A Proximity-based Recommender System for Indoor Spaces

A Thesis

submitted to the designated
by the General Assembly
of the Department of Computer Science and Engineering
Examination Committee

by

Christodoulos Asiminidis

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN DATA AND COMPUTER
SYSTEMS ENGINEERING

WITH SPECIALIZATION
IN DATA SCIENCE AND ENGINEERING

University of Ioannina
February 2021

Examining Committee:

- **Nikolaos Mamoulis**, Professor, Department of Computer Science and Engineering, University of Ioannina (Advisor)
- **Evaggelia Pitoura**, Professor, Department of Computer Science and Engineering, University of Ioannina
- **Panos Vassiliadis**, Professor, Department of Computer Science and Engineering, University of Ioannina

Dedication

I dedicate this master thesis report to my nephew, Thanos.

Acknowledgements

I would like to thank my advisor, Professor Nikolaos Mamoulis, for all the great deal of support and assistance I have received throughout the writing of this thesis. His expertise was priceless in formulating the research questions and methodologies. Also, I would like to thank him for all the patience and undestanding he provided and for all the chances he gave me to further my research. I would like to thank my parents for their support and love they have shown during this journey. Finally, I could not have completed this thesis without the support of my sister, Sophia and my friends who were the happiest distractions to rest my mind out of my research.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abstract

Christodoulos Asiminidis, M.Sc. in Data and Computer Systems Engineering, Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, February 2021.
A Proximity-based Recommender System for Indoor Spaces.
Advisor: Nikolaos Mamoulis, Professor.

Location-based recommender systems increasingly gain popularity, due to the fact that most users who seek information or recommendations are doing so via their mobile devices. In most previous work, GPS is used for locating the user and the recommended data are related to outdoor objects. On the other hand, there are many cases where users seek for information in indoor spaces. In this case, the use of GPS for location tracking is inaccurate and technologies such as Wi-Fi and BLE are more appropriate. In this thesis, we study the development of a proximity-based recommender system for indoor spaces. We present an accurate indoor localization approach based on BLE. In addition, we deploy a fused matrix factorization model which captures the geographical influence of a location in indoor spaces by modeling the probability of the user to be interested in as a Multi-Gaussian Model. The model takes into account the user's profile, the location and past actions. We evaluate the performance of our model for different resolutions in number of users, locations, the information which is associated with the location and time. We compare our model against baseline algorithms, which do not consider temporal and/or spatial information.

Εκτεταμενη Περιληψη

Χριστόδουλος Ασημινίδης, Δ.Μ.Σ. στη Μηχανική Δεδομένων και Υπολογιστικών Συστημάτων, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, Φεβρουάριος 2021.
Σύστημα σύστασης με βάση την εγγύτητα για εσωτερικούς χώρους.
Επιβλέπων: Νικόλαος Μαμουλής, Καθηγητής.

Το GPS χρησιμοποιείται για τον εντοπισμό του χρήστη όπου τα προτεινόμενα δεδομένα σχετίζονται με την τοποθεσία του σε εξωτερικούς χώρους. Από την άλλη πλευρά, το GPS δεν παρέχει ακρίβεια στον εντοπισμό του χρήστη σε εσωτερικούς χώρους οπότε και δεν μπορεί να λάβει πληροφορίες για τα αντικείμενα που βρίσκονται εντός κτιρίων. Έτσι, λοιπόν, για να λάβει πληροφορίες γι'αυτά τεχνολογίες όπως το Wi-Fi και το BLE είναι πιο ακριβείς όσον αφορά την ακρίβεια τοποθέτησης σε εσωτερικούς χώρους.

Λόγω του δυνητικά μεγάλου όγκου πληροφοριών που περιέχονται σε εσωτερικούς χώρους και του γεγονότος ότι ένας χρήστης πλοηγείται σε εσωτερικούς χώρους σε αυτή τη διατριβή μελετάμε την ανάπτυξη ενός ακριβούς συστήματος συστάσεων που βασίζεται στην πλοήγηση σε εσωτερικό χώρο με βάση το χώρο, το χρόνο και το προφίλ του χρήστη. Αναπτύσοουμε ένα σύνθετο μοντέλο παραγοντοποίησης μητρών που συλλαμβάνει τη γεωγραφική επίδραση μιας θέσης που λαμβάνει υπόψιν τη γεωγραφική επίδραση του χρήστη με βάση τη τοποθεσία του εσωτερικά των κτιρίων, μοντελοποιώντας την πιθανότητα του χρήστη να ενδιαφέρεται για μια πληροφορία. Σημαντικός παράγοντας σε αυτό παίζει ο χρόνος, καθώς η πληροφορία που συνίσταται στο χρήστη είναι συναρτήσει με αυτό. Συνεπώς, το μοντέλο μας λαμβάνει υπόψιν το προφίλ, το ιστορικό, καθώς και την τοποθεσία του χρήστη.

Αξιολογούμε την απόδοση του μοντέλου μας για διαφορετικές αναλύσεις όσον αφορά τον αριθμό των χρηστών, τοποθεσιών, τις πληροφορίες που σχετίζονται με την τοποθεσία και την ώρα. Τέλος, συγκρίνουμε το μοντέλο μας με βασικούς αλγορίθμους που δεν λαμβάνουν υπόψιν το χρόνο ή/και το χώρο.

Chapter 1

Introduction

In this thesis, we propose effective methods for recommending personalized and relevant information items to a target user, considering the user's location and the current time. In the following, we give a general overview of the problem and the work carried out throughout this thesis. Specifically, we describe our research motivation in Section 1.1, present the problem formulation in Section 1.2, the challenges in Section 1.3 and the organization of the thesis in Section 1.4.

1.1 Research Motivation

The increasing importance of the Internet as a medium for electronic and business transactions in conjuction with the amount of information sources and items which increases exponentially, necessitates filtering the information which is provided to the users. This is the driving force for the development of recommender systems technology. The information is based on the user's past activities or is derived from other users' preferences explicitly [1]. The goal of a recommender system is to offer effective personalized recommendations to the user that she might not even be aware of [2].

In particular, the location of the user plays a significant role in the accuracy of the recommended information. We develop a proximity-based recommender system which provides personalized information to the user while she navigates in indoor spaces, where the information items are associated to locations. Although there is a lot of research on location-based recommender systems for outdoor spaces (based on GPS technology) [3, 4, 5, 6], there is a limited research in providing recommendations for indoor spaces. We focus on the development of a location-based recommender system for indoor spaces using Bluetooth Low Energy (BLE) beacon technology, which helps us to achieve good indoor location tracking accuracy. We do not use GPS based tracking because it is inaccurate for indoor spaces [7].

To better understand the application scenario, consider the case where the user would like to learn about information items while navigating indoors. When the user enters a building, a smartphone application is launched at her mobile device. While the application is running on the smartphone, the position of the user is tracked indoors through the signals received from the beacons placed on the corridors of the building. Hence, when a user is outside of a particular office, relevant information items may appear on her screen of the smartphone (push-based notifications). The proximity-based recommender system that we develop in this thesis selects the most appropriate personalized information present to the user. The information provided to the user is a ranked list of information items based on user's profile, location and time. In our case, the recommendations target different users and the information provided may vary, from location to location and from time to time.

In the main use case of our scenario, users are students who walk in the building of our institute and information items are text snippets or URLs which correspond to the different offices, labs or rooms in the building. Multiple information items may correspond to a single location. For example, when a student visits the office of a teacher, the student may be interested in the office hours of the teacher, in his/her profile or homepage, in the courses that he/she teaches, in announcements related to the teacher or his courses, etc.

Besides applications that can directly be served by our system (e.g., recommendations in indoor office spaces, other domains could also benefit from our solution. For example, in proximity-based marketing, customers who walk in a department store may receive personalized notifications or information about products in their vicinity. In museums, visitors are notified with information about nearby exhibited items.

1.2 Problem formulation

In this thesis, we focus on the problem of location-based recommendation by using users' historical behavior indoors.

Defition 1 (Recommendation item) A recommended item is a piece of information associated to an indoor location (e.g., office, information desk).

Definition 2 (User Activity) A user activity is modeled by a quadruple (u, v, l_v, t) which indicates that the user u accessed information item v located at l_v at time t.

Problem (Recommendation) Given a target user u, her current location l and the current time t, our goal is to recommend a list of information items that u would be interested in.

Currently, there is not much work in information recommendation in indoor places. We present the challenges of our work in the following section.

1.3 Challenges

The first challenge is the accurate tracking of users' indoor activity. Previous work for recommendations in indoor spaces is based on the use of Wi-Fi for location tracking [8]. We improve the accuracy of location tracking by using BLE beacons. We compare BLE technology to Wi-Fi only because other technologies such as RFID are not built in smartphones, hence they are not applicable in the general case (at least without a cost). While Wi-Fi is easy to implement and does not require extra hardware, it consumes more power than BLE technology. In addition, BLE provides better accuracy in positioning indoors [9, 10]. Table 1.1 shows that BLE is more suitable for indoor localization compared to Wi-Fi in terms of accuracy [11]. Specifically, signals are not strongly influenced by the environment because of their lower transmission power. BLE adopts a channel hopping mechanism, leading to fewer package collisions. BLE has a much higher sampling rate, which makes it easier to filter out outliers. These advantages has led us to use and exploit BLE for precise indoor localization.

Table 1.1: Comparison of Wi-Fi and BLE wireless communication technologies.

Attribute	Wi-Fi	BLE
Signal Rate	54 Mbps	720 Kbps
Normal Range	100 m	30 m
Transmission Power	20 dBm	1 dBm
Energy Consumption	50-100 mA	15 mA
Hardware Cost	high	low
Indoor Accuracy	3-10 meters	1-2 meters

The number of users, locations, information items and time are the parameters that have been taken into consideration. All these different parameters make the problem space extremely sparse, which affects negatively the quality of recommendations. There could be many of different recommending items. Hence, the recommender system may have to choose from numerous information items which are associated to locations near the user.

The proximity-based recommender system for indoor spaces we are developing exploits users' indoor preferences on items associated with locations l_v. When users are navigating indoors through a smartphone application, information items appear on her screen in close proximity based on their current position and time. Our goal is to select and recommend to the user the nearby items that are the most relevant to the user, considering also the current time.

