LOCATION RECOMMENDATION USING CONTENT-AWARE COLLABORATIVE FILTERING

PROJECT REPORT

Submitted by

AKASH NATH	**NSS15CS003**
AMALDEV	**NSS15CS006**
LALKRISHNA S	**NSS15CS038**
SOURAV A S	**NSS15CS056**

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree of
Bachelor of Technology
In
Computer Science and Engineering

Department of Computer Science and Engineering

NSS College of Engineering,

Palakkad

May 2019

DECLARATION

We undersigned hereby declare that the project report "Location Recommendation Using Content-Aware Collaborative Filtering", submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Ms Chitra S Nair. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Palakkad
Date: 2019-05-25

AKASH NATH

AMALDEV

LALKRISHNA S

SOURAV A S

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NSS COLLEGE OF ENGINEERING PALAKKAD

CERTIFICATE

This is to certify that the project report titled "**LOCATION RECOMMENDATION USING CONTENT-AWARE COLLABORATIVE FILTERING"** submitted by **AKASH NATH, AMALDEV, LALKRISHNA S** and **SOURAV A S** to the APJ Abdul Kalam Technological University, in partial fulfilment of the requirement for the award of B.Tech Degree in Computer Science and Engineering is a bonafide record of the work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Mrs. Chitra S Nair

Guide

Dr. Maya Mohan

Project in Charge

Dr. Sindhu S

Head of Department

External Examiner

ACKNOWLEDGEMENT

We wish to express our wholehearted indebtedness to God Almighty for his gracious constant care and blessings showered over us during the course of this project.

We are grateful towards **Dr. Sudha T,** Principal, N.S.S College of Engineering, Palakkad, for shaping a vision of academic success for all students and for being the central source of leadership influence.

We are thankful towards **Dr. Sindhu S**, Head of the Department of Computer Science and Engineering, N.S.S College of Engineering, Palakkad, for providing and availing all the required facilities for undertaking the project in a systematic way.

We are deeply indebted to Ms. Chitra S Nair, Assistant Professor, Department of Computer Science and Engineering, N.S.S College of Engineering, Palakkad, for her assistance, guidance and contribution towards the project as a guide.

We would like to express our gratitude to committee members Professor Nandakumar P**,** Dr. Maya Mohan**,** Mrs. Sruthy Manmadhan for providing good suggestions to improve the project. Gratitude is extended to all teaching and non-teaching staffs of Department of Computer Science and Engineering, N.S.S College of Engineering, Palakkad, for the sincere directions imparted and the cooperation in connection with the project.

We are also thankful to our parents for the support given in connection with the project. Gratitude may be extended to all well-wishers and friends who supported us.

Akash Nath

Amaldev

Lalkrishna S

Sourav A S

ABSTRACT

Location recommendation plays an essential role in POI (Point Of Interest) recommendation for tourists and for leisure activities. It also can be utilized for designing location strategies for new business based on human traffic. This project focuses on addressing the cold start problem, a key disadvantage of collaborative filtering recommender systems. This project uses human mobility data, an implicit feedback dataset, which contains user location check-ins. As a user's negative preferences are not explicitly observable in most human mobility data, implicit feedback model is used. A combination of content-based recommendation and collaborative filtering is suggested. Location visits (Check-ins) are normally shared on a social network and from this information, a user's interest can be extracted. This is used to refine user similarity based on mobility. Locations are recommended based on mobile data and user's interest. The project utilizes large-scale LBSN (Location Based Social Network) dataset where users have profiles and text content. Thus, the project attempts to prove that content aware collaborative filtering is not only effective at improving recommendation but also helpful in coping with the cold-start problem.

CONTENTS

Contents **Page No.**

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CF: Collaborative Filtering

ICCF: Implicit Content aware Collaborative Filtering

LBSN: Location Based Social Network

POI: Point of Interest

CHAPTER 1
INTRODUCTION

1.1 General Background

As cities expand, the number of location or Point of Interest (POI) also grow which include hotels, museums, parks etc. People love to explore new neighbourhood and locations as novelty seeking is one of the most basic human activities. Location recommendation can be exploited to provide users with a list of locations to visit and also the corresponding set of activities to do. With the recent developments in Recommender systems, a more personalized recommendation is possible. Thus, location recommendation can help people discover places tailored to their interest and speed up their familiarization with the surroundings.

The rise in popularity of Location-Based Social Networks (LBSN) has led to the generation of a huge volume of human mobility data. This is generated as users of these websites like Foursquare, Yelp, Whrrl etc. are likely to share location visits and reviews. This has led to location recommendation becoming one of the hot research topics. Transparency and privacy of individual users are also taken into consideration as only publicly available data is used and proper agreements are signed beforehand. Human mobility data is used for tourist and leisure recommendations and can also be utilized for generating location strategies for new business. A peculiarity of human mobility data is lack of explicit negative feedback (rating) in the dataset. This implies either sentiment analysis of location reviews has to be performed or some other mechanism without the use of customer reviews has to be used to obtain negative feedback. Also, users have this account linked to other social networks such as Twitter, Facebook etc. where they share reviews and preferences. This social media data can be leveraged to obtain user profile which captures user behavior and also implies their interests [1].

The decisions of users to visit places depend on multiple factors, and different users may be affected differently by these factors. In general, the factors affecting user decision can be classified into 3 categories- social influences, temporal influences, geographical influences. Thus, location recommendation focuses on extracting this information from human mobility data. Each of these influences requires a different model of Recommender System. So, creating a model that exploits all the categories is a highly complex task and is a wide area of research. Recently contextual information is also considered as a factor of recommendation [3], [4] and it requires sensors to acquire them and is very domain specific. In the case of location recommendation, it refers to weather, traffic, time of day etc. It is comparatively a new area of research. Generally, recent location recommendation systems are more generalized that work for any type of locations and still provide a personalized recommendation to users.

1.2 Areas of Research

1.2.1 Recommender systems

Recommender systems are one of the widely used applications on machine learning because most big corporations want users to have a personalized experience while using their product which results in longer engagement time and ultimately more profit. Practical application of this can be seen in YouTube, Netflix, Spotify and Amazon recommendation engines which are popular topics of discussion in the recommender system community. These systems show how theoretical aspects of machine learning can be applied to real-time data and construct actual systems which are of great practical use. The popularity can also be attributed to the emergence of big data. Most of the big companies have released their user datasets which have further fueled the research in recommendation systems. The process of

recommendation is also called ‘discovery assistant’ as through recommendations new items are discovered by users [2].

1.2.2 Location Recommendation

Location recommendation focuses on recommending a list of places that the user is likely to visit based on user preferences. Its practical application lies in location-based services which incorporate location information. The popularity lies in mobile recommendation where users share their location visits (check-ins) on a social network like Facebook, Twitter etc and also in the advent of Location-based social networks (LBSN). A huge volume of human mobility data can be extracted from these networks which are used for location recommendation. Location-based services include location strategies for new business, location and activity recommendation for tourist or for adventure enthusiasts, restaurant and motel recommendation based on customer reviews etc. While prior research focused on mostly recommending a specific type of location like restaurants, recent studies focus on more generalized POI recommendation that exploits geographical, temporal, social and even contextual information [5], [6]. Combination of these factors is also considered for better performance which is generally evaluated based on the rating that the user is likely to give or the number of times the user is likely to visit the place in case of human mobility data. In addition to these factors, textual information is also available in the form of customer review. Topic modelling and sentiment analysis can also be performed to incorporate this information into the recommendation model.

Human mobility data does not provide explicit negative feedback as it only contains the location visit frequency. Thus, negative feedback has to be extracted from it. While past research focused on negative sampling, this project utilizes an implicit feedback model that

assigns lower confidence to negative preferences which are more practical as an unvisited location may indicate an unknown location or unfavored locations [7], [8].

