Human Interaction Discovery in Smartphone Proximity Networks

Trinh Minh Tri Do · Daniel Gatica-Perez

Abstract Since humans are fundamentally social beings and interact frequently with others in their daily life, understanding social context is of primary importance in building context-aware applications. In this paper, using smartphone Bluetooth as a proximity sensor to create social networks, we present a probabilistic approach to mine human interaction types in real life. Our analysis is conducted on Bluetooth data continuously sensed with smartphones for over one year from 40 individuals who are professionally or personally related. The results show that the model can automatically discover a variety of social contexts. We objectively validated our model by studying its predictive and retrieval performance.

1 Introduction

Social interaction plays an important role in our daily lives as a part of most of our activities (e.g., working, at home, doing sports, etc.). In sociology, social network analysis has become a key tool to analyze interaction between people at a large scale [35,39]. This type of analysis calls for efficient data collection methodologies,

T.M.T. Do
Idiap Research Institute, Switzerland
Tel.: +41 27 721 77 51
Fax: +41 27 721 77 12
E-mail: do@idiap.ch

D. Gatica-Perez
Idiap Research Institute, Switzerland
École Polytechnique Fédérale de Lausanne, Switzerland
Tel.: +41 27 721 77 33
Fax: +41 27 721 77 12
E-mail: gatica@idiap.ch

which are challenging task themselves. Each data type reflects some specific aspect of social behavior and contributes to the global understanding of human interaction. The amount of data can vary from small amounts collected manually via questionnaires in the early days of social science [34] to global networks of people collected via phones and the internet [26,28].

Recently, smartphones have emerged as a feasible device to sense daily life activities and events, including social interactions. These ubiquitous, programmable devices provide access to behavioral and contextual information through various built-in sensors [11,33] (e.g., GPS, accelerometer, Bluetooth, etc.). The main advantage of smartphones over dedicated devices for data collection and analysis also lies in the fact that most people do not have to carry an additional device (and the associated burden) and hence usually do not change their normal behavior. Further, these devices create an opportunity for continuous sensing of human behavior for long periods of time.

In this work, we have used Bluetooth sensors in smartphones to sense the proximity network between people over a long period of time. Bluetooth proximity is an acceptable approximation for social interactions, since people in proximity are more likely to interact with each other. Technically, Bluetooth-based proximity has advantages including low battery cost, the ability to work in both indoor and outdoor environments, and its availability in most phones and other mobile devices. Furthermore, it is perceived to be less privacy-sensitive than other data types such as audio and location, and people are happy to connect to others via Bluetooth to share both their presence and data, by ac-

tivating the "discoverable mode" in their devices. Bluetooth data is inherently relational, unlike other sensor data types like GPS.

Bluetooth proximity networks have various limitations when sensing social interactions. First, Bluetooth proximity data is noisy, since devices might sometimes fail to detect all nearby devices. This is more likely to happen when many devices are active at the same time and place. Second, proximity does not always mean actual interaction (e.g., in public transport). Finally, Bluetooth devices do not always have a one-to-one correspondence to individuals: Although phones are usually carried by a single owner, people sometimes forget them or put them aside [32], and some users might carry multiple Bluetooth-equipped devices (phones, laptops, etc).

We conduct our analysis on a large-scale Bluetooth proximity data, with the hypothesis that the longitudinal aspect of the collected data will overcome some of the limitations mentioned above, and propose an appropriate mining algorithm to discover relevant recurrent patterns of interaction between people in their social network. As people usually form groups in real life, we focus our analysis on group interaction rather than on pairwise relations.

The main goal of this work is to determine the social context of a given user based on the state of the proximity network around him and the current time. We develop an unsupervised approach for automatic interpretation of social contexts, which are not explicitly available but that can be discovered from the data. To this end, we introduce the concept of an *interaction type* for the interaction links that might exist among people, which defines the latent meaning of the link. Basically, an interaction type is characterized by who is present when; for example, an interaction type called "group meeting" might consist of the same same group of six people that from 10 to 11 every Tuesday morning, while the "being at home" interaction type could refer to the interaction of a person with his family during non-working hours.

From the above examples, clearly, these interaction types are user/group-specific (two groups of people might have two different group meetings). The discovery and the recognition of these interaction types could be useful for personalizing applications. First, we could predict who a user meets and when directly from the set of discovered interaction types. This recognized interaction type could then be used as an input to context-aware applications, for which user interest or user behavior depends on the social context. The set of interaction types of a person could also be used to infer his personality. Finally, the discovered interaction types

can be viewed as a summary of how people interact together, and facilitate the visualisation of interaction data.

In this paper, we describe a probabilistic framework to discover interaction types. The framework uses longitudinal, real-life Bluetooth data collected from a population of smartphone users as input. Our work makes the following three contributions. First, we discuss our model for interaction type discovery from proximity data (referred to as GroupUs), which utilizes the information from long-term observations of everyday proximity within a model, that accounts for uncertainty both in sensing and in the group interactions themselves. Second, our analysis is performed on a data set collected with smartphones, which encompasses one year in the life of 40 individuals. Finally, we analyze GroupUs' performance in detail. We show that the model can indeed infer both a set of meaningful interaction types and the individuals who are more prominent in those interactions, and compare it against an existing method using an objective evaluation procedure.

The structure of the paper is as follows. Section 2 reviews related work. We present the data collection framework and some basic analysis in Section 3. We present an overview of our method in Section 4 and the technical details is provided in Section 5. Our findings and validation results are presented in Sections 6, 7 and 8. Finally, Section 9 provides concluding remarks.

2 Related work

The idea of using Bluetooth as a way of detecting social interaction is not new in ubiquitous computing. As two examples, Terry et al. [36] investigated the use of pairwise proximity patterns over time to identify interests shared by individuals. In an urban context, O'Neill et al. proposed to use the number of detected BT devices in an environment as an indicator of the associated human density [31]. More recently, other related work has appeared, often under the umbrella term of reality mining [11]. Most of this work has used mobile phones to sense longitudinal human activity, as proposed by Raento et al. [33] and Eagle and Pentland [11].

Sensor data reflecting real face-to-face interaction has increasing value for social network research. Sensors used in the literature include Bluetooth, RFID, infrared, microphones, and cameras; each sensor presents advantages and limitations, especially regarding the accuracy in capturing real interaction. In the case of Bluetooth, Clauset et al. [5] analyzed a BT proximity network of a population recorded over nine months, and demostrated that quantities like periodicity can be inferred. Eagle et al. [10] analyed a network constructed

from BT links and phone call logs to identify friendship networks. Mardenfeld et al. [25] also studied a BT network to discover groups. Other works have relied on other mobile sensors, like infrared, and microphones, to address the limitations of Bluetooth to detect real face-to-face proximity, rather than just detecting people sharing an office or a large space [15,40,30]. As a tradeoff for the improved spatial resolution, many of these studies imposed restrictions on how people wear these devices. As a new possibility, it has been shown that the distance between wireless devices can be estimated by exploiting low level signals, and the spatial resolution can potentially be improved [3].

Also connected to our work is the problem of discovering places, which has been widely studied in mobile and ubiquitous computing using several types of location data [2,17,24,29]. The relation between places and interaction is evident in everyday life: specific interactions happen at specific places. This knowledge could clearly be used for place and event prediction purposes [23]. In our case, while we only rely on Bluetooth, the model presented here could, in an extension, integrate location data to relate places to interaction types as part of the discovery process.

Social network analysis for relational data is also an active topic in data mining and machine learning [39,21]. Some methods have been proposed to extract groups, which are mainly based on discovering block structure from interaction, but these methods have not been used for social network modeling from smartphone data. Using probabilistic framework, stochastic block structure models [21,1] aim at finding groups for each individual in a given network. To analyze dynamical networks, Fu et al. [13] extended these models by allowing model parameters to change over the global state of the network. Another approach for modeling dynamical network has been proposed in [38] for relational and text data, in which the group assignment are dynamical, 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. In a recent work, Dubois et al. [9] simplify the framework by considering 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 pairwise interactions were assumed to be independent

and identically distributed. This assumption, however, is not realistic for social network analysis applications in which people interact in group.

The GroupUs model is inspired from topic models like the Latent Dirichlet Allocation, proposed by Blei et al. [4]. LDA is a highly popular tool in text analysis to extract semantic topics from text corpora, and recently used in human activity modeling from mobile sensors by Farrahi et al. [12] and Huynh et al. [19]. We have extended these ideas for interaction data, where the set of links between people in a network, within a relatively short period of time, are assumed to correspond to a hidden interaction type (taken from a small number of possible types). Our work differs from standard LDA, as the observation space and the nature of the latent class to be recovered from data are both relational. As mentioned earlier, block structures are relevant in social network analysis for detecting communities. Our model captures these block structures by using a conditional independence assumption between observed variables, which also reduces the algorithmic complexity.

Our model was originally proposed in Do and Gatica-Perez [8] and validated on a dataset with 40 users. In this paper, we discuss it in more details, evaluate it thoroughly, and also include additional nearby Bluetooth devices in the analysis. In an earlier work, we also proposed a model which focuses on discovering emergent group structure of proximity networks [7]. The main ideas are that dynamical networks have a limited number of emergent structures, and that each structure corresponds to a mapping from the set of people in the network to the set of latent groups, in which group members have high probability to interact with each other. While this approach is a direct way to group people, the global network has potentially an exponential number of grouping modes (with respect to the number of nodes), and thus the model scalability is limited. To circumvent this problem, GroupUs does not model explicitly the global structure of the network, but focuses on extracting specific group interactions separately.

