Anomalous and Significant Subgraph Detection in Attributed Networks

Feng Chen [1], Petko Bogdanov [1], Daniel B. Neill [2], and Ambuj K. Singh [3]

[1] Department of Computer Science
College of Engineering and Applied Sciences
University at Albany - SUNY

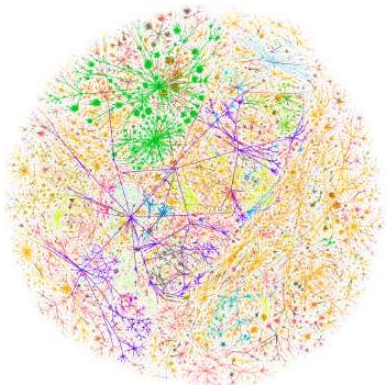
[2] Event and Pattern Detection Laboratory
H.J. Heinz III College
Carnegie Mellon University

[3] Department of Computer Science &
Biomolecular Science and Engineering
University of California at Santa Barbara

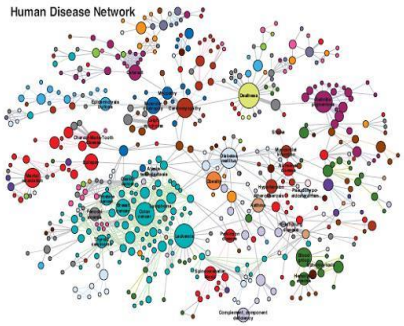
Roadmap

- Introduction and motivation
- Part 1: Subgraph detection in static attributed networks
- Part 2: Subgraph detection in dynamic attributed networks
- Conclusion and future directions

