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

Pedro M. Varela, *Student Member, IEEE,* Jihoon Hong, *Member, IEEE,* Tomoaki Ohtsuki, *Senior Member, IEEE,* and Xiaoqi Qin, *Student Member, IEEE*

***Abstract*—Co-localization of mobile users combines methods of detecting nearby users and providing them interesting and useful services or information. By exploiting the massive use of smartphones, nearby users can be co-localized using only their captured ambient radio signals. In this paper, we propose a real-time co-localization system, in a centralized manner, that leverages co-located users with high accuracy. We exploit the similarity of radio frequency measurements from users' mobile terminal. We do not require any further information about them. Our co-localization system 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. In addition, we propose a modified version of Gibbs sampling technique with an average similarity threshold to better fit user's group. We design our system in a completely centralized manner. Hence, it enables the network to control and manage the formation of the users' groups. We first evaluate the performance of our proposal numerically. Then, we carry out an extensive experiment to demonstrate the feasibility, and the efficiency of our approach with data sets from a real-world setting. Results on experiment favor our algorithm over the state-of-the-art community detection based clustering method.**

***Index Terms*—Co-location, Gaussian mixture model, mobile computing, clustering.**

I. Introduction

THE large-scale use of the smart devices has given an energetic impulse to a rapid development of a variety of mobile applications. Moreover, it has also triggered a lot of attention in the research community in the recent years. With this proliferation of mobile devices, new services are also provided to the customers, depending on their current location. One of them is known as location-based services (LBS), in which nearby places of interest are ubiquitously queried by users based on their current positions transmitted to the location server.

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 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 from the same ambient signals, will cluster mobile users into the same group.

It is worth noting that, in the localization system [1], [2], the absolute or relative position of an individual user is estimated and displayed on a surface of a map. However, in the co-location system, we aim at determining users who are geographically near one to the another, which can be somehow confusing. This confusion is mainly explained by the fact that, contrary to the localization system [1], [2], the absolute position of the users in the network is not necessary, and the fixed measure of vicinity among users to state that they are co-located is fuzzy. Therefore, depending on applications targeted, we can define how near two or more users can be considered as co-located.

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 people (we consider they are interacting, in some way) can serve as many purposes. For example, in the authentication scenarios [3] with nearby people, in wireless networks, to prevent eavesdropper and spoofer attacks; in gaining a better understanding of human social interactions; in mobile geosocial networking [4]; in providing real-time recommendations about people with the common interests; in detecting coworkers in the same place and deliver message on their smartphones.

In addition, by taking advantage of physically closely co-located mobile UEs, one can directly route data traffic between mobile users (e.g., sharing streaming video, pictures, etc.), which is known as Device-to-Device (D2D) [5] communication, for the purposes of proximity-based services [6] 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 [7], in improving throughput, delay, spectrum efficiency, as well as enhancing Quality of Experience (QoE) in LTE-Advanced networks [8].

Detecting such co-located group of people based on their geographical positions was proposed in [9]. However, it presents some issues and privacy concerns arise among them. In [9], the location of users is used to identify group of people and their associated places. By collecting positions of the users for a long period of time exposes them to be easily tracked with

Manuscript received ...; revised ...; accepted ....

P. Varela is with the Graduate School of Science and Technology, Keio University, Yokohama 223-8522, Japan (e-mail: pedro@ohtsuki.ics.keio.ac.jp).

J. Hong was with Keio University, Yokohama 223-8522, Japan. He is now with Kanagawa Institute of Technology, Kanagawa 243-0292, Japan (e-mail: jihoon.hong@cco.kanagawa-it.ac.jp).

T. Ohtsuki is with the Department of Information and Computer Science, Keio University, Yokohama 223-8522, Japan (e-mail: ohtsuki@ohtsuki.ics.keio.ac.jp).

X. Qin is with Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA (e-mail: xiaoqi@vt.edu).

today's technologies. Here, we design an algorithm to perform co-localization of mobile users that have been together for a certain amount of time. This algorithm is robust in infering co-located users, and does not disclose users' absolute position, which preserves users' location privacy.

For the purpose of realizing potential applications of co-located mobile users, we propose a new method to detect in real-time co-located users in wireless networks. It is based on a nonparametric Bayesian technique (NPB) called infinite Gaussian mixture modeling (IGMM) [10]. To classify users' measured radio signals a Markov chain Monte Carlo (MCMC) implementation of a hierarchical IGMM [11] is utilized. This method is built on spatial-temporal location of the mobile users, and infers co-located 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 users.

One main advantage of this method is that it avoids the need of the *a priori* knowledge of the input data, *i.e.*, the number of active devices operating in the network. Indeed, in a real-world scenario, the number of users that bands together in a room, for instance, is unpredictable and changes over time, which makes NPB an appealing technique to address this kind of problem. Besides, with this approach the number of clusters[1] in the input data is automatically detected, in contrast with other methods that need to be told how many clusters to find [12].