3 Large-scale proximity data and basic analysis

The dataset in this analysis stems from the Lausanne data collection campaign, which uses a server-client architecture built for the Nokia N95 8GB smartphone to collect data [22]. The software client was designed to detect and record Bluetooth scans approximately every 1-3 minutes, and store the logs (MAC addresses of nearby Bluetooth devices together with the timestamp) in the phone memory. This client which could run in the background in a non-intrusive manner was installed in the phone. The client started automatically at startup,

and recorded data continuously as long as the phone was powered on. The logged data was then uploaded daily to a server, typically at night, by connecting to a wifi network.

In order to optimize battery consumption, the client was designed using a state machine architecture [22], which adapted the sensor sampling rate based on the inferred phone state (e.g., static, moving, etc). Due to this the data was recorded continuously with the only restriction of having to recharge the phone once a day (which was typically done during nights).

We use Bluetooth data recorded continuously over 12 months on a set of 40 volunteer users (also called observers in the following discussions). 25 of these users were colleagues who worked for a mid-size organization and occupied a dozen office spaces in a building, spanning from single-person rooms to a lecture room. The remaining 15 users were family members of the 25 users. All volunteers were compensated for any costs associated to the data collection. Information about the users was anonymized, and only basic information about group membership was 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 October 2009 to the end of September 2010. This corresponds to more than 2 million non-empty Bluetooth scans.

Unlike previous works on Bluetooth proximity data, which mainly focused on pairwise interactions, we address the problem of mining group interaction behavior among people in daily life. In the next subsections, we address three specific aspects of our Bluetooth proximity data.

3.1 Block structure.

We expect BT proximity data to exhibit features similar to other types of data used to sense social network. One of the key concepts in social network analysis is the block structure, which is used to explain how people form groups. A matrix has a block structure if we can group its rows and columns into groups of similar vectors. For example, if there are N groups of rows and M groups of columns and the rows and columns are ordered by group, then the matrix has a structure of $N \times M$ blocks and the values in each block are similar. The problem of grouping people in social network can be expressed as finding the block structure of the interaction matrix, where we expect that people in the same group have similar vectors of interaction. For example, Airoldi et al. showed that block structure can be used to model the like-dislike matrix between monks in a monastery [1]. As a basic analysis, we show the
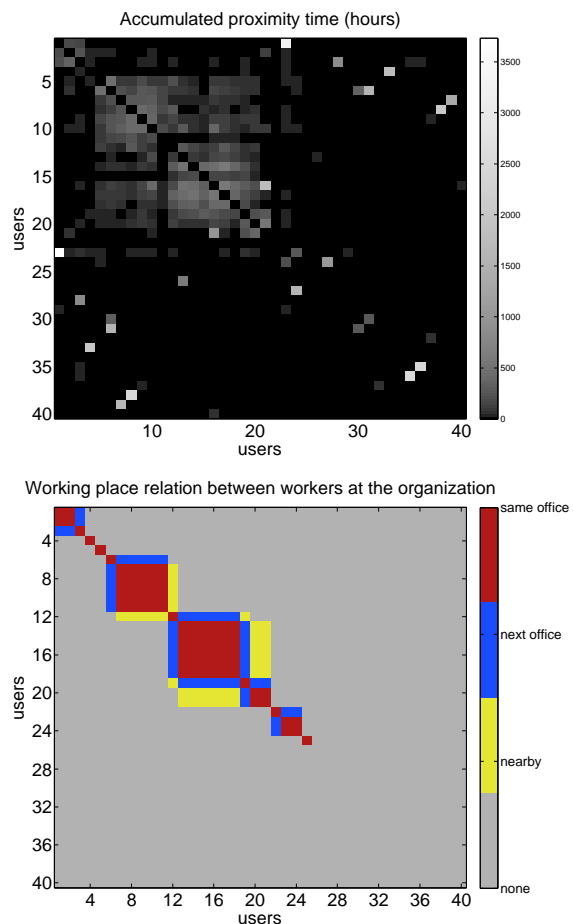

Fig. 1 Top: Accumulated proximity time between users according to BT sensor. Users 1-25 are co-workers, users 26-40 are some of their family members. Bottom: Working place relation between workers in the organization.

pairwise proximity time matrix between smartphone users in Figure 1 (top). The 25 workers in the organization are numbered from 1 to 25 and ordered by the office they nominally occupy. Figure 1(bottom) shows the working place relation between workers according to four cases: i) office co-workers (*same office*), whose phones should detect each other quite often; ii) workers in adjacent offices (*next office*) are likely to detect each other depending on their relative position; iii) workers in nearby offices (*nearby*), not as close as the two first cases, might detect each other sometimes; and (iv) none of the above. These plots reflect the fact that in reality co-workers have high chance of seeing each other if their offices are close, and that people spend more time with their relatives than with co-workers.

Some emergent block structures can be observed in Figure 1(top), which reflects the fact that people usually interact in groups. As the set of users was ordered by office, for the interactions between co-workers the
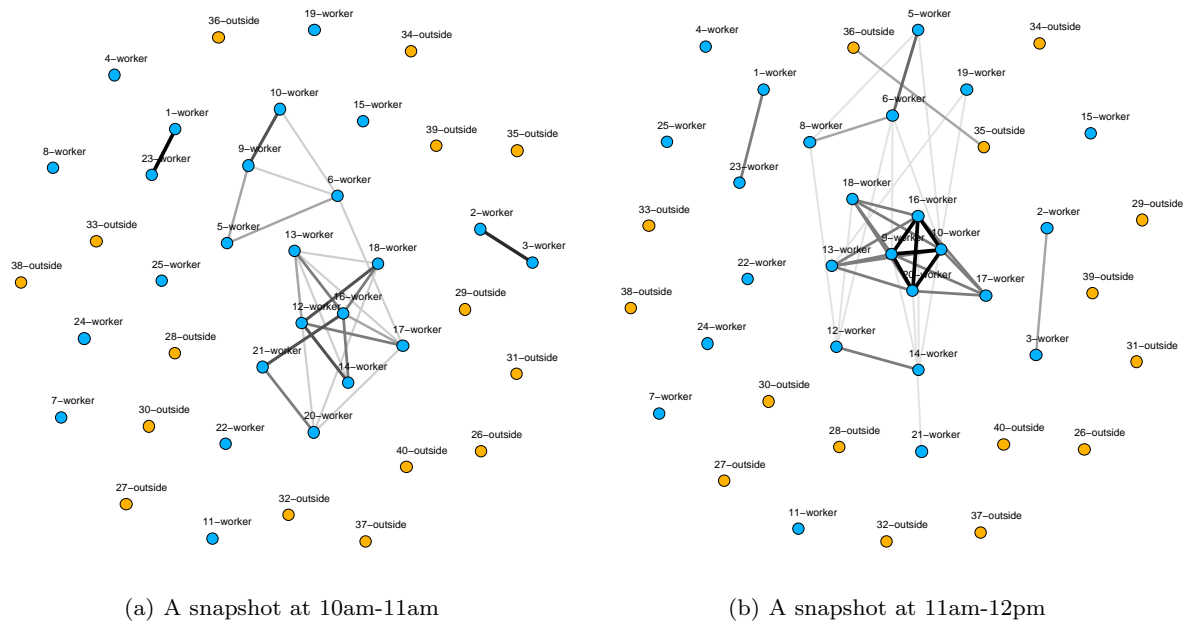
(a) A snapshot at 10am-11am   (b) A snapshot at 11am-12pm

Fig. 2 The proximity network of co-workers and others. Links represents proximity between people during the one-hour period, and the intensity of the link represents the amount of time people were in proximity.

block structure is relatively easy to detect visually, reflecting cases of people working in the same office who spent much time together. Interactions between people and their family members are also strongly highlighted, and would also form blocks if ordered appropriately. Besides, there are also other interaction patterns which do not dominate in terms of total amount of time but are key to understand the social behavior of the organization(e.g., project meetings, having lunch together, etc.). These patterns are relatively subtle in the statistics in Figure 1, and require a more sophisticated approach for discovery.

3.2 Dynamical network.

As human and social behaviors are strongly conditioned by time, it is important to consider the temporal information in the analysis. The Bluetooth proximity data can thus be viewed as a dynamical network that changes over time, where each link (i.e., each pairwise interaction) has a start-time and an end-time. This temporal dimension plays an important role for understanding the actual semantic meaning of the link, which is not observed in the automatic sensing framework.

The introduction of temporal information for each pairwise interaction, however, also makes group interaction analysis more challenging. For a given group meeting, proximity links between people usually have different start/end times, and so it is unclear how to determine the start/end time of the group interaction, specially in the case of noisy data. To avoid the cost of automatic segmentation, one can choose a slice-based representation, in which the dynamical network is divided into slices of short duration. With an appropriate setting of the duration of a slice, relevant interaction patterns could potentially emerge from various snapshot of the dynamical network. Figure 2 shows two snapshots at two consecutive hours on a given day. As can be seen, the network topology changes quickly according to the real life event. And in both snapshots, one can observe interactions between workers, with different intensity.

3.3 Sensing quality with Bluetooth

As discussed in the introduction, using Bluetooth as proximity sensor has many advantages, but the Bluetooth data source is unfortunately quite noisy. In practice, often, 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 data in real conditions, and use the results of the quality analysis as the input to set the slice duration parameter.

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
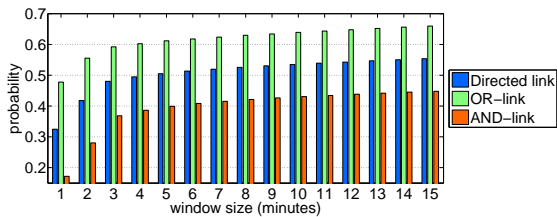
Fig. 3 Proximity detection rate of Bluetooth sensor for group meeting data for directed and undirected link cases and varying time slice duration.

the 12-month recording period. Based on this grouth truth data, we would like to estimate the rate at which the phone of each person successfully detects the other group members. 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:

- A directed link from user u to user v corresponds to the fact that u observed v during the time slice.
- An OR-link between u and v corresponds to the fact that (u observed v) OR (v observed u) during the time slice.
- An AND-link between u and v corresponds to the fact that (u observed v) AND (v observed u) during the time slice.

