# Current topics of Research

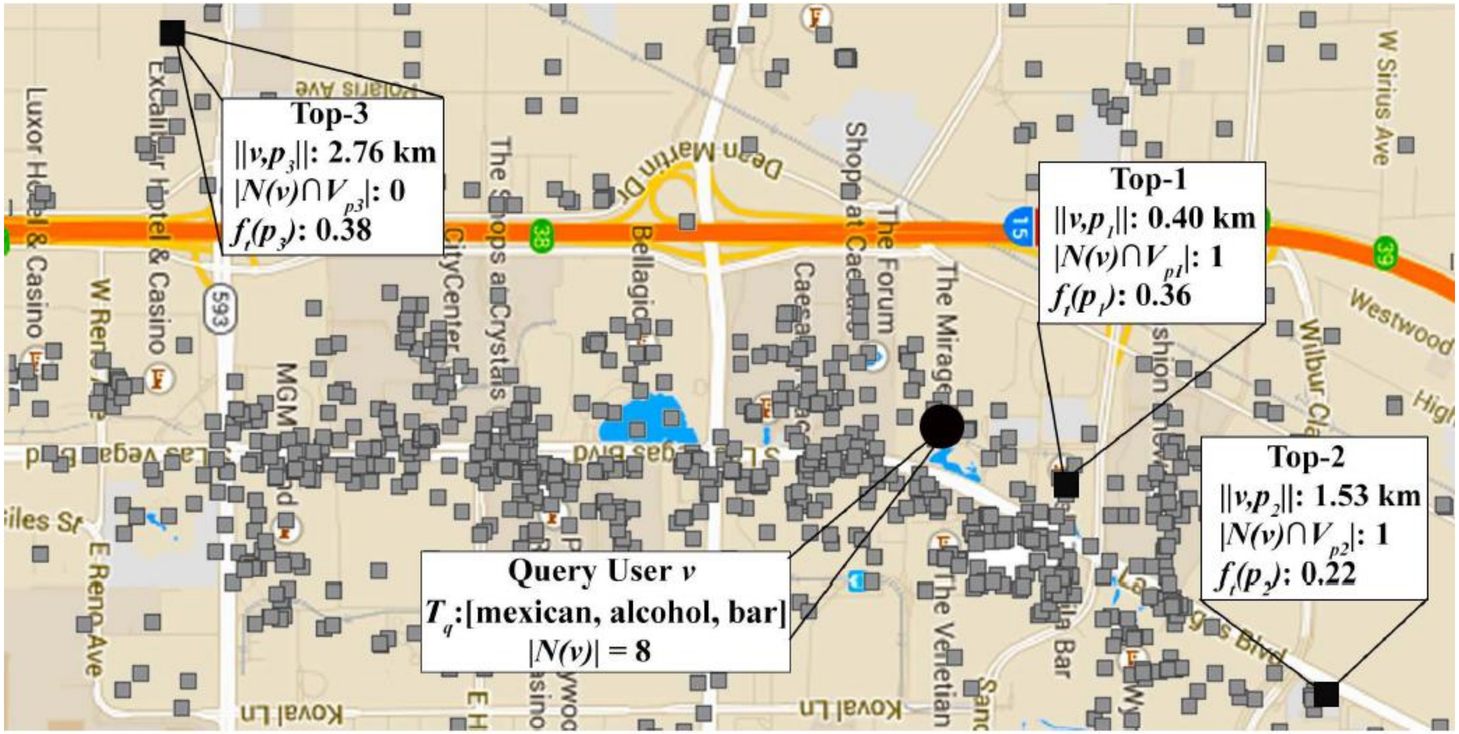
1. **Geo-social Networks**
2. **Graph Partitioning**
3. **Uncertain Graphs**

