# Geo-social Clustering of Places from Check-in Data

Shivam Srivastava*
Samsung Research
Bagmane Constellation Business Park
Bangalore, India
shivam.srivastava511@gmail.com

Shiladitya Pande
Dept. of CSE
IIT Madras
Chennai, India
spande@cse.iitm.ac.in

Sayan Ranu
Dept. of CSE
IIT Madras
Chennai, India
sayan@cse.iitm.ac.in

***Abstract*—In this paper, we develop an algorithm to cluster places not only based on their locations but also their semantics. Specifically, two places are considered similar if they are spatially close and visited by people of similar communities. With the explosion in the availability of location-tracking technologies, it has become easy to track locations and movements of users through user "check-ins". These check-ins provide insights into the community structure of people visiting a place, which is leveraged and integrated into the proposed *geo-social* clustering framework called *GeoScop*. While community detection is typically done on social networks, in our problem, we lack any network data. Rather, two people belong to the same community if they visit similar geo-social clusters. We tackle this chicken-and-egg problem through an iterative procedure of *expectation maximization* and *DBSCAN*. Extensive experiments on real check-in data demonstrate that GeoScop mines semantically meaningful clusters that cannot be found by using any of the existing clustering techniques. Furthermore, GeoScop is up to 6 times more pure in social quality than the state-of-the-art technique. The executables for the tool are available at http://www.cse.iitm.ac.in/∼sayan/software.html.**

## I. Introduction

Spatial clustering is an unsupervised algorithm to group places into clusters, such that those within a cluster are similar, and places across clusters are dissimilar. Traditionally, the distance function between two places quantifies the spatial distance between them and ignores any semantic similarity. In certain applications, such a distance function leads to spurious clusters. Consider Fig. 1. Each dot denotes a "check-in" by a user. Let the blue check-ins represent visits by students and the green check-ins represent tourists. If the check-ins are clustered using traditional spatial-clustering algorithms, then all check-ins in the zoo, university, book store, and the library would form a single cluster, and the two white check-ins would be outliers. However, it is easy to see that their are two *geo-social* clusters. While the university, library and book store are mostly visited by students, the zoo is frequented by tourists.

In this paper, we develop a technique called *GeoScop (GEO-Social Clustering Of Places)* to mine geo-social clusters from check-in data. Due to the ubiquity of GPS-enabled phones, there has been a spurt of services, such as Yelp and Zomato, which are built on check-in data. These check-ins provide a rich form of geographical data, the analysis of which provides new and interesting insights, compared to raw spatial data.

*The work was done at IIT Madras

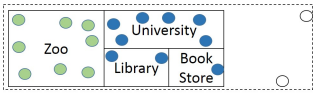



Fig. 1. A scenario where spatial clustering leads to spurious results.

The closest work to our problem is DCPGS[1]. Although the formulation of DCPGS is different, it has the same goal of grouping places based on both spatial distance and social distance. However, DCPGS assumes the presence of an observable social network to mine communities of users. In our problem, we have access to only the check-in data and lack the social network of users. While it is always useful to have additional information in the form of the social network, in many scenarios, this assumption is not realistic. It is well known that only a few companies, such as Facebook and Twitter, have access to large scale social networks. For the majority of the companies like Yelp, Zomato, TripAdvisor, and credit card agencies, the only source of data are the check-ins. Social network owners are known to not share their data due to competitive business advantage, its inestimable value[2], as well as privacy concerns. Therefore, the question that arises is the following: *How do we perform geo-social clustering without knowledge of the explicit network among users?* Essentially, we face a chicken-and-egg problem. Two users are part of the same community if they visit similar geo-social clusters. On the other hand, two places are similar if they are visited by people of similar communities. We solve this problem through the iterative procedure of community detection and geo-social clustering as outlined in Fig. 2. To summarize, the contributions of our work are as follows.

- We formulate the novel problem of *geo-social clustering* from check-in data. As a by-product of our problem, we also mine *geo-social* communities.
- We develop a technique called *GeoScop* to mine geo-social clusters from check-in data without relying on a social network.
- Extensive experiments on real check-in data establish that GeoScop is able to mine clusters that are semantically meaningful and up to 6 times more pure than the state-of-the-art techniques.

## II. Problem Formulation

The data input to GeoScop is a set of *check-ins*.

1550-4786/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDM.2015.16



IEEE computer society

***Definition** 1:* CHECK-IN. *A check-in is a tuple $\langle u, x, y, t\rangle$, where u represents the user id of the person generating the visit, x,y are the spatial coordinates of the location, and t is the timestamp at which the visit occurred.*

Let $\mathcal{S}$ be the set of all the unique locations in $\mathbb{D}$ in terms of their spatial co-ordinates. Thus, alternatively, a check-in is represented as $\langle u, p, t\rangle$, where $p \in \mathcal{S}$. Similar to places, we use the notation $\mathcal{U}$ to denote the set of unique users in $\mathbb{D}$.