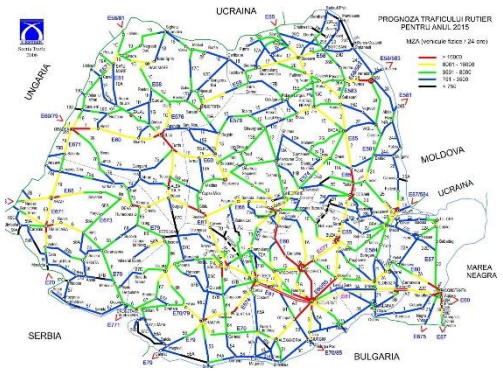
Real-world networks



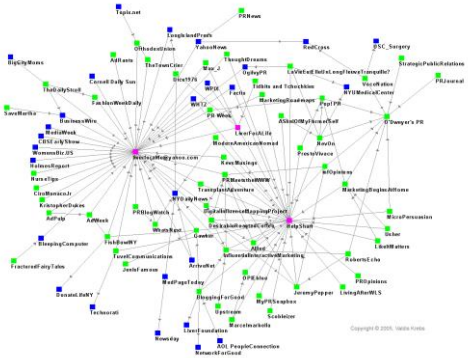
Internet map



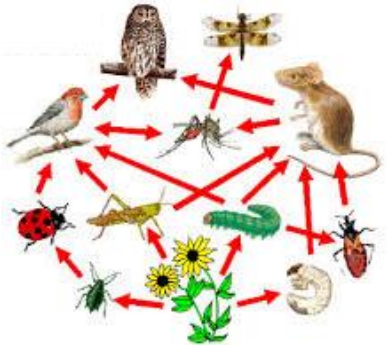
Biological networks



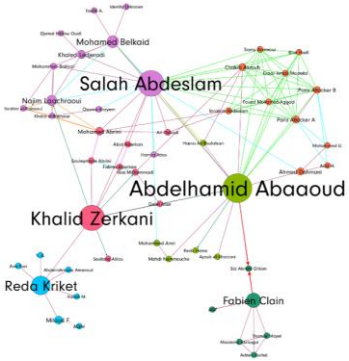
Road networks



Blog networks

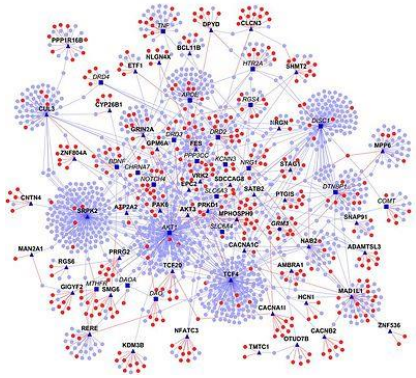


Food web

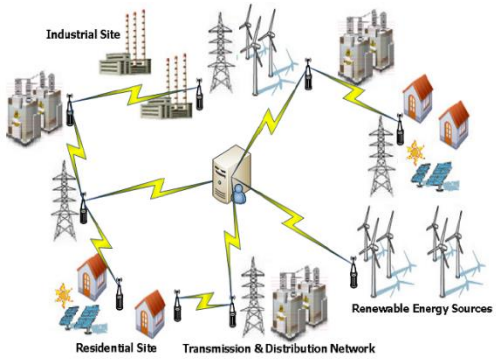


Terrorist networks

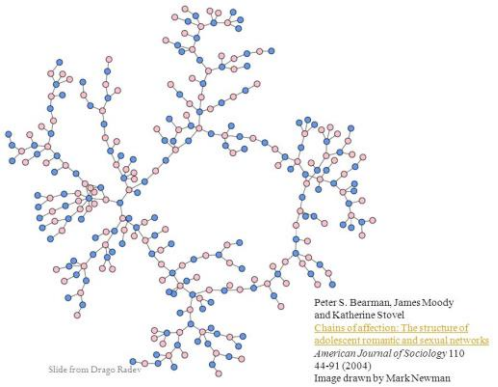
Real-world networks



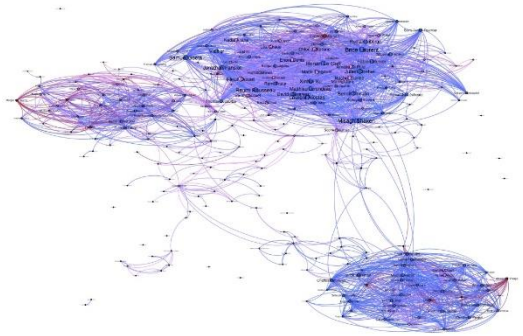
Protein-protein interaction networks



Power grid networks

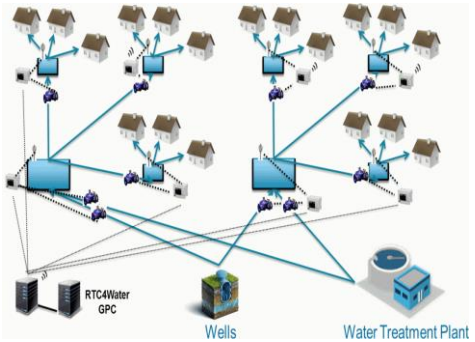


Dating networks