In general, location recommendation depends on the user information and location information extracted and matching that is used. User information includes the user profile and their preferences. Location information may include associated textual information in the form of reviews and description. User information is more dynamic and is more important in providing a recommendation to a new user. Thus, social media data is leveraged to obtain more refined user profile. It is seen that most users tend to visit a few places compared to all locations.

Thus, location recommendation suffers from sparsity problem.

1.2.3 Content-aware collaborative filtering

It is the integration of collaborative filtering and content-based filtering. While collaborative filtering provides improved performance, it suffers from Cold-Start problem; which is providing a recommendation to new users and recommending new locations to existing users. Thus, content-based is integrated to overcome these shortcomings and a hybrid model is used for better performances [10].

1.3 Objectives

To design a location recommender that takes in a set of users and a set of locations and gives a list of places based on preference score for each user. The given model is a generalized location recommender which works for any locations and provides users with personalized recommendations. Through the given model, the effects of user content and location content are examined. User content involves the user profile and preferences leveraged from social network data. Location information can also be refined in a similar manner.

Since human mobility data is used, which can be extracted from LBSN, which will also contain links to corresponding social network accounts, implicit feedback is used as past research. It has shown that explicit feedback which works on negative sampling tend to perform poorly compared to implicit feedback. Most real-world systems tend to use implicit feedback because of the lack of explicit feedback from users. Thus, implicit feedback is used to obtain a user-location rating and provide recommendations based on it.

Since both user content and location content are used, the recommendation to new users and scaling the model to involve new location is also possible as initially the user profile is used to match users to preferred locations. Thus, the effects of user content and location content are explored in both warm start and cold start cases.

1.4 Problem Identification

1.4.1 Lack of Personalization

Most location recommenders are based on user-location rating and are generally not personalized because they use collaborative filtering method where recommendations are based on similar users and past ratings. Problem-specific factors such geographical influences or social influences are not taken into consideration and also it is difficult to define similarity between locations of different types. Thus, rating based location recommenders are not used for general POI recommendation and instead, human mobility data is used for location recommendation. Lack of user and location information along with the sparsity problem associated with human mobility data also lead to this problem as users tend to visit the same locations more often and the unvisited locations maybe unfavorable or unknown to them.

1.4.2 Lack of Feedback

Most recommender systems are rating based where the user provides each item they consume with a preference score. Higher rating implies favorable item and thus recommendation of item similar characteristics is possible. However, in practical real-world systems, users rarely provide items with a score. In fact, asking users to rate every item they consume is proven to be counterproductive. In location recommendation, although users may leave reviews of the visited place, obtaining a preference score based on the textual information left behind is still a complex task involving topic modelling and sentiment analysis. Also, review based recommendations are not scalable as new locations will not have any reviews, to begin with. Thus, implicit feedback has to be used to infer preference score based on user location visit frequency.

1.4.3 Cold Start Problem

A commonly seen problem in recommender system in which recommendation related to new users and new items cannot be made since sufficient information about them is not available. Collaborative filtering fails completely in case of cold-start. To counter this a hybrid model is used which uses both user content and location content. The presence of user content which consists of user profile and preferences can be used to provide initial recommendations to new users while location content can be used to handle new items. This also helps in providing more personalized recommendation and making the given model scalable.

CHAPTER 2
LITERATURE SURVEY

In Lian., et al., [1], proposes an implicit feedback content-aware location recommender based on human mobility data collected from LBSN. It uses both user and location information to provide recommendation and addresses the cold-start problem normally seen in collaborative filtering by integrating both content-based recommendation and collaborative filtering. Users normally share their location visits in social media which are linked to LBSN. This information can be leveraged to capture user interests and behaviour. The given model uses implicit feedback where unvisited locations are considered as negative feedback with lower confidence and do not use negative sampling methods. The presented work mostly focuses on studying the effects of user profiles and textual content in improving recommendations in both warm-start and cold-start cases.

This paper acts as our base paper and serves as an introduction to location recommendation through collaborative filtering. It also specifies the advantages of using implicit feedback over explicit feedback and other negative sampling methods. The given model is scalable and uses data extracted from LBSN. This also served as an introduction to the cold-start problem, which is providing recommendations to new users and recommending new items. User profile and textual content can be used in improving recommendation during cold start case as they capture user interests. The given work also served as an introduction to matrix factorisation which is used to decompose the user-item matrix and obtain latent factors in user matrix and item matrix. Thus, the given model is a highly personalised, scalable and practical Point-Of-Interest Recommender, based on the mobility data accumulated from LBSN. It provides further opportunity for future research focusing on leveraging both location information and user-information from social network and exploring the improved

performance in a recommendation. Other geographical, temporal, social and contextual information can be integrated to further improve the performance.

Following sections contain explanations regarding the above-mentioned topics and possible alternatives. Here sections are arranged based on the order of references specified.

2.1 Collaborative Location Recommendation

In V. Zheng, et al., [2], proposes a user-centred collaborative location and activity filtering to find like-minded users and provide location and activity recommendations based on them. It mainly addresses the issue that data available on individual users is less but to make a useful and accurate recommendation, extensive annotated location and activity information from user trace data is needed. It uses a tensor representation to establish the user-locationactivity relation and uses a matrix decomposition to find the necessary solution. It tries to exploit the activity-location relationship as activities are usually location dependent and also addresses the data sparsity problem as ratings provided by users are sparse and insufficient. This model is capable of handling locations of different types and providing recommendations of interesting activities to be performed at the corresponding locations.

This served as an introduction to location recommendation through collaborative filtering which focuses on finding similar users. The given model did address the sparsity issue by proposing an algorithm which completes the incomplete tensor based on additional information matrices which implies that additional information is always needed to provide a more personalized recommendation. Using this, the model attempts to uncover user location activity relations and provide recommendations based on it. The given model does not address the cold start problem commonly seen in CF based recommenders. Also, it assumed that ample user and location information is available and did not focus on extracting this information from the real-world system or showed how the performance of the model is affected upon scaling.

Also, the model focused on using GPS information extracted from mobile applications to perform location-activity recommendation and did not exploit the data accumulated from LBSN which contains the same information in a more structured format.

2.2 Factors Affecting Location Recommendation

2.2.1 Exploiting geographical factors

In Lian., et al., [3], proposes CF based location recommender with implicit feedback that incorporates clustering phenomenon observed in human mobility behavior, i.e. individual visiting location tend to cluster together. By modelling the spatial clustering phenomenon, it addresses the problem of user-POI matrix sparsity and helps in improving recommendation performance. This model utilizes the mobility records of LBSN as implicit feedback for POI recommendation and exploits weighted matrix factorisation for this task. It focuses on augmenting users' and POIs latent factors with activity area vectors of users and influence area vectors of POIs, respectively to leverage geographical information in POI recommendation. It is based on the intuition that if a user often visits a certain POI but has never visited surrounding POIs, it is highly unlikely for the user to visit them in the future. The reason for using check-ins for implicit feedback is as follows:

- They provide positive examples.
- Visit frequency is a clear indication of confidence of user in visiting a location.
- Unvisited locations are either unfavorable or undiscovered.

This served as an introduction to implicit feedback-based location recommendation that leverages geographical information in improving POI recommendation and weighted matrix factorisation can be used for this task. It also served as an explanation of the advantages in using matrix factorisation in CF recommender with implicit feedback. The given work did

suggest the effect of contextual and textual information about locations and users in improving recommendations as future areas of research. It also didn't address the cold start problem normally seen CF based recommenders.

2.2.2 Context-aware recommender system

In Adomavicius, G., et al, [4], proposes a Recommendation model that incorporates context into the system. Context-Aware Recommender Systems are able to produce more personalized recommendations by adapting them to the situational context of the user. Depending on the type of information that the model knows, we are able to classify the contextual factors of the recommender system knowledge into three categories: Fully Observable, Partially Observable and Unobservable. There are mainly two types of Contextual factors that are affected over time, they are: Static and Dynamic. In Static Context, the structure and the contextual factors remain stable over a period of time. For example, Age of a person, Gender etc. In Dynamic Context, the factors affecting the context of a situation can change with respect to time. For example, while recommending a location to the user, he may prefer to go to different restaurants at different times of the day. Context can be obtained either explicitly or implicitly. The three main paradigms used for incorporating context into this model are Contextual Pre-filtering, Post-Filtering and Modelling.

