YELP	0.770	0.761	**0.722**	6.28%	5.13%
Movies and Videos	**0.101**	0.196	0.109	-7.36%	44.45%
Tools and Home Improvement	0.630	**0.622**	0.635	-0.83%	-2.06%
Baby	**0.667**	0.671	0.724	-8.55%	-7.80%
Toys and Games	**0.587**	0.613	0.611	-4.17%	0.29%
Cell Phones and Accessories	0.463	**0.471**	0.498	-7.70%	-5.71%
Beauty	0.226	**0.220**	0.250	-10.56%	-13.27%
Video Games	0.813	**0.760**	0.859	-5.72%	-13.12%
Sports and Outdoors	**0.605**	0.617	0.628	-3.81%	-1.82%
Grocery and Gourmet Food	0.681	**0.701**	0.760	-11.67%	-8.41%
Digital Music	0.771	0.713	**0.682**	11.58%	4.37%
Average	**0.574**	0.577	0.589	-2.61%	-2.08%

Table 4.5: The performance in term of RMSE of MF, TopicMF and the proposed method with $K = 40$ on all datasets.

Dataset	RMSE			Improvement of RMSE	
	MF	TopicMF	Proposed Method	vs MF	vs TopicMF
YELP	1.091	0.997	**0.970**	11.12%	2.77%
Movies and Videos	0.335	0.436	**0.323**	3.52%	25.98%
Tools and Home Improvement	0.911	<u>**0.891**</u>	0.923	-1.35%	-3.68%
Baby	**0.851**	0.855	0.934	-9.68%	-9.19%
Toys and Games	0.839	**0.832**	0.766	8.77%	7.95%
Cell Phones and Accessories	**0.672**	0.688	0.723	-7.51%	-5.12%
Beauty	0.388	<u>**0.380**</u>	0.444	-14.29%	-16.73%
Video Games	1.154	<u>**1.061**</u>	1.239	-7.31%	-16.71%
Sports and Outdoors	<u>**0.830**</u>	0.840	0.862	-3.87%	-2.66%
Grocery and Gourmet Food	<u>**0.928**</u>	0.946	1.019	-9.83%	-7.72%
Digital Music	1.113	<u>**0.933**</u>	0.958	13.90%	-2.74%
Average	0.828	**0.805**	0.833	-0.52%	-3.40%

4.4.4 Performance in Recommendation

According to the previous experimental results, we set K to 20 and 40 to conduct a detailed evaluation to the performance in rating prediction. Except MF, we also implemented TopicMF [10] which is an extension of HFT [77] as comparison. Following the setup of their experiments, we set $\lambda = 1$, $c_{i,j} = 1$ if $\exists r_{i,j}$ and $\lambda_u = \lambda_v = \lambda_B = 0.001$. Since the training of TopicMF is time consuming (3 to 5 minutes for a training set with a scale of 1,000 reviews for one epoch), we train it with 100 epochs and report its performance.

Table 4.3 summarizes the results of all datasets, with the best performance emphasized in boldface for each of the datasets. The last lines of the two tables present the average of MAE and RMSE. The improvement of the proposed method is calculated and presented in last four columns both for each dataset and average performance. When $K = 20$, the proposed method shows the best performance in term of RMSE on 10 datasets. Comparing with MF, the improvement of the proposed method is 3.77% in MAE, and 5.82% in RMSE in average. It indicates that the proposed method is effective in reducing the decisive failure of prediction. Especially on *YELP*, *Movies and Videos*, *Video Games* and *Digital Music*, the proposed method gains the improvement from 6.40% up to 9.03% in MAE. Also referring Table 4.3, averages of words in one review of these four datasets are all more than 65. It represents that their reviews are written in more detail than other datasets. Therefore, the topics could be more clearly inferred to make the latent factors well trained in learning. On the other hand, the proposed method also outperforms TopicMF on 11 datasets, with the improvement of 4.87% in MAE and 3.99% in RMSE in average. When $K = 40$, the performance of the proposed method declines on most of the datasets except *Digital Music* and *Sports and Outdoors*. It represents that for such datasets, setting K to 40 makes the topics not clearly derived, further affects the performance. In term of RMSE, the improvement also reduces to -0.52% comparing with MF, and -3.40% with TopicMF in average.