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 five minutes in order to obtain near optimal link detection rate with the Bluetooth sensor. Looking at the result for the directed link case, we found that the Bluetooth sensor has a proximity detection rate of 0.5 at 10-minute time slice. The rate can be improved by considering Bluetooth data from pairs of users in the case of OR-link, for which the proximity detection rates are roughly 25% better than the case of directed link. Finally, we also report the two-way-detection rates (AND-link case) which are obviously lower than the two other cases because of the strict condition of detection.

4 Overview of our method

Our approach to discover group interaction patterns is described in Figure 4. At the low level, we collected raw BT proximity links over months of real life to capture
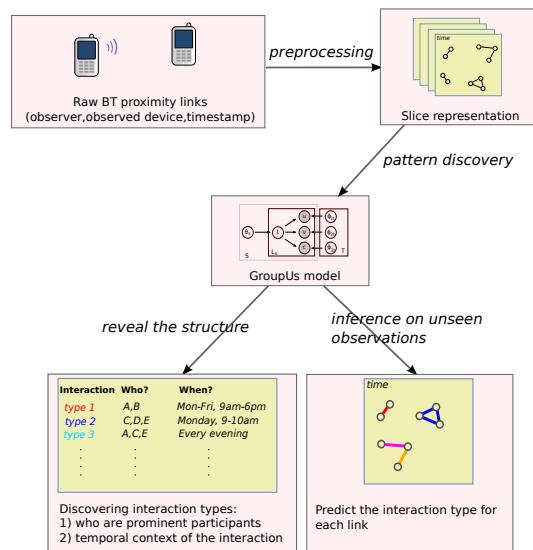

Fig. 4 Overview of our method.

actual events in the life of a community. The raw data was transformed into a time-slice based representation, where each slice is described by an interaction graph between users and its temporal context. The slice representation was then analyzed by the GroupUs model, which is designed to identify different interaction types like group meetings, lunches with family members, etc. in an unsupervised manner. The GroupUs model in turn gave the set of discovered interaction types and the assignments of interaction type for the set of links.

The GroupUs algorithm is presented in detail in the next section. Its application application on real-life BT data is then described in Sections 6 and 7.

5 GroupUs : A probabilistic model for sensing group interaction.

In this section, we present a probabilistic model for analyzing dyadic interaction data, which is usually represented by 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. Although the undirected OR-link has the best detection rate, it require a constraint that the set of observers and the set of observed devices are identical. In this study, we consider directed links to allow the fact that we could have two separate sets of observers and observed devices.

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 dynamical 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.

5.1 The probabilistic model

In many cases, the observed Bluetooth data is noisy. It may be due to technical problems of the sensor or due to 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. On the other hand, sometimes, a member of the group could be absent from the group meeting. We call this the "reality noise" of the group meeting.

In order to handle such stochasticity of the data, we use a probabilistic approach where observations are represented by random variables. A latent variable is introduced for capturing emergent patterns from the observations. This idea is inspired from topic models which was originally proposed for text [4,18] and had applications to other domains such as image retrieval and bioinformatics [20,41].

Our graphical model is illustrated in Figure 5, 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
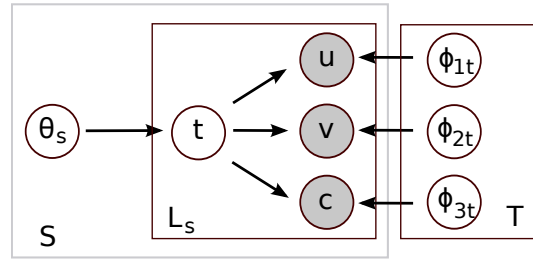

Fig. 5 Graphical model.

Table 1 Generative process.

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:

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

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. To simplify the presentation, the mathematical derivations have been omitted here but are available in the appendix.

5.2 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 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) [14]. In this work, we learn the model using collapsed Gibbs sampling [16], 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 the ith 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 :

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

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, $n_{st}^{\neg i} = \sum_{j \neq i} \mathbf{1}(s_j = s \text{ and } t_j = t)$. Given the interaction type assignments for all links, we can estimate the model parameters as follows:

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

Algorithm 1 GroupUs learning algorithm

1: **input:** interaction links $\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{s}$
2: **output:** model parameters, θ, ϕ, 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: $n_{st_i}\text{--}; m_{t_i u_i}\text{--}; p_{t_i v_i}\text{--}; q_{t_i c_i}\text{--};$
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: $n_{st_i}\text{++}; m_{t_i u_i}\text{++}; p_{t_i v_i}\text{++}; q_{t_i c_i}\text{++};$
10: 　　**end for**
11: **end while**
12: Compute θ, ϕ according to Eq. 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 [38,25] for which the complexity grows superlinearly (quadratically or sometimes 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.

5.3 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?. These questions are discussed in the following.

Inferring the participants of a given type of interaction. The learned parameter ϕ_{1t} tells us the probability of

---

Algorithm 2 Finding prominent users.

1: **Input:** $P(u|t)$
2: **Output:** most prominent users.
3: Sort users by $P(u|t)$
4: **for** $n = 1$ to #users **do**
5:     Compute Kullback Leibler divergence $KL(n)$ between:
      P_{proto}^n : the prototype distribution with n-top participants
      $P(u|z)$: the input distribution
6: **end for**
7: $n^* = \arg\max \; KL(n)$
8: Return the list of top n^* users.

---

observing user u given the type of interaction t, and thus we can answer the first question based on ϕ_{1t}. 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 that 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 $P(u|t)$ to determine the list of prominent users in the interaction 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:

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

Figure 6 shows an input distribution and three prototype distributions P_{proto}^n. Among these prototype distributions, P_{proto}^4 has lowest the Kullback Leibler divergence to the input distribution, and so the list of dominant users is the top 4 users.

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
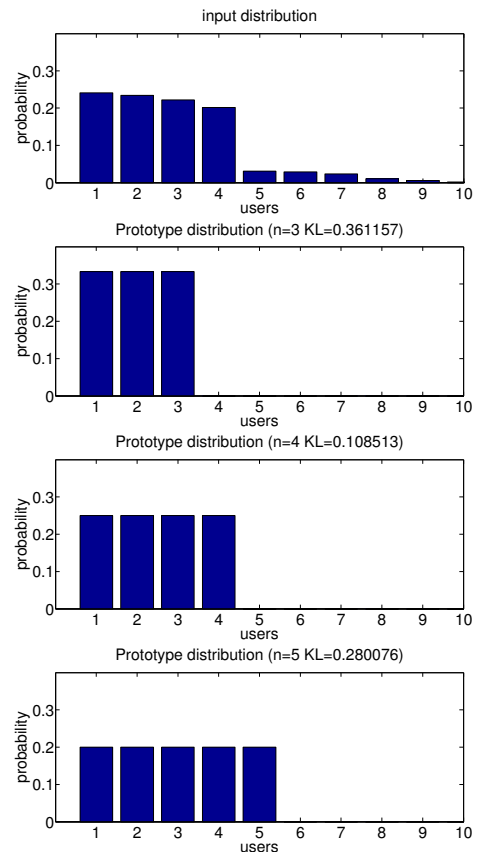

Fig. 6 The input distribution and three prototype distributions P_{proto}^3, P_{proto}^4 and P_{proto}^5. Optimal segmentation can be found by comparing the input distribution with prototype distributions. Kullback Leibler divergence is low for the prototype distribution that is close to the input distribution.

an interaction of type t occurs during working time can be computed as:

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

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 6.2, this information is helpful to visualize data or for further analysis. Clearly, if $P(H|t)$ is high then it is likely that the interaction type t corresponds to an office interaction.

6 Discovering interaction types between participants in the data collection campaign.

We first apply the GroupUs algorithm to the Bluetooth proximity data in order to discover emergent group interaction patterns. For our experiments, we set the slice

duration to 10 minutes which is a good tradeoff between link detection rate and temporal resolution. With this setting, we expected that the model can capture important events for context-aware applications. Note, however, that the time scale parameter is tunable, and it depends on the data and the goal of the analysis, for which we might be interested in extended time scales such as months or longer periods. Clearly, the choice of the number of interaction types, T, also influences the results. A small value of T leads to coarse interaction patterns and a large value of T produces fine-grained results. After manually studying the results with different settings of T (varying from 10 to 100), we chose $T = 40$ for reporting typical discovered interaction types, which produces a few interaction types per users (note that an interaction type involves generally many users). Finally, for other hyperparameter, we set $\alpha = 0.1, \beta = 0.1$ and $T_0 = 0.5$.

Starting from random initialization, our algorithm refines model parameters in each Gibbs sampling iteration. We observed that the convergence is reached after about 30 iterations. Using the classification method in Section 5.3, we found 15 office interaction types and 25 family interaction types. First we highlight some typical examples of discovered interaction types by visualizing the learned model parameters. We then study the evolution of interactions over time in real events.

