# Density-based Place Clustering in Geo-Social Networks

Jieming Shi, Nikos Mamoulis, Dingming Wu, David W. Cheung, SIGMOD 2014

Presenter: Muhammad Aamir Saleem

# Agenda

- Motivation
- Introduction
- Methodology
- Experiments and Results
- Conclusion
- Questions and Answers

# Motivation

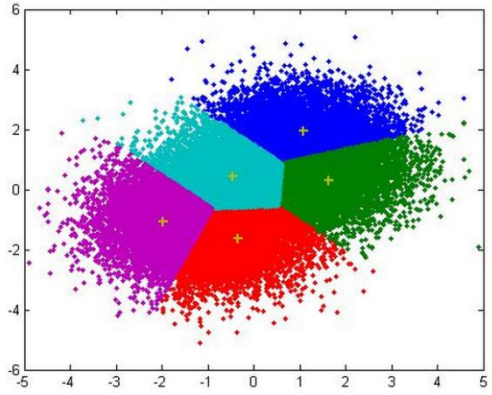
- Geo Social Networks are growing with rapid pace
  - Facebook, FourSquare, Gowalla etc.
- Wide variety of services are being provided
  - Transport management, Recommendation Systems


- These services differs with the diversity of people
  - Advertisement of skydiving
    - Young people close to the area, having strong social interactions

- Identification of these similar groups may significantly improve the services

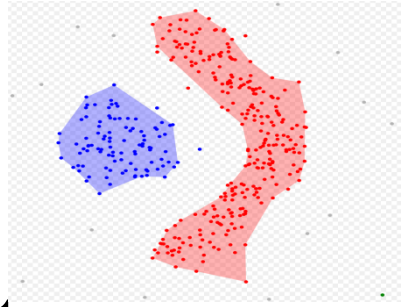
# Introduction
# What is Clustering ?

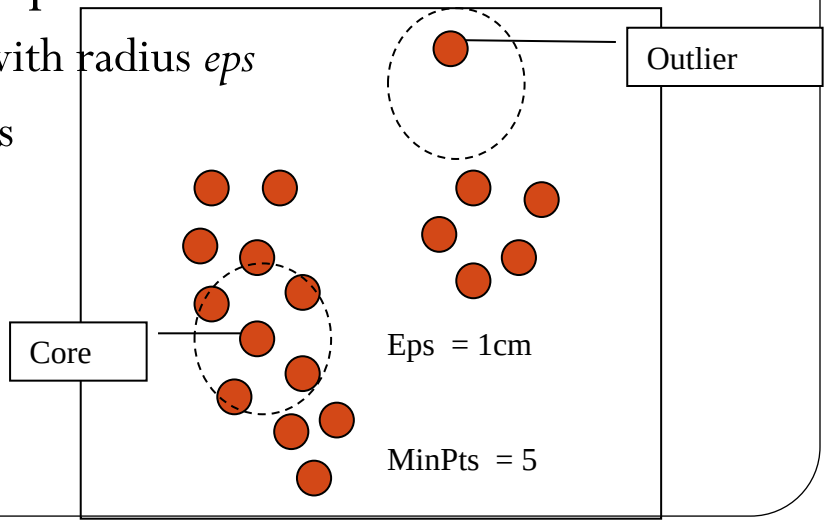
- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Grouping a set of data objects into clusters
- Clustering is unsupervised classification: no predefined classes
- Method for:
  - Data exploration
  - Characterization
  - Summarization



# Density based Clustering

- Divides large collection of points into densely populated regions
  - Most appropriate clustering paradigm for spatial data with low dimensionality
  - Arbitrary shapes and sizes and exclude outliers
- DBSCAN
  - Find spatial neighborhood of each point in dataset
    - Circular region centered at point *p* with radius *eps*
    - Core point if more than *MinPts* places