Our approaches are practical for several reasons and can be implemented efficiently with high accuracy, as discussed later on. First, we use ambient WiFi signals whose detection are available in nearly every smartphone, and increasingly, hot spots can be found anywhere we go. Second, the discovery of co-located users is centralized, which allows the co-location server to control the formation of co-located users. Third, by adopting a modified version of Gibbs sampling method with a similarity threshold, we effectively detect co-located users that have spent a certain amount of time in the same place.

Note that our method does not estimate the absolute position [13] of an individual user, which prevents him 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.

A. Our Contributions

We summarize the contributions of our paper as follows:

- We propose a new real-time approach to infer co-located mobile users, in a centralized manner, by exploiting the similarity of their measurements of the shared ambient radio signals, based on a nonparametric Bayesian method called IGMM. Furthermore, a modified version of Gibbs sampling is proposed as a key enabler to a high co-localization accuracy, in accordance with application requirements.
- In association with RSSI, MAC address, and arrival time of beacon packets from environmental WiFi signals, we analyzed the performance of our proposal not only numerically but also experimentally in order to demonstrate its feasibility. We also perform a comparison result. Beyond being practical and efficient, results on experiment with real data sets favor our algorithm over its counterpart modularity-based community-detection approach presented in [14].

The remainder of this paper is organized as follows. In Section II we overviewed the related works. In Section III we discuss the co-location system based on IGMM and ambient radio signals. Then, we provide a numerical results in Section IV. In Section V we present the experimental results and a comparison study, followed by a conclusion in Section VI.

II. Related Works

The co-localization system has been subjected to several researches in recent years, due to its importance on people-centric and place-centric mobile applications [15].

An easy way of thinking to address this 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 [9]. Despite the fact that this approach seems attractive at first, it presents several drawbacks associated with positioning systems to co-localize 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) [16]. Another drawback is that collecting people's position for a long period of time can allow them to be easily tracked. Therefore, robust techniques to infer groups of co-localized users are needed, without disclosing their absolute position.

Traditional approaches such as k-means [17] or Gaussian mixture modeling [18] provide also a way to solve this problem. However, both of them suffer from the same drawbacks. In fact, these algorithms require a fixed number of clusters, which they need to be told to find. As the number of users can change over time, and consequently the number of hidden clusters is unknown and may also vary, 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.

Dashti *et al*. [14] devised a method to co-localize mobile users based on the similarity of their radio frequency (RF) fingerprints, by exploiting their shared ambient radio signals. In [14], the authors constructed a connectivity graph by taking into account the similarity of user's measured signals. Then, a modularity-based community detection approach is applied to cluster users [19]. To maximize their modularity function,

[1]In this work, the words cluster and group are used interchangeably.

a heuristic technique called simulated annealing is utilized [20]. Simulated annealing is a randomized search process that avoids the problem of getting stuck in local optima-solutions that are better than any other neighbors, but are not the very best. In our work, we also exploit multiple ambient radio signals' features. However, we apply a Markov chain Monte Carlo (MCMC) method called collapsed Gibbs sampling technique for classification [11]. 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 [21].

Mardenfeld *et al.* [22], on the other hand, proposed to identify co-located users by using their Bluetooth traces. The proposed algorithm [22] identifies groups of users that have been spending a certain amount of time together and meeting for several times. To validate their approaches, one month of collected radio signals were used from many users' smartphones. Similar to them, we evaluate our approach in a real-world scenario, carrying out a vast experiment in an entire second floor of a building with several meeting rooms, an open space, and corridors. Results from experiment show the effectiveness of our approach.

III. Problem Statements

In this section, we introduce our system architecture. Features from environmental WiFi signals are extracted, combining with the current location of the mobile users, lead to the high clustering accuracy. Our algorithm based on IGMM for modeling and Gibbs sampling for classification is also explained in detail in the following.

A. System Model

Mobile users that have been together, for a certain amount of time, experience the similar WiFi radio signals from their shared ambient radio signals. Hence, we aim at detecting these users with similar RF measurements and cluster them into the same group.

In Fig. 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 work, WiFi radio signals are used because of their easy deployment and no extra cost, and their ability of working in both indoor and outdoor environments. However, other ambient radio signals such as Bluetooth, GSM, FM radio, LTE signals, *etc.*, can be exploited as well to co-locate mobile users.

In the following subsection, we discuss in detail our implementation based on IGMM. For ease of reference, Table I summarizes the notation of all the mathematical symbols used in this paper.

