Online Community Detection in Social Sensing

Guo-Jun Qi
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL
qi4@illinois.edu

Charu C. Aggarwal
IBM T.J. Watson Research Center
Yorktown Heights, NY
charu@us.ibm.com

Thomas S. Huang
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL
huang@ifp.uiuc.edu

ABSTRACT

The proliferation of location and GPS data streams which are collected in a wide variety of participatory sensing applications has created numerous possibilities for analysis of the underlying patterns of activity. Typically, the spatiotemporal patterns arising from such activity can be analyzed in order to determine the latent community structure in the underlying data. In this paper, we will examine the problem of online community detection from the location data collected from such social sensing applications in real time. Such data brings numerous challenges associated with it, in that they can be of a relatively large scale, and can be extremely noisy from the perspective of both data representation and analysis. Furthermore, the community structure in the underlying data cannot be directly inferred from the shape of the underlying trajectories, since a considerable amount of variation may exist in terms of trajectories of individuals belonging to the same community. In this paper, we will design *online* algorithms for community detection in social sensing applications. Our algorithm uses a robust and efficiently updateable model with the use of Gibbs sampling, and we will show its effectiveness and efficiency for social sensing applications.

Categories and Subject Descriptors

H.2.8 [**Databases Management**]: Database Applications

Keywords

Social Sensing

1. INTRODUCTION

The proliferation of mobile phones and a wide variety of other hardware with embedded and wearable sensors have resulted in a tremendous amount of GPS and trajectory data in a wide variety of applications. Such data can often be mined in order to determine the useful communities from the underlying data. A number of recent hardware advancements have lead to the increased importance of social sensing applications:

- The increasing accessibility of wearable sensors with GPS-enabled devices has made it much easier to track and collect individual locations in mobile sensing applications.
- The rapid popularization of smartphone technology with GPS abilities has increased our ability to collect spatio-temporal data in the context of a wide variety of applications.

Such spatiotemporal data can often be very useful for determining communities with the use of spatial and temporal co-location information between different entities. Nevertheless, the process of determining such communities can be extremely noisy and challenging because of a wide variety of reasons:

- At any given time, a pair of entities could be present at the same location purely by chance. It is only on the basis of repeated interactions between the pair of entities *over different times* that the entities may be said to belong to the same community.
- In many cases, the repeated co-location of communities may not occur over continuous periods in time. This is quite different from many trajectory clustering models, which tend to determine clustering behavior of trajectories on the basis of their shape, and the behavior over contiguous time stamps.
- GPS data is usually highly incomplete, as a result of which the locations of only a subset of the entities may be available at a given moment in time. Therefore, the community maintenance process must be highly cognizant of the errors and noise in the underlying data.
- Locations which have *higher selectivity* in terms of enabling the interaction of a small number of participants *at a given time* are often more indicative of community behavior. In locations containing an extremely large number of participants, the spatial proximity of a pair of participants may often not be quite as indicative of community behavior unless the two participants are extremely close to one another over a long period of time. In this context, the inherent errors of data collection may play a role, because an error of a few meters in location tracking in a spatial region of very high density may be significant, whereas it may not be quite

WSDM'13, February 4–8, 2013, Rome, Italy.
Copyright 2013 ACM 978-1-4503-1869-3/13/02 ...$15.00.

as significant in a spatial region of low density. Thus, the significance of the co-location of a pair of entities should be evaluated in the context of the density of the point of co-location.

- The underlying communities may often evolve over time, as new interactions are observed between different participants. Therefore, one can incorporate temporal information in the community detection process in order to perform the detection in real time. While recent interactions may be very relevant, the history of previous interactions should also play an important role in the community detection process.

The main challenge in the modeling process is to use all the different statistical indicators of community formation in a balanced way in order to continuously determine and track the underlying communities in real time. For example, while a considerable amount of work has been done recently in the area of trajectory clustering [4, 11, 13, 14, 17, 18], most of these methods are not designed for online community detection. Furthermore, most of these methods impose the constraint of either similar trajectory shape, or the occurrence of two objects at the same location over *consecutive* snapshots.

In this context, the problem of determining swarms in spatio-temporal data was proposed in [18], which determines groups of objects which occur together at possibly non-consecutive snapshots. However, this work requires significant global pre-processing of the trajectories, and has several constraints in the context of *real-time* community-detection:

- The work in [18] decouples the spatial and temporal aspects of the problem by performing a static (spatial) pre-clustering of the objects at *each time stamp* with the use of an off-the-shelf algorithm for further temporal analysis. In a sense, the spatial component of the problem is largely abstracted out by this work. This may not be practical in real applications, and the quality and speed of the approach is highly dependent on the quality of the (off-the-shelf) algorithm used for spatial pre-clustering. Thus, if hundreds of thousands of time-stamps are collected, such a spatial pre-clustering would need to be performed at *each time-stamp* to begin with. Such an approach cannot be used for real-time applications, especially in cases, where a large number of objects and time-stamps are involved. The de-coupling of the spatial and temporal aspects of an inherently spatio-temporal problem (and the subsequent de-emphasis on the spatial component by using an off-the-shelf spatial clustering algorithm) is also not qualitatively the most effective design. This can lead to a tremendous loss of information in the spatial pre-clustering phase.

- The purpose of clustering and community detection is to report a small number of groups of related entities, which further allows for easy assimilation and a concise representation. However, the work in [18] may determine hundreds of thousands of swarms by using a temporal frequent-pattern mining like approach on the pre-clustered data. The reported swarms may be highly overlapping, and in many cases, the total number of swarms reported may be greater than the number of objects being tracked. This does not serve the purposes of a problem such a community detection very well.

In this paper, we design a *real time* and *online* approach for determining the set of continuously evolving communities from the locations of a set of users. Different from traditional methods, the approach is real-time and continuous, and is not based on trajectory shape. It is also different from link prediction, which is typically based on very long term analysis, and typically cannot be performed in real time. Our approach uses relatively limited location information for the analysis process, and is also different from most of the (offline) pattern-based co-location clustering methods.

This paper is organized as follows. The remainder of this section discusses related work. In Section 2, we formulate and motivate the problem and introduce notations. In Section 3, we propose an *Infinite Community Dynamic Random Field (*IC-DRF*) model* to capture the community structure in the context of object locations and their dynamics. In Section 4, a Gibbs Sampling algorithm is presented for efficient incremental inference of community configurations of objects at each time stamp. In Section 5, we experimentally compare our approach with other state-of-the-art algorithms. Section 6 contains the conclusions and summary.

1.1 Related work

The problem of community detection is generally defined in the form of a clustering of the underling network [8, 19, 10, 16, 15]. A survey of a number of important algorithms for community detection is provided in [6]. Discussion of important statistical properties of web communities is discussed in [16]. Evolutionary characteristics of dynamic communities are studied in [2, 7, 8]. The problem of community detection has also been studied in the context of combining node and edge content in order to improve its effectiveness [21, 25, 27].

