Event detection in activity networks

Polina Rozenshtein\textsuperscript{1}
Aris Anagnostopoulos\textsuperscript{2}
Aristides Gionis\textsuperscript{1}
Nikolaj Tatti\textsuperscript{1}

\textsuperscript{1}HIIT, Aalto University \cdot \textsuperscript{2}Sapienza University of Rome

August 25, 2014
City events

- **City event**: macroscopic-level activity in the city, which takes place in space and time, and falls outside the normal city life cycle
  - social event, festival, traffic accident, weather disaster, ..
City events

15.11.2012: Normal day, no events

11.09.2012: National day of Catalonia; FC Barcelona - Igualada HC
Event detection problem formulation

- Locations \( V = \{v_1, \ldots, v_n\} \)
- Location \( v_i \) has coordinates \((x_i, y_i)\)
- Distance \( d(u, v) \) between location \( u \) and \( v \)
- Activity at location \( v_i \) recorded in time series \( t_i \)

- For fixed time snapshot:
  - Weight \( w(v) \): deviation from normal activity at location \( v \)
  - Estimated using a predictive model
  - Sophisticated models can be incorporated
Event detection problem formulation

- **Input:** graph $G = (V, E, d, w)$
  - Distance function $d : E \rightarrow R$
  - Weight function $w : V \rightarrow R$

- **Find** subset of locations $S \subseteq V$ with
  - High activity weight $W(S) = \sum_{v \in S} w(v)$
  - Compactness $D(S)$

- **Compactness** $D(S)$:
  1. All pair distances $D_{AP}(S) = \frac{1}{2} \sum_{u \in S} \sum_{v \in S} d(u, v)$
  2. Minimal spanning tree $D_T(S) = \min_T \sum_{(u, v) \in T} d(u, v)$

- **Maximize** $Q(S) = \lambda W(S) - D(S)$
Shifted objective

- Maximize $Q(S) = \lambda W(S) - D(S)$ is \textbf{NP}-hard
- Ensure non-negativity by adding a constant term

Shifted objective: all pairs

- Maximize $Q_{AP}(S) = \lambda W(S) - D_{AP}(S)$
- Maximize $Q_{AP}(S) = \lambda W(S) - D_{AP}(S) + D_{AP}(V)$

Shifted objective: spanning tree

- Maximize $Q_T(S) = \lambda W(S) - D_T(S)$
- Minimize $Q_T(S) = D_T(S) + \lambda W(V \setminus S)$
- Prize-Collecting Steiner Tree
Algorithms: all pair distances

\[ \frac{1}{2} \text{ – approximations} \]

- Function \( Q_{AP}(S) \) is **sub-modular** (but not monotone)
- **Trivial**: all or nothing
  - \( S = \arg \max \{ Q_{AP}(\emptyset), Q_{AP}(V) \} \)
Algorithms: all pair distances

$\frac{1}{2}$ – approximations

- Function $Q_{AP}(S)$ is sub-modular (but not monotone)
- Trivial: all or nothing
  - $S = \arg \max \{Q_{AP}(\emptyset), Q_{AP}(V)\}$
- Greedy algorithm
  - start with the empty set
  - iteratively add the best vertex until no improvement
Algorithms: all pair distances

$1/2$ – approximations

- Function $Q_{AP}(S)$ is **sub-modular** (but not monotone)
- **Trivial**: all or nothing
  - $S = \arg \max \{ Q_{AP}(\emptyset), Q_{AP}(V) \}$
- **Greedy** algorithm
  - start with the **empty set**
  - iteratively **add** the **best vertex** until no improvement
- **Double-greedy** algorithm by Buchbinder et al.[1] (BFNS)
  - Maintain two sets $X = \emptyset$ and $Y = V$
  - For each $v \in V$:
    - If $Q_{AP}(X \cup v) - Q_{AP}(X) > Q_{AP}(Y \setminus v) - Q_{AP}(Y)$,
      then add $v$ to $X$
    - otherwise, delete $v$ from $Y$
  - At the end $X = Y$, return $X$
Algorithms: all pair distances

- Algorithm based on SDP relaxations
  - Seminal work of Goemans and Williamson [3]
  - 0.878-approximation for $\text{MaxCut}$

- Possible to adapt the technique for our problem
- 0.878-approximation for $(s, t)$-$\text{MaxCut}$
Algorithms: Prize-Collecting Steiner Tree (PCST)

- Primal-dual algorithm
  - $2 - \frac{1}{n-1}$ – approximation (Goemans and Williamson [2])
Algorithms: Prize-Collecting Steiner Tree (PCST)

- **Primal-dual algorithm**
  - $2 - \frac{1}{n-1}$ – approximation (Goemans and Williamson [2])

- **Two-phase algorithm**
  - $2$ – approximation by Johnson et al.[4]
  - 1. **Merging phase:**
    - Start with each vertex as a component
    - Merge components based on weights and edge distances
  - 2. **Bottom-up pruning phase**
Algorithms: Prize-Collecting Steiner Tree (PCST)

- **Primal-dual algorithm**
  - $2 - \frac{1}{n-1}$ – approximation (Goemans and Williamson [2])

- **Two-phase algorithm**
  - $2$ – approximation by Johnson et al.[4]
    1. **Merging phase:**
       - Start with each vertex as a component
       - Merge components based on weights and edge distances
    2. **Bottom-up pruning phase**

- **Same simple greedy as with all pairs distance**
  - start with the empty set
  - iteratively add the best vertex at that step
  - repeat until there is no improvement
Experimental evaluation

- **Synthetic datasets**
  - Planted events + noise
- **Bicing datasets** (sensor networks)
  - Activity = number of bikes
  - Barcelona, Minneapolis, and Washington DC
- **Location-based social networks**
  - Geo-tagged tweets
  - Activity = number of tweets in city locations
  - 100 cities in US
  - Experiments with NY and LA
Typical behavior: algorithms for All Pairs model

<table>
<thead>
<tr>
<th>All pairs</th>
<th>Trivial</th>
<th>Greedy</th>
<th>Double Greedy (BFNS)</th>
<th>SDP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.878</td>
</tr>
</tbody>
</table>

![Graph showing cost vs. weight multiplier for different algorithms.](image-url)
Typical behavior: algorithms for Tree model

<table>
<thead>
<tr>
<th>PCST</th>
<th>Primal Dual</th>
<th>Simple Greedy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>no</td>
</tr>
</tbody>
</table>

![Graph showing cost vs. weight multiplier for GreedyT and PD algorithms.](image-url)
Some of the top event days detected

Barcelona: 18.09.12 festival of the Poblenou neighborhood
Barcelona: 01.06.12 Primavera sound music festival
Barcelona: 31.10.12 Halloween
New York: 6.09.10 Labor Day
Los Angeles: 31.05.10 Memorial Day
Washington, DC: 27.05.13 Memorial Day
Summary and future work

- Detecting events in activity networks
- Find high-activity compact subareas in the city
- Developed approximation algorithms and baselines
- Experimented with urban and social media datasets

- Scalability
  - Incorporate time dimension in the framework and discover events of varying temporal support
- Event evolution and tracking
References


Thank you!