Since our goal is to perform geo-social clustering among places, we define a distance function for the geographical world, denoted as $gdist(p_1, p_2)$ and the social world, denoted as $sdist(p_1, p_2)$. $gdist(p_1, p_2)$ is simply the Euclidean distance between $p_1$ and $p_2$. To define the social distance, assume there is some oracle that partitions all users in the check-in dataset into disjoint sets of *communities* $\mathbb{C} = \{C_1, \cdots, C_m\}$, where $C_i \in \mathbb{C}$ is a set of users, and $C_i \cap C_j = \emptyset$. Each community contains users of similar tastes. We now construct a community profile $pr(p)$ for each place $p \in \mathcal{S}$.

***Definition** 2:* COMMUNITY PROFILE. *The community profile $pr(p)$ of a place p is a vector of dimension $\|\mathbb{C}\|$. The $i^{th}$ dimension, denoted as $pr(p)[i]$, contains the number of visits to p from community $C_i \in \mathbb{C}$. Mathematically, $pr(p)[i] = \|\{u \in \mathcal{U} | \exists \langle u, p, t\rangle \in \mathbb{D},\ u \in C_i\}\|$.*

***Definition** 3:* SOCIAL DISTANCE. *Given, the community profiles of places $p_1$ and $p_2$, the social distance $sdist(p_1, p_2)$ is defined as the following.*

$$sdist(p_1, p_2) = 1 - \frac{\sum_{\forall C_i \in \mathbb{C}} \min\{pr(p_1)[i], pr(p_2)[i]\}}{\sum_{\forall C_i \in \mathbb{C}} \max\{pr(p_1)[i], pr(p_2)[i]\}} \quad (1)$$

GeoScop is modeled on DBSCAN[3]. A place $p$ is a *core* point if it contains at least $minPts$ points in its *geo-social* neighborhood $\mathcal{N}_{\delta_g, \delta_s}(p) = \{p' \in \mathcal{S} |\ gdist(p, p') \leq \delta_g,\ \ sdist(p, p') \leq \delta_s\}$, where $\delta_g$ and $\delta_s$ are user-provided geographical distance and social distance thresholds. A place $p'$ is *directly density reachable* from $p$ if $p' \in \mathcal{N}_{\delta_g, \delta_s}(p)$ and $p$ is a core point. $p'$ is *density reachable* from $p$ if there exists a chain of places $p_1, \cdots, p_n$, such that $p_1 = p$, $p_n = p'$ and each $p_{i+1}$ in the chain is directly density reachable from $p_i$. Finally, $p'$ and $p''$ are *density connected* if $\exists p \in \mathcal{S}$, such that both $p'$ and $p''$ are density reachable from $p$. Given the set of places $\mathcal{S}$, our goal is to partition it into a set of *geo-social clusters* $\mathbb{P} = \{P_1, \cdots, P_m\}$ with the following properties.

1) $\forall P_i \in \mathbb{P}, \forall p, p' \in P_i$, $p$ is density connected to $p'$.
2) If $p \in P_i, P_i \in \mathbb{P}$, and $p'$ is density reachable from $p$, then $p' \in P_i$.

## III. GEOSCOP

Geo-social clustering is straightforward if the set of communities is known. A community is a set of users who have similar interests in visiting places. In other words, if user $u$ visiting a geo-social cluster $C$ increases the chances of user $u'$ also visiting $C$, then they are part of the same community. However, we do not have access to the underlying social network and mine the communities. We need to mine communities from just the check-ins. Towards that goal, we base our inference procedure on the following assumptions.

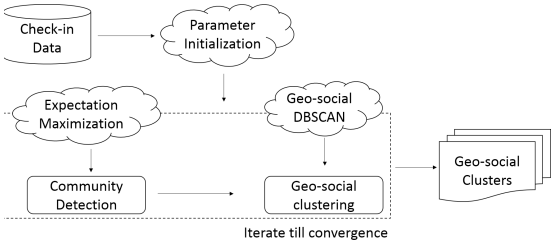



Fig. 2. The pipeline of GeoScop.

- There is an unobserved social network with well-defined communities of users. Connectivity among users in the community is dense and across communities is sparse.
- Each user $u$ influences all other members of his/her community $C$ with a certain intensity.
- Users' check-ins are governed by the *independent cascade* model[4]. Specifically, the chances of $u$ visiting a cluster $P$ depends on the influence exerted by the *active* members of $C$ on $u$. A user in $C$ gets active with respect to cluster $P$ after he/she visits $P$ for the first time.

A recent work called *C-IC* exists in mining communities from user activity data[5]. Each activity is represented by a tuple $\langle u, i, t\rangle$, where $i$ denotes a certain activity such as purchasing a product, sharing a photo, etc. C-IC builds on the assumptions that two users $u_1$, $u_2$ are likely to be part of the same community if $u_1$ performing an activity $i$ increases the chances of $u_2$ performing $i$ as well. Our problem maps to this same exercise where $i$ represents the place $p \in \mathcal{S}$ visited. However, in our problem, the influence exerted on a user is at the cluster level and not at the place level. The granularity of a place, which are the unique $\langle x, y\rangle$ co-ordinates, is extremely small. Two users visiting a zoo will not check-in at the exact same coordinates. The spatial extent of the zoo can be identified by clustering all check-ins from the zoo-loving community. We therefore adapt the C-IC model for our problem of geo-social clustering.