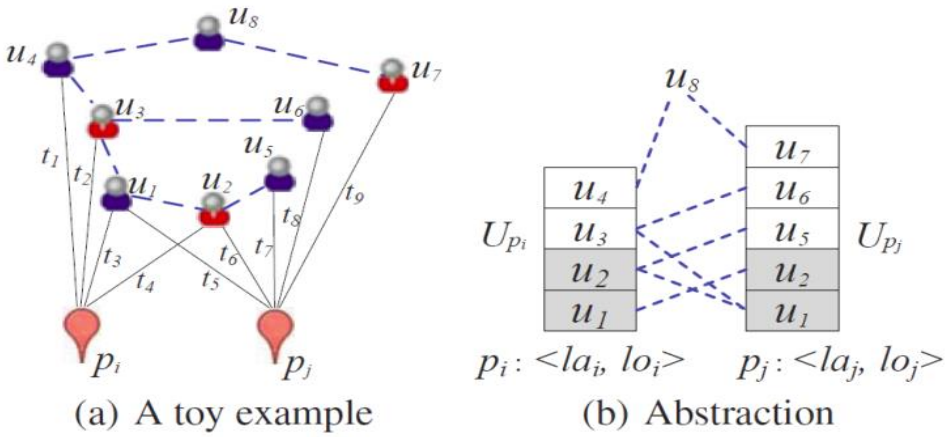


# Potential applications for DCPGS

- Generalization and Characterization of places
  - Land managers interest in identifying region of similar demographic statistics
    - Areas preferred by elderly people to visit
    - Certain religious communities
  - DCPGS fetches areas with social and spatial density
- Data Cleaning
  - Cleaning of semantics for a particular place
    - Big restaurant may have different tags
- Marketing
  - Discount for users visit same places in a geo-social cluster
    - Spatial and social interest

# Density based Clustering Places in Geo-Social Networks (DCPGS)

- Extension of density based clustering (DBSCAN) by considering spatial and social relationships of locations of visitors
  - Replace Euclidian distance threshold *eps* for the extents of dense region by a threshold ϵ
    - Consider both spatial and social relationships between places
    - Spatial distance is Euclidian distance among places
    - Social distance is social relationships between users of these places
- Example:




(a) A toy example (b) Abstraction

# Model and Definitions

- Social Network
  - Un-directed Graph $G=(U,E)$
    - Where $U$ is the set of users
    - $E$ is edges/ friendship links between users

- *Places:*
  - $P$ is the set of all the places visited by users
    - GPS co-ordinates
- Check-ins:
  - $CK$ is the set of all the check-ins generated by users in $U$

# DCPGS Model

- For each place find geo-social $\mathcal{E}$-neighborhood $N(p_i)$
  - $D_{gs}(p_i, p_j) \leq \mathcal{E}$
    - $D_{gs}(p_i, p_j) = f(D_S(p_i, p_j) , E(p_i, p_j) )$
  - $D_S(p_i, p_j) \leq \tau$
    - Social distance
  - $E(p_i, p_j) \leq maxD$
    - Euclidian distance
- If $N(p_i)$ contains places more than *MinPts* then $p_i$ is a core place
  - $P_i$ and all places in neighborhood belong to a cluster $r(p_i)$
  - If another core place belong to cluster $r(p_i)$, their clusters are merged
- DCPGS ends up with set of clusters and outliers

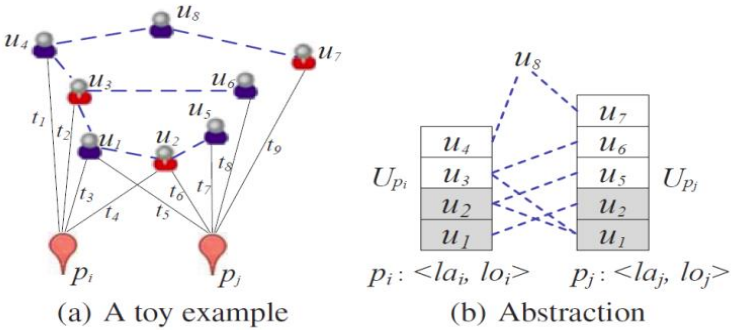
# Social Distance between places

- Social distance $D_s(p_i, p_j)$ between $p_i$ and $p_j$ depends on the social network relationships between users of $U_{pi}$ and $U_{pj}$
  - Contributing Users $CU_{ij}$
    - User $u_a$ is a contributing user if he has visited both places of $p_i$ and $p_j$
    - Users $u_a$ and $u_b$ are contributing users if they both are friends and $u_a$ have visited $p_i$ and $u_b$ have visited $p_j$
  - Social Distance

$$D_S(p_i, p_j) = 1 - \frac{|CU_{ij}|}{|U_{p_i} \cup U_{p_j}|}$$

    - Where, $p_i$ and $p_j$ are locations and $U_{pi}$ and $U_{pj}$ are visitors of these places respectively
    - Set of similarity between sets $U_{pi}$ *and* $U_{pj}$
    - Social relationships among users

