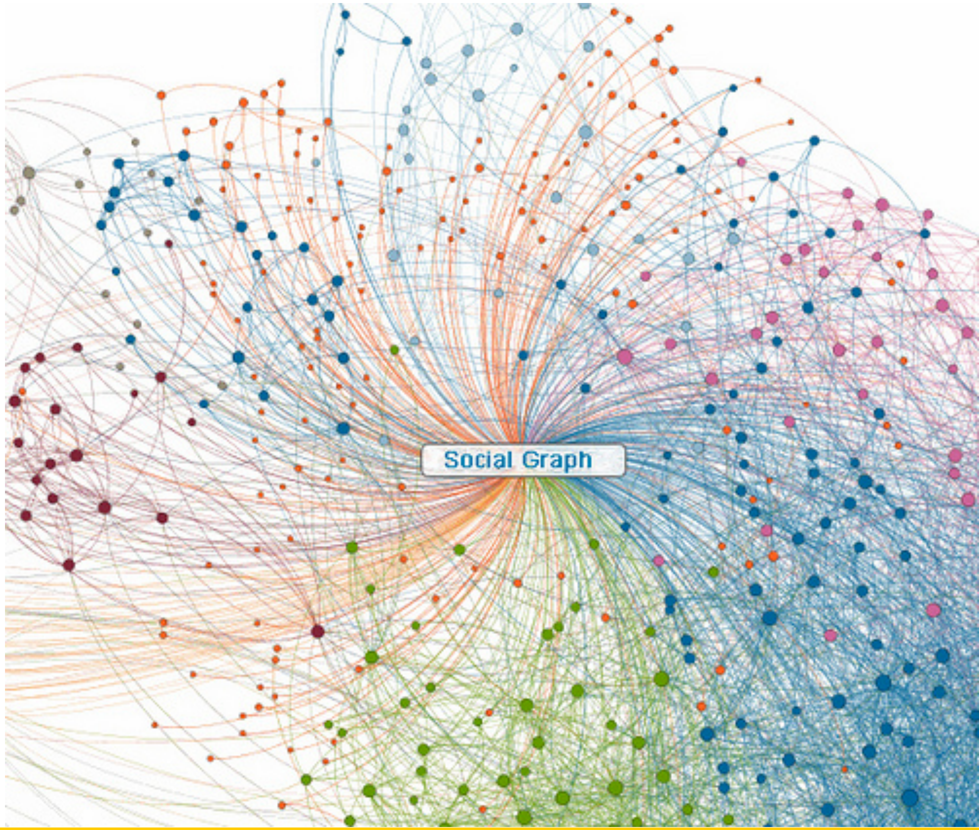
GeoSocial Networks



Cyrus Shahabi, Ph.D.
Professor of Computer Science, Electrical Engineering & Spatial Sciences
Chair, Department of Computer Science
Director, Integrated Media Systems Center (IMSC)
Viterbi School of Engineering
University of Southern California
Los Angeles, CA 900890781
shahabi@usc.edu

OUTLINE

- GeoSocial Queries [VLDB’13]
- Inferring Social from Geo [SIGMOD’13]
- GeoSocial Recommendation [SIGMOD’15]
- Future [SIGSPATIAL’15]

OUTLINE

- GeoSocial Queries [VLDB’13]
- Inferring Social from Geo [SIGMOD’13]
- GeoSocial Recommendation [SIGMOD’15]
- Future [SIGSPATIAL’15]

Geo-Social Networks (GeoSNs)

Geo-Social Query

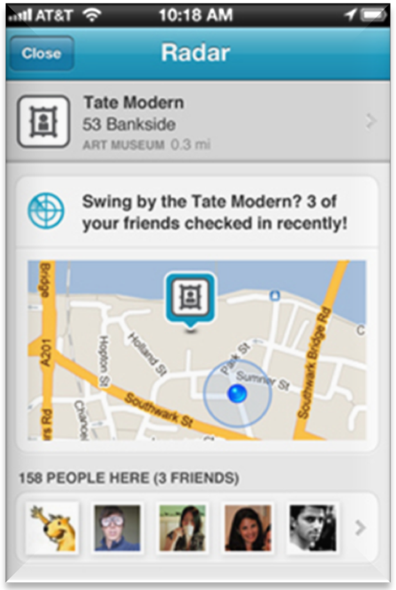
My Friends in range

Social network functionality

Location-based services

+

=



foursquare Radar

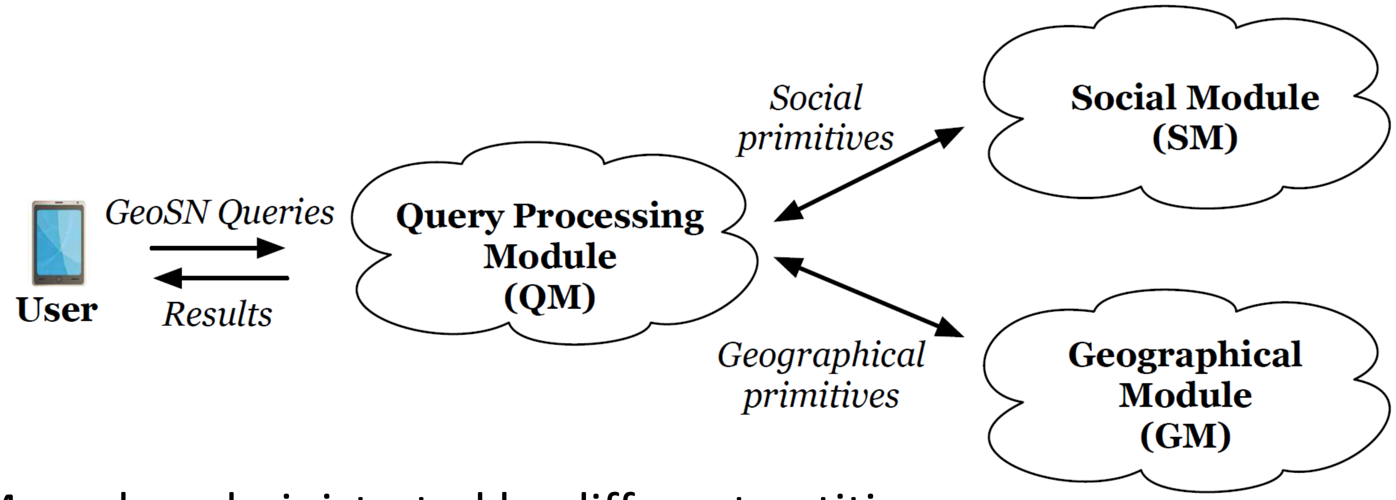
Industry & Academia

Data Management

Application/Paper	Storage Scheme
Social	
+	Adjacency lists in a Distributed Memory Hash Table
+	Adjacency lists in a Document-oriented database
[Y. Doytsher et al., WWW 2012]	Adjacency lists in Neo4j
[W. Liu et al., DASFAA 2012]	Adjacency matrix
[Y. Doytsher et al., LSBN 2010]	Edge lists in a RDBMS
Spatial	
	R*-Tree
	Grids & Geohashes
[J. Bao et al., ICDE 2012]	Grid
[A. Amir et al., PMC 2007]	Quad-Tree
[W. Liu et al., DASFAA 2012]	R*-Tree

