A Thesis for the Degree of Ph.D. in Engineering

A Study on Co-Location of Mobile Users Using Ambient and iBeacon Radio Signals in Wireless Network

February 2017

Graduate School of Science and Technology
Keio University

Pedro Moreira Varela

Acknowledgments

I would like to express my deep and sincere gratitude to all people who have helped and supported me throughout this thesis. Your cooperation was the crucial importance.

I am extremely grateful to Professor Tomoaki Ohtsuki, my supervisor, for his support, his comments during the preparation of this thesis, and for accepting me in his laboratory. Under his supervision, I learned how to do research and more important I learned how to turn a simple yet theoretical idea into a practical project.

I would like to warmly thank the Japanese Government for providing me the scholarship (MEXT) to realize this thesis. Thank you so much for your financial support. It allowed me to focus on the essential, i.e., my research.

In addition, I want to say a special thanks to all the committee members, Professor Iwao Sasase, Professor Yukitoshi Sanada, Professor Masaaki Ikehara, and Professor Pooi-Yuen Kam for their constructive comments and suggestions that helped me improving the quality of this thesis.

Finally, I was honored to represent my country and at the same time to discover the grandeur and beauty of Japan. I will certainly be your first ambassador in my country, Cape Verde.

Abstract

This dissertation provides a study on co-location system of mobile users. Co-location system combines methods of detecting nearby mobile users and providing them interesting and useful services or information within their respective groups. It has found several useful and real-world applications in proximity-based services. Aware of this new trend in our society and its impact in our daily life, we design two novel frameworks with the aiming at unleashing the potential of these proximity-based services.

We first devise a scheme that exploits the similarity of the environmental radio signals from multiple Wi-Fi access points when mobile users are in the same place, a room, for instance, to cluster them into the same group. The designed scheme is based on a nonparametric Bayesian method called infinite Gaussian mixture model that allows the model parameters to change with the observed input data. In addition, we apply a modified version of Gibbs sampling techniques with an average similarity threshold to better fit user's group. We evaluate the performance, in terms of clustering accuracy, of our proposal numerically and then experimentally. Through the experimental results we demonstrate the feasibility and the efficiency of this method. Results on experiment showed that it can even achieve a better accuracy when compared with the state-of-the-art community detection-based clustering method.

Then, we extend our first scheme to a new issue arising from the need to co-localize walking groups of people. That is, we give it now the ability of clustering groups of people even though their are walking together as part of the same group. This second devised framework is based on the analysis of the two key network properties, i.e., the edge betweenness and the shortest average path length among all pairs of mobile users in the wireless networks. It leverages Bluetooth low energy technology to achieve a high degree of co-location accuracy. From the collected radio signals, we construct a graph network in which the distance between pairwise vertices represents the connection strength between mobile users. Then, we apply a modified version of the edge betweenness techniques to cluster walking groups of mobile users into the same group. We assess our method with both computer-generated and experimental data sets. Through obtained results, we have shown that our method can be successfully applied to co-localize people walking as part of the same group in wireless networks.

Contents

List of Figures

List of Tables

Acronyms

AP Access Point. 13, 35, 39, 51, 54, 65–67, 76

APL Average Path Length. 13

BLE Bluetooth Low Energy. 10, 13, 66–68, 82, 90

BS Base Station. 35, 67

CD Community Detection. 27, 61

CRP Chinese Restaurant Process. 43

D2D Device-to-Device. 7

FGMM Finite Gaussian Mixture Model. 13, 37, 41, 45

GIW Gaussian Inverse Wishart. 39, 40

GMM Gaussian Mixture Model. 18, 19

GPI Group-place Identification. 22

GPS Global Positioning System. xi, 4, 18, 23

HMM Hidden Markov Models. 20

IARC Individual Activity Recognition Chain. 24–26

IGMM Infinite Gaussian Mixture Model. 13, 33–35, 37, 41, 45, 48, 54, 61

IoT Internet of Things. 69

LBS Location-based Services. 1, 2

LTE-Advanced Long-term Evolution Advanced. 7

MAC address Media Access Control Address. 34, 53

MCMC Markov Chain Monte Carlo. 34, 37, 43

MUE Mobile User Equipment. 35, 67

NPB Nonparametric Bayesian. 33

PDF Probability Density Function. 37, 38, 40, 45

ProSe Proximity-based Services. 1

RF Radio Frequency. 27, 31, 37, 61

RSSI Received Signal Strength Indicator. 13, 34, 51, 53, 54, 67, 81

SA Simulated Annealing. 28, 61

UE User Equipments. 2, 6, 7

UUID Universally Unique Identifier. 81

List of Symbols

Q Area of interest.

N Total number of users in the network.

D Number of access points data collected.

$\mathbf{y} = \{y_i\}_{i=1}^{N}$ Set of all observations.

$y_i \in \mathbb{R}^D$ The ith observation.

$\mathbf{y}_{-i}$ All observations except the current one.

K Number of mixture weights.

$\mathbf{z}$ Indicator parameters.

$\mathbf{z}_{-i}$ All indicators except the current one.

α Concentration parameter.

π Mixture weights.

μ_j, $\vec{\mu}_j$ Means and means vectors of jth component.

s_j, $\mathbf{\Sigma}_j$ Precisions and covariance matrix of jth component.

n_j Number of observations in the jth components.

$n_{-i,j}$ Number of observations in the jth components, without taking ith observation into account.

H Hyperparameters for Gaussian inverse Wishart (GIW) distribution prior on mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$.

$\mathbf{\Lambda}_0^{-1}$ Proportional to our prior mean for $\mathbf{\Sigma}$.

υ_0 How confident we are about the above prior.

$\vec{\mu}_0$ Our prior mean for $\boldsymbol{\mu}$.

κ_0 How confident we are about in this above prior mean.

Δ, δ Similarity thresholds for IGMM and community detection algorithm, respectively.

Θ Threshold to evaluate walking/not-walking users.

G Undirected graph.

V Set of all vertices corresponding to the mobile users.

E Set of all edges representing the connection strength between pair of mobile users in the network.

M Total number of edges in the graph network.

$\sigma_{i,j}(k)$ Partial betweenness of a vertex k with respect to a pair of vertices i and j in graph G, where $k, i, j \in V$.

$C_B(k)$ Betweenness centrality of a vertex $k \in V$.

$d_G(i, j)$ Distance between vertices i and j in graph G.

Δt Minimum period of time required to the users to be together to consider them as co-localized.

Chapter 1

Introduction

1.1 Background and Motivations

Humans are social beings. Consequently, they construct conscientiously or not many complex group structures cooperating and/or competing against each other. These human groups can be exploited for many purposes as in providing them interesting and useful services or information within their respective groups. With the massive use of smartphones, these social beings provide a way to be co-localized by using only their captured ambient radio signals. Thus, allowing scientists in gaining a better understanding of their behaviors and their social interactions.

This explosive use of smart devices has also given rise to an impulsive and rapid development of a variety of mobile applications. As a result, a wide range of services is now available on users' smart devices. Example of these services are proximity-based services (ProSe), location-based services (LBS), etc.

Services such as ProSe (e.g., mobile social network [1, 2], mobile healthcare [3–5], etc.) have been around for quite a while, and new services are expected to change all user experiences in the near future. Reflecting this trend, worldwide researchers have also shown their interests in this new kind of human mobility-based, and many interesting works have been done in this area in recent years.

With this proliferation of mobile devices, a lot of efforts have been deployed aiming to explore them to their full potentials in a broad variety of contexts, such as in co-location of contexts [6, 7], in co-location of physically nearby mobile users[1] [8, 9], and in extraction of the user social relationships [10, 11], etc. New services have also been provided to the customers depending on their current location, which is known as LBS. In LBS nearby places of interest are ubiquitously queried by mobile users based on their current positions transmitted to the location server. LBS answers the questions such as where are we (in terms of latitude and longitude)? what points of interest are near us? what businesses are near us? etc.

Another interesting application of this widespread adoption of powerful smart devices is to provide useful services and information to a co-located group of people, according to their local geographical proximity. One way to proceed is to allow mobile user equipments (UEs) to sense and transmit their shared ambient radio signals to the co-location server. Upon receipt, the co-location server, based on the similarity of the reported radio signals, or on the distance between pairwise mobile users, from the same ambient radio signals, will cluster mobile users into the same group. Then, the co-location server will inform them back, through an application installed on their devices, about their belonging group.

The co-location of contexts, as it is presented in [7], aim at delivering a rich contextual information, to a co-located group of people, for developing context-aware applications in pervasive computing [12]. The information is delivered to a group of people according to their context. Here, the focus is on the context in which people are. Examples of such context are social interaction, data delivering, daily routines, etc. Whereas, the co-location of physically nearby mobile users is designed for detecting communities in which people are physically and geographically close to one another and have been together for a certain time interval. Here, on the other hand, the emphasis is on physical proximity entities. In this dissertation, even though the former is of interest, we are mainly concerned with the latter.

Therefore, the co-location system under concern in this thesis focuses on detecting and clustering of mobile users who are in the same place (a room, for instance), and/or

[1] In this thesis when we refer to a mobile user we mean that a person holding a mobile device. It can be a smartphone, a tablet, etc.

are walking together, as part of the same group, for a certain amount of time, and are physically, geographically close to one another.

In the following subsections, we will discuss the main differences between co-localization and localization systems. We also highlight some of their numerous possible application areas.

1.1.1 Co-localization versus Localization Systems

In this subsection, we want to clarify the difference between two key words that may be confusion. They are co-localization and localization. It is important to have a clear definition of these two terms in our mind before going further in this dissertation.

Localization

The term localization is defined as the process of accurately estimating the geographical position of an object, also commonly called a target or node, in wireless networks and display it on a surface of a map. The localization system focus on accurately estimating the target's position coordinates in a reference system and display it on a map, see Figure 1-1 for an example. In the localization system, when a target can estimate its own position, it is called self-positioning. On the other hand, when a central unit (e.g., cloud server) estimates the target' position through the reported information, it is called remote-positioning.

A number of applications can benefit from an accurate position estimation of a target, such as location sensitive billing, intelligent transport systems, improved traffic management, intruder detection, tacking of fire-fighters and miners, patient monitoring, and many more [13, 14]. Also, with the popularity of the wireless information access and its wide spread utilization, accurate positioning in wireless networks is highly demanded in both indoor and outdoor environments [15].

Basically, there are two different position estimation schemes: direct positioning and two-step positioning. The former is used when the location estimation is performed directly from the radio signals traveling between nodes [16]. Whereas, in the

Figure 1-1: An example of a localization scenario. This figure exhibits an example of a localization scenario where the absolute position of a target is estimated (for example, by using GPS) and displayed on a surface of a map. Therefore, the absolute coordinates of a target, in terms of latitude and longitude, with respect to a map are henceforth known.

latter, certain signal parameters are extracted first from the observed radio signals, then the location of a node is estimated based on those signal parameters. In general, the two-step positioning method is suboptimal. However, when compared with direct positioning, it shows significantly lower complexity. For this reason, it is the most common method utilized in positioning systems.

As its name suggests, the two-step positioning approach is in fact a two steps positioning method. In the first step, and depending on accuracy requirements and system constraints, position related parameters of the radio signals such as time of arrival, angle of arrival, time difference of arrival or received signal strength are estimated. In the second step, the position of a target is estimated based upon the position related parameters of the radio signals estimated in the first step. The most commonly employed position estimation techniques, in this second step, are mapping, geometric or statistical methods.

Mapping methods requires an up-to-date database, constructed in an off-line phase, consisting of previous estimated position in a given environment before the actual position estimate of a target begins. On the other hand, the geometric and statistical methods do not requires any pre-existing database. The estimation of the position of a target is performed directly from the signal parameters estimated in the previous step by utilizing geometric relationships and statistical approaches, respectively.

Co-localization

The term co-localization is defined as how near (geographically close to) or how far two or more nodes are from one another. A node represents a person or an object. Figure 1-2 shows an example network with two groups of co-located mobile UEs: **Group 1** and **Group 2**. Thus, mobile users in the same group can enjoy all the applications provided by the co-location systems.

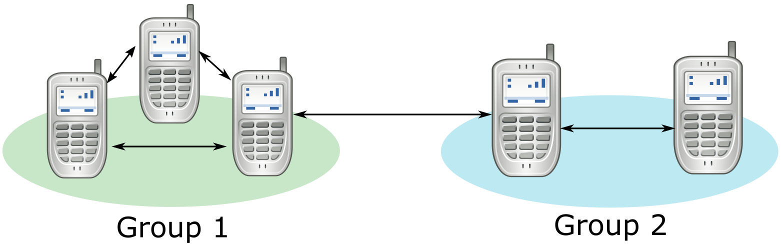


Figure 1-2: An example network with two co-located mobile user equipments (MUEs). In **Group 1**, there are three co-located MUEs. On the other hand, in **Group 2**, there are two co-located MUEs. The double-headed arrow represents the communication links between MUEs.

Co-location systems are primarily interested only in proximity objects, i.e., what is “near” to one another, and the term “near” is defined in accordance with the application requirements. There is no fixed measure of vicinity among nodes, with respect to a distance, to state whether they are co-located or not. Therefore, the term co-localization is more ambiguous that the term localization discussed earlier.

The co-location system under concern in this thesis focuses on detecting and clustering mobile users who are in the same space and/or are walking together for a certain amount of time, and are physically, geographically close to one another. It is worth noting that, contrary to the localization systems [13, 14], whose aim is at estimating the absolute or relative position of an individual user in the wireless networks and display it on a surface of a map, the co-location systems, on the other hand, seek ways of identifying vicinity users and clustering them into the same group.

Therefore, in accordance with the application requirements, one defines how closely users should be regarded as potentially co-located [17]. Note also that a mobile device is considered in proximity to another mobile device if a given proximity criterion is fulfilled. Examples of these proximity criteria are radio range, geographic range, etc.

A number of real-world applications can directly benefit from the automatic inference of co-localized groups of mobile devices. In the next subsection, we will present some of the potential areas of application of co-located groups of mobile users.

1.1.2 Applications of Co-location Systems

We are witnessing an incredible change in the way we interact with each other and with our physical world. Information collected on a co-located group of mobile user equipments (UEs) has found several useful and real-world applications. In this subsection, we present some of these application scenarios with the aim at showing how it can be applied to provide enhanced wireless services to the mobile users. This subsection serves also an introduction and the motivations of doing research on this topic.

Real-world example applications of co-localized groups of mobile users range from authentication scenarios [18, 19] (see Figure 1-3) with nearby people, in wireless networks, to prevent eavesdropping and spoofing attacks, to place recommendations (for people with common interests) and includes information about human social interactions, geosocial networking [20], opportunistic networks [21, 22] (in which the aim is at delivering data based on pairwise contact opportunities), and many more. It also shows promises in revolutionizing vehicular social networks [23–25].

Another particular interesting application of co-located groups of mobile users is at providing social network users with notification messages on their smartphone such as that their co-workers, acquaintances, friends, etc., are in close proximity with them (e.g., in the same room). This is performed by allowing the co-location server to estimate the proximity level of mobile devices and send a notification message to the mobile users that their co-workers, for example, are in the same room as they.

Furthermore, by taking advantage of physically closely co-localized mobile UEs,

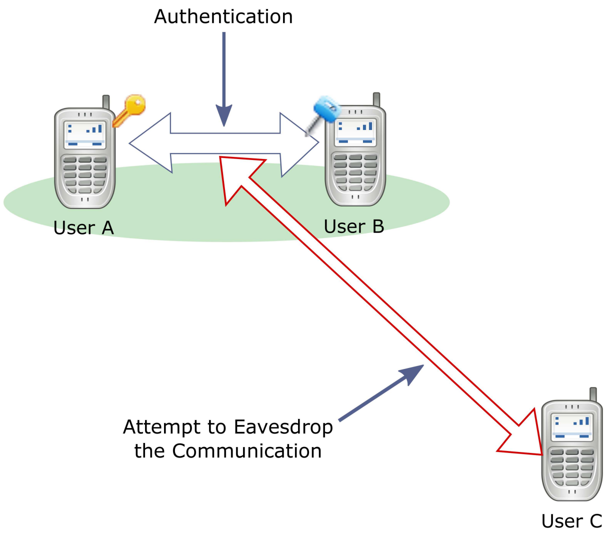


Figure 1-3: An example application scenario of two co-located mobile user devices (*User A* and *User B*) involving in an authentication process. As they are co-localized, according to some pre-established authentication requirements, if a third user, for example *User C*, wants to eavesdrop their communication with the aim at launching a spoofing attack, they can be aware of this malicious intent and take some appropriated measurements to counteract this kind of behavior. As an example where such application may be very useful is in mobile payment, where the payment services are performed via two nearby mobile devices. In this case, a very short distance around 10 centimeters is required.

one can directly route data traffic between mobile users that includes content sharing (e.g., sharing streaming video, pictures, etc.), connectivity extension, etc., which is known as device-to-device (D2D) [26] communication (see Figure 1-4), for the purposes of proximity-based services [17] in long-term evolution advanced (LTE-Advanced) system. Thus, co-located mobile UEs, in the context of D2D communication, can be exploited with the objective of minimizing the power consumption of mobile devices [27], in improving throughput, increasing network coverage, delay, spectrum efficiency, as well as enhancing quality of experience in LTE-Advanced networks [28]. It is also beneficial in spreading of information in social-aware mobile networks [29], in which the interactions among mobile users rely on both their movement as their social relationships.

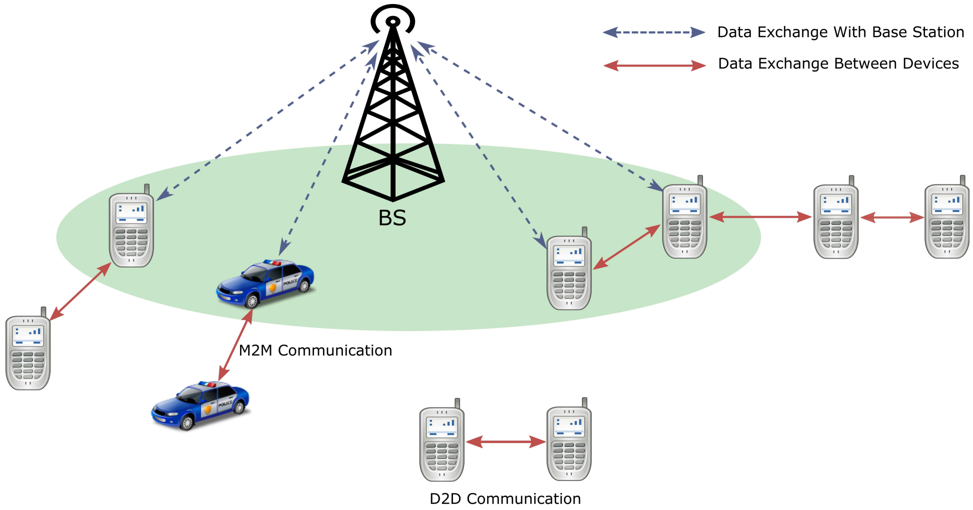

Data Exchange With Base Station
Data Exchange Between Devices
BS
M2M Communication
D2D Communication

Figure 1-4: An example network of device-to-device (D2D) communication and machine-to-machine (M2M) communication. This figure exhibits critical applications of co-located mobile devices. It shows that nearby mobile devices can directly communicate with one another without the need for the data to traverse the core network. This obviously brings some real advantages to the wireless networks. It aims at minimizing the power consumption of the mobile devices, improving throughput, increasing network coverage, etc.

An Illustrative Example

With the aim at helping visualize the concept and scope of the co-location systems, let us imagine a real-world application of a co-localized group of mobile users where one of them is watching a video on YouTube channel. If another one wishes to watch the same video, as they are in close proximity to one another, he can take streaming video directly from his neighbor, instead of downloading it directly from the YouTube server. It happens that the same line of reasoning can be adopted on the uploading case in a specific situation. That is, instead of having multiple connections on the server for the same purposes, it is better to have a reduced number of connections on the server and let the users routing their data traffic between themselves.

1.2 Contributions

This thesis starts with an introduction to the co-localization systems. A particular attention is devoted to the background and the motivations of doing research on this topic. Then, we clearly show in which aspects and to what extent the co-localization systems differ from localization systems, discuss several real-world applications, and the benefits that co-location systems bring to our ever-connected society.

In Chapter 2, we first discuss some existing methods that can be used in clustering process. Then, we review some related existing works that have been done on this topic. We describe, in each case, the approach undertaken in order to address this issue. Technical details about their implementation are also presented. The work that inspired us to do research on this topic is highlighted [30] in this chapter as well. Moreover, for each discussed approach, we emphasize its strengths and make clear its limitations.

In Chapter 3, we present our first method for clustering mobile users. It automatically discovers co-localized mobile users, when they are in the same place. To this end, we exploit the similarity of radio frequency measurements from users' mobile terminal. We do not require any further information about them.

The designed co-localization algorithm is based on a nonparametric Bayesian (NPB) method called infinite Gaussian mixture model (IGMM) that allows the model parameters to change with observed input data. IGMM possesses several attractive properties that make it an excellent choice for this kind of applications when compared with other existing techniques. One of them is actually that it can be used when the number of clusters in the input data is unknown or may vary over time. Indeed, this is always the case in the pervasive computing.

Based on the co-location criterion, we propose a modified version of Gibbs sampling technique with an average similarity threshold (which can be understood as level of the similarity of the measured radio signals) to better fit user's group. Finally, we carry out analysis and show that the proposed method is practical and can be implemented efficiently with high accuracy.

As we are interested, in this first proposal, only on mobile users that have been in the same place (e.g., in a room) for a certain amount of time, we derive a mathematical model to differentiate between walking and non-walking mobile users. The goal is to filter out the passing by mobile user who will not make part of any existing group.

For the purpose of co-location, we use ambient Wi-Fi radio signals whose detection is available in nearly every smartphone, and increasingly, hotspots can be found anywhere we go. Therefore, it can work in both indoor and outdoor environments. The proposed method is built on spatial-temporal location of the mobile users, and infers co-located mobile users using multiple ambient radio signals, which provides an unforgeable co-localization proof. In association with received signal strength indicator (RSSI), MAC address, and arrival time of beacon packets from multiple ambient radio signals, we show through simulation and experimental studies that the proposed method can efficiently detect co-located mobile users.

The discovery of the co-located mobile users is performed in real-time and in a centralized, which allows the co-location server to control the formation of the all co-localized mobile users. We analyze the performance of our proposal, in terms of clustering accuracy, not only numerically but also experimentally in order to demonstrate its feasibility. We also perform a comparison result.

With the aim at improving the framework proposed in Chapter 3, we specially design a novel method for clustering mobile users, in real-time, when they are walking together as part of the same group, in Chapter 4. However, it can also be applied when people remain in the same place as well.

The designed method is based on the edge betweenness techniques that allow the model to automatically infer the number of co-located mobile users in the input data. It exploits the period of time that mobile users have been walking together as part of a group, the frequency of their meetings, and finally the distance between pairwise mobile users for the same period of time. Furthermore, we propose a modified version of the edge betweenness algorithm with an average path length as a key enabler to a high co-location accuracy, in accordance with the application requirements.

The proposed method is designed in such a way that it allows us to exploit one of the most interesting findings in social networks analysis, i.e., most of real world groups have on average a short distance connecting people within groups [31].

We leverage the emerging and promising Bluetooth low energy (BLE) [32] technologies by collecting an array of signals broadcast by all nearby iBeacon [33] devices indexed by time. For this, we take into account the universally unique identifier (UUID), received signal strength indicator (RSSI), and the arrival time of radio signals transmitted by an iBeacon device. The iBeacon technologies are used owing to its very low cost, low power consumption, easy to deploy, and relatively long range. Furthermore, it is mainly designed for proximity-based services, contrary to the access points which implement the protocols for faster access.

We use the collected information to construct a matrix of interactions, in which each entry is a distance representing a pairwise connection strength among mobile users. Then, the groups of mobile users are inferred based on the analysis of the two key network properties, i.e., the edge betweenness and the average shortest distance among all pairs of users. Finally, we analyze our approach with both computer-generated and experimental data set to demonstrate its feasibility.

Note that both of the designed methods, in Chapters 3 and 4, do not estimate the absolute position of individual users, which prevents them from being tracked, thus protecting their location privacy. These methods require only a list of captured ambient and iBeacon radio signals to be reported to the co-location server, and do not spread the list among other users, consequently there is no privacy leakage. It is worth noting that, even though the co-location server informs users of the presence of other users in their vicinity, it does not disclose their exact location.

Chapter 5 draws a conclusion of this thesis and presents some directions for the future research on this topic.

We summarize the contributions of this thesis as follows:

- For the purpose of proximity-based services, we first propose a method able to automatically cluster mobile users, in real-time and in a centralized manner, while they are in the same place (a room, for instance). The proposed method exploits the similarity of the users' measured ambient radio signals to cluster them into the same group. We also consider that mobile users should be in that place for a certain amount of time in order to regard them as co-located. We apply IGMM and a modified version of Gibbs sampling techniques for inference of the class label of each observation.