In many application domains, a context-independent representation may lose predictive power as potentially useful information from multiple contexts are aggregated. This type of system can be useful in our project. The disadvantages of using this is that it is difficult to define what are the relevant contexts and utility of each context for a personalized recommendation. This is a relatively new research area and so, there are many unresolved issues and research challenges. Also, a collection of contexts in a practical system or even acquiring contexts for research purposes seems to be an issue.

In Karatzoglou, A., et al., [5], proposes a CF-based Tensor Factorization, a generalization of Matrix Factorization that allows for the integration of contextual information by modelling the data as a User-Item-Context N-dimensional. The factorization of this tensor results in a compact model of the data which can then be used to provide context-aware recommendations which addresses also the contextual recommendation problem. This enables the model to take advantage of the sparsity of the data while still exploiting the interaction between all users and items and context and also embed multiple contextual dimensions into a coherent model.

The given work further proved that performance of the recommender system generally increases with the amount of contextual information present. The given model does suffer from all the disadvantages of using a context-aware system as mentioned above. The paper suggested in applying the given model to hybrid recommender model but in its present form, it does not address the cold-start problem normally seen in CF based recommender. Adapting context aware recommenders to handle location recommendation is still a widely researched topic.

2.2.3 Time-Aware Spatio-Textual Recommender System

In Kefalas, P, et al., [6] proposes two novel unified models that provide POI and Review recommendations while considering the spatial, temporal and textual factors which is extracted from Location-Based Social Networks or LBSNs (used to post reviews and ratings on various topics and locations and can also alert or notify friends about these posts). In the First model, Item-based collaborative filtering is extended by combining the spatial influence of user reviews and the textual information amongst these reviews. In the second model, user-based collaborative filtering is improved by incorporating social influence of user reviews and

spatial influence of user check-in history. And finally, the temporal aspect is considered for both the models by measuring the impact of time on different intervals of time.

Traditionally, different kinds of models have been proposed for Point-Of-Interest recommendation based on both Explicit (Comments and ratings etc.) and Implicit (User Influence, Views, Score etc.) information. However, these models or methods do not sufficiently capture the user preferences as they tend to change according to the time and place the users are currently in. Time is an essential factor because user check-in behaviour can be time-dependent or periodic, like the venues they attend or words they use. For example, checkin at work in the morning or at home in the evening. The usage of textual influence in the recommender system can provide a more robust set of review recommendations that are far more likely to be preferred by the user. Thus, users who make use of the same vocabulary can be classified into similar groups. And by incorporating the situational context of the users, we are able to generate a more personalized recommendation for the users based on these factors. Hence, the users who check-in at nearby locations during the same time period are likely to have the same interests.

2.3 Implicit Feedback

In Hu, Y., et al., [7], collaborative filtering with implicit feedback is detailed. Implicit feedback refers to the data gathered from user behaviour. Implicit feedback is different from explicit which feedback which usually consists of a rating like system where the user will give a rating to products. Implicit data has some unique characteristics such as:

- Lack of negative feedback
- Inherently noisy
- Evaluation requires appropriate measures

- Numerical value indicates confidence whereas, in the case of explicit feedback, it indicates a preference

The paper proposes treating data as an indication of positive and negative preference associated with vastly varying confidence levels in order to evaluate implicit feedback. Implicit user observations should be transformed into two paired magnitudes: preference (indicates whether the user has consumed an item or not) and confidence levels associated with the item. An item which the user has interacted a lot will have higher confidence than an item which the user has interacted less with. The paper proposed a latent factor model. After finding preference and confidence, matrix factorization is done to find the user factors as well as item factors which indicates the characteristics of the user or the item. Alternating least square (ALS) is used to minimize the cost function while finding the factors. The output is two matrix, user matrix and an item matrix. When they're multiplied, the result is a user by item matrix which will have a score for a user for each item. Recommendations are made by taking the highest k scores.

The paper serves as an introduction to collaborative filtering with implicit feedback. It also specifies how to get good recommendations from implicit data containing noise. The given model is scalable and was implemented on a digital television service with 300,000 set-top boxes. The given paper also served as an introduction to matrix factorisation which is used to decompose the user-item matrix and obtain latent factors in user matrix and item matrix. This does not solve the cold start problem as additional content is needed to solve it.

In Kawai, K., et al., [8], collaborative filtering with implicit feedback is addressed. Implicit feedback is more important than explicit feedbacks as they are easier to obtain and more abundant as well. The downside is that implicit feedback is noisy. It is difficult to infer if the user liked a product or not as users could dislike a product after consuming it. The

papers proposed to address this noise problem. There is a hidden uncertainty for all observed feedback.

The uncertainty is in whether the user liked it or not. The effects of observed feedback of greater uncertainty are discounted. The work describes three discounting methods:

- User-oriented discounting
- Item-oriented discounting
- Time-oriented discounting

- User-oriented discounting: it assumes that the confidences of observed feedbacks are actually depended on the number of feedbacks a user provided for items, i.e., feedbacks by a user who gives feedback on a few items is more confident than that of a user which gives feedbacks to many items.
- Item-oriented discounting: it assumes that confidences of feedbacks depend on the number of feedbacks an item gets from users. Popular items will get more feedback. This method will help to recommend non-popular items as well.
- Time-oriented discounting: it assumes that confidence of feedbacks reduces with time. Users taste changes with time. So, feedbacks taken a long time ago may not represent a user's current preference correctly.

The model introduced serves as an introduction to implicit collaborative filtering and also address the noise issue with implicit feedbacks. The given methods can be easily combined with existing methods such as matrix factorization model and improve its recommendation accuracy. The given method is scalable and was implemented on three datasets: Rakuten Recipe (recipes) dataset, Lastfm Song (music) dataset and YOOCHOOSE(e-commerce) dataset. It was found that time-oriented discounting methods significantly improve the baseline model, which indicated that the feedbacks will be noisy with time. User-oriented discounting as well as item-oriented discounting increased the

performance. Future research area for this method includes selecting the best discounting methods depending on the user.

2.4 Alternate Evaluation Strategies

In S. Rendle, et al., [9], "BPR: Bayesian personalized ranking from implicit feedback" present a generic method for learning models for personalized ranking in recommender systems. So far recommender systems have focused only on improving recommendation accuracy. This causes overspecialization which can lead to frustration for the user. This work explores several ranking approaches that increase the aggregate diversity in recommender system as a general method for any recommender models. In the commonly used approach for ranking the items in a recommender system, the predicted rating is ranked from highest to lowest. Then the items with the highest predicted ratings are recommended to the user. This approach increases the accuracy in recommender system but not diversity for personalization. There are many methods for item recommendation from implicit feedback like matrix factorization or k-nearest-neighbour. Even though all these methods are designed for the item prediction task with personalized ranking, none of them is directly optimized for ranking purpose. Instead of applying rating prediction techniques, BPR ranks the items for a user without calculating a prediction rating. Instead of treating items as single entities they have taken pairwise comparison for the task of recommendation, where each value in the matrix indicates user's preference of one item over another.

In this work, they present a generic optimization criterion BPR-Opt by using Bayesian analysis of the problem for personalized ranking that is the maximum posterior estimator derived from the analysis. They also provide a generic learning algorithm with respect to BPROpt for optimizing models, which is based on stochastic gradient descent with bootstrap sampling. They show how to apply this method to two commonly used recommender models:

matrix factorization and adaptive kNN. The experimental results in this work indicate that for the task of personalized ranking the optimization technique discussed through the model outperforms the standard learning techniques for MF and kNN. The results show the importance of optimizing models for the right criterion for a better recommendation.