Additionally, we underline the best performance among the approaches in both cases of K for each dataset. For example, the proposed method obtains the smallest RMSE for YELP dataset, which is underlined in the first table of $K = 20$. Overall,

Table 4.6: The precision of the proposed method and MF in prediction of 5 star's rating.

Dataset	MF	Proposed Method	Improvement
YELP	0.555	0.594	6.977%
Movies and Videos	0.984	0.987	0.334%
Tools and Home Improvement	0.6	0.622	3.704%
Baby	0.439	0.449	2.326%
Toys and Games	0.773	0.818	5.882%
Cell Phones and Accessories	0.734	0.741	0.98%
Beauty	0.845	0.865	2.29%
Video Games	0.545	0.571	4.918%
Sports and Outdoors	0.689	0.722	4.808%
Grocery and Gourmet Food	0.579	0.592	2.273%
Digital Music	0.559	0.581	3.846%

the proposed method obtains the best performance on 8 datasets in term of RMSE, 7 datasets in MAE. It is also observed that only two of them is in the case of $K = 40$. Therefore, proper assignment of K (20 for most of the datasets) guarantees the proposed method to gain better performance than two existing methods.

In practical application, if the predicted rating of an unpurchased item is high, such item may be a future recommendation to the given user. Therefore, we particularly evaluate the accuracy of predictions to the actual ratings with the highest score. Considering that both in YELP and Amazon a user evaluates an item up to 5 stars, we take the feedback in the testing set with 5 stars' ratings as objective ones. The prediction is successful if the predicted rating locates in $[4.5, \infty)$. The precision is calculated as the proportion of successful predictions to the objective ratings.

Table 4.6 shows the precision of the proposed method and MF on each dataset with $K = 20$. For example, to 5 stars' ratings in YELP dataset, 55.5% of them are predicted in the range of $[4.5, \infty)$ by MF, and 59.4% by the proposed method respectively. Among all datasets the improvement of the proposed method is up to 6.977%. For *Movies and Videos*, since the precision of both MF and the proposed method is at a high level of more than 0.98, correspondingly the improvment is the slightest (0.334%). Also note that although the performance of the proposed method is worse than MF in *Sports and Outdoors* and *Grocery and Gourmet Food* (line 9 and line 10 in Table 4.3), the precision is higher than MF. It demonstrates that the proposed method has higher accuracy in the prediction of such highest ratings than MF.

4.5 Concluding Remarks

In this chapter, we propose a new method to predict ratings for recommendation, including a topic gradient descent method (TGD) for the MF model. From the given textual reviews, their topics are derived by Latent Dirichlet Allocation model. Using such topics, in the learning of the proposed method the latent factors of the users and items are iteratively updated by dynamically assigned updating steps. In the evaluation, we conduct a series of experiments utilizing 11 datasets, including YELP challenge dataset and per-category Amazon reviews. Firstly, the experimental results verified that the TGD certainly converges the squared error of the rating prediction. Secondly, it also shows that the proposed method outperforms MF in the recommendation. The accuracy of rating prediction improves up to 12.23% in term of RMSE, and 5.82% on average in all datasets. Comparing with TopicMF which is a state-of-the-art recommendation model for recommendation, it also achieves a superiority of performance. Finally, the proposed method is demonstrated to have higher accuracy than MF in the prediction of high-scored ratings, which is considered as an ordinary scene of recommendation.

In the futrue, we intent to develop a mechanism to automatically search the proper assignment of parameters corresponding with the given dataset. On the other hand, we hope to evaluate the ability to describe the predicted ratings by the learnt latent factors and derived topics. Not only for MF, we also plan to apply the proposed TGD method to tensor factorization to extend it as an optimization to general latent factor based model.

Chapter 5

Concluding Remarks

5.1 Summary

In this thesis, we studied the utilization of word of mouth in recommender systems, in order to improve their performance.

In chapter 1, we introduced the overview of recommender system, including their purpose and the main techniques. Additionally, we also described the shape of word of mouth, and pointed out the two types of information where we focus our attention: 1) the user-generated tweets in Twitter and 2) the textual reviews included in users' feedbacks.