# Example




(a) A toy example (b) Abstraction

- $D_s(p_i, p_j)$
  - $U_{pi} = \{u_1, u_2, u_3, u_4\}$
  - $U_{pj} = \{u_1, u_2, u_5, u_6, u_7\}$
  - $CU_{ij} = \{u_1, u_2, u_3, u_5, u_6\}$
    - $u_1$ and $u_2$ visited $p_i$ and $p_j$
    - $u_3$ who visited $p_i$ has friend $u_6$ who visited $p_j$
    - $u_6$ who visited $p_j$ has friend $u_3$ who visited $p_i$
    - $u_5$ has a friend $u_2$ having been to $p_i$
  - $D_s(p_i, p_j) = 1 - 5/7 = 0.2857$

# Alternatives to Ds

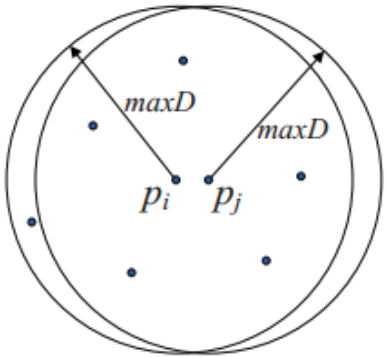
- Jaccard
  - $J(p_i, p_j) = (|U_{pi} \cap U_{pj}|)/(|U_{pi} \cup U_{pj}|)$
  - $D_s^{Jac} = (p_i, p_j) = 1 - J(p_i, p_j)$
  - Disregard the social network friends
- SimRank
  - Structural context model for measuring the similarity between nodes in graph
  - Two nodes are equivalent if they relate to equivalent nodes
  - $D_S^{sim}(p_i, p_j) = 1 - s(p_i, p_j)$
- Katz
  - Measure sums over all possible path form user $u_r$ to $u_s$
  - $D_S^{Katz}(p_i, p_j) = 1 - \frac{1}{|U_{p_i}||U_{p_j}|} \sum_{u_r \in U_{p_i}} \sum_{u_s \in U_{p_j}} \mathcal{K}_a(u_r, u_s)$
- Commute Time
  - Hitting time $h(u_r, u_s)$ from $u_r$ to $u_s$ in the expected number of steps required to reach from a random walk starting at $u_r$ to reach $u_s$ $ct^L(u_r,u_s) = h(u_r,u_s)+(u_s+u_r)$
    - Sensitive to long paths
    - $D_S^{ct}(p_i, p_j) = \frac{1}{|U_{p_i}||U_{p_j}|} \sum_{u_r \in U_{p_i}} \sum_{u_s \in U_{p_j}} ct^L(u_r, u_s)$

# *DCPGS-R* : R-tree based Algorithm

- Places in *GeoSN* are bulk loaded into an R-tree
- Given a place $p_i$, performs range query with radius *maxD* to get a set of candidate places that may fall in the geo-social ϵ-neighborhood i.e. $N(p_i)$
  - *maxD* is maximum allowed spatial distance between places pi and places in its geo-social ϵ-neighborhood
- DSPGS-R filters and keep in $N_\epsilon(p_i)$ only candidates which satisfies social distance τ and geo-social distance threshold ϵ

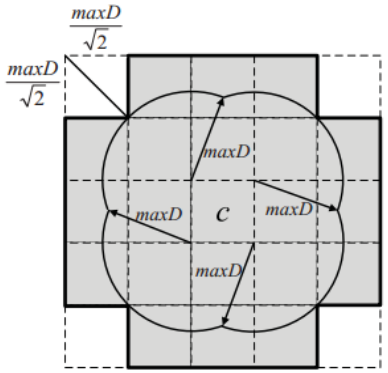
# *DCPGS-G:* Grid based Algorithm

- **Need**
  - *DCPGS-R* conducts spatial query for each place in *GeoSN* to obtain the candidate places for the purposes of discovering geo-social clusters
    - Millions of range queries required for each location
  - Figure shows two almost identical range queries with *maxD*
  - *DCPGS-R* will run 8 independent range queries on R-tree that search almost the same space
    - Redundant traversing paths and computations
  - To overcome this *DCPGS-G* has been proposed



# *DCPGS-G:* Grid based Algorithm

- **Grid Partitioning**
  - The area of locations is partitioned by grids of size: $maxD/\sqrt{2} * maxD/\sqrt{2}$
  - Non-empty grids are indexed by hash table with grid cell coordinates as search key
- **Neighbor Cells**
  - Cells that intersect the union of four cells, each centered at a corner of cell with radius *maxD*
  - *Example:* 20 cells except c are neighbors cells
  - Any place *p* inside *c*, the content of *p's* geo-social neighborhood is contained in *NC(c)* and *c* itself