6.1 Discovered interaction types

We start with some examples of discovered interaction types, visualized with the pairwise matrix of interaction $(\phi_{1t}{}^{\mathsf{T}}\phi_{2t})$ and the distribution of temporal context over the weekly calendar (ϕ_{3t}) in Figure 7, left and right columns, respectively. The first two interaction types Fig. 7(a-b) correspond to working place interactions, where these groupings (the first one involving users 1-3, and the second one involving users 7-11) clearly correspond to the working place ground truth (compare with Fig. 1 (bottom)). Note that these interactions spread over working times but have low probability at lunch time, which indicate that these co-workers 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. In the two next interaction types Fig. 1(c-d), more people are involved and the temporal context reveals that these are not daily interactions. Figure 7(c) corresponds to a weekly group meeting on Fridays followed by lunch that is correctly discovered by the algorithm. Note that this group is spread over 4 different offices, and some of its members also appear as the most prominent users of other discovered interaction types (e.g., Figure 7(b))
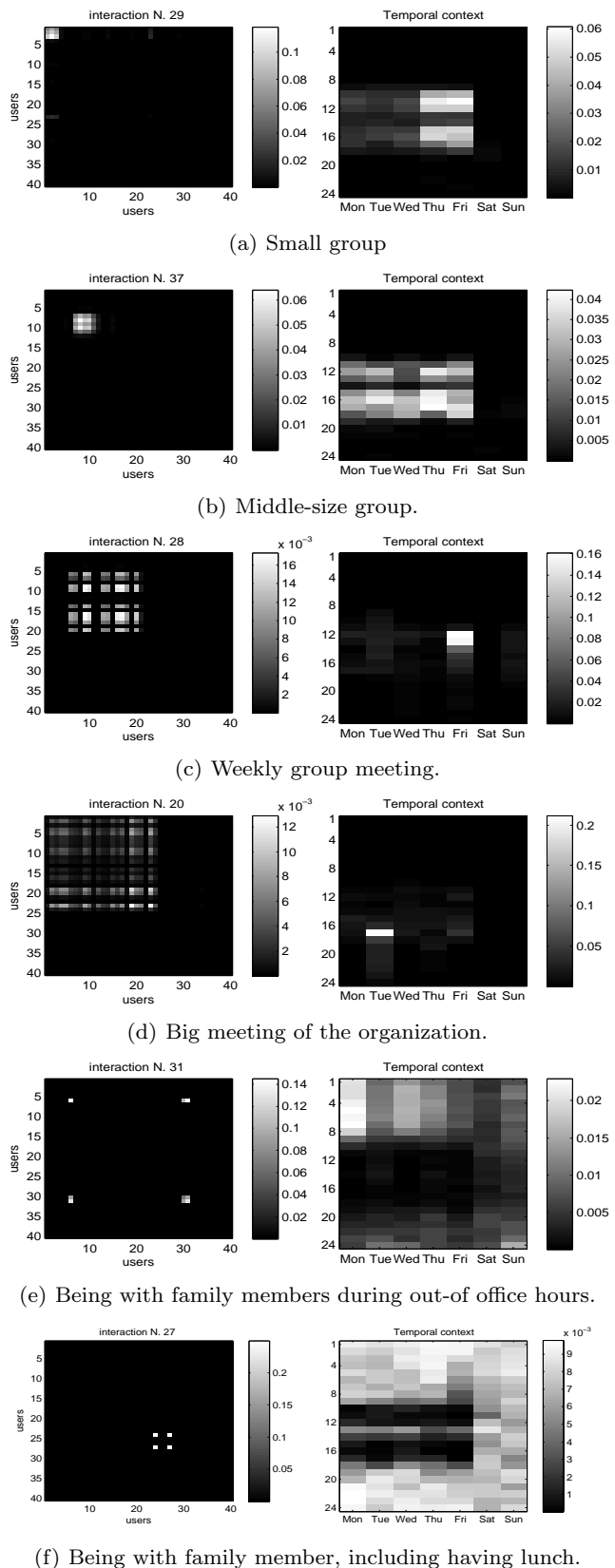

(a) Small group

(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.

Fig. 7 Typical discovered interactions visualized with pairwise interaction matrix $(\phi_{1t}{}^{\mathsf{T}}\phi_{2t})$ between users, shown in left column, and the distribution of temporal context (ϕ_{3t}), shown in right column.
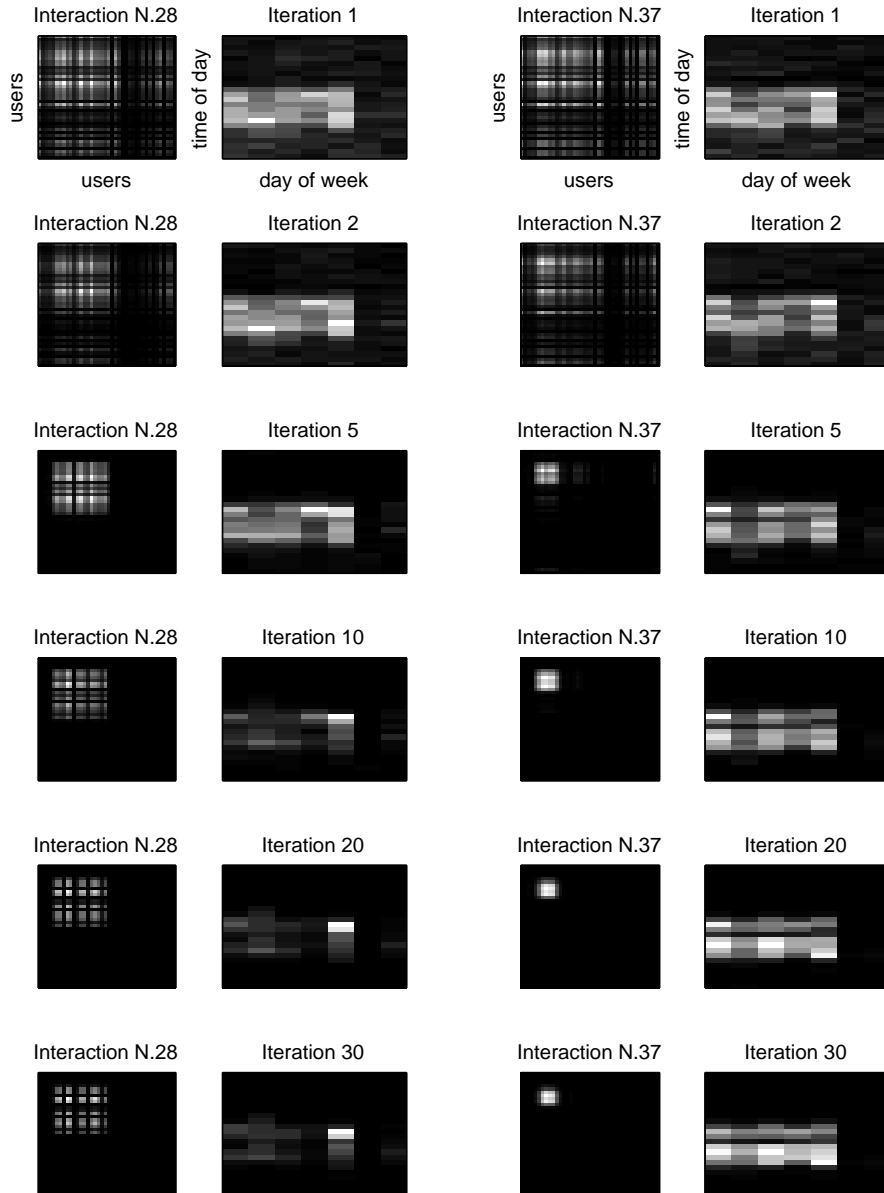
Fig. 8 The evolution of the model parameters associated with two interaction types (28 and 37) with respect to the number of iterations of Gibbs sampling.

which highlights the probabilistic advantage of GroupUs -people belong to multiple groups. The interaction type in Figure 7(d) reflects a weekly meeting of the whole 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 7(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.

Finally, to illustrate the convergence rate of Gibbs sampling, Fig. 8 shows the parameters of the model ($\phi_{1t}{}^\top \phi_{2t}$ and ϕ_{3t}) for 2 interaction types over various iterations (one iteration corresponds to one pass over the data). In the first iteration, the interaction type **t** is randomly assigned, therefore, the model parameters are similar for different interaction types. The parameters are then updated after each Gibbs sampling iter-
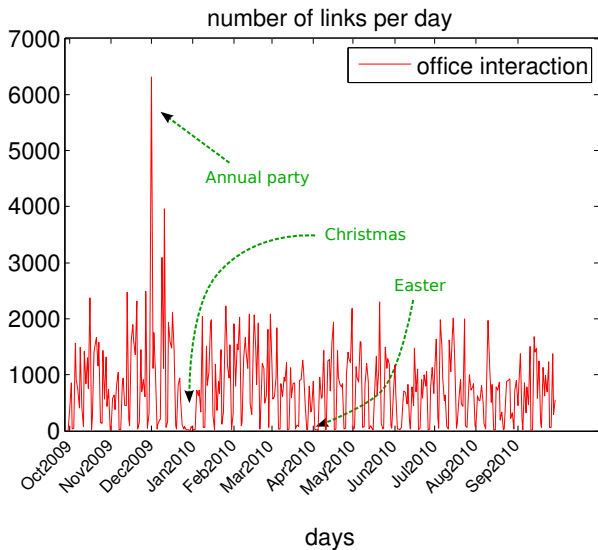
Fig. 9 Evolution of office interaction over time. Some emergent events such as holidays can be observed.

ation and converges to a stable state. As can be seen in Fig. 8, there was only a small change in model parameters between iteration 20 and 30, indicating that the algorithm converged after about 30 Gibbs sampling iterations.

6.2 Interaction over time

Although our method does not take into account absolute calendar temporal information (beyond the weekly schedule), we can nonetheless study the evolution of proximity interactions over time. Figure 9 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 5.3. As can be seen, we can observe some 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 in the 1-D sequence in Figure 9. The plot also shows big events of the organization. For instance, the highest peak occurring on December 1, 2009, was actually the annual party of the organization where people spend an afternoon and evening together.