Facebook friends' networks

Retail networks



Water distribution networks

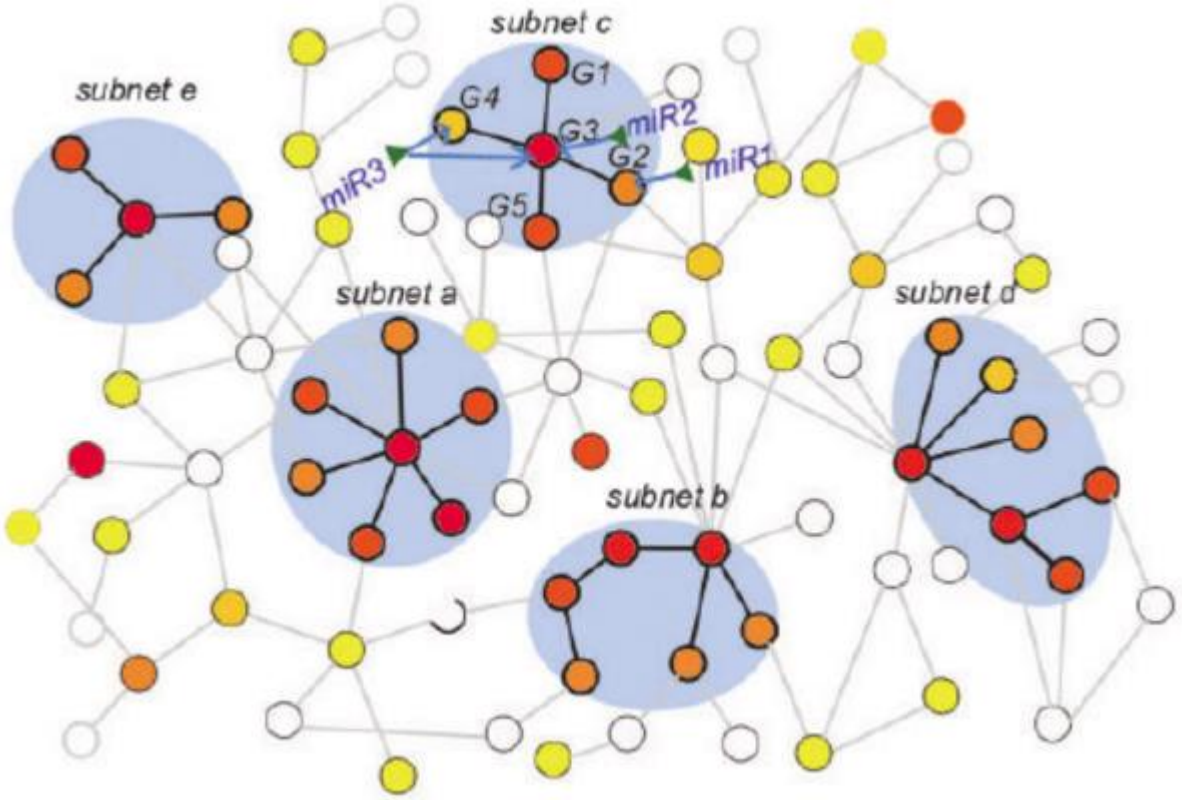
Anomalous & significant subgraphs

Anomalous and significant subgraphs refer to subgraphs, in which the behaviors (attributes) of the nodes or edges are significantly different from the behaviors of those outside the subgraphs.

This tutorial mainly reviews methods on detection of anomalous and significant subgraphs with connectivity constraint.

Anomalous & significant subgraphs

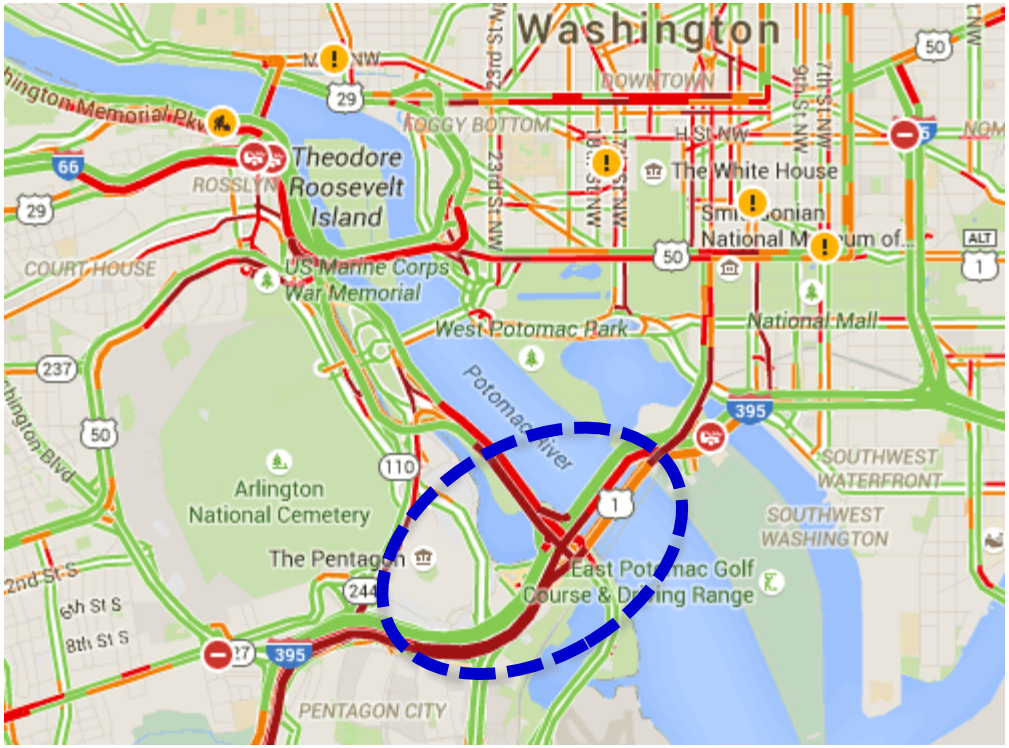
- Detection of subnetwork biomarkers



(Chuang et al. 2007)