### A. Mining Communities

Let user $u$ visit a geo-social cluster $P$, and soon after, we observe user $u'$ also visiting $P$. Due to the small time difference in their visits, we hypothesize that $u$ may have influenced $u'$ and therefore they may belong to the same community. Now, if this event happens multiple times across various clusters, then the probability of these two users lying in the same community should further increase. In the description that follows, we see how it is modeled to arrive at communities.

We hypothesize that actions of users are governed by a set of parameters $\Theta$. The likelihood of the check-in data $\mathbb{D}$ is therefore expressed as a function of $\Theta$, i.e., $\mathcal{L}(\Theta; \mathbb{D}) = \prod_{\forall u \in \mathcal{U}} P(u|\Theta)$ where $P(u|\Theta)$ is the likelihood of $u$'s check-ins relative to the parameters. Our learning problem is to find $\hat{\Theta} = \arg\max_{\Theta}\{\mathcal{L}(\Theta; \mathbb{D})\}$.

We assume $u$'s check-ins are dependent on $u$'s community membership. To model this, we have a hidden binary variable $m_{u,i}$ denoting users $u$'s membership in community $i$ with the

constraint that $\sum_{i=1}^{K} m_{u,i} = 1$, where $K$ is the total number of communities. The parameter set $\Theta$ can therefore be partitioned into $\{\pi_1, \cdots, \pi_K, \Theta_1, \cdots, \Theta_K\}$, where $\pi_i = \frac{\sum_{\forall u \in \mathcal{U}} m_{u,i}}{\|\mathcal{U}\|}$ represents the prior probability $P(m_{u,i}) = 1$ and $\Theta_i$ is the parameter set for community $C_i$. The data likelihood is therefore re-expressed as follows.

$$\mathcal{L}(\Theta; \mathbb{D}) = \prod_{\forall u \in \mathcal{U}} \sum_{i=1}^{K} P(u|\Theta_i)\pi_i \tag{2}$$

We optimize this likelihood through the standard EM algorithm. The data likelihood can be expressed as $P(\mathbb{D}, \mathbf{M}, \Theta) = P(\mathbb{D}|\mathbf{M}, \Theta).P(\mathbf{M}|\Theta).P(\Theta)$ where

$$P(\mathbb{D}|\mathbf{M}, \Theta) = \prod_{\forall u \in \mathcal{U}} \sum_{i=1}^{K} P(u|\Theta_i)^{m_{u,i}} \tag{3}$$

$$P(\mathbf{M}|\Theta) = \prod_{\forall u \in \mathcal{U}} \sum_{i=1}^{K} \pi_i^{m_{u,i}} \tag{4}$$

Now, following standard EM manipulation, the $\mathcal{Q}$-*function* is

$$\mathcal{Q}(\Theta|\Theta') = E[\log P(\mathbb{D}, \mathbf{M}, \Theta|\mathbb{D}, \Theta')] \tag{5}$$

$$\propto \sum_{\forall u \in \mathcal{U}} \sum_{i=1}^{K} \gamma_{u,i} \{\log P(u|\Theta_i) + \log \pi_i\} \tag{6}$$

$$\text{where } \gamma_{u,i} = \frac{p(u|\Theta_i)\pi_i}{\sum_{j=1}^{K} P(u|\Theta_j)\pi_j} \tag{7}$$

We optimize $\gamma_{u,i}$ using the standard EM procedure. The parameter set $\Theta$ is first initialized and the $E$ and $M$ steps are performed recursively till convergence.

**E Step:** Given $\Theta$, estimate $\gamma_{u,i}$ for each $u, i$.

**M Step:** Given $\gamma_{u,i}$, maximize the $\mathcal{Q}$-function (Eq.5).

We simulate $P(u|\Theta_i)$ based on the Independent Cascade model (IC). We assume time unfolds in discrete timestamps (in our case 1 second). A user $u$ becomes active with respect to a cluster $P$ after visiting a place in $P$ for the first time. Now, $u$ has a single shot at influencing all other inactive members of $u$'s community in visiting $P$. Since, we do not have access to the network, we assume that $u$ exerts influence globally on all members of his/her community $C_i$ with a weight $p_u^i \in [0, 1]$. The core idea is that the community-level influence of $u$ is highest in the community $u$ belongs to.

