GroupUs: Smartphone Proximity Data and Human Interaction Type Mining

Trinh Minh Tri Do
Idiap Research Institute, Martigny, Switzerland
do@idiap.ch

Daniel Gatica-Perez
Idiap Research Institute, Martigny, Switzerland
EPFL, Lausanne, Switzerland
gatica@idiap.ch

Abstract

There is an increasing interest in analyzing social interaction from mobile sensor data, and smartphones are rapidly becoming the most attractive sensing option. We propose a new probabilistic relational model to analyze long-term dynamic social networks created by physical proximity of people. Our model can infer different interaction types from the network, revealing the participants of a given group interaction, and discovering a variety of social contexts. Our analysis is conducted on Bluetooth data sensed with smartphones for over one year on the life of 40 individuals related by professional or personal links. We objectively validate our model by studying its predictive performance, showing a significant advantage over a recently proposed model.

1 Introduction

People carry their mobile phone almost everywhere. By exploiting built-in sensors, smartphones have become attractive options for large-scale sensing of human and social behavior [6, 20]. The automatically determination of a user's social context is a desirable functionality for the next generation of adaptive, personalized mobile phone applications.

Integrated in phones and other mobile devices, Bluetooth is an imperfect yet reasonable approximation for sensing social interaction. Bluetooth information can tell if two persons carrying Bluetooth devices are in proximity with high probability. Bluetooth-based proximity also offers some important technical advantages such as low battery consumption,and the ability to work in both indoor and outdoor environments. Furthermore, people are often willing to share Bluetooth data with others. Note that, even without installing a client software to record Bluetooth interaction logs, people can share their Bluetooth information by setting their device to discoverable mode, and in practice many users do so on a regular basis. This is a key difference between relational Bluetooth data and other self-sensor data such as GPS, where the observed data involves only the actual device holder.

Bluetooth has been used as a social sensor in the past [21, 20, 6, 5]. Perhaps the simplest example is the use of the number of discovered nearby devices as a measurement of the human density of the environment [19]. At a public place, one could observe many nearby devices from likely unknown people. A different challenge is that of discovering the recurrent patterns of interaction with people in our social network (work colleagues, family members) and the context (temporal and spatial) in which they occur in real life. Many of these interactions take place over multiple time scales and multiple groups: we might have breakfast and dinner with our family every day, meet our collaborators twice a week, our boss once a month, and our sport teammates every sunday. The robust discovery of real-life interaction types therefore call, on one hand, for methods that are able to handle uncertainty in a principled way, and on the other, for longitudinal data to discover these possibly long-term effects.

In this paper, we present a probabilistic framework to discover social context, such as a weekly group meeting or having lunch with family members. Based on Bluetooth information collected for a large population over several months of daily life, our framework automatically assigns an *interaction type* for each Bluetooth link between two persons while discovering what these different types of group interactions correspond to. Our work makes the following contributions: (1) we introduce a new model, called GroupUs, for interaction type discovery from proximity data, designed to overcome some of the limitations of Bluetooth instantaneous data by integrating longitudinal observations of real-life proximity; (2) we conduct our analysis on an interaction data set spanning the life of 40 people over 12 months of time; and (3) we show that GroupUs can infer different interaction types from the full Bluetooth proximity data set, and assign group membership to the individuals who best conform them.

The structure of the paper is as follows. Section 2 reviews related works on Bluetooth data analysis and group discovery on social networks. We present our method in Section 3. The data collection framework and experimental results are presented in Sections 4 and 5. Finally, we draw conclusions in Section 6.

2 Related work

Our work can be positioned within the emergingbody of work on reality mining, which analyzes human behavior at large scale using mobile phones as sensors of activity [6, 20, 11]. However, the idea of using Bluetooth as a proximity sensor is not new. For instance, Terry et al [21] looked for pairwise proximity patterns over time. Raento et al. [20] were among the first to propose the use of mobile phones for large-scale context sensing, a first step towards reality mining [6]. In [5], Eagle et al. proposed to use Bluetooth and phone calls data to define pairwise links between people and in this way infer friendship networks, as an alternative to questionnaire-based, self-reported data. More recently, Mardenfeld et al. [17] proposed an algorithm for group discovery which is based on fully connected components of a Bluetooth proximity network. This method, however, has a number of drawbacks such as sensitivity to noise, the inability to discover very large groups (e.g., a lecture in an auditorium), and a complexity that grows exponentially with respect to group size.

Several other works address face-to-face interaction discovery by using other types of dedicated mobile devices, partly due to the limitations of Bluetooth to sense actual face-to-face proximity (instead of simply detecting people sharing an office or in adjacent offices) [9, 24, 18]. While these dedicated devices provide a definite advantage over Bluetooth to sense the actual interaction in terms of spatial resolution, and use voice and infrared sensors, they need to be worn in specific conditions to work in practice. Furthermore, they typically represent an additional device that many people might not be willing to carry in daily life.

There is a clear connection between discovering interaction types and discovering places, which is a problem that has been widely studied in mobile and ubiquitous computing using GPS or other types of location data [2, 12, 16]. Clearly, knowing that specific interactions tend to occur at certain places represents a strong prior - friends meet at restaurants and bars, families with children go to the park. Our GroupUs model could be extended to include location data in order to anchor the discovery of interaction types to geographic or semantic locations.