In chapter 2, we went through the related works of content-based and collaborative filtering recommender system. For the former, we concentrated our attention on tourism domain, to present the structure of relevant systems including the techniques applied. For the latter, we reviewed the both the primary and state-of-the-art model-based approaches of recommendation, and presented their advantages and drawbacks.

In chapter 3, we focused on using Twitter and Wikipedia on content-based tourism recommender system, to realize a seasonal recommendation of sightseeing spots. For the spots, we proposed a generation of seasonal feature vectors to characterize their seasonal variant features. By using such vectors as the profiles of spots, we developed a new recommender system which provides a list of spots to fit both the user's preference and the travel season. Additionally, we also discussed the ignorance of the noise included in tweets. In experiments, the effectiveness of the proposed system is especially evaluated by questionnaire of participants.

In chapter 4, we proposed a method to predict ratings of the user's unpurchased items for recommendation, including a topic gradient descent method (TGD) for the matrix factorization (MF) model. From the given textual review in the feedback, its topics are derived by Latent Dirichlet Allocation model. Using such topics, in the learning of the proposed method the latent factors of the users and items are iteratively updated by dynamically assigned updating steps. In the evaluation, we conduct a series of experiments utilizing 11 datasets, including YELP challenge dataset and per-category Amazon reviews.

5.2 Contributions

The contributions of this thesis are summarized as follows:

- We proposed a generation of seasonal feature vectors (SFVs) for sightseeing spots, and a new tourism recommender system to provide a list of sightseeing spots with the awareness of the user's travel season. The results of evaluations indicate that the proposed SFVs certainly characterize the variable seasonal features of the spots. More concretely, the variance of SFVs follows Gaussian distribution and the similarity of SFVs reflects the similarity of the features of the corresponding spots in a designated season. For the proposed system, the result of questionnaire verifies that its seasonal recommendation fit users' demand better than the comparison.

- We propose topic gradient descent method (TGD) for the MF model. By using TGD, we further proposed a method of rating prediction of the user's unpurchased item for recommendation. Firstly, the experimental results verified that TGD certainly converges the squared error of the rating prediction. Secondly, it also shows that the proposed method of rating prediction outperforms MF in the recommendation. Especially, the proposed method is demonstrated to have higher accuracy than MF in the prediction of high-scored ratings, which is considered as an ordinary scene of recommendation. Even comparing with state-of-the-art recommendation model of TopicMF, it also achieves a superiority of performance.

5.3 Future Work

Although we applied several techniques in the analysis of word of mouth, they are term level approaches and with limitations, such as the cost of time in dealing with large volume of information, and the tolerance of noise. As the development of researches in Natural Language Processing, more powerful tools or techniques (e.g. ontologies, thesaurus analysis or word2Vec) are expected to be applied. On the other hand, the geometry information of tweets should also be used to realize a geo-temporal recommendation. For example, machine learning techniques, such as multi-dimensional Gaussian regression can be used to model the relationship of topics and geometry and temporal information. In the decision of tourism recommendation, we should also apply collaborative filtering techniques to develop a hybrid recommender system, and further improve the performance.

Bibliography

[1] Rahim A Abbaspour and Farhad Samadzadegan. Time-dependent personal tour planning and scheduling in metropolises. *Expert Systems with Applications*, 38(10):12439–12452, 2011.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[3] Jonathas Magalh aes, Rubens Pessoa, Cleyton Souza, Evandro Costa, and Joseana Fechine. A recommender system for predicting user engagement in twitter. In *Proceedings of the Recommender Systems Challenge*, page 52, 2014.

[4] Fabio Aiolli. Efficient top-n recommendation for very large scale binary rated datasets. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 273–280. ACM, 2013.

[5] Ángel García-Crespo, José Luis López-Cuadrado, Ricardo Colomo-Palacios, Israel González-Carrasco, and Belén Ruiz-Mezcua. Sem-fit: A semantic based expert system to provide recommendations in the tourism domain. *Expert Systems with Applications: An International Journal*, 38(10):13310–13319, September 2011.

[6] Johan Arndt. Role of product-related conversations in the diffusion of a new product. *Journal of marketing Research*, pages 291–295, 1967.