Now, given that $u$ has visited cluster $P$, we identify the potential users who could have influenced $u$. These potential influencers are those who visited $P$ within a short time span prior to $u$. Mathematically, the influencers $F_{u,P}^{+}$ are

$$F_{u,P}^{+} = \{v \in \mathcal{U} | \exists \langle u, p_u, t_u \rangle, \langle v, p_v, t_v \rangle \in \mathbb{D}, \\ 0 \leq t_u - t_v \leq \Delta,\ p_u \in P,\ p_v \in P\} \tag{8}$$

In the same manner, $F_{u,P}^{-}$ defines the users who visited $P$, but potentially failed to influence $u$.

$$F_{u,P}^{-} = \{v \in \mathcal{U} | \exists \langle u, p_u, t_u \rangle, \langle v, p_v, t_v \rangle \in \mathbb{D}, \\ t_u - t_v \rangle \Delta,\ p_u \in P,\ p_v \in P\} \tag{9}$$

Thus, $P(u|\Theta_i) = \prod_{\forall P \in \mathbb{P}_u} P_+(P|u, \Theta_k) P_-(P|u, \Theta_k)$ where $\mathbb{P}_u = \{P \in \mathbb{P} | \exists \langle u, p, t \rangle \in \mathbb{D}, p \in P\}$ is the set of clusters visited by $u$, $P_+(P|u, \Theta_i) = 1 - \prod_{v \in F_{u,P}^{+}} (1 - p_v^i)$ is the probability that some of the potential influencers actually influenced $u$ in visiting $P$, and $P_-(P|u, \Theta_i) = \prod_{v \in F_{u,P}^{-}} (1 - p_v^i)$ is the probability that users in $F_{u,P}^{-}$ did not influence $u$.

The next task is therefore to perform the $M$ step, where we update the set of variables $p_u^i$ for all $u$ and $i$ to maximize the $\mathcal{Q}$-function (Eq. 5). Since the derivation for the updated $p_u^i$ is identical to the C-IC model [5], we skip the proof and directly present the result and the intuition behind it.

$$p_u^i = \frac{\sum_{\{\langle v, P \rangle | u \in F_{v,P}^{+}\}} \gamma_{v,i} * \eta_{P,u,v,i}}{S_{u,i}^{+} + S_{u,i}^{-}} \tag{10}$$

where $S_{u,i}^{+} = \sum_{\{\langle v, P \rangle | u \in F_{v,P}^{+}\}} \gamma_{v,i}$, $S_{u,i}^{-} = \sum_{\{\langle v, P \rangle | u \in F_{v,P}^{-}\}} \gamma_{v,i}$ and

$$\eta_{P,u,v,i} = \frac{p_u^i}{1 - \prod_{w \in F_{v,P}^{+}} (1 - p_w^i)} \tag{11}$$

Here, $\{\langle v, P \rangle | u \in F_{v,P}^{+}\}$ denotes the set of users who have been potentially influenced by $u$ over some cluster $P$.

$\eta_{P,u,v,i}$ stands for the "responsibility" of user $u$ in triggering $v$'s visit to cluster $P$ in the context of the community $C_i$. Hence, the numerator in Eq. 10 is proportional to the probability of all users who have been influenced by $u$ in community $i$. The denominator factors in the probabilities that the users who have been influenced by $u$ and those whom $u$ failed to influence truly belong to community $i$. Thus, to summarize, $p_u^i$ increases if $u$ is responsible for influencing many people in community $C_i$ to visit a cluster already visited by $u$.

### *B. Completing the GeoScop pipeline*

Revisiting Fig. 2, to complete the pipeline for GeoScop, we need to formulate the initialization module and the clustering module. Alg. 1 lays out the pseudocode for the entire GeoScop algorithm and how these two modules fit in. Initialization is performed once at the start (lines 3-8).

From the formulation of our geo-social clusters, if two points in a geo-social cluster are density connected, then they are also density connected in the geographical world. This property lies at the core of our initialization procedure. For the initial set of clusters, we simply perform DBSCAN without considering the social aspect (line 3 in Alg. 1) and feed these clusters to the community initialization algorithm. Next, we perform a crude community detection based on the geographical clusters. Specifically, for each user $u$, we construct a *cluster profile*, denoted as $cr(u)$. A cluster profile is identical to the idea of community profile. It is a feature vector of dimension size equal to the number of geographical clusters. The $i^{th}$ dimension in a cluster profile contains the number of visits to the $i^{th}$ cluster. The similarity between two users, $usim(u_1, u_2)$ is now quantified using the same MinMax measure.

$$usim(u_1, u_2) = \frac{\sum_{\forall P_i \in \mathbb{P}} \min\{cr(u_1)[i], cr(u_2)[i]\}}{\sum_{\forall P_i \in \mathbb{P}} \max\{cr(u_1)[i], cr(u_2)[i]\}} \tag{12}$$