Dimitris Papadias and my students

Hong Kong University of Science and Technology

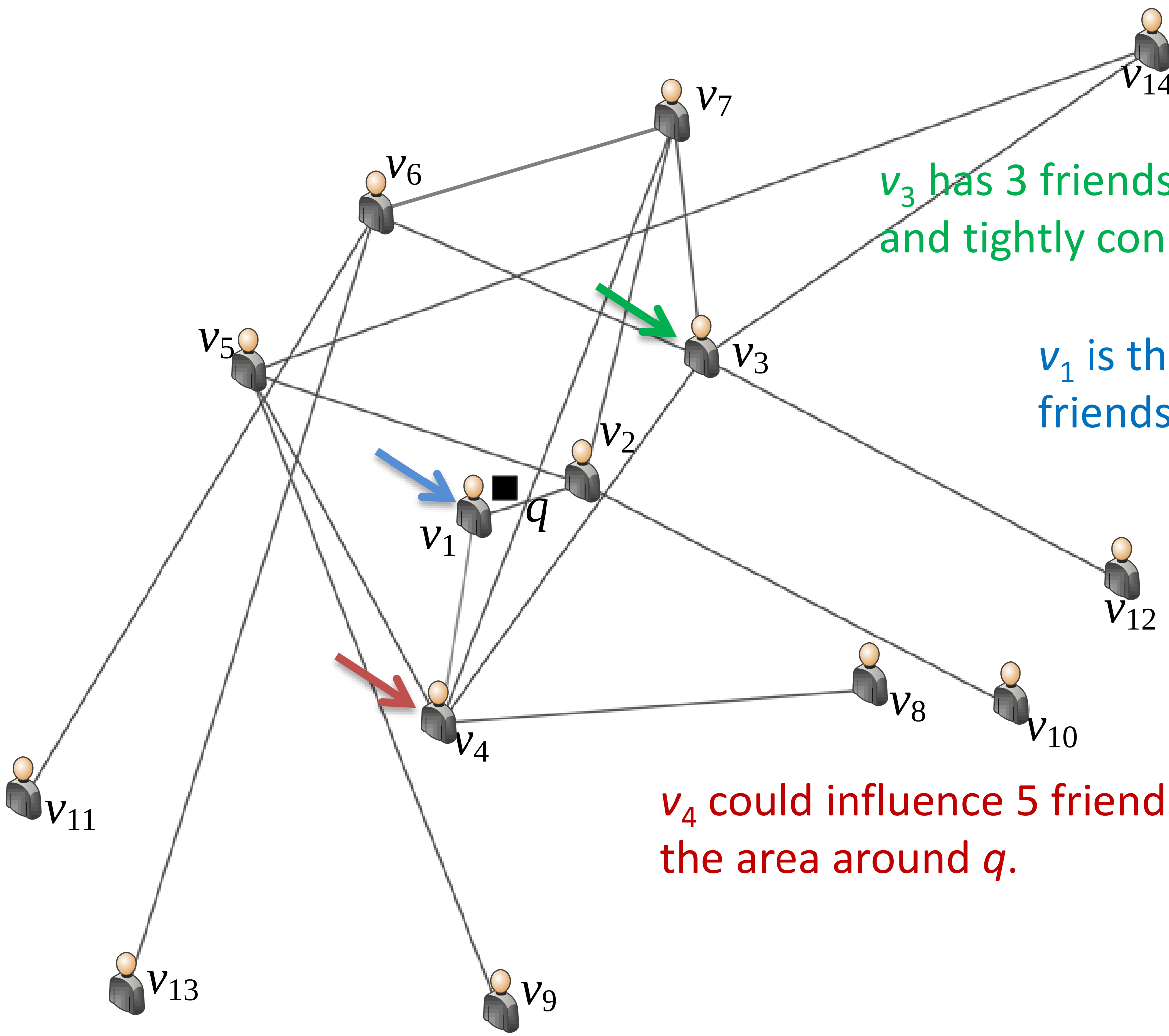
# 1. Geosocial Networks (VLDB 13, SSTD 15)

- General architectures
- Various query types that combine social , geographic and textual aspects



# Geo-Social Ranking (VLDBJ 15)

Example: Who is the top-1 user for query location *q* ?




$v_3$ has 3 friends ($v_4$, $v_6$, $v_7$), reasonably close to $q$, and tightly connected to each other.

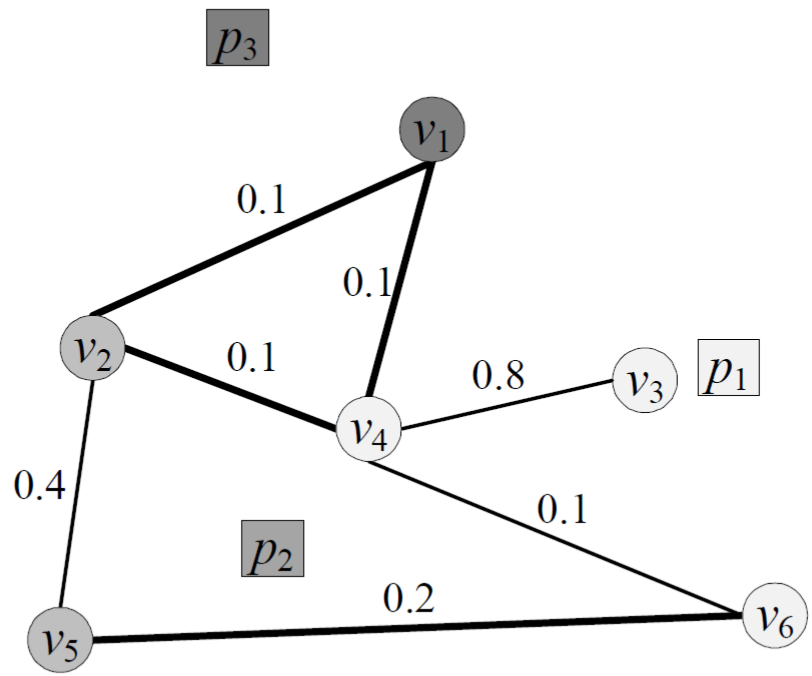
$v_1$ is the closest to $q$, and has two friends ($v_2$, $v_4$) that are very near $q$.