Social sensing [1] has recently found increasing interest because of the increasing importance of mobile phones and participatory sensing technology. Recent work has extensively studied common properties [9, 20, 22] of common spatio-temporal social networks, which reveal the link patterns between people in these networks. The work in [24] is more closely related to the problem of link prediction, though is not designed for real-time analysis. In this context, a considerable amount of work has been performed for trajectory clustering [4, 11, 13, 14, 17, 18]. Most of this work is either designed for discovering trajectories with *similar shapes* or in determining objects which occur together at consecutive time stamps. In practice, objects in the same community very rarely show similar trajectories, and the communities may be determined only on the basis of proximity to one another over significant periods of time. Furthermore, most of the existing work is applicable only for *offline* community detection, where most of the data is already available. For example, the swarms proposed in [18], determine *temporally co-occurring patterns of entities* from *spatially pre-clustered data*, though such an approach has limited applicability for real time community detection, because of the reasons discussed in the introduction section.

2. PROBLEM DEFINITION

We denote the set of actors tracked by the social sensing application by $\mathcal{O} = \{o_1, \cdots, o_n, \cdots\}$. Their geographical locations at each time stamp are denoted by $\mathcal{L}^{(t)} = \{l_1^{(t)}, \cdots, l_n^{(t)}, \cdots\}$. Since we propose a probabilistic model in this paper, we will readily handle the missing location data, which may occur as a result of failure of the device. Thus $\mathcal{L}^{(t)}$ does not need to contain the complete set of locations for all objects at each time. Our goal is to dynamically maintain a community partition of these moving objects into a set of the communities based on the trajectory information of their locations up to the current time stamp.

Problem 1. *Given a set of actors $\mathcal{O}$, along with a set of streaming locations $\mathcal{L}$, continuously maintain the set of communities from the actors $\mathcal{O}$ in real time.*

We denote the community assignment variable of each of the actors in the social sensing application by $\mathcal{Z}^{(t)} = \{z_1^{(t)}, \cdots, z_n^{(t)}, \cdots\}$, where $z_i^{(t)} \in \{1, 2, \cdots,\}$ is the community index for o_i at time t. In order to determine the community membership of these different participants, we use their location information, such that the objects, which are frequently close to one another are more likely to be in the same community. Intuitively, a good community structure will have low intra-community distance, and high inter-community distance across different communities. This can be quite challenging because of the temporal non-transitivity of distances; two objects which are consistently far away from one another, may frequently each be close to a third object, but never at the same time. This leads to challenges about how the community structure may be consistently defined in online fashion. In the next section, we will define a Dynamic Random Field (DRF) which uses this criterion to yield a probabilistic measurement over the possible configurations of community membership of all the participants.

Moreover, as a dynamic model, DRF should capture the evolution of objects from one community to another over time. We know that an object does not evolve between different communities with equal probability. For example, as long as the object location does not change too much, an object is more likely to stay in the current community unless its locations relative to the other objects in the same community changes dramatically. If the community membership of an object changes, we can expect that it is more likely to join in join in a closer community than the other distant ones. The DRF model will be able to seamlessly use the spatial and temporal components in online fashion in order to capture the dynamic changes of moving objects between communities. This increases the robustness and effectiveness of the community detection model.

The community structure of the moving participants evolves dynamically as their locations change over time. For example, some communities may merge with each other as their member objects encounter each other more frequently; on the other hand, a new community might appear as some objects depart from an existing community in order to show an independent pattern of behavior. Therefore, the number of communities can change dynamically as the geographical layout of objects change over time. This suggests that we cannot pre-decide the number of communities, especially when the natural granularity of the underlying communities is inherently dynamic. Therefore, the number of communities needs to be dynamically determined, on the basis of their spatial patterns and the corresponding evolution dynamics.

In the next section, we will present an Infinite Community - Dynamic Random Field (*IC-DRF*) which derives the evolving community structure from both the spatial trajectory patterns and the community evolution dynamics. The number of communities is automatically determined both from the current spatial dynamics and the temporal history of previous community structures.

3. INFINITE COMMUNITY - DYNAMIC RANDOM FIELDS

In this section, we will define an Infinite Community - Dynamic Random field (*IC-DRF*) model to determine the dynamic community structure in the social sensing problem. Specifically, the *IC-DRF* defines a sequence of probabilistic measures on the community configuration of objects over time based on their locations. The model derives the most probable configuration of community assignment to the moving objects by effectively comparing the intra-community distances between objects with inter-community distances over time.

This model has the following advantages:

1. As compared with many spatial community models, *IC-DRF* does not need to assume any specific prior knowledge about the spatial distribution of the members of each community. Actually, the spatial distribution of members can vary significantly in different communities. For example, people who travel together in dense regions (such as a mall) are usually expected to be much closer to one another as compared with those that travel in a mountain area. The *IC-DRF* model is flexible to varying community scales without the need for specializing to the spatial scale of a particular community.

2. The nonparametric Bayesian construction of this paper obviates the need to pre-decide the number of communities in the *IC-DRF* model. Instead, we assume a possibly infinite number of communities, so that the number of communities are dynamically determined in online fashion based on the spatio-temporal patterns of objects. In real-life applications, the number of communities are rarely constant, since they can evolve, merge and split because of variations in the underlying social interactions. A model which assumes countably infinite communities can simultaneously capture the dynamic changes in the number and structure of communities in a principled manner;

3. The *IC-DRF* model can maintain the community configuration of objects in an online manner. This is critical for social sensing applications, in which real-time analysis is usually an important goal.

To the best of our knowledge, *IC-DRF* is the first model to develop an online and dynamic approach for integrated spatio-temporal evolution and dynamic community analysis. Furthermore, this approach requires no prior knowledge. Straightforward applications of existing nonparametric Bayesian models assume prior knowledge about the spatial distribution of the objects. This is usually not flexible enough to capture the dynamic changes of object locations over time. The *IC-DRF* model defines a prior-independent energy function for minimization, which yields the most probable configuration of community membership of the underlying objects.

3.1 Complete Probabilistic Model

Next, we will introduce the *IC-DRF* model for community detection. Let the community assignment from time instants 1 through T be denoted by $\mathcal{Z}^{(1:T)} = \{\mathcal{Z}^{(1)}, \mathcal{Z}^{(2)}, \cdots, \mathcal{Z}^{(T)}\}$. We define the following probability distribution $P(\cdot)$ for community assignment model from time instants 1 to T with the use of the notations introduced in the previous section:

$$\begin{aligned}&P(\mathcal{Z}^{(1:T)}|\mathcal{L}^{(1:T)}, \boldsymbol{\gamma}, \boldsymbol{\pi})\\ &\propto \prod_{t=1}^{T} \exp(-E(\mathcal{Z}^{(t)}|\mathcal{L}^{(t)})) \prod_{t=2}^{T} P(\mathcal{Z}^{(t)}|\mathcal{Z}^{(t-1)}, \boldsymbol{\pi}) P(\mathcal{Z}^{(1)}|\boldsymbol{\gamma})\end{aligned} \tag{1}$$

The intuitive explanation for the different components of this model are as follows:

- The expression $E(\mathcal{Z}^{(t)}|\mathcal{L}^{(t)})$ (which is described in more detail in the next subsection) defines an energy function which measures the consistency of object locations for the community structure at each time t. In practice, we will use the exponentiated potential function $\exp(-E(\mathcal{Z}^{(1)}|\mathcal{L}^{(1)}))$ of the energy function for analysis.