In data mining and machine learning, there is much interest in relational data [23]. Some methods have been proposed to extract groups, which are mainly based on discovering block structure from interaction, but have not been used for social network modeling from smartphone data. Stochastic block structure models [14, 1] aim at finding groups for each individual in a given network. Fu et al. [8] extended these models to dynamic networks by allowing model parameters to change over the global state of the network. The group-topic model by Wang et al. [22] used dynamic group assignment based on text-data where people form groups depending on the actual topic of discussion. In the context of group interaction discovery, these models have two common limitations: first, there is a scalability issue, and second these models focus on global structure of the network rather than finding local interactions of groups. Importantly, the latter point makes block structure models inefficient for extracting local parts of the network that corresponds to specific group interactions. Recently, Dubois et al. [4] proposed to consider individual pairwise interactions rather than the whole network at the same time. This simple model allows to extract local blocks of the network and overcome the drawback of block structure models. However this advantage comes at a price as it cannot infer the latent interactions in a collaborative fashion, taking into account the set of links in the network when assigning interaction type to a pairwise link.

Our model is inspired from topic models such as Latent Dirichlet Allocation [3]. These models were popularized in text analysis for finding relevant latent topics from a corpus and have been recently used in individual activity modeling tasks [7, 13]. We have reformulated and extended this idea for interaction data, where the set of links between users at a given time are assumed to belong to a small number of interaction types. Our work differs from standard topic models on the modeling of the observation space and the nature of the latent class that we want to recover from data (the block structure). As mentioned earlier, the block structure is of high relevance in social network analysis for detecting communities with high intra-community interactions. This is captured naturally by our model by using a conditional independence assumption between observed variables, which also reduces the algorithmic complexity, making our model scalable compared to existing work.

3 GroupUs : A probabilistic model for sensing group interaction.

We present in this section a new probabilistic model for analyzing dyadic interaction data, which are usually represented as a set of links between pairs of users together with the interaction timestamp. In our framework, a user may have multiple links to others for a given timestamp, depending on the number of nearby devices that the Bluetooth scan detected. In this study, we consider directed links, but our method also works with undirected links.

Data representation. The main insight in this work is that to infer the interaction type between two users at a given time, one could exploit not the only links involving the two considered users but also the links between other nearby users. We conduct our analysis with a slice-based approach, where all links within a short period (e.g. 10 minutes) are grouped together, forming a slice of the dynamic network. Duplicate links are removed, which means that there are at most 2 directed links between any two users in a slice. Furthermore, the time of the interaction is also key to deduce the interaction type, hence we include the temporal information in the description of the link. A link i is thus characterized by:

u_i : the head of the link (observer device).

v_i : the tail of the link (observed device).

c_i : the temporal context of the link, a discrete value that describes the corresponding time of the day and day of the week. It always corresponds to one of $24 * 7 = 168$ cases of the 24×7 grid of a weekly calendar.

s_i : the identifier of the time slice that the link belongs to. $s_i \in \{1..S\}$ where S is the total number of time slices.

3.1 The probabilistic model

In many cases, the observed Bluetooth data are noisy. This may be due to technical problems of the sensor as well as the presence of real noise. Considering a group meeting as an example, even if all members attended the meeting, it could happen that some links between members could be lost due to sensor failures. Furthermore, a member of the group could be absent from the group meeting for a few times, we call this "reality noise" of the group meeting.

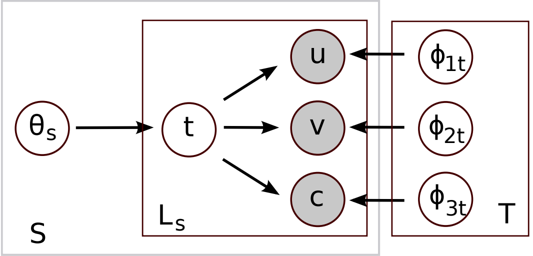


Figure 1: Graphical model.

```
Initialization:
    Draw distribution $\theta_s \sim Dirichlet(\boldsymbol{\alpha})$ for each slice $s$.
    Draw distribution $\phi_t \sim Dirichlet(\boldsymbol{\beta})$ for each interaction type $t$.
For each link of the slice $s$:
    Draw an interaction type $t|s \sim Multinomial(\theta_s)$.
    Draw a first person $u|t \sim Multinomial(\phi_{1t})$.
    Draw a second person $v|t \sim Multinomial(\phi_{2t})$.
    Draw a temporal context $c|t \sim Multinomial(\phi_{3t})$.
```

Table 1: Generative process.