Framework

Architecture



- SM and GM can be administrated by different entities.
  - Implement GeoSN queries without owning geo-social data.
- Independent functionality of social and geographical structures.
- Easy integration of new, more efficient data structures without modifications.
- Novel GeoSN query types = either a different combination of existing primitives or new ones

Framework

primitive Operations

Social Primitives

GetFriends(u)

AreFriends(u_i, u_j)

GetDegree(u)

Geographical Primitives

GetUserLocation(u)

RangeUsers(q, r)

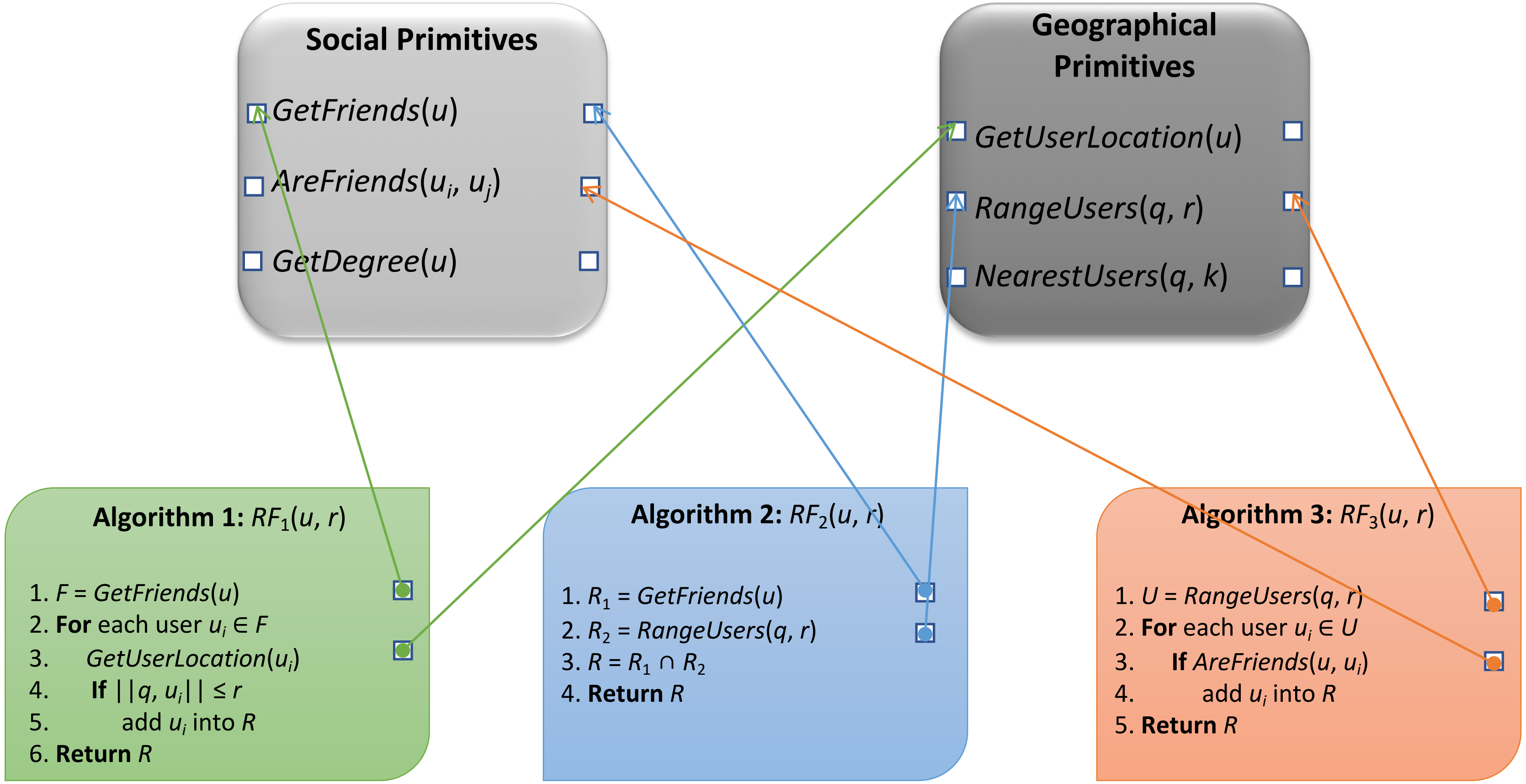
NearestUsers(q, k)

- Any primitive must be treated as an *atomic* operation.
  - No *states.*
  - *NextNearestUser* = multiple calls of *NearestUsers* – keep data locally.
  - Find more!
- Efficiency depends on the underlying storage scheme.
  - *AreFriends* - Adjacency matrix
  - *GetFriends* - Adjacency Lists
  - *GetUserLocation* – Hash Table
  - *RangeUsers* & *NearestUsers* – Spatial Indices
- They are supported by commercial GeoSNs' APIs.

Query processing

range friends

Friends of user u within range r of q



No Spatial Index
Adjacency list
Independent of check-ins
Dense social network

Spatial Index
Adjacency list
Sparse check-ins
primitives

Spatial Index
Adjacency matrix
Sparse check-ins
primitives

Query processing

NEAREST FRIENDS

k nearest friends of user *u* to location *q*.

Social Primitives

- $GetFriends(u)$
- $AreFriends(u_i, u_j)$
- $GetDegree(u)$

Geographical Primitives

- $GetUserLocation(u)$
- $RangeUsers(q, r)$
- $NearestUsers(q, k)$

Algorithm 1: $NF_1(u, q, k)$

1. F = $GetFriends(u)$
2. **For** each user $u_i \in F$
3. $\quad GetUserLocation(u_i)$
4. Sort F (asc. $||q, u_i||$)
5. R = top-k of F
6. **Return** R

No Spatial Index
Adjacency list
Independent of check-ins
Dense social network

Algorithm 2: $NF_2(u, q, k)$

1. F = $GetFriends(u)$
2. **While** $|R| < k$
3. $\quad u_i = NextNearestUser(q)$
4. **If** $u_i \in F$, add u_i into R
5. **Return** R

Spatial Index
Adjacency list
Sparse check-ins
primitives

Algorithm 3: $NF_3(u, q, k)$

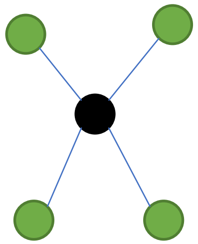
1. **While** $|R| < k$
2. $\quad u_i = NextNearestUser(q)$
3. **If** $AreFriends(u, u_i)$
4. $\quad$ add u_i into R
5. **Return** R

Spatial Index
Adjacency matrix
Sparse check-ins
primitives

nearest star group

(NSG Query)

"the next group of five people who come to the restaurant will receive 20% discount"