- The notations $P(\mathcal{Z}^{(1)}|\boldsymbol{\gamma})$ and $P(\mathcal{Z}^{(t)}|\mathcal{Z}^{(t-1)}, \gamma, \mu, \alpha)$ (for $t > 1$) represent the probability measures for the initial community assignment $\mathcal{Z}^{(1)}$ and the dynamic community assignment $\mathcal{Z}^{(t)}$ at time $t > 1$, conditional on the assignment $\mathcal{Z}^{(t-1)}$ at time $t-1$, respectively. We use the naive Bayes assumption to factorize these probability functions over the different objects.

$$P(\mathcal{Z}^{(1)}|\boldsymbol{\gamma}) = \prod_{o_i \in \mathcal{O}} P(z_i^{(1)}|\boldsymbol{\gamma}) \tag{2}$$

Also, for each $t > 1$, we have:

$$P(\mathcal{Z}^{(t)}|\mathcal{Z}^{(t-1)}, \boldsymbol{\pi}) = \prod_{o_i \in \mathcal{O}} P(z_i^{(t)}|z_i^{(t-1)}, \boldsymbol{\pi}) \tag{3}$$

Here $\boldsymbol{\gamma}$ is the model parameter for the initial community assignment with $P(Z_i^{(1)} = k|\boldsymbol{\gamma}) = \gamma_k$, and $\boldsymbol{\pi} = \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots\}$ are the parameters for the community evolution probability, where $\boldsymbol{\pi}_k = [\pi_{k1}, \pi_{k2}, \cdots,]$ gives the probability that a member of community k evolves to another community l in the next time stamp, i.e., $P(z_i^{(t)} = l|z_i^{(t-1)} = k, \pi) = \pi_{kl}$.

In the following, we will impose the Hierarchical Dirichlet Process (HDP) prior on the model parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\pi}$, and give a joint probability of community configurations by marginalizing out $\boldsymbol{\gamma}$ and $\boldsymbol{\pi}$ over the HDP prior:

$$\begin{aligned}&P(\mathcal{Z}^{(1:T)}|\mathcal{L}^{(1:T)}) \propto \int_{\boldsymbol{\gamma},\boldsymbol{\pi}} \prod_{t=1}^{T} \exp(-E(\mathcal{Z}^{(t)}|\mathcal{L}^{(t)}))\\ &\times \prod_{t=2}^{T} P(\mathcal{Z}^{(t)}|\mathcal{Z}^{(t-1)}, \pi) P(\mathcal{Z}^{(1)}|\gamma) P(\boldsymbol{\gamma}, \boldsymbol{\pi}) d\boldsymbol{\gamma} d\boldsymbol{\pi}\end{aligned} \tag{4}$$

We can see that marginalization of the model parameters couples all the community configurations $\mathcal{Z}^{(1:T)}$ together, where the community configuration at each time stamp no longer depends only on its directly preceding community configuration as in Eq. (1). Therefore, the *IC-DRF* model captures the community evolution dynamics in the history in order to determine the most probable community assignments at each time by maximizing the above probability. This will become clear after we define the priors for $\boldsymbol{\lambda}$ and $\boldsymbol{\pi}$.

3.2 Location-Sensitive Energy Function

In this subsection, we will formally define the afore-mentioned location-sensitive community energy function to model the spatial structure of communities. Given the current community configuration $\mathcal{Z}^{(t)}$ of objects, we define the energy function as follows:

$$E(\mathcal{Z}^{(t)}|\mathcal{L}^{(t)}) = E_{\text{intra}}(\mathcal{Z}^{(t)}|\mathcal{L}^{(t)}) - E_{\text{inter}}(\mathcal{Z}^{(t)}|\mathcal{L}^{(t)}) \tag{5}$$

where

$$E_{\text{intra}}(\mathcal{Z}^{(t)}|\mathcal{L}^{(t)}) = \frac{\sum_{i \neq j} d_{ij}^{(t)} \delta \left[\left[z_i^{(t)} = z_j^{(t)} \right]\right]}{\sum_{i \neq j} \delta \left[\left[z_i^{(t)} = z_j^{(t)} \right]\right]} \tag{6}$$

and

$$E_{\text{inter}}(\mathcal{Z}^{(t)}|\mathcal{L}^{(t)}) = \frac{\sum_{i \neq j} d_{ij}^{(t)} \delta \left[\left[z_i^{(t)} \neq z_j^{(t)} \right]\right]}{\sum_{i \neq j} \delta \left[\left[z_i^{(t)} \neq z_j^{(t)} \right]\right]} \tag{7}$$

The last two equations represent the average of inter- and intra-community distances, and where $d_{ij}^{(t)}$ is the distance between object o_i and o_j computed from their locations l_i and l_j at time stamp t, $\delta\,[\![\cdot]\!]$ is the indicator function which outputs 1 if its condition holds, and outputs 0 otherwise. The first term on the right hand side of the equation computes the average distance between objects in the same community, and the second term computes the average distance between objects in the different community. Minimizing the above energy function will lead to a community configuration for objects that minimizes the average intra-community distance relative to the average inter-community distance over time.

It is worth noting that the number of communities do not need to be pre-decided a-priori. The afore-mentioned energy function handles different numbers of communities in a graceful and uniform way. As we will see in the next subsection, the number of communities can be automatically determined from the dynamic model for community evolution, as it relates to the spatial layout of the objects over time. The energy function also does not make any assumptions about the spatial scales of the underlying communities. In other words, no assumptions are made about how close two objects need to be to one another over time in order to belong to the same community. Thus the proposed model is adaptive and flexible in its ability to discover communities over a wide range of spatial scales.

Finally, we also notice that in some cases, some additional content (or meta-information) about moving community members is available, which can be valuable for the community detection process. For example, in a mobile application, the transportation modes of mobile devices may be available. It is clear that when two objects often have the same transportation mode, in addition to spatial locality, this is more likely to be indicative of similar community membership. Such hints can be seamlessly incorporated into our model by incorporating the following additional term into the energy function:

$$E_{\text{status}}(\mathcal{Z}^{(t)}|S^{(t)}) = \sum_{i \neq j} \delta \left[\left[z_i^{(t)} = z_j^{(t)} \right]\right] \delta \left[\left[s_i^{(t)} \neq s_j^{(t)} \right]\right] \tag{8}$$

Here, $s_i^{(t)}$ is the indicator from a set of L statuses $\{1, 2, \cdots, L\}$, and $\mathcal{S}^{(t)} = \{s_i^{(t)}|i = 1, 2, \cdots\}$ is the set of the statuses of all objects at time t. This energy function supports the membership of objects into the same community, when they frequently have similar status.