In Li, X., et al., [10], "Rank-GeoFM" present a ranking based geographical factorization method for POI recommendation. The online systems such as location-based social networks (LBSN) enable people to check in and share their experiences with friends when they visit a point of interest (POI), e.g., restaurant, and shopping mall. The huge amount of data contains valuable information about POIs, and human preference, which can be exploited for POI recommendation. Recommendation of POI suffers some problems, the check-in data in LBSN is very sparse, and thus recommendation methods suffer from the data scarcity problem, also the check-in is a type of implicit feedback, so the check-ins offer only positive examples that a user likes, and the POIs without check-ins, which are unknown, are either unattractive or undiscovered but potentially attractive. In other words, we need to infer the user's preference and non-preference based on the check-in data. In POI recommendation, different types of context information are available, e.g., geographical coordinates of POIs, time stamps of check-ins, categories of POIs, etc. It is important to exploit context information to improve the recommendation accuracy. For example, geographical coordinates are used as an important type of context information are exploited for POI recommendation and not available in conventional recommendation tasks,

Previous works for POI recommendation has different approaches to exploiting the different types of context information. However, these approaches are usually developed for a particular type of context and it is difficult to generalize them to handle another type of context information like geographic coordinate time etc. This model addresses the above-mentioned problems for POI recommendation and improves the recommendation accuracy by

optimizing the problem for ranking also which have previously shown to increase the recommendation performance. The results show that by using the methods mentioned in this work, more contextual information can be added easily to the model without compromising the recommendation efficiency also solving the scarcity issue of the recommender, at the same time producing better results.

The main characteristics of a recommendation system are personalization and generalization. In most CF based location recommenders, the data sparsity issue, cold start problem and the lack of explicit feedback are issues that need to be addressed. By using a hybrid model that uses implicit feedback dataset extracted from LBSN, most of these issues are addressed. To handle cold start case and improve performance in warm start case, additional information is needed such as social media data, contextual information etc. Geographic, temporal and other social factors may be further integrated to the given location recommender to obtain better performance and more personalized recommendations. In most of the given models, a variant of matrix factorization was used to model the given problem and satisfactory results were obtained. Therefore, the remaining chapters focus on studying the effects on additional location and user information leveraged from social media or other sources in improving performance, when integrated with the given model.

CHAPTER 3
THEORY

The following section covers the theoretical aspects of implicit feedback content aware collaborative filtering. This section also provides an overview of the model used for analysis named ICCF (Implicit Content aware Collaborative Filtering) which takes as input human mobility data obtained from LBSN's, additional user information and location information and does implicit matrix factorization to provide top k recommendation of locations to each user based on the calculated preference score.

3.1 Collaborative Filtering for Implicit Feedback Datasets

Given human mobility data of U users visiting total of V locations, location recommendation first converts it into a user-location frequency sparse matrix $\mathbf{C} \in \mathbf{N}^{U \times V}$, with each entry $\mathbf{c_{u,i}}$ indicating the count of visits of a user **u** to a location **i**. $\mathbf{R} \in \{0, 1\}^{U \times V}$ is a preference matrix, for which each entry $\mathbf{r_{u,i}}$ is set to 1 if the user **u** has visited the location **i**; otherwise it is set to zero.

One technique that is commonly used for the recommendation problem is to project the matrix C into a low rank approximation, and then compute distances in that space. The idea is to take the original user-item visit frequency matrix, and then reduce that down to two much smaller matrices that approximate the original when multiplied together. By reducing the dimensionality of the data like this, in effect compression the input matrix down to two much smaller matrices occurs. Better understanding of the data is obtained from generalization caused by the compression of the data.

Most of the MF models used in recommender systems operate on explicit data, where the user has rated both things they like and dislike using a 5-star rating scale or point based rating indicating their preference. They work by assuming the missing data as unknown, and then minimizing the reconstruction error associated with matrix factorisation. For implicit feedback datasets, the missing values cannot be treated as unknown. To handle this case where negative data is not known with much confidence, implicit matrix factorisation learns a factorized matrix representation using different confidence levels on binary preferences: unseen items are given negative preference with a low confidence, where consumed items are treated positive preference with a much higher confidence.

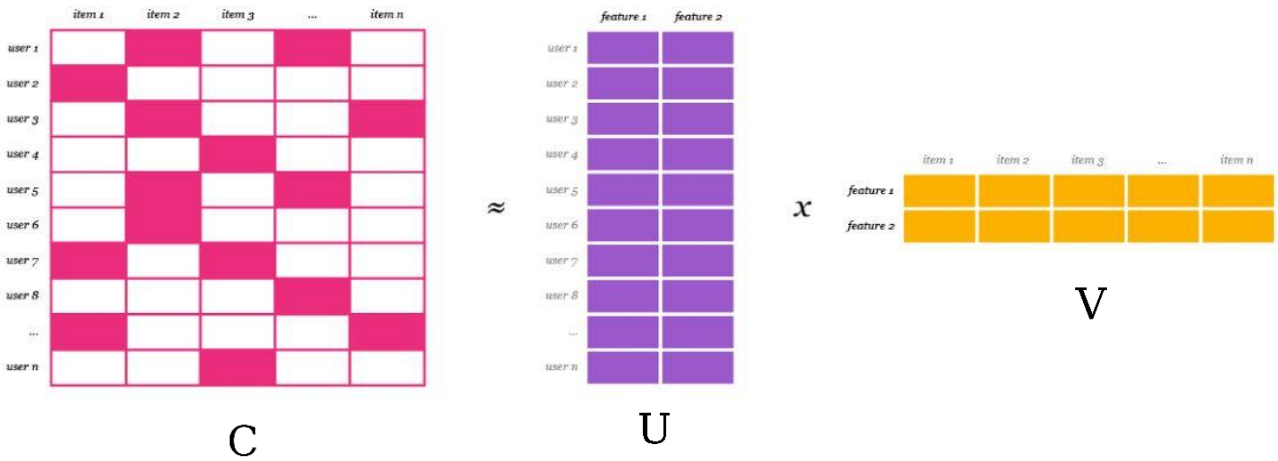


Figure 3.1: Implicit Matrix Factorisation

In Figure 3.1, the recommendation problem reduces to learning user factors and item factors. There are many matrix factorization techniques available but it is to be ensured that the technique chosen should be applicable to implicit feedback datasets by handling the missing data in sparse matrix. The dot product of $\mathbf{U_u}$ and $\mathbf{V_i}$ will give the preference score of user **u** to location **i**. Based on this top k locations can be inferred and recommended to each user. This will however fail in the case of the cold-start problem, specifically, recommending locations for new users. A general solution is to integrate collaborative filtering with content-based filtering.

3.2 Implicit Feedback Based Content-Aware Collaborative Filtering

In addition to the user-item check-in matrix C, users have features, such as profiles and textual content, provided in social networks like Twitter and Facebook, and locations have features, like category hierarchy and geographical information. After performing tokenization on textual content and discretizing continuous features (e.g., ages), all user features are encapsulated into a sparse user-feature matrix $\mathbf{X} \in \mathbf{R}^{\mathbf{M}\times\mathbf{F}}$, where **F** is the number of user features. Similarly, location features are also encapsulated into a sparse location-feature matrix $\mathbf{Y} \in \mathbf{R}^{\mathbf{N}\times\mathbf{L}}$, where **L** is the number of location features. Each entry $\mathbf{x_{u,f}}$ in matrix **X** is the value of the $\mathbf{f^{th}}$ feature of user **u** and $\mathbf{y_{i,l}}$ is the value of the $\mathbf{l^{th}}$ feature of location **i**. However, before feeding them into the model, these two feature matrices may need further preprocessing by applying tf-idf transformation and normalization (or standardization). After feeding them as input to the ICCF users and locations, as well as their features, are mapped into a joint latent space. For optimizing ICCF, multiple methods are taken into consideration and the best one is chosen based on predictive accuracy, time taken for training and ease of programmability.