# *DCPGS-G:* Grid based Algorithm

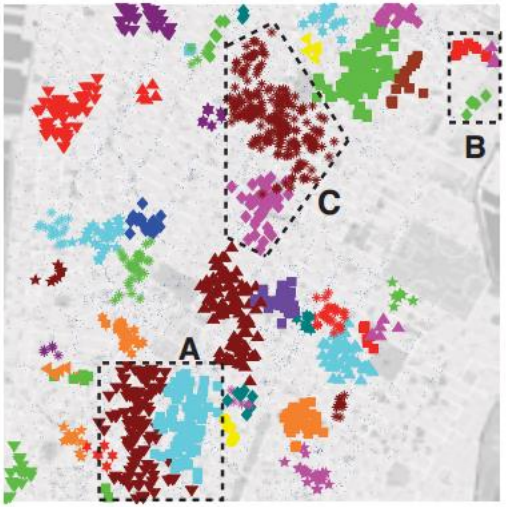
- **Cluster Discovery**
  - Algorithm maps all places into grid cells
  - Obtains geo-social ∈-neighborhood of all places
    - For each cell retrieve its neighbor cell
    - Filters out the pairs of places with spatial distance greater that *maxD*
      - $p_i \in c$, $p_j \in NC(c)$ and $p_i \neq p_j$
      - Step is not needed if $p_i$ and $p_j$ are in same grid cell
      - Pairs of places that satisfies the social and geo-social distance are selected
      - If $p_i$ and $p_j$ are in each other's geo-social neighborhood, their corresponding $N_\epsilon(p)$ are updated
  - Discovers all geo-social clusters in *GeoSN*

# Qualitative Analysis

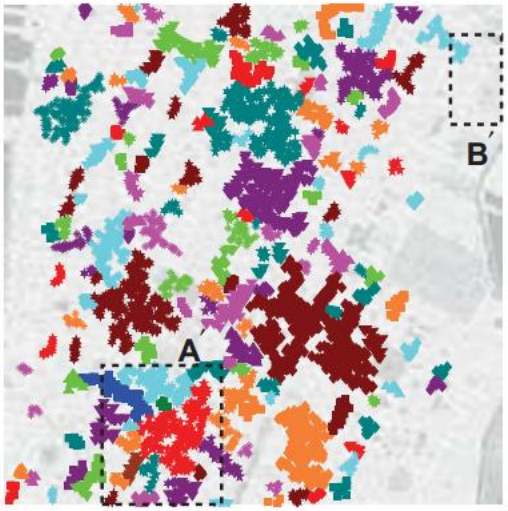
- **Data**
  - Gowalla:
    - Users $|U|=196,59$
    - Friendship edges $|E|=950,327$
    - Checkins $|CK|=6,44,892$
    - Places $|P|=1,280,969$ from Feb, 2009 to Oct. 2010
  - BrightKite:
    - Users $|U|=58,228$
    - Friendship edges $|E|=214,078$
    - Checkins $|CK|=4,491,143$
    - Places $|P|=772,783$ from Apr. 2008 to Oct. 2010

# Visualization based Analysis

- Geo-Social Splitting/Merging Criteria
  - Geo-social clusters that are very closed to each other are split correctly by *DCPGS*
    - *DBSCAN* may consider them as single cluster due to their spatial closeness
    - Clusters split by *DBSCAN* due to relatively low spatial density between them are merged by *DCPGS* because of their strong social ties



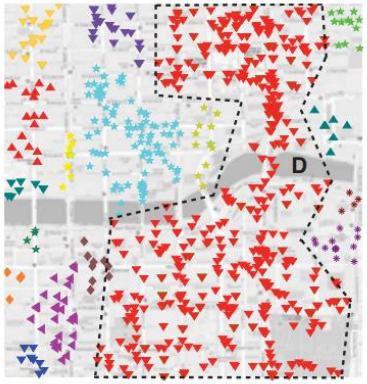

(a) DCPGS: $\epsilon = 0.4$, $\tau = 0.7$, $maxD = 100m$



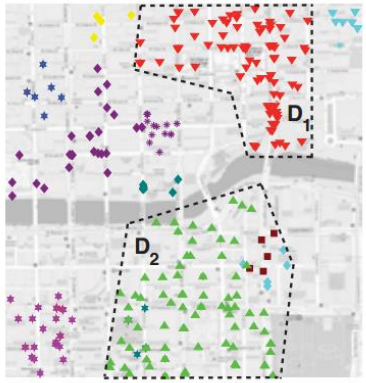

(b) DBSCAN: $eps = 40m$

# Visualization based Analysis

- Spatially dense clusters may be split by *DCPGS* because of some natural barriers such as rivers and walls
  - Region *D* in following figure