3.3 Infinite Community Evolution Model

To model the possibly infinite number of communities in *IC-DRF*, we adopt a Hierarchical Dirichlet Process (HDP) distribution to impose the prior on the model parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\pi}$ in Eq. (1). We also use the *stick-breaking construction* [23] in order to perform the modeling. While a detailed introduction of this construction is beyond the scope of this paper, we briefly describe how it is applied in the context of this paper. The stick breaking construction imposes a prior on the probability $\boldsymbol{\gamma}$ of community assignment for each object. Suppose we have a stick with unit length. At each time, the stick-breaking construction draws γ_i by breaking a portion, which is determined by the Beta distribution $\text{Beta}(1, \alpha)$, from the remaining stick. The larger the concentration parameter α, the more the number of communities that are constructed. The HDP process introduces a two-level prior to capture the community evolution. In the first level, $\boldsymbol{\gamma}$ is drawn from the stick-breaking construction [23] with a concentration parameter α, i.e.,

$$\boldsymbol{\gamma} \sim \text{GEM}(\alpha) \tag{9}$$

Here, GEM denotes the initials of the authors[1] of the stick breaking convention [23].

Each probability $\boldsymbol{\pi}_k$ from $\boldsymbol{\pi}$, denotes the transition probability from community k. This probability is assumed to be drawn from the Dirichlet Process $\text{DP}(\beta, \boldsymbol{\gamma})$ with concentration parameter θ.

$$\boldsymbol{\pi}_k \sim \text{DP}(\theta, \boldsymbol{\gamma}), k = 1, 2, \cdots$$

Thus, all transition probabilities $\{\boldsymbol{\pi}_k\}_{k=1}^{+\infty}$ are drawn from Dirichlet Processes with the same base distribution $\boldsymbol{\gamma}$.

As in Eq. (4), we can obtain the joint probability of the community configurations by marginalizing out $\boldsymbol{\gamma}$ and $\boldsymbol{\pi}$ over the afore-mentioned HDP prior. This results in a two-level probabilistic model that can be explained by the Chinese Restaurant process [3]. Suppose there exists an object in community k at the current moment. Then, the HDP prior assumes that at the top level, the probability that an object in community k evolves to community l is proportional to the number of the times that the same transition occurs before; and with probability proportional to θ, an oracle sampling process is invoked for modeling at the bottom level. At the bottom level, the oracle process determines the probability that an object evolves to community l is proportional to the number of the times that community l has been sampled by this oracle process; and with probability proportional to α, the object becomes the solitary member of a newly created community. We refer interested readers to [3] for more details on the HDP prior.

4. GIBBS SAMPLING FOR IC-DRF

The Gibbs sampling process is critical in efficient online maintenance of the social sensing model, while retaining a high level of accuracy. Based on the probabilistic model of Eq. 4, our goal is to incrementally maintain the most probable community configurations. In other words, we aim to maximize the following conditional probability given the previous community configurations and the current member locations:

$$\begin{aligned}\mathcal{Z}^{(T)\star} &= \arg\max_{\mathcal{Z}^{(T)}} P(\mathcal{Z}^{(T)} | \mathcal{Z}^{(1:T-1)}, \mathcal{L}^{(T)}) \\ &= \arg\max_{\mathcal{Z}^{(T)}} P(\mathcal{Z}^{(T)}, \mathcal{Z}^{(1:T-1)} | \mathcal{L}^{(T)})\end{aligned} \tag{10}$$

The above optimization problem is computationally intractable, because it has an exponentially large solution space in terms of the number of objects. Therefore, we use Gibbs sampling to perform a faster approximation. In Gibbs sampling, we need to compute the conditional probability of the community assignment for one object given the community assignments of the other objects in each step. Then, a new community assignment of each object is sequentially drawn from this conditional probability to replace the old one.

In our *IC-DRF* approach, consider the community configuration $\mathcal{Z}^{(1:T-1)}$ at time $T-1$. At this point, we need to sample the community assignment $z_j^{(T)}$ for each object o_j at time T in sequence conditioned on the community assignments $\mathcal{Z}_{-j}^{(T)}$ of the other objects. Here, $\mathcal{Z}_{-j}^{(T)}$ denotes the community assignments of the objects at time T leaving out $z_j^{(T)}$. According to the Chinese Restaurant process, the conditional probability $P(z_j^{(T)} = l | \mathcal{Z}^{(1:T-1)}, \mathcal{Z}_{-j}^{(T)})$ can be computed by a two-level process depending on whether the oracle process is invoked. We use $u_j^{(T)} = 1$ to indicate that the oracle is invoked or $u_j^{(T)} = 0$ otherwise.

At the top level, when the oracle is not invoked ($u_j^{(T)} = 0$), the probability of assigning object o_j to community l is as follows:

$$\begin{aligned}&P(z_j^{(T)} = l, u_j^{(T)} = 0 | \mathcal{Z}^{(1:T-1)}, \mathcal{Z}_{-j}^{(T)}, z_j^{(T-1)} = k) \\ &\propto n_{k\to l,-j}^{(1:T)} \exp(-E(z_j^{(T)}, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)}))\end{aligned} \tag{11}$$

Here, we emphasize that object o_j has been assigned to community k at time $T-1$ in the condition, i.e., $z_j^{(T-1)} = k$; and $n_{k\to l,-j}^{(1:T)}$ is the number of the times of community evolutions from community k to l, leaving out the evolution of object o_j at time $T-1$, i.e.,

$$\begin{aligned}n_{k\to l,-j}^{(1:T)} &= \sum_{o_i \in O} \sum_{t=2}^{T-1} \delta \left[\left[z_i^{(t-1)} = k, z_i^{(t)} = l \right]\right] \\ &+ \sum_{o_i \in O, i \neq j} \delta \left[\left[z_i^{(T-1)} = k, z_i^{(T)} = l \right]\right]\end{aligned} \tag{12}$$

Otherwise, the probability of invoking the oracle process is as follows:

$$\begin{aligned}&P(u_j^{(T)} = 1 | \mathcal{Z}^{(1:T-1)}, \mathcal{Z}_{-j}^{(T)}, z_j^{(T-1)} = k) \\ &\propto \alpha f_{\text{oracle}}(z_j^{(T)}, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)})\end{aligned} \tag{13}$$

Here, $f_{\text{oracle}}(z_j^{(T)}, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)})$ denotes the expected value of the potential function when the oracle process is invoked:

$$\begin{aligned}&f_{\text{oracle}}(z_j^{(T)}, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)}) \\ &= \sum_{l=1}^{L_{\max}} \frac{m_{l,-j}^{(1:T)}}{m_{\cdot,-j}^{(1:T)} + \theta} \exp\left(-E(z_j^{(T)} = l, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)})\right) \\ &+ \frac{\theta}{m_{\cdot,-j}^{(1:T)} + \theta} \exp\left(-E(z_j^{(T)} = \text{new}, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)})\right)\end{aligned} \tag{14}$$

where $L_{\max}$ is the number of the existing communities so far, and $m_{l,-j}^{(1:T)}$ is the number of the times that community