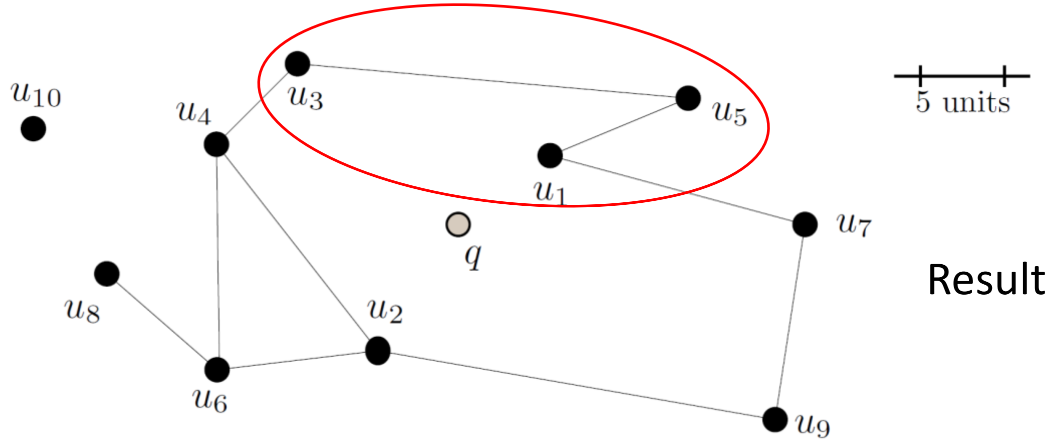


Ideally:
- Socially connected! → Have a common friend (**star**).
- Close to the restaurant → **Min**. sum of **distances** to the restaurant

Output: k nearest groups of m users to q, such that the users in every group are connected through a common friend (star).

Nearest star group

Example (k = 1, m = 3)



Result: { $\underline{u_5}$, u_1, u_3 }

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}
$\|\|q, u_i\|\|$	5	6.4	10	10.4	11.3	12.2	15	15.5	15.9	18.9

<u>Observation:</u>

The **best** group of a user contains himself and his m – 1 closest friends to q.

<u>NSG is not an NP-Hard problem!</u>

NSG query processing

<u>Basic Notation</u>

b_s : the current lower aggregate distance achieved by the already examined users (***seen***).

b_{un}: the lower aggregate distance that can be achieved by non-retrieved users (***un**seen*).

<u>Skeleton for NSG algorithms</u> (Branch and Bound - BnB)

Input: Location q, positive integers m, k

Output: Result set R

1. Initialize R, b_s, b_{un}
2. **While** $b_{un} < b_s$
3. Get the next nearest user to q
4. Construct his best group
5. Update result R and b_s, b_{un}
6. Refine R
7. **Return** R

Eager	**Lazy**	**Eager***
Simple b_{un}	Simple b_{un}	Aggressive b_{un}
☑	☑	☑
☑	☑	☑
Find the group	Construct the graph	Find the group
☑	☑	☑
		☑
☑	☑	☑

Ex

Experiments

Linux, C++

- Storage Schemes
  - Disk-based + Cache
    - Social:
      - **Adjacency List**: user → sorted list of friends' ids. (**document** per user)
    - Geographical:
      - user → coordinates (**document** per user)
      - Index: **Geohashes & Grids**
    - Cache: **Linux's caching** mechanism
  - Memory-based
    - Social:
      - (**Hash Table**) Adjacency List: user → sorted list of friends' ids.
    - Geographical:
      - (**Hash Table**) user → coordinates
      - Index: **Grid** (CPM)

- Machine Architecture
  - Centralized: All modules at a single server.
  - Distributed: Separate server for each module (100 Mbps Ethernet)

Experiments

- Real Dataset (Foursquare & Twitter)
  - Check-ins:
    - 12,652 users
    - *same* day (May 30th, 2012)
    - in New York City (1,112 km^2).
  - Social Graph:
    - 12,652 + 2M (non checked-in friends) users
    - Avg. # of friends: 437.

- Synthetic Dataset (1M, 2M, 3M, 4M, 5M)
  - Check-ins

    [Cho et al., *SIGKDD* '11]
    - "The distribution of the distance between two friends follows a power law."
    - BFS – assign locations: distance is randomly derived by the distribution in:
    - Area: 7,853 km^2
  - Social Graph: Barabási-Albert preference model
    - Power-law degree distribution.
    - Small-world phenomenon.
    - Avg. # of friends: 100.

Experiments

Friends of user *u* within range *r*.

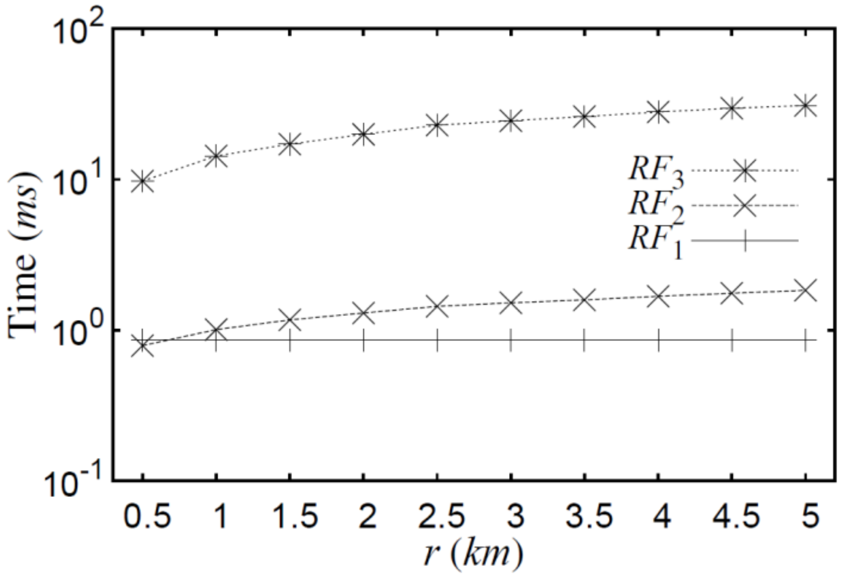
Algorithm 1: $RF_1(u, r)$
One *GetFriends*(*u*)
Multiple *GetUserLocation*(u_i)

Algorithm 2: $RF_2(u, r)$
GetFriends(*u*) ∩
RangeUsers(*q*, *r*)

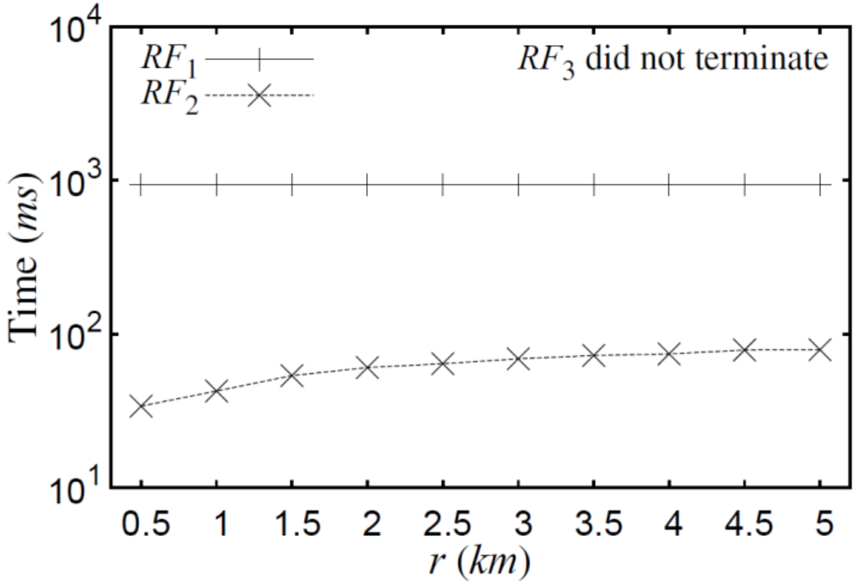
Algorithm 3: $RF_3(u, r)$
One *RangeUsers*(*q*, *r*)
Multiple *AreFriends*(*u*, u_i)