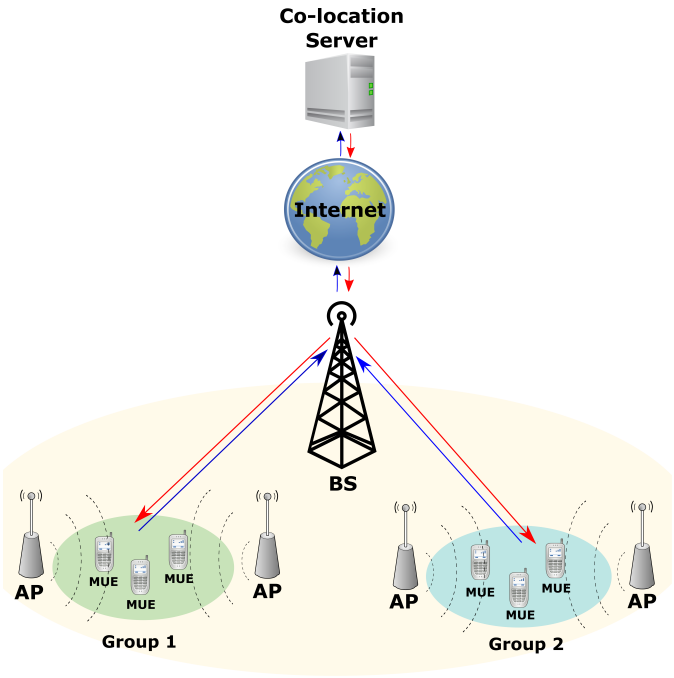

Fig. 1. An example network architecture of co-localized mobile equipments. The blue arrows indicate the transmission of the ambient radio signals to the co-location server. The red arrows represent information of co-localized mobile equipments sent by the server.

B. 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 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 explain how to generalize this model for the multivariate case later on.

Farrahi *et al.* [23] 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. Thus, one Gaussian mixture model will be used to model each class.

1) Fixed number of classes: Our co-localization technique is implemented with infinite Gaussian mixture model (IGMM) for modeling, and Gibbs sampler for classification. In [10], 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, the number of classes is known, and then explore the model when the number of the classes is unknown. So, let's 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 (1), and derive the model later when $K \to \infty$.

TABLE I
LIST OF SYMBOLS AND NOTATIONS USED IN THIS PAPER

Symbol	Definition
N	Total number of observations
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 interacting/non-interacting users

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

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 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

$$p(\mu_j|\lambda, r) \sim \mathcal{N}(\lambda, r^{-1}), \tag{2}$$

whose mean, λ, and precision, r, are hyperparameters of the model. Their priors are given by $p(\lambda) \sim \mathcal{N}(\mu_y, \sigma_y^2)$ and $p(r) \sim \mathcal{G}a(1, \sigma_y^{-2})$, 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

$$p(s_j|\beta, \omega) \sim \mathcal{G}a(\beta, \omega^{-1}), \tag{3}$$

whose shape, β, and mean, ω^{-1}, are also hyperparameters of the model. Their priors are given by $p(\beta^{-1}) \sim \mathcal{G}a(1,1)$, and $p(\omega) \sim \mathcal{G}a(1, \sigma_y^2)$, which are inverse Gamma and Gamma, respectively.

Following [10], 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[2] of the Multinomial distribution, whose joint pdf is in the following form

$$\begin{aligned} p(\boldsymbol{\pi}|\alpha) &\sim \text{Dir}(\alpha/K, \alpha/K, \ldots, \alpha/K) \\ &= \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \prod_{j=1}^{K} \pi_j^{\frac{\alpha}{K}-1}, \end{aligned} \tag{4}$$

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 (4) encode our beliefs about how uniform or skewed the class mixture weights will be [21].

In our experiment, we collected WiFi signals sent by three different access points. 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 $\mathbf{\Sigma}_j$. For the remainder of this paper, all discussion will be focused on the multidimensional space, *i.e.*, $D = 3$.

According to [24], the conjugate prior distribution of the mean vector $\vec{\mu}_j$ and covariance matrix $\mathbf{\Sigma}_j$, can be computed with Gaussian inverse Wishart (GIW) distribution, with hyperparameters $H = (\mathbf{\Lambda}_0^{-1}, \upsilon_0, \vec{\mu}_0, \kappa_0)$, and they are denoted as

$$\begin{aligned} \mathbf{\Sigma}_j &\sim \text{IW}_{\upsilon_0}(\mathbf{\Lambda}_0^{-1}) \\ \vec{\mu}_j|\mathbf{\Sigma}_j &\sim \mathcal{N}(\vec{\mu}_0, \mathbf{\Sigma}_j/\kappa_0), \end{aligned} \tag{5}$$

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

$$p(\boldsymbol{\mu}, \mathbf{\Sigma}) = \text{GIW}(\boldsymbol{\mu}, \mathbf{\Sigma}|\mathbf{\Lambda}_0^{-1}, \upsilon_0, \vec{\mu}_0, \kappa_0), \tag{6}$$