$v_4$ could influence 5 friends ($v_1$, $v_3$, $v_5$, $v_7$, $v_8$) in the area around $q$.

# 2. Multi-criteria graph partitioning (SIGMOD 15)

Example: We wish to <u>promote</u> (recommend) upcoming events.
Assign each user to an event that minimizes

- the distance/travel time between the user and the event, and
- the social connectivity between users assigned to different events.




| | $\|\|v_i, p_1\|\|$ | $\|\|v_i, p_2\|\|$ | $\|\|v_i, p_3\|\|$ |
|---|---|---|---|
| $v_1$ | 0.48 | 0.6 | 0.27 |
| $v_2$ | 0.8 | 0.34 | 0.44 |
| $v_3$ | 0.1 | 0.54 | 0.67 |
| $v_4$ | 0.47 | 0.2 | 0.54 |
| $v_5$ | 0.94 | 0.3 | 0.8 |
| $v_6$ | 0.34 | 0.67 | 0.99 |

Event

User

- Another criterion: Textual (dis)similarity
- Combination of criteria: Euclidean distance + Textual dissimilarity

# Game theoretic solution

Each user is a player who has a cost function that depends on the event that he will attend and his friends' decisions.

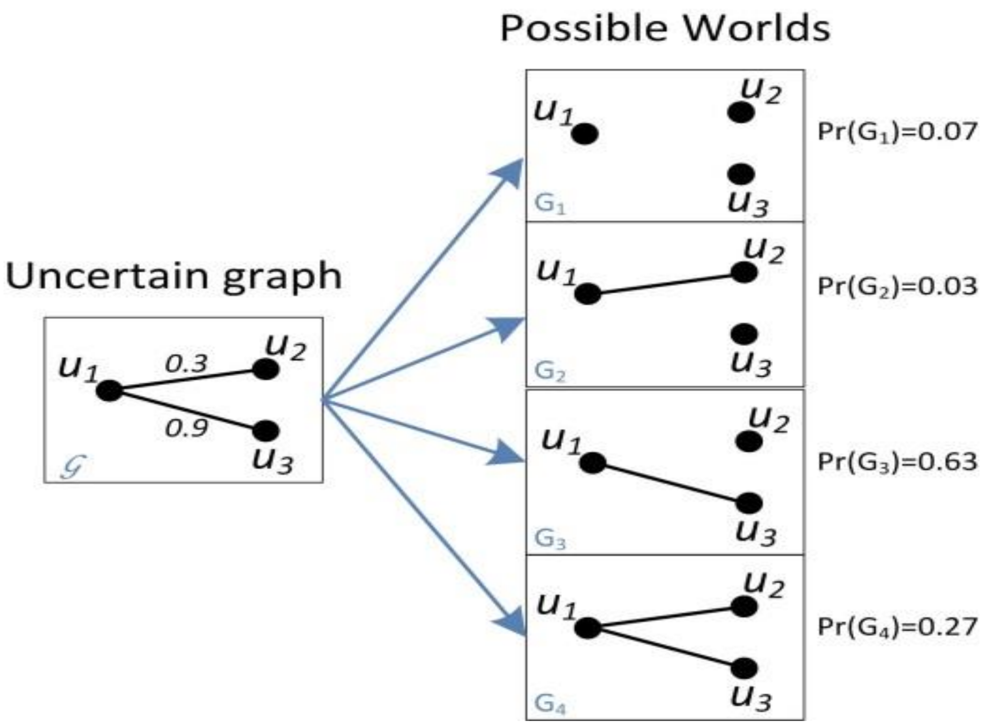
- His goal is to attend the event that minimizes his own cost function.

Algorithm (Best-Responses)

1. Assign a random strategy (event) to each player
2. Repeat
3. For each player $v \in V$
4. compute *v*'s best event wrt the other players´ strategies
5. let *v* follow his best strategy
6. Until no player has incentive to change strategy (Nash equilibrium)
7. Return the strategy of each player

- Several optimizations for centralized and distributed architectures
- Can partition large graphs in seconds or minutes

# 3. Uncertain graphs: edge probabilities




- Possible world semantics: interprets uncertain graphs as a collection of $2^{|E|}$ deterministic graphs (possible worlds).
- Expected degree of a node u: the sum of the probabilities of the edges incident to u (e.g. $[deg_{u1}] = 1.2$).