- We design a novel method that extends the capability of the previous method by given it now the ability to cluster mobile users even though they are walking together as part of the same group. In this case, we utilize the radio signals transmitted by iBeacon devices. Thus, we exploits the connection strength between iBeacon devices to cluster users into the same group. Moreover, the period of time that mobile users spend together is also taken into account. The set of mobile users belonging to the same group is inferred by applying two key network properties, namely the edge betweenness and the average path length.

- In both cases, we first present numerical results. Then, we carry out experiments and analyze these methods with data sets from real-world settings. Thus, we demonstrate, through numerical and experimental evaluation, their robustness and effectiveness, and show that they can be successfully applied to co-localize people in wireless networks.

1.3 Outline of Dissertation

This thesis is structured into five Chapters as it is shown in Figure 1-5. In the first chapter, we introduce the background and motivations of the co-location systems. We highlight some key differences between localization and co-localization systems. Then, the related works are reviewed in Chapter 2. In Chapter 3, we provide a technical analysis on how to infer co-located groups of people, while they are in the same place, by applying a nonparametric Bayesian method called infinite Gaussian mixture model. Analysis on numerical and experimental results are conducted in order to demonstrate its effectiveness.

Then, we extend the aim of Chapter 3, in Chapter 4, by proposing a novel method able to co-localize users even though they are walking as part of the same groups. This newly devised method is based on the analysis of the two key network properties, i.e., the edge betweenness techniques and the average path length. We carry out numerical and experimental analysis and show its performance in terms of clustering accuracy. Finally, we conclude and present the direction for future research including possible improvements in Chapter 5.

We portray the relationship between chapters and techniques used in this thesis in Figure 1-6. It illustrates that radio signals are sensed from multiple Wi-Fi access points (APs) and Bluetooth low energy (BLE) devices. In fact, received signals strength indicators (RSSI) are collected and processed in both cases. Similarity of the measured radio signals from different APs are exploited for the inference of the co-located groups of people that are in the same place, in Chapter 3. Whereas the connection strength between pairwise of mobile users using BLE devices are used to co-locate walking groups of people in Chapter 4.

Radio signals from APs are extracted and modelled with finite Gaussian mixture model (FGMM) when the number of clusters in the data set is known. However, when the number of clusters is unknown or may vary over time, infinity Gaussian mixture model (IGMM), which is an extreme case of FGMM, becomes a better choice. In this work, the latter is utilized.

Both FGMM and IGMM use Gaussian distribution to model the observations. Gibbs sampling method is utilized to infer the class label of each observation. However, to effectively cluster mobile users, in accordance with the application requirements, we compute the average similarity value, which represents the centroid of each discovered cluster, and accept a new membership, into this particular cluster, if the distance of this new incoming observation to the center of that cluster is less than or equal to a predefined similarity threshold. In this case, different distance metrics can be utilized. The similarity threshold defines our co-location criterion, i.e., how close mobile users should be considered as potential co-located.

From the collected BLE radio signals, we construct a graph network in which each vertex corresponds to a mobile user and the distance between pairwise vertices represents the connection strength between mobile users. Then, the edge betweenness, which is a generalization of the vertex betweenness, is used to classify walking groups of mobile users. Based on average path length (APL) of each discovered cluster, with a similarity threshold, set of vertices that belong to the same cluster are extracted from the constructed graph network. These set of vertices represent co-localized walking groups of people. Here again, the similarity threshold defines our co-location criterion.

In Chapter 5, we draw a general conclusion of this thesis and highlight several other challenging issues that need to be addressed in order to fulfill the potential of co-location systems.

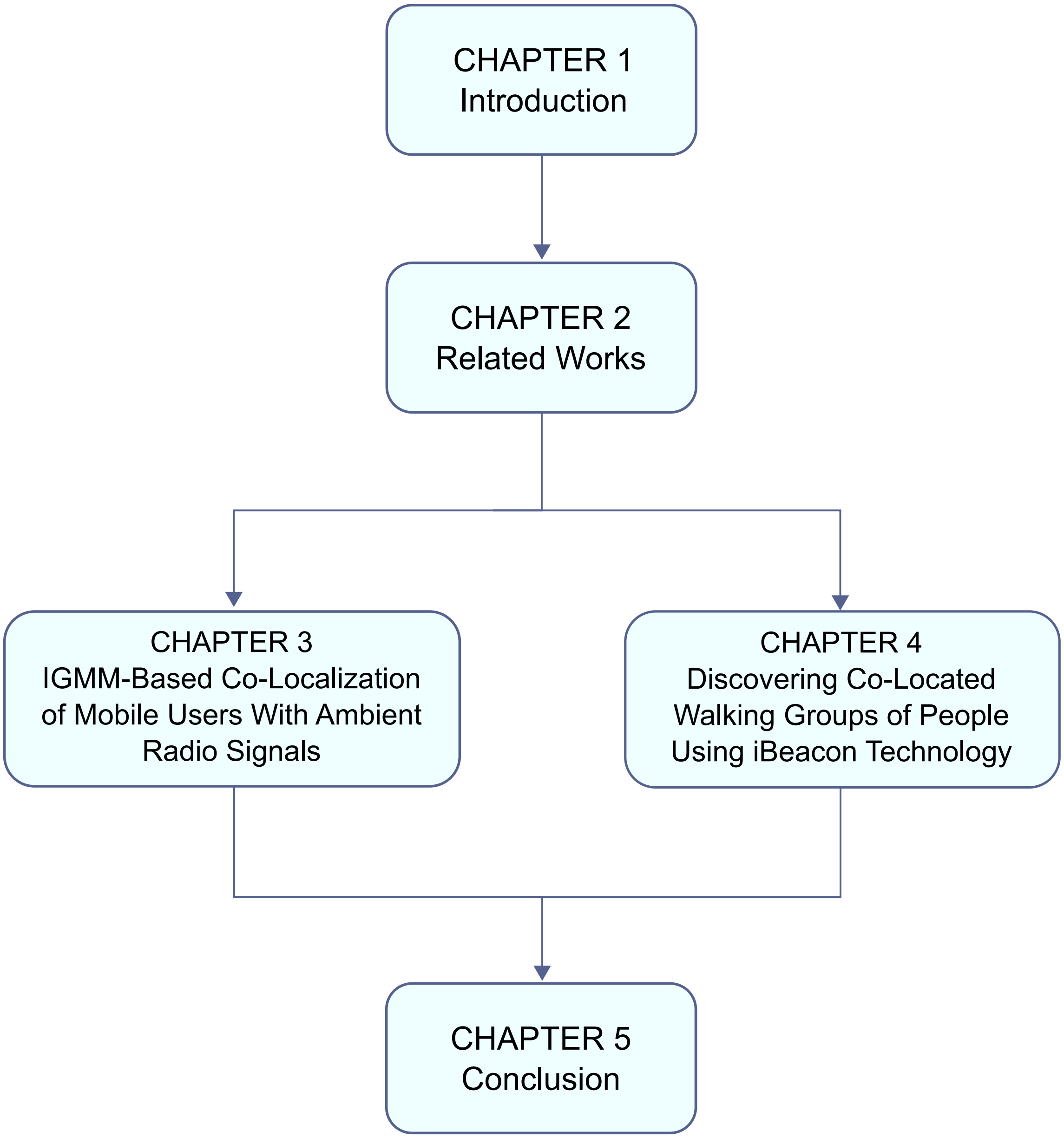

CHAPTER 1
Introduction
CHAPTER 2
Related Works
CHAPTER 3
IGMM-Based Co-Localization of Mobile Users With Ambient Radio Signals
CHAPTER 4
Discovering Co-Located Walking Groups of People Using iBeacon Technology
CHAPTER 5
Conclusion

Figure 1-5: Outline of this dissertation.

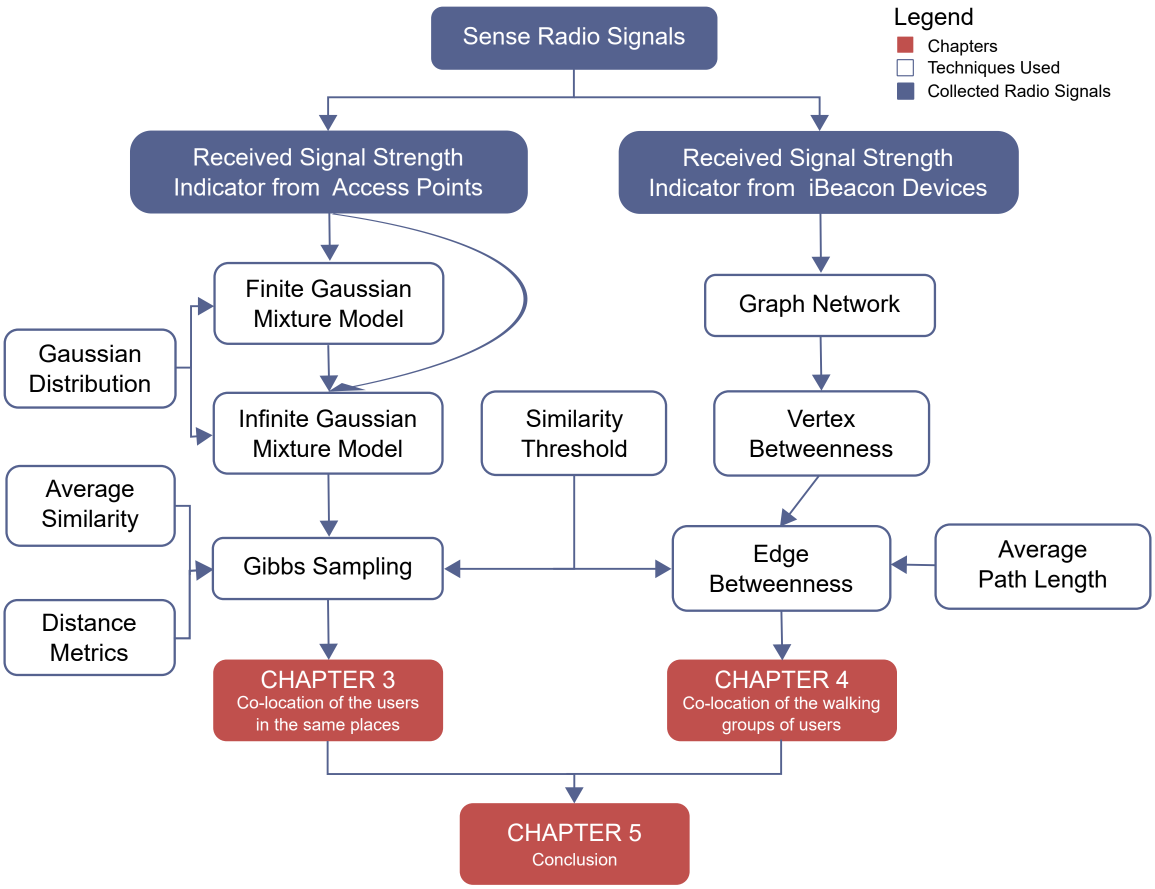

Sense Radio Signals
Legend
🟥 Chapters
□ Techniques Used
🟦 Collected Radio Signals
Received Signal Strength Indicator from Access Points
Received Signal Strength Indicator from iBeacon Devices
Finite Gaussian Mixture Model
Graph Network
Gaussian Distribution
Infinite Gaussian Mixture Model
Similarity Threshold
Vertex Betweenness
Average Similarity
Gibbs Sampling
Edge Betweenness
Average Path Length
Distance Metrics
CHAPTER 3
Co-location of the users in the same places
CHAPTER 4
Co-location of the walking groups of users
CHAPTER 5
Conclusion

Figure 1-6: Relationship between techniques and chapters.

Chapter 2

Related Works

2.1 Introduction

The co-localization system has been subject to several research studies in recent years, due to its importance on people-centric and place-centric mobile applications [34]. However, it is indeed a recent research topic and many works still remain to be done, as we will see later on.

In this chapter, we first discuss some traditional clustering methods able to tackle this issue. Then, we provide a review on some interesting works already done in co-location system and explain different techniques utilized. In both cases (traditional and conventional methods), we highlighted their strengths as well as their limitations.

Co-location system faces several key design challenges that should be careful addressed in order to fulfill its potential. Following are some of them:

- The designed algorithm should be able to automatically discover co-located groups of mobile users, with high accuracy, in the wireless networks, without the need to be specified how many clusters to find. Indeed, in the real-world scenario, we do not have any knowledge of the number of active mobile users in the network. Moreover, it is unpredictable and changes over time.

- As users are becoming more and more concerned with their privacy, the designed algorithm should not track or allow a third-party application to trace them.

Therefore, the designed schemes should be inherently users' privacy preserving.

- Instead of estimating the position of a mobile user, as localization engines do, co-localization techniques seek to detect physically and geographically close mobile users who have been together for some amount of time, and cluster them into the same groups. Therefore, two key co-location parameters should be taken into account, i.e., the duration and/or the frequency of the group meeting.

In the following sections, we will present some existing clustering methods and related works and explain how they deal with the aforementioned design challenges.

2.2 Traditional Clustering Methods

An easy way of thinking to address the co-location issue is to use an already built-in positioning system equipped with each smartphone to estimate the current position of the users. Then, using the current obtained position to state whether or not they are co-located [35]. Despite the fact that this approach seems simple and attractive at first, it presents several drawbacks associated with positioning systems to co-localize mobile users. One of them is actually that the position of a target is not accurately assessed and changes place to place (in indoor environment, it is even not available when using GPS) [36]. Another drawback is that collecting people's position for a long period of time can allow them to be easily tracked with today's technologies (e.g., by using data mining). Therefore, robust techniques to infer groups of co-localized mobile users are needed, without disclosing their absolute position.

Traditional clustering approaches such as K-means [37], Gaussian mixture modeling (GMM) [38], or hidden Markov model [39] provide also a way to solve this problem. However, all of them suffer from the same drawbacks. In fact, these algorithms require a fixed number of clusters[1], which they need to be told to find. As the number of users in pervasive computing can change over time, and consequently the number of hidden clusters in the input data set is unknown and may also vary,

[1] Throughout this thesis, the words cluster and group are used interchangeably.

these algorithms become inappropriate for this kind of problems. In addition, in real-world settings we do not have any knowledge of the input data, and the model chosen depends heavily on the data sets.

In the following subsections, we present some of these algorithms and provide their mathematical foundation. We skip the discussion of GMM in this chapter because our derived framework is based on it, which we thoroughly discuss in Chapter 3.

2.2.1 *K*-means Algorithm

Parametric clustering methods such as K-means has been thoroughly used in the literature since its establishment in 1967 by MacQueen *et al.* [40]. Its widely adoption is due to the fact that its procedure is easily programmed and computationally economical [40].

The main objective of this algorithm is to partition a given data set into K subsets, where K is the number of clusters in the dataset. K is also a parameter to be specified. For example, given a set of N observations, $\mathbf{y} = \{y_1, y_2, \ldots, y_N\}$, in an D-dimensional feature space, K-means algorithm partitions each observation y_i into a cluster c_j, where $j \leq K$. This partitioning is performed by minimizing the sum of the squared distances to the cluster centers, i.e., to minimize the following objective function

$$\sum_{j=1}^{K} \sum_{y_i \in c_j} \parallel y_i - \mu_j \parallel^2 \tag{2.1}$$

where μ_j is the centroid of the cluster c_j. The algorithm operates in the following steps to classify a given set of the observations [40]

1. Put K observations into the space represented by the observations that are being clustered. These observations represent initial cluster centroids.

2. Assign each observation to the cluster that has the closest centroid.

3. When all observations have been assigned, recalculate the positions of the K centroids.

4. Repeat steps 2 and 3 until the centroids no longer move.

The algorithm is expected to converge at a certain number of iterations, i.e., when no more assignment changes are happening with each iteration. However, it does not necessarily find the most optimal partition, i.e., it can get stuck in local minima. Moreover, it is also significantly sensitive to the initial selected cluster assignments. There are techniques for choosing initial assignments effectively and keeping the algorithm from converging in local minima. The Bradley-Fayyad algorithm [41] is one of these techniques for choosing refined initial assignments.

2.2.2 Hidden Markov Models

Hidden Markov models (HMM) are another class of algorithms that can be used for clustering. They are stochastic methods for modelling temporal and sequence data. The basic idea of HMM was introduced by Baum and Petrie in the late 1960s [42, 43]. Since then, it has been extensively applied to a wide variety of problems, as in automatic speech recognition [44, 45], gesture recognition [46, 47], sequence clustering [48], computer vision [49], and many more.

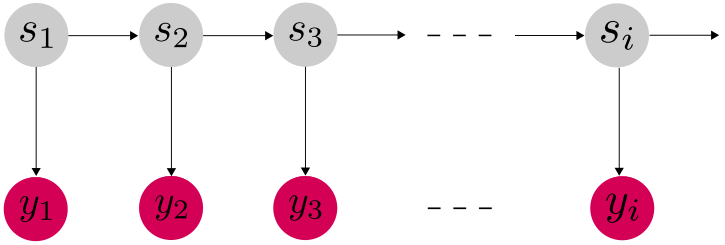


Figure 2-1: Illustration of the sequence of hidden Markov model where each observation y_i corresponds to a hidden state s_i.

The HMM can be considered as a specific instance of the state space model represented in Figure 2-1, in which the latent variables, $S = \{s_i\}_{i=1}^{K}$, are discrete. In this figure, we can notice that each observation y_i is generated by a specific hidden state s_i. In fact, under Markov assumption, the latest observation is assumed to be influenced by the current state of the system. Therefore, a discrete-time HMM is defined by a set of hidden states, $S = \{s_1, s_2, \ldots, s_K\}$, where each state is characterized by a state transition probability distribution, also known as transition matrix A. K is the number of the hidden states. The values of the transition matrix A denoted by a_{ij} represent the transition probabilities of going from one state the to another, i.e., from state s_i to state s_j. They are given by

$$a_{ij} = p(s_{t+1,j} = 1 | s_{t,i} = 1), \tag{2.2}$$

which means that the probability of being in state s_j at time $t + 1$ given that at time t we were at state s_i. As the values of a_{ij} are probabilities, they take their values from $0 \leqslant a_{ij} \leqslant 1$, with $\sum_K a_{ij} = 1$. At each time instant t, which T denoting the length of observation sequence, there is a set $V = \{v_1, v_2, \ldots, v_M\}$ of possible discrete observation symbols that can be made. The probability of observing these symbols is denoted by $B = \{b_j(l)\}$, where $b_j(l) = p(v_{t,l} | s_t = j)$ is the probability of observing a particular symbol v_l given that at time t we are at state s_j.

As the model is formulated as sequential of hidden states, from Figure 2-1 we can see that the initial state s_1 has no parent node. It has, however, a marginal distribution $p(s_1)$ given by a vector of probabilities $\pi = \{\pi_i\}$, in which $\pi_i = p(s_1 = i)$, i.e., the probability of being in state s_i at time $t = 1$.

The observation sequences made at each time instant t is denoted by O_t. The HMM is usually denoted in a compact form as a triplet $\lambda = (A, B, \pi)$.

Given a set of observed sequences $\{O_t\}$, the values of the HMM parameters can be efficiently estimated using the Baum-Welch algorithm [39] or Baldi-Chauvin algorithm [50]. Baum-Welch algorithm determines the parameters maximizing the likelihood $p(O_i | \lambda)$. It is an example of a forward-backward algorithm [39] used to compute $p(O | \lambda)$, given the model λ and a sequence O.

A standard approach to cluster sequences of observations using HMM is known as proximity-based method [51]. It computes the similarity between sequences of observations and pairwise distance matrix-based approaches to obtain clusters of sequences.

Considering a set of T observed sequences $\{O_t\}_{t=1}^{T}$, the algorithm operates as follows:

1. Train the model for each sequence;
2. Compute the distance matrix $\{D(O_i, O_j)\}$, expressing the similarity measure between sequences by using forward probability $p(O_j, \lambda_i)$;
3. Using pairwise distance matrix-based method to perform clustering.

Even though HMM is a well known and studied technique, it is unsuitable for the co-location problem. Indeed, it needs to be specified the value of K, i.e., the number of clusters to be found in the input data. As discussed earlier and thoroughly emphasized in this dissertation, in pervasive computing we do not have any knowledge of the number of clusters in the input data, and it may vary over time. Consequently, HMM becomes not the best choice for this kind of applications.

2.3 Group-place Identification Algorithm

The first work on using community mobility traces to automatically infer social groups members and group-place associations that have some importance for a group of people goes back to the work published by Gupta *et al.* [52] in 2007. The authors in [52] designed an algorithm, called group-place identification (GPI), that takes advantage of the location of users to infer their corresponding groups and associated places.

The GPI algorithm performs using community mobility traces acquired from any localization system to achieve its goal. It relies on repeatedly discovering users' copresence at the same place to determine the group members, and in turn deduce their meeting places. The basic assumption behind GPI is that group members have a much higher degree of copresence (DCP) than non-group members. The DCP is rather defined as the total number of times two members were copresent divided by the total number of group meetings.

The GPI algorithm operates by identify each user with his respective places. For each place visited by a user u_i, the algorithm verifies if there are groups associated with that place. If so, the group members are identified using copresence information. This is done by analyzing the place visit data from all other users u_k to check potential copresence with user u_i at place P. The information obtained from this analysis is then used to build a copresence matrix with respect to user u_i at place P. Two users identified at the same place are considered co-located if the distance between them is less than a threshold Δ, and the time overlap between their visits is at least Δt.

Finally, the place where the group is formed is to be identified. To this end, the average of the geographical coordinates of all trace points by all users at place P is computed, and called a point C. Then, the place is determined by looking at actual geographies area of radius E around the point C. The radius E is defined as the maximum error in determining the point C, which is introduced by the localization engine. Based on the proposed scheme, the GPI algorithm is evaluated with respect to the two following goals: *i*) high percentage of group member identification, and *ii*) high accuracy of the place of the group meetings.

From their evaluation, the authors showed that GPI algorithm is accurate and exhibits low false positives. However, it presents some issues and privacy concerns arise among them. In fact, the location of the mobile users is not accurately assessed and its accuracy changes with places; by collecting positions of the users for a long period of time exposes them to be easily tracked with today's technologies; and finally, their approach requires a location engine (e.g., GPS) [53] installed on every user's device, which constrains its usability. Moreover, the frequently computation of the users' location and delivering it to the server could significant reduce the battery lifetime of a mobile devices.

2.4 Detection of Walking Groups of Users

Later in 2011, Roggen *et al.* in [54] proposed to detect groups of walking people by analyzing the data signals collected from an ensemble of people wearing on-body sensors. During their experiments, people were wearing each one an accelerometer.

The authors in [54] formulated the co-location problem as a series of processing steps, called crowd behavior recognition chain, that can be used to infer collective crowd behavior from on-body sensors. From the collected data signals, machine learning techniques are used to infer users with similar patterns while they are walking together.

Crowd behavior is defined as coordinated movement of a large number of individuals to which a semantically relevant meaning can be attributed. Examples of these behaviors include people queuing, people clogging and forming lanes, people walking in groups, running, etc. In their work [54], the collective behavior is restricted to walking groups of people. Therefore, all discuss hereafter will be on that latter. The recognition chain, on the other hand, is defined as the task of identifying which individuals participate to that crowd behavior.

From the on-body sensor data measurements, the characteristics of each user are inferred. Then, these characteristics are analyzed pairwise for each pair of users. The aim is at finding out whether the behavior of these two users may be the outcome of their participation to the specific crowd behavior (e.g., walking together). Finally, the users that participate in the common crowd behavior are determined among all the others. This is achieved by analyzing the pairwise measure of disparity using graph visualization and graph clustering. That is, the inference of all groups of walking users.

2.4.1 Individual Activity Recognition

Following the individual activity recognition chain (IARC) [55] processing principles, the human activities are inferred from raw sensor data. The IARC is used for recognizing one or more user behaviors from the on-body sensor data measurements. Its role is to map low-level sensor data S^u (e.g., body-limp acceleration) of a users u to a meaningfully human activity (e.g., do a step). This is generically referred as the user “individual behavior” B^u. Formally,

$$IARC : S^u \rightarrow B^u \tag{2.3}$$

The individual behavior at given time t is estimated using the data available up to that time point. Thus, the behavior B^u corresponds to time series is done by

$$B^u = \{B^u_t : t \in T^u\}, \tag{2.4}$$

where B^u_t is the behavior of the user u at a given time t, and $T^u = \{T^u_1, T^u_2, \ldots\}$ is the time instants T^u_t at which the behavior B^u_t is estimated. The behavior is further represented by a tuple $B^u_t = (b^u_t, p^u_t)$, where b^u_t is the set of activities and p^u_t representing the confidence of the system in the decision.