In order to handle such stochasticity of the data, we use a probabilistic approach where observations are represented by random variables. A latent variable model is introduced for capturing emergent patterns from the observations. The graphical model is illustrated in Figure 1, where observed random variables u, v and c are represented by shaded nodes. The latent variable t corresponds to the *interaction type* (a cluster of related links) of the link. The latent interaction types are not explicit but are characterized by model parameters ϕ defining which users are likely to be observer and observed person for each interaction type (ϕ_{1t} and ϕ_{2t}), and which temporal contexts that interactions of a given types are likely to happen (ϕ_{3t}). Finally, θ_s corresponds to the conditional distribution of interaction types given the slice s. Once learned, these hidden variables can be used as a summary of the observation or to generalize the observation. Note that we use a plate representation where each node corresponds to a number of random variables, and the capital letters in the corners stand for the number of variables that the node represents. More specifically, S stands for the number of slices in the data, L_s is the number of links in slice s, and T is the number of interaction types that we want to discover. The generative process for a set of links is shown in Table 1 where we use a Dirichlet prior distribution (with parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) for model parameters θ and $\phi = \{\phi_1, \phi_2, \phi_3\}$. The Dirichlet distribution is the conjugate prior of the Multinomial, which is chosen for algebraic convenience.

Let L be the total number of links, $(\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{s}) = (u_i, v_i, c_i, s_i)_{i=1..L}$ be the set of observed links, and $\mathbf{t} = (t_i)_{i=1..L}$ be the interaction type assignment for each link. The joint probability of $\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{s}$ and $\mathbf{t}$ can be obtained by integrating over hidden parameters:

$$\begin{aligned} &P(\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{s}, \mathbf{t}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \int_{\theta,\phi} P(\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{s}, \mathbf{t}, \theta, \phi; \boldsymbol{\alpha}, \boldsymbol{\beta}) \partial\theta\partial\phi \\ &= \int_{\theta} P(\mathbf{t}|\theta) P(\theta; \boldsymbol{\alpha}) \partial\theta \int_{\phi} P(\mathbf{u}, \mathbf{v}, \mathbf{c}|\mathbf{t}, \phi) P(\phi; \boldsymbol{\beta}) \partial\phi \\ &= \prod_{s=1}^{S} \frac{B(\boldsymbol{\alpha}+\mathbf{n}_s)}{B(\boldsymbol{\alpha})} \prod_{t=1}^{T} \frac{B(\boldsymbol{\beta}+\mathbf{m}_t)}{B(\boldsymbol{\beta})} \frac{B(\boldsymbol{\beta}+\mathbf{p}_t)}{B(\boldsymbol{\beta})} \frac{B(\boldsymbol{\beta}+\mathbf{q}_t)}{B(\boldsymbol{\beta})}. \end{aligned} \tag{1}$$

where $B(.)$ is the multinomial Beta function, $\mathbf{n}_s$ is a T-dimensional interaction type count vector for slice s, and $\{\mathbf{m}_t, \mathbf{p}_t, \mathbf{q}_t\}$ are the observation count vectors of interaction type t. Mathematically, the counts are defined by:

$$\begin{aligned} n_{st} &= \textstyle\sum_{i=1}^{L} \mathbf{1}(s_i = s \wedge t_i = t), & m_{tu} &= \textstyle\sum_{i=1}^{L} \mathbf{1}(t_i = t \wedge u_i = u), \\ p_{tv} &= \textstyle\sum_{i=1}^{L} \mathbf{1}(t_i = t \wedge v_i = v), & q_{tc} &= \textstyle\sum_{i=1}^{L} \mathbf{1}(t_i = t \wedge c_i = c). \end{aligned} \tag{2}$$

where $\mathbf{1}(.)$ denotes the indicator function. Note that the integration over hidden parameters θ and ϕ in Eq. 1 can be computed efficiently since we use conjugate priors in each elementary distribution. For space reasons, the mathematical derivations have been omitted from the paper but are available in a supplementary appendix.

3.2 Inference and parameter estimation

The proposed probabilistic model defines relations between observed variables and latent variables. These relations are parameterized by ϕ and θ, for instance ϕ_{1t} tells which users are likely to appear as observer in the interaction of type t, ϕ_{2t} tells which users are likely to be observed in the interaction of type t, and ϕ_{3t} tells which time slots in

Algorithm 1 GroupUs learning algorithm

```
1: **input:** interaction links $\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{s}$
2: **output:** model parameters, $\theta$, $\phi$, and interaction type for each link,$\mathbf{t}$.
3: **initialization:** Randomly assign interaction type $t_i$ for each link $i$
4: Compute the count $n_{st}, m_{tu}, p_{tv}, q_{tc}$ according to Eq. 2
5: **while** not converged **do**
6:   **for** each link $i$ **do**
7:     $s := s_i$. Decrement the counts: $n_{st_i}$--; $m_{t_i u_i}$--; $p_{t_i v_i}$--; $q_{t_i c_i}$--;
8:     Sample the interaction type assigment $t_i$ according to
           $P(t_i = t|\mathbf{t}_{\neg i}, \mathbf{u}, \mathbf{v}, \mathbf{c}; \alpha, \beta)$
           $\propto (\alpha + n_{st}) \frac{\beta + m_{tu_i}}{\sum_u (\beta + m_{tu})} \frac{\beta + p_{tv_i}}{\sum_v (\beta + p_{tv})} \frac{\beta + q_{tc_i}}{\sum_c (\beta + q_{tc})}$
9:     Updating the counts: $n_{st_i}$++; $m_{t_i u_i}$++; $p_{t_i v_i}$++; $q_{t_i c_i}$++;
10:  **end for**
11: **end while**
12: Compute $\theta$, $\phi$ according to Eq. 4
```

the weekly calendar interactions of type t are likely to occur. Discovering the interaction type is the process of fitting model parameters to observed data, and then visualizing the learned patterns based on the model parameters.