Users express their interest by clicking on an recommending item. Hence, the user-item pairs form a non-negative utility matrix, where the entries are unary ratings because user clicks on preferences derived from their activities indoors by clicking information items.

To overcome the data sparsity issue, the proximity-based recommender system we propose combines collaborative model-based methods and content-based recommender techniques.

The data sparsity issue is faced by adapting the Probabilistic Factor Model which belongs to the family of Matrix Factorization models. We use the particular model since it is a promising model for non-negative data such ours.

A related issue is the cold-start problem, in case the user has just started navigating and the user has clicked a few or no items. We address this by disregarding the user's profile and instead use just the user's current location and time together with historical data from other users, for recommendation.

Our contributions can be sumarized as follows:

- We accurately track users' behavior indoors using BLE technology.
- We develop a fused matrix factorization technique to model the users' preference and behavior in space and time.
- We evaluate our model under different resolutions in space and time, different number of users and information items.

1.4 Thesis outline

The rest of this master thesis report consists of four chapters. Chapter 2 offers an overview of indoor localization and describes existing related work found in the literature. Also, it gives an overview about recommender systems based on the solutions, on information collection methods and evaluation methods and compares our system to the ones found in the literature. It also provides a brief overview about collaborative filtering recommender systems and describes related work on location-based recommender systems found in the literature. Chapter 3 describes our research methodology which consists of two parts, the indoor localization and the recommender system. Chapter 4 evaluates our approach under different resolutions in space and time, different number of users and information items. Chapter 5 concludes our work.

Chapter 2

Related Work

In this Chapter, we discuss the goal, techniques of indoor localization and compare our approach to the ones found in the literature in Section 2.1. Section 2.2 gives an overview of recommender systems, classifies them based on their solutions, information collecting methods, describes the evaluation methods and compares related work found in the literature to our methodology.

2.1 Indoor Localization

Indoor localization aims at continuously tracking the locations of mobile users indoors. Indoor localization facilitates recommendations based on the user's location and preference. In order to provide more accurate products, services and information items, the coordinates of the locations associated to information items and different temporal states are exploited in order to determine and rank the recommendation list.

Without utilizing the indoor localization process, there would be too many recommendations that could be provided to the user. The indoor localization method used in this thesis helps tackle this issue by suggesting to the user only items which are relevant to her location and preferences.

In Section 2.1.1 we give a brief overview about the indoor localization techniques and present previous work found in the literature.

2.1.1 Indoor Localization Techniques and Existing Related Work

Two of the main techniques employed to locate a user in an indoor setting or environment are fingerprinting and trilateration. The fingerprinting technique focuses on obtaining fingerprints or features of the environment where the localization system is to be used [12]. Initially, different RSSI measurements are collected during an offline phase. The online measurements, which are obtained in real-time, are compared with the offline measurements to estimate the user's location as shown in Figure 2.1. In order to position the user indoors, the most popular positioning algorithm is the weighted k-NN. This particular algorithm calculates the Euclidean distance between the online RSSI measurements and the offline ones. The distance vector that is created is sorted in an ascending order. The smallest k distances are chosen and the inverse of the distance is assigned as weight to each beacon reference. Hence, the k nearest neighbors are found based on the online positioning [13]. The trilateration method focuses on estimating the distance between the user and the at least three beacon references. Since, that is the technique we use for the purpose of this master thesis report, we provide more information and discuss how we adapt it to our problem in Chapter 3.

The fact that mobile devices are becoming prevalent has put great stress on much of the research undertaken in the field of indoor localization, and the use of these devices can hardly be ignored in this day. Recent research [14] found that most of the people spend 87% of their time indoors, meaning that the quantity of the time spent indoors and the use of mobile devices is proportional.

There is an in-depth analysis in [15] on the different types of indoor positioning systems, including Wi-Fi and Bluetooth. The process of trilateration is a technique of estimating the indoor location using the approximate location of the short range wireless devices and the distance between the device and the user. This has led to indoor localization solutions in order to find the user's preference on location [16].

Ng et al. [17] propose an interactive framework, called Notify-and-Interact which

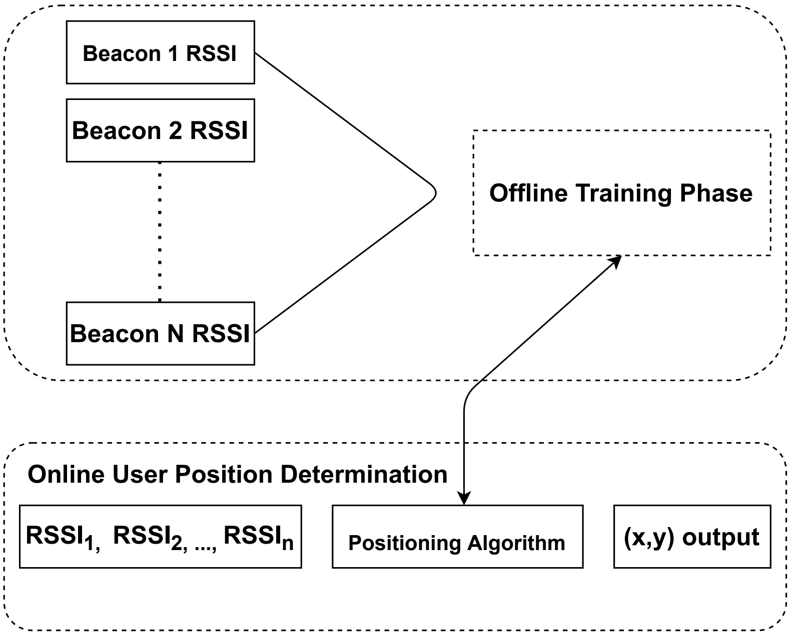


Figure 2.1: Fingerprinting method.

consists of BLE beacon devices to notify users and smartphones to interact with them. The framework is presented in the Ping Yuan and Kinmay W Tang gallery in Hong Kong. The framework is classified into three parts. The first part is the "Notify" via Beacon to advertise signals where time users spent in front of an exhibit is taken into consideration. If a specific period of time exceeds a pre-defined time, then a notification invites users to interact. The second part is the "Interact" via Smartphone. Users' smartphones scan, notify and interact events between the smartphone, the beacons and the cloud. The third and last part includes a content management system which manages and synchronizes the content from beacons and smartphones. Experimental results have shown that private notifications on users' smartphones are more efficient in increasing users' awareness and receive higher acceptance than the public ones. The functionality difference between our approach and is that in our case the time duration has not been taken into consideration because the goal is provide to the target user items that are more relevant and personalized in order to minimize the time spent indoors and find the appropriate information as fast as possible. The difference on the notifications is that the Notify-and-Interact framework pushes notifications coming from a CMS, whereas our approach suggests a list with the top recommended items based on location and time.

Another research found in the literature is by Uttarwat el at. [18], who propose the BeaLib system in which notifications are pushed to user's smartphone when close to a beacon device. It is a Beacon Enabled Smart Library System which consists of three beacons and an Android application. The work took place on the third floor in Auckland University of Technology library. When a user is close to a particular beacon, a notification appears on the smartphone with the information related to the book. In case a user passes by another beacon, then the application pushes a different notification related to another book depending on the location inside the library. For the purpose of the thesis, based on user's location, a push notification appears on her screen. The drawback in this approach is that no recommendations are made based on user's preferences and past activity. Also, pushing notifications all the time might bother the user inside the library.

Previous works have shown that different techniques such as RFID and Wi-Fi can estimate the position of the user indoors [19, 20, 21]. In order to use RFID appropriate equipment is required and is not in the scope of this thesis.

Section 2.2 gives an overview of recommender systems, Section 2.2.1 describes the fundamentals of recommender systems, Section 2.2.1 refers to the basic models of the recommender systems, Sections 2.2.2, 2.2.4 describe the solutions and information collecting methods respectively. Sections 2.2.5, 2.2.6 give a brief overview on the different accuracy estimation either based on ratings or rankings. Section 2.2.8 analyzes existing related work found in the literature and compares our model to it.

2.2 Recommender Systems Overview

Since the mid-90s, the interest in recommending items to users has been elevated in industry and academia especially when the first scientific papers on collaborative filtering were published [22, 23, 24]. Netflix has contributed significantly to the research community as a result of the Netflix Prize contest. This contest was designed to provide a forum for competition among various collaborative filtering algorithms contributed by contestants. Netflix offered a prize of $1m to the first person or team to beat the recommender algorithm they developed themselves.

There are three attributes in a data table used to develop a recommender system: user, item and preference. We are illustrating MovieLens which is non-commerical, free of advertisements, personalized and one of the most popular open recommender systems [25].

Table 2.1: Term definition for MovieLens and proximity-based RSs.

	MovieLens	Proximity-based RS
user	users sign up on the system	users sign up on the app
item	movies on the online system	items shown on the app
pref.	ratings given by users to movies	clicks given by users to items

Table 2.2: Example of the proximity-based RS.

User	Item 1	Item 2	Item 3
4	2020-10-10T13:45:30	2020-10-09T12:30:32	
23		2020-11-09T09:45:43	
57	2020-10-06T14:15:14		2020-12-01T09:30:45

Table 2.1 summarizes the differences between Movie Lens RS and proximity-based RS in terms of the definition of the three terms. The users refer to the audience who rate the movies they like and are the ones who receive the recommendations. The items refer to different movies available for the users to choose. The preference refers to the degree that a user likes a certain item. One way to represent the preference is to explicitly rate values between 1 and 5, which is used by the Movie Lens online system. Preferences are also implied by user behaviors such as clicks and purchases.

The data that are used in any recommender system is represented by a matrix which is usually referred to as utility matrix. An example for proximity-based RS is shown in Table 2.2. In the leftmost column, users are referred to their IDs such as 23, the items are represented by the information items associated to a location that the user expressed her interest while navigating indoors. Each location may have different number of information items. The items colored as blue indicate that they belong to the same location and the item colored as red indicates that they are relevant to a different location. Few values in the utility matrix are left empty indicating that the user has never clicked for the corresponding item at a specific location.

2.2.1 Basic Models of Recommender Systems

The unknown preference of a user to an item can be estimated in various ways based on the available data as follows (a). user-item interactions, such as ratings or buying

Table 2.3: Goals of different types of RSs.