The IARC embraces a series of processing stages described as follows: *a*) the sensor data are collected, which correspond to a time series $S = \{s_1, s_2, s_3, \ldots\}$; *b*) the time series S is pre-processed, which leads to time series $P = \{p_1, p_2, p_3, \ldots\}$. In this case, the time series P is segmented into sections within which a characteristic of the user behavior is computed. Each section i delimited by a start time t^s_i and an end time t^e_i, yielding a segmented time series $W_i = \{p_{t^s_i}, \ldots, p_{t^e_i}\}$; *d*) features are extracted from these sections to discriminate the activities. The outcome is a feature vector $X_i = \Psi(W_i)$. The feature vector X_i is then mapped into an individual behavior b_i as $X_i \rightarrow (b_i, p_i)$. b_i represents a discrete individual behavior and p_i is the classification likelihood, i.e., the confidence in the classification result.

The classification is carried out using a machine learning classifier. According to the authors, any machine learning classifier such as Support Vector Machine [56], Naïve Bayes classifiers [57], etc. can be adopted.

2.4.2 Pairwise Disparity Analysis

After inferring the behavior of each single user, the measure of disparity between a pair of users is carried out at time T from the behavior B^u and B^v. The aim is at recognizing pair of users that participate to a common crowd behavior. The computation of the disparity follows this model:

$$C_T^{u,v} = g(Corr(f(B^u, T), f(B^v, T))). \tag{2.5}$$

The function $Corr(\cdot, \cdot)$ calculates the measure of similarity between the input data. In turn, the function g maps it to a disparity value, which can be 0 for the same crowd behavior and 1 for different crowd behavior. Thus, the resulting disparity matrix at time T is computed as $C_T = [C_T^{u,v}]_{n \times n}$, for n users. Its values are lower when users participate in the same crowd behavior and higher in different crowd behavior. The pre-processing function $f(\cdot, \cdot)$ defines a slide window w_1 within which the disparity is computed. It is formulated as follows:

$$f(B^u, T) = \{B_t^u : t \in T^u, T - w_1 \le t \le T\}. \tag{2.6}$$

The functions $Corr(\cdot, \cdot)$, $g(\cdot)$, and $f(\cdot, \cdot)$ and their parameters are determined based on the training data set.

2.4.3 Global Crowd Behavior

With the computation of the disparity matrix, $C_T = [C_T^{u,v}]_{n \times n}$, in the previous subsection, the task now is to find the global crowd behavior from this disparity matrix, i.e., to find out set of users who participate in the same crowd behavior at a given time T. According to the authors [54], different methods can be adopted. However, they opted for graph clustering method to objectively identify clusters of users.

By applying graph clustering method, set of users performing the same activity are identified by the proposed scheme, i.e., people participating to the common crowd behavior, which is in fact the inference of walking groups of people.

Note that the proposed method first identifies activity of each user by applying IARC techniques. Then, try to cluster users with the same activities together. However, it should be noted that IARC does not guarantee a perfect recognition of the individual activities. The proposed techniques, in its own way, do not provide any mechanism to assess how close people are from one another. By asking people to wear a particular kind of sensors in order to determine their on-going activities reduces the practicability of the proposed scheme. Moreover, the time duration that people should pass together in order to state that they are co-located is not taken into account in their proposals.

2.5 Method Based on Community Detection Tools

More recently, in 2015, Dashti *et al.* [30] devised a real-time clustering method to co-localize mobile users based on the similarity of their radio frequency (RF) fingerprints. The authors assume the mobile users are in the same place and propose to exploit their shared ambient radio signals. From the reported RF fingerprints, community detection (CD) tools are applied to infer co-located groups of users. Mobile users are considered potentially co-localized if their reported RF fingerprints differ less than a predefined threshold. The co-location is performed by calculating the distance (in signal space) between reported fingerprints from each pairs of mobile users. In this proposal, the time traces of fingerprints are also taken into account in order to infer the length of users' interaction.

To apply CD tools for inferring groups of co-located mobile users, a connectivity graph is first constructed by taking into account the similarity of user's measured radio signals. The connectivity graph is constructed with the distance computed (in signal space) between reported fingerprints from each pairs of mobile users. If the distance between pairwise users, $d_{i,j}$, is less than a preset threshold δ, the two mobile users are connected by an edge, i.e., $C'_{i,j} = 1$ in the estimated connectivity matrix, otherwise $C'_{i,j} = 0$. The groups discovering process in the constructed graph aims at dividing the vertices (users) in such a way that within each cluster the most similar vertices are observed. To this end, an objective function called "modularity" function is defined which the aim at measuring the fraction of the edges that falls within the given groups minus the expected fraction of edges if they were distributed at random. Thus, by maximizing this modularity function, the graph is partitioned into many within-community links and few possible between-community links.

2.5.1 Modularity Function

The modularity function is defined as follows

$$M = \sum_{n=1}^{N} \left(\frac{l_n}{L} - \frac{d_n^2}{4L^2} \right) \tag{2.7}$$

where N is the number of communities, L is the number of links in the graph, l_n is the number of links between vertices in community n, and d_n is the sum of the degrees of the vertices in community n. The objective is to find a community assignment for each vertex in the graph such that the modularity function M is maximized. By maximizing this modularity function, the number of clusters within the constructed connectivity graph can be inferred automatically. In the next subsection, we discuss the technique used to maximize this modularity function.

2.5.2 Simulated Annealing Method

As aforementioned, the proposed algorithm needs to maximize a modularity function M for inferring the number of clusters in the constructed connectivity graph. In fact, this maximization is performed with a heuristic technique called simulated annealing (SA) [58]. SA is a stochastic optimization technique for finding a global low-cost configuration of an objective function that may have several local minima. It was inspired by the process of annealing in metalwork. The annealing process consists of heating and cooling a metal so that its physical properties can be altered owing to the changes in its internal structure.

SA was first proposed as an optimization technique by Kirkpatrick in 1983 [59] and Černý in 1984 [60]. It is a well known randomized search process used for finding a good solution (not necessarily the best one) to an optimization problem. It exhibits, however, an attractive property, i.e., it avoids the problem of getting stuck in local optima-solutions that are better than any other neighbors, but are not the very best.

In order to achieve a global low-cost configuration, a computational temperature T

is introduced in the algorithm. At high temperature T, the algorithm can explore high cost configurations, whereas at low temperature T, the algorithm explores low cost configurations. Normally, we start with high temperature T and then slowly "cool", decrease, it. As the temperature is slowly reduced, the system decreases gradually toward the minima solutions. In this case, the chance of accepting worse solutions is also reduced. Thus, the algorithm gradually concentrate on the region of search where hopefully a optimum solution can be found.

With the aim at identifying co-located groups of people in the wireless networks, the objective function M is maximized. Thus, the cost $C = -M$, where M is the modularity function that we want to maximize, as defined in (2.7). At each temperature T, assuming the current best cost is C_i, the algorithm randomly chooses a new neighbour solution C_j, and accepts this newly solution as the better one with the following probabilities [58]

$$p = \begin{cases} 1 & \text{if } C_j \leq C_i, \\ \exp\left(-\frac{C_j - C_i}{T}\right) & \text{if } C_j > C_i \end{cases} \tag{2.8}$$

where C_i is the current cost of the system and C_j is the cost of choosing a new solution that maybe is better than the previous one.

The authors in [30] evaluated their proposal with real-world data sets and showed that it provides accurate people co-location information with sub-meter accuracy. Moreover, the proposed scheme was also analyzed with different distance metrics (e.g., Euclidean distance, Manhattan distance, Minkowski distance, etc.), and demonstrated that these distance metrics impact differently the co-location system.

The algorithm presented in this section inspired us to do our work. In Chapter 3, we will compare our first proposal with this algorithm. Therefore, for the sake of comparison, we will call henceforth this method CDSA-based clustering method.

2.6 CrumblR Algorithm

In [61], Vanderhulst *et al.* built a framework, called CrumblR, that associates places with services. That is, users opportunistically share their locations with a place in order to obtain associated *Proxemic Services*. The authors defined the Proxemic Services as a "temporal service that automatically provides the user with value at a specific place."

CrumblR algorithm operates by first presenting to the users with an overview of places of interest (e.g., mall, hospital, airport, etc.) near their current location. By checking in to a place, the user' device begins to drop wireless signal fingerprints at that place. In return, proxemic services associated with that place are pushed to the user' device (e.g., alerts, coupons, interactive controls). Once the user has left the place, he automatically loses all these associated services.

To achieve its objectives, CrumblR implements two different algorithms able to determine mobile device' location and group. The first one is called place detection algorithm. It checks in to the previously trusted places by detecting a mobile device' coarse location. This task is accomplished by exploiting two key operations of mobile devices: cellular and Wi-Fi probing techniques. Thus, enabling a mobile device to learn about the identities of Wi-Fi APs and cell towers within radio range. The second one is called point-in-place algorithm which is based on co-location. It is used to detect mobile devices' location and cluster them into the same group. The central idea of co-location techniques here, according to the authors, is that the multipath structure of a radio channel is unique to every location and can be considered as a signature of the location. Therefore, co-localized mobile devices experience a similar multipath environment and exhibit similar multipath profiles.

From the measured RF fingerprints, the algorithm computes the distance (in signal space) between every two mobile devices. In this case, different distance metrics can be utilized as a measure of the distance between devices. The mobile devices whose RF fingerprints differ less than a predefined similarity threshold δ are regarded to be potentially co-localized.

Although such an approach seems interesting, it needs to collect RF fingerprint at a specific place beforehand, which reduces its practicability. Indeed, like the algorithm discussed in Section 2.3, it presents also a trade-off between disclosing users' location and the benefit of services it provides to them in return.

2.7 Summary of the Reviewed Clustering Methods

In the previous sections, we first reviewed some popular traditional clustering methods and showed for each one of them why they fail to be applied to the co-location systems. Then, we presented some existing conventional methods for co-location of mobile users. We also highlight some of their strengths as well as their weaknesses.

In this section, we draw a summary of all these techniques presented earlier by showing a comparison study between them. In this comparison, we are mainly concerned with some of their key properties needed in co-location system. Table 2.1 presents a summary of each one of them. A hyphen in different cells means that the information is missing.

2.8 Conclusion

In this chapter, we explain several design challenges that face co-location systems and discuss how some existing methods can be applied to address these design challenges. Existing works on co-location systems are also reviewed. We provide in each case the mainly idea behind each proposal and their mathematical foundation.

In the next chapters, we will present and evaluate our methods and show how our proposals deal with these design challenges that face co-location systems.

Table 2.1. Comparison between different clustering methods (traditional and conventional)

	Methods	Number of Clusters	Duration/ Frequency	Privacy Issue	Accuracy
Traditional	*K*-means	Not Automatic	-	No	-
	HMM	Not Automatic	-	No	-
	GMM	Not Automatic	-	No	-
Conventional	GPI	Automatic	Yes	Yes	High
	IARC-based	Automatic	Yes	No	High
	CDSA-based	Automatic	Yes	No	High
	CrumbR	Automatic	Yes	Yes	-

HMM - Hidden Markov Model;
GMM - Gaussian Mixture Model;
GPI - Group-place Identification;
IARC-based - Individual Activity Recognition Chain-based clustering;
CDSA-based - Community Detection-based Simulated Annealing

Chapter 3

IGMM-Based Co-Localization of Mobile Users With Ambient Radio Signals

3.1 Introduction

In this chapter, for the purpose of realizing potential applications of co-localized mobile users, we present a method able to detect, in real-time and in a centralized manner, co-localized mobile users in wireless networks. It is based on a nonparametric Bayesian (NPB) method called infinite Gaussian mixture modeling (IGMM) [62]. We chose IGMM because it offers several attractive proprieties, when compared with its counterpart, that make it a potential candidate for the co-location problem. These properties are summarized as follows

- It is known that Bayesian methodology avoids overfitting problem. Thus, the task of adjusting model complexity disappears;

- It avoids selecting a statistical model from a set of candidate models, given the input data;

- It avoids the need of the *a priori* knowledge of the input data, i.e., the number of active devices operating in the network;

- It can be used when the number of clusters in the input data is unknown or may vary over time. In other words, it can automatically infer the number of clusters in the data set; etc.

IGMM exploits the similarity of the users' measured ambient radio signals from different Wi-Fi hotspots to cluster them into the same group. To classify users' measured radio signals a Markov chain Monte Carlo (MCMC) [63] implementation of a hierarchical IGMM [64] is utilized. An MCMC is used because it simulates a Markov chain whose equilibrium distribution is the posterior distribution. Therefore, sampling from this posterior distribution avoids the problems of local optima solutions. Furthermore, a modified version of Gibbs sampling is proposed as a key enabler to a high co-localization accuracy, in accordance with application requirements.

As we are interested in the groups of users in the same place, we also proposed a method for inferring walking and non-walking mobile users based on a period of time Δt. This Δt is defined as the minimum period of time required by the mobile users to be together, in the same place, in order to regard them as potential co-located. As stated earlier, in this chapter, we are only interested in clustering of mobile users who spend a certain amount of time together in the same place. Therefore, we need to filter out passing by users who will not make part of any of these groups.

The proposed method, which is based on IGMM, is built on spatial-temporal location of the mobile users and infers co-located groups of mobile users using multiple ambient radio signals, which provides an unforgeable co-localization proof. In association with received signal strength indicator (RSSI), MAC address, and arrival time of beacon packets from multiple ambient radio signals, we show through simulation and experimental studies that the proposed method can efficiently detect co-located groups of users. Moreover, through a comparative analysis we have shown that the proposed method can even outperform the state-of-the-art clustering method.

Note that, contrary to the other existing techniques [35], our method does not estimate the absolute position [65] of individual users, then to cluster them into the same group, which prevents them from being tracked, thus protecting location privacy. The method requires only a list of captured ambient radio signals to be reported to the co-location server, and does not spread the list among other users, consequently there is no privacy leakage. It is worth noting that, even though the co-location server informs users of the presence of other users in their vicinity, it does not disclose their exact location.

3.2 System Model

Mobile users that have been together, for a certain amount of time, in the same place, experience the similar Wi-Fi radio signals from their shared ambient radio signals [30]. Hence, we aim at detecting these mobile users with similar RF measurements and cluster them into the same group.

In Figure 3-1, we present an example network of our co-location system. In this figure, there are several mobile user equipments (MUEs), organized in two groups: **Group 1** and **Group 2**. Mobile users in the same group are expected to experience similar radio signals from their nearest access points (APs). Periodically, they will report to the nearest base station (BS) their measured radio signals. Upon receipt, the base station will in turn transmit the reported measurements to the co-location server through the Internet. The server will perform the task of group formation detection from the received data sets, and will inform back the mobile users, through an application installed on their devices, about their belonging group.

In this thesis, we propose to exploit the Wi-Fi radio signals to cluster mobile users when they are in the same place, a room, for instance. This approach is explained by the fact of their easy deployment and no extra cost, and their ability of working in both indoor and outdoor environments. However, other radio signals can be exploited as well to co-localize mobile users [30].

In the following subsections, we discuss in detail our implementation based on IGMM. For ease of reference, we summarize the notation of all the mathematical symbols used in this chapter in the beginning of this thesis in Section “List of Symbols”.

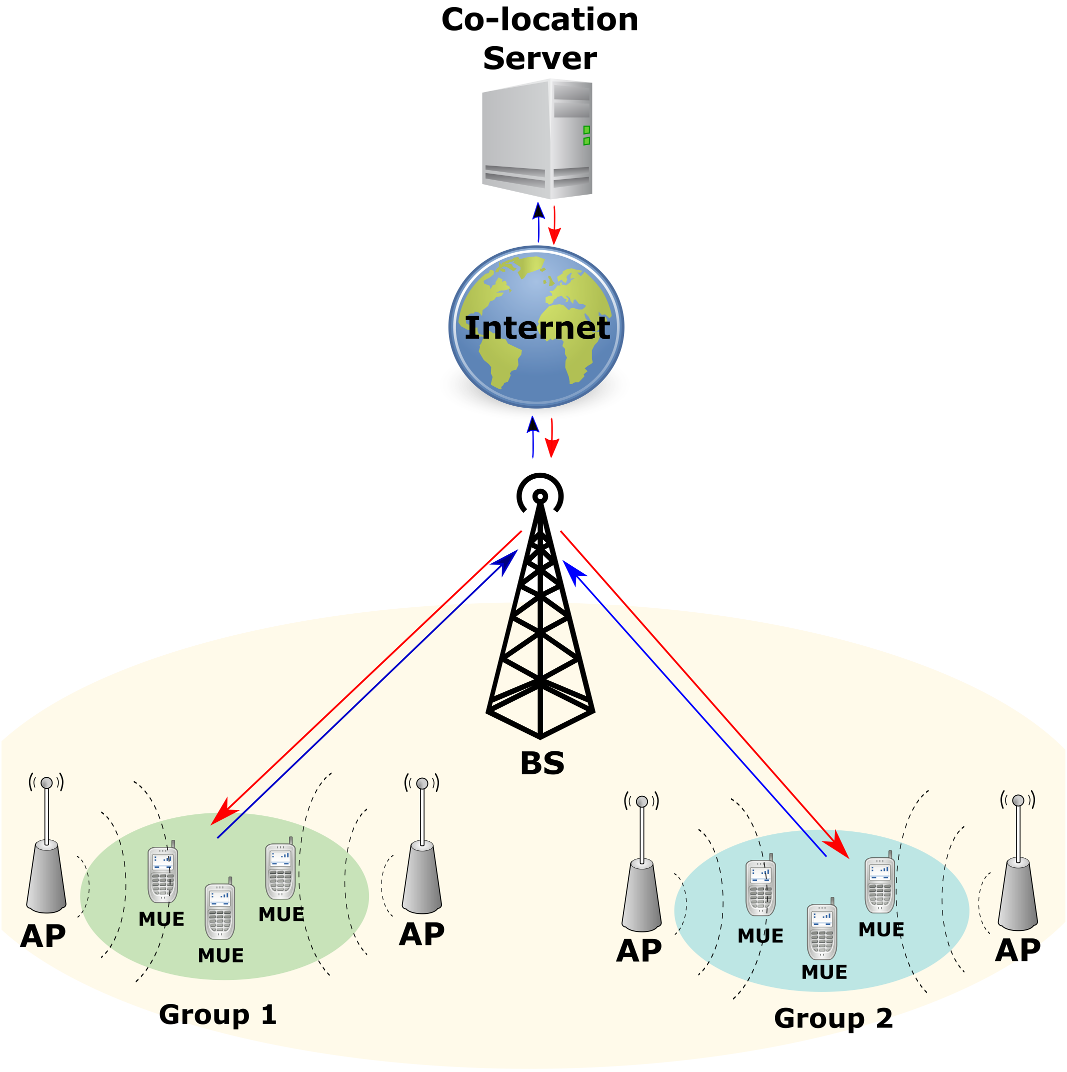


Figure 3-1: An example network architecture of two co-localized groups of mobile users equipments: **Group 1** and **Group 2**. The blue arrows indicate the transmission of the ambient radio signals to the co-location server. The red arrows represent information of co-localized mobile equipments sent by the server.

3.2.1 IGMM-Based Co-location

Consider $\mathbf{y} = \{y_1, y_2, \ldots, y_N\}$ are our set of all observations from N mobile users in the area of interest $\mathcal{Q}$, where each $y_i \in \mathbb{R}^D$ is a feature vector of ith user in a D-dimensional space. For the sake of simplicity, we will first present our model for one dimensional space ($D = 1$), and, then, explain how to generalize this model for the multivariate case later on.

Farrahi *et al.* [66] showed through 72 individuals over nine month period collecting

Bluetooth signals, that the distribution of users that have been in physical proximity fits Gaussian distribution. Based on this finding, and as we are only interested in users' physical proximity, we assume that the received RF measurements can be well modeled by a multivariate Gaussian distribution. Thus, one Gaussian mixture model will be used to model each class.

Fixed number of classes

Our co-localization technique is implemented with infinite Gaussian mixture model (IGMM) for modeling. Then, we apply an MCMC method called collapsed Gibbs sampling technique for classification. It simulates a Markov chain whose equilibrium distribution is the posterior distribution. Sampling from this posterior distribution circumvents the problems with initialization and local optima [67].

In [62], Rasmussen has shown that, even though we do not have any knowledge of our input data, we can start with a finite Gaussian mixture model (FGMM). That is, we assume that the number of classes is known, and then explore the model when the number of the classes is unknown. So, let us assume that we have K mixture weights to model our input data $\mathbf{y}=\{y_i\}_{i=1}^{N}$, which the probability density function (PDF) given in (3.1), and derive the model later when $K \to \infty$.

$$p(y_i) = \sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mu_j, s_j^{-1}\right), \tag{3.1}$$

where π_j are the mixture weights, with $0 \leq \pi_j \leq 1$, and $\sum_{j=1}^{K} \pi_j = 1$. The mixture weights represent the probability of y_i observation belongs to one of the K classes. The parameters μ_j and s_j are the means and precisions (inverse covariance) of the jth Gaussian $\mathcal{N}$, respectively.

The mixture means, μ_j, have Gaussian priors in the following form

$$p(\mu_j|\lambda, r) \sim \mathcal{N}(\lambda, r^{-1}), \tag{3.2}$$

whose mean, λ, and precision, r, are hyperparameters of the model. Their priors are given by

$$p(\lambda) \sim \mathcal{N}(\mu_y, \sigma_y^2) \tag{3.3}$$

and

$$p(r) \sim \mathcal{G}a(1, \sigma_y^{-2}), \tag{3.4}$$

which are Gaussian and Gamma, respectively. The mean, μ_y, and the variance, σ_y^2 are computed from the observations.

The mixture precisions, s_j, are given by the Gamma priors as

$$p(s_j|\beta, \omega) \sim \mathcal{G}a(\beta, \omega^{-1}), \tag{3.5}$$

whose shape, β, and mean, ω^{-1}, are also hyperparameters of the model. Their priors are given by

$$p(\beta^{-1}) \sim \mathcal{G}a(1, 1), \tag{3.6}$$

and

$$p(\omega) \sim \mathcal{G}a(1, \sigma_y^2), \tag{3.7}$$

which are inverse Gamma and Gamma, respectively.

Following [62], we use a symmetric Dirichlet distribution to compute the mixture weights $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_K)$. In fact, Dirichlet distribution is a conjugate prior[1] of the Multinomial distribution, whose joint PDF is in the following form

[1] A prior is conjugate if it yields a posterior that is the same family as the prior (a mathematical convenience) [64].

$$
\begin{aligned}
p(\boldsymbol{\pi}|\alpha) &\sim \mathrm{Dir}(\alpha/K, \alpha/K, \ldots, \alpha/K) \\
&= \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \prod_{j=1}^{K} \pi_j^{\frac{\alpha}{K}-1},
\end{aligned}
\tag{3.8}
$$

where $\Gamma(\cdot)$ is the Gamma function. The mixtures π_j are positive and sum to one, and α is the concentration parameter whose prior has an inverse Gamma shape as $p(\alpha^{-1}) \sim \mathcal{G}a(1, 1)$. The symmetric Dirichlet hyperparameters $\frac{\alpha}{K}$ in (3.8) encode our beliefs about how uniform or skewed the class mixture weights will be [67].

At this point, we presented our model for one dimensional ($D = 1$) feature space, as stated earlier. However, in our experiments, we collected Wi-Fi radio signals sent by three different APs. Therefore, the model presented so far should be modified to fit the multivariate case. Hence, to adapt the model to the multivariate case, with $D = 3$ features, some modifications are needed, which is straightforward. We replace the normal and Gamma variables with multivariate Gaussian and Wishart distribution, respectively. Therefore, the normal variables μ_j become multinormal random vectors $\vec{\mu}_j$. The Gamma variables s_j become Wishart random matrices $\boldsymbol{\Sigma}_j$. For the remainder of this chapter, all discussion will be focused on the multidimensional space, i.e., $D = 3$.