[1] Griffiths, Engen and McCloskey

l has been chosen by the oracle for the objects excluding object j at time T, i.e.,

$$\begin{aligned} m_{l,-j}^{(1:T)} &= \sum_{o_i \in O} \sum_{t=1}^{T-1} \delta \left[\left[z_i^{(t)} = l, u_i^{(t)} = 1 \right]\right] \\ &+ \sum_{o_i \in O, i \neq j} \delta \left[\left[z_i^{(T)} = l, u_i^{(T)} = 1 \right]\right] \end{aligned} \tag{15}$$

and

$$m_{\cdot,-j}^{(1:T)} = \sum_{l=1}^{L_{\max}} m_{l,-j}^{(1:T)} \tag{16}$$

Combining Eq. (11) and Eq. (13), we have the following sampling probability to decide which community is sampled to be assigned to object o_j at the top level, or instead the oracle is invoked as follows:

$$\begin{aligned} & P(z_j^{(T)} = l, u_j^{(T)} = 0 | \mathcal{Z}^{(1:T-1)}, \mathcal{Z}_{-j}^{(T)}, z_j^{(T-1)} = k) \\ &= \frac{n_{k\to l,-j}^{(1:T)} \exp(-E(z_j^{(T)}, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)}))}{\alpha f_{oracle}(z_j^{(T)}, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)}) + \sum_{s=1}^{L_{\max}} n_{k\to s,-j}^{(1:T)} \exp(-E(z_j^{(T)} = s, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)}))} \end{aligned} \tag{17}$$

and

$$\begin{aligned} & P(u_j^{(T)} = 1 | \mathcal{Z}^{(1:T-1)}, \mathcal{Z}_{-j}^{(T)}, z_j^{(T-1)} = k) \\ &= \frac{\alpha f_{\text{oracle}}(z_j^{(T)}, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)})}{\alpha f_{\text{oracle}}(z_j^{(T)}, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)}) + \sum_{s=1}^{L_{\max}} n_{k\to s,-j}^{(1:T)} \exp(-E(z_j^{(T)} = s, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)}))} \end{aligned} \tag{18}$$

If $u_i^{(T)} = 1$ is sampled at the top level, the oracle is invoked, and we have the following conditional probability of sampling an existing community for $z_j^{(T)}$:

$$P(z_j^{(T)} = l | \mathcal{Z}_{-j}^{(T)}, u_j^{(T)} = 1) \propto m_{l,-j}^{(1:T)} \exp(-E(z_j^{(T)}, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)})) \tag{19}$$

Otherwise, we have a new community sampled for o_j:

$$\begin{aligned} & P(z_j^{(T)} = \text{new} | \mathcal{Z}_{-j}^{(T)}, u_j^{(T)} = 1) \\ & \propto \theta \exp(-E(z_j^{(T)} = \text{new}, \mathcal{Z}_{-j}^{(T)} | \mathcal{L}^{(t)})) \end{aligned} \tag{20}$$

Algorithm 1 summarizes the Gibbs sampling procedure for community configuration $\mathcal{Z}^{(T)}$ at each time T. We can see that the only role of the previously determined configurations $\mathcal{Z}^{(1:T-1)}$ in this sampling procedure is to count the statistics n and m for each object. Hence, we only need to store and keep these counts updated at each time. Actually, we only need these counts and the community assignments for objects at the previous time $\mathcal{Z}^{(T-1)}$ in order to perform the sampling procedure. The other information in $\mathcal{Z}^{(1:T-1)}$ can be disregarded. This will save space for storing the past community information and make the online inference process of the *IC-DRF* model more efficient.

In each sampling step, only the community assignment for one object changes, and the assignments for the other objects remain the same. Therefore, when we compute the location-sensitive energy, we do not need recompute it. Instead, we can store the total values of intra-community and inter-community distances in two auxiliary variables and update them with the changed community assignment in each sampling step. Moreover, since we will use DBSCAN to cluster all the GPS locations to a much smaller number of semantic locations in real systems, the distances between these semantic locations can be precomputed and stored in a lookup matrix. This will save a lot of computing time in sampling procedure. Thus, the computational complexity in each time stamp is $O(|\mathcal{O}|L_{\max}^2 D)$, which is linear to the number of objects $|\mathcal{O}|$. This shows that our algorithm can well scale to large number of objects.

The model of this paper also ensures that the community membership of an object is determined by its presence in a community over a significant period of time rather than at one occasional moment, which we refer to as *temporal consistency.* To reflect such temporal consistency, we add a positive value n_0 to the community evolution count $n_{k\to k,-j}^{(1:T)}$ for each community k. The larger n_0 is, the longer one community tends to last over time with a stronger temporal consistency effect. This is because it is not sufficient to determine the community assignment of an object only based on its location at a particular moment. Community members may be co-located at a location by chance. Moreover, the reported object locations can be erroneous and noisy, because of the inherent limitations of such location tracking technology. The temporal consistency of our community assignment approach can reduce the negative impact of the limitations of data collection, and make our approach more robust in real scenarios.

5. EXPERIMENTAL RESULTS

In this section, we evaluate the proposed *IC-DRF* model compared with the other state-of-the-art algorithms. We will evaluate the approach for effectiveness, efficiency, and the effectiveness of incorporating content information.

5.1 Data Sets

We evaluate the proposed algorithm on the *GeoLife* data set [28], which recorded the trajectories of 164 users with GPS devices from April 2007 to October 2009 in the city of Beijing, China. It contains more than $12,000$ different GPS trajectories over $139,310$ kilometers in geographical scale.

Most of these trajectories were recorded at a sampling of 1-5 seconds or every 5-10 meters per point. This is not necessarily desirable, since locations do not change very significantly in small time frames. In the experiments, we apply the DBSCAN to cluster all recorded geo-locations to a set of "semantic" places. Our idea is the frequently visited locations correspond to some places with specific activity semantic meaning. For example, they can be a meeting room in a business building, a fitness facility, or a restaurant. Such geo-location clustering algorithms have been applied in literature [28] to automate the discovery of semantic places, and good performances have been observed on GeoLife dataset. Thus, we believe people who visit these semantic places simultaneously can be considered to be involved in the same activity. We construct the trajectory for each user by connecting their visited semantic places, and sample the trajectories every 10 minutes to form the dynamic networks at a set of discrete time stamps for community detection. The data set also contains the transportation modes, such as walk, bike, bus and car & taxi, associated with some users. We can use them as meta-information for user status in the energy function as Subsection 3.2. However, the transportation modes are incomplete in the data set, and we will simply skip them in the energy function when they are missing.