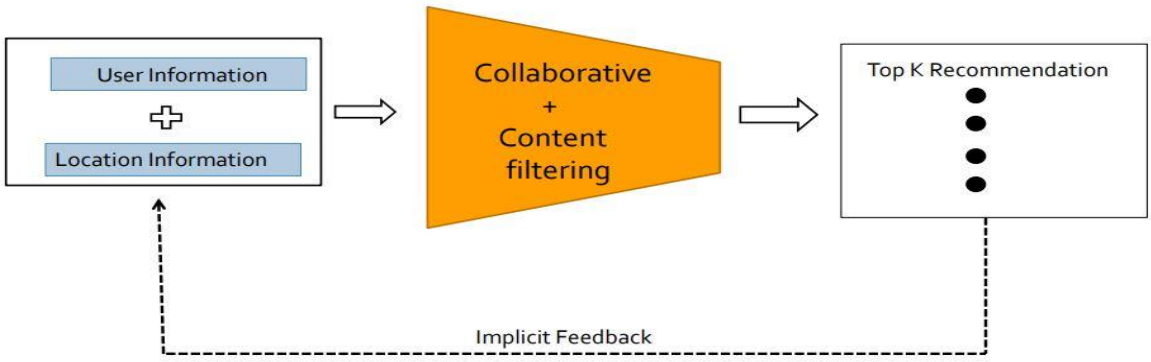


Figure 3.2: Overview of ICCF location recommender

In Figure 3.2. provides a broad overview of ICCF and its working in recommending location to users, both existing users (warm start case) and new users (cold start case). The following sections covers the theoretical aspects of ICCF in detail.

3.2.1 Content + Collaborative filtering

To overcome the disadvantages of CF based models, a hybrid model is to be used which contains both the mobility history, required by CF, and additional information, which provides the description of the item. Using this, the cold start problem normally seen in CF based models can be addressed. A system which combines content-based filtering and CF could take advantage from both the representation of the content as well as the similarities among users.

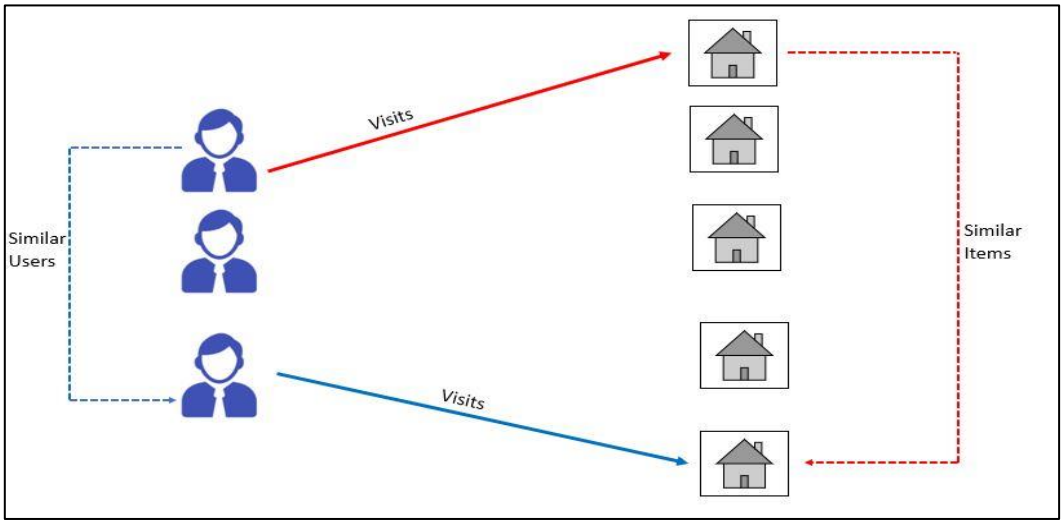


Figure 3.3: Hybrid Location Recommendation System

In collaboration via content both the rated items and the content of the items are used to construct user profile. Content-based techniques are used for the selection of terms which best describe the content or characteristics of the items.

In Figure 3.3, both user similarity and location similarity are checked. Hence it is a hybrid model that uses both CF and content-based filtering in calculating the user preference for individual locations.

3.2.2 User Content + Location Content

Users are often linked to other social networks in LBSN, which can be used in leveraging social media data to imply users' interest, an essential element for capturing users' visiting behavior. Moreover, additional textual review information and descriptive information about the location can be obtained in similar manner. This can be used in building location profile and user profile which significantly helps in improving recommendations. It is to be noted that, mostly sematic content is to be used in building these profiles, thus topic modelling or sentiment analysis is to be performed on this social network data. Both the information is correlated to each other; thus, one can be used to infer the other. Social, temporal and geographical patterns can be further observed in the collected information which can be leveraged to improve recommendation.

Figure 3.4: Correlation between Location information and user information leveraged from social network

In Figure 3.4, the user information leveraged from social networks such as twitter is shown to be correlated to the location category. Sentiment analysis or topic modelling may be needed on the additional information collected as it is the semantic or sentiment of the actual tweet relating to the location that is of use in recommendation.

3.2.3 Implicit Feedback

Human mobility data is form of implicit feedback dataset as users generally don't leave explicit ratings for the location they visit. Greater the number of visits to a specific POI, greater the user preference towards that specific POI. Hence it is an indirect reflection of user behavior and can be used in recommendation. Some of the specific characteristics of implicit feedback:

- No Negative Feedback: It is hard to reliably infer which items a user did not like. For example, if a user didn't visit a particular location, it may be because didn't like or the user was unaware of the location. It is of utmost importance to address the "missing" data, which indicates most of the negative feedback expected to be found.
- Noisy: Some method of positive feedback discounting is needed as implicit feedback is inherently noisy.
- Preference Vs Confidence: The numerical value of explicit feedback indicates preference, whereas the numerical value of implicit feedback indicates confidence.

  Let r_{ui} be the user-location visit frequency.

  p_{ui} (preference)is obtained by binarizing r_{ui} values.

$$p_{ui} = \begin{cases} 1 & , r_{ui} > 0 \\ 0 & , r_{ui} = 0 \end{cases} \quad (3.1)$$

  c_{ui} (confidence) is obtained from r_{ui}

$$c_{ui} = \begin{cases} \alpha(r_{ui}) + 1 & , r_{ui} > 0 \\ 1 & , r_{ui} = 0 \end{cases} \quad (3.2)$$

where $\alpha(r_{ui})$ is a monotonically increasing function with respect to r_{ui} so that positive confidence increases with visit frequency.

In a mobility dataset, a user's visit to a location only implies her positive preference, thus users' visited locations are considered positive examples and the visit frequency to locations determines the confidence level of positive preference. However, since their negative

preference for unvisited locations has not explicitly been observed, all unvisited locations are considered "pseudo" negative and the confidence in the negative attitude of unvisited locations is significantly less than the positive attitude of visited locations. This is done by assigning the confidence level of the preference for negative locations to the same value, e.g., 1, as done in equation (3.1) and (3.2).

As mentioned above, ICCF takes a user-location visit count matrix C, a user-feature matrix X, and a location-feature matrix Y as inputs. Based on these, ICCF first generates the weighting matrix W and the preference matrix R according to equation (3.1) and (3.2). It then defines the prediction preference of a user u for a location i as (3.3):

$$\hat{r}_{u,i} = (p_u + U\, x_u)\,(q_i + V\, y_i) \tag{3.3}$$

where each row of latent matrices $\mathbf{U} \in \mathbf{R}^{F \times K}$ and $\mathbf{V} \in \mathbf{R}^{L \times K}$ represents latent factors of user features and location features. Consequently, not only users and locations, but also their features are mapped into a joint latent space, where the inner product between them indicates one's preference for another. If the ids of both users and locations are also considered as features and encapsulated into $\{\tilde{x}_u\}$ and $\{\tilde{y}_i\}$ the prediction preference is simplified as (3.4):

$$\hat{r}_{u,i} = \tilde{x}_u \tilde{U} \tilde{V} \tilde{y}_i \tag{3.4}$$

where $\tilde{U} \in \mathbf{R}^{(M+F) \times K}$ is obtained by concatenating $\{p_u\}$ and $\mathbf{U}$ by rows ($\tilde{V}$ shares a similar meaning). Based on the prediction function (3.4), an objective loss function, taking into account the varying confidence of preference with visit frequency, is then formulated as (3.5):

$$L = \frac{1}{2} \sum_{u,i} w_{u,i} \left(r_{u,i} - \tilde{x}_u' \tilde{U} \tilde{V}' \tilde{y}_i \right)^2 + \frac{\lambda_1}{2} \left| \tilde{U} \right|_F^2 + \frac{\lambda_2}{2} \left| \tilde{V} \right|_F^2 \tag{3.5}$$

CHAPTER 4
EXPERIMENTATION

The following section covers the experimental aspects of the project which include the dataset description, the evaluation methodology used for evaluating the recommendation generated and the experimental setup which describes the models tested, parameter setting and the general overview of the model tested. The various technologies used for the project are also listed in this chapter.