**Algorithm 1** GeoScop($\mathbb{D}$)

**Ensure:** return the set of geo-social clusters $\mathbb{P}$
1: $\mathcal{U} \leftarrow$ unique users in $\mathbb{D}$
2: $\mathcal{S} \leftarrow$ unique places in $\mathbb{D}$
3: $\mathbb{P} \leftarrow DBSCAN(\mathcal{S})$
4: $\mathbb{C} \leftarrow CommunityInitialization(\mathbb{D}, \mathbb{P})$
5: **for** $\forall C_i \in \mathbb{C}$ **do**
6: **for** $\forall u \in C_i$ **do**
7: $\gamma_{u,i} = 1$
8: $p_u^i = usim(centroid(C_i, u))$
9: **repeat**
10: $\forall C_i \in \mathbb{C}, \pi_i = \frac{\sum_{\forall u \in \mathcal{U}} \gamma_{u,i}}{\|\mathcal{U}\|}$
11: **repeat**
12: **for** $\forall C_i \in \mathbb{C}$ **do**
13: **for** $\forall u \in \mathcal{S}$ **do**
14: $F_{u,i}^{+} \leftarrow$ positive influencers of $u$ in $C_i$
15: $F_{u,i}^{-} \leftarrow$ members of $C_i$ who failed to influence $u$
16: Compute $\gamma_{u,i} \backslash\backslash$ (Eq. 7)
17: **for** $\forall C_i \in \mathbb{C}$ **do**
18: **for** $\forall u \in C_i$ **do**
19: Compute $p_u^i \backslash\backslash$ (Eq. 10)
20: **until** Convergence of parameter set $\Theta$
21: **for** $\forall p \in \mathcal{S}$ **do**
22: **for** $\forall C_i \in \mathcal{C}$ **do**
23: $pr(pl)[i] = \sum_{\forall \langle u, pl, t \rangle \in \mathbb{D},} \gamma_{u,i}$
24: $\mathbb{P} \leftarrow$ geo-social DBSCAN using $pr(pl)$
25: **until** Convergence of $\mathbb{P}$
26: **return** $\mathbb{P}$

Now, if we cluster users based on their cluster profiles, we would have a rough estimate of the underlying communities. Following this intuition, we cluster users using Eq. 12 as the similarity function in a manner analogous to DBSCAN. A random user, $u$, is picked and the user's top-$k$ similar users are found. If the $k$-th user has a similarity greater than a certain threshold $minSim$, then user $u$ and the top-$k$ users are grouped together into a community. The intuition here is that in a community, a user must have at least $k$ users who are similar ($k = 20,\ minSim = 0.4$ in our experiments). If not, $u$ must be an outlier. The top-$k$ users are then pushed into a queue and one by one, each of them is picked for further expansion of the community. This is repeated until the queue gets empty. A new user who has not yet been processed is then picked and the procedure is repeated. Finally, the set of communities is returned.

Once the initial communities are computed, we initialize $p_u^i$ and $\pi_i$. $\pi_i$ is simply $\frac{\|C_i\|}{\|\mathcal{U}\|}$, where $C_i$ is the $i^{th}$ community (line 7, line 10 in Alg. 1). We use a centroid based approach to initialize $p_u^i$. For each community $C$, we calculate the mean cluster profile vector where $cr(centroid)[j] = \sum_{\forall u \in C} \frac{cr[j]}{\|C\|}$. Conceptually, the centroid is assumed to be the most influential a user can get. Thus, $p_u^i$, for user $u$ and community $C_i$, is set to $p_u^i = usim(centroid, u)$ (line 8 in Alg. 1). Owing to the MinMax formulation, $usim$ lies in the range $[0, 1]$. In other words, users close to the centroid are set as influential and those lying on the borders do not exert much influence on the community members.

## IV. Experiments

We use two datasets for benchmarking purposes: *Gowalla*[6] and *Brightkite*[6]. Gowalla contains 196,591 users with 6,442,892 check-ins performed on 1,280,969 places over a period from February 2009 to October 2010. Brightkite contains 58,228 users with 4,491,143 check-ins from April 2008 to October 2010 across 772,783 places. Both these datasets are accompanied with a social network connecting their users. Gowalla contains 950,327 edges and Brightkite has 214,078 edges. Although, GeoScop does not use the network, its availability allows us to evaluate the performance against a technique like DCPGS[1], which relies on the network to perform geo-social clustering.

| City | Code | Source | ‖Users‖ | ‖Places‖ | ‖Check-ins‖ |
|---|---|---|---|---|---|
| Chicago | C | Gowalla | 4697 | 16721 | 95154 |
| Philadelphia | P | Gowalla | 2574 | 11857 | 47538 |
| Boston | B | Gowalla | 3142 | 15977 | 74708 |
| San Francisco | S | Brightkite | 3598 | 14042 | 85758 |
| New York City | N | Brightkite | 4029 | 21448 | 128397 |
| Washington | W | Brightkite | 2668 | 20129 | 117298 |