According to [68], the conjugate prior distribution of the mean vector $\vec{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$, can be computed with Gaussian inverse Wishart (GIW) distribution, with hyperparameters $H = (\boldsymbol{\Lambda}_0^{-1}, \upsilon_0, \vec{\mu}_0, \kappa_0)$, and they are denoted as

$$
\begin{aligned}
\boldsymbol{\Sigma}_j &\sim \mathrm{IW}_{\upsilon_0}(\boldsymbol{\Lambda}_0^{-1}) \\
\vec{\mu}_j|\boldsymbol{\Sigma}_j &\sim \mathcal{N}(\vec{\mu}_0, \boldsymbol{\Sigma}_j/\kappa_0),
\end{aligned}
\tag{3.9}
$$

where IW is the inverse Wishart distribution and $\mathcal{N}$ is the multivariate Gaussian distribution. The hyperparameters, denoted by H, delineate our knowledge of the observations. Thus, the fully conjugate prior density is given by

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathrm{GIW}(\boldsymbol{\mu}, \boldsymbol{\Sigma} | \boldsymbol{\Lambda}_0^{-1}, \upsilon_0, \vec{\mu}_0, \kappa_0), \tag{3.10}$$

where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$ is the covariance matrix of a multivariate Gaussian. The GIW is given by

$$\begin{aligned} \mathrm{GIW}(\boldsymbol{\mu}, \boldsymbol{\Sigma} | H) &\triangleq \mathcal{N}(\boldsymbol{\mu} | \vec{\mu}_0, \boldsymbol{\Sigma}/\kappa_0) \cdot \mathrm{IW}(\boldsymbol{\Sigma} | \boldsymbol{\Lambda}_0^{-1}, \upsilon_0) \\ &= \frac{|\boldsymbol{\Sigma}|^{-\frac{\upsilon_0 + D + 2}{2}}}{Z_{GIW}} \exp\left[-\frac{\kappa_0}{2} (\boldsymbol{\mu} - \mu_0)^2 \boldsymbol{\Sigma}^{-1} - \frac{\mathrm{Tr}(\boldsymbol{\Sigma}^{-1} \boldsymbol{\Lambda}_0^{-1})}{2} \right] \end{aligned} \tag{3.11}$$

where

$$Z_{GIW} = 2^{\frac{\upsilon_0 D}{2}} \Gamma_D(\upsilon_0/2)(2\pi/\kappa_0)^{D/2} |\boldsymbol{\Lambda}_0^{-1}|^{-\upsilon_0/2}, \tag{3.12}$$

and $\Gamma_D(\cdot)$ is the multivariate Gamma function. The complete derivation can be found in [69, Ch. 4, pp 133].

The choice of the inverse Wishart distribution is because it is fully conjugate prior for the multivariate Gaussian. The hyperparameters, denoted by H, for the inverse Wishart have the following interpretations: $\vec{\mu}_0$ is our prior mean for $\boldsymbol{\mu}$, and κ_0 indicates how strongly we are confident about that. The hyperparameters $\boldsymbol{\Lambda}_0^{-1}$ is proportional to our prior mean for $\boldsymbol{\Sigma}$, and υ_0 encodes our confidence about that.

For reference, the PDF of the inverse Wishart distribution is given in (3.13), where υ is the number of degrees of freedom of the distribution, $\boldsymbol{\Lambda}$ is a $D \times D$ scale matrix, and $\mathrm{Tr}(\cdot)$ denotes the trace.

$$p(\boldsymbol{\Sigma}) = \frac{|\boldsymbol{\Lambda}^{-1}|^{\upsilon/2} |\boldsymbol{\Sigma}|^{-\frac{\upsilon + D + 1}{2}} \exp\left[-\frac{1}{2} \mathrm{Tr}(\boldsymbol{\Lambda}^{-1} \boldsymbol{\Sigma}^{-1}) \right]}{2^{\frac{\upsilon D}{2}} \Gamma_D(\upsilon/2)}. \tag{3.13}$$

For the sake of completeness, we also provide here the PDF of the multivariate Gaussian distribution in (3.14), where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$ is a $D \times D$ covariance matrix.

$$p(y|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(y-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(y-\boldsymbol{\mu})\right]. \tag{3.14}$$

Our purpose is to infer the class of each one of our N observations, $\mathbf{y}$, from the feature space. So, let us define a set of N indicator parameters $\mathbf{z} = \{z_1, z_2, \ldots, z_N\}$ which encode each data point y_i, i.e., z_i encodes y_i, indicating which class it belongs to. This specifically means that, when z_i belongs to class j, so does y_i with probability $p(z_i = j) = \pi_j$.

Non-fixed number of classes

So far, we assumed a fixed number of classes, K, as explained earlier. In reality, we do not know the exact number of classes in our input data, and here is where the IGMM comes, which is actually an extreme case of FGMM when $K \to \infty$.

We have chosen the $p(\boldsymbol{\pi}|\alpha)$ and $p(\vec{\mu}_j, \boldsymbol{\Sigma}_j|H)$ to be our conjugate prior, therefore one may integrate out the model parameters $\boldsymbol{\pi}$, $\vec{\mu}_j$ and $\boldsymbol{\Sigma}_j$, and sample the indicator parameters $\mathbf{z}$ to infer the class of each one of our N mobile users.

The indicator parameters $\mathbf{z}$ can be sampled according to the Bayesian principle. Indeed, Bayes' rule tells us that the posterior probability of the indicator parameters $\mathbf{z}$ given the input data $\mathbf{y}$ is proportional to the prior probability of $\mathbf{z}$ times the likelihood. Hence, the posterior distribution of the classification indicators is given by

$$\begin{aligned} & p(z_i = j|\mathbf{z}_{-i}, \mathbf{y}, \alpha, H) \\ & \quad \sim p(\mathbf{z}|\alpha)p(\mathbf{y}|\mathbf{z}, H) \\ & \quad \sim p(z_i = j|\mathbf{z}_{-i}, \alpha)p(\mathbf{y}|z_i = j, \mathbf{z}_{-i}, H) \\ & \quad \sim p(z_i = j|\mathbf{z}_{-i}, \alpha)p(y_i|\mathbf{y}_{-i}, H), \end{aligned} \tag{3.15}$$

where $\mathbf{y}_{-i}$ means that all other observations except the current one.

In order to determine the value of the posterior probability in (3.15), we should derive the expressions of the first and the second terms on the right side.

To infer the expressions for prior $p(z_i = j|\mathbf{z}_{-i}, \alpha)$, we need to integrate out the mixture weights and write the prior in terms of indicators

$$p(\mathbf{z}|\alpha) = \int_{\boldsymbol{\pi}} p(\mathbf{z}|\boldsymbol{\pi})p(\boldsymbol{\pi}|\alpha)d\boldsymbol{\pi}, \tag{3.16}$$

where the first term

$$p(\mathbf{z}|\boldsymbol{\pi}) = \prod_{j=1}^{K} \pi_j^{n_j}, \tag{3.17}$$

and the second term is given in (3.8). Hence, following [69] we have

$$\begin{aligned} p(\mathbf{z}|\alpha) &= \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \int_{\pi} \prod_{j=1}^{K} \pi_j^{n_j + \frac{\alpha}{K} - 1} \\ &= \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)} \prod_{j=1}^{K} \frac{\Gamma(n_j + \alpha/K)}{\Gamma(\alpha/K)}, \end{aligned} \tag{3.18}$$

where n_j is the number of observations belonging to class j.

We have applied Dirichlet distribution to model the distribution of the mixture weights, $\boldsymbol{\pi}$, when the number of the clusters is supposed known. However, in IGMM, it can be hard to sample the mixture weights directly from Dirichlet distribution, i.e., when the number of the clusters goes to infinity. Another method, called stick-breaking construction [70] can be, instead, utilized. It is simply given as follows:

$$\beta_j \sim \text{Beta}(1, \alpha) \tag{3.19}$$

$$\pi_j = \beta_j \prod_{l=1}^{j-1} (1 - \beta_j). \tag{3.20}$$

We already know that the sum of the mixture weights is equal to one, $\sum_{j=1}^{K} \pi_j = 1$. Hence, starting with a stick of length one and break it into two separate parts at β_1, assigning π_1 to be equal to the length of either one of the two parts. β_1 is sampled according to the Beta distribution. Then, repeat the same process on the other portion to obtain π_2, π_3 and so on.

Our goal is to sample from posterior distribution over the model when the limit $K \longrightarrow \infty$. An Markov chain Monte Carlo (MCMC) technique known as Gibbs sampling [71] [72] is used to sample the distribution and determine the class label of each mobile user. Gibbs sampler makes this possible, by repeatedly replacing each component with a value taken from its conditional distribution on the current values of all other components. Therefore, to use Gibbs sampling for the indicators, z_i, we need conditional prior for a single indicator given all the others. By keeping all but a single indicator fixed in (3.18), we obtain

$$p(z_i = j|\mathbf{z}_{-i}, \alpha) = \frac{n_{-i,j} + \alpha/K}{N - 1 + \alpha}, \tag{3.21}$$

where $\mathbf{z}_{-i}$ are the classes for the observations other than y_i, and $n_{-i,j}$ represent the number of observations in class j before y_i belonging to.

When $K \longrightarrow \infty$ in (3.21), the conditional prior reaches the followings limits

$$p(z_i = j|\mathbf{z}_{-i}, \alpha) = \begin{cases} \frac{n_{-i,j}}{N-1+\alpha}, & \text{if } n_{-i,j} > 0, \\ \frac{\alpha}{N-1+\alpha}, & \text{if } n_{-i,j} = 0 \end{cases} \tag{3.22}$$

where $n_{-i,j} = 0$ means that, no observation has been assigned yet to class j. The generative model in (3.22) is a characterization of Dirichlet process known as Chinese restaurant process (CRP) [73] [74].

The CRP metaphor is described as follows. Imagine a Chinese restaurant with an infinite number of tables. Each table corresponds to a specific cluster. The customers that enter the restaurant are, in our case, the observations. The first customer enters and chooses a table, i.e., selects a cluster. Then, the ith customer enters and chooses an empty table or an occupied one. Therefore, the probability of choosing an occupied table is given by $\frac{n_{-i,j}}{N-1+\alpha}$, which is proportional to the number of customers, $n_{-i,j}$, who have already chosen this particular table. In turn, the probability of choosing an empty table is given by $\frac{\alpha}{N-1+\alpha}$, which is proportional to the concentration parameter α. From this, we can notice that, the more customers are at a particular table, it is more likely the new customer will join it. In the contrary, the probability of joining a completely new table is very small.

Same as the first term in (3.15) (right side) follows two cases, described in (3.22), we may also find two expressions for the second term. Indeed, following [67] and [69], the second term in (3.15) is obtained by the multivariate Student-t distribution, because of our previous choice of conjugate prior. Therefore,

$$p(y_i|\mathbf{y}_{-i}, H) \sim t_{\upsilon_n - D+1}\left(\vec{\mu}_n, \frac{\mathbf{\Lambda}_n(\kappa_n + 1)}{\kappa_n(\upsilon_n - D + 1)}\right), \tag{3.23}$$

where t is the multivariate Student-t distribution. The subscript $\upsilon_n - D + 1$ is its number of degrees of freedom. The rest of the parameters in (3.23) are defined as follows

$$\begin{aligned}
\vec{\mu}_n &= \frac{\kappa_0}{\kappa_0 + N}\vec{\mu}_0 + \frac{N}{\kappa_0 + N}\bar{y} && (3.24)\\
\kappa_n &= \kappa_0 + N && (3.25)\\
\upsilon_n &= \upsilon_0 + N && (3.26)\\
\mathbf{\Lambda}_n &= \mathbf{\Lambda}_0 + S + \frac{\kappa_0 n}{\kappa_0 + N}(\bar{y} - \vec{\mu}_0)(\bar{y} - \vec{\mu}_0)^T && (3.27)
\end{aligned}$$

and $\bar{y}$ is the mean of observations, D is the dimensionality. $\mu_l, \kappa_l, \upsilon_l$ and $\mathbf{\Lambda}_l$ are the updated hyperparameters after observing samples, and S is defined as

$$S = \sum_{i=1}^{N}(y_i - \bar{y})^2. \tag{3.28}$$

For the case where no user has been assigned to a cluster, we need to find $p(y_i, H)$. In fact, it has the same form as $p(y_i|\mathbf{y}_{-i}, H)$, given in (3.23), with the hyperparameters before updating

$$p(y_i, H) \sim t_{\upsilon_0 - D+1}\left(\vec{\mu}_0, \frac{\mathbf{\Lambda}_0(\kappa_0 + 1)}{\kappa_0(\upsilon_0 - D + 1)}\right). \tag{3.29}$$

For reference, the PDF of the multivariate Student-t distribution is given in (3.30), where υ is the degrees of freedom, $\boldsymbol{\mu}$ is the mean, and $\mathbf{\Lambda}$ is a $D \times D$ scale matrix.

$$t_\upsilon(y|\boldsymbol{\mu}, \mathbf{\Lambda}) = \frac{\Gamma(\frac{D+\upsilon}{2})}{\Gamma(\frac{\upsilon}{2})} \frac{|\mathbf{\Lambda}|^{1/2}}{(\pi\upsilon)^{D/2}} \left[1 + \frac{(y - \boldsymbol{\mu})^2 \mathbf{\Lambda}^{-1}}{\upsilon}\right]^{-\frac{\upsilon+D}{2}}. \tag{3.30}$$

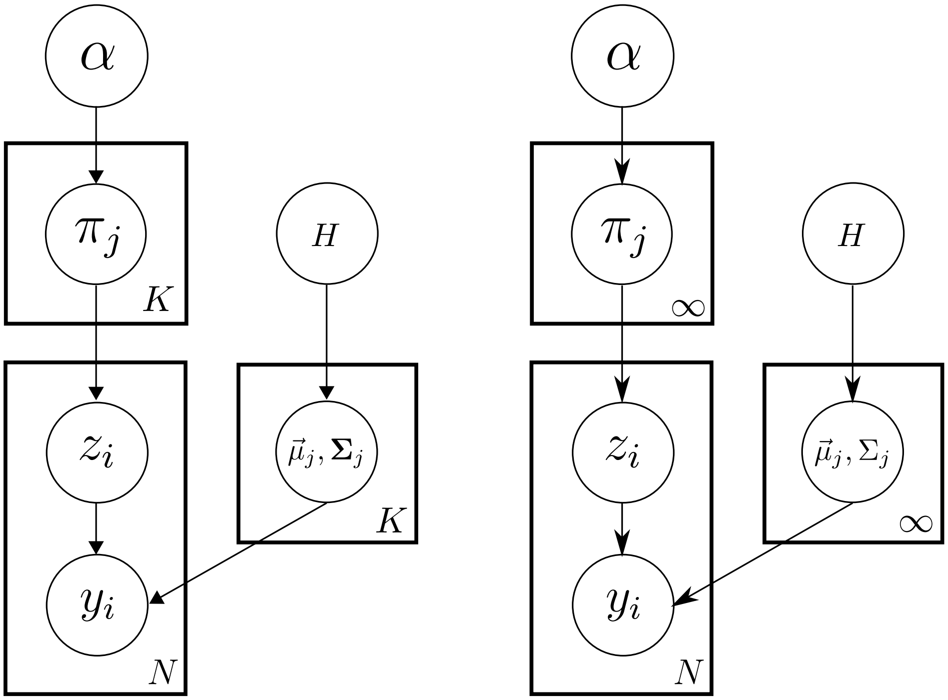
As a conclusion, we can say that, to be able to compute the posterior probability for our indicators $\mathbf{z}$, we need to determine the posterior distribution when there are observations assigned to an existing cluster. This is done by

$$p(z_i = j|\mathbf{z}_{-i}, \mathbf{y}, \alpha, H) \sim \frac{n_{-i,j}}{N - 1 + \alpha} t_{\upsilon_0 - D+1}\left(\vec{\mu}_0, \frac{\mathbf{\Lambda}_0(\kappa_0 + 1)}{\kappa_0(\upsilon_0 - D + 1)}\right), \tag{3.31}$$

and when there is no observation assigned to a cluster. That one is given by

$$p(z_i \neq z_{i'}, \forall i \neq i'|z_{-i}, \mathbf{y}, \alpha, H) \sim \frac{\alpha}{N - 1 + \alpha} t_{\upsilon_0 - D+1}\left(\vec{\mu}_0, \frac{\mathbf{\Lambda}_0(\kappa_0 + 1)}{\kappa_0(\upsilon_0 - D + 1)}\right). \tag{3.32}$$

Figure 3-2 depicts the graphical representation of this model. In Figure 3-2(a), the FGMM is portrayed and its nonparametric version, which is the one used in this work to co-localize mobile users, is exhibited in Figure 3-2(b). It illustrates the conditional relationships among parameters, hyperparameters and input data in IGMM. For example, it shows that the indicator z_i depends on π_j, which in turn depends on the parameter α. The rectangular blocks represent the repetition, and the symbol in the lower right corner indicates the number of repetitions.



(a) Finite

(b) Infinite

Figure 3-2: Graphical model representation of Bayesian Gaussian mixture model: (a) finite and (b) infinite. We adopt the infinite model in our co-localization system.

3.2.2 Modified Gibbs Sampling

The proposed co-location algorithm exploits the similarity of users' measurements of their shared ambient radio signals. So, they are assigned to the same cluster depending on their reported Wi-Fi radio signals.

As we mentioned above, depending on application requirements, one can define how near two users should be considered as co-located. In the sense that there is no precise distance of nearness between two users, for instance, to deduce that they are co-located.

The two posterior distributions discussed so far permit us applying Gibbs sampler to sample the values of the indicator parameters $\mathbf{z}$, to infer the class label of each user. To take into account how near two users should be considered as co-located or not, we have introduced a similarity threshold denoted by Δ (explained in detail later on) in our algorithm. That is, when two users' measurements differ less than the similarity threshold Δ, we regard these users as co-located. More specifically, we first compute the average similarity denoted by $AVGSIM$ of each existing cluster as follows

$$AVGSIM_j = \frac{1}{n_j} \sum_{i=1}^{N} \delta_{\text{Kronecker}}(z_i, j), \tag{3.33}$$

where $AVGSIM_j$ denotes the average similarity of the jth cluster, and $\delta_{\text{Kronecker}}(z_i, j)$ is the Kronecker delta function representing the ith observation encoded by indicator parameter z_i, belonging to the jth cluster. It has the task of retaining all the observations that belong to a specific cluster j, when $z_i = j$ in the summation. That is, when the observation y_i encoded by z_i belongs to the class j, this observation is taken in the summation, otherwise not.

Then, for a new incoming observation, y_i, the Euclidean distance denoted by $DIST_{(i,j)}$, i.e., the distance between the ith observation, y_i, and the jth average similarity, $AVGSIM_j$, is computed in signal domain. If the computed distance, $DIST_{(i,j)}$, is less than or equal to the predefined similarity threshold Δ, the user is accepted in that cluster, i.e., $DIST_{(i,j)} \leq \Delta$.

Note that, n_j is the number of observations in cluster j, and N is the total number of observations. z_i is our indicator parameter that encodes the ith observation indicating with cluster the observation belongs to. With this approach we were able to leverage our co-location accuracy.

With respect to a moving user, who is walking around or just passing by, we noticed that his measured ambient radio signals change a lot over time compared with users that are interacting with others. So, we define a period of time, Δt, that users should have been together in order to classify them into the same cluster. Δt should be set large enough in order to ensure that people have spent time together.

Algorithm 1 shows the necessary steps of our modified Gibbs sampling for IGMM-based co-location. The variable T indicates the number of iterations to be accomplished by the algorithm. It should be set large enough to ensure accurate sampling.

Algorithm 1 Collapsed Gibbs sampler for IGMM-based co-location

```
Input : Data sets from $N$ users, and pre-set threshold $\Delta$.
Output: Users co-located in $K$ clusters.
Initialize: Set all users into the same cluster, $K = 1$.
for $t = 1$ to $T$ do
    for $i = 1$ to $N$ do
        Remove $y_i$ from its current class.
        for $j = 1$ to $K$ do
            Compute prob. of an existing class as in (3.31).
            $AVGSIM_j \leftarrow$ (3.33)
            $DIST_{(i,j)} \leftarrow$ distance to cluster $j$.
        end for
        Compute prob. of a new class as in (3.32).
        $z_i \leftarrow$ class with highest prob. and $DIST_{(i,j)} \leq \Delta$.
        Remove any empty class, and decrease $K$.
    end for
end for
```

3.2.3 Co-location Scheme Detection

To detect and cluster co-located users, we propose the following scheme (see Figure 3-3). Ambient radio signals are sensed for a period of time Δt, and the collected data signals are sent to the co-location server to be processed. Upon receiving the data signals, the server will create distinct lists of users with the same APs. Then, for each user a $\Delta\sigma_j$ (fleshed out later on) is calculated in order to determine if a user is interacting or not with others. Note that, in this case, we compare $\Delta\sigma_j$ with a threshold denoted by Θ. More on this threshold Θ will be discussed later on.

In the next step, the mean of received signals of each user is computed and assigned all the users to the same class, $K = 1$, to start the classification process. Then, hyperparameters and parameters of IGMM are computed, as well as the average similarity of each cluster. For an incoming observation, Gibbs sampling will give us its cluster, i.e., it will belong to an existing cluster or a new one. Based on a predefined similarity threshold Δ we assign this incoming observation into an existing cluster predicted by Gibbs sampler or a new one. This is performed by comparing its distance to the center of the predicted cluster. The optimum value of the similarity threshold Δ is estimated in offline analysis in Section 3.4. Finally, the users with the strongest $\Delta\sigma_j$ are assigned to different classes at the end of the algorithm.

As can be noticed, so far, we mainly focus on the clustering of mobile users who have been spent time together in the same place. In Chapter 4, we will discuss a challenging case when users are walking together as part of the same group and show how to cluster them in real-time.

The proposed scheme has several advantages. One of them is that mobile users who experience radio signals from different APs, in terms of the placement where these APs are, will never be clustered together. Another one is that by introducing the similarity threshold Δ in our clustering process, we are able to determine all existing clusters in the input data, as it is shown in the next sections from our numerical and experimental results. The proposed approach is also robust to deal with the varying number of clusters and users over time. Indeed, this is one of the many appealing properties of NPB, i.e., the ability to automatically infer the number of clusters in the input data.

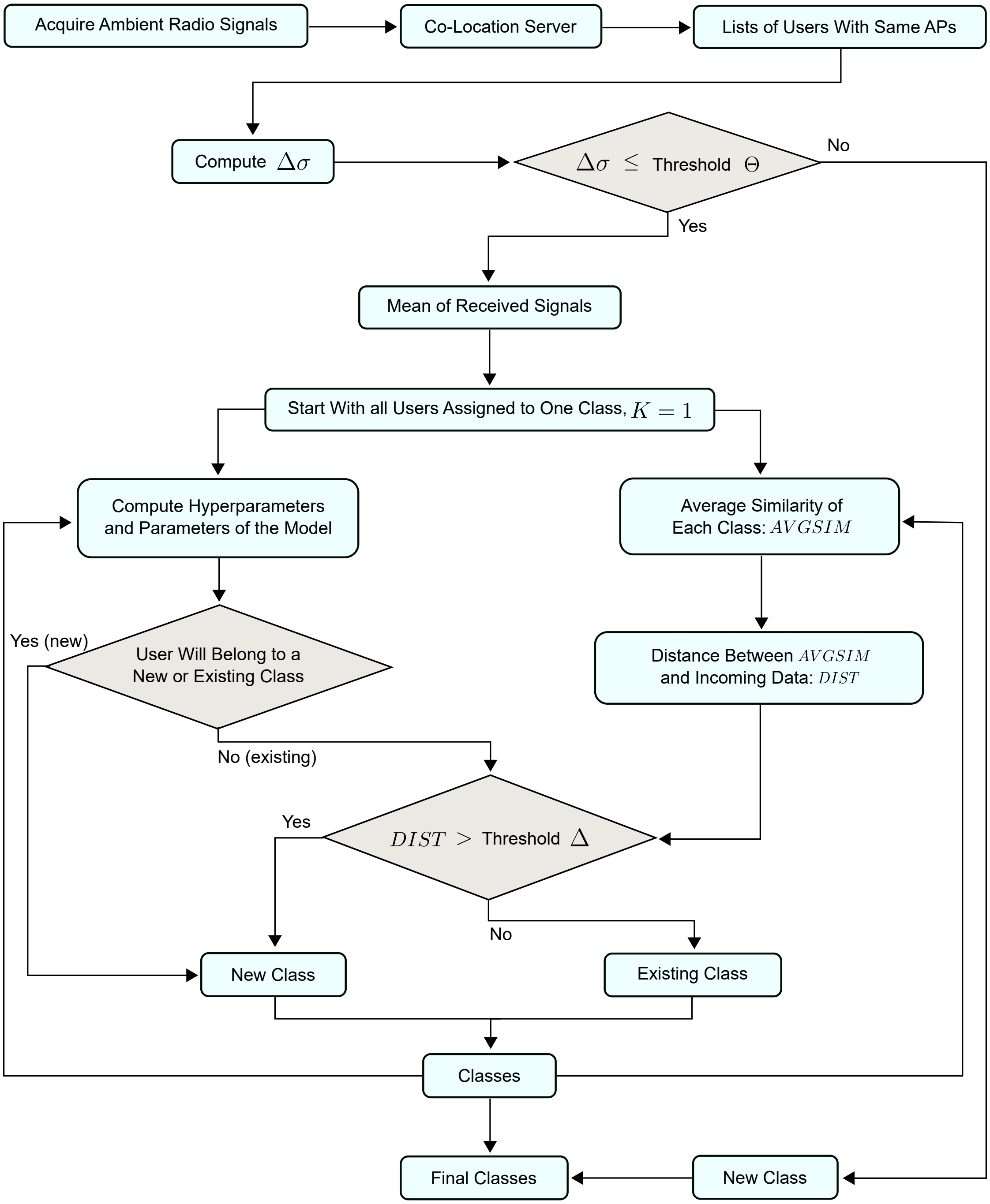

Acquire Ambient Radio Signals
Co-Location Server
Lists of Users With Same APs
Compute $\Delta\sigma$
$\Delta\sigma \leq$ Threshold Θ
No
Yes
Mean of Received Signals
Start With all Users Assigned to One Class, $K = 1$
Compute Hyperparameters and Parameters of the Model
Average Similarity of Each Class: $AVGSIM$
Yes (new)
User Will Belong to a New or Existing Class
Distance Between $AVGSIM$ and Incoming Data: $DIST$
No (existing)
Yes
$DIST >$ Threshold Δ
No
New Class
Existing Class
Classes
Final Classes
New Class

Figure 3-3: Flow chart of co-localization algorithm based on IGMM.