The problem of finding optimum model parameters is intractable in general. However, a wide variety of approximation techniques can be used, including Laplace approximation, variational approximation, and Markov chain Monte Carlo (MCMC). In this work, we learn the model using collapsed Gibbs sampling [10], which samples the posterior distribution $P(\mathbf{t}|\mathbf{u}, \mathbf{v}, \mathbf{c}; \boldsymbol{\alpha}, \boldsymbol{\beta})$ from the conditional distribution $P(t_i = t|\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{t}_{\neg i}; \boldsymbol{\alpha}, \boldsymbol{\beta})$ where $\mathbf{t}_{\neg i}$ denotes the type assignment for all links but i^{th} link. Although our method works for general Dirichlet priors, we assume symmetric Dirichlet priors to simplify the presentation, and we denote the scalar value of elements of the two vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}$ by α, β. Omitting derivation details for space reasons, the Gibbs sampling equation can be written by :

$$\begin{aligned} & P(t_i = t|\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{t}_{\neg i}; \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto \\ & \quad (\alpha + n_{s_i t}^{\neg i}) \frac{\beta + m_{tu_i}^{\neg i}}{\sum_u (\beta + m_{tu}^{\neg i})} \frac{\beta + p_{tv_i}^{\neg i}}{\sum_v (\beta + p_{tv}^{\neg i})} \frac{\beta + q_{tc_i}^{\neg i}}{\sum_c (\beta + q_{tc}^{\neg i})}, \end{aligned} \tag{3}$$

where $n_{st}^{\neg i}, m_{tu_i}^{\neg i}, p_{tv_i}^{\neg i}$ and $q_{tc_i}^{\neg i}$ are the counts for $n_{st}, m_{tu_i}, p_{tv_i}$ and q_{tc_i} without considering the link i. For instance, $n_{st}^{\neg i} = \sum_{j \neq i} \mathbf{1}(s_j = s \; and \; t_j = t)$. Given the interaction type assignments for all links, we can estimate the model parameters as follows:

$$\begin{aligned} \theta_{st} &= \frac{\beta + n_{st}}{\sum_{t'} (\beta + n_{st'})}, & \phi_{2tv} &= \frac{\beta + p_{tv}}{\sum_{v'} (\beta + p_{tv'})}, \\ \phi_{1tu} &= \frac{\beta + m_{tu}}{\sum_{u'} (\beta + m_{tu'})}, & \phi_{3tc} &= \frac{\beta + q_{tc}}{\sum_{c'} (\beta + q_{tc'})}. \end{aligned} \tag{4}$$

The full learning algorithm is summarized in Algorithm 1.The algorithm starts with random interaction type assignments $\mathbf{t}$ for the set of links. Then, the interaction type for each link is resampled iteratively until convergence. We maintain the counts $n_{st}, m_{tu}, p_{tv}, q_{tc}$ over iterations, which are updated after each sampling step so that each iteration requires only a few computations. Note that in the equation at line 8 - Algorithm 1 is equivalent to sampling equation in Eq. 3, since the counts were decreased just before the sampling step and correspond to the counts without considering the link i. After the sampling process, the algorithm outputs the interaction type for each link as well as estimates of the parameters θ, ϕ. The overall complexity of Algorithm 1 is $O(KLT)$ where K is the number of sampling iterations (we set $K = 100$ in our experiments). Compared to previous works [22, 17] for which the complexity grows superlinearly (quadratically or even exponentially) with the problem size, GroupUs scales well with the number of links and the number of interaction types, and hence it can learn from large-scale data in linear time.

Algorithm 2 Finding prominent users.

```
**Input:** $P(u|t)$
**Output:** most prominent users.
Sort users by $P(u|t)$
**for** $n = 1$ to #users **do**
  Compute Kullback Leibler divergence $KL(n)$ between:
  $P_{proto}^{n}$ : the prototype distribution with $n$-top participants
  $P(u|z)$ : the input distribution
**end for**
$n^* = \text{argmax } KL(n)$
Return the list of top $n^*$ users.
```

3.3 Interpreting interaction types

Our method represents interaction types in a probabilistic fashion. In most applications, one may want to know what a discovered interaction type represents in real life. This section shows how we interpret the learned model by considering two fundamental questions for each discovered interaction type: (1) Who are involved?, (2) Is the interaction happening at work?. This is discussed in the following.

Inferring the participants of a given type of interaction. The learned parameter ϕ_{1t} tells us the probability of observing user u given the type of interaction t, and thus we can answer the first question based on this. Due to the variability of group size, we need a method to extract the top users who are likely to participate in a given interaction type. A simple method is to take the minimal set of top users who cover at least $X\%$ (e.g. 90%) of the probability mass. However, this method is quite sensitive to the threshold and might fail to find the relevant members of a group.