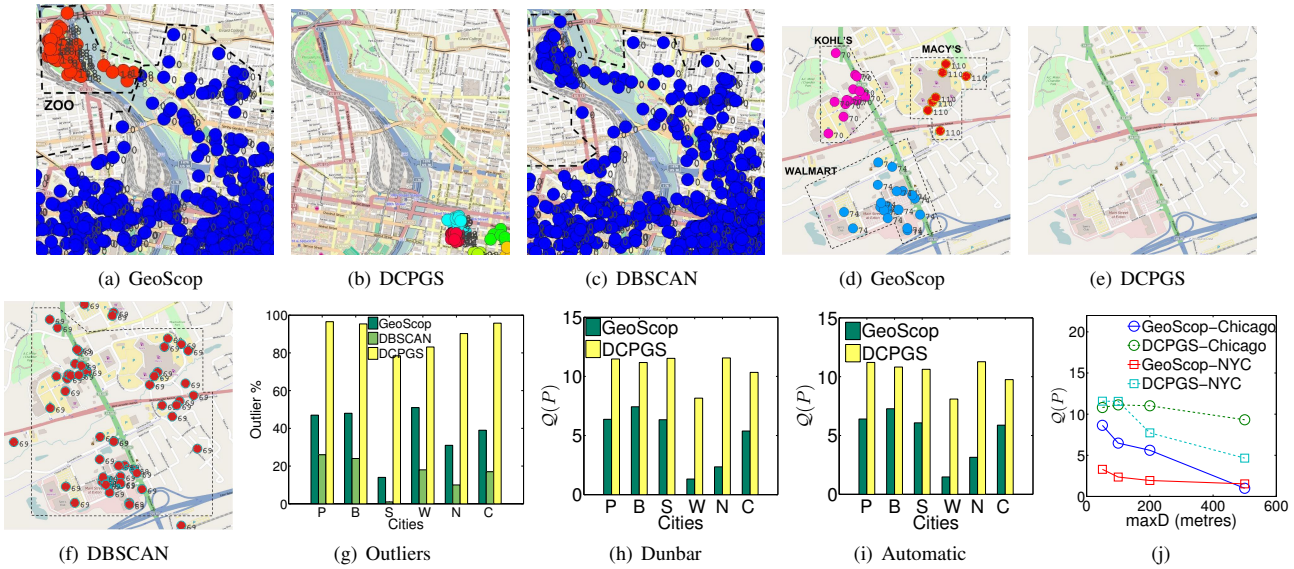
TABLE I
STATISTICS OF VARIOUS CITY CHECK-INS.

We compare GeoScop's performance with DCPGS. In addition to DCPGS, we also include DBSCAN in our benchmarking studies to showcase the need to capture semantics. Since clusters across two different cities are always independent, we perform our analysis on a city-by-city basis. Table I summarizes the check-ins in cities used for our experiments.

**Parameters:** The parameters that are common to GeoScop, DCPGS, and DBSCAN are $minPts$ and the geographical radius $\delta_g$ . We use the default parameter values recommended in DCPGS since the same datasets are used in their evaluation as well. These values are $minPts = 5$ and $\delta_g = 120m$. DCPGS requires three other parameters. DCPGS performs DBSCAN with a weighted combination (weight $w$) of geographical distance and social distance with two additional thresholds on the weighted distance ($\epsilon$) and the social distance ($\tau$). We set these parameters as recommended by the authors in [1], which are $w = 0.5, \epsilon = 0.4, \tau = 0.7$. GeoScop needs to know a social distance threshold $\delta_s$, which we learn automatically from the dataset. More specifically, following the initialization step, we sample 10% of the users and compute the social distance between them. $\delta_s$ is set to 80 percentile, which is the value where 80% of the social distances are larger. In EM, we construct $F^+$ and $F^-$ using the time threshold $\Delta$ set to 30 days.

### A. Visualization-based Analysis

Figs. 3(a) presents an example from Philadelphia where two geo-social clusters are found (color coded in blue and orange). Notice in Fig. 3(c) that DBSCAN merges the two into one single cluster. On closer inspection, we find that while the larger blue cluster generally corresponds to the downtown area, the orange cluster represents the zoo. This is a perfect example demonstrating the need to incorporate social information and capture the semantics. Zoo is frequented by tourists compared to the community of residents who routinely visit the downtown areas. GeoScop is able to make this distinction due to mining communities through the proposed iterative




(a) GeoScop (b) DCPGS (c) DBSCAN (d) GeoScop (e) DCPGS

(f) DBSCAN (g) Outliers (h) Dunbar (i) Automatic (j)

Fig. 3. (a-i) Comparison of the clusters obtained using GeoScop, DCPGS, and DBSCAN. Colored viewing is recommended for these plots. (a-c) GeoScop correctly separates zoo from downtown Philadelphia, which DCPGS and DBSCAN fails to detect. Similar results are obtained in (d-f) where GeoScop separates the check-ins at Walmart and Target into two clusters and in (g-i) where GeoScop is able to detect three communities going to Macy's, Walmart, and Kohl's. (j) The percentage of outliers for the three clustering techniques. (h-i) The social quality of the clusters obtained from GeoScop and DCPGS. (j) The variation in the social quality with the maxD parameter.

framework. Now, notice in Fig. 3(b), that DCPGS fails to detect any geo-social cluster in the zoo. It also classifies most of the check-ins in the downtown area as outliers. This results due to the restrictive 1-hop based social distance measure.