[7] Richard Atterer, Monika Wnuk, and Albrecht Schmidt. Knowing the user's every move: user activity tracking for website usability evaluation and implicit

interaction. In *Proceedings of the 15th international conference on World Wide Web*, pages 203–212. ACM, 2006.

[8] Ricardo A. Baeza Yates and Berthier R. Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[9] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[10] Yang Bao, Hui Fang, and Jie Zhang. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In *Proceedings of Twenty-Eighth AAAI Conference on Artificial Intelligence*, volume 14, pages 2–8, 2014.

[11] Ranieri Baraglia, Claudio Frattari, Cristina Ioana Muntean, Franco Maria Nardini, and Fabrizio Silvestri. A trajectory-based recommender system for tourism. In *International Conference on Active Media Technology*, pages 196–205. Springer, 2012.

[12] Alejandro Bellogín, Iván Cantador, and Pablo Castells. A study of heterogeneity in recommendations for a social music service. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pages 1–8. ACM, 2010.

[13] Daniel Billsus and Michael J Pazzani. Learning collaborative information filters. In *Proceedings of 15th International conference on Machine Learning*, volume 98, pages 46–54, 1998.

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

[15] Jesús Bobadilla, Fernando Ortega, and Antonio Hernando. A collaborative filtering similarity measure based on singularities. *Information Processing & Management*, 48(2):204–217, 2012.

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

[17] M Dylan Bould, Emily S Hladkowicz, Ashlee-Ann E Pigford, Lee-Anne Ufholz, Tatyana Postonogova, Eunkyung Shin, and Sylvain Boet. References that anyone can edit: review of wikipedia citations in peer reviewed health science literature. *BMJ*, 348:g1585, 2014.

[18] Keith Bradley, Rachael Rafter, and Barry Smyth. Case-based user profiling for content personalisation. In *Proceedings of International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 62–72. Springer, 2000.

[19] Matthias Braunhofer, Marius Kaminskas, and Francesco Ricci. Recommending music for places of interest in a mobile travel guide. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 253–256. ACM, 2011.

[20] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52, 1998.

[21] Gülçin Büyüközkan and Buse Ergün. Intelligent system applications in electronic tourism. *Expert systems with applications*, 38(6):6586–6598, 2011.

[22] Iván Cantador, Alejandro Bellogín, and Pablo Castells. A multilayer ontology-based hybrid recommendation model. *AI Communications*, 21(2-3):203–210, 2008.

[23] David Carmel, Haggai Roitman, and Naama Zwerdling. Enhancing cluster labeling using wikipedia. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 139–146. ACM, 2009.

[24] Kannan Chandrasekaran, Susan Gauch, Praveen Lakkaraju, and Hiep Luong. Concept-based document recommendations for citeseer authors. In *Proceedings of Adaptive hypermedia and adaptive web-based systems*, pages 83–92. Springer, 2008.

[25] Shuo Chang, F Maxwell Harper, and Loren Gilbert Terveen. Crowd-based personalized natural language explanations for recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 175–182, September 2016.

[26] Ting Chen, Wei-Li Han, Hai-Dong Wang, Yi-Xun Zhou, Bin Xu, and Bin-Yu Zang. Content recommendation system based on private dynamic user profile. In *Proceedings of 2007 International Conference on Machine Learning and Cybernetics*, volume 4, pages 2112–2118. IEEE, 2007.

[27] Keunho Choi and Yongmoo Suh. A new similarity function for selecting neighbors for each target item in collaborative filtering. *Knowledge-Based Systems*, 37:146–153, 2013.

[28] William B Claster, Malcolm Cooper, and Philip Sallis. Thailand–tourism and conflict: Modeling sentiment from twitter tweets using naïve bayes and unsupervised artificial neural nets. In *Proceedings of Second International Conference on Computational Intelligence, Modelling and Simulation*, pages 89–94, 2010.

[29] Dan Cosley, Shyong K. Lam, Istvan Albert, Joseph A. Konstan, and John Riedl. Is seeing believing?: How recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 585–592. ACM, April 2003.