(Average over 100 random queries)

Real Dataset



Memory - Centralized



Memory - Distributed

Experiments

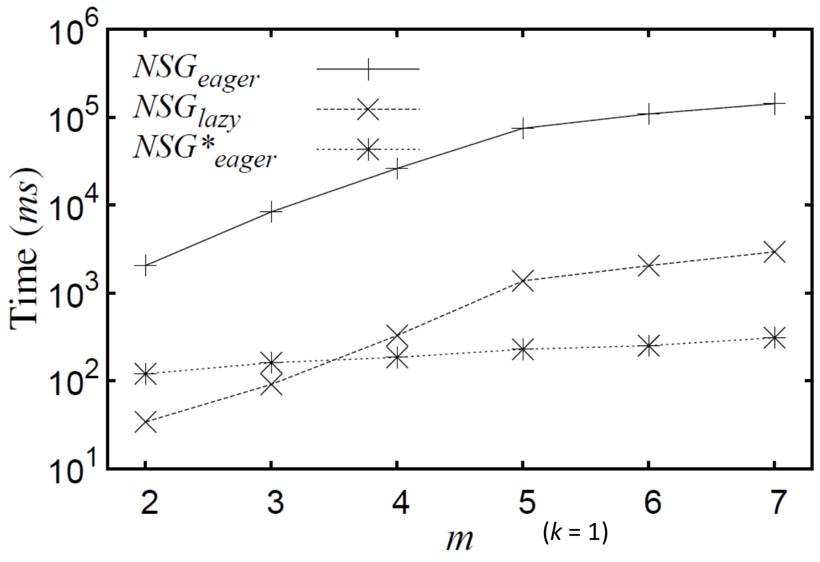
NEAREST STAR Group (NSG)

***NSG*$_{eager}$**
For each newly retrieved user compute his best group eagerly.

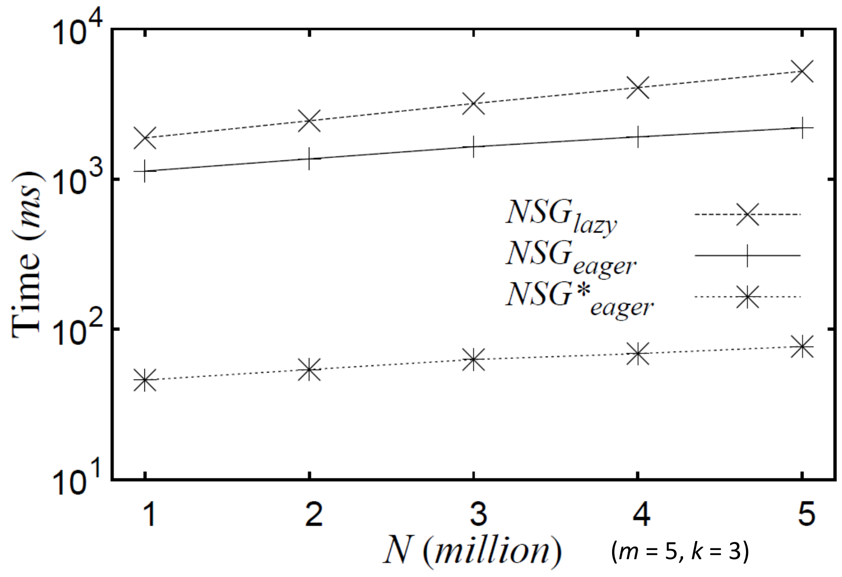
***NSG*$_{lazy}$**
Construct the social graph around *q* iteratively.

***NSG**$_{eager}$**
Similar to *NSG*$_{eager}$, but more aggressive bounds.
Refinement step.

(Average over 100 random queries + warm up)



Real Dataset : Disk - Centralized



Synthetic Dataset : Memory - Centralized

- In the most of the cases *NSG**$_{eager}$ is the best.
- Performance scales well with the dataset size.

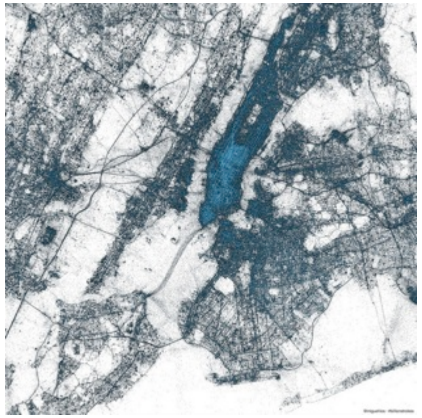
OUTLINE

- GeoSocial Queries [VLDB’13]
- Inferring Social from Geo [SIGMOD’13]
- GeoSocial Recommendation [SIGMOD’15]
- Future [SIGSPATIAL’15]

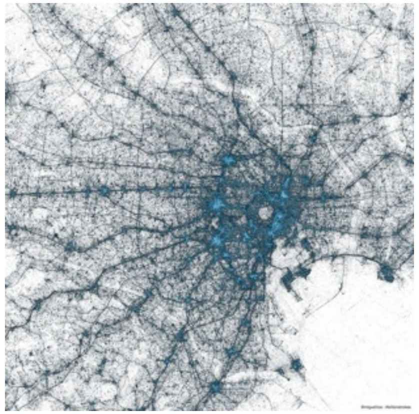
Location-Enriched Datasets

- Popularity of Location-Based Services

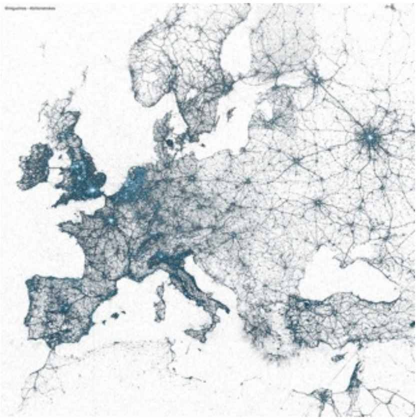
Twitter: 10M+ geo-tagged tweets/day mashable.com
Foursquare: 5M check-ins/day venturebeat.com/2015/08/09/