We present another set of results from Philadelphia, where GeoScop picks out Walmart (blue), Macy's (red) and Kohl's (pink) as three different geo-social clusters (Fig. 3(d)) from a single cluster as identified by DBSCAN (Fig. 3(f)). These three are well known stores in USA. Both Macy's and Kohl's sell different kind of products than Walmart. Between Macy's and Kohl's, Macy's caters to the upper income group, whereas Kohl's core customer base is the middle-class section[7], [8]. This separation in the communities of people visiting the same neighborhood is picked by GeoScop and appropriately segregated. DCPGS (Fig. 3(e)) classifies all check-ins in this area as outliers.

One consistent theme that we observe across all these results is that in DCPGS, most places get classified as outliers. Intrigued by this observation, we study deeper and quantify what percentage of the places in each of these cities are outliers. Fig. 3(g) presents the results for all three clustering techniques across 6 major US cities. We use the city codes shown in Table I as $x$-axis labels. It is natural that both GeoScop and DCPGS would have more outliers than DBSCAN since they enforce social constraints in addition to the geographical constraints of DBSCAN. The most striking observation in this study though is the high outlier percentage in DCPGS. This result stems from the low overlap in 1-hop neighborhoods of users visiting places, which translates to high social distance. GeoScop, on the other hand, finds much less outliers due to community level analysis.

### B. Quantitative Analysis

Our goal in the following experiments is to measure the social quality across all clusters in a city. Note that the geographical quality is already bounded by DBSCAN. What is unknown is the following question: *Do geo-social clusters exist that attract an entire community?* We answer this question in a systematic and quantitative manner with respect to the communities that exist in the actual social network rather than those mined by GeoScop. Thus, the following experiments also evaluate to what extent GeoScop is able to overcome the inaccessibility to the social network.

Intuitively, a cluster $P$ is good if a person $u$ visiting $P$ indicates a high likelihood of the remaining members of $u$'s community also visiting $P$. We formalize this property. Let $\mathbb{C}$ be the set of all communities visiting a cluster $P$ and $\mathbb{P}$ be the set of all geo-social clusters. For any community $C \in \mathbb{C}$, $visits(C,P) = \{\langle u,p,t\rangle \in \mathbb{D} | u \in C, p \in P\}$ denotes the set of visits from $C$ to $P$. Therefore, higher the value of $\frac{\|visits(C,P)\|}{\sum_{\forall P' \in \mathbb{P}} visits(C,P')}$, better is the social quality of $P$ with respect to community $C$. A high value indicates that there is a strong affinity of visiting $P$ in community $C$. However, this ratio ignores the fact that a high number of visits from a single member of the community could bias the overall ratio if the other members are not as outgoing. Hence, we compute $people(C,P) = \{u \in \mathbb{C} | \langle u,p,t\rangle \in \mathbb{D}, p \in P\}$, which is the set of members who visited $P$. For a "good" geo-social cluster, we would also like $\frac{\|people(C,P)\|}{\|C\|}$ to be high. We combine these intuitions into the following metric:

$$\mathcal{Q}(P) = -\sum_{\forall C \in \mathbb{C}} log(p(C)) \times \frac{\|C\|}{\sum_{\forall C' \in \mathbb{C}} \|C'\|} \qquad (13)$$

where $p(C) = \frac{\|visits(C,P)\|}{\sum_{\forall P' \in \mathbb{P}} vists(C,P')} \times \frac{\|people(C,P)\|}{\|C\|}$

$p(C)$, which lies in the range $[0,1]$, is proportional to the representation of a community in cluster $P$ and also to the proportion of visits paid by the community to $P$. We take the negative log of $p(C)$ for each community visiting cluster $P$ and weight it based on its size so that a larger community has a higher say. The lower the value of $\mathcal{Q}(P)$, the better is the social quality. Given the entire set of geo-social clusters $\mathbb{P}$, the quality is summarized as $\mathcal{Q}(\mathbb{P}) = \sum_{\forall P \in \mathbb{P}} \mathcal{Q}(P) \frac{\|P\|}{\sum_{\forall P' \in \mathbb{P}} \|P'\|}$.

We compute $\mathcal{Q}(\mathbb{P})$ with respect to the actual communities that exist in the social networks of Gowalla and Brightkite. The communities are mined using *Metis*[9]. Metis requires the number of communities as an input. We use two approaches to determine this parameter. GeoScop proposes an automated way of detecting the number of communities in the initialization step. In the second approach, we use the heuristic proposed by DCPGS. DCPGS uses the *Dunbar number* to set the number of communities, which is simply $\frac{\|\mathcal{U}\|}{150}$.

Figs. 3(h) and 3(i) demonstrate the results in 6 major cities of USA. Regardless of the number of communities provided to Metis, GeoScop performs better. The difference in performance is most evident in San Francisco and New York City, where GeoScop is 6 times better than DCPGS. This result brings out two key conclusions. First, it is indeed possible to perform geo-social clustering even when the network is not available. Second, computing social similarity based on community produces better results than the one-hop neighborhood.