We also compare the periodicity of office interactions with personal interactions. Figure 10 shows the autocorrelation of these two kinds of interactions for the population of 25 workers. As can be seen, the weekly periodicity of work interaction is very clear, while the weekly periodicity of family interaction is quite weak. Note that this analysis offers an automatic way to distinguish between periodic group interaction (such as
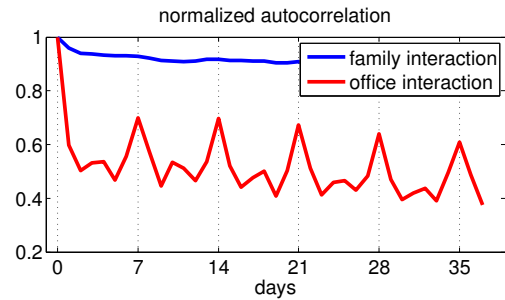

Fig. 10 Autocorrelation of family interaction (blue) and office interaction (red). The x-axis corresponds to days, y-axis corresponds to autocorrelation value

weekly meetings) and occasional group interactions. These results confirm early findings by Eagle and Pentland [11] but on a different organization and with a robust probabilistic approach that significantly reduces the presence of noise in Bluetooth data.

Interestingly, we also observed that the evolution of each interaction type over time also offers additional features besides the basic weekly temporal context. Figure 11 shows the evolution of some specific interaction types over time by plotting the number of links assigned to each interaction types for each day. For instance, the plot in Figure 11-(b) corresponds to some weekly events that occurred on Thursdays for only a few months in reality (mainly in spring and early summer). The time dimension of the interaction type number 20 (Figure 11-(c)) clearly indicates that these are truly unique events (e.g., party). The GroupUs model can infer not only the emergent interaction types of the network, but also when these interactions occurred (or not) in the past.

7 Including unknown Bluetooth devices in the data

Bluetooth sensors have the ability to detect other Bluetooth devices in proximity, as long as these devices are in discoverable mode. This is a usual situation in practice, where many unknown devices may be observable in public spaces, public transport, etc. In this section, we include these unknown devices in the analysis, thus considering an extended population of both volunteer data providers and others. While more data provides the opportunity to infer more accurately social context, one key challenge is to work with many unknown devices.

Device filtering. On the collected data, we found that each of our users observed a huge number of Bluetooth devices during the data collection campaign (in the order of thousands). These huge numbers come mainly from strangers' devices which were observed only once
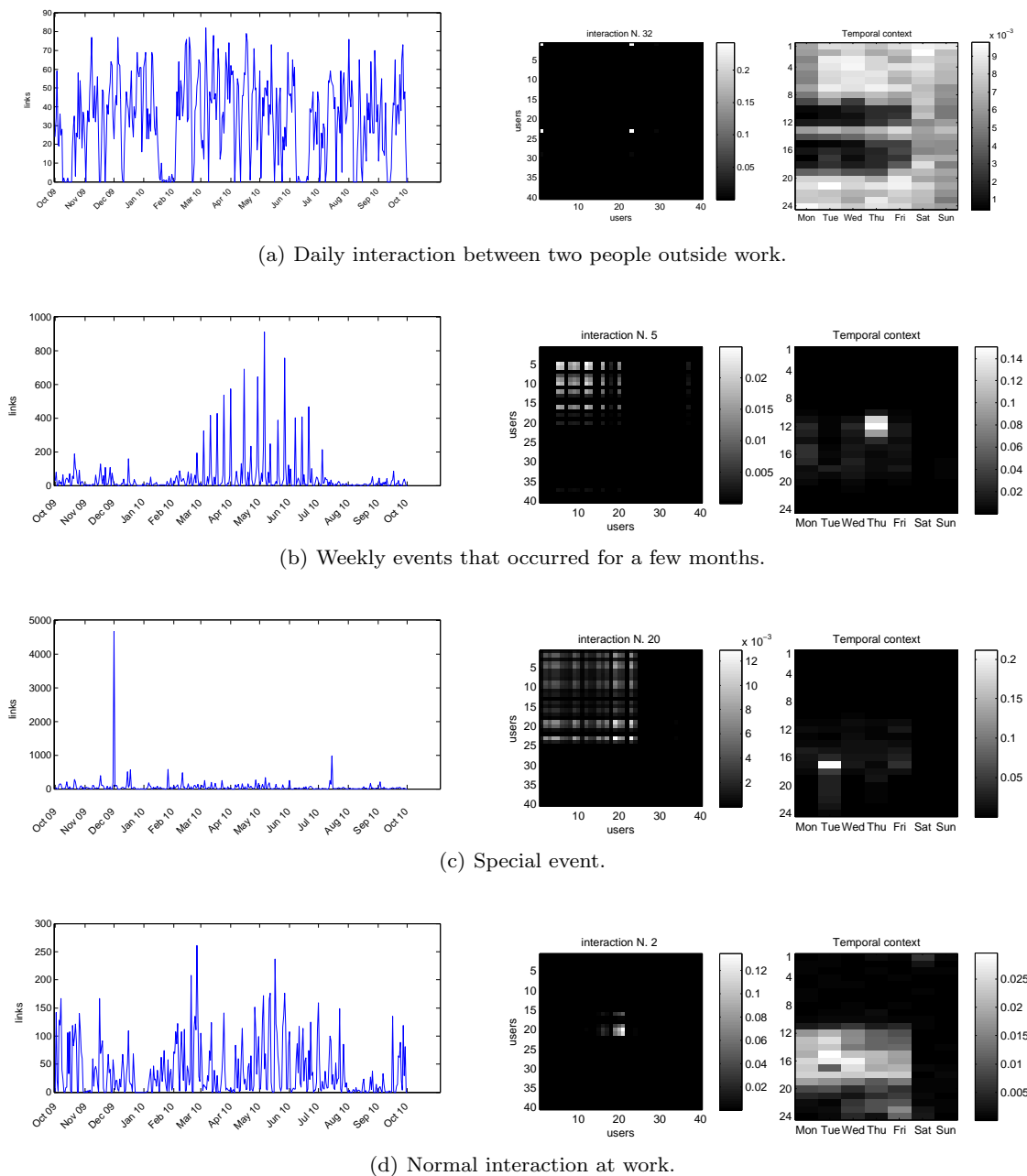
(a) Daily interaction between two people outside work.

(b) Weekly events that occurred for a few months.

(c) Special event.

(d) Normal interaction at work.

Fig. 11 Evolution of some interaction types over time. The left column shows the number of links observed per day for each interaction type. The center and right columns show the model parameters of the interaction type.

or a few times. As our goal is to discover regular patterns, we selected for this analysis only unknown devices which were observed frequently. For each device, we computed the weekly observing frequency as the fraction of the number of days that we observed the BT device and the total number of sensing weeks. By filtering out devices that were observed least than once a week, we get a set of 120 additional devices. Figure 12 shows the weekly observing frequency for the set of BT

devices in the extended data set. For each device, we also report the average observing time computed on the set of days that the device was observed. In order to determine the type of these unknown Bluetooth devices, we also utilized the location information derived from GPS data. Since the GPS and BT data collection were not synchronized, we aggregated GPS traces that were closest in time (within a 1-minute time window), to the time-instances when a given Bluetooth device was de-
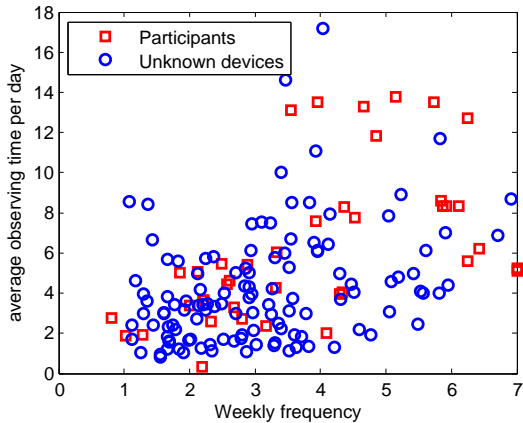
Fig. 12 Observing frequency of observed BT devices and participants' devices.

tected. This gave us a list of GPS location data points at which the BT device was likely to be detected. Interestingly, most of these frequently observed devices are found in multiple places, meaning that they are mobile devices (e.g., phone, laptop) while a few devices were found at only one place.

We used the same GroupUs setting as in the previous section for the extended data set with 160 observed devices (40 volunteer users and 120 extra unknown Bluetooth devices). It is not surprising that the algorithm found similar interaction types as the previous sections. For instance, the interactions in Figure 13-(a,b) are an extended version (with some extra unknown Bluetooth devices in the background) of some of the interaction types discovered previously. Interestingly, the model was able to discover some new types of interactions, which involve only one observer and his/her frequently observed Bluetooth devices. Figure 13-(c) corresponds to an interaction pattern at home (since it occurred every night), whose temporal context is very special compared to the interaction types we discovered from the original 40-person population. Finally, Figure 13-(d) shows an interaction type discovered for a user (number 4) whose office is not close to the offices of other users in the original population, but who consistently appears in proximity to several unknown devices (labeled 71, 118, 119) in office hours. Note that the two last interaction types cannot be discovered without extending the initial population with the list of nearby unknown Bluetooth devices. Therefore, the key role of including nearby Bluetooth devices for practical uses of GroupUs is to complete the list of possible interaction types discovered for each user.

8 Objective evaluation

In this section, we evaluate numerically our model by studying the predictive performance on unseen data and the retrieval performance. First, we consider the likelihood of the model on unseen data as a measurement, and compare GroupUs with a recently proposed model. Subsequently, we show how to use the ground truth group meeting schedule for understanding the performance of GroupUs on retrieval task.

8.1 Predictive performance

Prediction is a very important task in context-aware mobile applications [6,37,42]. Our main goal is to validate the learning capability of the proposed model by computing the likelihood on unseen data. For this reason, we do not consider a real-time prediction task neither compare GroupUs with predictive models such as ARIMA [27]. For the evaluation of predictive performance, the last two months of data for the 40-person population are used for testing, and we learn the model with different training sets, varying from 2 to 10 months of data from all users, starting from the most recent data and adding data backwards in time.