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

[31] Ruihai Dong, Markus Schaal, Michael P O'Mahony, and Barry Smyth. Topic extraction from online reviews for classification and recommendation. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1310–1316, 2013.

[32] Frederico Durao and Peter Dolog. Improving tag-based recommendation with the collaborative value of wiki pages for knowledge sharing. *Journal of Ambient Intelligence and Humanized Computing*, 5(1):21–38, February 2014.

[33] Douglas Eck, Paul Lamere, Thierry Bertin-Mahieux, and Stephen Green. Automatic generation of social tags for music recommendation. In *Proceedings of Advances in neural information processing systems 2008*, pages 385–392, 2008.

[34] Charles Elkan. Using the triangle inequality to accelerate k-means. In *Proceedings of International Conference on Machine Learning*, pages 147–153, 2003.

[35] Sandra Garcia Esparza, Michael P O'Mahony, and Barry Smyth. Mining the real-time web: a novel approach to product recommendation. *Knowledge-Based Systems*, 29:3–11, 2012.

[36] Giuseppe Fenza, Enrico Fischetti, Domenico Furno, and Vincenzo Loia. A hybrid context aware system for tourist guidance based on collaborative filtering. In *Proceedings of IEEE International Conference on Fuzzy Systems*, pages 131–138. IEEE, 2011.

[37] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. A generic semantic-based framework for cross-domain recommendation. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pages 25–32. ACM, 2011.

[38] Adam Field, Pieter Hartel, and Wim Mooij. Personal dj, an architecture for personalised content delivery. In *Proceedings of the 10th International Conference on World Wide Web*, pages 1–7, 2001.

[39] Susan Gauch, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli. User profiles for personalized information access. *The adaptive web*, pages 54–89, 2007.

[40] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 330–339. ACM, 2007.

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

[42] Jon Atle Gulla, Arne Dag Fidjestøl, Xiaomeng Su, and Humberto Nicolás Castejón Martínez. Implicit user profiling in news recommender systems. In *Proceedings of 10th International Conference on Web Information Systems and Technologies*, pages 185–192, 2014.

[43] Robert Győrödi, Cornelia Győrödi, and Mihai Deridan. An extended recommendation system using data mining implemented for smart phones. *International Journal of Computers and Technology*, 11(3):2360–2372, 2013.

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

[45] 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.

[46] Gregor Heinrich. Parameter estimation for text analysis. *Technical Report, University of Leipzig*, 2008.

[47] Thorsten Hennig-Thurau, Kevin P Gwinner, Gianfranco Walsh, and Dwayne D Gremler. Electronic word-of-mouth via consumer-opinion platforms: what motivates consumers to articulate themselves on the internet? *Journal of interactive marketing*, 18(1):38–52, 2004.

[48] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[49] Daniel Herzog and Wolfgang Wörndl. A travel recommender system for combining multiple travel regions to a composite trip. In *Proceedings of Workshop on New Trends in Content-Based Recommender Systems*, pages 42–48. Citeseer, 2014.

[50] Yoshinori Hijikata. Implicit user profiling for on demand relevance feedback. In *Proceedings of the 9th international conference on Intelligent user interfaces*, pages 198–205. ACM, 2004.

[51] Rune Hjelsvold, Subu Vdaygiri, and Yves Léauté. Web-based personalization and management of interactive video. In *Proceedings of the 10th international conference on World Wide Web*, pages 129–139. ACM, 2001.

[52] Chen Hongbo, Chen Zhiming, Mohammad Shamsul Arefin, and Yasuhiko Morimoto. Place recommendation from check-in spots on location-based online social networks. In *Proceedings of the 3rd International Conference on Networking and Computing*, pages 143–148, 2012.

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

[54] Marius Kaminskas and Francesco Ricci. Location-adapted music recommendation using tags. *User Modeling, Adaption and Personalization*, pages 183–194, 2011.

[55] Jaeyong Kang and Hyunju Lee. Modeling user interest in social media using news media and wikipedia. *Information Systems*, 65:52–64, 2017.

[56] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, pages 79–86. ACM, September 2010.

[57] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 79–86, 2010.