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

$$\begin{aligned} &\text{GIW}(\boldsymbol{\mu}, \mathbf{\Sigma}|H) \triangleq \mathcal{N}(\boldsymbol{\mu}|\vec{\mu}_0, \mathbf{\Sigma}/\kappa_0) \cdot \text{IW}(\mathbf{\Sigma}|\mathbf{\Lambda}_0^{-1}, \upsilon_0) \\ &= \frac{|\mathbf{\Sigma}|^{-\frac{\upsilon_0+D+2}{2}}}{Z_{GIW}} \exp\left[-\frac{\kappa_0}{2}(\boldsymbol{\mu}-\mu_0)^2\mathbf{\Sigma}^{-1} - \frac{\text{Tr}(\mathbf{\Sigma}^{-1}\mathbf{\Lambda}_0^{-1})}{2}\right] \end{aligned} \tag{7}$$

where $Z_{GIW} = 2^{\frac{\upsilon_0 D}{2}} \Gamma_D(\upsilon_0/2)(2\pi/\kappa_0)^{D/2}|\mathbf{\Lambda}_0^{-1}|^{-\upsilon_0/2}$, and $\Gamma_D(\cdot)$ is the multivariate Gamma function. The complete derivation can be found in [25, 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 $\mathbf{\Lambda}_0^{-1}$ is proportional to our prior mean for $\mathbf{\Sigma}$, and υ_0 encodes our confidence about that. For reference, the pdf of the inverse Wishart distribution is given in (8), where

[2]A prior is conjugate if it yields a posterior that is the same family as the prior (a mathematical convenience) [11].

υ is the number of degrees of freedom of the distribution, $\mathbf{\Lambda}$ is a $D \times D$ scale matrix, and $\mathrm{Tr}(\cdot)$ denotes the trace.

$$p(\mathbf{\Sigma}) = \frac{|\mathbf{\Lambda}^{-1}|^{\upsilon/2}|\mathbf{\Sigma}|^{-\frac{\upsilon+D+1}{2}} \exp\left[-\frac{1}{2}\mathrm{Tr}(\mathbf{\Lambda}^{-1}\mathbf{\Sigma}^{-1})\right]}{2^{\frac{\upsilon D}{2}}\Gamma_D(\upsilon/2)}. \quad (8)$$

For the sake of completeness, we also provide here the pdf of the multivariate Gaussian distribution in (9), where $\boldsymbol{\mu}$ is the mean and $\mathbf{\Sigma}$ is a $D \times D$ covariance matrix.

$$p(y|\boldsymbol{\mu}, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(y-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(y-\boldsymbol{\mu})\right]. \quad (9)$$

Our purpose is to infer the class of each one of our N observations, $\mathbf{y}$, from the feature space. So, let's 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 $p(z_i = j) = \pi_j$.

2) 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 infinite Gaussian mixture model (IGMM) comes, which is actually an extreme case of FGMM when $K \to \infty$.

We have chosen the $p(\boldsymbol{\pi}|\alpha)$ and $p(\vec{\mu}_j, \mathbf{\Sigma}_j|H)$ to be our conjugate prior, therefore one may integrate out the model parameters $\boldsymbol{\pi}$, $\vec{\mu}_j$ and $\mathbf{\Sigma}_j$, 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

$$\begin{aligned} p(z_i = j|\mathbf{z}_{-i}, \mathbf{y}, \alpha, H) &\sim p(\mathbf{z}|\alpha)p(\mathbf{y}|\mathbf{z}, H) \\ &\sim p(z_i = j|\mathbf{z}_{-i}, \alpha)p(\mathbf{y}|z_i = j, \mathbf{z}_{-i}, H) \\ &\sim p(z_i = j|\mathbf{z}_{-i}, \alpha)p(y_i|\mathbf{y}_{-i}, H), \end{aligned} \quad (10)$$

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

In order to determine the value of the posterior probability in (10), we should derive the expressions of the first and the second terms on the right side.

To educe 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

$$p(\mathbf{z}|\alpha) = \int_{\boldsymbol{\pi}} p(\mathbf{z}|\boldsymbol{\pi})p(\boldsymbol{\pi}|\alpha)d\boldsymbol{\pi}, \quad (11)$$

where the first term $p(\mathbf{z}|\boldsymbol{\pi}) = \prod_{j=1}^{K}\pi_j^{n_j}$, and the second term is given in (4). Hence, following [25] we have

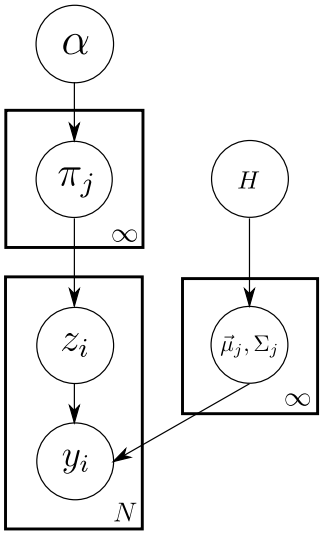

Fig. 2. Graphical model representation of Bayesian infinite Gaussian mixture model in our co-localization system.

$$\begin{aligned} p(\mathbf{z}|\alpha) &= \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K}\int_{\boldsymbol{\pi}}\prod_{j=1}^{K}\pi_j^{n_j+\frac{\alpha}{K}-1} \\ &= \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}\prod_{j=1}^{K}\frac{\Gamma(n_j+\alpha/K)}{\Gamma(\alpha/K)}, \end{aligned} \quad (12)$$