New York City



Tokyo

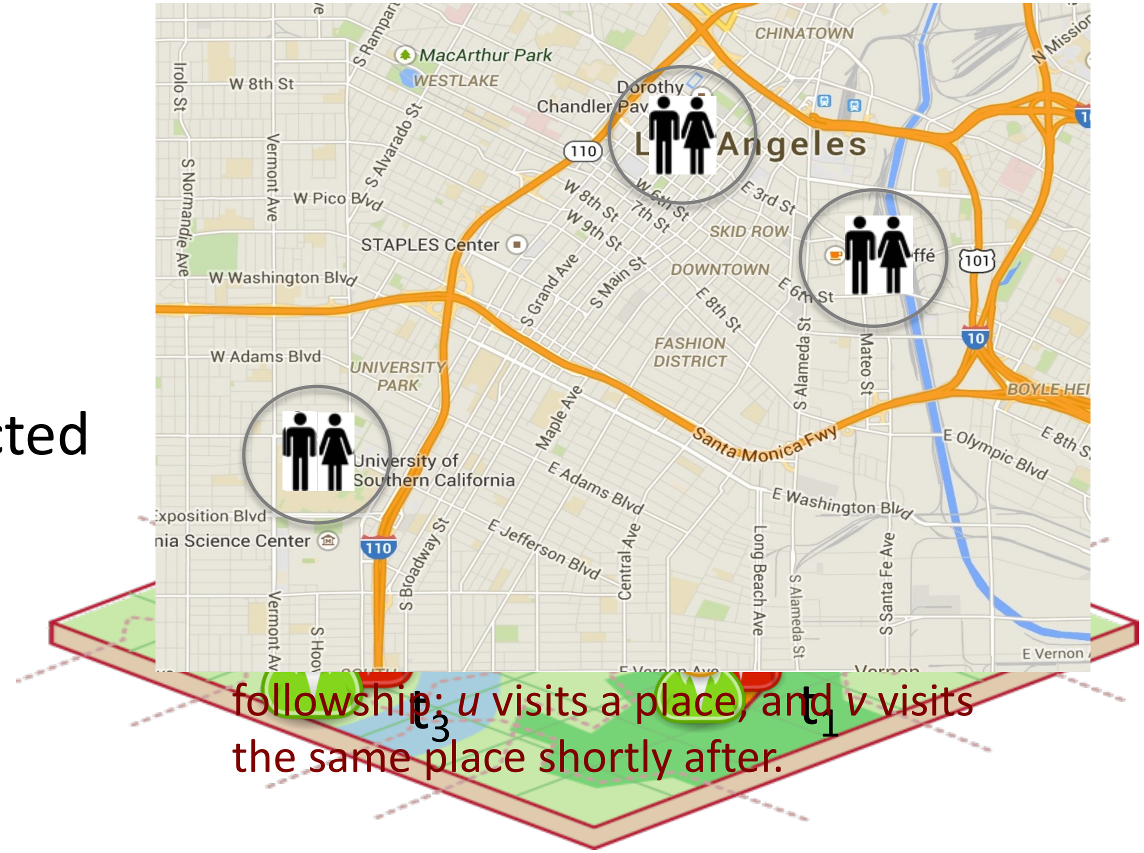


Europe

Geo-Tagged Tweets on Map
by Twitter mashable.com

Social Relationship Inference from Location Data

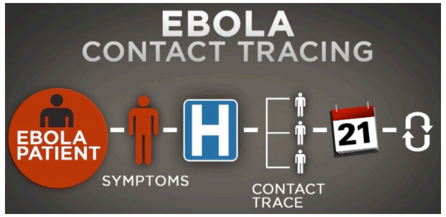
- Reachability [VLDB'12]
  - *u* is reachable to *v* in time period *T*
  - if there is a contact path
- Social Strength [SIGMOD'13]
  - *u* and *v* are socially connected
  - how often they meet and where
- Spatial Influence [ICDE'16]
  - *u* influences *v*
  - if *v* follows *u*



Applications

- Social Network
  - Marketing
  - Friendship suggestions
  - Social and cultural studies

- Geo-social Network
  - Criminology
    - identify the new or unknown members of a criminal gang or a terrorist cell
  - Epidemiology
    - spread of diseases through human contacts
  - Policy
    - induce local influence in electing a tribal representative



Real-World Social Strength - Intuition

Co-occurrence

From Real-World ***Co-occurrences*** to ***Social Strength***

Inferring friendship network structure by using mobile phone data (PNAS'09)

N. Eagle, A. Pentland, D. Lazer

- Study traces of 94 subjects using mobile phones
  - Subjects also reported their data: proximity and friendships
  - Analyzes proximity and friendships (inferred from recorded data) vs. ones that were self-reported by users

  - Conc-1: Two data sources is overlapping but distinct
  - Conc-2: Accurately infer 95% of friendships based on the observational data alone, where friend dyads demonstrate distinctive temporal and spatial patterns in their physical proximity and calling patterns.

Inferring social ties from geographic coincidences (in PNAS'10)

David J. Crandall, Lars Backstromb, Dan Cosleyc, Siddharth Surib, Daniel Huttenlocher, and Jon Kleinberg

- Probabilistic Model
  - Infer the probability of two people being friends given their co-occurrences in space and time
  - Does not consider the frequency of co-visit
  - Simplifies the social network: one connection for each person

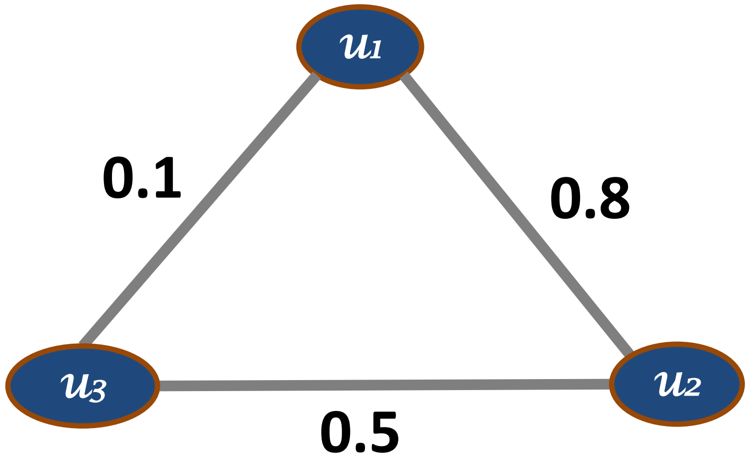
Bridging the Gap between Physical Location and Online Social Network (Ubicomp '10)

J. Cranshaw, E. Toch, J. Hong, A. Kittur, N. Sadeh

- Introduces a novel set of location based features for analyzing the social context of a geographical region
- **Location Entropy**: analyzes the context of the social interactions at that location: crowdedness and diversity
- **Regularity (Schedule_Entropy)**: High value reflects irregular movements, which produce high chance of making new friends
- Establishes a model of friendship in an online social network based on contextual features of co-locations

Example

	USC	SM Pier	Kodak Theater
(u_1, u_2)	4	3	2
(u_2, u_3)	2	2	1
(u_1, u_3)		5	



Problem Definition

Social strength is a quantitative measure that tells how socially close two people are.

Input: Users: $U = (u_1, u_2, \dots, u_M)$

Locations : $L = (l_1, l_2, \dots, l_N)$

Spatiotemporal records $< user\_id, location, time >$: $< u, l, t >$

Output: a weighted social graph where the weights of the edges define social strengths.