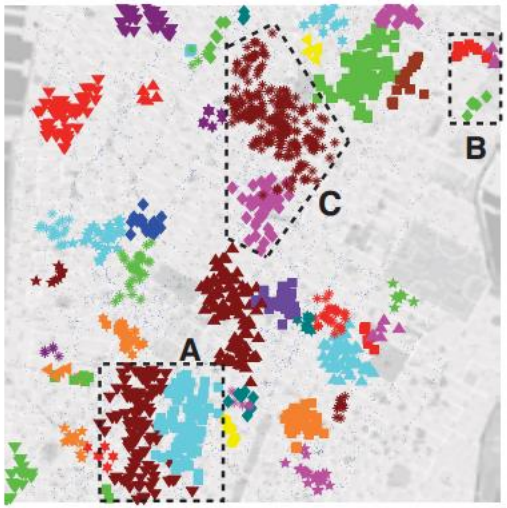

(a) DBSCAN: $eps = 60m$



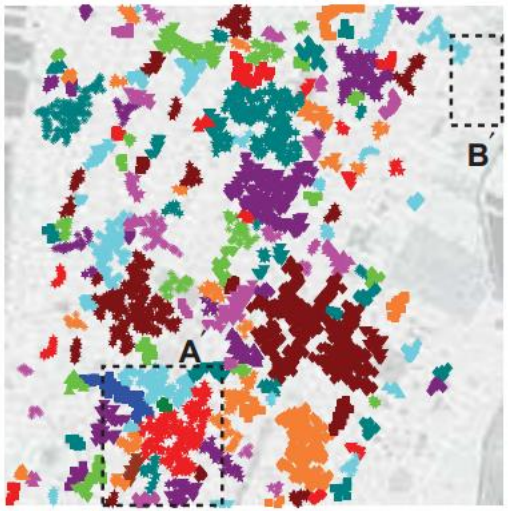

(b) DCPGS: $\epsilon = 0.4$ s.t. $\tau = 0.7$, $maxD = 120m$

# Visualization based Analysis

- **Spatially Loose Clusters**
  - Clusters detected by *DCPGS* are considered as outlier by *DBSCAN* i.e. B and B' in following figure
    - Users who checked in those places have strong social relationship
    - By decreasing density parameters *DBSCAN* can discover these clusters but this would merge too many cluster



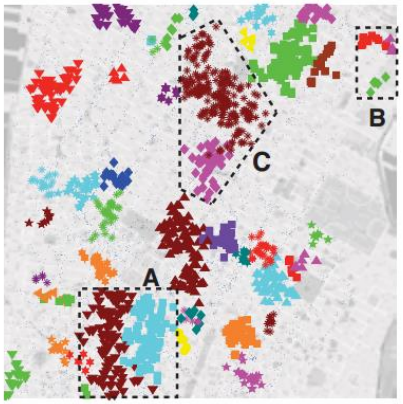

(a) DCPGS: $\epsilon = 0.4$, $\tau = 0.7$, $maxD = 100m$



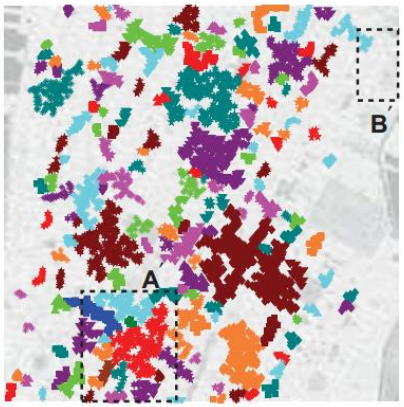

(b) DBSCAN: $eps = 40m$

# Visualization based Analysis

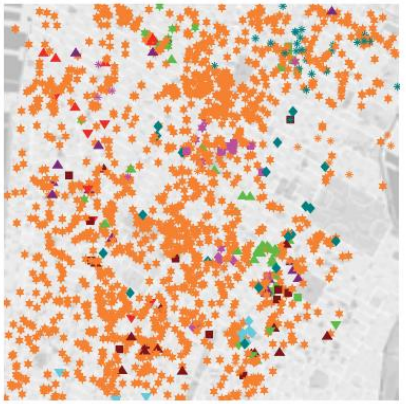
- **Fuzzy Boundary Clusters**
  - *DCPGS* discover clusters having fuzzy boundaries with each other e.g. clusters *C* and *C'* in following figure
    - Groups of socially connected users may spatially overlap
    - DBSCAN produces clusters with strict boundaries
    - Pure social clusters also produce fuzzy clusters but spatially indistinguishable thus not interesting Figure *c*