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

Our goal is to sample from posterior distribution over the model when the limit $K \to \infty$. An MCMC technique known as Gibbs sampling [26] 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 (12), we obtain

$$p(z_i = j|\mathbf{z}_{-i}, \alpha) = \frac{n_{-i,j} + \alpha/K}{N - 1 + \alpha}, \quad (13)$$

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 $K \to \infty$ in (13), the conditional prior reaches the followings limits

$$p(z_i = j|\mathbf{z}_{-i}, \alpha) = \begin{cases} \frac{n_{-i,j}}{N-1+\alpha}, & \text{if} \quad n_{-i,j} > 0, \\ \frac{\alpha}{N-1+\alpha}, & \text{if} \quad n_{-i,j} = 0 \end{cases} \quad (14)$$

where $n_{-i,j} = 0$ means that, no observation has been assigned yet to class j. The generative model in (14) is a characterization of Dirichlet process known as Chinese restaurant process (CRP) [27].

Same as the first term in (10) (right side) follows two cases, described in (14), we may also find two expressions for the

second term. Indeed, following [21] and [25], the second term in (10) is obtained by the multivariate Student-t distribution, because of our previous choice of conjugate prior. Therefore,

$$p(y_i|\mathbf{y}_{-i},H) \sim t_{\upsilon_n-D+1}\left(\vec{\mu}_n, \frac{\mathbf{\Lambda}_n(\kappa_n+1)}{\kappa_n(\upsilon_n-D+1)}\right), \quad (15)$$

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 (15) are defined as follows

$$\vec{\mu}_n = \frac{\kappa_0}{\kappa_0+N}\vec{\mu}_0 + \frac{N}{\kappa_0+N}\bar{y} \quad (16)$$

$$\kappa_n = \kappa_0 + N \quad (17)$$

$$\upsilon_n = \upsilon_0 + N \quad (18)$$

$$\mathbf{\Lambda}_n = \mathbf{\Lambda}_0 + S + \frac{\kappa_0 n}{\kappa_0+N}(\bar{y}-\vec{\mu}_0)(\bar{y}-\vec{\mu}_0)^T \quad (19)$$

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

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 (15), with the hyperparameters before updating

$$p(y_i,H) \sim t_{\upsilon_0-D+1}\left(\vec{\mu}_0, \frac{\mathbf{\Lambda}_0(\kappa_0+1)}{\kappa_0(\upsilon_0-D+1)}\right). \quad (20)$$

For reference, the pdf of the multivariate Student-t distribution is given in (21), where υ is the degrees of freedom, $\boldsymbol{\mu}$ is the mean, and $\mathbf{\Lambda}$ is a $D \times D$ scale matrix.

$$t_\upsilon(y|\boldsymbol{\mu},\mathbf{\Lambda}) = \frac{\Gamma(\frac{D+\upsilon}{2})}{\Gamma(\frac{\upsilon}{2})}\frac{|\mathbf{\Lambda}|^{1/2}}{(\pi\upsilon)^{D/2}}\left[1+\frac{(y-\boldsymbol{\mu})^2\mathbf{\Lambda}^{-1}}{\upsilon}\right]^{-\frac{\upsilon+D}{2}}. \quad (21)$$

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

$$p(z_i = j|\mathbf{z}_{-i},\mathbf{y},\alpha,H) \sim \frac{n_{-i,j}}{N-1+\alpha} t_{\upsilon_0-D+1}\left(\vec{\mu}_0, \frac{\mathbf{\Lambda}_0(\kappa_0+1)}{\kappa_0(\upsilon_0-D+1)}\right), \quad (22)$$

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

$$p(z_i \neq z_{i'}, \forall i \neq i'|z_{-i},\mathbf{y},\alpha,H) \sim \frac{\alpha}{N-1+\alpha} t_{\upsilon_0-D+1}\left(\vec{\mu}_0, \frac{\mathbf{\Lambda}_0(\kappa_0+1)}{\kappa_0(\upsilon_0-D+1)}\right). \quad (23)$$

Fig. 2 depicts the graphical representation of this model in order to co-localize mobile users. 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 parameter α. The rectangular blocks represent the repetition, and the number in the lower right corner indicates the number of repetitions.