Methods	Goal
Collaborative Filtering	Recommendations based on collaborative approach that leverages the ratings and actions of my peers/myself
Content-based	Recommendations based on the content I have favored in my past ratings and actions
Hybrid	Recommendations based on my ratings, of the kind of content I want

behavior, and (b). the attribute information about the users and items such as textual profiles or relevant keywords [26]. In this section, we pay attention to three aspects that characterize recommender system, and are used to identify the recommender system developed in this thesis. These factors are the recommendation solutions, the information gathering techniques and the evaluation methods.

2.2.2 Basic Models of Recommender Systems on solutions

An empty preference is predicted in various ways exploiting techniques from machine learning. Recommender system methods refer to the approaches the recommender system. Recommender systems can be classified into the following categories based on their approach:

- Collaborative Filtering Models: The RS recommends items that users with similar behavior preferred in the past.
- Content-Based Recommender Systems: The RS recommends items similar to the ones the user preferred in the past.
- Hybrid Recommender Systems: The RS combines the strength of collaborative filtering models and content-based recommender systems to create techniques that can perform more robustly in a wide variety of settings.

Table 2.3 describes the conceptual differences in the three categories of recommendations [26].

Collaborative Filtering methods use the collaborative power of ratings provided by multiple users to make recommendations. They leverage the similarity of the user's ratings for co-rated items. The main challenge is that the underlying utility matrix is sparse. The number of ratings that have already gained is very small in contrast to the number of ratings that have to be estimated. For instance, in the MovieLens database, users rate the movies etiher liking or disliking of specific movie. Most users would have viewed only a small fraction of the large universe of available movies. As a result, most of the ratings are unspecified.

In content-based RS, the descriptive attributes of items are used to make recommendations. The term 'content' refers to these descriptions. In content-based methods, the ratings and clicking behavior of users are combined with the content information available for each item. For instance, in MovieLens database, consider a case where John has rated a movie highly, but we do not have access to the ratings of other users. Therefore, collaborative filtering methods are ruled out. However, the item description of the movie contains similar keywords as other movies. In such cases, these movies can be recommended to John. There are plenty of techniques to automatically extract characteristics from items, among which the most widely used on is Term Frequency/Inverse Document Frequency (TF/IDF) [27] that is used to specify keyword weights in text-based items.

Content-based models work well in practice, though it has some drawbacks in making recommendations for new items. This is because other items with similar attributes might have been rated by an active user. The model should be able to leverage these ratings in combination with the item attributes to make recommendations even when there is no history data for that item.

Hybrid RS combines various aspects from different types of RS, such as collaborative filtering and content-based as described. They have the power that they deploy different types of machine learning algorithms to create a more robust model.

2.2.3 Basic Models of Recommender Systems based on information collecting methods

The ways the user gives feedback correspond to the information collecting methods. Recommender systems can be classified based on the user's involvement as follows:

- Intrusive recommender system: An important level of user's participation is required to get the feedback.

- Non-intrusive recommender system: Little or no explicit user involvement is required to get the feedback.

A way to collect user feedback is to explicitly ask users to rate the items they have reviewed or purschased. In the MovieLens database users are asked to rate movies in the interval-based ratings scale of 5 point that the system draws ratings from a 5-point ratng scale.

A different approach to collect data is to predict the real rating a user will give for an item. Minimizing intrusiveness and keeping the accuracy of recommendations remains a significant subject in which researches find interesting because of its difficulty and promising potential. In online systems, the most common approach that is used is the non-intrusive. The common approach to collect data is the clicking behaviour which is the same approach of collecting data for our proximity-based recommender system. The proximity-based recommender system for indoor spaces is a non-intrusive recommender system because the information item clicks are collected from users while navigating indoors and storing their current location and time.

2.2.4 Basic Models of Recommender Systems based on evaluation methods

Recommender systems are deployed to provide the best recommendations. Before offering recommendations, the system should be able to decide which recommendations are the best ones for the target user. Various recommender systems try to achieve a specific purpose that results in the best recommendations. Researchers examine the performance of a recommender system comparing different evaluation methods. The main components of accuracy evaluation are the following ones:

- Designing the accuracy evaluation: The specified ratings are divided into training the model and evaluating the accuracy.

- Accuracy metrics: Accuracy metrics are used to evaluate either the accuracy of estimating the ratings of user-item interactions or the accuracy of the top-k ranking predicted by a recommender system.

2.2.5 Accuracy of estimating ratings

The error between the actual rating and the predicted one is given by Equation 2.1.

$$e_{uj} = \hat{r_{uj}} - r_{uj} \tag{2.1}$$

This error can be computed over the set E of entries in the ratings matrix on which the evaluation is performed. Mean Squared Error is one of the most common metric denoted by MSE and given in Equation 2.2:

$$MSE = \frac{\sum_{(u,j)\in E} e_{uj}^2}{|E|} \tag{2.2}$$

The square-root of the aforementioned quantity is referred to as the root mean error denoted by RMSE and given in Equation 2.3.

$$RMSE = \sqrt{\frac{\sum_{(u,j)\in E} e_{uj}^2}{|E|}} \tag{2.3}$$

2.2.6 Accuracy of estimating rankings

Many recommender systems do not estimate ratings, but they estimate the top-k items returned for each user. Assume that one selects the top-k set of ranked items to recommend to the user. For any given value k of the size of the recommended list, the set of recommended items is denoted by $S(k)$ so that $|S(k)| = k$. Therefore, as k changes, the size of the $S(k)$ changes. Let G represent the true set of relevant items (ground-truth positives) that are clicked by the user. For any given size k of the recommended list, the precision is defined as the percentage of recommended items that truly turn out to be relevant (clicked by the user) as given in Equation 2.4.

$$Precision(k) = 100 \times \frac{|S(k) \cap G|}{|S(k)|} \tag{2.4}$$

The recall metric is the proportion of good recommendations that appear in top recommendations. The metric is given in Equation 2.5.

$$Recall(k) = 100 \times \frac{|S(k) \cap G|}{|G|} \tag{2.5}$$

The trade-offs between precision and recall are not necessarily proportional. An increase in recall does not mean the value of precision will be reduced. Both, precision and recall can be summarized in F_1 score as given in Equation 2.6.

$$F_1(k) = \frac{2 \times Precision(k) \times Recall(k)}{Precision(k) + Recall(k)} \tag{2.6}$$

In this thesis, the precision, recall and F_1 metrics are used to evaluate the information needed and provide personalized recommendations to a target user u. For this reason, after receiving the top-k information items from a model-based collaborative filtering algorithm, the recommender system determines the precision, recall and F_1 scores.

2.2.7 Collaborative Filtering Recommender Systems

The goal of a collaborative recommender system is to estimate the unobserved ratings based on the items previously rated by other users. A collaborative filtering recommender system depends on the ratings rather than any domain specific information of items. There are two collaborative filtering methods as follows [28, 26]:

- Neighborhood-based/Memory-based: These methods count on similatiry patterns of ratings between users and items.
- Model-based: These methods use models that derive from machine learning to make rating predictions.

Neihgborhood-based collaborative filtering methods, also mentioned as memory-based, are the earliest methods for collaborative filtering. These methods are based on the fact that similar users behave the same and similar items are rated similarly. They can be categorized into two types:

- User-based CF methods: In order to make recommendations for a target user, ratings are estimated based on similar user.
- Item-based CF methods: In order to make recommendations for a target item, ratings are estimated based on the target item and similar items.

The model-based collaborative filtering methods are most commonly used in machine learning which are the focus of this thesis. Examples of these methods are decision trees, rule-based methods, Bayes classifiers, regression models, support vector machines, neural networks and latent factor models [29, 30]. The goal of the aforementioned techniques is to complete a matrix in which entries might be missing.

The attention of this thesis is paid on latent factor models [31]. The advantage of these methods is that they exploit the important similarities between users and items that are highly correlated. The resulting data matrix is estimated by low-rank matrices. Compared to the original data matrix, the fully low-rank matrix is estimated even with a small subset.

Our approach is based on clickstreams of information items associated to a location gathered by the users who are navigating in indoor spaces at a speific temporal state. That means that we deal with a non-negative Matrix Factorization (MF) problem in which we use multiplicative update rules to the entries of the matrices U and V in order to minimize the objective function.

The low-rank MF methods are very efficient in training since they assume that in the user-item preference matrix, only a small number of factors influence the preference by how each factor applies to that user.

In recommender systems we denote a set of users by $\mathcal{U} = \{u_1, u_2, ..., u_m\}$, a set of items $\mathcal{V} = \{v_1, v_2, ..., v_n\}$, a set of locations by $\mathcal{L} = \{l_1, l_2, ..., l_{\mathrm{k}}\}$ associated with the information items v_i and a set of temporal states by $\mathcal{T} = \{t_1, t_2, ..., t_{\mathrm{h}}\}$. The user-item preference is encoded in $R^{m \times n}$, where the entries $r_{u,v} \in R$ represent the previous information item and thus, the previous locations visited of a user $u \in \mathcal{U}$ to information item $v \in \mathcal{V}$. Also, l_v shows the information item v associated with the location l.

The Singluar Value Decomposition (SVD) estimates the matrix R by minimizing the quantity as given in Equation 2.7:

$$\underset{U,V}{\text{minimize}} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i V_j^T)^2 \tag{2.7}$$

,where U_i and V_j are row vectors with d values, I_{ij} is the indicator function that is equal to 1 if the user i rated item j or equal to 0 otherwise.

Another popular method in recommender systems is the Probabilistic Matrix Factorization(PMF) [32]. The distribution over the observed ratings is defined as given in Equation 2.8.

$$p(R|U, V, \sigma_R^2) = \prod_{i=1}^{m} \prod_{j=1}^{n} [N(R_{ij}|U_i V_j^T, \sigma_R^2)]^{I_{ij}} \tag{2.8}$$

where $N(x|\mu, \sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2. In both Equations 2.7 and 2.8 I_{ij} denotes the indication function that is equal to 1 if the user i rated the item j or equal to 0 otherwise.

When modeling the specified rating matrix, both SVD and PMF as described assume that the feature vector U and V follow the Gaussian distribution. In our case, this assumption is not appropriate when working on clickstream data. PFM as adjusted and described for our case is given in Appendix A.