4.1 Data description

ICCF is evaluated on a large-scale location-based social network dataset crawled from Jiepang, a Chinese location-based social network. The following table 4.1 shows the summary of the check-in data collected.

Table 4.1: Check-in Dataset properties

No. of unique users	55650
No. of unique places	213673
Total no. of check-ins	3464798
Sparsity	0.9998324518253869

Since many Jiepang users are linked to Weibo, a Chinese Microblog, rich semantic content, such as tweets and tags, and profile information, including age and gender, from users was scrapped using weibo-scraper[11], a python based weibo tweet scraper by which weibo tweets was crawled without authorization. This gives us the user profile dataset which contains the user id and profile information along with user tweets. In similar manner location dataset was generated which contains the location id and basic location information. The user id and location id can be used to merge with the check-in dataset.

The user profile dataset contains basic profile information like age, place, gender and horoscope along with tweets. Because of authorization restrictions and lack of tweets, only

53.85 % of the users had tweets which were capable of being crawled. This was followed by performing RAKE (Rapid Automatic Keyword Extraction) on tweets to extract keywords using Rake_For_Chinese[12]**,** a Python implementation of the RAKE algorithm. This package required the chinese text segmentation package Jieba[13] to perform word segmentation.

The location profile dataset has geographic information like latitude, longitude and name of place along with two levels of category hierarchy, where the first level contains 8 coarse categories and the second level contains 157 fine categories .16.6% of the category hierarchy is missing for the entire location dataset.

4.2 Evaluation Methodology

Each user is presented with top k location recommendation by the model based on the user item preference score calculated. The evaluation of the recommendations is based on hit ratio, ratio of number of users for which the model was able to recommend locations that the user actually did visit which were hidden during input (hit count) to the total number of users. Two different related measures are defined based on entity that is hidden before input to the model:

i. One item hide@k: here for each user, one location out of all the unique locations the user visited is hidden and the hit count get incremented only if the model is able to recommend the hidden location. Users who visit only one location are exempted from this evaluation. It is more restrictive than the other measure and is an indication of how many new places the model is able to recommend thereby helping in location discovery.

ii. One visit hide@k: here for each user, one visit out of all the user check-ins is hidden and the hit count get incremented only if the model is able to recommend the hidden location visit. It is more relaxed than the other measure and indicates the general accuracy of the model.

In both these measures, k indicates the number of locations recommended to each user based on preference score. Model tries to maximize the score as much as possible as better score implies a better performing model.

4.3 Experimental Setup

Figure 4.1 shows the general overview of the model used. The model is implemented in Google Colaboratory, a free Jupyter notebook environment that requires no initial setup and runs entirely on the cloud and is written in python. Two scenarios are taken for consideration:

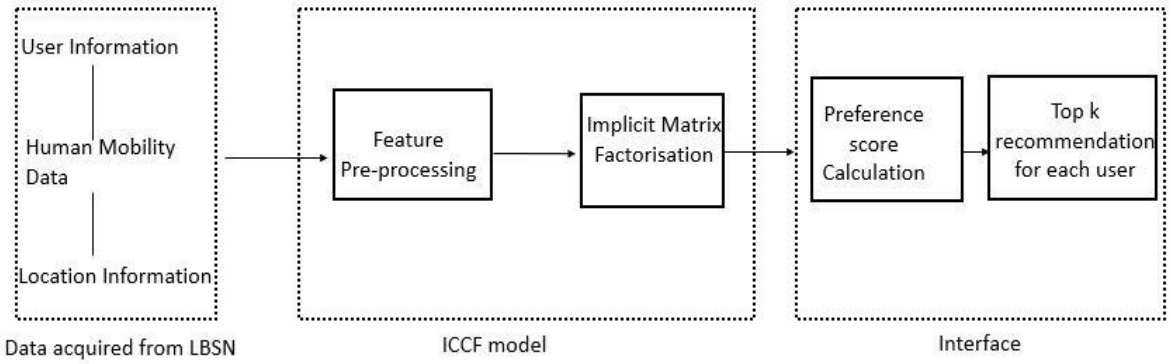


Figure 4.1: General overview of the model used

i. Warm start scenario, where the check-in data itself is given as input to the model and the accuracy of the model is evaluated.
ii. Cold start scenario where new users are defined and model is executed to generate recommendation for new users.

In warm start scenario, 10% of the dataset is hidden, based on the evaluation measure chosen, as the held-out set and the remaining 90% forms the training set. Results are evaluated based on recommendations generated by model. For purposes of uniformity, top 10 recommendations for each user is taken for consideration i.e. k = 10 for the evaluation measures.

The following algorithms are tested:

i. Lightfm [14] - A Python implementation of a recommendation algorithm suitable for both implicit and explicit feedback, including efficient implementation of some common ranking loss algorithm. It's easy to use, fast, supports multithreaded model estimation and produces high quality results. It also supports incorporating both item and user metadata into the traditional matrix factorization algorithms. This algorithm represents each user and item as the sum of the latent representations of their features. This allows recommendations to generalise to new items (via item features) and to new users (via user features).

ii. Implicit [15] - A Python implementation of recommendation algorithm for implicit feedback datasets known as Alternating Least Squares which is described in the papers Collaborative Filtering for Implicit Feedback Datasets and Applications of the Conjugate Gradient Method for Implicit Feedback Collaborative Filtering.

iii. MRec [16] - A Matlab implementation of recommendation algorithm targeting item recommendation from implicit feedback and rating prediction-based recommendation. It supports cross validation and holdout evaluation. It includes several important algorithms tailored for implicit feedback and content-aware collaborative filtering for implicit feedback. Also, it includes some state-of-the-art algorithms useful for location recommendation.

Each algorithm is executed against the same dataset and both the evaluation measures are used to evaluate the accuracy and quality of the recommendation algorithm chosen and is used to study the effects of user information and location information on recommendations in both warm start and cold start case. Each algorithm is executed with similar parameter setting and also results are averaged out. The number of latent factors for user and item factor matrix

is taken as 50 and the number of iterations in each model is taken as 20. All other parameters are kept as default as given in the algorithm. This setting is intended to reflect the balance between model accuracy and the computational cost of larger vectors in production systems.

CHAPTER 5
RESULTS AND DISCUSSION

The following chapter presents the various experimental results obtained using the ICCF model on the various recommendation algorithms along with suitable explanations for the results. Each result contains 2 entries: hits and hit ratio. Hit refers to the total number of recommendations that the model predicted found within the test or hidden set for each user and Hit ratio corresponds to the ratio of Hit and the total number of users. It indicates the average number of recommendation accurate for each user.