Finally, we next investigate the impact of the parameter $\delta_g$, which is common to both GeoScop and DCPGS. Fig. 3(j) studies how the social quality changes with $\delta_g$ in Chicago and NYC. As can be seen, the social quality improves with increase in $\delta_g$. This is natural since as $\delta_g$ grows, the geographical connectivity constraint gets relaxed leading to far-away, but socially similar places getting grouped into the same geo-social cluster.

## V. Related work

Here, we overview the existing works that overlap with our problem. Using mobility data to analyze places has been used for urban planning[10], marketing[11], traffic congestion modeling[12], trajectory estimation[13], and ranking importance of places[14]. A significant volume of work has been done on the correlation between geographical and social relationships. The algorithm ComeTogether, [15], finds clusters of points-of-interests (POIs) based on the correlation of the sequence of visits by mobile users. In contrast to our work, this algorithm ignores the social relationship between users. Rather, the focus is on connecting places that are visited in same sequence by users. Scellato et al. study this correlation in Geo-Social networks [16] and use it for link prediction[17]. Backstrom et al. use geo-social relations to predict user locations[18]. Along similar lines, Pham et al. [19] predict social connections from geographical data, and Cho et al.[20] explain human movements from social relationships. Although related to our work, none of these techniques solve the problem of geo-social clustering of places.

## VI. Conclusion

In this paper, we showed how user check-ins can lead to useful insights about communities and places. Our problem is motivated from practical real-life scenarios where the social network is inaccessible to majority of the companies that collect check-in data. To solve this problem, we developed a technique called *GeoScop (GEO-Social Clustering Of Places)*, which uses an iterative framework of community detection and clustering. We conducted extensive experiments on check-ins across 7 major cities in USA. The empirical results demonstrated that even in the absence of a social network, GeoScop is up to 6 times better than the state-of-the-art technique. Overall, GeoScop unleashes the capability to capture semantics in place clustering, which till now, was handicapped due to the reliance on an observable social network.

## References


[1] J. Shi, N. Mamoulis, D. Wu, and D. W. Cheung, "Density-based place clustering in geo-social networks," in *SIGMOD*, 2014, pp. 99–110.

[2] "http://techcrunch.com/2013/01/24/my-precious-social-graph/."

[3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *SIGKDD*, vol. 96, no. 34, 1996, pp. 226–231.

[4] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *SIGKDD*, 2003, pp. 137–146.

[5] N. Barbieri, F. Bonchi, and G. Manco, "Influence-based network-oblivious community detection," in *ICDM*, 2013, pp. 955–960.

[6] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *SIGKDD*, 2011.

[7] "http://www.bloomberg.com/news/articles/2015-03-30/kohl-s-backs-big-name-brands-to-boost-shares-faster-than-macy-s."

[8] "http://www.washingtonpost.com/blogs/wonkblog/wp/2013/11/14/how-wal-mart-and-macys-explain-the-economy/."

[9] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, pp. 359–392, 1998.

[10] S. Mitra, S. Ranu, V. Kolar, A. Telang, A. Bhattacharya, R. Kokku, and S. Raghavan, "Trajectory aware macro-cell planning for mobile users." in *INFOCOM*, 2015.

[11] D. Pennerstorfer and C. Weiss, "Spatial clustering and market power: Evidence from the retail gasoline market," *Regional Science and Urban Economics*, vol. 43, no. 4, pp. 661–675, 2013.

[12] A. Diker and E. Nasibov, "Estimation of traffic congestion level via fn-dbscan algorithm by using gps data," in *Problems of Cybernetics and Informatics*, 2012, pp. 1–4.

[13] P. Banerjee, S. Ranu, and S. Raghavan, "Inferring uncertain trajectories from partial observations," in *ICDM*, 2014, pp. 30–39.

[14] V. Kolar, S. Ranu, A. P. Subramainan, Y. Shrinivasan, A. Telang, R. Kokku, and S. Raghavan, "People in motion: Spatio-temporal analytics on call detail records," in *Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on*, 2014, pp. 1–4.

[15] I. Ramalho Brilhante, M. Berlingerio, R. Trasarti, C. Renso, J. A. F. de Macedo, and M. A. Casanova, "Cometogether: Discovering communities of places in mobility data," in *MDM*, 2012, pp. 268–273.

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

[17] S. Scellato, A. Noulas, and C. Mascolo, "Exploiting place features in link prediction on location-based social networks," in *SIGKDD*, 2011.

[18] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: Improving geographical prediction with social and spatial proximity," in *WWW*, 2010, pp. 61–70.

[19] H. Pham, C. Shahabi, and Y. Liu, "Ebm: An entropy-based model to infer social strength from spatiotemporal data," in *SIGMOD*, 2013.

[20] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *SIGKDD*, 2011.