2.2.8 Existing Related Work on Location-based Recommender Systems

Related work found in the literature is mostly based on the location using GPS technology for outdoor spaces rather than technologies, such as BLE or Wi-Fi for indoor spaces. It has been proved that the location of users/items has been special importance to suggest recommendations [33]. Location-based recommender systems take into consideration the spatial attributes of users to make recommendations. The location can be associated to position of the user when she rates, buys or clicks an item. The location-based ratings can be classified into three categories according to [34] as described:

- Spatial ratings for non-spatial items.
- Non-spatial ratings for spatial items.
- Spatial ratings for spatial items.

Levandoski et al. [5] presented a location-aware probabilistic generative model that exploits location-based ratings to model user profiles to produce recommendations. The difference between the proposed model and ours is that they do not take time into consideration whereas in our case we examine the performance of our model into different temporal states.

Ye et al. [35] focus on POI recommendations on the LBSNs data. Their work takes into account the geographical influence assuming a power-law distribution between the check-in probability and the distance along the whole check-in history. Our appoach takes into account users' mobility indoors by placing them into multicenters in order to recommend information items in the nearby area within a radius. In addition, the proposed collaborative method computes all pairwise distances of the whole visiting history which makes it hard to solve large-scale datasets.

Hongzhi et al. [36] propose the LCARS spatial item recommender system which its goal is to recommend to a target user a set of items (e.g. restaurants and shopping malls) taking into consideration the personal interest and local preference. Their difference to our approach is that they did not consider time as an important factor to recommend a set of items to a target user and set centers within a radius.

Salakhutdinov et al. [32] proposed a probabilistic graphic model by assuming Gaussian observation noises on observed user-item ratings. The proposed model achieved promising prediction results. Low-rank MF methods are very efficient in training since they assume that in the user-item ratings matrix, only a small number of factors influence preferences, and that a user's preference vector is determined by how each factor applies to that user. Low-rank matrix approximations which minimize the sum-squared errors can be solved using Singular Value Decomposition (SVD) [37]. The drawback of these methods compared to ours is that the observed rating data, both of them, have the underlying assumption about Gaussian distribution. This assumption is not appropriate when dealing with clickstreams data since the utility matrix consists of non-negative values. Therefore, these models encounter problems in recommending items properly.

Chapter 3

Research Methodology

In this chapter, we introduce the indoor localization methodology in Section 3.1 and the proximity-based recommender system methodology in Section 3.2.

3.1 Indoor Localization Methodology

In order to recommend information items based on user's mobility in indoor spaces, we firstly accurately track the target user.

For the purpose of this master thesis, we adapted the trilateration or N-point lateration technique examined to estimate the position of a user indoors with the help of three beacon devices as shown in the example of 3.1.

Assume that the beacons are able to transmit data within a range of a few meters depending on the device specifications. The indoor environment can mathematically be modeled as a set of circles, such that each beacon is considered as the centre of a circle and the signal range is the radius. Then, the intersection of the three circles is the estimated indoor location of the user. The coordinates of a user navigating indoors can be calculated via the construction of Equation 3.1.

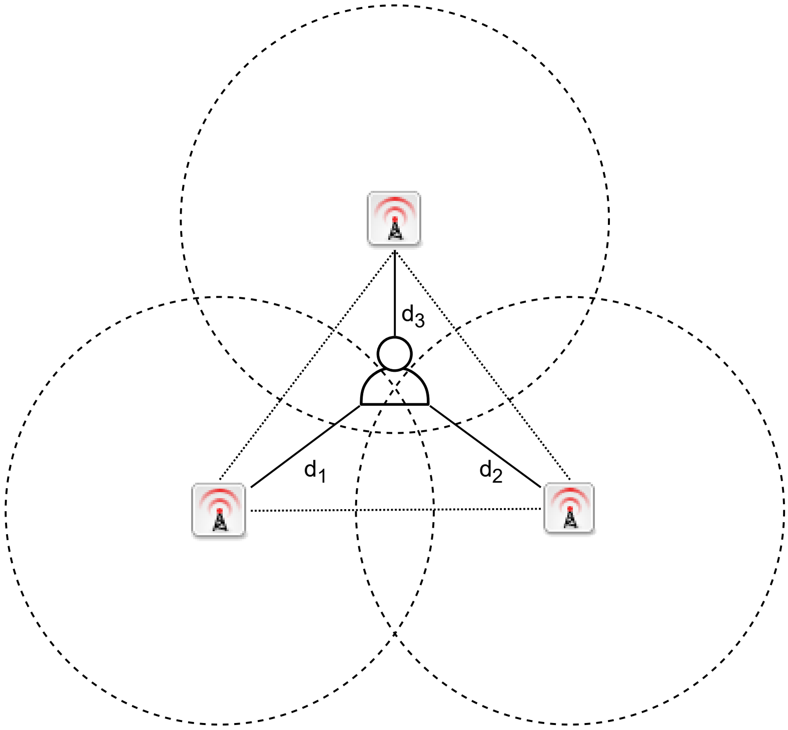


Figure 3.1: Example of trilateration method.

$$
\begin{aligned}
(x - x_0) + (y - y_0) &= d_1^2 \\
(x - x_1) + (y - y_1) &= d_2^2 \\
(x - x_2) + (y - y_2) &= d_3^2
\end{aligned}
\tag{3.1}
$$

We can expand the squares in each one as given in Equation 3.2:

$$
\begin{aligned}
x^2 - 2x_1x + x_1^2 + y^2 - 2y_1y + y_1^2 &= d_1^2 \\
x^2 - 2x_2x + x_2^2 + y^2 - 2y_2y + y_2^2 &= d_2^2 \\
x^2 - 2x_3x + x_3^2 + y^2 - 2y_3y + y_3^2 &= d_3^2
\end{aligned}
\tag{3.2}
$$

If we subtract the second equation from the first and likewise the third one from the second, we get Equations 3.3, 3.4.

$$(-2x_1 + 2x_2)x + (-2y_1 + 2y_2)y = d_1^2 - d_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 \tag{3.3}$$

$$(-2x_2 + 2x_3)x + (-2y_2 + 2y_3)y = d_2^2 - d_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 \tag{3.4}$$

The solution to the two Equations are given below:

$$x = \frac{(d_1^2 - d_2^2 - x_1^2 + x_1^2 - y_1^2 + y_2^2)(-2y_2 + 2y_3) - (d_2^2 - d_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2)(-2y_1 + 2y_2)}{(-2y_2 + 2y_3)(-2x_1 + 2x_2) - (-2y_1 + 2y_2)(-2x_2 + 2x_3)} \tag{3.5}$$

$$y = \frac{(d_1^2 - d_2^2 - x_1^2 + x_1^2 - y_1^2 + y_2^2)(-2x_2 + 2x_3) - (-2x_1 + 2x_2)(d_2^2 - d_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2)}{(-2y_1 + 2y_2)(-2x_2 + 2x_3) - (-2x_1 + 2x_2)(-2y_2 + 2y_3)} \tag{3.6}$$

The distances d_1, d_2, d_3 between the beacon and the user are calculated using the Equation 3.7.

$$d_i = 10^{(A-RSSI)/10n} \tag{3.7}$$

where $d_i, i = 1, 2, 3$ is the undetermined distance, A indicates the signal strength which is received from the beacon at a test distance d_0, the RSSI value is the signal value received from the beacon and n indicates the path loss exponent (which varies from 2 in free space to 4 in indoor environments). In our case, we set $d_0 = 1$ meter and the A value received at this distance equals to -56 dBm.

We implemented the trilateration method by building an Android Application that runs on Android version of 8 and above (>26 API Level). The beacons are configured under the iBeacon protocol setting the UUID equal to 0x7777772e6b6b6d 636e2e636f6d000001 and the Major Number equals to 0x0001 which indicates the floor number of the building. The Minor Number equals to 0x0001, 0x0002 and 0x0003 indicating the three beacons. Beacons have been placed in the corridor of the department from the South to the North Side.

Location is the most important factor while the user reaches to an area that finds interesting. After that, information items appear on her screen as a recommendation list. In addition, while the user is walking along the corridor, she is probably not interested in any information items.

The beacons were placed on the walls at a distance of 9 meters facing each other to cover the area of the corridor.

In this experiment, we have placed three beacons at a corridor and Figure 3.2 shows the directionality of a user while navigating indoors. The user is walking down the corridor to reach to a destination where there are infromation items associated with the particular location. As the user starts navigating indoors, the Left Sided Beacon which is nearer to the user transmits the lowest RSSI values equal to -60 dBm

Figure 3.2: Example of Directionality of a user.

meaning that the user is close to 1 meter compared to the rest of the three. When the user approaches the centre of the building, the RSSI values of the central beacon receive the lowest of the RSSI values and the rest receive the values approximately -80 dBm meaning that the user is heading from the one side of the building to the other. The user keeps navigating indoors because the Right Sided Beacon and the Central Beacon receive the values of -62 dBm and -82 dBm correspondingly. The time needed in order to go through the half of the corridor inside the building is 30 seconds. The average speed equals to 1,2m/sec.

The movement of the user whether she is on one side of the wall or on the other is found from the accurate tracking of the user using all three beacons.

3.2 Proximity-based Recommender System Methodology

Our methodology focuses on the fact, that while the user is navigating indoors, our proximity-based recommender system is deployed to provide personalized recommendations based on her current location, information items associated with the location, time and her profile.

Considering the fact that the items v are associated with locations l_v, our research focuses on examining the performance of our model on different values of the number of users, locations, information items through time.

Our proximity-based recommender system is modeled considering four factors: the users' preference, the information items associated with each location, the location and the time the user clicked the information items in the training data.

In order to capture the geographical influence on user's preference on locations and offer more accurate top-k information items we deploy a fused Probabilistic Factor Model framework. We deploy the particular framework since matrix factorization techniques only model users' preference and do not explore users' geographical influence received from users' preference. The particular model captures the geographical influence of the users by restricting the most frequent locations within a radius and considering these locations as a set. The most frequent locations are called centers.

FINDING THE CENTERS In order to find the centers, we model users' behavior in a spatial-temporal manner, a temporal multi-center clustering algorithm among each user's locations is based on the Pareto principle [38]. For each user u and a temporal state $t \in \mathcal{T}$ all l_v of a user u are identified and the most preferrable locations l_v associated with items v are found, whose distance is less than d meters from the selected location into a radius. In case the ratio of the user's total location within a radius to the user's total location number is greater than a threshold θ, then we set these locations as a center. Algorithm 3.1 shows the process of discovering multiple centres within a temporal state t. In our case, we set θ to 0.02 and the distance d to 2 meters. The frequency control parameter α is set to 0.2.