5.1 Effects of User and Item Feature Matrices

Results with the entire dataset with and without user and item features with number of latent factors k = 20, number of iterations = 50 and one visit hide@10 measure are as follows:

Table 5.1: Evaluation results with and without user and item features

	Without user and item features	With user and item features
Implicit	3218 0.0949	----
LightFm	5577 0.1643	6782 0.2000
Mrec	8772 0.2581	9750 0.2896

This clearly shows a significant increase in accuracy when using user and item features along with mobility dataset as user and item features are used to refine the preference score for user item pairs based in which recommendations are generated. Mrec seems to perform comparatively better than the other two recommendation algorithms because of the method it uses to compute and optimize the user factor and item factor matrices. Implicit doesn’t have

any facility to accept user and item feature matrices and its performance is comparatively worse than the other two used. User feature matrix consists of the user profile and tweet keywords extracted and the item feature matrix consists of categorical description along with location.

5.2 Effects Related to Visit Hide and Item Hide

Results with the entire dataset without user features and item features with number of latent factors k = 20, number of iterations = 50 and both one visit hide@10 and one item hide@10 measure are as follows:

Table 5.2: Evaluation results one visit hide@10 and one item hide@10

	One visit hide@10	One item hide@10
Implicit	3218 0.0949	295 0.0063
LightFm	5577 0.1643	1147 0.024
Mrec	8772 0.2581	7451 0.2581

This indicates how one item hide@10 is stricter measure than the other as it measures how much the model is accurate in predicting new locations that the user is likely to visit. In all cases one visit hit@10 values seems to greater than one item hide@10.Mrec seems to perform extremely well in one item hide@10 when compared to other two as indicated by its improved Hit ratio.

5.3 Comparing Time Taken to Factorise

Results with the entire dataset with and without user and item features with number of latent factors k = 20, number of iterations = 50 and time take to train or fit the model in seconds taken as measure are as follows:

Table 5.3: Time taken to factorise with and without user and item features

	Without user and item features	With user and item features
Implicit	1.94s	----
LightFm	109.17s	292.78s
Mrec	338.20s	453.61s

All algorithms were executed in multi-threaded mode where in parallel execution of code occurred. Implicit seems to designed for fast execution as its training time was very less when compared to other algorithms. In other algorithms it was observed, addition of user and item features seems to increase the training time significantly.

5.4 Effect of Increased Recommendations for Each User

Results with the 1000 users of the dataset without user features and item features with number of latent factors k = 20, number of iterations = 50 and one visit hide@k measure are as follows:

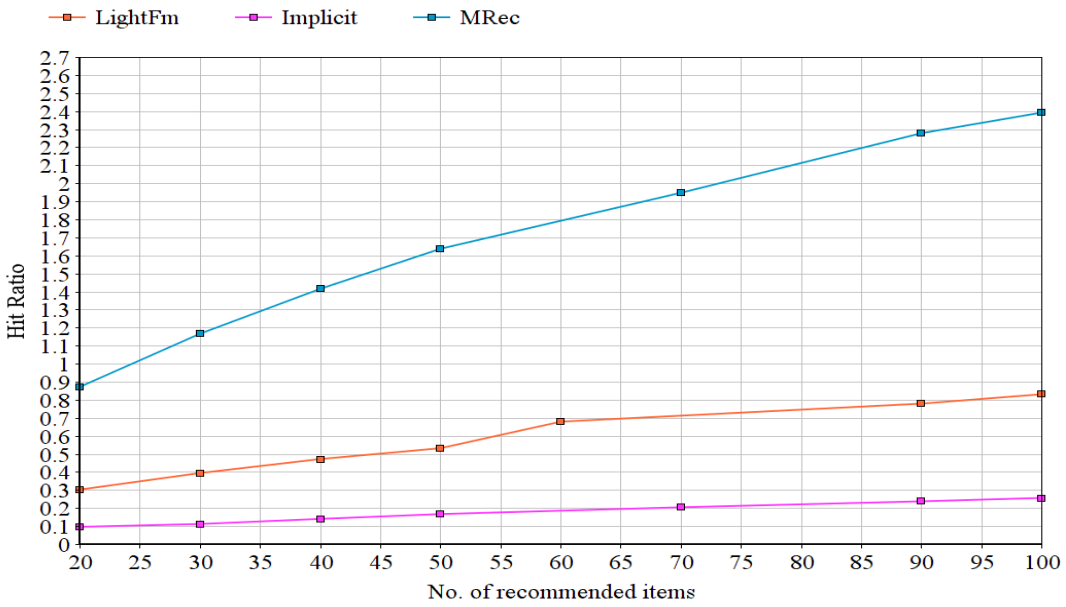
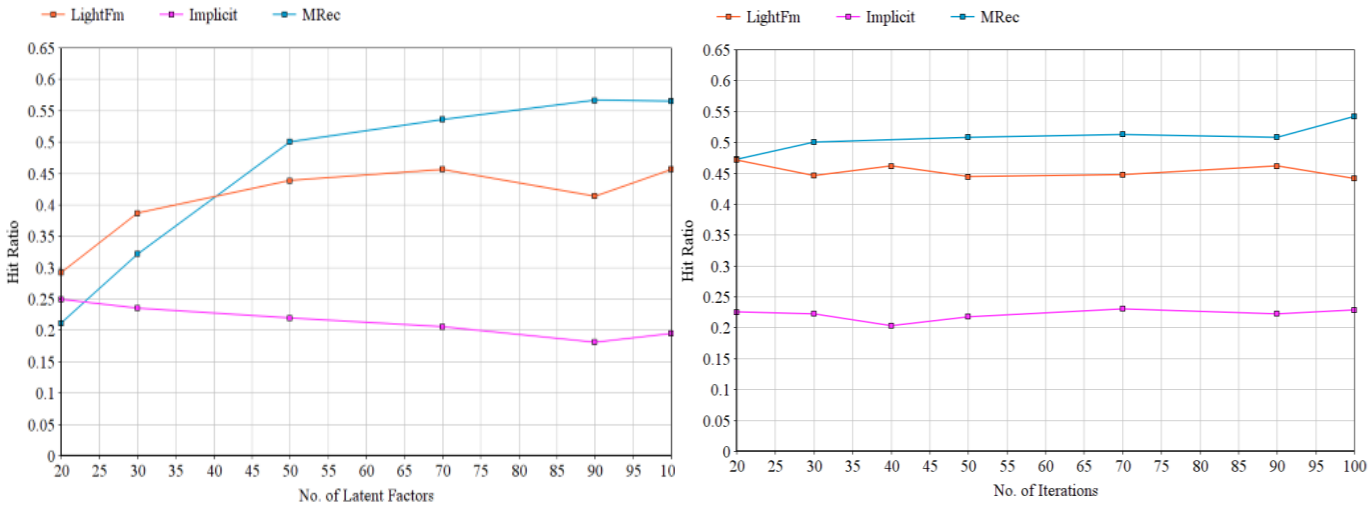


Figure 5.1: Evaluation results of varying number of recommendations for each user

It is seen upon increasing the number of predicted items the accuracy seems to increase in all cases as more items predicted, more the chances of a Hit. Mrec again seems to perform really well by achieving hit ratio over 2 indicating it rightly predicts 2 locations for every user of the dataset. For practical purposes, number of locations predicted for each user should be set somewhere around 40 to 20 locations and ranking maybe used to refine the results further.

5.5 Effect of Increased Latent factors and Increased Number of Iterations

Results with the 1000 users of the dataset without user features and item features, default number of iterations is 50, default number latent factors is set to 50 and one visit hide@10 measure are as follows:



(a) (b)

Figure 5.2: Evaluation results with varying (a) number of latent factors and (b) number of iterations

The two main model parameters are the number of latent factors in user factor and item factor matrices and the number of iterations to be performed. Generally, as number of latent factors increases, accuracy also increases up to a certain limit after which the accuracy either reduces or stays constant. The ideal number of latent factors is between 50 -70 factors. Similar observations are found for the number of iterations or epochs the model performs. Ideal value

for the number of iterations is found somewhere between 20-40 after which it there is so significant increase or decrease in accuracy of the model. Other results are directly inferred from graph.