Our solution is described in Algorithm 2. The algorithm takes as input the conditional distribution over users given an interaction type $P(u|t)$ and outputs the list of prominent users as follows. First, the list of users is sorted by their probabilities. Then the algorithm finds the best segmentation of the list of users into participants and non-participants. As scoring function for a given segmentation with n prominent users, we use Kullback Leibler divergence between a prototype distribution with n users and the input distribution. The prototype distributions are defined based on the ideal case where the top n users have equal probabilities, and the probabilities of all others are zero. Formally:

$$P^n_{proto}(u) = \begin{cases} 1/n & \text{if } u \text{ belong to the top } n \text{ users} \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Office interaction vs personal interaction. A person may have many social interaction types in their daily life. Based on the temporal context, we can infer the meaning of the discovered interaction types. For instance, a work interaction should mainly occur on working hours and not on weekend days. Implementing this idea in our model is particularly easy given the learned parameters. Let H be the set of office-hour time slots, i.e. from 9am-6pm Monday to Friday. The probability that an interaction of type t occurs during working time can be computed as:

$$P(H|t) = \sum_{c \in H} P(c|t) = \sum_{c \in H} \phi_{3tc}. \quad (6)$$

Clearly, if $P(H|t)$ is high then it is likely that the interaction type t corresponds to an office interaction. We define an *office interaction* as an interaction type t for which $P(H|t) > T_0$ where T_0 expresses the certainty of t being an office interaction. As we will see in Section 5, this information is helpful to visualize data or for further analysis.

4 Large-scale proximity data

We use data from the Lausanne data collection campaign, which uses a server-client architecture built for the Nokia N95 8GB smartphone in order to collect data [15]. The software client was designed to detect and record Bluetooth scans, storing the logs in the phone's memory. The client was installed in the phone and runs in the background in a non-intrusive way, starting automatically at startup, and recording data on a 24/7 basis as long as the phone is on. The log data are then uploaded daily to a server, typically done at night, via a user-defined wifi connection.

To allow for real-life usage with respect to battery consumption, the client is designed using a state machine architecture [15], which adapts the sensor sampling rate depending on the inferred phone state (e.g. static, moving, etc). The data are recorded continuously with the only restriction of having to recharge the phone once a day (typically done during nights). The mobile phone scans to detect nearby Bluetooth devices every 1-3 minutes.

We used Bluetooth data recorded continuously over 12 months of real-life on a set of 40 volunteer users (also called observers in the following). 24 of the users are colleagues who work for a mid-size organization and occupy a dozen office spaces in a building, spanning from single-person rooms to a lecture room. The remaining 16 users are family members from the 24 users. All volunteers were compensated for any costs associated to the data collection. All information about the users has been anonymized, and only basic information about group membership has been kept for experiments. Users carried their device as their actual (and only) phone and therefore used them in real conditions. The data was recorded from September 2009 to August 2010 and corresponds to more than 2 million non-empty Bluetooth scans.

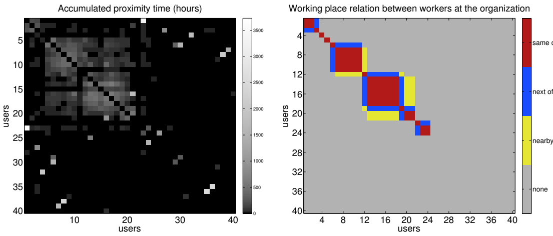


Figure 2: Left: Accumulated proximity time between users according to BT sensor. Users 1-24 are co-workers, users 25-40 are some of their family members. Right: Working place relation between workers in the organization.

5 Experimental results and discussion

We begin by presenting a global view of the data. Figure 2(left) shows the accumulated proximity time between users in the population according to the Bluetooth sensor. The 24 workers in the organization are numbered from 1 to 24 and ordered by the office they nominally occupy.

Figure 2(right) shows the working place relation between workers according to four cases: i) co-office workers (*same office*), whose phones should detect each other quite often; ii) workers in adjacent offices (*next office*) are very likely to detect each other, but depending on their relative position; iii) workers in nearby offices (*nearby*) but not as close as the two first cases; and (iv) none of the above. These plots reflect the fact that in reality co-workers have high chance to see each other if their offices are close, and people spend more time with their relatives than with co-workers.

5.1 Robustness of Bluetooth as proximity sensor

Bluetooth data is quite noisy, it often happens that a Bluetooth device does not detect all nearby devices in a scan. We present in this section a basic analysis of robustness of Bluetooth proximity sensor in a real condition.

We start by considering a subset of the data consisting of the weekly meetings of a group of 10 members for whom we know the exact meeting schedule over the recording period. Based on this, we would like to estimate the rate at which the phone of each person successfully detects other participants. To this end, we divide each group meeting into time slices of short duration, and draw links between people within each time slice. The ground truth for each group meeting is simply a fully connected graph using the people present at the meeting. We consider both directed and undirected graphs for the evaluation:

- An asymmetric link from user u to user v corresponds to the fact that u observed v during the slice.