MULTI-CENTER GAUSSIAN MODEL (MGM)

An important aspect of recommending information items is that they are usually located around several locations.

These two aspects show that geographical influence is a strong indicator on users' presense behavior by clicking an information item at a specific l_v. We use the Gaussian distributions to model the locations where users are interested in receiving information about items and click on information items. The probability of a user u, clicking an information item v associated with the location l_v at a temporal state t given the multi-center set $C_{u,t}$ is defined by the Equation 3.8:

$$P(u, l_v|C_{u,t}) = \sum_{C_{u,t}=1}^{|C_{u,t}|} \frac{1}{dist(l_v, C_{u,t})} \frac{fc_{u,t}^{a}}{\sum_{i \in\ C_{u,t}} f_i^a} \frac{\mathcal{N}(u, l_v|\mu_{C_{u,t}}, \Sigma_{C_{u,t}})}{\sum_{i \in C_{u,t}} \mathcal{N}(u, l_v|\mu_{C_i}, \Sigma_{C_i})} \tag{3.8}$$

where, l_v denotes the location corresponding to information an item v, $C_{u,t}$ is the set of centers for the user u at a temporal state t.

For each center, Equation 3.8 consists of three terms:

1. Quantity $\frac{1}{dist(l_v, C_{u,t})}$ determines the distance between the l_v belonging to the center $C_{u,t}$ which is inversely proportional to the distance between the l_v and the center $C_{u,t}$.
2. Term $\frac{f^a_{C_{u,t}}}{\sum_{i\in\ C_{u,t}} f^a_i}$ denotes the normalized effect of the location l_v frequency $f_{C_{u,t}}$ on the center $C_{u,t}$. The parameter $\alpha \in$ (0,1] keeps the frequency levels stable meaning that very high l_v frequency is not important. In order to keep the frequency levels stable, an appropriate value for the α paramater is set 0.2.s
3. The third term denotes the normalized probability of a l_v belonging to the center $C_{u,t}$, where $\mathcal{N}(u, l_v|\mu_{C_i}, \Sigma_{C_i})$ is the probability density function of the Gaussian distribution, where $\mu_{C_{u,t}}$ and $\Sigma_{C_{u,t}}$ correspond to the mean and the covariance matrix of the centers $C_{u,t}$ accordingly.
4. μ_{C_i} denotes the mean vector of the center $C_{u,t}$.
5. $\Sigma_{C_{u,t}}$ denotes the covariance matrix of the center $C_{u,t}$.

MATRIX FACTORIZATION

MF is one of the most popular model-based collaborative filtering method for recommender systems. In the case of information item recommendations such ours, even if a user has enough data, she often appears to new transitions based on time. These issues make the traditional solutions, such as SVD, ineffective because in the case of SVD the sum-squared distance is computed only for the observed entries of the target matrix $\mathcal{R}$. In our case, the rows of our utility matrix R_t represent the users, while the columns represent the information items clicked while the user is navigating indoors. The entries of the matrix R_t represent the time the information item has been clicked. The user-item clicking matrix is divided into t sub-matrices where t corresponding to the different temporal states belonging to the set T. So, we model MF to each R_t to compute user's preference on information item v at time t.

So, given the partial observed entries in a $|U_t| \times |V_t|$ clicking matrix R_t, the goal is to find two low-rank matrices $U_t \in R^{K\times|U_t|}$ and $V_t \in R^{K\times|V_t|}$, K $\ll U_t$, V_t such that $R_t \propto U_t^T V_t$. The product of the two matrices $U_{u,t}^T V_{v,t}$ captures the correlation between user u and preference v which is the predicted probability of u's preference on information item v at time t. The predicted probability of a user u, click an information item v at time t, is determined by Equation 3.9:

$$P(R_{\mathrm{u,v,t}}) \propto U_{u,t}^{T} V_{v,t} \tag{3.9}$$

In order to accurately approximate the probabilities that users would follow certain location preferences, the objective function of MF is to minimize the quantity as defined in Equation 3.10:

$$\Omega = \sum_{i=1}^{|U_t|} \sum_{k=1}^{K} ((\alpha_{\mathrm{k}} - 1) ln(U_{\mathrm{ik}}/\beta_{\mathrm{k}}) - V_{\mathrm{ik}}/\beta_{\mathrm{k}}) + \sum_{j=1}^{|V_t|} \sum_{k=1}^{K} ((\alpha_{\mathrm{k}} - 1) ln(U_{\mathrm{jk}}) - V_{\mathrm{jk}}/\beta_{\mathrm{k}}) + \sum_{i=1}^{|U_t|} \sum_{j=1}^{|V_t|} (R_{\mathrm{ij}} ln(U^T V)_{\mathrm{ij}} - (U^T V)_{\mathrm{ij}}) + c \tag{3.10}$$

where α = $(\alpha_1, \alpha_2, ..., \alpha_K) > 0$, β = $(\beta_1, \beta_2, ..., \beta_K) > 0$ are parameters for Gamma distributions and c is a constant term.

FUSED MATRIX FACTORIZATION WITH MULTI-CENTER GAUSSIAN MODEL Simple MF methods do not perform very well when users' preferences on location is to be predicted. They do not explore the geographical influence. Users tend to find items that interest them around their location centers. So, we utilize the fusion MF with multi gaussian model [39]. We fuse users' preference on an information item and the probability a user will visit a place to determine whether a user u prefers an information item v associated to a location l_v at a temporal state t is given by the Equation 3.11.

$$P_{u,v,l_v,t} = P(R_{u,v,t}) P(l_v | C_{u,t}) \tag{3.11}$$

So, once the FMFMGM is built and the probability predictions are obtained, we rank the recommending information items in descending order based on the predicted probabilities. We choose the top-k most likely items and subsequently deploy them for personalized recommendations.

In order to improve overall accuracy of the recommending information items, we filter our data in the data set based on time and space.

Algorithm 3.1 Multi-center Discovering Algorithm.

for user $i \in U$ in $t \in T$ **do**
    Rank all locations l in |L| according to frequency of the clicks received based on the information item at a specific temporal state t
    $\forall\ l_k \in L$, set $l_k.center = -1$
    $CenterList = \emptyset$,
    $CenterTotalFrequency = 0$
    for $i = 1 \in L$ **do**
        if $l_i.center == -1$ **then**
            $centerCounter++$
            $Center = \emptyset$
            $Center.totalFrequency = 0$
            $Center.add(l_i)$
            $Center.totalFrequency+ = l_i.frequency$
            for $j = i + 1$, $j \in L$ **do**
                if $l_j.center == -1$ & $distance(l_i, l_j) \leq d$ **then**
                    $l_j.center = centerCounter$
                    $Center.add(l_j)$
                    $Center.totalFrequency+ = l_j.frequency$
                end if
            end for
            if $\frac{Center.totalFrequency}{|L_{ik}|} \geq \theta$ **then**
                $CenterList.add(Center)$
            end if
        end if
    end for
    return $CenterList\ for\ user\ u$
end for

Chapter 4

Evaluation

In this Chapter, Section 4.1 describes the experimental scenario of our approach and Section 4.2 analyzes the results.

4.1 Experimental Scenario

Though the hardware of the computing platform has no influence on the metrics taking into account for the evaluation of our recommender system, it plays an important role in the execution time. We run our Python scripts on a laptop with 8GB DDR4 memory and AMD Ryzen 3 2200U running at 3.4GHz. It is a dual-core processor which is able to run a maximum of 4 threads per core. The operating system is Ubuntu 19.10 64-bit.

As discussed in Section 2.2.6, the metrics of precision, recall and F_1-score have been taken into consideration. The metrics are used to evaluate the proximity-based recommender system only. For all the testing cases, the temporal 90% of the information items clicked by users are used for training the recommender system. We then use the remaining 10% of the temporal clicks for testing.

Due to the lack of large-scale real data for our problem, we used synthetically generated data in our evaluation. The data generator creates synthetic random walks in a building plan which is assumed to be a 100×100 grid. Locations which hold information items are randomly generated on the edges (i.e. lines) of the grid. These lines model the corridors of the building and the locations model places of interest, such as offices. Users click different information items on different times. Also, we group users' behavior based on space and time. In our data, we discarded the users who have not clicked on information items on the test set. We evaluated our recommender system by comparing its recommendations to users who clicked one or more items in the test set with the clicked information.

We tested the system under different scenarios where the number of users. locations, information items and temporal states vary. We examine the performance of our model compared to two baseline algorithms, the Multi-Gaussian Probabilistic Factor Model and the Probabilistic Factor Model. The difference between the two algorithms and our model is that the first algorithm does not take time into account which is an important factor since different information items appear at different times. The second algorithm is the simplest form of our model where time and the geographical influence are not taken into consideration.

A series of test cases with different variations are examined to compare the performance of our approach in the proximity-based recommender system by varying one of the four parameters (number of users, number of times, number of locations and number of items per location) and keeping the other three fixed to their default values. Table 4.1 shows the range of tested values for each parameter. Their default values are shown in bold. More specifically, we analyze the performance of four eperimental cases as described below.

Case 1: Examining the performance between the proximity based recommender system and the two baseline algorithms on different number of users.

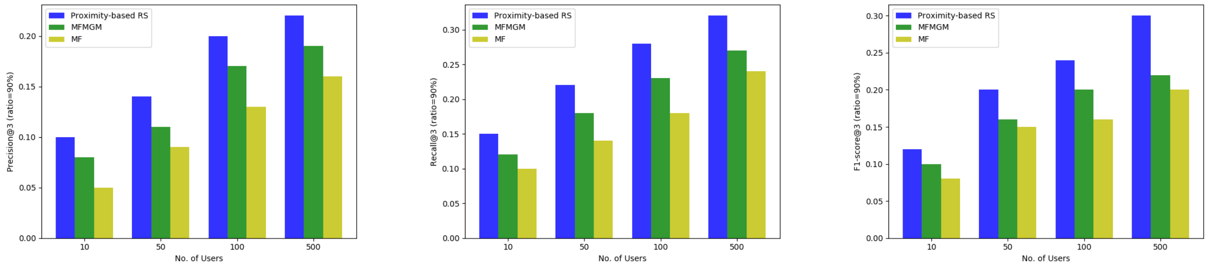
Case 2: Examining the performance between the proximity of our model on different number of locations while we keep the number of users, information items and time stable.