5.6 Cold Start Evaluation

New users are represented by zero vector in the check-in matrix indicating user hasn't visited any location yet. Implicit doesn't have the ability to handle new users as collaborative filtering algorithm fails to handle user – item zero vector as it cannot be factorised to user factor and item factor values. Lightfm and Mrec does support recommendations for new users as it performs content aware collaborative filtering and user feature and item feature matrices are provided as input. If no additional matrices other than the check-in matrix are provided, the algorithm fails to generate recommendations. User feature and item feature matrices can be refined based on the features available and additional features may be added in future based on availability. Evaluation of cold start recommendations needs to be based on real time testing or live A/B testing and this is included as a future area of work. Recommendations are based on user features which include basic profile information such as gender, age, user location etc. along with social networking content which includes tweets extracted from the user's weibo account (similar to twitter) and basic location information such as categorical information describing the place along with general location.

The inferences from the above results are as follows. Implicit algorithm is optimized for speed and not accuracy when compared to other algorithms. It can be used for commercial use if speed of recommendations is of primary importance. Also, it fails cold start evaluation as it is not possible to generate recommendations for new users as collaborative filtering fails. Lightfm offers great performance in terms of prediction accuracy and intermediate performance in terms of speed of factorization. This algorithm is ideal for practical use and

real time application. New users are handled as the algorithm accepts user feature and item feature matrices. Additional features may be added if made available to improve recommendation accuracy. Mrec offers great prediction accuracy for both existing and new users. It also accepts user feature and item feature matrices and is capable for generating recommendation for new users. However, the algorithm is difficult in terms of programmability and takes comparatively takes longer to factorise. Also, the ideal values for the model parameters was inferred based on experimental results. It was also inferred that increasing the number of recommendations for each user tend to increase the predictive accuracy. Also, the inclusion of user and item feature matrices tends to increase the predictive accuracy.

CHAPTER 6
CONCLUSION AND FUTURE WORKS

In this project, a content-aware collaborative filtering from implicit feedback dataset framework called ICCF was used to perform comparative study of some state-of-the-art recommender algorithms in providing top k recommendation to each user based on mobility history. User features are used to refine mobility similarity between users and also handle the cold start case and thereby provide recommendation to new users and existing users. The input of ICCF for location recommendation is a large-scale LBSN dataset that contains check-in information and user information and item information dataset. User features includes basic profile information like age, gender, etc. and user tweets collected from social networking sites and location feature involves categorical description of the location. Further processing of tweets was needed to extract important tags and keywords which reflected the user preferences.

Human mobility data is an example of implicit feedback datasets. While most CF based recommenders operates on explicit feedback dataset based on 5-star rating which have explicit negative feedback. Thus, there is a need to extract negative preference from the mobility data. This is achieved by assigning lower confidence to unvisited locations. Matrix factorization for implicit feedback datasets takes user preference as well as the confidence for that preference. Implicit matrix factorisation factorises the user-item matrix to user-factor and item-factor matrices and then is used to calculate user preference score for each user item pair and the top-k values are recommended to each user and these topics were reviewed in this project.

Through this project, some of the problems associated with modern recommendation systems, location recommendation and implicit feedback datasets were explored and possible solutions were looked into. This project also focused on the importance of hybrid recommendation combing collaborative filtering and content-based filtering for overcoming the disadvantages of each other. By studying the effects of user profiles and semantic content,

it was found that they improve recommendation in warm-start cases and help address the cold-start problems. User features are more helpful in providing a personalized recommendation then compared to item features especially in the case of new users and absence of mobility history. Moreover, when integrating them together, it makes further improvement in recommendation performance, indicating they complement each other. Nevertheless, the impact of incorporating content with profile is not as large as expected where the textual content refers to the item information and profile corresponds to the user information. One reason is that textual content is not as strongly correlated with recommending novel and attractive locations as profiles. To quantify the improvement, this project used two evaluation measures based on hit ratio which measured both the accuracy of recommendation and the ability of algorithm to recommend new places and assist in location discovery. Evaluation involves hiding location visited and checking whether the model was able to recommend the hidden location based on remaining check-in history for each user.

For further work, the evaluation measure needs to be extended so that it can evaluate cold start case as well. This may include incorporating live A/B testing of recommendation to get immediate feedback from users and also being able to quantify the novelty and popularity of location so that they can be exploited in recommending new locations and providing recommendation to new users. This may be used in fine tuning of parameters so as to get optimal recommendation for each user making the model more robust and also providing a personalized experience even for new users without any mobility history. Additionally, more complex user and item features may be added to test the improvement and changes in recommendation. This may include live data collected at the point of user login such as recent tweets, weather conditions, user location with respect to item location etc. It may also include incorporating audio and visual features provided they can be properly encoded. The scalability

of the algorithms and performance of the model in a real time environment is a promising future area of work.

The topics discussed in this project are also applicable to other recommendation applications such as product recommendation or movie recommendation. Most of the data being generated these days is implicit feedback in nature. They require processing technique similar to the ones discussed in this project. Extending the model so as to support inputs other than the human mobility dataset and evaluating performance of recommendation is also a future area of research.

REFERENCES

[1] D. Lian, Ge, Y., Zhang, F., Yuan, N., Xie, X., Zhou, T. and Rui, Y, 2018, ”Scalable Content-Aware Collaborative Filtering for Location Recommendation”, IEEE Transactions on Knowledge and Data Engineering, 30(6), pp. 1122-1135

[2] V. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang, 2010,"Collaborative filtering meets mobile recommendation: A user-centred approach”, Proceedings of AAAI’10.

[3] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, 2014, “Geomf: joint geographical modeling and matrix factorization for Point-of-interest recommendation." in Proceedings of KDD'14. ACM, pp. 831– 840.

[4] Adomavicius, G., and Tuzhili, A., 2011, “Context-Aware Recommender Systems”, In Recommender Systems Handbook, pp. 217– 256.

[5] Karatzoglou, A., Amatriain, X., Baltrunas, L. and Oliver, N., 2010, “Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering”, In Proceedings of fourth ACM on Recommender systems, pp. 79-86.

[6] Kefalas, P. and Manolopoulos, Y., 2017, “A time-aware spatio-textual recommender system”, Expert Systems with Applications, 78, pp. 396-406.

[7] Hu, Y., Koren, Y. and Volinsky, C., 2008, “Collaborative Filtering for Implicit Feedback Datasets”, Eighth IEEE International Conference on Data Mining.

[8] Kawai, K. and Kitagawa, H., 2016, “Collaborative Filtering with Implicit Feedbacks by Discounting Positive Feedbacks”, In Multimedia Big Data (BigMM), 2016 IEEE Second International Conference, pp. 41-48.

[9] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, Lars Schmidt-Thieme, 2016 “BPR: Bayesian personalized ranking from implicit feedback”, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 452-461.

[10] Li, X., Cong, G., Li, X., Pham, T. and Krishnaswamy, S., 2015, “Rank-GeoFM”, Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15.

[11] X. Xarrow, “Xarrow/weibo-scraper,” GitHub, 2019. [Online]. Available: https://github.com/Xarrow/weibo-scraper. [Accessed: 01-April-2019].

[12] X. Ruoyang, “Cryptum169/Rake_For_Chinese,” GitHub, 2018. [Online]. Available: https://github.com/Cryptum169/Rake_For_Chinese. [Accessed: 01-May-2019].

[13] S. Junyi, “fxsjy/jieba,” GitHub, 2018. [Online]. Available: https://github.com/fxsjy/jieba. [Accessed: 01- April -2019].

[14] M. Kula, “lyst/lightfm,” GitHub, 2019. [Online]. Available: https://github.com/lyst/lightfm. [Accessed: 01- April -2019].

[15] B. Frederickson, “benfred/implicit,” GitHub, 2018. [Online]. Available: https://github.com/benfred/implicit. [Accessed: 01- April -2019].

[16] D. Lian, “DefuLian/recsys,” GitHub, 2019. [Online]. Available: https://github.com/DefuLian/recsys. [Accessed: 01- April -2019].