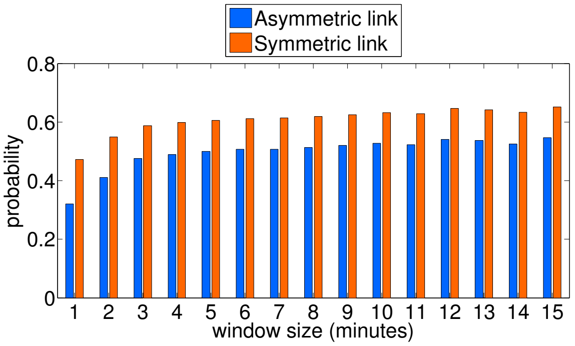


Figure 3: Proximity detection rate of Bluetooth sensor for group meeting data.

- A symmetric link between u and v corresponds to the fact that u observed v or v observed u.

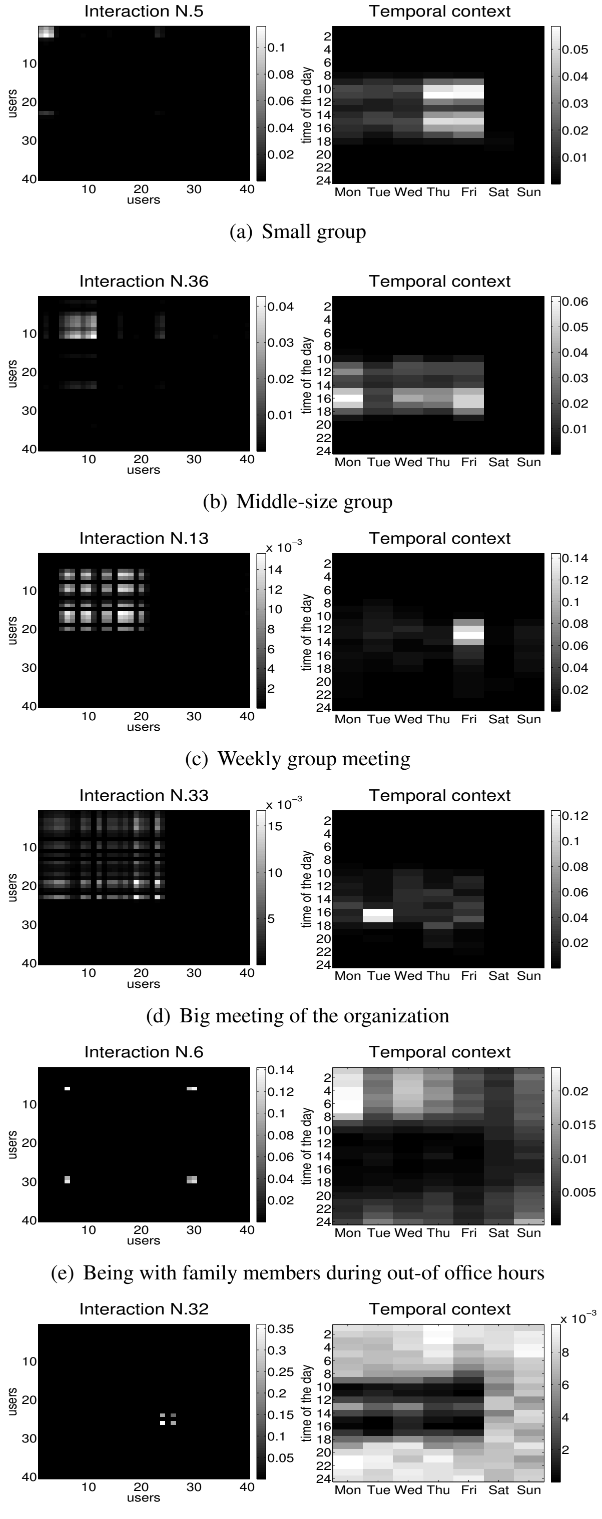
Figure 3 reports the rate of link detection as a function of time slice duration. As can be seen, the duration of the slice is crucial as increasing the "observation" period also increases the rate of link detection. The plot also suggests to consider a slice duration of at least 5 minutes in order to obtain near optimal link detection rate with Bluetooth sensor. Looking at the result for asymmetric link, we found that the Bluetooth sensor has a proximity detection rate of 0.5 at 10 minutes time slice. The rate can be improved by considering Bluetooth data from two users, this corresponds to the case of symmetric link where the proximity detection rate are roughly 25% better than the case of asymmetric link. Slices of 10 minutes are therefore a conservatively good choice.

We continue with the analysis with GroupUs algorithm. First we highlight some typical examples of discovered interaction types, we then study the evolution of interactions over time in real events. Finally, we evaluate objectively the predictive performance of GroupUs.

5.2 Discovered interaction types

We ran GroupUs with $T = 40$ in order to capture a few family interactions and office interactions, and we set $\alpha = 0.1, \beta = 0.1$ and $T_0 = 0.5$. Starting from random initialization, the algorithm refines model parameters in each Gibbs sampling iteration. We observed that the convergence is reached after about 30 iterations (see movie in supplementary marterial).