Case 3: Examining the performance of our model on different number of information items while we keep the number of users, locations time stable.

Case 4: Examining the performance of our model on different temporal states while we keep the number of users, locations and information items stable.

Table 4.1: Different resolutions on number of users, locations, information items and temporal states

No. of Users	**10** , 50 , 100 , 500
No. of Locations	**10** , 50 , 100 , 500
No. of Information Items	**10** , 50, 100 , 500
No. of Temporal States	**2** , 4 , 7 , 24



(a) (b) (c)

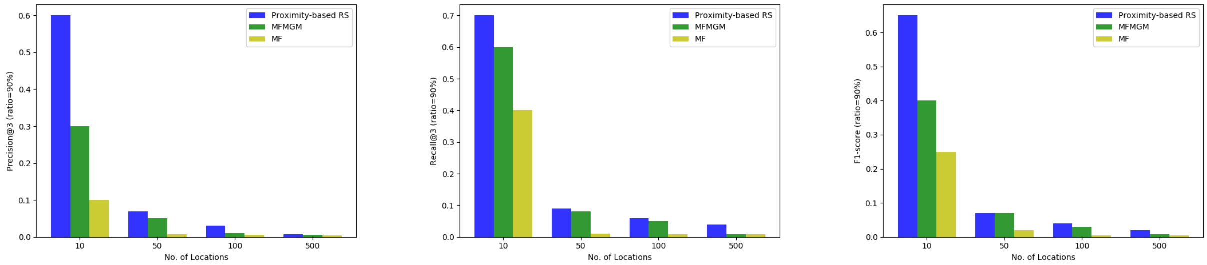
Figure 4.1: Case 1: Comparison between Proximity-based RS and baseline algorithms on different no. of users.

4.2 Experimental Results

Case 1: In this case we present the experimental results of the proximity-based recommender system and the two baseline algorithms while we keep the number of locations, information items and time slots stable. As shown in Figure 4.1 the proximity based recommender system outperforms both of the baseline algorithms comparing all three metrics examined. The more users are generated, the more accurate the recommender system becomes because it is more likely to find similar users to the target user.

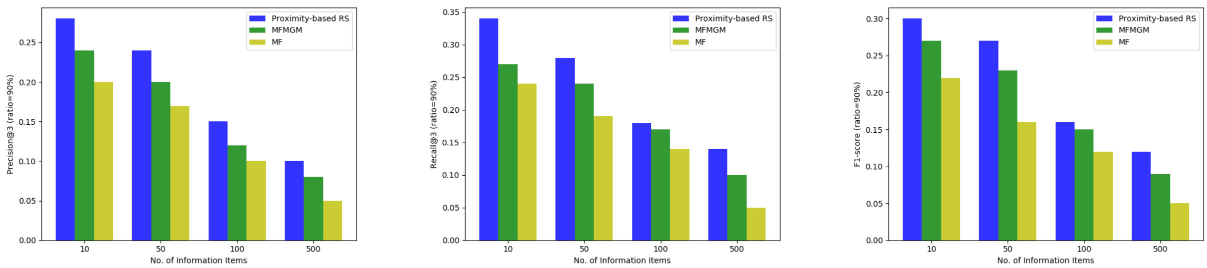
Case 2: In this case we present the experimental results while we keep the number of users, information items and time slots stable. As shown in Figure 4.2 again the proximity-based recommender system outperfoms both of the baseline algorithms since it captures the geographical influence of the users and considers the time. The difference between the number of locations considering the three metrics is because when the number of locations increases, it is more likely for the recommender system to suggest less relevant information items based on the location.

Case 3: In this case we present the experimental results while we keep the number



(a) (b) (c)

Figure 4.2: Case 2: Comparison between Proximity-based RS and baseline algorithms on different no. of locations.

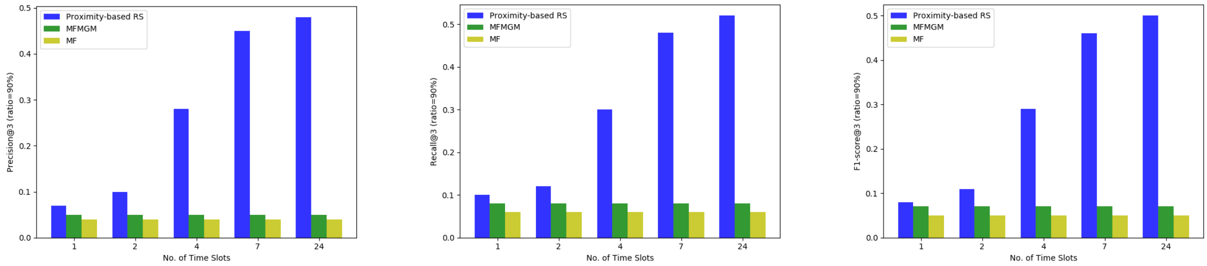


(a) (b) (c)

Figure 4.3: Case 3: Comparison between Proximity-based RS and baseline algorithms on different no. of information items.

of users, locations and time slots stable. As shown in Figure 4.3, the proximity-based RS outperforms the two baseline algorithms. The values of the three metrics decrease because the number of information items increases. This makes it harder for the recommender system to provide relevant information items because more items are associated to the location which may not be of interest to the user.

Case 4: In this case, we present the experimental results while we keep the number of users, location and information items stable as shown in Figure 4.4. We examine the performance of our model under five different time resolutions. When the number of time slots equals to one means that time is not taken into consideration. Therefore, the proximity-based recommender system provides better recommendations based on the three metrics considered. When the time slots equal to 2, 4, 7 and 24 the two baseline algorithms are examined in case there is one temporal state. The performance of the two baseline algorithms does not change because the baseline models do not take time into consideration. Nonetheless, the proximity-based recommender system recommends more revelant information items when the time slot is more specific than



(a) (b) (c)

Figure 4.4: Case 4: Comparison between Proximity-based RS and baseline algorithms on different time slots.

there is none.

Experimental results have shown that the recommender system becomes more accurate once the number of users increases because it finds similar users to the target user. In addition, when the number of locations within an indoor space increases, it is more likely for the recommender system to suggest less relevant information due to high space density. The recommender system again suggests less relevant information to the target user once the number of information items associated to each location increases which may not be of interest to the user. It is proven that time is an important factor in order to provide more relevant information to the user. The more specific the time is, the better recommendations the system provides. The proximity-based recommender system takes into account the space and time and it is proven that it provides better recommendations compared to the rest two baseline algorithms which do not take space and time into consideration.

Chapter 5

Conclusions

5.1 Conclusions

In this thesis, we have proposed a proximity-based recommender system which provides personalized recommended information to the user while navigating in indoor spaces using beacons to estimate user's position. We compare the performance of our model compared to two baseline algorithms which do not take time and space into account. The performance of the three models has conducted on synthetically generated data, varying the number of users, locations, information items and time.

Results have shown that the proximity-based recommender system outperforms the rest of the two baseline algorithms in all of the testing cases considered. That is because it takes the user's past actions while navigating indoors and time into consideration. It is shown that time plays significant role to recommend relevant information to the target user.

5.2 Future Work

There are several directions worthy of considering for future study: 1) capture the direction of the user in indoor spaces and model it into the proximity-based recommender system to provide even more accurate recommendations, 2) expand the MF technique to combine explicit and implicit feedback.

Bibliography

[1] N. Polatidis and C. Georgiadis, "Recommender systems: The importance of personalization in e-business environments," *International Journal of E-Entrepreneurship and Innovation*, vol. 4, pp. 32–46, October 2013.

[2] R. Burke, A. Felfernig, and M. Göker, "Recommender systems: An overview," *Ai Magazine*, vol. 32, pp. 13–18, September 2011.

[3] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System. Theory and practice.*, February 2001.

[4] H. Wang, M. Terrovitis, and N. Mamoulis, "Location recommendation in location-based social networks using user check-in data," November 2013, pp. 374–383.

[5] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, "Lars: A location-aware recommender system," in *2012 IEEE 28th International Conference on Data Engineering*, 2012, pp. 450–461.

[6] Y. Qian, Z. Lu, N. Mamoulis, and D. Cheung, "P-lag: Location-aware group recommendation for passive users," July 2017, pp. 242–259.

[7] K. Pahlavan, X. Li, and J.-P. Mäkelä, "Indoor geolocation science and technology," *Communications Magazine, IEEE*, vol. 40, pp. 112 – 118, March 2002.

[8] Z. Lin, "Indoor location-based recommender system," Master's thesis, University of Toronto, 2013.

[9] M. Elkhodr, S. Shahrestani, and H. Cheung, "Emerging wireless technologies in the internet of things : A comparative study," *International Journal of Wireless and Mobile Networks*, vol. 8, pp. 67–82, 11 2016.

[10] F. Zafari, A. Gkelias, and K. Leung, "A survey of indoor localization systems and technologies," 2019.

[11] Y. Yun, J. Lee, D. An, S. Kim, and Y. Kim, "Performance comparison of indoor positioning schemes exploiting wi-fi aps and ble beacons," in *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*, 2018, pp. 124–127.

[12] F. Zafari, I. Papapanagiotou, and K. Christidis, "Micro-location for internet of things equipped smart buildings," *IEEE Internet of Things Journal*, vol. 3, January 2015.

[13] S. Subedi and J.-Y. Pyun, "Practical fingerprinting localization for indoor positioning system by using beacons," *Journal of Sensors*, vol. 2017, pp. 1–16, 12 2017.

[14] N. Klepeis, W. Nelson, W. Ott, and J. Robinson, "The national human activity pattern survey (nhaps): A resource for assessing exposure to environmental pollutants," January 2001.

[15] K. Nuaimi and H. Kamel, "A survey of indoor positioning systems and algorithms," 05 2011, pp. 185 – 190.

[16] N. A. B. Nawale, Swapnesh Patel, "Beacon for proximity target marketing," *International Journal of Engineering and Computer Science*, vol. 5, no. 5, Dec. 2017. [Online]. Available: http://103.53.42.157/index.php/ijecs/article/view/1125

[17] P. Ng, J. She, and S. Park, "Notify-and-interact: A beacon-smartphone interaction for user engagement in galleries," July 2017, pp. 1069–1074.