(a) DCPGS: $\epsilon = 0.4$, $\tau = 0.7$, $maxD = 100m$




(b) DBSCAN: $eps = 40m$

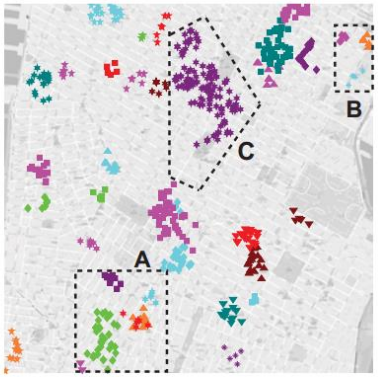


(c) PureSocialDistance: $\epsilon = 0.2$, $\tau = 1$, $maxD = 1000m$

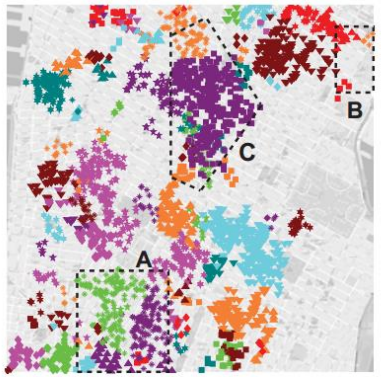
# Visualization based Analysis

- *LinkClustering* produces thousands of small clusters not spatially well separated
  - Due to sparsity of geo-social network data, constructed place network contains many connected components that are disconnected with each other
  - *Jaccard* discovers extremely small clusters and many outliers
  - *SimRank* is skewed towards low values