# How can we answer common queries (e.g. kNN, shortest path) on uncertain graphs?

The exact answer requires materialization of all possible worlds and query execution in each world.

*Monte Carlo sampling*

1. Generate numerous samples
2. Process the query on each sample
3. Aggregate partial results

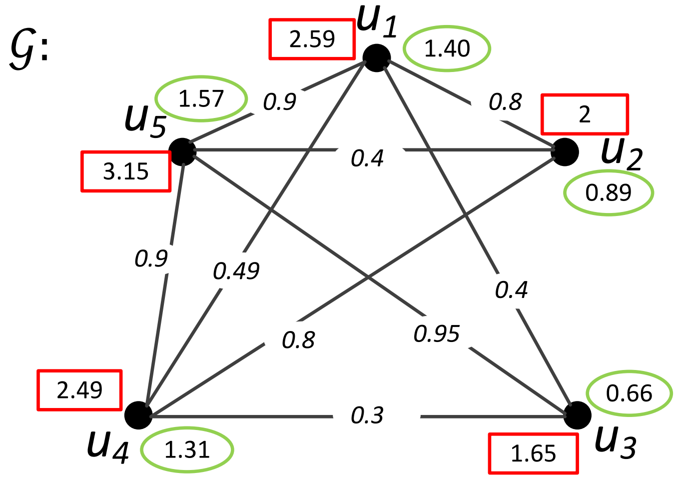
- Extremely expensive
  - E.g., queries such as betweeness centrality require all-pairs shortest path computations, which must be performed on all samples

# Our first solution (SIGMOD 14, TODS15)

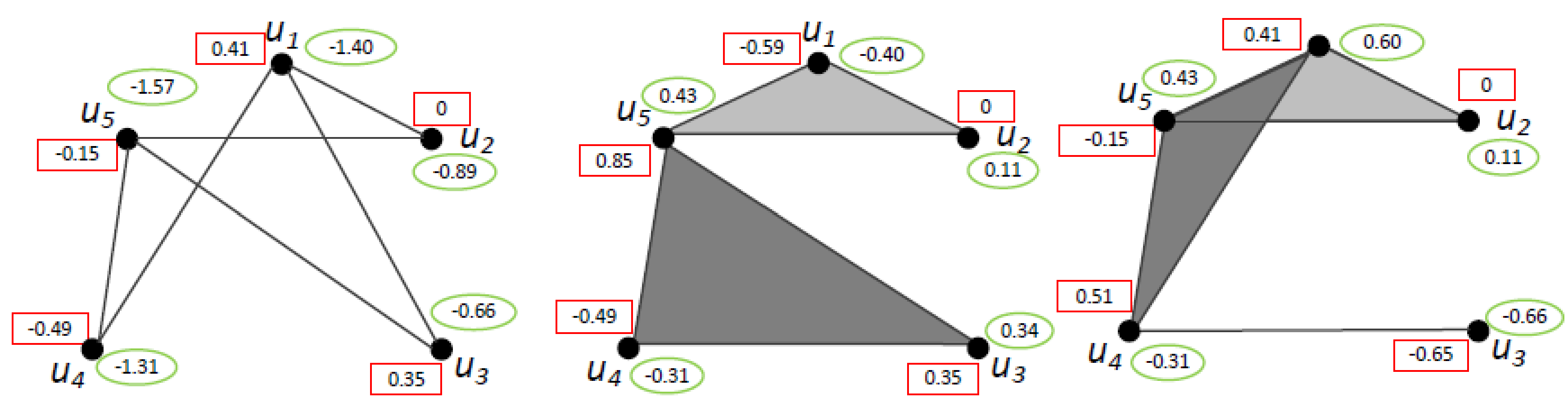
- From all possible worlds extract a representative instance that preserves the structural properties of the uncertain graph.
  - Preserve the expected degree of every node
  - Preserve the $n$-qlique cardinality of every node.

- Queries are then processed approximately on the representative using conventional (deterministic) query processing methods.
  - Very efficient and accurate

# Example




| representative | $\Delta_2$ | $\Delta_3$ | $\Delta_2 + \Delta_3$ |
|---|---|---|---|
| $G_2^*$ | **1.40** | 5.83 | 7.23 |
| $G_3^*$ | 2.28 | **1.59** | 3.87 |
| $G_{2,3}^*$ | 1.72 | 2.11 | **3.83** |




(a) $G_2^*$ (b) $G_3^*$ (c) $G_{2,3}^*$

# Our second solution (on going work)

- Sparsify uncertain graphs.
- Reduce the number of edges in the graph and modify the probabilities of the remaining ones to preserve the structural properties.

Queries are then processed approximately on the sparse graph using Monte Carlo sampling.