[18] M. Uttarwar, A. Kumar, and P. Chong, "Bealib: A beacon enabled smart library system," *Wireless Sensor Network*, vol. 09, pp. 302–310, January 2017.

[19] A. Diallo, Z. Lu, and X. Zhao, "Wireless indoor localization using passive rfid tags," *Procedia Computer Science*, vol. 155, pp. 210–217, 2019, the 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919309457

[20] H. Xu, Y. Ding, P. Li, R. Wang, and Y. Li, "An rfid indoor positioning algorithm based on bayesian probability and k-nearest neighbor," *Sensors*, vol. 17, p. 1806, August 2017.

[21] C. Chen, Y. Chen, H. Lai, Y. Han, and K. J. R. Liu, "High accuracy indoor localization: A wifi-based approach," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6245–6249.

[22] U. Shardanand and P. Maes, "Social information filtering: Algorithms for automating "word of mouth"." ACM Press, 1995, pp. 210–217.

[23] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews." ACM Press, 1994, pp. 175–186.

[24] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazon's highly available key-value store," in *Proceedings of 21st ACM SIGOPS Symposium on Operating Systems Principles (SOSP)*, 2007, pp. 205–220.

[25] G. Research, "Movielens," 2021. [Online]. Available: https://movielens.org/

[26] C. Aggarwal C., *Recommender Systems*, 2nd ed. Springer International Publishing Switzerland, 2016.

[27] S. Qaiser and R. Ali, "Text mining: Use of tf-idf to examine the relevance of words to documents," *International Journal of Computer Applications*, vol. 181, July 2018.

[28] B. Schafer, B. J, D. Frankowski, Dan, Herlocker, Jon, Shilad, and S. Sen, "Collaborative filtering recommender systems," January 2007.

[29] C. Aggarwal, "Data mining: The textbook." Springer International Publishing, 2015.

[30] A. A. Konaté, H. Pan, H. Ma, X. Cao, Y. Yevenyo Ziggah, M. Oloo, and N. Khan, "Application of dimensionality reduction technique to improve geophysical log data classification performance in crystalline rocks," *Journal of Petroleum Science and Engineering*, vol. 133, pp. 633–645, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0920410515300425

[31] C. C. Aggarwal and S. Parthasarathy, "Mining massively incomplete data sets by conceptual reconstruction," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. New York, NY, USA: Association for Computing Machinery, 2001, pp. 227–232. [Online]. Available: https://doi.org/10.1145/502512.502543

[32] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, ser. NIPS'07. Red Hook, NY, USA: Curran Associates Inc., 2007, pp. 1257–1264.

[33] T. Horozov, N. Narasimhan, and V. Vasudevan, "Using location for personalized poi recommendations in mobile environments," in *International Symposium on Applications and the Internet (SAINT'06)*, 2006, pp. 6–129.

[34] F. Rehman, O. Khalid, and S. A. Madani, "A comparative study of location-based recommendation systems," *The Knowledge Engineering Review*, vol. 32, pp. 1–30, 2017.

[35] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 325–334. [Online]. Available: https://doi.org/10.1145/2009916.2009962

[36] H. Yin, B. Cui, Y. Sun, Z. Hu, and L. Chen, "Lcars: A spatial item recommender system," *ACM Trans. Inf. Syst.*, vol. 32, no. 3, Jul. 2014. [Online]. Available: https://doi.org/10.1145/2629461

[37] M. Kurucz, A. Benczúr, and K. Csalogány, "Methods for large scale svd with missing values," *ACM KDDCup 2007*, January 2007.

[38] T. Nisonger, "The "80/20 rule" and core journals," *Serials Librarian - SERIALS LIBR*, vol. 55, pp. 62–84, July 2008.

[39] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, ser. AAAI'12. AAAI Press, 2012, pp. 17–23.

[40] B. Liu, H. Xiong, S. Papadimitriou, Y. Fu, and Z. Yao, "A general geographical probabilistic factor model for point of interest recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1167–1179, 2015.

[41] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, ser. NIPS'00. Cambridge, MA, USA: MIT Press, 2000, pp. 535–541.

[42] A. Inc., "ibeacon," 2020. [Online]. Available: https://developer.apple.com/ibeacon/

[43] G. Inc., "Eddystone," 2020. [Online]. Available: https://developers.google.com/beacons/eddystone

Appendix A

Probabilistic Factor Model

Probabilistic Factor Model

Let R be an $m \times n$ data matrix whose element $r_{i,j}$ is the observed click of an item j by user i. Y is a matrix of expected clicks with the same dimensions as R, and $y_{i,j}$ denotes an element in matrix Y. Every observed element $r_{i,j}$ in matrix R is assumed to follow the Poisson distribution with the mean $y_{i,j}$ in matrix Y, respectively. The matrix Y is factorized into two matrices U and V, where U is an $m \times d$, V is an $n \times d$ matrix and d is the dimensionality of the latent factors. Each element in u_{ik} (k = 1, ..., d) in U encodes the preference of the user i to the latent item k, and each v_{jk} can be interpreted as the affinity of the location l to the latent item k. Finally, u_{ik} and v_{jk} are given by the Gamma distributions as the empirical priors.

There are two reasons we use Gamma distributions to model u_{ik} and v_{jk} instead of Gaussian or other distributions: (1) Gamma distribution is suitable for modeling non-negative values, while Gaussian distribution can model both negative and non-negative values. If we allow negative values in u_{ik} and v_{jk}, potentially, the model will generate negative preference values, which is truly unreasonable in the real world problems. (2) The Gamma distribution is already proved to be effective in modeling items over user-clicks [39, 40], where the user-item relation is also represented as a preference matrix in our case.

Therefore, the generative process of an observed user-item preference r_{ij} in our model follows:

- Generate $u_{ik} \sim$ Gamma(α_k, β_k), $\forall$ k.

- Generate $v_{jk} \sim$ Gamma(α_k, β_k), $\forall$ k.
- Generate y_{ij} occurrences of item j from user i with outcome y_{ij} = $\sum_{k=1}^{d}$ u_{ik} v_{kj}.
- Generate $r_{ij} \sim$ Poisson(y_{ij}).

The Gamma distributions of U and V follow the probabilistic functions are given in EquationA.1,A.2:

$$p(U|\alpha, \beta) = \prod_{i=1}^{m} \prod_{k=1}^{d} \frac{u_{ik}^{\alpha_k - 1} exp(-u_{ik}/\beta_k)}{\beta_k^{\alpha_k} \Gamma(\alpha_k)} \tag{A.1}$$

$$p(V|\alpha, \beta) = \prod_{i=1}^{n} \prod_{k=1}^{d} \frac{v_{jk}^{\alpha_k - 1} exp(-v_{jk}/\beta_k)}{\beta_k^{\alpha_k} \Gamma(\alpha_k)} \tag{A.2}$$

where α = (α_1, ..., α_d), β = (β_1, ..., β_d), u_{ik} $\geq$ 0 , v_{jk} $\geq$ 0, α_k $\geq$ 0 and β_k $\geq$ 0, $\Gamma(\cdot)$ function.

The Poisson distribution of R given Y can then be defined as shown in Equation A.3.

$$p(R|Y) = \prod_{i=1}^{m} \prod_{j=1}^{n} \frac{y_{ij}^{r_{ij}} exp(-y_{ij})}{r_{ij}!} \tag{A.3}$$

where y_{ij} = $\sum_{k=1}^{d}$ u_{ik} v_{jk}.

Since Y = UV^T, the posterior distribution of U and V given R can be modeled as given in Equation A.4.

$$p(U, V|R, \alpha, \beta) \propto p(R|Y)p(U|\alpha, \beta)p(V|\alpha, \beta) \tag{A.4}$$

Hence, we infer the log of the posterior distribution p(U, V | R, α, β) over the user u and item v latent factors as given in Equation A.5.

$$\Omega(U, V; R) = \sum_{i=1}^{m} \sum_{j=1}^{n} (r_{ij} ln(y_{ij}) - (y_{ij})) + \sum_{i=1}^{m} \sum_{k=1}^{d} ((\alpha_{\mathrm{k}} - 1) ln(u_{ik}/\beta_k) - u_{ik}/\beta_k) + \sum_{j=1}^{n} \sum_{k=1}^{d} ((\alpha_k - 1) ln(v_{jk}) - v_{jk}/\beta_{jk}) + c \tag{A.5}$$

Taking the derivatives on L with respect to $u_i k$ and $v_j k$, we have the Equations A.6, A.7.

$$\frac{\partial \Omega}{\partial u_{ik}} = \sum_{j=1}^{n} (r_{ij} v_{jk}/y_{ij} - v_{jk}) + (\alpha_k - 1)/u_{ik} - 1/\beta_k \tag{A.6}$$

$$\frac{\partial \Omega}{\partial v_{jk}} = \sum_{i=1}^{m} (r_{ij} u_{ik}/y_{ij} - u_{ik}) + (\alpha_k - 1)/v_{jk} - 1/\beta_k \tag{A.7}$$

We set the learning rates as denoted in Equations A.8, A.9.

$$\frac{u_{ik}}{\sum_{j=1}^{n} v_{jk} + \frac{1}{\beta_k}} \tag{A.8}$$

$$\frac{v_{jk}}{\sum_{i=1}^{n} u_{ik} + \frac{1}{\beta_k}} \tag{A.9}$$

respectively, we obtain the multiplicative updating rules [41] given in Equations A.10.

$$u_{ik} < -u_{ik} \frac{\sum_{j=1}^{n}(r_{ij}v_{jk}/y_{ij}) + (\alpha_k - 1)/u_{ik}}{\sum_{j=1}^{n} v_{jk} + 1/\beta_k} \tag{A.10}$$

$$v_{jk} < -v_{jk} \frac{\sum_{i=1}^{m}(r_{ij}u_{ik}/y_{ij}) + (\alpha_k - 1)/v_{jk}}{\sum_{1=1}^{m} u_{ik} + 1/\beta_k} \tag{A.11}$$

The matrices U and L are learnt using multiplicative update rules and not the additive ones, such as gradient descent, because the convergence is faster and are easy to implement. Other methods such as conjugate gradient have faster convergence at least at finding the local minima, but are more complicated to implement than gradient descent. Also, the convergence of gradient based mathod has a drawback of being very sensitive to the choice of step size, which is not convenient for large applications such ours. The multiplicative update rules as described in Equations A.10, A.11 are a good tactic between speed and ease of implementation for solving the problem given in Equation A.5.