As a baseline, we adapted the Marginal Product Mixture Model (MPMM) which was proposed recently for analyzing phone call record data [9]. 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 links between pairs of nodes in a graph. 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.

Note that GroupUs model has two sets of parameters: $(\phi_{1t}, \phi_{3t}, \phi_{3t})$ for the discovered interaction types, and θ_s for each slice s in the training data. To compute the likelihood of unseen data, we first need to estimate the parameter θ_s for each unseen slice. This can be done using a process similar to Gibbs sampling in Algorithm 1, except that we do not need to resample interaction type t for links that were in the training data. The likelihood of test data $\bar{\mathbf{u}}, \bar{\mathbf{v}}, \bar{\mathbf{c}}, \bar{\mathbf{s}}$ given the GroupUs parameters ϕ, θ is computed as:

$$P(\bar{\mathbf{u}}, \bar{\mathbf{v}}, \bar{\mathbf{c}}, \bar{\mathbf{s}}|\phi, \theta) = \sum_{\bar{\mathbf{t}}} P(\bar{\mathbf{u}}, \bar{\mathbf{v}}, \bar{\mathbf{c}}, \bar{\mathbf{s}}, \bar{\mathbf{t}}|\phi, \theta) \qquad (7)$$

$$= \prod_{i=|\bar{\mathbf{t}}|} \sum_{t=1..T} \phi_{1t\bar{u}_i} \phi_{2t\bar{v}_i} \phi_{3t\bar{c}_i} \theta_{\bar{s}_i t} \qquad (8)$$

where $\bar{\mathbf{t}}$ denote the interaction type assignment for the set of test links. Since $\mathbf{t}$ is unknown, we take the sum over all possible assignments $\bar{\mathbf{t}}$. This sum can be factor-

(a) Interaction at work

(b) Interaction at home

(c) A special interaction type that is likely to occur at home

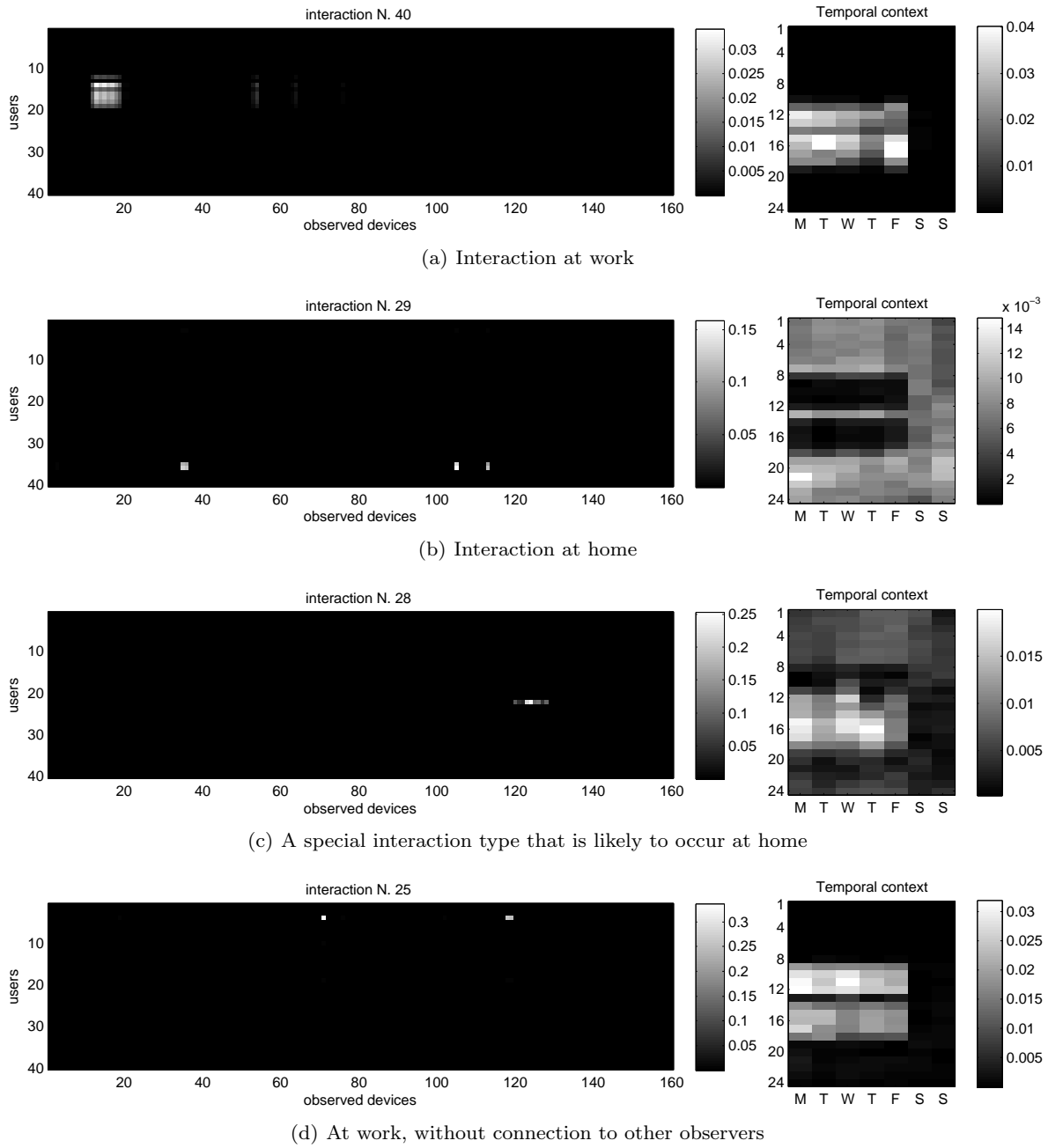(d) At work, without connection to other observers

Fig. 13 Examples of discovered interaction types when unknown Bluetooth devices are included in the analysis.

ized, so that it can be computed efficiently as given in Eq. (8).

Figure 14 plots the test log-likelihood for different training sizes. As can be seen, GroupUs clearly outperforms the MPMM model in term of predictive performance due the more accurate modeling assumptions of our model. In general, the more training data, the more accurate the inferences of GroupUs can be, but note that using "too old" data (see e.g., the case of

10 months of training data in Fig. 14) might not help improving predictive performance.

8.2 Retrieval performance on real events

In this section, we use the limited grouth truth of group meeting data (which was used in Section 3.3 for evaluating the robustness of Bluetooth data) for validating how well GroupUs recognize these meetings. The experiment is conducted as follows. First, we select an
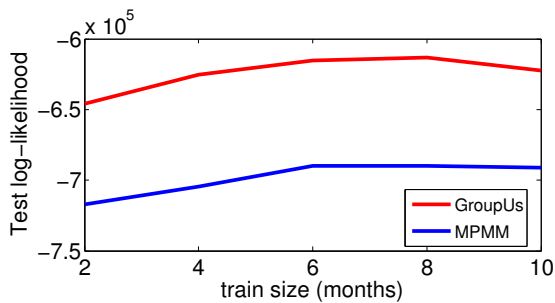
Fig. 14 Log-likelihood on the test data for varying size of training data.

interaction type which most resembles the real group meeting based on the set of participants. Then we retrieve the set of "significant occurences" of the selected interaction type. Finally, we evaluate the retrieval performance by comparing the list of dates of actual (reported) meetings and the list of dates retrieved. First, we describe how to retrieve significant occurences of a given interaction type. Then we analysis the results on retrieval performances of GroupUs.

Retrieving significant occurences of an interaction type. Thus far, we have seen how GroupUs model assigns an interaction type to each link. We have found that if many links in a group of users are assigned the same interaction type, this interaction type is more likely to have occurred in real life. If there are n users who are in proximity, in a given time slice, then there should be $n \times (n-1)$ links connecting them. However, since Bluetooth scans do not discover all devices in proximity (in Section 3.3, we found that in our dataset, the detection rate was found to be about 0.5), some of the links might be absent from the group. Further, even though the current slice belongs to a certain interaction type pertaining to a group event, some of the participants might be missing from the group event, causing these links to go missing. Hence, in order to determine the presence of interaction type t for each slice of time s, we define the link rate τ_{link} as the ratio of the number of observed links to the expected number of links for that interaction type, and the user rate τ_{user} as the ratio of the number of people actually detected by the Bluetooth scan to the number of participants, who are expected to be a part of the interaction as:

$$\tau_{link} = \frac{l}{m \times (m-1)}$$
$$\tau_{user} = \frac{m}{n}$$

where l is the number of links assigned to the considered interaction type, m is the number of people involved in the set of l links, and n is the number of prominent participants of the interaction type t defined in Section 5.3. For example, considering the interaction type #5

in Fig. 7(a), there are $n = 3$ prominent participants. Assume that in a slice s, there were 5 directed links of type #5 between these 3 people, then the link rate of interaction type #5 in slice s is $\tau_{link} = 5/6 = 0.83$ and the participant rate is $\tau_{user} = 3/3 = 1$.

Finally, we determine whether the group event of interaction type t actually happened in slice s using the two thresholds λ_{link} and λ_{user}, and the following conditions:

$$\tau_{link} \geq \lambda_{link}$$
$$\tau_{user} \geq \lambda_{user}$$

Among the 40 discovered interaction types, for the evaluation we selected the one who most resembles the real group meeting in the organization for which we known the exact calendar (which was used in Section 3.3 for evaluating the robustness of Bluetooth data), based on the set of participants. Then, we use the above method to retrieve the list of prominent slices where the event is likely to have occurred, and then map the list of slices to the corresponding list of calendar days to compare it with the actual list of days when the meetings occurred.