(d) LinkClustering: $\tau = 0.7$, $maxD = 100m$



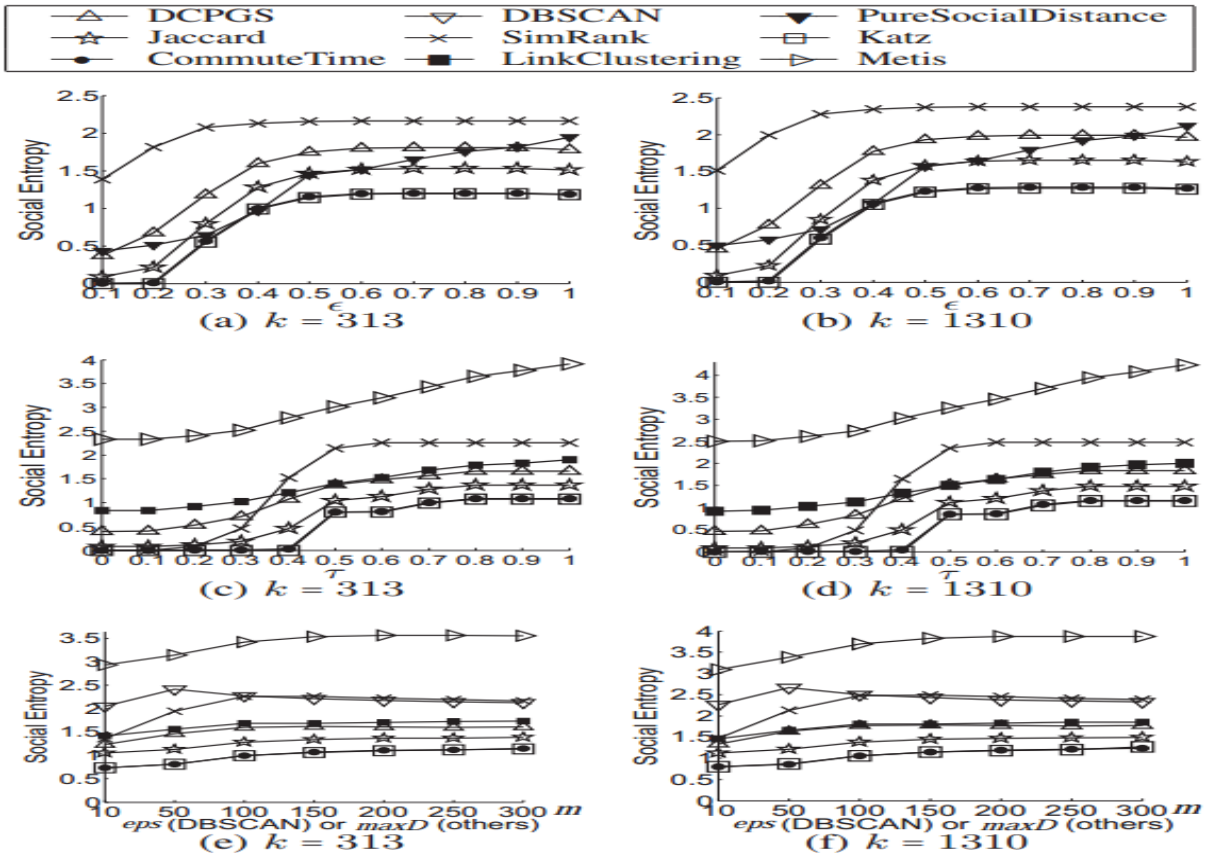

(e) Jaccard: $\epsilon = 0.4$, $\tau = 0.7$, $maxD = 100m$




(f) SimRank: $\epsilon = 0.3$, $\tau = 0.7$, $maxD = 100m$

# Social Quality Evaluation

- Social Entropy based Evaluation
  - It measure the social quality of clusters based on network communities that the GeoSN



# Related Work

- Spatial Clustering
  - Partitioning
    - K-mean, k-mediods and CLARANS need predefined k to specify number of clusters
  - Hierarchical
    - Don't have well defined termination criteria and cannot correct the result if some objects assigned to wrong clusters at an early stage
  - Density based Clustering
    - Discovers clusters of arbitrary shapes
    - DBSCAN, GDBSCAN,DENCLUE, OPTICS
    - This is adopted and extended in this work

# Related Work

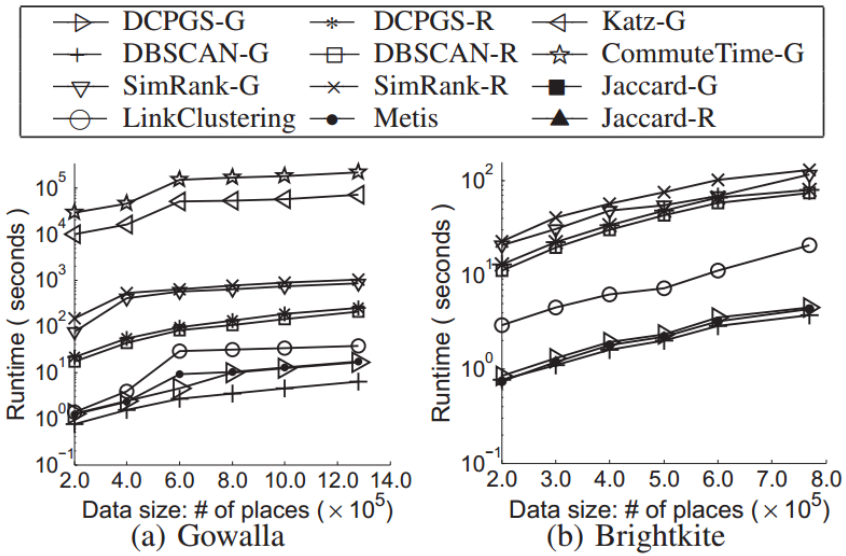
- Analysis of places based on Mobility Data
- Clustering based on Spatial and Non-Spatial Attributes
  - [39] cluster pixels consider RGB colors and spatial proximity that is useful in natural image segmentation
  - [32] use spectral clustering to identify communities in graph
- Spatial-Social Relationship
  - [27] performed a quantities on socio-spatial properties of GeoSN and proposed a link prediction model
  - [37] designed a location recommendation system
  - [5] predict the location of an individual from a sparse set of known user location using the relationship between geography and friendship

# Related Work

- Detecting and Evaluating Communities in Network
  - SCAN[35] detects clusters hubs and outliers in networks
  - [38] proposed partitioning hierarchinal and density based algorithms to cluster objects on spatial networks based on shortest path distace
  - [18] summarized and empirically evaluated algorithms for network community detection