[58] Byeong Man Kim, Qing Li, Chang Seok Park, Si Gwan Kim, and Ju Yeon Kim. A new approach for combining content-based and collaborative filters. *Journal of Intelligent Information Systems*, 27(1):79–91, 2006.

[59] Hyoung R Kim and Philip K Chan. Learning implicit user interest hierarchy for context in personalization. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 101–108. ACM, 2003.

[60] Masaru Kitsuregawa and Toyoaki Nishida. Special issue on information explosion. *New Generation Computing*, 28(3):207–215, 2010.

[61] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, August 2008.

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

[63] Kaushal Kurapati, Srinivas Gutta, David Schaffer, Jacquelyn Martino, and John Zimmerman. A multi-agent tv recommender. In *Proceedings of the UM 2001 workshop Personalization in Future TV*, 2001.

[64] Yohei Kurata, Yasutaka Shinagawa, and Tatsunori Hara. Ct-planner5: a computer-aided tour planning service which profits both tourists and destinations. In *Proceedings of the Workshop on Tourism Recommender Systems in 9th ACM Conference on Recommender Systems (RecSys 2015)*, volume 15, pages 35–42. ACM, 2015.

[65] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference on machine learning*, volume 10, pages 331–339, July 1995.

[66] Joonseok Lee, Mingxuan Sun, and Guy Lebanon. A comparative study of collaborative filtering algorithms. *arXiv preprint arXiv:1205.3193*, 2012.

[67] Kwan Hui Lim. Recommending tours and places-of-interest based on user interests from geo-tagged photos. In *Proceedings of the 2015 ACM SIGMOD on PhD Symposium*, pages 33–38. ACM, 2015.

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

[69] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1043–1051. ACM, 2013.

[70] Qi Liu, Haiping Ma, Enhong Chen, and Hui Xiong. A survey of context-aware mobile recommendations. *International Journal of Information Technology and Decision Making*, 12(1):139, January 2013.

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

[72] Chunliang Lu, Wai Lam, and Yingxiao Zhang. Twitter user modeling and tweets recommendation based on wikipedia concept graph. In *Proceedings of Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[73] Claudio Lucchese, Raffaele Perego, Fabrizio Silvestri, Hossein Vahabi, and Rossano Venturini. How random walks can help tourism. In *European Conference on Information Retrieval*, pages 195–206. Springer, 2012.

[74] Heng Luo, Changyong Niu, Ruimin Shen, and Carsten Ullrich. A collaborative filtering framework based on both local user similarity and global user similarity. *Machine Learning*, 72(3):231–245, 2008.

[75] Marcin Maleszka, Bernadetta Mianowska, and Ngoc Thanh Nguyen. A method for collaborative recommendation using knowledge integration tools and hierarchical structure of user profiles. *Knowledge-Based Systems*, 47:1–13, 2013.

[76] W Glynn Mangold and David J Faulds. Social media: The new hybrid element of the promotion mix. *Business horizons*, 52(4):357–365, 2009.

[77] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, October 2013.

[78] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, August 2015.

[79] Colin Meek and William P Birmingham. Automatic thematic extractor. *Journal of Intelligent Information Systems*, 21(1):9–33, 2003.

[80] Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence(AAAI-2002)*, pages 187–192, 2002.

[81] J. C. L. Menchavez and K. J. P. Espinosa. Fun in the philippines: Automatic identification and sentiment analysis of tourism-related tweets. In *Proceedings of 2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 660–667, Nov 2015.

[82] Dunja Mladenic and Browsing Assistant Personal Webwatcher. Machine learning used by personal webwatcher. 1999.

[83] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 529–533, 2011.

[84] Kenta Oku, Koki Ueno, and Fumio Hattori. Mapping geotagged tweets to tourist spots for recommender systems. In *Proceedings of the IIAI 3rd International Conference on Advanced Applied Informatics*, pages 789 – 794, 2014.

[85] Dmitry Y Pavlov and David M Pennock. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains. In *Proceedings of Advances in neural information processing systems*, pages 1465–1472, 2003.

[86] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

[87] Saverio Perugini, Marcos André Gonçalves, and Edward A Fox. Recommender systems research: A connection-centric survey. *Journal of Intelligent Information Systems*, 23(2):107–143, 2004.