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

Input : Data sets from N users, and pre-set threshold Δ.
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 (22).
      $AVGSIM_j \leftarrow$ (24)
      $DIST_{(i,j)} \leftarrow$ distance to cluster j.
    end for
    Compute prob. of a new class as in (23).
    $z_i \leftarrow$ class with highest prob. and $DIST_{(i,j)} \leq \Delta$.
    Remove any empty class, and decrease K.
  end for
end for

C. 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 WiFi 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

$$AVGSIM_j = \frac{1}{n_j}\sum_{i=1}^{N}\delta_{\text{Kronecker}}(z_i, j), \quad (24)$$

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 similar-

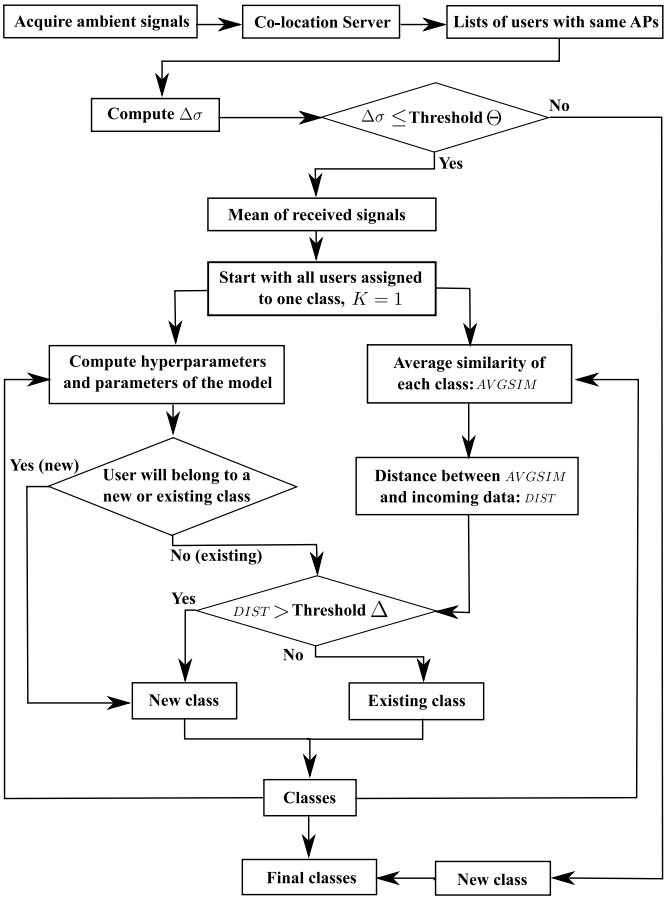

Fig. 3. Flow chart of co-localization based IGMM algorithm.

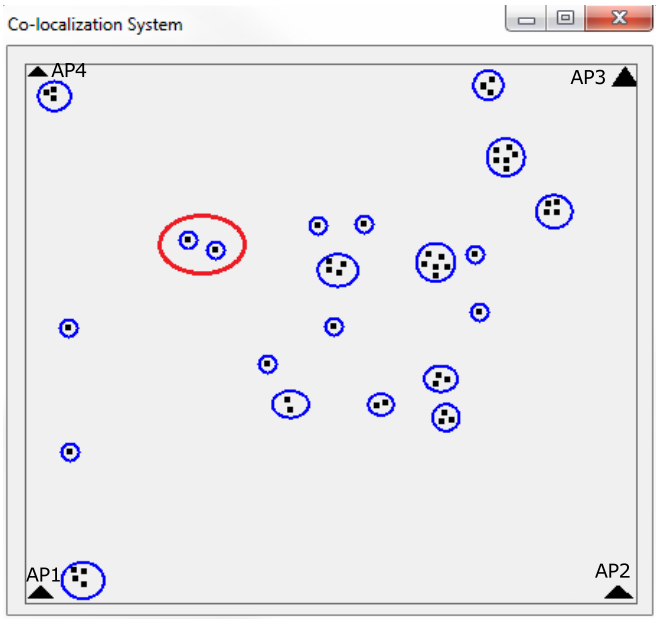

Fig. 4. Numerical results of our co-localization system. Each black dot represents a user in the wireless network. The blue circles indicate the actual co-located group of users. The red circle shows the misclassification case. We consider four APs in this simulation.

ity 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.

D. Co-location Scheme Detection

To detect and cluster co-located users, we propose the following scheme (see Fig. 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 V. Finally, the users with the strongest $\Delta\sigma_j$ are assigned to different classes at the end of the algorithm.

In the case when two or more users are walking together, the proposed scheme cannot be applied directly, because clustering group of walking users requires different approaches, which are beyond the scope of this paper.

The proposed scheme has several advantages. One of them is that the users that experience different APs radio signals will never be clustered together. Another one is that by introducing the similarity threshold Δ, in our process of clustering, we are able to determine all existing clusters. The proposed approach is also robust to deal with varying number of clusters and users over time.

IV. 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 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.

Fig. 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. As the moving users are not considered in this simulation, the threshold Θ is not used.

As can be seen in Fig. 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.

For this simulation, we chose the similarity threshold Δ, by trial-and-error process, that gave us the best results. However, as we explain in the next section, this threshold can be determined in offline analysis, and set according to the application requirements.

V. Experimental Setup and Results

In this section, we first discuss our experimental setup and present the obtained results using collected real-world WiFi signals. Then, in subsection V-E, we compare the performance of our method, in terms of clustering accuracy, against community detection-based approach proposed in [14] to co-locate mobile users.

A. 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 WiFi 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 Fig. 5).