Using the classification method in Section 3.3, we found 15 office interaction types and 25 family interaction types. We start with some examples of discovered interaction types, visualized with the pairwise matrix of interaction (i.e. ${\phi_{1t}}^{\intercal}\phi_{2t}$) and the distribution of temporal context over the weekly calendar (ϕ_{3t}) in Figure 4. The first two interaction types (a-b) correspond to working place interactions, where these groupings (the first one involving users 1-3, and the



(a) Small group

(b) Middle-size group

(c) Weekly group meeting

(d) Big meeting of the organization

(e) Being with family members during out-of office hours

(f) Being with family member, including having lunch

Figure 4: Typical discovered interactions visualized with pairwise interaction matrix (${\phi_{1t}}^{\intercal}\phi_{2t}$) between users and the distribution of temporal context (ϕ_{3t}).

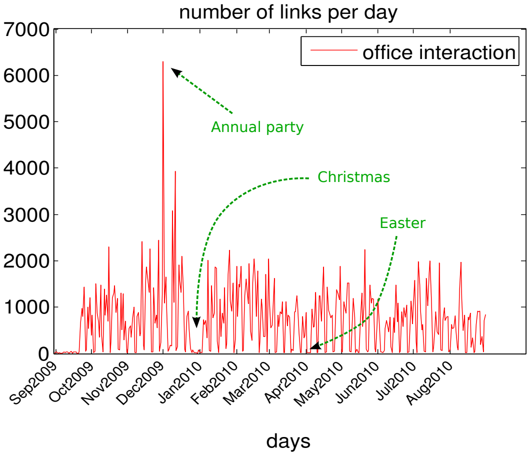


Figure 5: Evolution of office interaction over time. Some emergent events such as holidays can be observed.

second one involving users 5-11) clearly correspond to the working place ground truth (compare with Fig. 2(right). Note that these interactions spread over working times but have low probability at lunch time, which indicate that these coworkers do not eat together often. The low probabilities for some days of the week reflect the fact that some workers telecommute and so do not come to the organization every day.The two next interaction types (c-d) are interesting as more people are involved and the temporal context reveals that these are not daily interactions. Figure 4(c) corresponds to a weekly group meeting on Fridays followed by lunch in reality. Note that this group is spread over 4 different offices, and some of its members appear as most prominent users of other discovered interaction types (e.g. Figure 4(b)) which highlights the probabilistic advantage of GroupUs. The interaction type in Figure 4(d) reflects a weekly big meeting of the organization on Tuesday afternoons where all members are expected to attend. This is an example of a highly localized type of event that is correctly inferred by GroupUs. Note that some occasional interactions between workers are also assigned to this type of global interaction, explaining why there is some "noise" in the weekly calendar. Finally, we show two examples of family interaction in Figure 4(e-f). Note that, while many family interaction types were discovered, they have similar temporal context and differ mainly in the set of involved users.

5.3 Interaction over time

Although our method does not take into account absolute calendar temporal information (that is, beyond weekly schedule), we can nonetheless study the evolution of proximity interactions over time. Figure 5 plots the number of work interactions for each day of the data collection period. Office interactions were inferred according to the method described in Section 3.3. As can be seen, we can see some

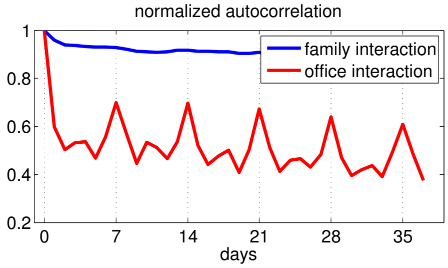


Figure 6: Autocorrelation of family interaction (blue) and office interaction (red). The x-axis corresponds to days, y-axis corresponds to autocorrelation value

emergent events from the plot such as Christmas vacation (Dec 23 - Jan 4), Easter weekend(April 2 - 5), and other holidays. All of these are characterized by lower values. The plot also shows big events of the organization. For instance, the highest pick at December 1, 2009 was actually the annual party of the organization.

We also compare the periodicity of office interactions with personal interactions. Figure 6 shows the autocorrelation of these two kinds of interactions for the population of 24 workers. As can be seen, the weekly periodicity of work interaction is very clear while weekly periodicity of family interaction is quite weak. Note that this analysis can be applied to each interaction type to distinguish between periodic group interaction (such as weekly meetings) and occasional group interactions (examples not shown for space reasons). These results confirm some findings by Eagle and Pentland [6] but on a different organization and with a robust probabilistic approach that significantly reduces the presence of noise in Bluetooth data.

5.4 Predictive performance

In this section, we evaluate our method by studying the predictive performance unseen data, an very important task in context-aware mobile application. Our main goal is to validate the learning capability of the proposed model by studying the likelihood on unseen data. For this reason, we do not consider a real-time prediction task and compare with predictive models such as ARIMA. The last two months of data are for testing, and we learn the model with different training sets, varying from 2 (last) months to 10 months of data from all users.

As a baseline, we adapted the Marginal Product Mixture Model (MPMM) which was proposed recently for analyzing phone call data [4]. As discussed in Section 2, this model also aims at finding latent classes of interaction, but it can only infer the latent class from a single link. On the contrary, our model infers the interaction type of a user based on his interactions with others and also based on the interactions among other people in the group.