Challenges

1. What features of co-occurrences matter?
   - Richness?
   - Frequency?
   - Coincidences?
2. Location
   - Popularity?
   - Semantics?
3. Quantify friendships
   - Social Strength in between [0,1]

Baseline Solution - Richness

Counting the number of unique locations

Co-occurrence Vectors	Richness
C_{12}= (10, 1, 0, 0, 9)	**3**
C_{23}= (2, 3, 2, 2, 3)	**5**
C_{13}= (10, 0, 0, 0, 10)	**2**

✘ Ignore multiple co-occurrences @ same places

Baseline Solution - Frequency

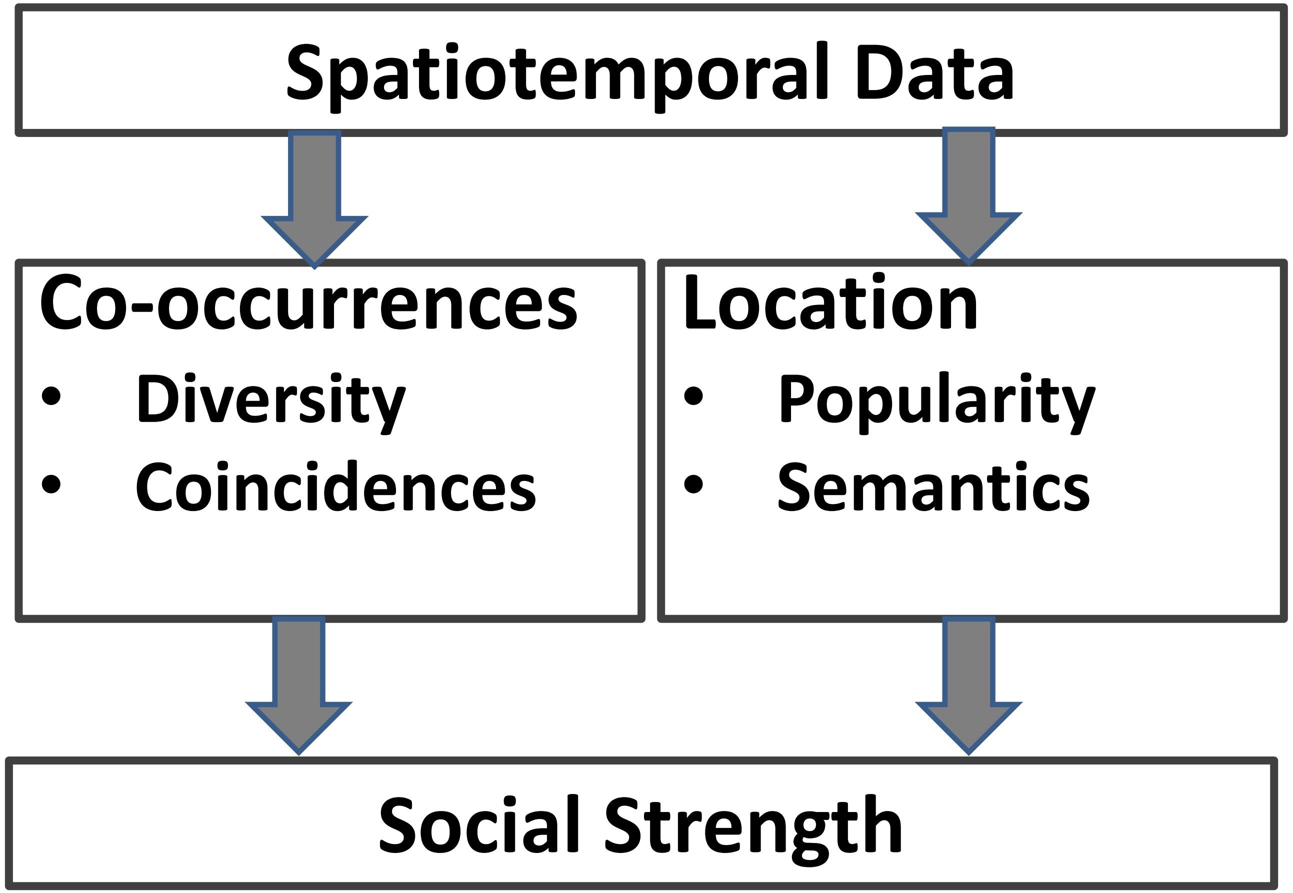
Counting the number of co-occurrences

Co-occurrence vectors	Frequency
C_{13}= (10, 1, 0, 0, 9)	**20**
C_{23}= (2, 3, 2, 2, 3)	**13**
C_{31}= (10, 0, 0, 0, 10)	**20**

✓ Captures local frequency

✗ Cannot capture the diversity of co-occurrences

EBM Model [SIGMOD'13]



Shannon Entropy

$$H_{ij}^{S} = -\sum_{l} P_{ij}^{l} \log P_{ij}^{l}$$

- If we select a random location, how predictable is whether i and j co-occurred there?
- More diverse places they co-occurred → Low predictability → High entropy

Co-occurrence vectors	H_{ij}^{s}
C_{12}= (10, 1, 0, 0, 9)	**0.86**
C_{23}= (2, 3, 2, 2, 3)	**1.59**
C_{13}= (10, 0, 0, 0, 10)	**0.69**

✓ The more locations, the higher entropy.

✓ The more diverse, the higher entropy.

✗ No control on diversity vs. frequency, e.g., may put too much weight on outliers (coincidences)

Rényi Entropy

We want to control the impact of diversity vs. frequency

$$H_{ij}^{R} = \left(-\log \sum_{l} \left(P_{ij}^{l} \right)^{q} \right) / (q-1)$$

Order of diversity

- $q > 1$ – Renyi entropy more favorably considers high local frequencies. (less diversity)
- q < 1 – in *opposite*, it gives more weight to low local frequencies.
- q = 1 – Renyi entropy is *undefined*, but its limit exists and becomes ***Shannon*** entropy, where it is unbiased.
- q = 0 the entropy is *insensitive* to local frequencies ⬄ giving pure number of unique locations – ***richness***.

✓ Captures the diversity of co-occurrences.

✓ Limits impact of coincidences (outliers).

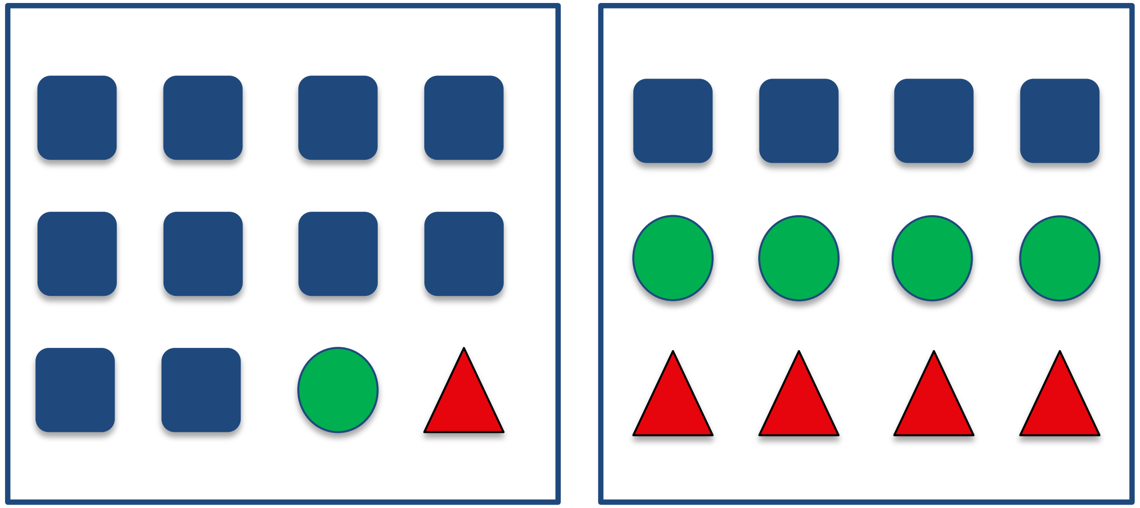
✗ Still considers all locations equally important. We need to consider:

- Location popularity

Location Entropy for Location Popularity

Frequency = 12
Diversity = 3

Less Popular
LE = 0.566



Location 1 **Location 2**

More Popular
LE = 1.099

Location Entropy (LE)

$$H_l = -\sum_{u, P_{u,l} \neq 0} P_{u,l} \log P_{u,l}$$

- LE indicates the popularity of a location *Cranshaw, J., et al., (2010). Bridging the gap between physical locations and online social networks. UBICOMP, 119-128.*
- The more popular, the higher entropy, and vice versa
- LE captures how diverse the visitors of a *location* are
  - E.g., your home is not diverse as only 2-4 users visited there; Eifel tower is the opposite
- Pick a random visit v at location l; high entropy means:
  - less predictable who made v
  - *The location has more diverse set of visitors*

The Entropy Based Model (EBM)

- Renyi Entropy

$$H_{ij}^{R} = \left(-\log \sum_{l} \left(P_{ij}^{l}\right)^{q}\right) / (q-1)$$

(How often i and j meet in how diverse of locations)

- Location Entropy

$$H_l = -\sum_{u, P_{u,l} \neq 0} P_{u,l} \log P_{u,l}$$

(How popular a location is)

- Weighted Frequency

$$F_{ij} = \sum_{l} c_{ij,l} \times \exp(-H_l)$$

(More weights to meetings in unpopular locations)

- Social Strength

$$s_{ij} = \alpha.\exp\left(H_{ij}^{R}\right) + \beta.\sum c_{ij}^{l} \times \exp(-H^{l}) + \gamma$$

Social Strength (EBM model)

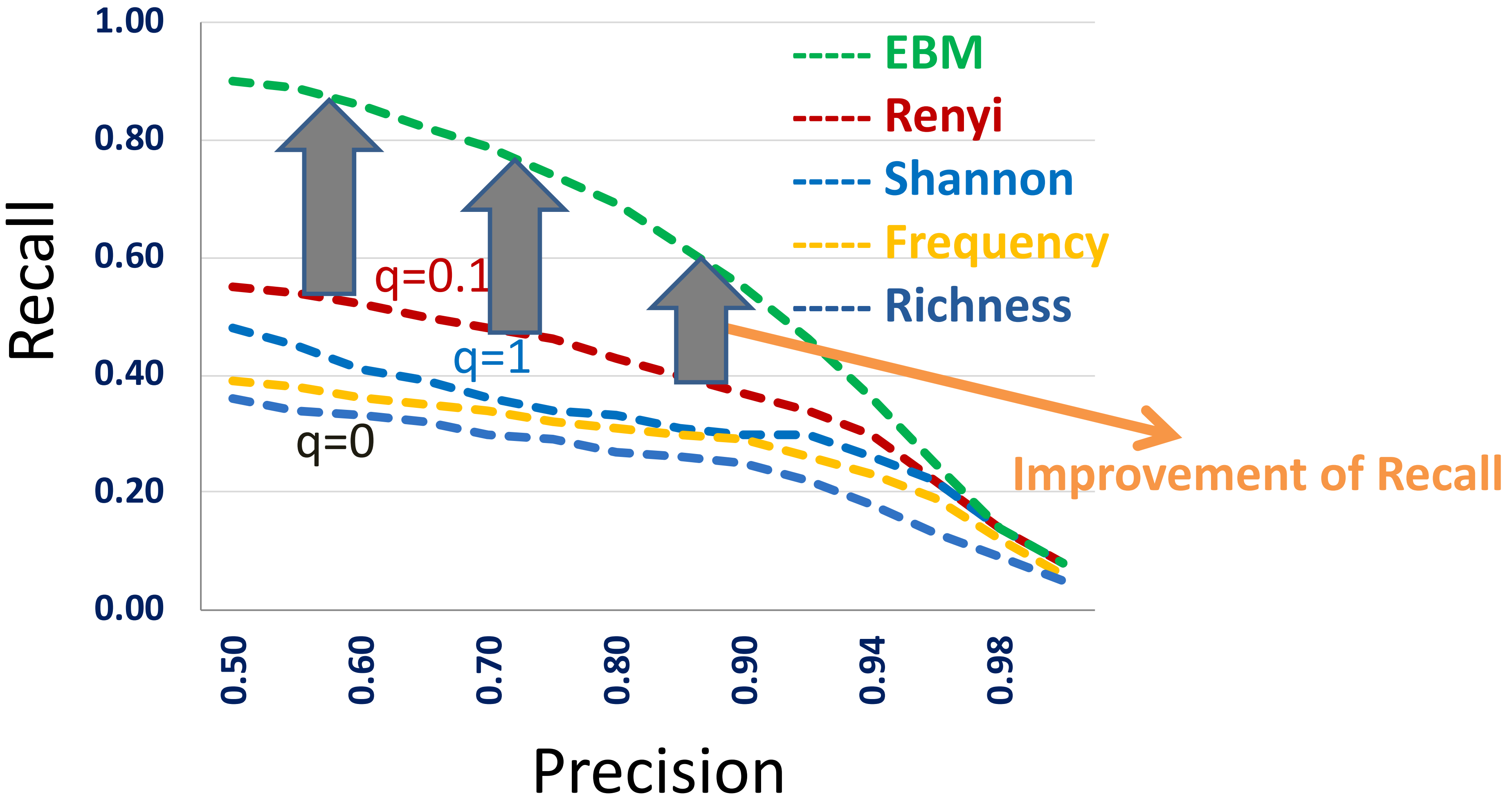
$$s_{ij} = \alpha.\exp\left(H_{ij}^{R}\right) + \beta.\sum c_{ij}^{l} \times \exp(-H^{l}) + \gamma$$

where parameter α, β and ϒ *can be learned from training data.*

Have addressed all the challenges mentioned earlier.

- ✓ Eliminate the impact of coincidences.
- ✓ Take into account the impact of locations.
- ✓ Data Sparseness.

Comparison of Various Social Strength Measures



OUTLINE

GeoSocial Queries [VLDB’13]

Inferring Social from Geo [SIGMOD’13]