Evaluation. The retrieval performance is evaluated in terms of precision, recall, and F1 measure for varying threshold values. Let $\mathbf{d}$ be the list of calendar dates retrieved with the selected interaction type and $\mathbf{d}^*$ be the actual list of dates when the meetings occurred (there were 18 meeting dates). The three measures were computed as follows:

$$precision = \frac{|\mathbf{d} \cap \mathbf{d}^*|}{|\mathbf{d}|}$$
$$recall = \frac{|\mathbf{d} \cap \mathbf{d}^*|}{|\mathbf{d}^*|}$$
$$F_1 = 2 \cdot \frac{recall \cdot precision}{recall + precision}$$

Intuitively, precision is the fraction of retrieved dates that belong to the list of 18 meetings, and recall is the fraction of actual meeting dates that are retrieved. Note that the model might reach a high precision by retrieving only a few most likely dates, but some meeting dates will probably missed. Also, it is trivial (but useless) to achieve recall of 100% by returning all calendar dates. Hence, F1-score is usually used for evaluating retrieval task in practice, which can be viewed as a tradeoff between precision and recall. Figure 15 shows the results for various threshold combinations. It is interesting to note that the best threshold λ_{link} (in terms of F1 score) coincides with the Bluetooth detection rate (0.5). Given that the number of participant in real data varied widely from 5 to 10, the best value for λ_{user} is also quite low (0.6).

Using the best threshold combination ($\lambda_{link} = 0.5$, $\lambda_{user} = 0.6$), the method retrieved a list of 22 calendar dates, of which 12 belonged to the list of 18 ground

truth meetings. Looking at the 6 meetings that were not retrieved, we found that the τ_{user} for the corresponding dates are actually low, suggesting that some people did not carry their phone to the meeting or they were absent from the meetings in reality. Despite the presence of these reality noises, the GroupUs model could learn and retrieve the set of 12 other meeting dates correctly the selected interaction type alone. Note that besides the regular group meetings, there were also some occasional meetings or activities that were not reported in the list of 18 meetings. This would explain also why GroupUs retrieved 10 additional dates on which there were group interactions similar to the group meetings (with $\tau_{link} \geq 0.5$ and $\tau_{user} \geq 0.6$). Note, however, that we did not have ground truth data for these events and it is not possible to verify them automatically.

9 Conclusion

In this paper, we have presented and analyzed a probabilistic model, called GroupUs, designed to discover interaction types from large-scale dyadic data such as proximity, phone call network, or email network. Using a Bayesian framework, GroupUs infers the latent meaning of each interaction based on the set of observed interactions over slices of time. Importantly, the model exhibits some desirable properties such as scalability to the volume of data, and the ability to deal with noisy data. Our analysis was conducted on Bluetooth proximity data involving 40 users over 12 months of real-life, and we also show how the inclusion of observed but unknown Bluetooth devices can enrich the automatic inference of social context. We objectively evaluated our method by studying predictive performance, showing a significant advantage over a recently proposed model. Using only Bluetooth data, we showed that GroupUs can infer relevant interactions such as office interactions and family interactions in an unsupervised manner. One could incorporate other types of data (e.g., GPS or other location data) into GroupUs in order to enrich the description of the context of specific interactions, therefore providing more details about the discovered interaction types. We will pursue this as part of future work. While our model is generic, it has some hyperparameters (i.e. the number of interaction types T and the slice duration) which depend on the application. Finding these hyperparameters automatically from the data (by for example, using Chinese restaurant process) is also an interesting direction to explore.
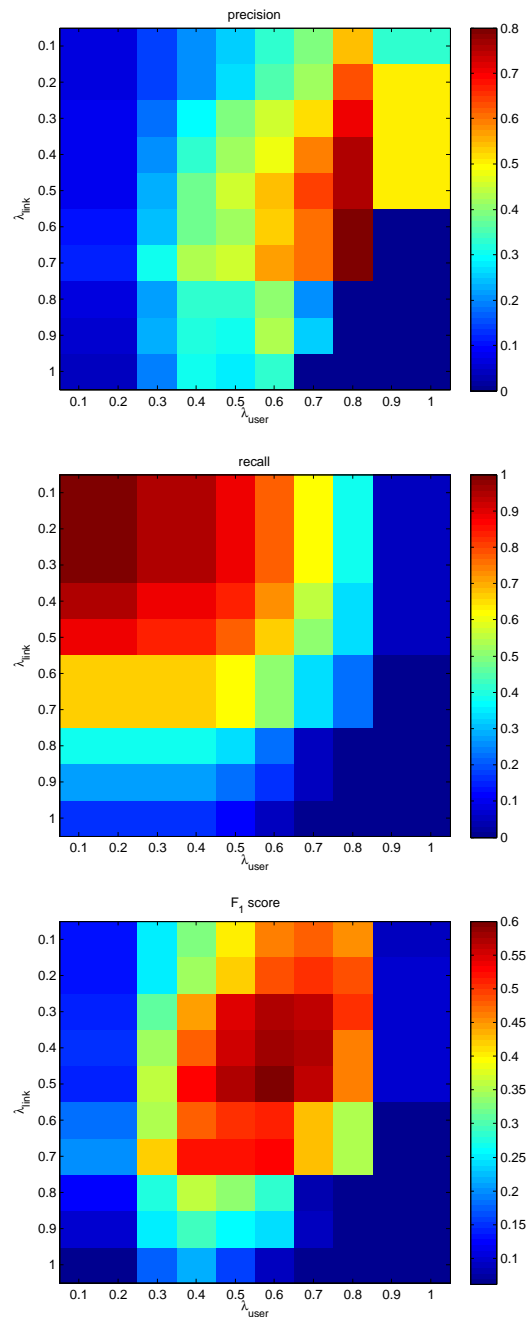
Fig. 15 Retrieving the set of group meetings with various thresholds.

References

1. Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. J. Mach. Learn. Res. **9**, 1981–2014 (2008)
2. Ashbrook, D., Starner, T.: Using gps to learn significant locations and predict movement across multiple users. Personal Ubiquitous Computing **7**, 275–286 (2003)
3. Banerjee, N., Agarwal, S., Bahl, V., Chandra, R., Wolman, A., Corner, M.D.: Virtual Compass: relative positioning to sense mobile social interactions. In: Proc.

Pervasive Computing, pp. 1–21 (2010)

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

5. Clauset, A., Eagle, N.: Persistence and periodicity in a dynamic proximity network. In: DIMACS Workshop on Computational Methods for Dynamic Interaction Networks (2007)

6. Dey, A.K.: Understanding and using context. Personal Ubiquitous Computing **5**, 4–7 (2001)

7. Do, T.M.T., Gatica-Perez, D.: Contextual grouping: discovering real-life interaction types from longitudinal bluetooth data. In: 12th International Conference on Mobile Data Management (2011)

8. Do, T.M.T., Gatica-Perez, D.: Groupus: Smartphone proximity data and human interaction type mining. In: 15th annual International Symposium on Wearable Computers (2011)

9. DuBois, C., Smyth, P.: Modeling relational events via latent classes. In: Proc. KDD, pp. 803–812 (2010)

10. Eagle, N., Pentland, A.S., Lazer, D.: Inferring social network structure using mobile phone data. PNAS **106(36)**, 15,274–15,278 (2009)

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

12. Farrahi, K., Gatica-Perez, D.: What did you do today?: discovering daily routines from large-scale mobile data. In: ACM Multimedia, pp. 849–852 (2008)

13. Fu, W., Song, L., Xing, E.P.: Dynamic mixed membership blockmodel for evolving networks. In: Proc. ICML, pp. 329–336 (2009)

14. Gilks, W.R.: Markov Chain Monte Carlo In Practice. Chapman and Hall/CRC (1999)

15. Gips, J., Pentland, A.: Mapping human networks. In: Proc. Pervasive Computing and Communications, pp. 159–168. IEEE Computer Society (2006)

16. Griffiths, T.L., Steyvers, M.: Finding scientific topics. PNAS **101**(Suppl. 1), 5228–5235 (2004)

17. Hightower, J., Consolvo, S., Lamarca, A., Smith, I., Hughes, J.: Learning and recognizing the places we go. In: Proc. UbiComp, pp. 159–176 (2005)

18. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 50–57. ACM, New York, NY, USA (1999). DOI http://doi.acm.org/10.1145/312624.312649

19. Huynh, T., Fritz, M., Schiele, B.: Discovery of activity patterns using topic models. In: Proc. Ubiquitous computing, pp. 10–19. ACM (2008)

20. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: Proc. SIGIR, pp. 119–126. ACM, New York (2003). DOI http://doi.acm.org/10.1145/860435.860459

21. K., N., B., S.T.A.: Estimation and prediction for stochastic blockstructures. JASA **96**, 1077–1087 (2001)

22. Kiukkonen, N., Blom, J., Dousse, O., Gatica-Perez, D., Laurila, J.: Towards rich mobile phone datasets: Lausanne data collection campaign. In: Proc. ICPS. Berlin (2010)

23. Krumm, J., Horvitz, E.: Predestination: Inferring destinations from partial trajectories. In: Proc. Ubiquitous Computing, pp. 243–260 (2006)

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

25. Mardenfeld, S., Boston, D., Juan Pan, S., Jones, Q., Iamnitchi, A., Cristian, B.: Gdc: Group discovery using co-location traces. In: SCA (2010)

26. McGovern, A., Friedland, L., Hay, M., Gallagher, B., Fast, A., Neville, J., Jensen, D.: Exploiting relational structure to understand publication patterns in high-energy physics. SIGKDD Explor. Newsl. **5**(2), 165–172 (2003)

27. Mills, T.C.: Time series techniques for economists. Cambridge University Press, Cambridge (1990)

28. Minkov, E., Cohen, W.W.: An email and meeting assistant using graph walks. In: CEAS (2006)

29. Montoliu, R., Gatica-Perez, D.: Discovering human places of interest from multimodal mobile phone data. In: Proc. International Conference on on Mobile and Ubiquitous Multimedia (2010)