Since our metrics will require a ground-truth for evaluation purposes, we used the social similarity between users [29]. The degree of social closeness ranged from $+1$ (close) to -1 (not close). Figure 1 illustrates the social relations be-

Algorithm 1: Online inference for community configuration

```
Input: The community configurations $\mathcal{Z}^{(1:T-1)}$ from time 1 to $T-1$, $\alpha$, $\theta$ and the number of sampling loops $D$
Output: The sampled community assignments.
Randomly initialize $\mathcal{Z}^{(T)}$ and compute the number of existing community $L_{\max}$ according;
for $d = 1$ to $D$ do
    foreach object $o_j$ do
        Count $n^{(1:T)}_{k\to l,-j}$ and $m^{(1:T)}_{l,-j}$ based on $\mathcal{Z}^{(1:T-1)}$ and the currently sampled $\mathcal{Z}^{(T)}$;
        Compute the conditional probabilities in Eq. (17) and (18):
            $P(z_j^{(T)} = l, u_j^{(T)} = 0 | Z^{(1:T-1)}, \mathcal{Z}_{-j}^{(T)}, z_j^{(T-1)} = k) = \frac{n^{(1:T)}_{k\to l,-j}\exp(-E(z_j^{(T)}, \mathcal{Z}_{-j}^{(T)}|\mathcal{L}^{(t)}))}{\alpha f_{\text{oracle}}(z_j^{(T)}, \mathcal{Z}_{-j}^{(T)}|\mathcal{L}^{(t)}) + \sum_{s=1}^{L_{\max}} n^{(1:T)}_{k\to s,-j}\exp(-E(z_j^{(T)}=s, \mathcal{Z}_{-j}^{(T)}|\mathcal{L}^{(t)}))}$
        and
            $P(u_j^{(T)} = 1 | \mathcal{Z}^{(1:T-1)}, \mathcal{Z}_{-j}^{(T)}, z_j^{(T-1)} = k) = \frac{\alpha f_{\text{oracle}}(z_j^{(T)}, \mathcal{Z}_{-j}^{(T)}|\mathcal{L}^{(t)})}{\alpha f_{\text{oracle}}(z_j^{(T)}, \mathcal{Z}_{-j}^{(T)}|\mathcal{L}^{(t)}) + \sum_{s=1}^{L_{\max}} n^{(1:T)}_{k\to s,-j}\exp(-E(z_j^{(T)}=s, \mathcal{Z}_{-j}^{(T)}|\mathcal{L}^{(t)}))}$ ;
        Sample $z_j^{(T)}$ and $u_j^{(T)}$ based on the above probabilities;
        if $u_j^{(T)} = 1$ is sampled then
            Compute the conditional probabilities according to Eq. (19) and (20):
                $P(z_j^{(T)} | \mathcal{Z}_{-j}^{(T)}, u_j^{(T)} = 1)$
                $= \begin{cases} \frac{m^{(1:T)}_{l,-j}\exp(-E(z_j^{(T)}=l, \mathcal{Z}_{-j}^{(T)}|\mathcal{L}^{(t)}))}{\sum_{s=1}^{L_{\max}} m^{(1:T)}_{s,-j}\exp(-E(z_j^{(T)}=s, Z_{-j}^{(T)}|\mathcal{L}^{(t)})) + \theta\exp(-E(z_j^{(T)}=\text{new}, \mathcal{Z}_{-j}^{(T)}|\mathcal{L}^{(t)}))}, & \text{an existing community } l \text{ is sampled for } z_j^{(T)} \\ \frac{\theta\exp(-E(z_j^{(T)}=new, \mathcal{Z}_{-j}^{(T)}|\mathcal{L}^{(t)}))}{\sum_{s=1}^{L_{\max}} m^{(1:T)}_{s,-j}\exp(-E(z_j^{(T)}=s, \mathcal{Z}_{-j}^{(T)}|\mathcal{L}^{(t)})) + \theta\exp(-E(z_j^{(T)}=\text{new}, \mathcal{Z}_{-j}^{(T)}|\mathcal{L}^{(t)}))}, & \text{a new community is sampled for } z_j^{(T)} \end{cases}$ ;
            Sample $z_j^{(T)}$ based on the above probabilities;
            if $z_j^{(T)} = new$ is sampled then
                $L_{\max} \leftarrow L_{\max} + 1$;
                $z_j^{(T)} = L_{\max}$;
            end
        end
    end
end
```

tween these 164 users. The colors near the red end denote stronger social relations between users, whereas the colors near the blue end illustrate the weaker social relations.

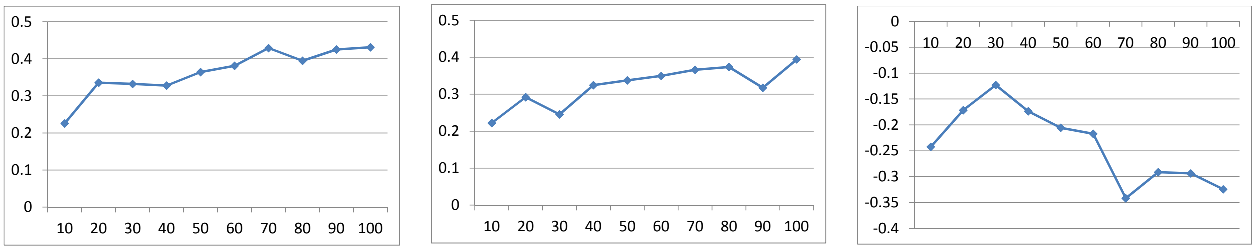
5.2 Evaluation Metrics

We used the following three metrics at different time stamps to evaluate the community detection results, and reported the average values of the metrics over the intervals of three years (i.e., 2007, 2008 and 2009) for reporting purposes. This compares the performance changes of different algorithms over years.

- **Pearson:** The first measure was the Pearson's coefficient between the ground truth social relations and the detection results. For each pair of users, we denote their *detected* social relation as +1 if they belong to the same community (as found by the algorithm), and −1 otherwise. Then, we computed the Pearson's coefficient between the *ground truth* social relations and the *detected* social relations. Higher values of the Pearson's coefficients are indicative of good community structure in terms of correspondence with the ground truth.
- **INTRA:** This is the average of intra-community (ground-truth) social closeness. This metric is computed by averaging the social closeness over all pairs of intra-community users. A large value of INTRA suggests that the users assigned to the same community are more likely to be familiar with one another. In an indirect sense, this metric measures the true positives in the community detection results.
- **INTER:** This is the average of inter-community social closeness. This metric is computed by averaging the (ground-truth) social closeness over all pairs of inter-community users. A small value of INTER indicates that the users assigned to the different communities are not likely to be socially close to each other. In this sense, the negated INTER measures the true negatives in the community detection result.

5.3 Baselines

We compared the proposed *IC-DRF* model with three other community detection algorithms: (1) *ObjectGrowth* [18] which finds the swarms of moving objects in the same community;(2) *Dynamic stochastic block model (DSBM)* [26] which extends the stochastic block model to dynamic social



(a) Pearson v.s. α

(b) INTRA v.s. α

(c) INTER v.s. α

Figure 2: Parameter sensitivity of quality with respect to α, which influences new community formation.