We utilized wireless adapters *AirPcap Nx* [28] and a free and open-source packet analyzers *Wireshark* [29] to simultaneously capture environmental radio signals. WiFi 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.

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

The concentration parameter, α, and the hyperparameters denoted by $H = (\mathbf{\Lambda}_0^{-1}, \upsilon_0, \vec{\mu}_0, \kappa_0)$ in IGMM model express our prior belief on the distribution and need to be specified roughly [25]. 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.

B. 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.

Fig. 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

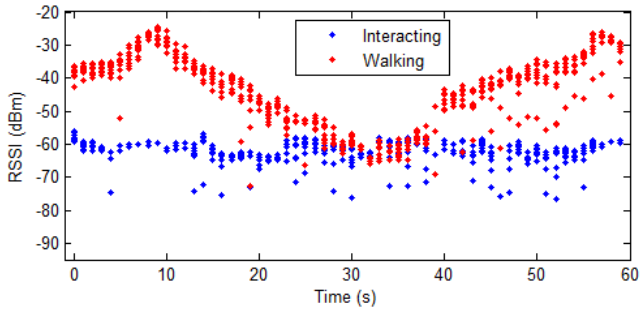

Fig. 6. RSSIs extracted from interacting (blue dots) and walking (red dots) users, for a period of time (Δt) of one minute. Interacting users were in the same room, while a user was walking in the corridor.

TABLE II
$\Delta\sigma_j$ 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	-

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 (25)

$$\Delta\sigma_j = \sqrt{\sum_{i=1}^{D} \sigma_{j,i}^2} \tag{25}$$

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

Table II 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 II 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 WiFi 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.

TABLE III
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

C. 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 III 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 III.

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

Fig. 7 depicts the effect of 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.

It should be pointed out that, the optimal value of the simi-

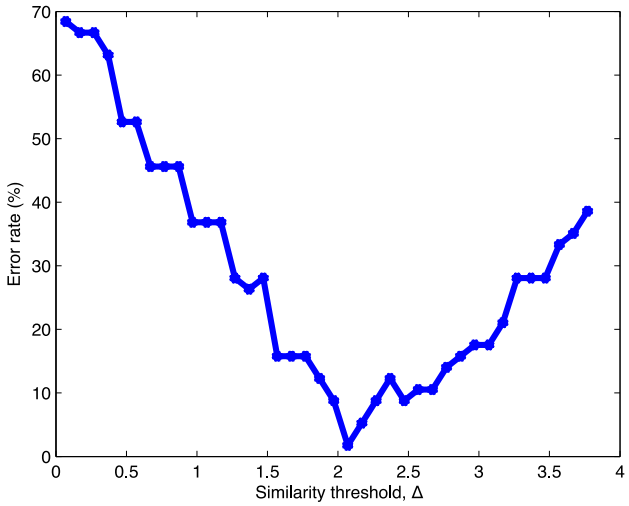

Fig. 7. Effect of the similarity threshold Δ on users co-localization. The value of Δ is computed in the signal domain for Euclidean distance metric.

larity 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.

In fact, 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 [6].

D. 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 Fig. 8.

Fig. 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.

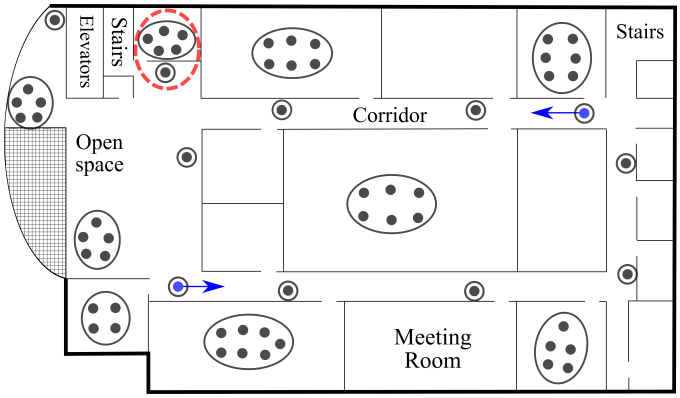

Fig. 8. Entire floor plan where the experiments were conducted and the obtained results. Each black or blue dot exemplifies a user in wireless network. The blue dots with a blue arrow indicate walking or just passing by users in the corridor. The black circles surrounding dots represent actual co-located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

E. Comparative Results

In this subsection, we will perform a comparative study between our proposal and the community detection-based approach presented in [14], on our measured WiFi signals.