3.3 Numerical Results

Our co-localization algorithm is first assessed numerically, and then experimentally. In this section, we will present our numerical results.

We considered a square area of interest Q of 460 m^2 with four access points (APs), located each one on its corner. Then, we randomly deployed 50 nodes (users) in different regions of that testing area. The received signal strength indicator (RSSI) is sampled 20 times per seconds, and then we took the average. Each node reports its measured RSSI from each AP, and the proposed algorithm tests the similarities among the reported RSSIs to decide the cluster of each one of them, according to their similarity measurements.

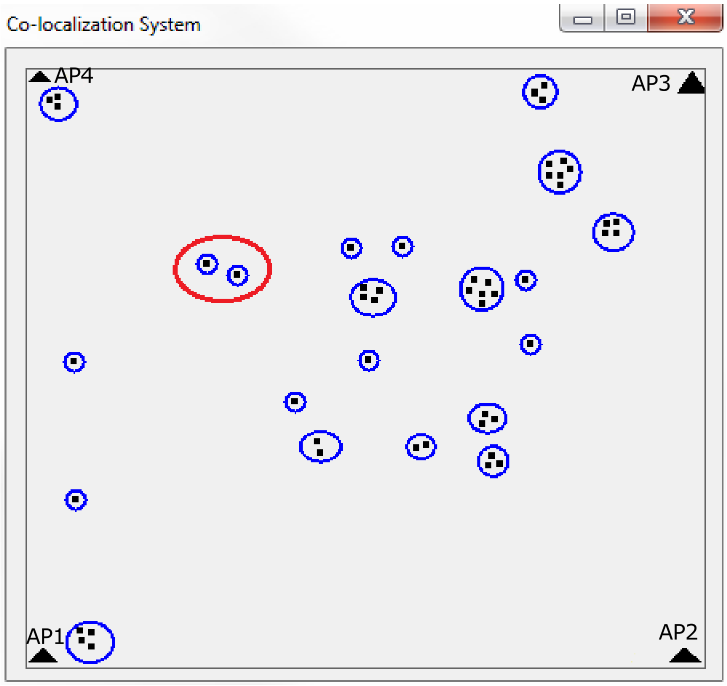


Figure 3-4: Numerical results of our co-localization system. Each black dot represents a user in the wireless network. The blue circles indicate the actual co-located group of users. The red circle shows the misclassification case. We consider four APs in this simulation.

Figure 3-4 depicts the obtained results. Each black point on this figure is considered as a user, and the blue circles indicate the true clusters. The red circle means the misclassification case. To obtain a such result, we set the similarity threshold Δ

to 1.05. This optimum value of Δ is obtained by trial-and-error process. That is, the inference of the optimum value of the similarity threshold Δ is computed in function of the number of users correctly clustered at each step size. The correct clustering of mobile users, at each step size, is defined as the number of mobile users clustered together by the proposed algorithm that is in accordance with our numerical setup. That is, if *User A* and *User B* are inferred by the proposed algorithm to be in the same group and it turns out that, in our numerical setup, these users were actually together, we consider this inference as a correct clustering by the algorithm. As the moving users are not considered in this simulation, the threshold Θ is not used. It will be discussed in Subsection 3.4.2.

As can be seen in Figure 3-4, the algorithm was able to detect the correct cluster of almost all nodes. Only two out of 50 nodes were wrongly clustered (red circle). In fact, these two nodes form each one its own cluster. Thus, 98% of nodes were correctly clustered.

In this simulation, we chose the value of the similarity threshold Δ, by trial-and-error process, that gave us the best results. However, as it is explained more thoroughly in the next section, the value of this threshold can be determined in offline analysis, and set according to the application requirements. It should also be noted that different environments (indoor and outdoor) have different effect on the choice of the threshold. We did not carry out experiments to show how it varies with different environments. However, we discuss this issue as part of the future research on this topic in Chapter 5.

3.4 Experimental Setup and Results

In this section, we first discuss our experimental setup and present the obtained results using collected real-world Wi-Fi signals. Then, in subsection 3.4.5, we compare the performance of our method, in terms of clustering accuracy, against community detection-based approach proposed in [30] to co-locate mobile users.

3.4.1 Experimental Setup

To evaluate the performances of the proposed algorithm with a real-world setting, we carried out an extensive experiment in an entire second floor of a building with six participants, collecting Wi-Fi signals in different places in ten different time-stamps. The testing area is a 1200 m^2 of a floor in a building composed with several meeting rooms, an open space, and corridors (see Figure 3-5).

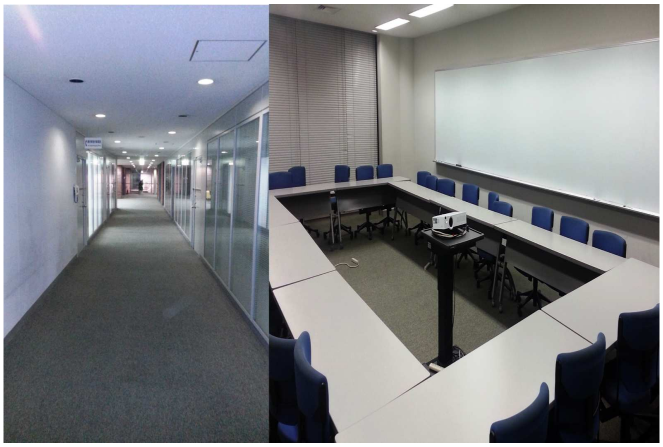


Figure 3-5: A corridor (left side) and a meeting room (right side) of a 2nd floor of a building where the experiments were conducted.

We utilized wireless adapters *AirPcap Nx* [75] and a free and open-source packet analyzers *Wireshark* [76] to simultaneously capture environmental radio signals. Wi-Fi signals were recorded for a period of time of one minute. Then, all measurements were put together to be processed on a computer.

RSSI, MAC address, and time arrival of beacon packets at 2.437 GHz from the same APs were extracted for one minute. For this experiment, users' measurements from three different APs deployed in a typical office building were considered. In this work, three different APs were considered because it is large enough to represent the unique signature of the location where the radio signals were captured. The fact that we collected ambient radio signals during a period of time of one minute for each user, and then took the average of each user, allows us to considerably reduce the measurement errors.

The concentration parameter, α, and the hyperparameters denoted by $\boldsymbol{H} = (\boldsymbol{\Lambda}_0^{-1}, \upsilon_0, \vec{\mu}_0, \kappa_0)$

in IGMM model express our prior belief on the distribution and need to be specified roughly [69]. Therefore, in our implementation we proceeded as follows. We used the standard setting for the concentration parameter α, i.e., $p(\alpha^{-1}) \sim \mathcal{G}a(1,1)$. The mean vector $\vec{\mu}_0$ is set from our data sets. The hyperparameter κ_0 that encodes how confident we are about our mean is set to 0.5. Λ_0 is chosen to be a diagonal matrix of 0.1, and υ_0 that represents our confidence about Λ_0 is set to 20.

3.4.2 Inferring Interacting Users

We investigated the effect of walking users on a group of other users within a room, i.e., while there is a group of users in a room, other users are walking in a corridor. The purpose of this investigation is to evaluate the group detection process, when a user is walking around and does not interact with the group.

As group meeting time is an important characteristic of co-location, we evaluated the radio signals when users are interacting or sharing a certain amount of time together, and when users are walking around or just passing by. The goal is to be able to differentiate between interacting and non-interacting users.

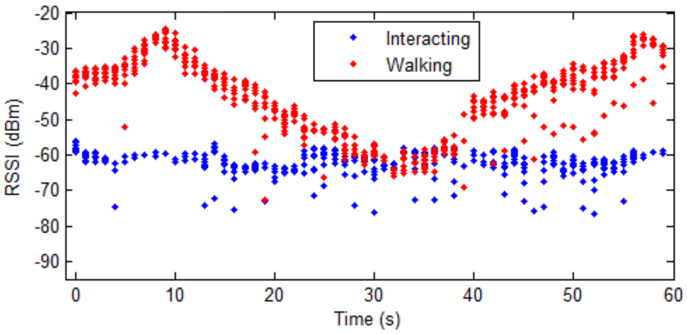


Figure 3-6: RSSIs extracted from interacting (blue dots) and walking (red dots) users, for a period of time (Δt) of one minute. Interacting users were in the same room, while a user was walking in the corridor.

Figure 3-6 shows the collected RSSIs from the same APs when a user is interacting or sharing some amount of time with other users (i.e., belonging to a cluster of users, blue dots), and when the same user is walking in a corridor (red dots), during the same period of time (one minute). As one can observe, on this figure these two measurements have different power levels. Therefore, we propose a method for their detection in real-time based on a predefined threshold denoted by Θ.

Unsurprisingly, when the user is interacting with others, i.e., the user does not move a lot over time (for a period of time Δt), the measured radio signals are almost the same (blue dots). On the contrary, when the same user is walking in a corridor, the experienced radio signals change a lot over time (red dots). Therefore, we differentiate these two kinds of users (interacting and non-interacting) as follows: the standard deviation $\sigma_{j,i}$ for each user of each AP is computed; then, we square and sum the obtained value of $\sigma_{j,i}$ from each user; and finally, a square root of it is computed. Hence, the $\Delta\sigma_j$ for each user is obtained, as it is shown in (3.34)

$$\Delta\sigma_j = \sqrt{\sum_{i=1}^{D} \sigma_{j,i}^2} \tag{3.34}$$

where D is the dimension of the observation, $\sigma_{j,i}$ is the standard deviation of the jth user for ith AP.

Table 3.1. $\Delta\sigma_j$ According to user actions (a hyphen means that no measurement was collected)

Users	Interacting	Walking
A	6.12	18.42
B	7.38	-
C	6.11	-
D	6.63	18.73
E	6.85	-
F	6.49	-

Table 3.1 shows the obtained values of $\Delta\sigma_j$ for two different kinds of users' actions (interacting and walking). As expected, their values are quite different. Accordingly, any value that can unambiguously differentiate these two kinds of users' actions can be chosen between these two sets of values. In our implementation, we set the threshold Θ

to 12.5. The dash lines in the walking column of Table 3.1 mean that no measurement was collected for this particular user concerning that action. This is explained by the fact that, in our experiment, we have chosen only two distinct users to collect Wi-Fi signals while they were walking.

It is worth noting that the obtained values of interacting users are almost the same, and also the values of walking users are almost the same, which comfort us in our choice of the value of the threshold Θ.

3.4.3 Similarity Threshold Δ

The proposed algorithm clusters users based on the similarity of their measured radio signals and physical proximity. As previously mentioned, there is no fixed measure of nearness between two users to affirm that they are co-located. Consequently, when measurements from two distinct users differ less than the predefined similarity threshold Δ, they are regarded to be potentially co-located. Therefore, we performed an offline analysis in order to determine the best value of the similarity threshold Δ for users to be part of the same group, i.e., how near two or more users should be considered as co-located.

We started by calculating the Euclidean distance between each pair of user's measurement. Thus, we noticed that when two or more users belong to the same group, their computed Euclidean distances are shorter than those from the other groups. It means that, by setting up a suitable value for the threshold Δ, we can accurately cluster co-located users.

Table 3.2 displays the minimum and the maximum Euclidean distances found in each cluster with two or more users. This table exhibits the values of nine clusters, because actually there are nine clusters with two or more users. The minimum distance of all clusters is found to be 0.07, and the maximum distance is found to be 3.8. They are printed in bold in Table 3.2.

According to the above obtained values (minimum and maximum), we defined the similarity threshold interval, i.e., the range on which the optimum value of the

Table 3.2. Max and min Euclidean distance in each cluster

	Minimum	**Maximum**
Cluster 1	0.32	2.31
Cluster 2	0.16	1.67
Cluster 3	**0.07**	1.79
Cluster 4	0.30	1.87
Cluster 5	0.89	**3.8**
Cluster 6	0.59	2.7
Cluster 7	0.71	2.13
Cluster 8	0.09	1.74
Cluster 9	1.03	2.27

similarity threshold Δ can be found. Otherwise, the scope will be too large to easily find one.

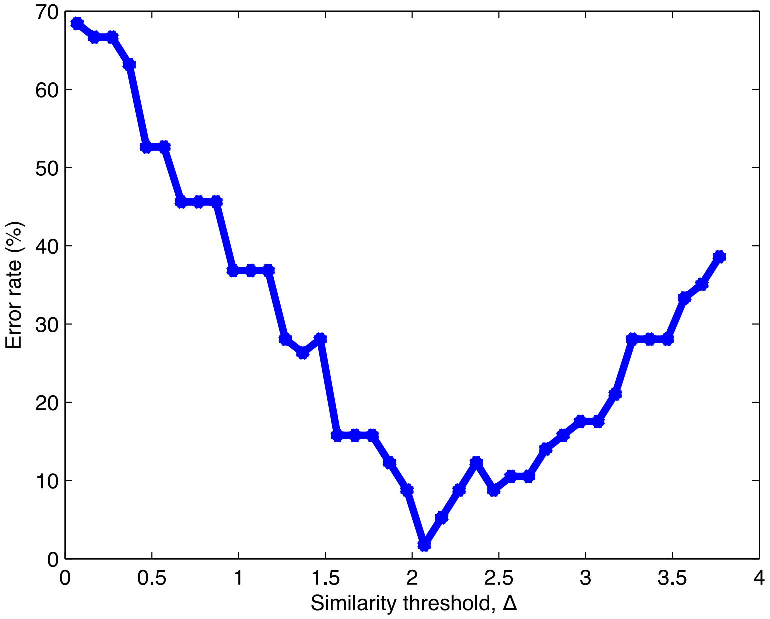


Figure 3-7: Effect of the similarity threshold Δ on users co-localization. The value of Δ is computed in the signal domain for Euclidean distance metric.

Figure 3-7 depicts the effect of the threshold Δ on classification accuracy for the normalized Euclidean distance metric. In this figure, one can notice that, when the value of the threshold Δ increases, the error rates decrease until attain its optimum value at approximately the middle of the interval, and then it retakes its growth. This corroborate our proposal of clustering co-located users by computing the average similarity of each cluster, and accept an incoming user if his distance to the center of that cluster is less than the similarity threshold Δ. Therefore, the optimal value of Δ is found to be 2.07, i.e., the value that the best minimizes the error rate. As we shall see, in the next chapter, when the value of the error rate is of the zero percent, one hundred percent of accuracy is achieved.

During the experiments, we demanded the mobile users in the same group to stay apart from one another at a maximum distance of two meters. They also stayed in that place during three minutes, i.e., the period of time Δt required to the mobile users to be together in order to consider them as co-located. In our implementation, however, we took the data signals collected during one minute. From this, we consider that two mobile users are co-localized (i.e., forming a cluster) if the distance between them is less than or equal to two meters and they pass a certain amount of time together (Δt = 1 min).

For the inference of the optimum value of the similarity threshold Δ, we compute its values in function of the number of users correctly clustered at each step size. The correct clustering of mobile users at each step size is defined as the number of users clustered together by the proposed algorithm that is in accordance with our experimental setup. That is, if *User A* and *User B* are inferred by the proposed algorithm to be in the same group and it turns out that, in our experiments, these users were actually together, we consider this inference as a correct clustering by the algorithm.

It should be pointed out that, the optimal value of the similarity threshold Δ is chosen in accordance with the application setup. In fact, if we envisage a reduced distance between members of the same clusters, the value of the threshold Δ can be decreased. Consequently, more clusters will be found with smaller size. On the other hand, by increasing the value of the threshold Δ (more than the optimal) we also increase the intra-cluster distances, i.e., we increase the distance between members within clusters, which in turn produces small number of clusters, but with bigger size. In this sense, the threshold Δ must be regarded as a key parameter to take into account in this kind of applications.

As aforementioned, our analysis on the threshold is focused on its optimal value. That is, the one that can give us back our co-located groups of people with the highest accuracy possible. However, any other values of the threshold that cannot retrieve accurately our co-located groups of mobile users can be considered as suboptimal. The aim of our analysis is to show that the proposed method can be applied to correctly infer co-located groups of people with high accuracy. Nevertheless, different environments, applications, and purposes may require different values of the threshold Δ, which should be taken into consideration to fulfill the potential of the proximity-based services [17].

3.4.4 Experimental Results

In this subsection, we present our experimental results. All the pre-computed thresholds are considered, and the setup is as described previously.

By taking into account the two predefined thresholds (Θ and Δ), our algorithm was able to detect almost all clusters, and classify users into their correct classes, as it is shown in Figure 3-8.

Figure 3-8 depicts the map of the entire floor where the experiment was conducted and the obtained results. The black and blue dots on this map represent users in wireless network. The black circles surrounding dots illustrate the actually co-located users, and the red dash circle means the misdetection group. The blue dots with a blue arrow each one, surrounding by a black circle, indicate the users that were walking in the corridor while we conducted the experiments.

For the misclassification case (red dash circle), we noticed that the users in the room were separated from the user in the corridor by a plate thin glass, which made some trouble to the algorithm to differentiate these to clusters.

From this result, we can see that the value of the threshold (which is in fact the

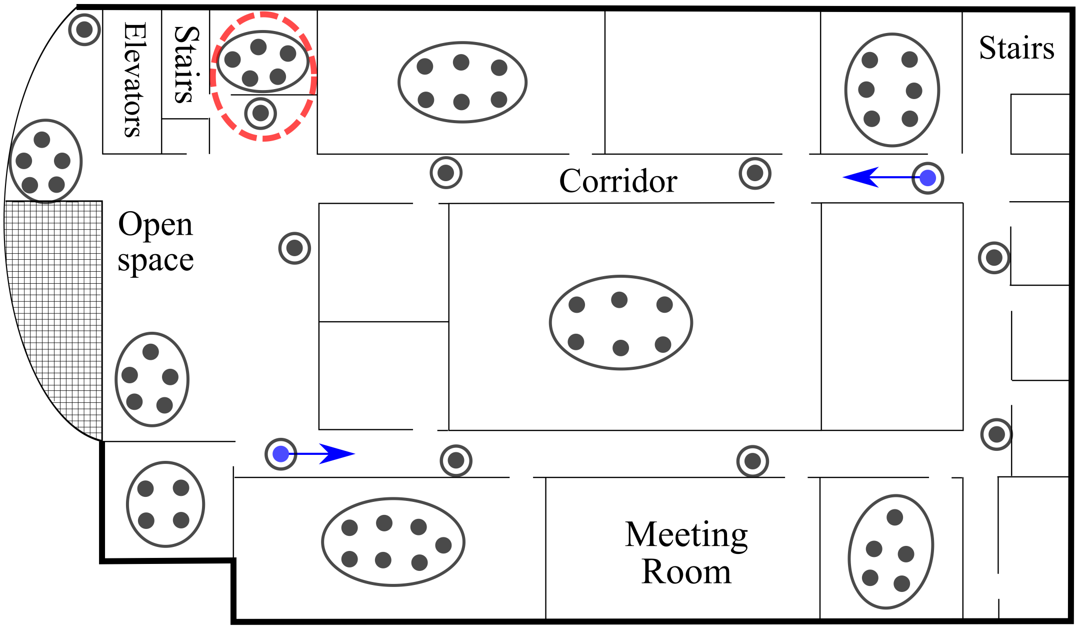


Figure 3-8: Entire floor plan where the experiments were conducted and the obtained results. Each black or blue dot exemplifies a user in wireless network. The blue dots with a blue arrow indicate walking or just passing by users in the corridor. The black circles surrounding dots represent actual co-located users. The red dash circle denotes the misclassification case.

level of similarity between users' measurements) is an important parameter to take into account in this kind of applications. It defines how close or how far we want users to be considered as co-located. If we choose its value to be less than the optimum (i.e., less than 2.07), more number of clusters will be found in the data set but with scanty number of users into it. On the other hand, when its value is set higher than the optimum (i.e., higher than 2.07), less number of clusters will be found in the data set. However, each one of these discovered clusters has an important number of users into it.

3.4.5 Comparative Results

In this subsection, we will perform a comparative study between our proposal and the community detection-based approach presented in [30], on our measured Wi-Fi signals.

As mentioned earlier, the authors in [30] proposed to co-locate mobile users by constructing a connectivity graph that represents the potential co-located users, based on pairwise similarity of RF measurements. Then, they applied community-detection [58] tools to cluster users into the same group. Moreover, an objective function called “modularity” is used. This modularity function is optimized with a heuristic method called simulated annealing [59]. As they utilized community detection (CD) tools and simulated annealing (SA) method to co-locate mobile users, henceforth we will call their approach CDSA-based.

In this work, we also exploited the similarity of user’s RF measurements from their mobile phones to cluster them into the same group. However, we do not consider any connectivity graph among them. Instead, we leverage co-located users by applying a nonparametric Bayesian method called IGMM with a modified version of Gibbs sampling to infer users’ corresponding groups. Throughout these comparative studies we will call our approach IGMM-based, and the one proposed in [30] CDSA-based.

For the sake of comparison, we performed an offline analysis to obtain the optimum value of similarity threshold denoted by δ, for CDSA-based, using the Euclidean distance metric. As the similarity threshold δ depends on the data signals and is set in accordance with application requirements, we determined its best value from our measured Wi-Fi signals. Therefore, we computed the best value of δ between an interval of $[min, max]$ with step size denoted by $\Delta\delta$, as the authors suggested to do in [30]. We used our predefined similarity threshold interval in this case. With the obtained value of the threshold δ, we proceeded with the evaluation process.

Notice that, in this comparative studies, we compared the performance of the algorithms with users that are interacting with others, i.e., users that have been together for some amount of time, and in the same place. We do not consider the users that are walking or just passing by.

Figure 3-9 shows the impact of δ on connectivity errors for the normalized Euclidean distance metric. The value of step size $\Delta\delta$ is set to 0.01. The optimal similarity threshold δ is chosen to minimize both false negative (misdetection) and false positive connectivity errors. The best value of the threshold δ for our data set is found to be 1.48.

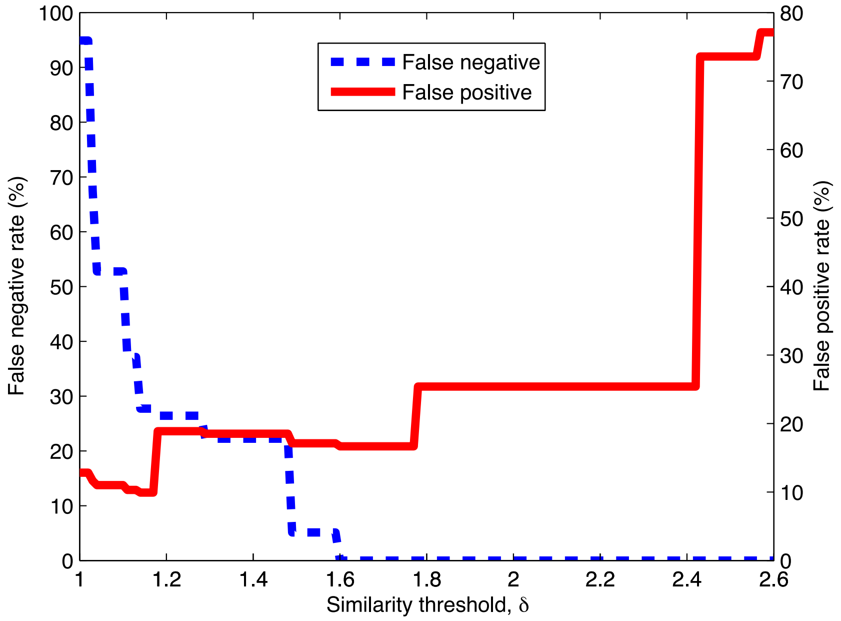


Figure 3-9: Impact of the similarity threshold δ on connectivity errors, for Euclidean distance metric. The step size $\Delta\delta$ is set to 0.01. The value of δ is computed in the signal domain.