Anomalous & significant subgraphs

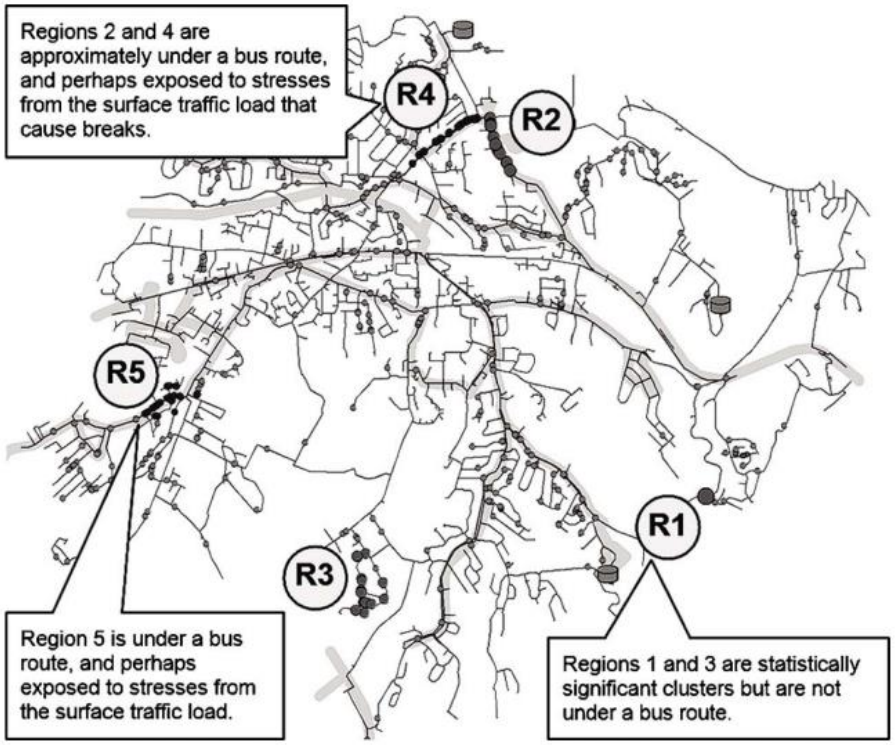
- Detection of road traffic congestion events



https://mikethemadbiologist.com/2015/08/08/the-ripple-effects-of-mass-transit/

Anomalous & significant subgraphs

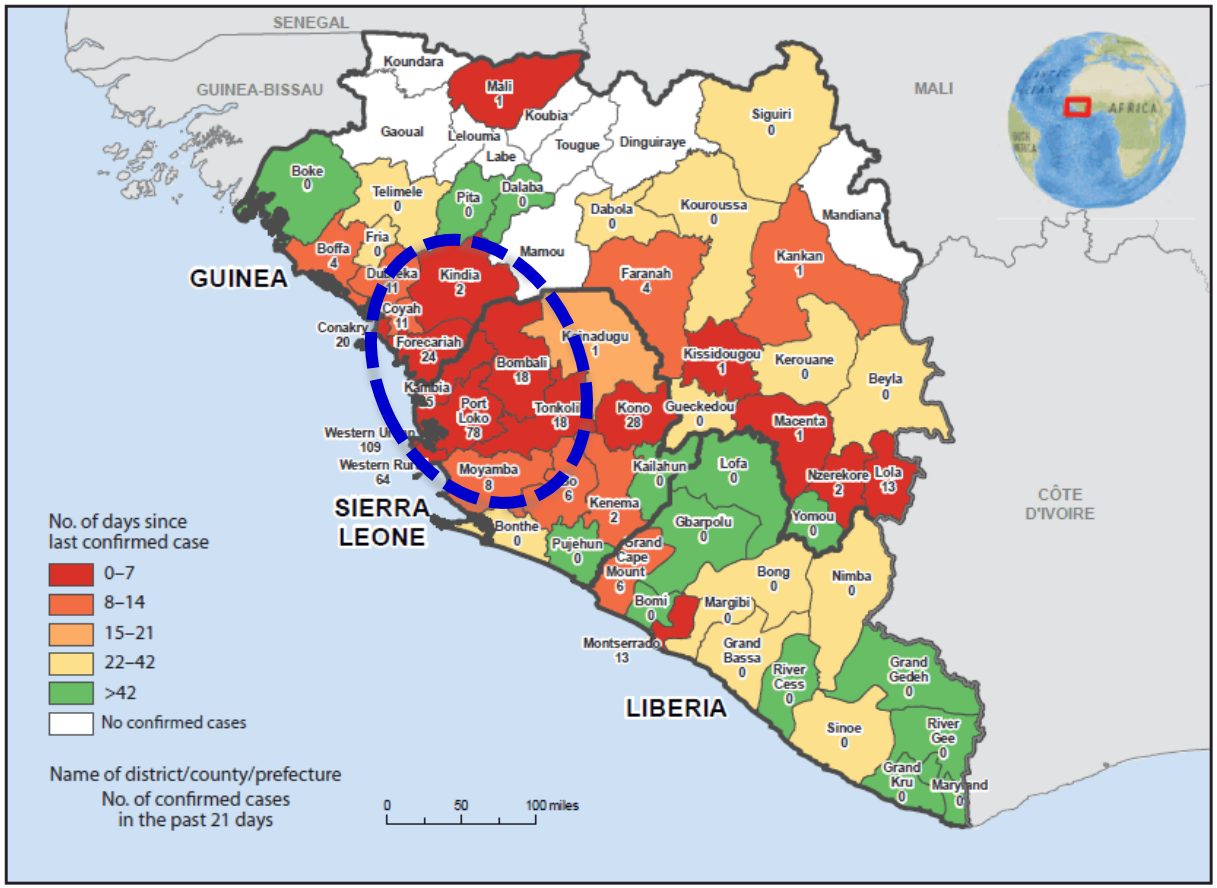
- Detection of abnormally high breakage in a distribution network