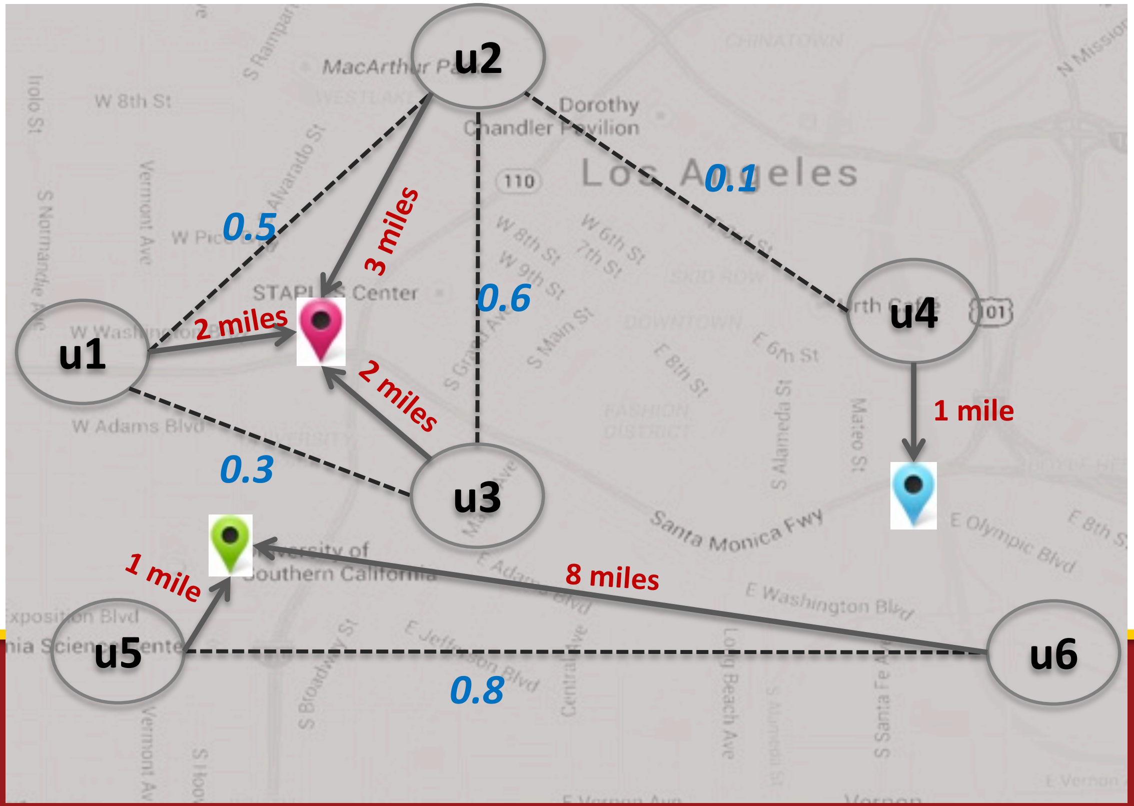
GeoSocial Recommendation [SIGMOD’15]

Future [SIGSPATIAL’15]

GeoSocial Recommendation [SIGMOD'15]

Real-Time Multi-Criteria Social Graph Partitioning

$$\underbrace{RMGP(G,P,\alpha)}_{\text{Minimizing Total Cost}} = \alpha \cdot \underbrace{\sum_{v \in V} c(v, s_v)}_{\text{Travel Cost}} + (1-\alpha) \cdot \underbrace{\sum_{\substack{e=(v,f)\in E \wedge \\ s_v \neq s_f}} w_e}_{\text{Loss of Social Connectivity}} \quad (1)$$



Related Work

- Graph Partitioning
  - Attribute-based [J. Sun et al., SIGKDD '07]
  - Connectivity-based [J. Shi et al., TPAMI '00], [M. E. Newman et al., Physical Review '04]
  - Attribute & Connectivity-based [Y. van Gennip et al. SIAM JAP '13]

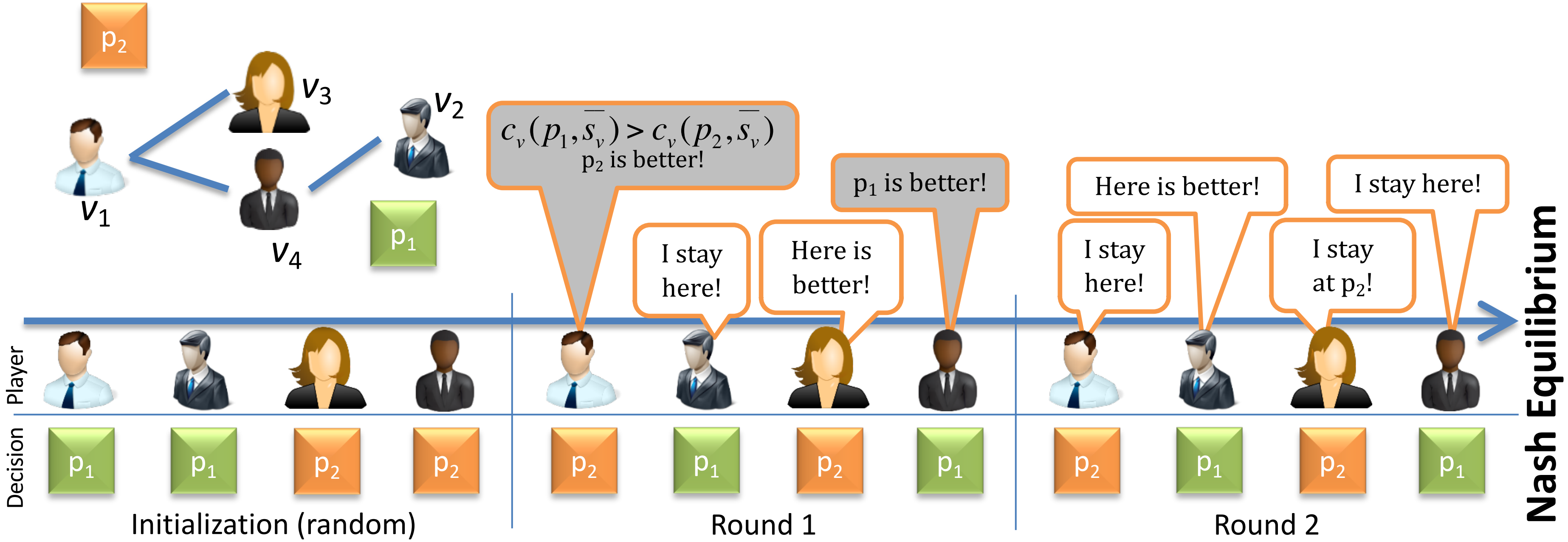
- Uniform Metric Labeling: Same objective function as RMGP, but studied only in theory. Solutions:
  - Linear Programming (UML_{lp}), and [J. Kleinberg et al., JACM '02]
  - Greedy (UML_{gr}). [E. C. Bracht et al., JEA '05]

GAME THEORETIC APPROACH

Every user $v \in V$ is a greedy player, who wants to attend the event s_v that minimizes his cost:

$$c_v(s_v, \overline{s_v}) = \alpha \cdot c(v, s_v) + (1-\alpha) \cdot \sum_{(e=(v,f)\in E)\wedge(s_v \neq s_f)} w_e$$

<u>Algorithm/Example</u>: α=0.5, equally weighted social edges



The game mimics the behavior of individual real-world users ☺

THEORETICAL RESULTS

1. Our game is an ***exact potential game*** -> always converges.

- Potential function:

$$\Phi(S)=\alpha\cdot\sum_{v\in V}c(v,s_v)+(1-\alpha)\cdot\frac{1}{2}\sum_{(e=(v,f)\in E)\wedge(s_v\neq s_f)}w_e$$

- When a user v moves from s_v to s'_v then:

$$C_v(s_v,\overline{s_v})-C_v(s'_v,\overline{s_v})=\Phi(s_v,\overline{s_v})-\Phi(s'_v,\overline{s_v})$$

[D. Monderer et al., Games and economic behavior, 1996]

2. Price of anarchy is upper-bounded:

$$\frac{cost\ of\ worst\ equilibrium}{global\ optimum}\leq 1+\frac{(1-\alpha)}{\alpha}\cdot\frac{deg_{avg}\cdot w_{avg}}{2\cdot c^{*}_{avg}}$$

deg_{avd} average degree

OUTLINE

- GeoSocial Queries [VLDB'13]
- Inferring Social from Geo [SIGMOD'13]
- GeoSocial Recommendation [SIGMOD'15]
- Future [SIGSPATIAL'15]

Two Sides of the Coin

Protecting against location disclosure
**** But allow for Social Inference***