[88] Oksana B Petrina, Dmitry G Korzun, Aleksey G Varfolomeyev, and Aleksandrs Ivanovs. Smart spaces based construction and personalization of recommendation services for historical e-tourism. *The International Journal on Advances in Intelligent Systems*, 9(1&2):85–95, 2016.

[89] Owen Phelan, Kevin McCarthy, Mike Bennett, and Barry Smyth. Terms of a feather: Content-based news recommendation and discovery using twitter. *Advances in Information Retrieval*, pages 448–459, 2011.

[90] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, pages 385–388. ACM, 2009.

[91] Sanjay Purushotham, Yan Liu, and C-c J Kuo. Collaborative topic regression with social matrix factorization for recommendation systems. In *Proceedings of the 29th International Conference on Machine Learning*, pages 759–766, June 2012.

[92] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B Kantor. *Recommender systems handbook*. Springer, 2015.

[93] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[94] Tae Hyup Roh, Kyong Joo Oh, and Ingoo Han. The collaborative filtering recommendation based on som cluster-indexing cbr. *Expert systems with applications*, 25(3):413–423, 2003.

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

[96] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, July 2008.

[97] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

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

[99] Bo Shao, Dingding Wang, Tao Li, and Mitsunori Ogihara. Music recommendation based on acoustic features and user access patterns. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(8):1602–1611, 2009.

[100] Kazutaka Shimada, Shunsuke Inoue, and Tsutomu Endo. On-site likelihood identification of tweets for tourism information analysis. In *Proceedings of the IIAI International Conference on Advanced Applied Informatics*, pages 117–122, 2012.

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

[102] Chang Tan, Qi Liu, Enhong Chen, Hui Xiong, and Xiang Wu. Object-oriented travel package recommendation. *ACM Transactions on Intelligent Systems and Technology*, 5(3):43:1–43:26, September 2014.

[103] Poonam B Thorat, RM Goudar, and Sunita Barve. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, 110(4), 2015.

[104] Lyle H Ungar and Dean P Foster. Clustering methods for collaborative filtering. In *AAAI workshop on recommendation systems*, volume 1, pages 114–129, 1998.

[105] David Vallet, Iván Cantador, and Joemon Jose. Personalizing web search with folksonomy-based user and document profiles. *Advances in information retrieval*, pages 420–431, 2010.

[106] Robin Van Meteren and Maarten Van Someren. Using content-based filtering for recommendation. In *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, pages 47–56, 2000.

[107] Bernadette Varga and Adrian Groza. Integrating dbpedia and sentiwordnet for a tourism recommender system. In *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*, pages 133–136. IEEE, 2011.

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

[109] Fei Wang, Zhe Zhang, Hailong Sun, Richong Zhang, and Xudong Liu. A cooperation based metric for mobile applications recommendation. In *Proceedings of 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, volume 3, pages 13–16. IEEE, 2013.

[110] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1235–1244. ACM, August 2015.

[111] Zheng Xiang and Ulrike Gretzel. Role of social media in online travel information search. *Tourism management*, 31(2):179–188, 2010.

[112] Markus Zanker, Markus Jessenitschnig, and Wolfgang Schmid. Preference reasoning with soft constraints in constraint-based recommender systems. *Constraints*, 15(4):574–595, 2010.

[113] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[114] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92. ACM, July 2014.

[115] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pages 928–935, August 2003.

Publication

International Journal

1. Guanshen Fang, Sayaka Kamei, and Satoshi Fujita. "A Japanese Tourism Recommender System with Automatic Generation of Seasonal Feature Vectors", *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, no. 6, pp. 347–354, 2017.

2. Guanshen Fang, Sayaka Kamei, and Satoshi Fujita. "Rating Prediction with Topic Gradient Descent Method for Matrix Factorization in Recommendation", *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, no. 12, pp. 469–476, 2017.

International Conference

1. Guanshen Fang, Sayaka Kamei, and Satoshi Fujita. "Automatic Generation of Temporal Feature Vectors with Application to Tourism Recommender Systems", In *International Workshop on Advances in Networking and Computing (WANC), in conjunction with CANDAR 2016*, pp. 676–680, November, 2016