(de Oliveira et al., 2010)

Anomalous & significant subgraphs

- Detection of disease outbreaks



http://alfa-img.com/show/ebola-epidemic-map-2015.html

Other applications

Societal events in social media

Malicious cargo

Image/video surveillance

New business discovery

Auction fraud, fake reviews, email spams, false advertising

Extreme weather events

Crime hotspots

Brain activities

Disease diagnosis

Animal activities

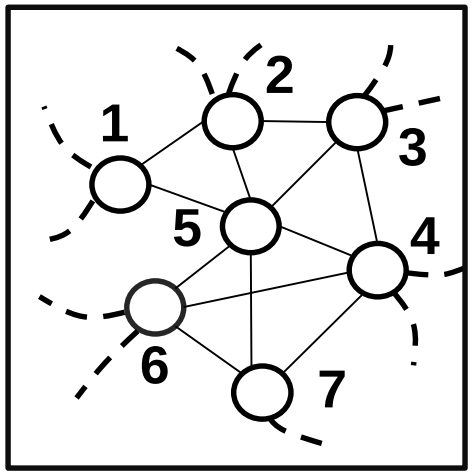
New chemical structures

New knowledge discovery

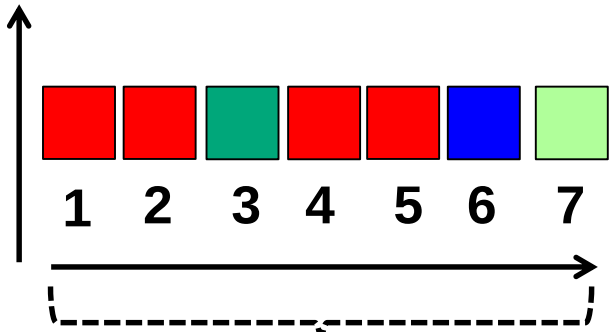
Subgraph detection: definition

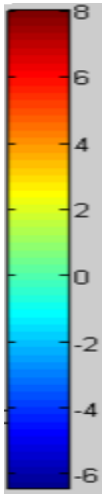
- Univariate static networks

Network topology $\mathbb{G} = (\mathbb{V}, \mathbb{E})$



Attributes (w)





$F(S)$ characterizes the level of anomalousness of S based on attributes.