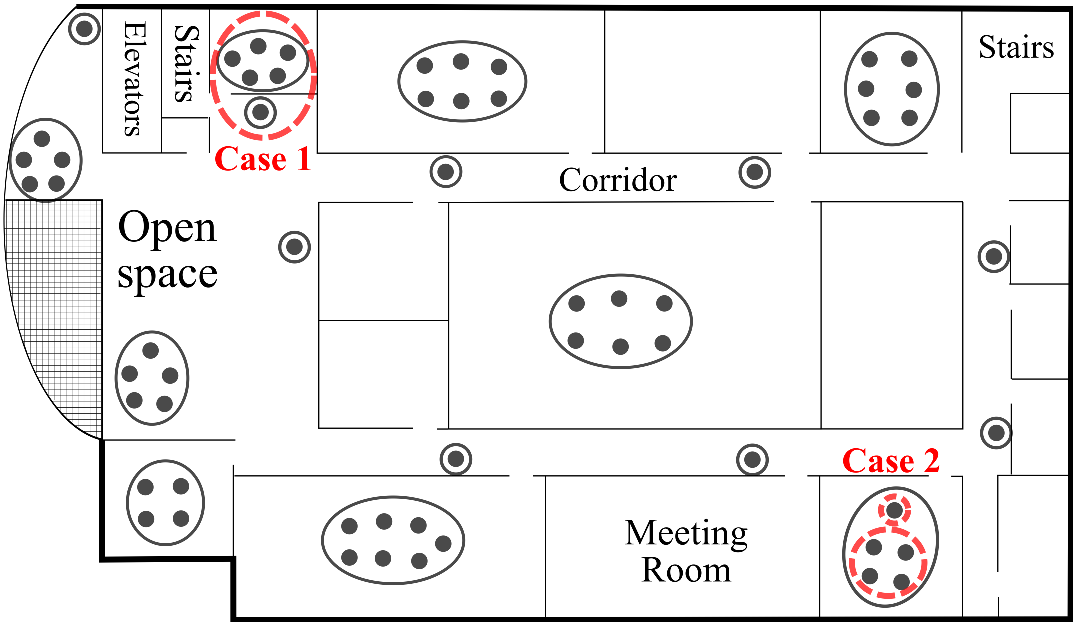


Figure 3-10: Entire floor plan where the experiments were conducted and the obtained results, using CDSA-based approach. The red dash circles indicate the misclassification cases. The setup is the same as in Figure 3-8, but without walking users.

Figure 3-10 shows the obtained results applying CDSA-based algorithm, with the value of the threshold δ set to 1.48. As one can see, both algorithms (IGMM-based and CDSA-based) misclassified a user in **Case 1**. However, CDSA-based approach in addition misclassified a user in **Case 2**.

We mainly believe that this misclassification in **Case 2**, on the one hand, is due to the predefined similarity threshold δ. On the other hand, the heuristic technique called simulated annealing used to maximize the modularity function, i.e., to maximize the intra-cluster edges, avoids getting stuck in local optima-solutions that are better than any others nearby, but are not the very best one.

Table 3.3. Performance comparison using Euclidean and Minkowski distance metrics

	Euclidean		**Minkowski** ($p = 1.5$)	
	Threshold	Accuracy	Threshold	Accuracy
IGMM-based	2.07	98.27%	1.97	94.82%
CDSA-based	1.48	96.55%	1.70	94.82%

Table 3.3 presents the performance comparison between the IGMM-based and CDSA-based algorithms using Euclidean and Minkowski distance metrics. The Minkowski distance (l_p-norm, $p \geq 1$) [77] can be considered as a generalization of the Euclidean distance, and is calculated in the signal domain as

$$d_{Mink} = \sqrt[p]{\sum_{i=1}^{D} \left| RSSI_i^{(k)} - RSSI_i^{(m)} \right|^p} \tag{3.35}$$

where $RSSI_i^{(k)}$ and $RSSI_i^{(m)}$ denote the RSSI values observed by the kth and mth users, respectively, from the ith AP. The order $p = 2$ for the Euclidean distance (l_2-norm).

As one can observe in Table 3.3, IGMM-based achieves similar performance as CDSA-based algorithm when Minkowski distance of order $p = 1.5$ is used. However, with Euclidean distance it performs better. We believe that, this is due to the fact that we used the average similarity of each cluster to accept a new incoming membership. As can be seen, IGMM-based algorithm uses almost the same similarity thresholds with both distance metrics, whereas, CDSA-based has different similarity thresholds. This is explained again by the fact that we made use of the centroid of cluster to accept a new member.

3.5 Conclusion

Throughout this chapter we have analyzed a framework for clustering mobile users, when they are in the same place, by exploiting the similarity of their measured ambient radio signals. It is designed to operate in real-time and in a centralized manner. We have shown that by using a nonparametric Bayesian method called infinite Gaussian mixture model (IGMM) with a modified version of Gibbs sampler, the proposed algorithm can accurately co-locate mobile users.

We carried out numerical and experimental analysis and showed that it can effectively detecting and clustering group of co-localized mobile users. We also conducted a comparative study where it has been shown that the proposed framework can achieve a better clustering accuracy than it counterpart community detection-based clustering. When users are walking together, as part of the same group, however the same approach can not be adopted. We discuss this issue is the next chapter and proposed a novel method to solve this problem.

The framework presented in this chapter is specially conceived for detecting co-located mobile users using ambient Wi-Fi radio signals, however it can be easily adapted to other radio signals.

Chapter 4

Discovering Co-Located Walking Groups of People Using iBeacon Technology

4.1 Introduction

In the previous chapter, we have designed and evaluated a framework mainly to cluster mobile users when they are in the same place. We have shown that it is able to identify clusters of mobile users based on the similarity of their measured Wi-Fi radio signals when people are in the same place. However, the same model can not be adopted when users are walking together, as part of the same group, using Wi-Fi hotspot radio signals. This is because the measured radio signals do not fit the same distribution [66]. Moreover, the protocols implemented by APs are conceived for faster access rather than proximity-based services.

Therefore, we extend the framework designed in the previous chapter by proposing a novel method for clustering people walking together, as part of the same group, in wireless networks. This newly devised method is based on the edge betweenness techniques presented in [78] by Girvan *et al.*, which is a generalization of the centrality betweenness algorithm proposed by Freeman in [79]. It is formalized as a graph network [80] in which each mobile user is represented by a vertex, and the connection strength between pairwise users is expressed by an undirected weighted edge.

We propose and derive a co-location algorithm based on edge betweenness techniques because it has some attractive properties. That is, with this technique, mobile users who do not discover one another through the Bluetooth discovering process will never be clustered together; by applying the edge betweenness techniques in conjunction with the average path length, the number of co-located groups of mobile users is automatically inferred from the input data, contrary to the parametric methods that need to be specified how many clusters to find in the input data; it needs only to know the connection strength between pairwise users to infer their clusters.

A graph network is constructed with information collected from all nearby Bluetooth low energy (BLE) [32, 81, 82] devices (e.g., iBeacon devices [33]), and the collected information is fed thereafter into the algorithm. To get information on walking group of users, we leverage the emerging and increasingly widely available BLE, owing to its very low cost, low power consumption, easy to deploy, and relatively long range. Moreover, the iBeacon devices that implement the BLE protocols are mainly designed for proximity-based services, which makes them the first choice to be considered in this kind of applications.

4.2 Problem Statements

We commence, in this section, with the explanation of our system architecture. Environmental radio signals are extracted from APs and iBeacon devices and processed to be fed into the proposed algorithm. Then, the basic idea behind our proposals is explicated, following by a full description of our modified version of the edge betweenness techniques for clustering mobile walking groups of users.

4.2.1 System Model

Mobile devices that have been in close proximity to each other, for a certain amount of time, detect one another for several times. They also experience similar radio signals from their environmental Wi-Fi hotspot [30]. Hence, our principal objective is to detect these proximity closely neighboring devices and cluster them into the same group.

In Figure 4-1, we present an example network of our co-located mobile devices. In this figure, there are several mobile user equipments (MUEs) and iBeacon devices organized in two groups: **Group 1** and **Group 2**. We consider the situation in which a person is equipped with, in addition to a mobile device (e.g., an iPhone), an iBeacon that broadcasts its radio signals. The radio signals broadcast by an iBeacon are received by all nearby mobile devices apart from iBeacons. This is explained by the fact that iBeacons only have the ability to broadcast their radio signals.

The reasons for this architecture are twofold: first, as in Chapter 3, we envisage to exploit the captured ambient radio signals to co-localize users who have been spending time together in the same place (in a room, for instance); and second, we aim at utilizing the emerging BLE technologies to cluster users while they are walking together for the same amount of time. Nevertheless, this latter approach can also be applied to co-localize users that are in the same place. It should be highlighted the fact that an iPhone, for example, can also be used as an iBeacon [83], therefore no need for an additional device.

Mobile users in the same group are expected to experience similar radio signals from their nearest access points (APs), and the radio signals broadcast by all nearby iBeacon devices. Thus, on a periodical basis, they will report to the nearest base station (BS) their measured radio signals from APs, and a matrix in which each entry is a distance from pair of users computed with the signals detected from iBeacon devices. The distance between pair of users are used as the measure of the strength of the link between them, and it is calculated using RSSI signals broadcast by each iBeacon. The higher the connection strength between users is, the closer they are to one another. Upon receipt, the BS will in turn transmit the reported information to the co-location server. The co-location server will perform the task of group formation detection from the received data set, and will inform back the mobile users, through an application installed on their devices, about their belonging group.

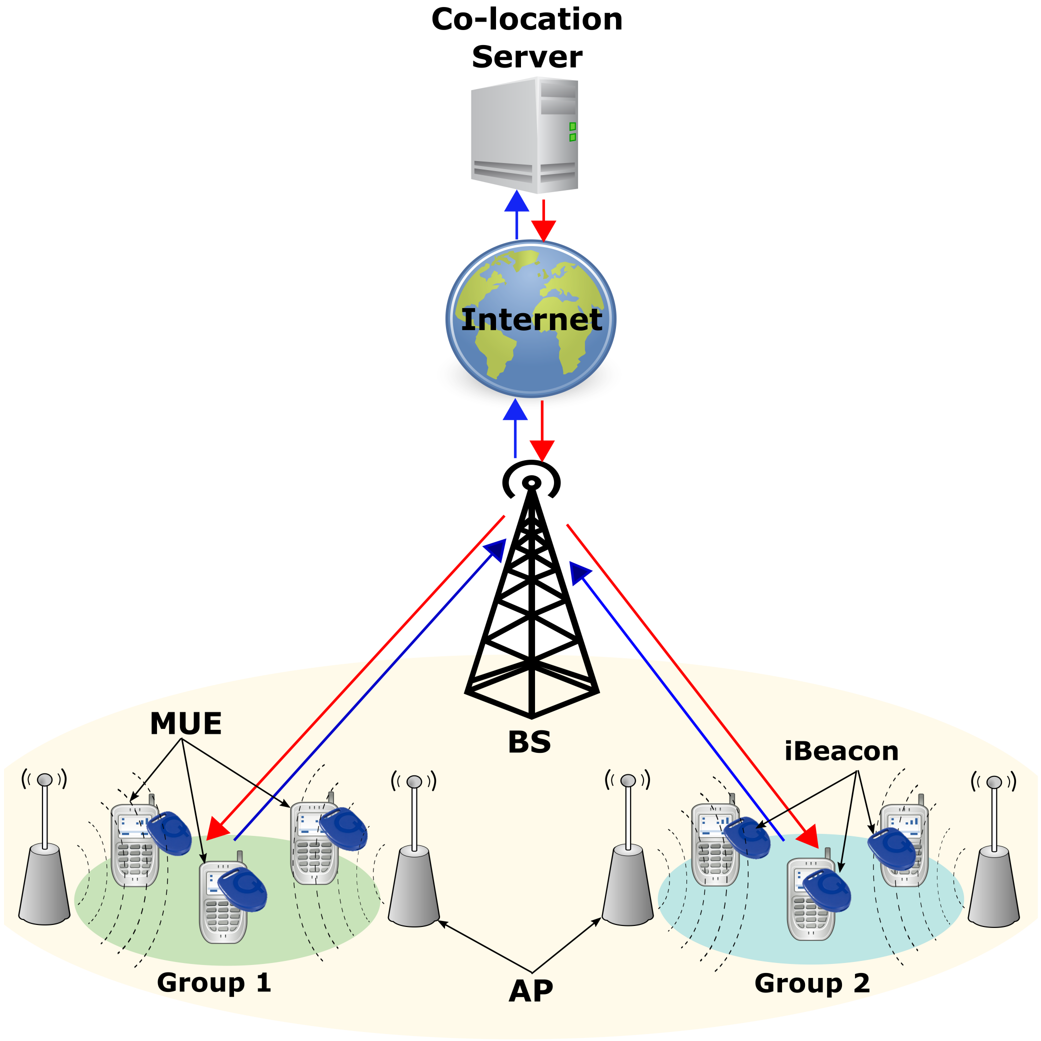


Figure 4-1: An example network architecture of co-localized mobile user equipments. The blue arrows indicate the transmission of the collected radio signals to the co-location server. The red arrows represent information of co-localized mobile user equipments sent by the co-location server. The small iBeacons here serve as the peripherals for MUEs.

In the likeness of the framework designed in Chapter 3, we propose to use Wi-Fi radio signals to cluster users that have been together for a certain amount of time in the same place. This is because of their easy deployment and no extra cost, and their ability of working in both indoor and outdoor environments. On the other hand, we exploit the emerging BLE technologies to cluster walking groups of users together, because it has some advantages over its counterpart. That is, when a device receives signals from a nearby iBeacon, it knows the sender and can compute the distance from it with high degree of accuracy. Moreover, it offers a lot of more possibilities than existing wireless technologies, and it will be certainly a leading candidate to implement the future Internet of Things (IoT), as IoT requires low power communication to fulfill its potential [84].

4.2.2 Inferring Co-located Group of Users

In this subsection, we will explain our algorithm based on the edge betweenness method to cluster walking groups of users in wireless networks.

As mentioned before, we aim at extending the capabilities of the framework proposed in Chapter 3 by giving it now the ability to cluster groups of users even though they are walking together. With this objective in mind, we provide design feature to enhance that framework. This new feature is based on the method proposed by Girvan *et al.* in [78] for detecting group of vertices in graph. Therefore, finding a distinct group of vertices within a graph is a key function to identify proximity nodes into the network, and, by extension, finding co-localized groups of people.

Hence, the problem of finding co-located walking groups of people is formulated thereby as a group discovery process in graph, in which each mobile user is represented by a vertex and the connection strengths among them are expressed by weighted edges. As stated before, these weighted edges are computed using the radio signals received from each iBeacon device on each mobile user. To find such groups within a network, the algorithm exploits the idea that, edges connecting inter-cluster (different clusters) have the highest betweenness scores than intra-cluster (same cluster) edges. Therefore, proceeding with the removal of these edges, the network will be split into tightly connected subgroups.

In the light of this observation, we design our co-localization system as a network of undirected graph, defined as

$$G = (V, E), \tag{4.1}$$

where V is the set of all vertices corresponding to the mobile users, and E is the set

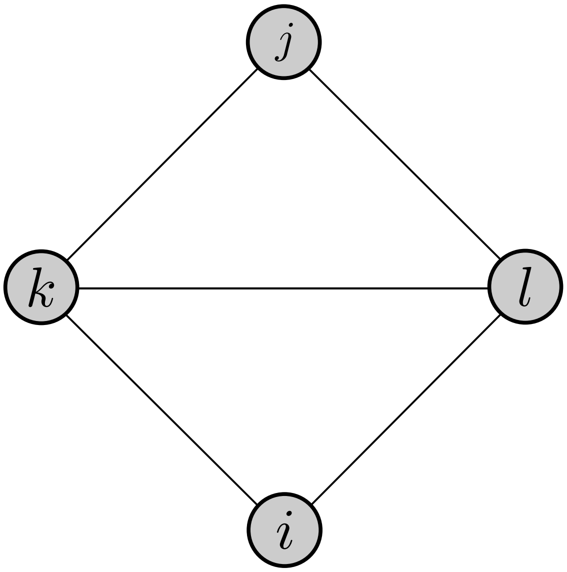


Figure 4-2: An example of an undirected graph with four vertices and five edges.

of all weighted edges representing the connection strength between pair of users in the network [80]. The number of vertices and edges in the network are denoted by N and M, respectively.

The central idea of group detection process presented in [78] is based on the vertex betweenness as a measure of the centrality and influence of a vertex, with respect to information flow, within the network, proposed by Freeman in [79]. In [79], Freeman defines the partial betweenness, $\sigma_{ij}(k)$, of a vertex k with respect to a pair of vertices, i and j, in graph G as follows. In case when the vertices i and j are not reachable, i.e., k is not between them, $\sigma_{ij}(k) = 0$. When the vertices i and j are reachable, using the shortest path length, there may exist multiple paths with the same length connecting these two vertices. Thus, the probability of using one of these paths is $\frac{1}{p_{ij}}$, where p_{ij} is the number of the shortest paths connecting vertices i and j. Therefore, the probability of vertex k falls on any one of the shortest path between vertices i and j is given by

$$\sigma_{ij}(k) = \frac{p_{ij}(k)}{p_{ij}}, \tag{4.2}$$

where $p_{ij}(k)$ is the number of shortest path length between vertices i and j containing the vertex k. An illustration of this network analysis technique is presented in

Figure 4-2. Hence, the overall measure of betweenness centrality of a vertex $k \in V$ is defined in [79] as

$$C_B(k) = \sum_{i \in V}^{N} \sum_{j \in V \setminus \{i\}}^{N} \sigma_{ij}(k). \tag{4.3}$$

Therefore, the Freeman's betweenness centrality is generalized to the edge betweenness as the number of shortest paths between pairs of vertices, i and j, that contain it. It has been shown that by successively removing edges with the highest betweenness, the network can be split up into many separate sub-networks [78]. This is explained by the fact that most of real world networks, arising in nature and technology, are characterized by a very short average path length within themselves. Thus, the concept of small-world phenomenon [31] [85] is introduced where people are connected with one another through a very short path.

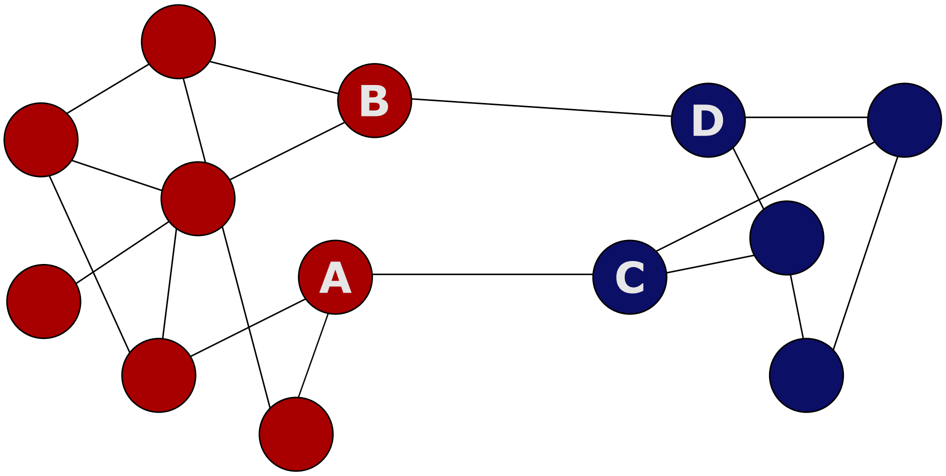


Figure 4-3: An example of a graph network with two sub-networks.

Figure 4-3 depicts an example graph network where vertices are connected between them through edges to form a population structure. In fact, this figure exhibits two groups of vertices (red and blue) connected between them by two edges: $\{A, C\}$ and $\{B, D\}$. The heart of our aim is to be able to partition this network into distinct sub-networks where each one of them is regarded as a potential co-located group of walking users. In this figure, each round circle (red or blue) designates a mobile user in the wireless network. The black lines between pair of round circles indicate the connection strength between users. That is, these users detect and connect with each other, otherwise no connection between them.

Based on the edge betweenness techniques, the algorithm finds the edge with the highest betweenness score and removes it from the network. As the algorithm repeatedly searches for these edges and removes them from the network, we will end up with the entire network partitioned into several sub-networks.

More specifically, by way of example, let us take Figure 4-3 and suppose the following. In the first iteration, the edge {A, C} is found to have the highest betweenness score. Consequently, this edge will be removed from the network. Then, in the next iteration, the edge {B, D} will be found with the maximum betweenness score and will be in turn removed from the network. At this point, we have divided the entire network into two sub-networks. If we keep running the algorithm, we will end up with this entire network split into its number of elements.

4.2.3 Modified Edge Betweenness Algorithm

The algorithm presented in the previous subsection actually divides a given network into a K sub-networks. However, to efficiently apply this algorithm on the issue at hand, i.e., co-location problem, we need to change its behavior. To do so, we introduce a notion of average path length, APL, into it. That is, each time the algorithm finds a new cluster in the network, as explained earlier, we compute the average path length of that cluster. If the computed average path length is less than or equal to a predefined similarity threshold Δ, i.e., $APL \leq \Delta$, we consider that a new cluster has been discovered and proceed with the output followed by the removal of all the elements of this newly found cluster in the network. The similarity threshold Δ defines how near two or more mobile users should be regarded as potentially co-located. More on this threshold Δ will be explained later.

In this work, we propose to use the average shortest path length to cluster mobile users into the same group because, as highlighted earlier, most real world groups are characterized by the shortest path length. Thus, we define the average path length,

APL, [86] in terms of the shortest path lengths as follows

$$APL = \frac{1}{N(N-1)} \sum_{i \neq j} d_G(i, j), \tag{4.4}$$

where $d_G(i, j)$ is the shortest path length between each pair of vertices, i and j, regarding the communication path separating them physically. N is the number of vertices in graph G.

In our implementation, we opt for the simplest, rapid, and efficient way to measure the edge betweenness, which is based on the shortest paths. However, other measures can be adopted which fit well with the application requirements [87]. It should be noted that there are several optimized versions of the edge betweenness algorithm, which make it a robust technique [87, 88].

It should be noticed that, even though the algorithm presented so far is able to successfully discover potentially co-located group of people in the wireless network, it does not take into account an important characteristic of the co-localization systems, i.e., how long people have to be together in order to be clustered into the same group. This issue is explained in the next subsection.

Algorithm 2 Edge betweenness-based co-location algorithm

```
1: Input : Data set from $N$ users, and pre-set threshold $\Delta$.
2: Output: Users co-localized into $K$ clusters.
3: Initialize: Set all users into the same cluster, $K = 1$.
4: Compute the edge betweenness score for all edges in the network as in (4.3).
5: Remove the edge with the highest betweenness.
6: Compute the average path length, $APL$, of each existing cluster as in (4.4).
7: Output identified cluster, and remove it from the network.
8: Recompute the edge betweenness.
9: Repeat step 5.
```

In the algorithm 2 we show the necessary steps of the modified version of the edge betweenness method to cluster walking groups of users.

4.2.4 Duration and Frequency of Encounters

The algorithm proposed in this work does effectively detect and cluster co-located group of people in the network. However, as it is mentioned in the previous subsection, it does not take into account the time that people spend together, which is an important criterion of the co-location systems. Therefore, we propose to cluster mobile users not only based on the strength of their connections but also on the time duration and frequency of their meetings. In fact, one can use only time duration and frequency of meetings to cluster mobile users into the same group. However, such an approach fails when it comes to assessing how close people are to one another.

Toward this end, we proceed as follows, when a device receives a signal from a nearby iBeacon, it registers the time of the reception and sets the frequency of meeting to one. Next time this device receives the signals from the same iBeacon, it just increases the duration and the frequency of meeting, as it has already received the signals from the same iBeacon for that period of time, Δt. Δt is defined as the minimum period of time that is required to the users to be together in order to consider them as co-located.

Since we are dealing with walking group of people, certainly, they will encounter many other people during the predefined period of time. Therefore, it is crucial that the time traces of their measurements should be compared to infer the duration of their interaction. Consequently, helping filter out these one-time encounters that will not make part of the same cluster. The analysis carried out on the measured data signals reveals that, when people are walking together for long time, the number of times they detect one another is much longer than when a user just passes by them. Therefore, the values of these two parameters should be tuned in order to achieve a desired accuracy in line with the application requirements.