Table 1: Comparison of different community detection algorithms on the *GeoLife* data set between 2007-2009. The best performance is highlighted in bold.

Algorithms	2007			2008			2009		
	Pearson	*INTRA*	*INTER*	*Pearson*	*INTRA*	*INTER*	*Pearson*	*INTRA*	*INTER*
ObjectGrowth	0.2537	0.3043	-0.1124	0.2654	0.2374	-0.153	0.2759	0.2941	-0.1361
DSBM	0.2453	0.1829	-0.1657	0.2631	0.2971	-0.1232	0.2748	0.2637	-0.1489
dIRM	0.2542	0.2818	-0.1208	0.2742	0.2816	-0.1398	0.297	0.2941	-0.1569
IC-DRF	**0.3430**	**0.3302**	**-0.1864**	**0.3606**	**0.3256**	**-0.2081**	**0.3956**	**0.3168**	**-0.2557**

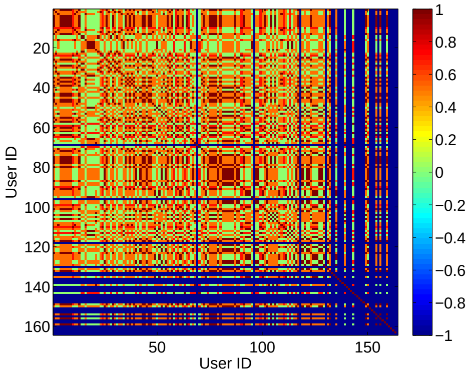


Figure 1: Illustration of social closeness between users. The social closeness between the user ranges from from a value of 1 (the closest) to -1 (the least close). Several users with IDs after 130 **have very weak social connections with others. These users tend to form isolated communities with smaller size.**

networks. This model also considers the dynamic evolution between communities in a transition probabilistic model;(3) *Dynamic Infinite Relational Model (dIRM)* [12] which also extends the *DSBM* to consider the dynamic and time-sensitive changes in the structure of the relational data.

5.4 Results

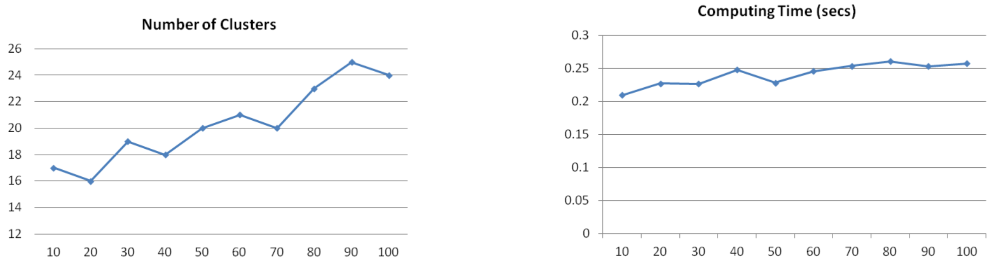
In Table 1, we report the results of the four compared algorithms on three time segments of the *GeoLife* data set. In order to show the experimental results over different periods, we evaluated the community detection results on three segments, corresponding to the 2007, 2008, and 2009 years. We reported the average value of Pearson, INTRA and INTER over these periods.

It is evident that the proposed *IC-DRF* model achieves the best performance over all the temporal segments among the compared algorithms in terms of all metrics. Furthermore, the performance improves over the different years, because a larger number of GPS data points progressively improves the incrementally updated probabilistic model over time. In general, the community evolution counts became more stable and robust as the repeated patterns in the community evolution dynamics were learned by the probabilistic model of *IC-DRF*. Thus the model estimation improved over time, and is manifested in the more accurate results for later periods in the community detection process. This is particularly useful in the context of applications in which the community detection process is likely to be executed over long periods of time.

The results reported above are obtained with $\alpha = 80$, $\theta = 80$ and $n_0 = 20$. While the number of communities is determined in a data-driven manner, the value of α still continues to have some indirect influence on it, since it affects the rate of new community formation. Therefore, it is useful to test the impact of α on the quality of the underlying communities. In Figure 2, we illustrate the parameter sensitivity of the *IC-DRF* model in terms of three metrics when α changes from 10 to 100. We can see that as long as α is not too small, the performance does not change too much. This shows the robustness of the approach. Moreover, Figure 2(a) illustrates the variation in the number of communities with respect to different values of α. As we expect, the number of communities tend to increase with larger values of α.

5.5 Computational Efficiency Results

We also studied the computational efficiency of the *IC-DRF* model for updating the community configurations in each time stamp. Since our approach uses an online incremental inference paradigm, it is critical for the model to be efficient enough to make simultaneous decisions about the evolution dynamics of community membership of different users in an incremental way. Recall that our use of



(a) Number of clusters v.s. α

(b) Computational time v.s. α

Figure 3: The variation in (a) the number of communities and (b) computational time (seconds) of *IC-DRF* with α.

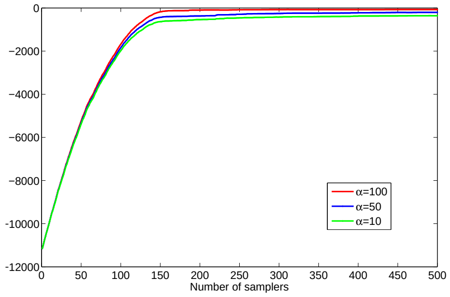


Figure 4: The potential function with increasing number of samples. From about the 160th samplers, the sampling algorithm begins to converge. After about 300 samplers, the potential function closely converges to the maximum value. This indicates that the most probable community configuration can be robustly obtained at that point.

the Gibbs sampling process is motivated by its efficiency in maintaining the incremental model. Our approach is influenced by some parameters in Gibbs sampling process, such as α, since it indirectly influences the number of communities. Figure 3(b) illustrates the computational time versus α. The value of α is illustrated on the X-axis, whereas the computational time is illustrated on the Y-axis. It is evident that while the computational time changes with α, the sensitivity is relatively small. Therefore, the computational efficiency also tends to be quite robust with α.

We also illustrated the efficiency of the Gibbs sampling process in terms of its convergence behavior over a small number of samples. Specifically, we plot the logarithm of the potential function with increasing number of Gibbs samplers in Figure 4. Recall that the potential function is the sum of the energy function and the logarithm of HDP likelihood. The number of Gibbs samples are illustrated on the X-axis, whereas the potential function is illustrated on the Y-axis. We can see that after about 160 samples, the potential function converges to the maximum value (as averaged over all users). We further note that the potential function values were recorded at the early second loop of the Gibbs sampling. Therefore, after the second loop of Gibbs sampling, we can stop and obtain the most probable community configuration with these parameters. This demonstrates the Gibbs sampling process can efficiently infer the most probable community assignment.