Figure 7 plots the test log-likelihood for different training sizes. As can be seen, GroupUs outperforms clearly

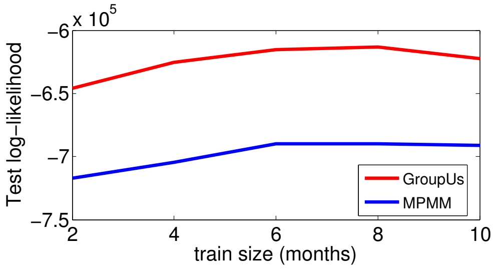


Figure 7: Log-likelihood on the test data.

the MPMM model in term of predictive performance thanks to the more accurate modeling assumption. In general, the more data the more accurate GroupUs is, but note that using "too old" data (the case of 10 months) might not help improving predictive performance.

6 Conclusion

We proposed a new probabilistic model for discovering interaction types from large-scale proximity data. We conducted our analysis on Bluetooth proximity data involving 40 users in which 24 of them are co-workers at the same organization. We objectively evaluated our method by studying predictive performance, showing a significant advantage over a recently proposed model. We are interested in extending this work to include additional nearby Bluetooth devices in the analysis, thus considering an extended population of volunteer data providers and others. While more data provides the opportunity for more accurate learning of social context, one key challenge will be to work with unknown devices. Using only Bluetooth data, we showed that GroupUs can infer relevant interactions such as office interactions and family interactions without any supervision. One could incorporate other type of data (e.g. GPS) into GroupUs in order to enrich the context of the interaction, therefore providing more details on the discovered interaction types.

Acknowledgments

This work was funded by Nokia Research Center Lausanne (NRC) through the LS-CONTEXT project.

References

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, 2008.

[2] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal Ubiquitous Computing*, 7:275–286, 2003.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[4] C. DuBois and P. Smyth. Modeling relational events via latent classes. In *Proc. KDD*, pages 803–812, 2010.

[5] N. Eagle, A. S. Pentland, and D. Lazer. Inferring social network structure using mobile phone data. *PNAS*, 106(36):15274–15278, 2009.

[6] N. Eagle and A. (Sandy) Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, 2006.

[7] K. Farrahi and D. Gatica-Perez. What did you do today?: discovering daily routines from large-scale mobile data. In *ACM Multimedia*, pages 849–852, 2008.

[8] W. Fu, L. Song, and E. P. Xing. Dynamic mixed membership blockmodel for evolving networks. In *Proc. ICML*, pages 329–336, 2009.

[9] J. Gips and A. Pentland. Mapping human networks. In *Proc. Pervasive Computing and Communications*, pages 159–168. IEEE Computer Society, 2006.

[10] T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101(Suppl. 1):5228–5235, April 2004.

[11] C. A. Hidalgo and C. Rodriguez-Sickert. The Dynamics of a Mobile Phone Network. *Physica A*, 387(12):3017–3024, Feb 2008.

[12] J. Hightower, S. Consolvo, A. Lamarca, I. Smith, and J. Hughes. Learning and recognizing the places we go. In *Proc. UbiComp*, pages 159–176, 2005.

[13] T. Huynh, M. Fritz, and B. Schiele. Discovery of activity patterns using topic models. In *Proc. Ubiquitous computing*, pages 10–19. ACM, 2008.

[14] N. K. and S. T. A. B. Estimation and prediction for stochastic blockstructures. *JASA*, 96:1077–1087, September 2001.

[15] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila. Towards rich mobile phone datasets: Lausanne data collection campaign. In *Proc. ICPS*, Berlin, 2010.

[16] L. Liao, D. Fox, and H. Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *International Journal of Robotics Research*, 26, 2007.

[17] S. Mardenfeld, D. Boston, S. Juan Pan, Q. Jones, A. Iamnitchi, and B. Cristian. Gdc: Group discovery using co-location traces. In *SCA*, 2010.

[18] D. O. Olguin, B. N. Waber, T. Kim, A. Mohan, K. Ara, and A. Pentland. Sensible organizations: Technology and methodology for automatically measuring organizational behavior. *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, 39:43–55, 2009.

[19] E. O'neill, V. Kostakos, T. Kindberg, A. Schiek, A. Penn, D. Fraser, and T. Jones. Instrumenting the city: Developing methods for observing and understanding the digital cityscape. In *Proc. UbiComp*, pages 315–332, 2006.

[20] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen. Contextphone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing*, 4(2):51–59, 2005.

[21] M. Terry, E. D. Mynatt, K. Ryall, and D. Leigh. Social net: using patterns of physical proximity over time to infer shared interests. In *Proc. CHI*, pages 816–817, 2002.

[22] X. Wang, N. Mohanty, and A. Mccallum. Group and topic discovery from relations and their attributes. In *Proc. NIPS*, pages 1449–1456, 2006.

[23] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[24] D. Wyatt, T. Choudhury, and H. Kautz. Capturing spontaneous conversation and social dynamics: A priv. sensi. data collec. effort. In *Proc. ICASSP*, 2007.