Impact of dimensionality and parameters α_k and β_k Choosing an appropriate dimensionality plays a significant role in the performance of our model. On the one hand, larger dimensions give us more flexibility to represent both user and items latent vectors. On the other hand, choosing large dimensionality, we experience severe overfitting problems. Since the clickstream of users is fixed, if we use a larger dimensionality, we need to employ some smaller values of u_ik and v_jk. In the case where the d = 10, the optimal parameter settings are α_k = 20 and β_k = 0.2. In case, where d = 20, we set α_k = 20 and β_k = 0.05. The best parameter settings in our case is to set d = 10 and α_k = 10 and β_k = 0.2 for our model to perform the best.

Appendix B

BLE protocols

Our approach in indoor localization technique is based on the iBeacon protocol [42] powered by Apple in 2013. This method as shown in Figure B.1 utilizes the Received Signal Strength Indicator (RSSI) of the BLE signal. Beacon interval is usually configured by the owner since it depends on the indoor environment and the purpose of its usage in order to achieve the best optimal accuracy in the user's position. Small beacon transmission intervals means better accuracy, but it consumes more power comsuption.

The iBeacon prefix consists of the following data: 0x0201061AFF004C0215. The 0x020106 belong to the Adv. Flags, the 0x1AFF belong to the Adv. Header, the 0x4C00 belong to the Company ID, the 0x02 belongs to the iBeacon Type and the 0x15 belongs to the iBeacon Length. The aforementioned hex data have the following meaning:

- 0x02 denotes that the rest of the Adv. Flags consists of two bytes.
- 0x01 denotes that the current device supports the BLE Peripheral Role meaning that it acts like a BLE advertiser.
- 0x06 denotes the following advertising packet is BLE discoverable, non-connectable and undirected.
- 0x1A denotes that the remainder of the packet consists of 26 bytes.
- 0xFF denotes the Manufacturer Specific Data which identifies the beacon as iBeacon.

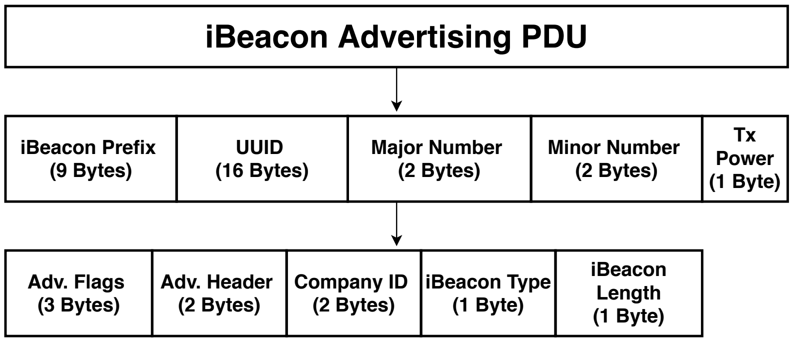


Figure B.1: iBeacon Advertising PDU.

- 0x4C00 is Apple's Bluetooth Sig ID and is the part of this spec that the protocol belongs to Apple.
- 0x02 is a secondary ID that denotes a proximity beacon, which is used by all iBeacons.
- 0x15 denotes that the remaining length is 21 Bytes.

The rest of the hex data, after the iBeacon prefix, are described as follows:

- UUID consists of 16 bytes/128 bit and is typically unique to an organization, set by the beacon owner.
- The Major Number which can be used to define a sub-region within the larger region by the UUID. A value of 0x0000 means the Major Value has not been set.
- The Minor Number which can be used to further subdivide the region defined by the Major field. A value of 0x0000 means that the Minor value has not been set.
- The TxPower denotes the calibrated value of the RSSI at 1 meter.

For completeness purposes we describe the Eddystone protocol that can be used. The Eddystone protocol developed by Google [43] in 2015 is shown in Figure B.2.

The Eddystone prefix consits of the following hex data: 0x0201060303AAFE. The hex data: 0x020106 correspond to the Advertisement Flags.

The hex data: 0x0303AAFE act as Beacon identifiers and have the following meaning:

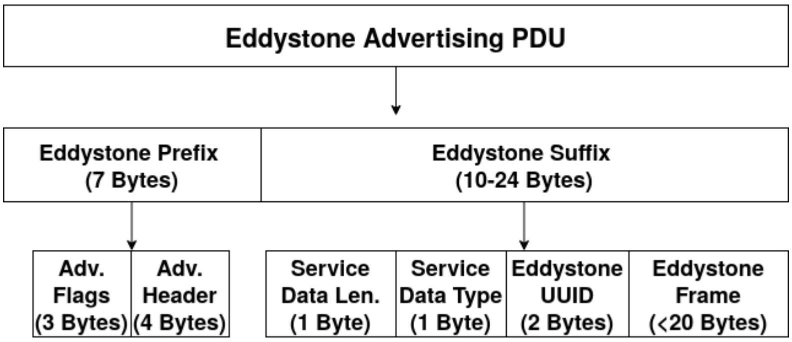


Figure B.2: Eddystone Advertising PDU.

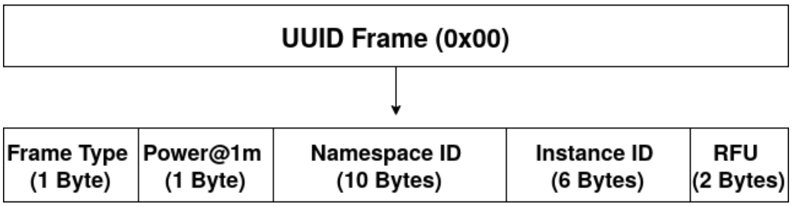


Figure B.3: UUID Frame Type.

- 0x03 denotes that the length of the following Service Data has a length of three bytes.
- 0x03 denotes a data type value from the complete list of 16-bit UUID identifier which is allocated by the Bluetooth SIG for use.
- 0xAAFE denote Eddystone UUIDs.

The Eddystone suffix consists of the Service Data Length, Service Data Type, the Eddystone UUID and Eddystone Frame. The Eddystone Service Data Length does not have a specific length value because it depends on the frame type and its date. There are currently three different Eddystone frames. The Service Data Type equals 0x16 byte denoting that Service Data data type is a 16 bit UUID. The Eddystone UUID consits of the 0xAAFE bytes as aforementioned.

As aforementioed the Eddystone frames are the UUID frame which consists od 20 bytes, the URL frame which its length varies between 6 and 20 bytes and the TLM frame which contains 14 bytes. The Eddystone UUID frame is given in Figure B.3

The Eddystone UUID frame type is a 16-byte unique beacon ID broken into a 10-byte namespace identifier and a 6-byte instance identifier, both assigned by the beacon owner. The frame length is fixed and uses the entire advertising packet, so

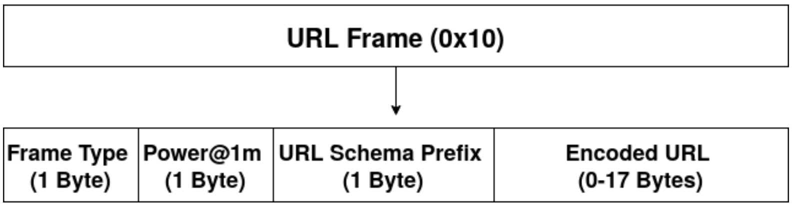


Figure B.4: URL Frame Type.

in this case the Service Data length is 0x17. The Eddystone UUID is described as follows:

- Frame type in UUID frame equals to 0x00 which is 1 byte.
- Power@1m indicates the beacon's calibrated Tx power at 0 meter. This can be calculated by measuring Tx power at 1 meter and adding 41 dBm which is 1 byte.
- Namespace ID is used to to group a particular set of beacon which are 10 bytes.
- Instance ID is used to identify individual devices in the group of beacons which are 6 bytes.
- RFU is a 2 byte field and is reserved for future use.

The Eddystone URL frame type advertises a URL using a compressed format resources accessible with hyper-text transef protocol (HTTP or HTTPS). The Eddystone URL frame is given in Figure B.4.

The Eddystone URL frame type is described as follows:

- Frame type in URL frame equals to 0x10 which is 1 byte.
- Power@1m indicates the beacon's calibrated Tx power at 0 meter. This can be calculated by measuring Tx power at 1 meter and adding 41 dBm which is 1 byte.
- The URL Scheme Prefix byte defines the identifier scheme.
- Encoded URL consists of a sequence of characters that is used to designate Internet resource accessible using HTTP.

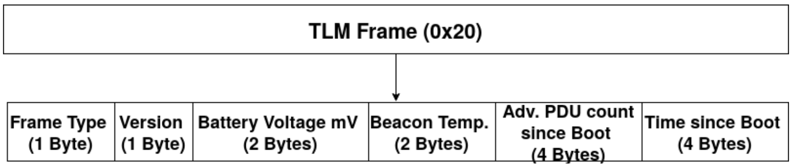


Figure B.5: TLM Frame Type.

The Eddystone TLM broadcasts telemetry information about the beacon. This frame is used to monitor the health of the beacon. The TLM frame has a fixed Length so the Service Data length byte should be always 0x11. The frame type is given in Figure B.5.

The Eddystone TLM is described as follows:

- Frame Type equals to 0x20 and shows that the frame type is a TLM one.
- The TLM version equals to 0x00 indicating the version and should always be set to 0x00. This allows future enhancements.
- Battery Voltage consists of two bytes indicating the battery volrage with a resolution of 1mV/bit. If not supported, the value should be 0x000.
- The Beacon Temperature consists of two bytes indicating the beacon temperature in Celsius degrees.
- The Advertising PDU count since Boot indicates the number of advertising events of all frame types since the last reset. This field consists of four bytes.
- Time since Boot indicates the elapsed time since the last reset in 0.1 second increments. A four byte value provides enough scope to show 13.5 years of continuous operation.

Short Biography

I hold a B.Sc. degree in Mathematics from the Department of Mathematics of the University of Ioannina based in Greece. I currently am a postgraduate in the Department of Computer Science & Engineering attending the Master's degree on "Data and Computer Systems Engineering". In the last few months I am interested in developing recommender systems.