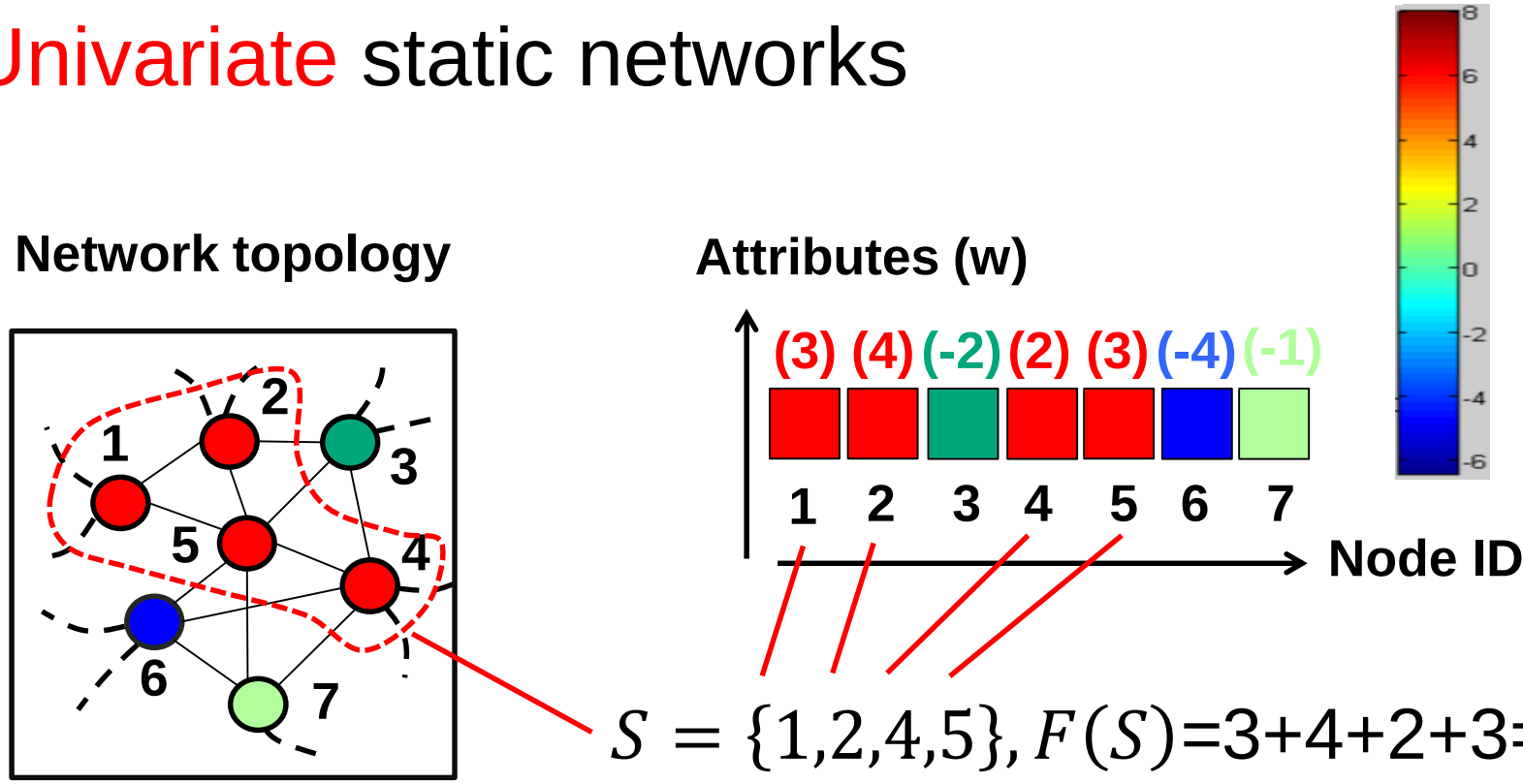
Constraint is defined based on network topology.

$$\max_{S \subseteq \mathbb{V}} F(S)$$

$s.t.$ S satisfies a predefined topological constraint (e.g. connectivity).