4.2.5 Co-location Scheme Detection

Aiming at detecting and clustering co-located group of people not only when they are in the same place but also when they are walking together, we propose the

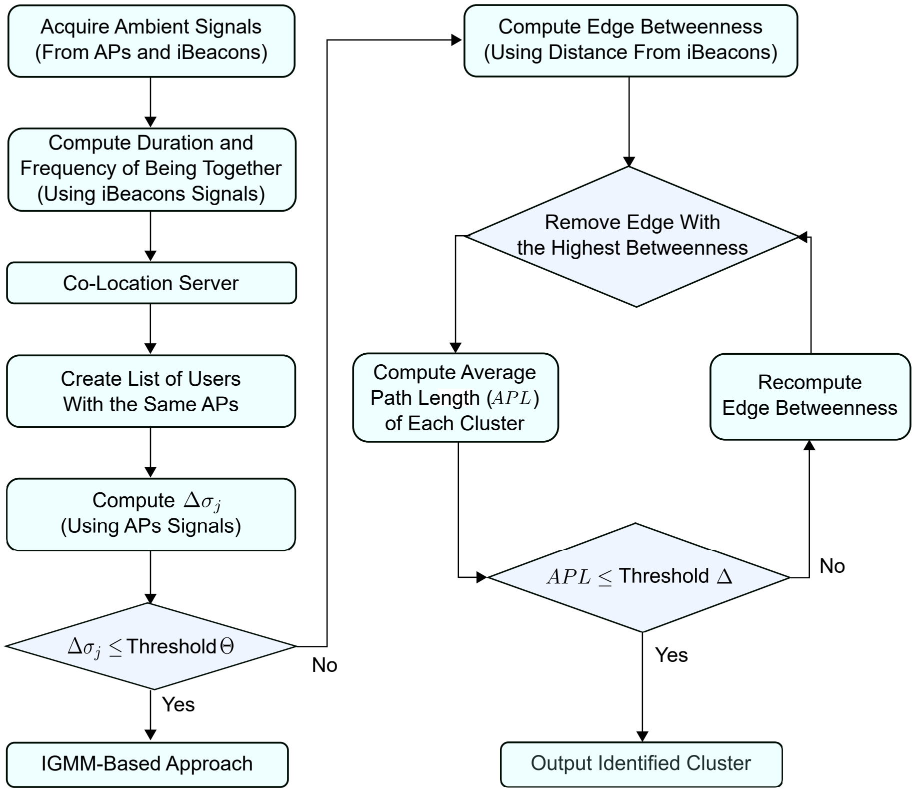

Acquire Ambient Signals (From APs and iBeacons)
Compute Duration and Frequency of Being Together (Using iBeacons Signals)
Co-Location Server
Create List of Users With the Same APs
Compute $\Delta\sigma_j$ (Using APs Signals)
$\Delta\sigma_j \leq$ Threshold Θ
Yes
No
IGMM-Based Approach
Compute Edge Betweenness (Using Distance From iBeacons)
Remove Edge With the Highest Betweenness
Compute Average Path Length (APL) of Each Cluster
Recompute Edge Betweenness
$APL \leq$ Threshold Δ
No
Yes
Output Identified Cluster

Figure 4-4: Algorithm flow chart of our co-location system.

following scheme (see Figure 4-4). Ambient radio signals (from APs and iBeacons) are sensed for a period of time, Δt. Then, the mobile device computes the distance to each one of the nearby users, the duration, and the frequency of being together, using data signals from iBeacons. If the computed duration and frequency of being together satisfy the predefined criteria, as explained earlier, the collected data signals from both APs and iBeacons are sent to the co-location server to be processed.

Upon receiving the data signals, the co-location server will create distinct lists of users with the data signals from the same APs, and a sparse symmetric matrix in which each entry is a distance between a pair of users. We use this matrix to cluster walking group of users. For each user a $\Delta\sigma_j$ is calculated in order to determine whether he or she is walking or remaining in the same place, for the specified period of time, Δt. To compute the value of $\Delta\sigma_j$, we use radio signals collected from APs, and to determine whether a user is walking or not, we compare the value of $\Delta\sigma_j$ with a threshold denoted by Θ. Note that, in Subsection 3.4.2 on page 54, we have defined a mathematical model to compute $\Delta\sigma_j$ and discussed how to choose the value of the threshold Θ.

When the proposed model determines that users are staying in the same place, we use the algorithm proposed in Chapter 3 to cluster them into the same group. Otherwise, we apply our newly proposed scheme. That is, we apply the edge betweenness-based algorithm on the transmitted data signals to cluster group of walking users into the same group. To this end, we start with the computation of the edge betweenness score. Then, we identify the edge with the highest betweenness score and remove it from the network. Next, we calculate the average path length, APL, of each discovered cluster. If the computed APL is less than or equal to a predefined similarity threshold Δ, ($APL \leq \Delta$), we consider that a new cluster has been discovered and output its elements followed by the removal of all the elements that belong to it. For the remaining elements in the data set, we recompute the edge betweenness in order to find the edge with the highest score and remove it from the network. This procedure is repeated until the algorithm discovers all the existing groups in the data set.

It should be noticed that the proposed scheme enjoys several advantages. In fact, mobile users who experience different AP radio signals and do not detect each other will never be clustered together. Another one is that by introducing the similarity threshold Δ, in our clustering process, we were able to discover all existing clusters. The proposed approach is also robust in dealing with varying the number of clusters and users over time in the network.

4.3 Numerical Results

Our co-localization of walking groups of users' algorithm is first assessed numerically, and then experimentally. In this section, we will present and discuss our numerical results.

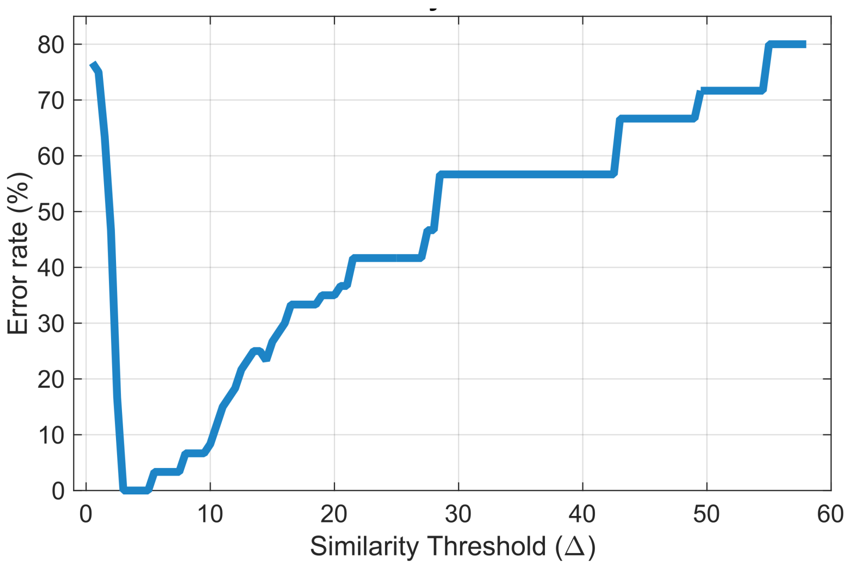


Figure 4-5: Effect of the similarity threshold Δ on users co-location. The error rate decreases until it attains its lowest level and then increases to its highest level.

4.3.1 Setup

For the purpose of evaluating the performance, in terms of the clustering accuracy of our proposals, a computer-generated graph similar to that one presented in Figure 4-3 is fed into the algorithm. The generated graph is a random modular graph with 60 vertices divided into 14 groups of vertices (each vertex represents a mobile user). Each group contains a different number of vertices.

We adopted the following procedures to place edges between vertices. If the distance between pair of vertices is less than or equal to 25 meters, we consider they are detecting each other. Thus, an edge is placed between them. The length of this edge represents the strength of their connection. In the case where the distance is greater than 25 meters, no edge is placed between them, which means no connection between pair of users. With this approach, a graph is generated which simulates the network of mobile users with known groups of vertices but in which its fundamental aspects keep its randomness.

4.3.2 Similarity Threshold Δ

In this subsection, we describe the steps undertaken to determine the optimum value of the similarity threshold Δ, to co-localize walking groups of mobile users.

With the firm purpose to get the best value possible for our threshold Δ, we perform an offline analysis. To do so, we define an interval in which the search will be operated. Thus, we perform a search over this interval with the step size of 0.5 meter. The optimum value of the threshold Δ is computed in function of the number of the users correctly clustered at each step size. Figure 4-5 depicts the effect of the similarity threshold Δ on the co-location accuracy over the defined interval. As it can be seen, from this figure, as the value of the threshold Δ increases, the error rate decreases until it attains its minimum percentage value, i.e., zero percent, and after that it increases gradually to its highest level. Therefore, we take the value of the threshold Δ where its effect on co-location accuracy is the best, i.e., where its error rate is of zero percent.

It should be noticed that, even though in this simulation, the accuracy of the proposed algorithm reaches error-free, it should not be always the case. Indeed, in a more realistic situation, more underlying parameters should be taken into account, which may have different effects on the accuracy. We will have more discussion on these matters later on in this chapter.

From this analysis, we observed that more than one value of the threshold Δ can be chosen in order to achieve the highest accuracy possible. In fact, this suggestion is in perfect tune with our setup. We want also to emphasize the fact that the value of this threshold is chosen in accordance with the application requirements. Here, we take the one that gives us the higher accuracy possible for our setup, i.e., the one that gets back our co-located group of users.

It is worth noting that when the value of the threshold Δ is chosen to be less than the optimum value, more number of clusters are found in the data set but with scanty number of users. In some cases, even singleton clusters are discovered. On the other hand, when the value of the threshold Δ is set higher than the optimum value, less

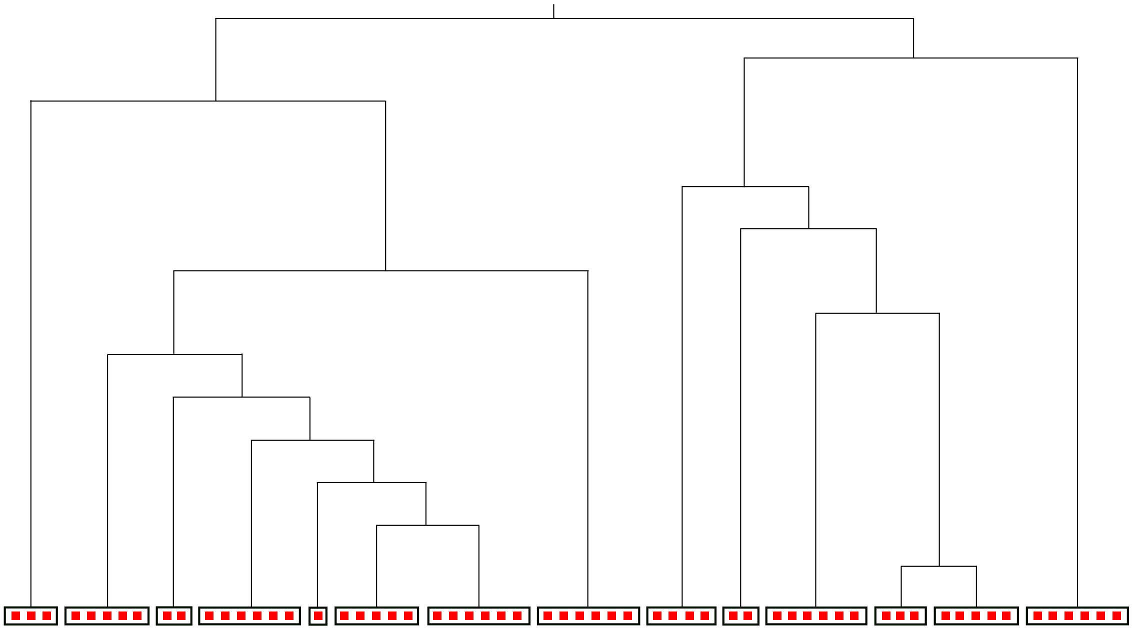


Figure 4-6: Numerical results of our co-localization system. Each red square dot represents a vertex (mobile user) in the graph network. The rectangles surrounding square dots indicate the detected co-located group of vertices. The algorithm reliably discovers all groups of vertices.

number of clusters are found in the data set. However, each one of these discovered clusters has an important number of users into it. Therefore, one should set the value of this threshold that best fits the application targeted [17].

4.3.3 Results

Figure 4-6 shows the obtained results in the form of a tree. Each red square dot at the bottom of the tree represents a vertex (a user), and the black rectangles surrounding them indicate detected groups of vertices in the graph. In this evaluation, and in accordance with our offline analysis in the previous subsection, the value of the similarity threshold Δ is set to 3.5 meters. That is, each time the algorithm partitions the graph into subgraphs, we test whether or not the newly found subgraphs satisfy our co-location criterion. If so, we output the vertices of these subgraphs as a new co-located group of people and proceed with the removal of its vertices from the network.

As it can be noticed, in this evaluation process, the algorithm correctly clusters all vertices into their respective groups. This result is justified by our earlier analysis on the similarity threshold Δ, in the previous subsection. That is, when the error rate is at its lowest level, the highest accuracy possible is achieved.

4.4 Experimental Setup and Results

In this section, we first describe our experimental setup to co-localize people walking together as part of the same group, and then we present and discuss the obtained results.

4.4.1 Experimental Setup

We carried out an experiment on a corridor of our department building, collecting iBeacon radio signals, to demonstrate the effectiveness of the proposed scheme. Thus, we evaluated the performance in terms of clustering accuracy of the designed framework with data set from this experiment.

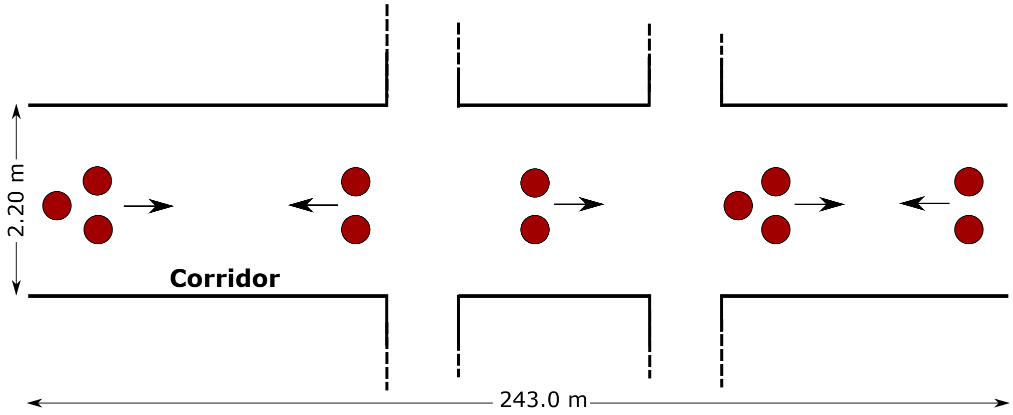
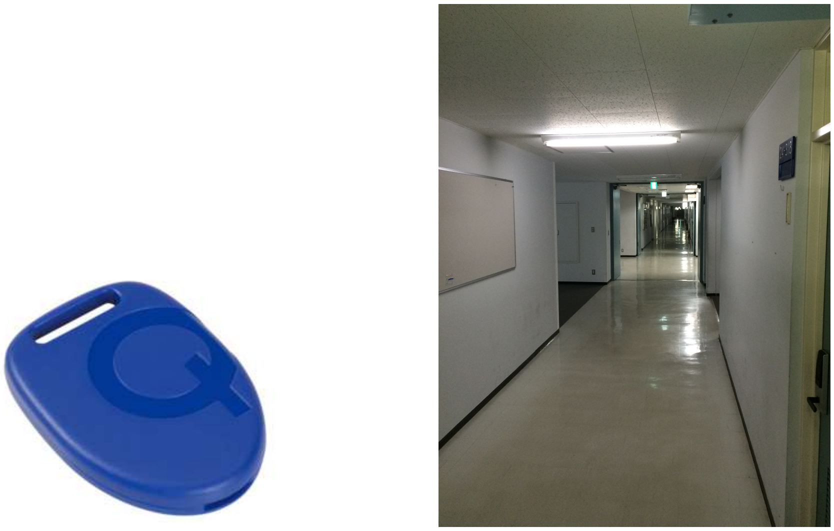


Figure 4-7: Corridor of a 3rd floor of a building where 12 users were walking during the experiment. Each red dot represents a walking user, and the arrows indicate the direction in which those groups of users were walking. We consider five walking groups of users in this experiment.

To this end, we developed a smartphone application, for both Android and iPhone OS devices, capable of collecting radio signals broadcast by all nearby iBeacon devices. We installed this application on smartphone of 12 students and equipped each one of them with an iBeacon device. Each iBeacon device is associated with a student's smartphone. Then, we demanded these students to walk in the corridor in groups of different sizes, in different directions, in a third floor of our department building during ten minutes. Thus, they can meet one another several times.

In Figure 4-7, we illustrate our configuration settings. Each red dot, in this figure, corresponds to a walking user, and the arrows indicate the directions in which they were walking in groups. There are five distinct groups of people. People in the same group were walking apart each other at a distance of around two meters. During the experiment, each group passed by each one several times. The size of this testing area was a 2.20 x 243.0 m^2 (see Figure 4-8 (b)).

The application installed on students' smartphone collects iBeacon universally unique identifier (UUID), date and time of received signals, and the received signal strength indicator (RSSI) from each iBeacon device. After that, all information is put together on a computer to be processed.



(a) Gimbal iBeacon [89]

(b) Corridor

Figure 4-8: Experimental setup. (a) Gimbal iBeacon device used in this experiment. (b) Corridor where we conducted the experiment.

4.4.2 iBeacons

An iBeacon is a low cost, low power consumption, and a 2.4 GHz radio transmitter using Bluetooth Smart [32]. It is also known as Bluetooth 4.0 low energy (BLE) device with one way transmitter capabilities to the receiver devices. iBeacons neither communicate with each other nor communicate with smartphones. Only a device with an application installed on it and specifically designed to detect the radio signals broadcast by iBeacons can do so. Its transmitted radio signals can be utilized to infer proximity devices as well as in providing context-aware services.

Moreover, the range of an iBeacon depends on manufacturer. For some manufacturers the range can be on the order of 70 meters, which is considered as standard. Whereas, long range iBeacons can reach hundreds of meters. The one we used in this work, its range in line of site is up to 50 meters, but the range decreases if there are some obstacles between an iBeacon and the devices that are supposed to detect it.

In this experiment, we use Gimbal proximity beacons series 10, measuring 40 x 28 x 5.5 millimeter [89], [90]. Figure 4-8 (a) shows an exemplar of them. More information about iBeacons can be found in [33].

4.4.3 Experimental Results

In this subsection, we present and discuss the obtained experimental results.

As mentioned earlier, the proposed scheme, which is based on the edge betweenness techniques, alone is not enough to state whether or not people are co-localized. Therefore, in our evaluation process we further consider that people are co-localized if they spend at least three minutes walking together (remember that, in tis experiment, we collect radio signal during ten minutes). In fact, these three minutes are the time required for a person to walk from one end to the other of the corridor where the experiment was conducted (see Figure 4-7). The period of time should be set large enough in order to ensure that people really pass time together. It is a tuning parameter. Here, we consider this period of time to be three minutes because, from our experiments, it seems a reasonable choice.

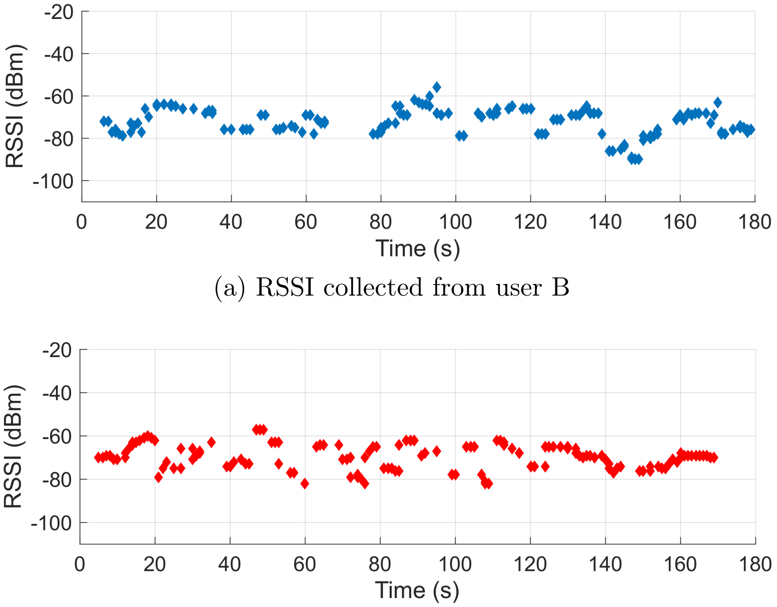
During this period of time, we noticed that the number of times that people in the same group detect each other is really high, compared with the number of times when a person just passes by, on the one hand. On the other hand, it also depends on the transmission interval of the signals configured in the iBeacon side. Therefore, the frequency of meeting should be set in accordance. It is also a tuning parameter. From our configuration, we noticed that users in the same group detect one another for more than 200 times. Thus, we set the frequency of meeting to this value.

Signal Measurements

As users were walking together at the same speed and keeping the same distance to one another, around two meters, we also observed that, the collected radio signals during this time period (three minutes) do not vary a lot. An illustration of this is given in Figure 4-9. In this figure, we plotted the collected radio signals by user A on two different users, B and C, when they were walking together, in the same group, during the experiment (users A, B, and C were walking in the same group). As can be seen, from this figure, the collected radio signals by user A on users B and C do not vary a lot over time. Therefore, we took the average of these measured data signals and computed the distance in signal space to each other. Thus, a sparse matrix of interactions is constructed in which each entry is a distance between pairwise users.

In the ideal case, a sparse symmetric matrix should be obtained from the collected radio signals, as we did in Section 4.3. However, a such approach here may be incorrect due to the fluctuation of the radio signals. Therefore, we constructed a sparse matrix with the distance computed directly from the data signals obtained from iBeacons on each user.

It should be pointed out that when users do not satisfy the two aforementioned criteria, i.e., the duration and the frequency of meeting, we do not consider a link between them. Thus, there is no connection between them for this predefined period of time. Therefore, they will not be clustered together in the same group.



(a) RSSI collected from user B

(b) RSSI collected from user C

Figure 4-9: RSSI values that user A measured from user B and C while they were walking. (a) RSSI from user B. (b) RSSI from user C. These users (A, B, and C) were walking in the same group. Here, we omit the RSSI values of user A because we consider its own values as zero in our setup.

Obtained Results

During our experiments, as stated earlier, we demanded the mobile users in the same group to walk apart from one another at a maximum distance of two meters. They keep walking with this distance during ten minutes, i.e., the period of time Δt required to the mobile users to be together in order to consider them as co-located. In our implementation, however, we took the data signals collected during three minute, as explained earlier. From this, we consider that two mobile users are co-localized (i.e., forming a cluster) if the distance between them is less than or equal to two meters and they pass a certain amount of time together ($\Delta t = 3$ min).

For the inference of the optimum value of the similarity threshold Δ, here again we follow our standard procedure. That is, we compute its values in function of the number of users correctly clustered at each step size. The correct clustering of mobile users at each step size is defined as the number of users clustered together by the proposed algorithm that is in accordance with our experimental setup. That is, if

User A and *User B* are inferred by the proposed algorithm to be in the same walking group and it turns out that, in our experiments, these users were actually walking together, we consider this inference as a correct clustering by the algorithm.

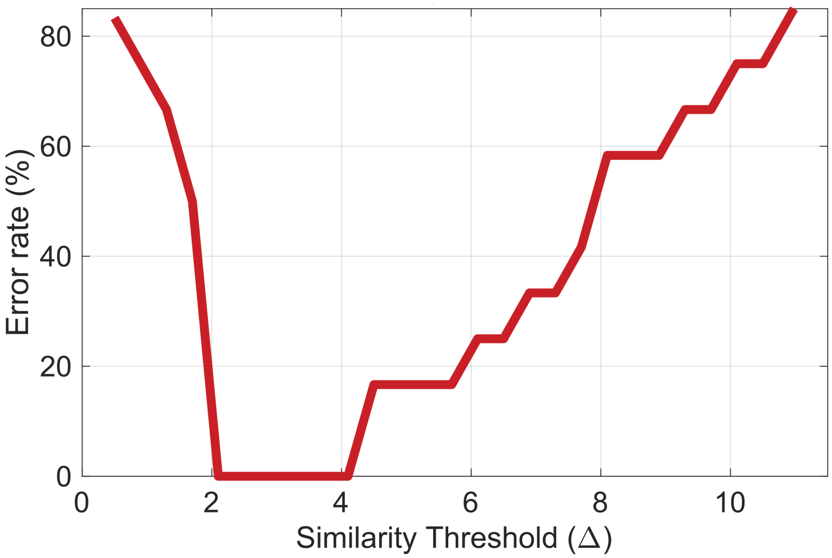


Figure 4-10: Effect of the similarity threshold Δ on co-localized walking groups of users from the experiments.

To set the optimum value of the similarity threshold Δ for our configuration setup, we also performed an offline analysis, as we did in Subsection 4.3.2. Figure 4-10 depicts the impact of this threshold on co-location accuracy. Its values are computed in function of the number of users correctly clustered at each step size. Here again, the similar analysis, as we did in Subsection 4.3.2, is carried out to choose the optimum value of this threshold. Thus, we take the value of the threshold Δ where its effect on co-location accuracy is the best, i.e., we set its value to 3 meters. However, according to the conducted analysis, other values can also be chosen, as it can be seen in Figure 4-10.