5.6 Incorporating Content Information: Transportation Modes

Finally, we show the impact of incorporating additional content-based meta-information into the community detection process, as discussed at the end of subsection 3.2. We will investigate that the additional labeling information can improve the community detection accuracy. Specifically, we use the transportation modes in the data set as the meta-information. These are incorporated in the energy function, as discussed in subsection 3.2. Table 2 presents the results by comparing the models with and without transportation modes in the *IC-DRF* model. We can see that the model with transportation modes performs better than its counterpart, which did not use the transportation modes. We can also see that the value of INTER metric is improved more than the value of INTRA metric. It is reasonable and expected, because users that belong to different communities can sometimes be recognized by very wide variation in their transportation modes. On the the other hand, the improvement of INTER is not obvious since two users in the same transportation mode do not necessarily belong to the same community. It is important to recognize that this additional meta-information is simply additional knowledge to *improve* the community detection process, for which the primary model is the probabilistic spatio-temporal analysis. The transportation modes simply provide additional information for more effective community discovery. Nevertheless, this is a useful feature to incorporate in the detection process, since additional meta-information is often available in many sensing scenarios.

6. CONCLUSION

In this paper, we investigated the dynamic and evolving community formation from the social structure implied by the distance behavior of mobile sensors. The proposed *IC-DRF* algorithm defines dynamic random fields over time that can determine and adjust the community structures on the basis of the evolving spatio-temporal locality. At the same time, the number of communities is determined dynamically and automatically at any given time, so that the model is

Table 2: Comparison of *IC-DRF* model with and without content information (transportation modes). The best performance is highlighted in bold.

	without using transportation modes			with using transportation modes		
Dataset	Pearson	INTRA	INTER	Pearson	INTRA	INTER
2007	0.3430	**0.3302**	-0.1864	**0.3458**	0.2924	**-0.2116**
2008	0.3606	0.3256	-0.2081	**0.3811**	**0.3494**	**-0.2172**
2009	0.3956	0.3168	**-0.2557**	**0.3999**	**0.3309**	-0.2508

able to effectively adapt to global changes in the granularity of the natural community structure. An efficient Gibbs sampling algorithm is used in order to speed up the approach, so that it can be used in an online incremental manner. This makes the *IC-DRF* sufficiently scalable in order to handle large amounts of incoming GPS data in real time. The experimental results on the real dataset show the effectiveness of the proposed model as compared to other state-of-the-art algorithms.

Acknowledgments

Research was sponsored by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 and National Science Foundation (NSF) under Grant IIS-1144111. Mr. Guo-Jun Qi was also supported in part by an IBM fellowship.

7. REFERENCES

[1] C. Aggarwal, T. Abdelzaher. *Social Sensing.* Managing and Mining Sensor Data, Springer, 2013.

[2] L. Backstrom, D. P. Huttenlocher, J. M. Kleinberg, X. Lan. Group formation in large social networks: membership, growth, and evolution, *KDD*, 2006.

[3] M.J. Beal, Z. Ghahramani, C. Rasmussen. The infinite hidden Markov Model, *NIPS Conference*, 2001.

[4] M. Benkert, J. Gudmundsson, F. Hubner, T. Wolle. Reporting flock patterns, *COMGEO*, 2008.

[5] C. Aggarwal, H. Wang. *Managing and Mining Graph Data*, Springer, 2010.

[6] C. Aggarwal. *Social Network Data Analytics*, Springer, 2011.

[7] D. Chakrabarti, R. Kumar, A. Tomkins. Evolutionary clustering, *KDD*, 2006.

[8] Y. Chi, X. Song, D. Zhou, K. Hino, B. L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness, *KDD*, 2007.

[9] E. Cho, S. Myers, J. Leskovec. Friendship and Mobility: User Movement In Location-Based Social Networks. *KDD*, 2011.

[10] A. Clauset, M. E. J. Newman, C. Moore. Finding community structure in very large networks, *Phys. Rev. E 70, 066111*, 2004.

[11] J. Gudmundsson, M. van Kreveld. Computing longest duration flocks in trajectory data. *GIS*, 2006.

[12] K. Ishiguro, T. Iwata, N. Ueda, J. Tenenbaum. Dynamic Infinite Relational Model for Time-varying Relational Data Analysis, *NIPS*, 2010.

[13] H. Jeung, M. L. Yiu, X. Zhou, C. Jensen, H. Shen. Discovery of Convoys in Trajectory Databases. *VLDB*, 2008.

[14] P. Kalnis, N. Mamoulis, S. Bakiras. On discovering moving clusters in spatio-temporal data. *SSTD*, 2005.

[15] R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins. Trawling the web for emerging cyber-communities, *Computer Networks*, 31(11-16), pp. 1481–1493, 1999.

[16] J. Leskovec, K. J. Lang, A. Dasgupta, M. W. Mahoney. Statistical properties of community structure in large social and information networks, *WWW*, 2008.

[17] J.-G. Lee, J. Han, K.-Y. Whang. Trajectory Clustering: A Partition and Group Framework, *SIGMOD*, 2007.

[18] Z. Li, B. Ding, J. Han, R. Kays. Swarm: Mining Relaxed Temporal Moving Object Clusters, *VLDB Conference*, 2010.

[19] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, A. Kelliher. Extracting community structure through relational hypergraphs, *WWW*, 2009.

[20] A. Noulas, S. Scellato, C. Mascolo, M. Pontil. An empirical study of geographic user activity patterns in foursquare, *ICWSM*, 2011.

[21] G. Qi, C. Aggarwal, T. Huang. Community Detection with Edge Content in Social Media Networks, *ICDE Conference*, 2012.

[22] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo. Socio-spatial properties of online location-based social networks, *ICWSM*, 2011.

[23] J. Sethuraman. A constructive definition of dirichlet priors, *Statistica Sinica*, 4:639–650, 1994.

[24] D. Wang, D. Pedreschi, C. Song, F. Giannotti, A. Barabasi. Human mobility, social ties, and link prediction, *KDD*, 2011.

[25] T. Yang, R. Jin, Y. Chi, S. Zhu. Combining link and content for community detection: a discriminative approach, *KDD*, 2009.

[26] T. Yang, Y. Chi, S. Zhu, Y. Gong, R. Jin. Detecting communities and their evolutions in dynamic social networks: a Bayesian approach, *Maching Learning*, 82: pp. 157–189, 2011.

[27] Y. Zhou, H. Cheng, J. X. Yu. Graph clustering based on structural/attribute similarities, *PVLDB*, 2(1), pp. 718–729, 2009.

[28] Y. Zheng, L. Zhang, X. Xie, W.-Y. Ma. Mining interesting locations and travel sequences from GPS trajectories, *WWW*, 2009.

[29] V. Zheng, B. Cao, Y. Zheng, X. Xie, Q. Yang. Collaborative Filtering Meets Mobile Recommendation: A User centered Approach, *AAAI*, 2010.