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

31. O'neill, E., Kostakos, V., Kindberg, T., Schiek, A., Penn, A., Fraser, D., Jones, T.: Instrumenting the city: Developing methods for observing and understanding the digital cityscape. In: Proc. UbiComp, pp. 315–332 (2006)

32. Patel, S.N., Kientz, J.A., Hayes, G.R., Bhat, S., Abowd, G.D.: Farther than you may think: An empirical investigation of the proximity of users to their mobile phones. In: Proc. Ubiquitous Computing, P. Dourish, pp. 123–140. Springer (2006)

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

34. Sampson, F.S.: A novitiate in a period of change: An experimental and case study of social relationships. Ph.D. thesis, Cornell University (1968)

35. Scott, J.P.: Social Network Analysis. London: SAGE (1991)

36. Terry, M., Mynatt, E.D., Ryall, K., Leigh, D.: Social net: using patterns of physical proximity over time to infer shared interests. In: Proc. CHI, pp. 816–817 (2002). DOI http://doi.acm.org/10.1145/506443.506612

37. Vetek, A., Flanagan, J.A., Colley, A., Keränen, T.: Smartactions: Context-aware mobile phone shortcuts. In: INTERACT (1), pp. 796–799 (2009)

38. Wang, X., Mohanty, N., Mccallum, A.: Group and topic discovery from relations and their attributes. In: Proc. NIPS, pp. 1449–1456 (2006)

39. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press (1994)

40. Wyatt, D., Choudhury, T., Kautz, H.: Capturing spontaneous conversation and social dynamics: A priv. sensi. data collec. effort. In: Proc. ICASSP (2007)

41. Zheng, B., Jr, D.C.M., Lu, X.: Identifying biological concepts from a protein-related corpus with a probabilistic topic model. BMC Bioinformatics **7**, 58 (2006)

42. Ziebart, B.D., Maas, A.L., Dey, A.K., Bagnell, J.A.: Navigate like a cabbie: probabilistic reasoning from observed context-aware behavior. In: Proc. UbiComp '08, pp. 322–331 (2008)

Appendix : mathematical derivation of GroupUs learning algorithm.

Begin with the joint distribution:

$$
\begin{aligned}
&P(\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{s}, \mathbf{t}; \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
&= \int_\theta P(\mathbf{t}|\theta) P(\theta; \boldsymbol{\alpha}) \partial\theta \\
&\quad \times \int_\phi P(\mathbf{u}|\mathbf{t}, \phi_1) P(\mathbf{v}|\mathbf{t}, \phi_2) P(\mathbf{c}|\mathbf{t}, \phi_3) P(\phi; \boldsymbol{\beta}) \partial\phi \\
&= \int_\theta \left(\prod_s \prod_t \theta_{st}^{n_{st}} \prod_s \frac{\prod_t \theta_{st}^{\alpha-1}}{B(\boldsymbol{\alpha})} \right) \partial\theta \\
&\quad \times \int_{\phi_1} \left(\prod_t \prod_u \phi_{1tu}^{m_{tu}} \prod_t \frac{\prod_u \phi_{1tu}^{\beta-1}}{B(\boldsymbol{\beta})} \right) \partial\phi_1 \\
&\quad \times \int_{\phi_2} \left(\prod_t \prod_v \phi_{2tv}^{m_{tv}} \prod_t \frac{\prod_v \phi_{2tv}^{\beta-1}}{B(\boldsymbol{\beta})} \right) \partial\phi_2 \\
&\quad \times \int_{\phi_3} \left(\prod_t \prod_c \phi_{3tc}^{m_{tc}} \prod_t \frac{\prod_c \phi_{3tc}^{\beta-1}}{B(\boldsymbol{\beta})} \right) \partial\phi_3 \\
&= \prod_s \int_{\theta_s} \frac{\prod_t \theta_{st}^{\alpha+n_{st}-1}}{B(\boldsymbol{\alpha})} \partial\theta_s \times \prod_t \int_{\phi_{1t}} \frac{\prod_u \phi_{1tu}^{\beta+m_{tu}-1}}{B(\boldsymbol{\beta})} \partial\phi_{1t} \\
&\quad \times \prod_t \int_{\phi_{2t}} \frac{\prod_v \phi_{2tv}^{\beta+p_{tv}-1}}{B(\boldsymbol{\beta})} \partial\phi_{2t} \times \prod_t \int_{\phi_{3t}} \frac{\prod_c (\phi_{3tc})^{\beta+q_{tc}-1}}{B(\boldsymbol{\beta})} \partial\phi_{3t}
\end{aligned}
$$

where the term inside integration has similar form as Dirichlet distribution. Note that $\int_x \frac{\prod_i x_i^{\alpha-1}}{B(\boldsymbol{\alpha})} \partial x = 1$, we have:

$$
\begin{aligned}
&P(\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{s}, \mathbf{t}; \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
&= \prod_s \frac{B(\boldsymbol{\alpha}+\mathbf{n}_s)}{B(\boldsymbol{\alpha})} \times \prod_t \frac{B(\boldsymbol{\beta}+\mathbf{m}_t)}{B(\boldsymbol{\beta})} \times \prod_t \frac{B(\boldsymbol{\beta}+\mathbf{p}_t)}{B(\boldsymbol{\beta})} \times \prod_t \frac{B(\boldsymbol{\beta}+\mathbf{q}_t)}{B(\boldsymbol{\beta})} \\
&= \prod_s \frac{B(\boldsymbol{\alpha}+\mathbf{n}_s)}{B(\boldsymbol{\alpha})} \times \prod_t \frac{B(\boldsymbol{\beta}+\mathbf{m}_t)}{B(\boldsymbol{\beta})} \frac{B(\boldsymbol{\beta}+\mathbf{p}_t)}{B(\boldsymbol{\beta})} \frac{B(\boldsymbol{\beta}+\mathbf{q}_t)}{B(\boldsymbol{\beta})}
\end{aligned}
$$

The conditional probability can be computed efficiently based on the fact that they are proportional to the joint probability:

$$
\begin{aligned}
&P(t_i|\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{t}_{\neg i}; \alpha, \beta) \\
&= \frac{P(\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{t}; \alpha, \beta)}{P(\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{t}_{\neg i}; \alpha, \beta)} \\
&\propto \frac{P(\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{t}; \alpha, \beta)}{P(\mathbf{u}_{\neg i}, \mathbf{v}_{\neg i}, \mathbf{c}_{\neg i}, \mathbf{t}_{\neg i}; \alpha, \beta)} \\
&\propto \frac{\prod_s \frac{B(\boldsymbol{\alpha}+\mathbf{n}_s)}{B(\boldsymbol{\alpha})} \times \prod_t \frac{B(\boldsymbol{\beta}+\mathbf{m}_t)}{B(\boldsymbol{\beta})} \frac{B(\boldsymbol{\beta}+\mathbf{p}_t)}{B(\boldsymbol{\beta})} \frac{B(\boldsymbol{\beta}+\mathbf{q}_t)}{B(\boldsymbol{\beta})}}{\prod_s \frac{B(\boldsymbol{\alpha}+\mathbf{n}_s^{\neg i})}{B(\boldsymbol{\alpha})} \times \prod_t \frac{B(\boldsymbol{\beta}+\mathbf{m}_t^{\neg i})}{B(\boldsymbol{\beta})} \frac{B(\boldsymbol{\beta}+\mathbf{p}_t^{\neg i})}{B(\boldsymbol{\beta})} \frac{B(\boldsymbol{\beta}+\mathbf{q}_t^{\neg i})}{B(\boldsymbol{\beta})}} \\
&\propto \frac{B(\boldsymbol{\alpha}+\mathbf{n}_{s_i})}{B(\boldsymbol{\alpha}+\mathbf{n}_{s_i}^{\neg i})} \times \frac{B(\boldsymbol{\beta}+\mathbf{m}_{t_i})}{B(\boldsymbol{\beta}+\mathbf{m}_{t_i}^{\neg i})} \times \frac{B(\boldsymbol{\beta}+\mathbf{p}_{t_i})}{B(\boldsymbol{\beta}+\mathbf{p}_{t_i}^{\neg i})} \times \frac{B(\boldsymbol{\beta}+\mathbf{q}_{t_i})}{B(\boldsymbol{\beta}+\mathbf{q}_{t_i}^{\neg i})} \\
&\propto \frac{\alpha+n_{s_i t_i}^{\neg i}}{\sum_t (\alpha+n_{s_i t}^{\neg i})} \frac{\beta+m_{t_i u_i}^{\neg i}}{\sum_u (\beta+m_{t_i u}^{\neg i})} \frac{\beta+p_{t_i v_i}^{\neg i}}{\sum_v (\beta+p_{t_i v}^{\neg i})} \frac{\beta+q_{t_i c_i}^{\neg i}}{\sum_c (\beta+q_{t_i c}^{\neg i})}
\end{aligned}
$$

Since the denominator $\sum_t (\alpha + n_{s_i t}^{\neg i})$ is invariant for any value of t_i, we obtain the final sampling equation :

$$
\begin{aligned}
P(t_i|\mathbf{u}, \mathbf{v}, \mathbf{c}, \mathbf{t}_{\neg i}; \alpha, \beta) &\propto (\alpha + n_{s_i t_i}^{\neg i}) \frac{\beta+m_{t_i u_i}^{\neg i}}{\sum_u (\beta+m_{t_i u}^{\neg i})} \\
&\times \frac{\beta+p_{t_i v_i}^{\neg i}}{\sum_v (\beta+p_{t_i v}^{\neg i})} \frac{\beta+q_{t_i c_i}^{\neg i}}{\sum_c (\beta+q_{t_i c}^{\neg i})}
\end{aligned}
$$