Figure 4-11 shows the obtained result, after feeding the observed data set into the algorithm, in the form of a tree. Each one of the red circle at the bottom of the tree corresponds to a walking user, and each black rectangle surrounding red circles indicates walking users as part of the same group discovered by the algorithm.

Note that, from this experiment, the proposed algorithm successfully discovered

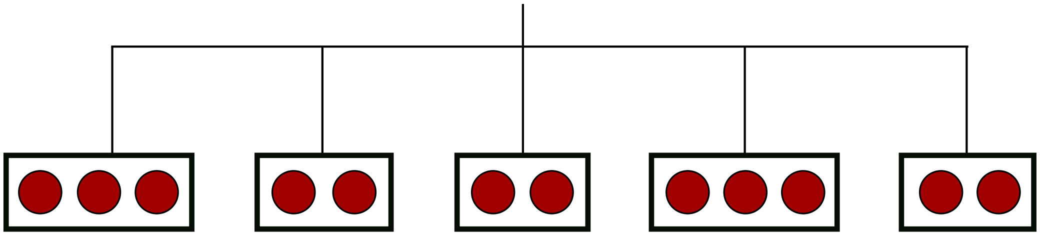


Figure 4-11: Experimental result. Each red circle at the bottom of the tree represents a user walking in the corridor during the experiment. The black rectangles surrounding red circles indicate the discovered group of users walking in the same group.

the cluster of all users, and the obtained result is in tune with our offline analysis on the similarity threshold. It should also be emphasized the fact that the value of the threshold changes in different situations. Indeed, different environments (indoor and outdoor) may have different effect on the choice of the threshold. Its value also varies respecting application requirements. More underlying situations need to be investigated, in the evaluation process, such as people are walking together sometimes and then they separate into two or more walking groups. It would be very interesting to see how the algorithm behaves in this kind of situation. We did not carry out experiments to show how the value of the threshold varies with different environments and how the accuracy of the algorithm varies in different situations. However, it can be addressed in the future research.

4.5 Conclusion

This chapter focused on analyzing a new method able to cluster people when they are walking together as part of the same group. It is especially designed as an extension of the method proposed in Chapter 3. Therefore, the proposed schemes, discussed throughout this thesis, are not only able to cluster mobile users when they remain in the same place but also when they are walking together as part of the same group, for a predefined period of time.

We proposed to exploit the environmental radio signals when users are in the same place, and the strength of their connections over time to cluster them when they are walking together. To overcome this latter challenge, we inferred walking groups of users by analyzing two key network properties, i.e., the edge betweenness and the average shortest path length among all pairs of users in the wireless networks. The connection strength between pairs of mobile users is constructed with the radio signals broadcast by Bluetooth low energy device (e.g., iBeacon).

We first evaluated the proposed algorithm, in this chapter, with computer-generated data set. Then, we carried out experiment to demonstrate its performance, in terms of clustering accuracy, with data set from real-world settings. In both cases, the proposed algorithm correctly identified and co-localized all groups of mobile walking users.

Chapter 5

Conclusions and Directions for Future Work

5.1 Conclusions

The core objective of this thesis was to show how to cluster mobile users, for the purposes of proximity-based services, when they are in the same place or are walking together as part of the same group, during the same time interval. In Chapter 1, we gave the background and the motivations for doing research in this topic. We highlighted some real-world applications and the benefits that this research brings to our ever connected society. The contributions and the relationship among techniques and chapters of this dissertation are also discussed on this chapter. A review on related works is presented in Chapter 2.

In Chapter 3, we derived a method that clusters mobile users based on the similarity of their measured environmental radio signals. It is conceived for mobile users that have been in the same place. To this end, we applied a nonparametric Bayesian method called infinite Gaussian mixture model to model the observed data signals and used Gibbs sampling technique to classify these observed radio signals while users are in the same place. A modified version of Gibbs sampling method is proposed with a similarity threshold to best fit the application requirements. The proposed framework operates in real-time to infer co-located mobile users. Its design allows the co-location server to manage all the aspects of the formation of the user groups in a centralized manner. We first analyzed the proposed algorithm numerically. Then, we carried out experiments to demonstrate its performance, in terms of clustering accuracy, with data signals from a real-world setting. We have also presented a comparative analysis on its performance against the state-of-the-art clustering method. Results on experiments favor our approach.

With the aim at improving the framework designed in Chapter 3, we investigated in Chapter 4 a new way of clustering method when users are walking together as part of the same group. In this case, we exploited the distance between pairwise users to construct a matrix of interactions in which each entry represents a connection strength between a pair of mobile users. This matrix of interactions is constructed with the radio signals transmitted by Bluetooth low energy (BLE) devices (e.g., iBeacon device). The co-located mobile users are then inferred based on the analysis of two key network properties, i.e., the edge betweenness techniques and the average path length among all pairs of users in the network. Then, we derived a modified version the edge betweenness techniques in order to fit the requirements of the co-location systems. Finally, we evaluated the proposed algorithm with computer-generated and experimental data set. In both cases, we demonstrated that the proposed framework is robust in inferring co-located mobile users and it can even achieve one hundred percent of accuracy in some situations. It should be emphasized that even though this algorithm is mainly designed for walking groups of people, it can also be applied when users remain in the same place.

5.2 Directions for Future Work

Throughout this thesis, two new robust frameworks have been proposed, designed, and evaluated for the purpose of proximity-based services. We have shown, through numerical and experimental analysis, that they are indeed effective in inferring co-localized group of mobile users. However, our focus was only on physical proximity entities. Many other factors can influence the cluster of people such as their social attributes, for instance. In the following, we will highlight some important related issues that need to be addressed in order to fulfill the potential of proximity-based services.

The models presented in this thesis can thus be extended and improved in several deferent ways, as for examples:

- It is well known that in many natural groups, there is a hierarchical structure. Therefore, further analysis in each discovered group of mobile users may uncover more important information such as group dynamics, role of each mobile user, etc. It would be very interesting, for example, to know in each group of people who is the leader, as in many natural groups it happens to be. Thus, the leader, who has an overall knowledge of his group, can be in charge of that group and be able to predict and prevent unwanted situations within his respective group and take command, if it is necessary.

- Throughout this thesis, we analyze the case where mobile users are physically close to one another. That is, we only consider their physical proximity. However, the context in which they are is another important issue. Normally, people form a group with an objective, a purpose, for example for data exchange. Another important point is that their social relationships can also influence the group formation. Therefore, their physical distance, their purposes, and their social attributes should also be object of analysis as well in the clustering process.

- The best value of our co-location criterion, i.e., the value of the similarity threshold, is determined in an off-line analysis from the data signals obtained through our experiments. However, different environments (indoors and outdoors), buildings, materials, the number of people, their activities, and so on, may have different impact on the choice of the value of the similarity threshold. In addition, the value of the similarity threshold may vary with application requirements. Further research should be carried out in order to show how the value of the similarity threshold varies with these aforementioned conditions.

- In our first proposal, we utilized Wi-Fi radio signals transmitted at 2.4 GHz, from different access points, to cluster mobile users, when they are in the same place. However, the proposed scheme can also be analyzed with dual band Wi-Fi radio signals. The fact is that many Wi-Fi networks suffer from increase wireless interference and degrade performance due to the predominance of 2.4 GHz consumer gadgets. Therefore, utilizing 5.0 GHz on a dual band Wi-Fi router can help circumvent these issues. Moreover, power delay profile measurements can also be utilized as radio frequency fingerprints. It fully characterizes the multipath channel features, which are widely applied in the localization systems, as it represents a more unique location signature than received signal strength.
- It will be very interesting to see the implementation of these algorithms running in real life.

Bibliography

[1] A. Didwania and Z. Narmawala, “A comparative study of various community detection algorithms in the mobile social network,” in *Proc. 5th Nirma University International Conference on Engineering (NUiCONE)*, Nov. 2015, pp. 1–6.

[2] N. Kayastha, D. Niyato, P. Wang, and E. Hossain, “Applications, architectures, and protocol design issues for mobile social networks: A survey,” *Proc. IEEE*, vol. 99, no. 12, pp. 2130–2158, Dec. 2011.

[3] X. Liang, R. Lu, L. Chen, X. Lin, and X. S. Shen, “PEC: A privacy-preserving emergency call scheme for mobile healthcare social networks,” *J. commun. and netw.*, vol. 13, no. 2, Apr. 2011.

[4] Z. Lu, X. Chen, Z. Dong, Z. Zhao, and X. Zhang, “A prototype of reflection pulse oximeter designed for mobile healthcare,” *IEEE J. Biomedical and Health Informatics*, vol. 20, no. 5, pp. 1309–1320, Sep. 2016.

[5] V. Goverdovsky, D. Looney, P. Kidmose, C. Papavassiliou, and D. P. Mandic, “Co-located multimodal sensing: A next generation solution for wearable health,” *IEEE Sensors Journal*, vol. 15, no. 1, pp. 138–145, Jan. 2015.

[6] K. Farrahi, R. Emonet, and A. Ferscha, “Extraction of latent patterns and contexts from social honest signals using hierarchical dirichlet processes,” in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2013, pp. 47–55.

[7] T. Nguyen, S. Gupta, S. Venkatesh, and D. Phung, “Continuous discovery of co-location contexts from bluetooth data,” *Pervasive Mobile Comput.*,

vol. 16, Part B, pp. 286–304, Jan. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119214001941

[8] S. Chung and I. Rhee, "An unobtrusive interaction interface for multiple co-located mobile devices," in *Proc. IEEE 11th Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, Oct. 2015, pp. 683–690.

[9] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proc. 2nd ACM/IEEE Int. Workshop Mobility Evolving Internet Architecture (MobiArch)*, no. 7. New York, NY, USA: ACM, Aug. 2007, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/1366919.1366929

[10] D. Quercia, J. Ellis, and L. Capra, "Using mobile phones to nurture social networks," *IEEE Pervasive Computing*, vol. 9, no. 3, pp. 12–20, Jul. 2010.

[11] K. Tsubouchi, O. Saisho, J. Sato, S. Araki, and M. Shimosaka, "Fine-grained social relationship extraction from real activity data under coarse supervision," in *Proc. ACM International Symposium on Wearable Computers*, ser. ISWC '15, 2015, pp. 183–187. [Online]. Available: http://doi.acm.org/10.1145/2802083.2808402

[12] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 414–454, First Quarter 2014.

[13] L. Cheng, C. Wu, Y. Zhang, H. Wu, M. Li, and C. Maple, "A survey of localization in wireless sensor network," *Int. J. Distrib. Sens. Netw.*, vol. 2012, pp. 1–12, 2012.

[14] S. Gezici, "A survey on wireless position estimation," *Wirel. Pers. Commun.*, vol. 44, no. 3, pp. 263–282, Feb. 2008.

[15] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning

techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[16] A. J. Weiss, "Direct position determination of narrowband radio frequency transmitters," *IEEE Signal Processing Letters*, vol. 11, no. 5, pp. 513–516, May 2004.

[17] 3GPP, "3rd generation partnership project; technical specification group services and system aspects; Feasibility study for proximity services (ProSe); Release 12," Sophia Antipolis Cedex, France, 3GPP TR 22.803, Tech. Rep., Jun. 2013. [Online]. Available: http://www.3gpp.org/

[18] L. Xiao, Q. Yan, W. Lou, G. Chen, and Y. T. Hou, "Proximity-based security techinques for mobile users in wireless networks," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 12, pp. 2089–2100, Dec. 2013.

[19] Y. Zheng, M. Li, W. Lou, and T. Hou, "Location based handshake and private proximity test with location tags," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1–1, 2015.

[20] H. P. Li, H. Hu, and J. Xu, "Nearby friend alert: Location anonymity in mobile geosocial networks," *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 62–70, Oct. 2013.

[21] M. Conti, S. Giordano, M. May, and A. Passarella, "From opportunistic networks to opportunistic computing," *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 126–139, Sep. 2010.

[22] J. Hu, L. L. Yang, K. Yang, and L. Hanzo, "Socially aware integrated centralized infrastructure and opportunistic networking: a powerful content dissemination catalyst," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 84–91, Aug. 2016.

[23] X. Hu and V. C. M. Leung, "Towards context-aware mobile crowdsensing in vehicular social networks," in *Proc. 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, May 2015, pp. 749–752.

[24] A. M. Vegni and V. Loscrí, "A survey on vehicular social networks," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2397–2419, Jul. 2015.

[25] K. M. Alam, M. Saini, and A. E. Saddik, "Toward social internet of vehicles: Concept, architecture, and applications," *IEEE Access*, vol. 3, pp. 343–357, 2015.

[26] J. Liu, N. Kato, J. Ma, and N. Kadowaki, "Device-to-device communication in lte-advanced networks: A survey," *IEEE Commun. Surveys and Tutorials*, vol. 17, no. 4, pp. 1923–1940, Nov. 2015.

[27] Y. Guan, Y. Xiao, L. J. C. Jr., and C.-C. Shen, "Power efficient peer-to-peer streaming to co-located mobile users," in *Proc. IEEE 11th Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2014, pp. 321–326.

[28] J. Liu, S. Zhang, N. Kato, H. Ujikawa, and K. Suzuki, "Device-to-device communications for enhancing quality of experience in software defined multi-tier lte-a networks," *IEEE Netw. Mag.*, vol. 29, no. 4, pp. 46–52, Jul. 2015.

[29] J. Wang, C. Jiang, T. Q. S. Quek, X. Wang, and Y. Ren, "The value strength aided information diffusion in socially-aware mobile networks," *IEEE Access*, vol. 4, pp. 3907–3919, Aug. 2016.

[30] M. Dashti, M. A. A. Rahman, H. Mahmoudi, and H. Claussen, "Detecting co-located mobile users," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 1565–1570.

[31] S. Milgram, "The small world problem," *Psychol. Today*, vol. 1, no. 1, pp. 61–67, May 1967.

[32] Bluetooth SIG, Inc. (2014, Dec.) Bluetooth Core Specification 4.2. [Online]. Available: http://www.bluetooth.com

[33] Apple, Inc. (2016, Jun.) Getting started with iBeacon. [Online]. Available: https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf

[34] A. Gupta, A. Kalra, D. Boston, and C. Borcea, “MobiSoC: a middleware for mobile social computing applications,” *Mobile Netw. Applicat.*, vol. 14, no. 1, pp. 35–52, Feb. 2009.

[35] A. Gupta, S. Paul, Q. Jones, and C. Borcea, “Automatic identification of informal social groups and places for geosocial recommendations,” *Int. J. Mobile Netw. Des. Innovation*, vol. 2, no. 3, pp. 159–171, Dec. 2007.

[36] S. A. R. Zekavat and R. M. Buehrer, *Handbook of Position Location: Theory, Practice and Advances.* New Jersey, USA: Wiley-IEEE Press, 2012.

[37] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proc. 5th Berkeley Symp. Math. Stat. Prob.*, vol. 1. Univ. of Calif. Press, Berkeley, 1967, pp. 281–296.

[38] A. W. Moore. (2016, Jun.) Clustering with gaussian mixture models. [Online]. Available: http://www.autonlab.org/tutorials/gmm14.pdf

[39] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[40] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proc. 5th Berkeley Symp. Math. Stat. Prob.*, vol. 1. Univ. of Calif. Press, Berkeley, 1967, pp. 281–296. [Online]. Available: http://projecteuclid.org/euclid.bsmsp/1200512992

[41] P. S. Bradley and U. M. Fayyad, “Refining initial points for k-means clustering,” in *Proc. of the Fifteenth International Conference on Machine Learning (ICML 1998)*, vol. 66, 1998.

[42] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finte state markov chains,” *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, 1966. [Online]. Available: http://projecteuclid.org/euclid.aoms/1177699147

[43] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, “A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains,” *Ann. Math. Statist.*, vol. 41, no. 1, pp. 164–171, Feb. 1970. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177697196

[44] J. H. L. Hansen and T. Hasan, “Speaker recognition by machines and humans: A tutorial review,” *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, Nov. 2015.

[45] P. Chandrasekar, S. Chapaneri, and D. Jayaswal, “Automatic speech emotion recognition: A survey,” in *2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA)*, Apr. 2014, pp. 341–346.

[46] H. Cheng, L. Yang, and Z. Liu, “Survey on 3d hand gesture recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1659–1673, Sep. 2016.

[47] S. Mitra and T. Acharya, “Gesture recognition: A survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, May 2007.

[48] T. X. Society, S. Wang, Q. Jiang, and J. Z. Huang, “A novel variable-order markov model for clustering categorical sequences,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2339–2353, Oct. 2014.

[49] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, “A survey of vision-based traffic monitoring of road intersections,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2681–2698, Oct. 2016.

[50] P. Baldi and Y. Chauvin, “Smooth on-line learning algorithms for hidden markov models.” vol. 6, no. 2, pp. 307–318, Mar. 1994.

[51] M. Bicego, V. Murino, and M. A. Figueiredo, “Similarity-based clustering of

sequences using hidden markov models," in *Machine Learning and Data Mining in Pattern Recognition, vol. LNAI 2734*. Springer, 2003, pp. 86–95.

[52] A. Gupta, S. Paul, Q. Jones, and C. Borcea, "Automatic identification of informal social groups and places for geo-social recommendations," *Int. J. Mobile Netw. Design Innovation*, vol. 2, no. 3–4, pp. 159–171, Jan. 2007.

[53] S. A. R. Zekavat and R. M. Buehrer, *Handbook of Position Location: Theory, Practice and Advances*. New Jersey, USA: Wiley-IEEE Press, 2012.

[54] D. Roggen, M. Wirz, G. Tröster, and D. Helbing, "Recognition of crowd behavior from mobile sensors with pattern analysis and graph clustering methods," *Netw. and Heterogeneous Media (NHM)*, vol. 6, no. 3, pp. 521–544, Sep. 2011.

[55] A. Nazábal, P. García-Moreno, A. Artés-Rodríguez, and Z. Ghahramani, "Human activity recognition by combining a small number of classifiers," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 5, pp. 1342–1351, Sep. 2016.

[56] H. Qian, Y. Mao, W. Xiang, and Z. Wang, "Recognition of human activities using svm multi-class classifier," *Pattern Recognition Letters*, vol. 31, no. 2, pp. 100–111, Jan. 2010.

[57] C. Randell and H. Muller, "Context awareness by analysing accelerometer data," in *Proc. Fourth International Symposium on Wearable Computers (ISWC)*, Oct. 2000, pp. 175–176.

[58] R. Guimerà and L. A. N. Amaral, "Functional cartography of complex metabolic networks," *Nature*, vol. 433, no. 7028, pp. 895–900, Feb. 2005.

[59] S. Kirkpatrick, J. C. Daniel Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–80, May 1983.

[60] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, Jan. 1985.

[61] G. Vanderhulst, M. Dashti, A. Mashhadi, and F. Kawsar, "Crumblr: Enabling proxemic services through opportunistic location sharing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2015, pp. 256–259.

[62] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Advances in Neural Inform. Process. Syst.* MIT Press, 2000, pp. 554–560.

[63] C. Robert and G. Casella. (2008, Aug.) A history of markov chain monte carlo – subjective recollections from incomplete data. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00311793/document

[64] C. M. Bishop, *Pattern Recognition and Machine Learning.* New York, NY, USA: Springer Press, 2006.

[65] J. Wang, Q. Gao, Y. Yu, P. Cheng, L. Wu, and H. Wang, "Robust device-free wireless localization based on differential rss measurements," *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5943–5952, Dec. 2013.

[66] K. Farrahi, R. Emonet, and A. Ferscha, "Socio-technical network analysis from wearable interactions," in *Proc. 16th Int. Symp. Wearable Comput. (ISWC)*, Jun. 2012, pp. 9–16.

[67] F. Wood and M. J. Black, "A nonparametric bayesian alternative to spike sorting," *J. Neurosci. Methods*, vol. 173, no. 1, pp. 1–12, Aug. 2008.

[68] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*, 3rd ed. London, U.K.: Chapman & Hall/CRC Press, 2014.

[69] K. P. Murphy, *Machine Learning: A Probabilistic Perspective.* Cambridge, Mass.: MIT Press, 2012.

[70] Y. W. Teh. (2016, Oct.) Dirichlet process. [Online]. Available: http://www.gatsby.ucl.ac.uk/~ywteh/research/npbayes/dp.pdf

[71] H. Kamper. (2013, Nov.) Gibbs sampling for fitting finite and infinite gaussian mixture models. [Online]. Available: http://www.kamperh.com/notes/kamper_bayesgmm13.pdf

[72] P. Resnik and E. Hardisty, "Gibbs sampling for the uninitiated," Univ. of Maryland, College Park, Tech. Rep. LAMP–TR–153, 2010.

[73] T. L. Griffiths and Z. Ghahramani, "Infinite latent feature models and the indian buffet process," in *Proc. Neural Inf. Process. Syst. (NIPS).* MIT Press, 2005, pp. 475–482.

[74] D. J. Aldous, *Exchangeability and related topics.* Berlin: Springer-Verlag: École d'Été de Probabilités de Saint-Flour XIII - 1983, 1985.

[75] Riverbed Technology, Inc. (2016, Sep.) AirPcap Nx. Available: http://www.riverbed.com.

[76] The Wireshark team. (2016, Sep.) Wireshark. Available: https://www.wireshark.org.

[77] S.-H. Cha. (2016, Sep.) Comprehensive survey on distance/similarity measures between probability density functions [Online]. Available: http://www.gly.fsu.edu/~parker/geostats/Cha.pdf.

[78] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, 2002.

[79] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[80] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows: theory, algorithms, and applications.* New Jersey, USA: Prentice Hall, 1993.

[81] J. DeCuir, "Introducing bluetooth smart: Part I: A look at both classic and new technologies." *IEEE Consum. Electron. Mag.*, vol. 3, no. 1, pp. 12–18, Jan. 2014.

[82] J. Decuir, “Introducing bluetooth smart: Part II: Applications and updates.” *IEEE Consum. Electron. Mag.*, vol. 3, no. 2, pp. 25–29, Apr. 2014.

[83] M. Kohne and J. Sieck, “Location-based services with ibeacon technology,” in *Proc. 2nd Int. Conf. Artificial Intell. Modelling Simulation (AIMS)*, Nov. 2014, pp. 315–321.

[84] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, “How low energy is bluetooth low energy? comparative measurements with zigbee/802.15.4,” in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2012, pp. 232–237.

[85] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets Reasoning About a Highly Connected World.* New York, NY, USA: Cambridge University Press, Jul. 2010.

[86] S. Boccaletti, V. Latora, Y. Moreno, and D.-U. H. Martin Chavez, “Complex networks: Structure and dynamics,” *Phys. Rep.*, vol. 424, no. 4, pp. 175–308, Feb. 2006.

[87] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E*, vol. 69, no. 2, p. 026113, Feb. 2004.

[88] U. Brandes, “A faster algorithm for betweenness centrality,” *The Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001. [Online]. Available: http://dx.doi.org/10.1080/0022250X.2001.9990249

[89] Gimbal, Inc. (2016, Jul.) Gimbal beacons. [Online]. Available: http://www.gimbal.com

[90] ——. (2014, Nov.) Gimbal proximity beacon specification sheet. [Online]. Available: https://cdn.shopify.com/s/files/1/0680/0893/files/GimbalBeaconSpecRevC.pdf

Chapter 6

Publication List

6.1 Journals

[1] Pedro M. Varela and T. Ohtsuki, “Discovering Co-Located Walking Groups of People Using iBeacon Technology,” *IEEE Access*, vol. 4, pp. 6591–6601, Oct. 2016. DOI: 10.1109/ACCESS.2016.2615863

[2] Pedro M. Varela, J. Hong, T. Ohtsuki, and X. Qin, “IGMM-Based Co-Localization of Mobile Users With Ambient Radio Signals,” *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, May 2016. DOI: 10.1109/JIOT.2016.2568258

6.2 Conferences Proceedings (peer-reviewed)

[1] Pedro M. Varela and T. Ohtsuki, “Co-Location of Walking Groups of Users,” in *Proc. IEEE International Conference on Communications (ICC)*, 2017 (*under review*).

[2] Pedro M. Varela, J. Hong, and T. Ohtsuki, “IGMM-Based Approach for Discovering Co-located Mobile Users,” in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, Dec. 2016.

6.3 Conferences Proceedings (without peer-review)

[1] Pedro M. Varela, J. Hong, and T. Ohtsuki, “A Nonparametric Bayesian Approach for Co-location of Mobile Users,” in *Proc. IEICE-CS Technical Report*, ASN2015-83, vol. 115, no. 437, pp. 21 – 26, Jan. 2016.