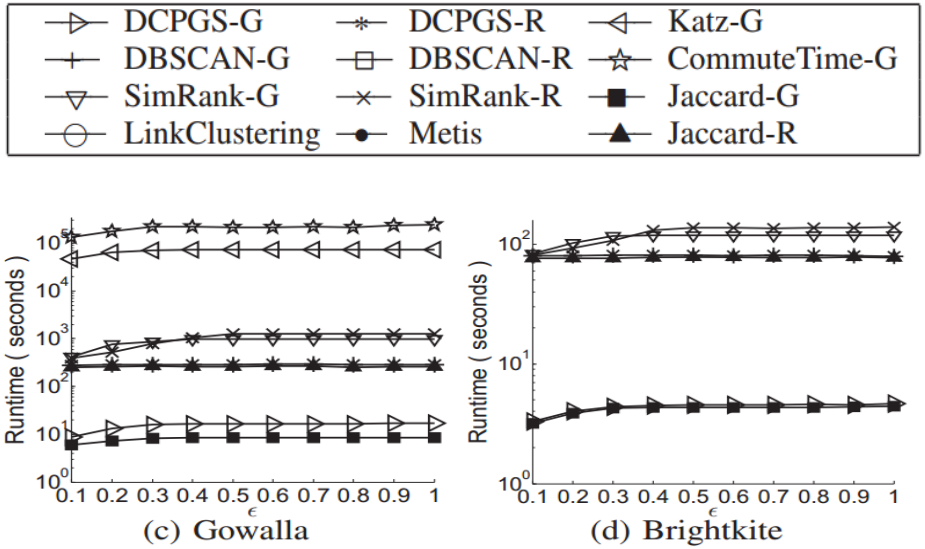
# Efficiency Evaluation

- **Effect of data size**
  - *DCPGS-G* is much faster than *DCPGS-R* and maintain its performance gap as the data grows
  - *DBSCAN-G* is fastest in all cases



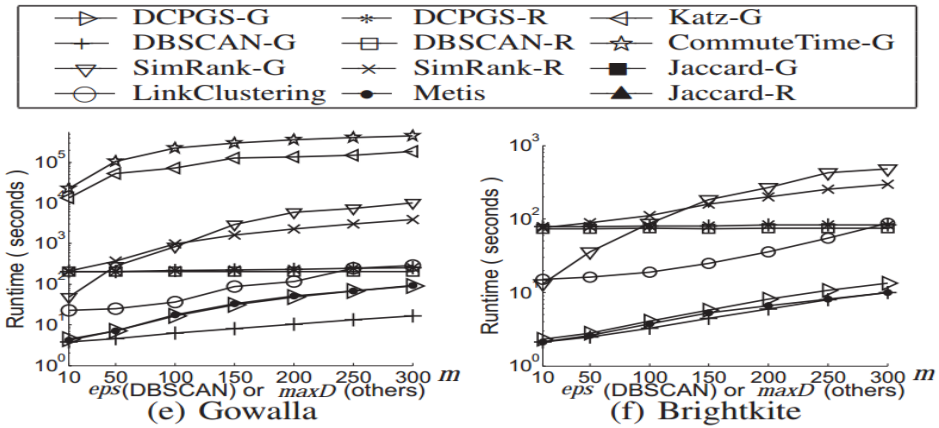
# Efficiency Evaluation

- **Effect of ϵ**
  - *DCPGS-G* is faster than *DCPGS-R* in all cases
  - *DCPGS-G* and *DCPGS-R* are more expensive than *Jaccard-G* and *Jaccard-R*
  - *DCPGS* and *DBSCAN* remains quite stable since computational cost of social distance is insignificant



# Efficiency Evaluation

- **Effect of *maxD* (*eps* in *DBSCAN*)**
  - *maxD* decides the spatial ranges of geo-social neighborhoods of places
  - *SimRank* and *Katz* and *CommuteTime* are more sensitive to *maxD* due to the high cost of their functions
  - Performance gap of *DCPGS-G* and *DCPGS-R* narrows with *maxD*
    - Grid based computation become more expensive as the grid size increases



# Conclusion

- First time the problem of Density based Clustering Places in Geo-Social Networks is discussed
- New measure for the social distance between places considering social ties between users who visited them
  - More effective and efficient
- DCPGS discovers clusters with interesting properties i.e. barrier-based, splitting , spatially loose clusters, cluster with fuzzy boundaries which cannot be found by spatial clustering
- Two evaluation metrics have been proposed, social entropy and community detection based score
- Grid based approach is proposed for further efficient computation

- Future Work
  - Incorporate influence of time span of users visit to places and multiplicity of user visits to a place to cluster places