As mentioned earlier, the authors in [14] proposed to co-locate mobile users by constructing a connectivity graph that represents the potential co-located users, based on pair-wise similarity of RF measurements. Then, they applied community-detection [20] 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 [30]. 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 [14] 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 WiFi 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 [14]. 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

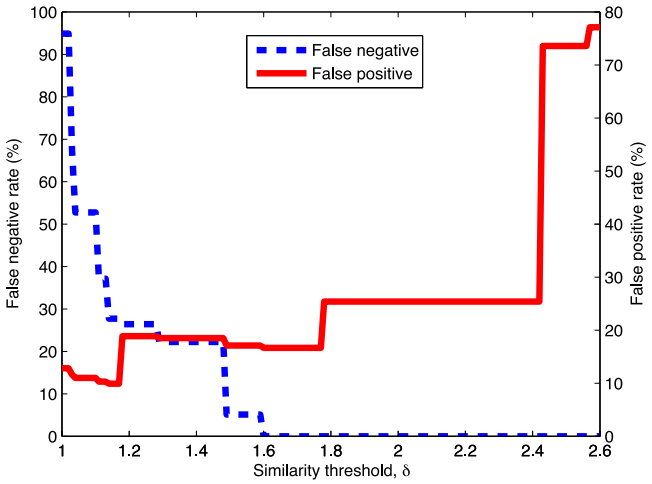

Fig. 9. Impact of the similarity threshold δ [14] on connectivity errors, for Euclidean distance metric. The step size $\Delta\delta$ is set to 0.01. The value of δ is computed in the signal domain.

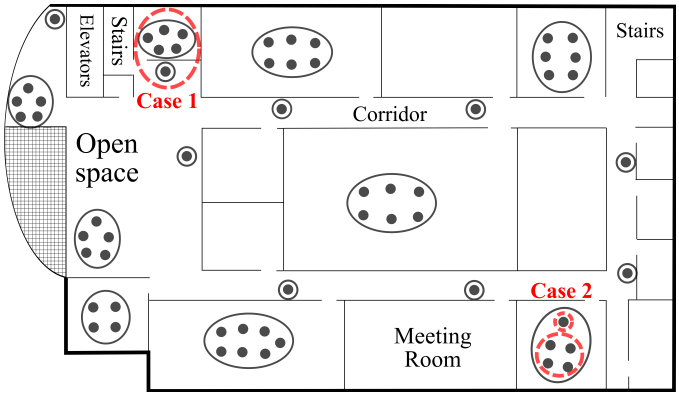

Fig. 10. Entire floor plan where the experiments were conducted and the obtained results, using CDSA-based approach. The red dash circles indicate the misclassification cases. The setup is the same as in Fig. 8, but without walking users.

of time. We do not consider the users that are walking or just passing by.

Fig. 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.

Fig. 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 IV presents the performance comparison between the IGMM-based and CDSA-based algorithms using Euclidean and Minkowski distance metrics. The Minkowski distance (l_p-norm, $p \geq 1$) [31] can be considered as a generalization of the Euclidean distance, and is calculated in the signal domain as

$$d_{Mink} = \sqrt[p]{\sum_{i=1}^{D} \left| RSSI_i^{(k)} - RSSI_i^{(m)} \right|^p} \qquad (26)$$

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 IV, 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.

TABLE IV
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%

VI. CONCLUSION

Throughout this paper, an efficient solution has been presented and evaluated to realize in real-time the co-localization of mobile users, by exploiting the similarity of their radio signals. It has been shown that by using a nonparametric Bayesian method called infinite Gaussian mixture model (IGMM) with a modified version of Gibbs sampler, our algorithm can accurately co-locate mobile users. The proposed design allows the co-location server to control and manage all aspects of the formation of the user groups in a centralized manner.

First, we numerically assessed our proposal. Then, we carried out an extensive experiment to demonstrate its performance with data sets from a real-world setting. In both cases, we have shown that our method can efficiently cluster co-located users. We have also compared our framework against a state-of-the-art community detection based clustering method. Results on experiments with real data sets favor our approach.

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

ACKNOWLEDGMENT

The authors would like to thank all students from our laboratory for their precious help during the setup experiment phases and in collecting data sets.

REFERENCES

[1] 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.

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

[3] L. Xiao, Q. Yan, W. Lou, G. Chen, and Y. T. Hou, "Proximity-based security using ambient radio signals," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2013, pp. 1609–1613.

[4] 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.

[5] 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.

[6] "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., 2013.

[7] 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.

[8] 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.

[9] 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.

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

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

[12] R. Xu and D. W. II, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[13] 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.

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

[15] 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.

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

[17] 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.

[18] A. W. Moore. Clustering with gaussian mixture models [Online]. Available: http://www.autonlab.org/tutorials/gmm14.pdf.

[19] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.

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

[21] 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.

[22] S. Mardenfeld, D. Boston, S. J. Pan, Q. Jones, A. Iamntichi, and C. Borce, "GDC: Group discovery using co-location traces," in *Proc. 2nd IEEE Int. Conf. Social Computing (SocialCom)*, Aug. 2010, pp. 641–648.

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

[24] 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.

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

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

[27] 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.

[28] Riverbed Technology, Inc. AirPcap Nx. Available: http://www.riverbed.com.

[29] The Wireshark team. Wireshark. Available: https://www.wireshark.org.

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

[31] S.-H. Cha. (2007) Comprehensive survey on distance/similarity measures between probability density functions [Online]. Available: http://www.gly.fsu.edu/~parker/geostats/Cha.pdf.