Subgraph detection: definition

- Univariate static networks



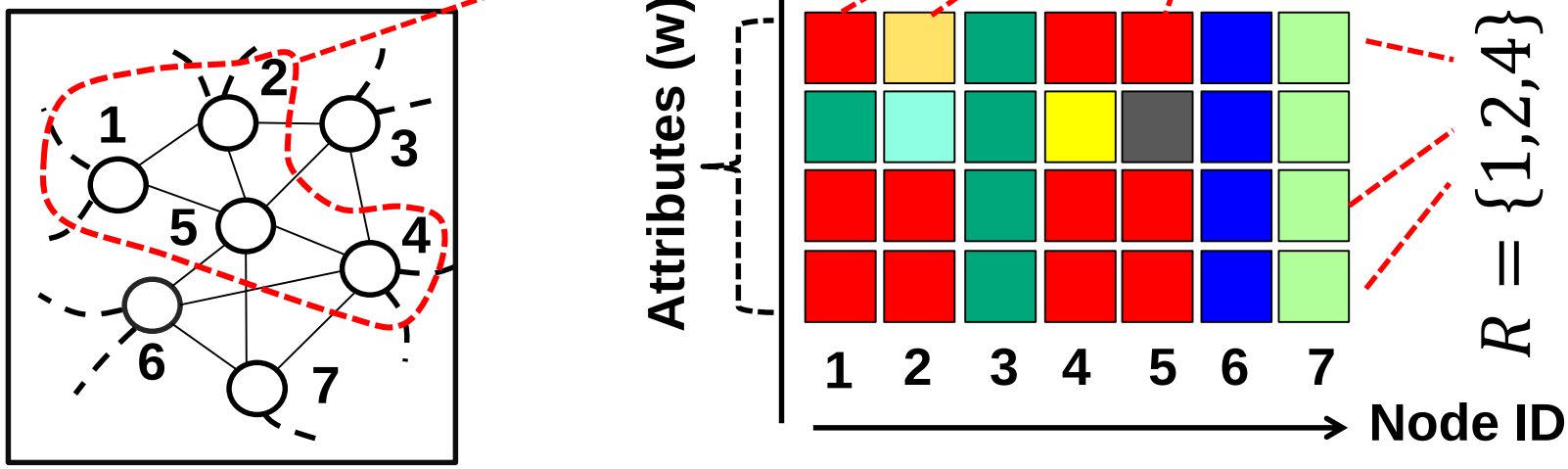
$S = \{1,2,4,5\}, F(S)=3+4+2+3=12$

$$\max_{S \subseteq \mathbb{V}} F(S) = \sum_{i \in S} w(i)$$

$$s.t.\ S \text{ is connected}$$

Subgraph detection: definition

- Multivariate static networks



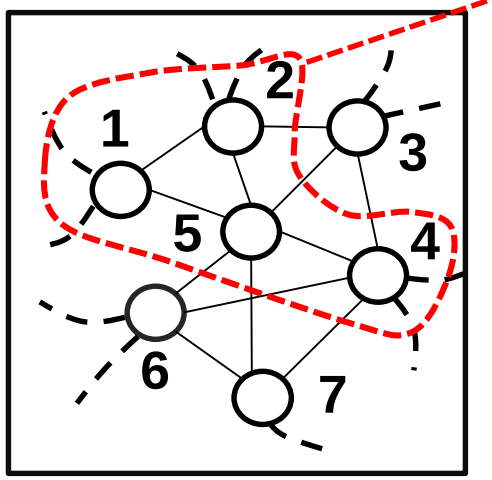
Constraint is defined based on network topology.

$$\max_{S,R} F(S,R)$$

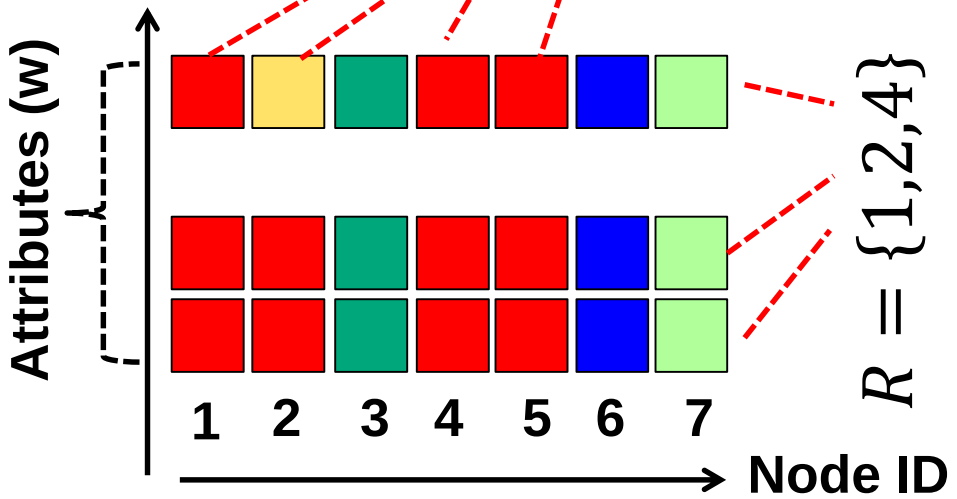
$s.t.$ S satisfies a predefined topological constraint

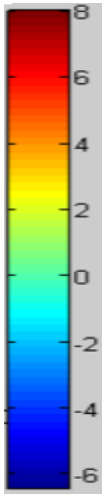
Subgraph detection: definition

- Multivariate static networks



$S = \{1,2,4,5\}$





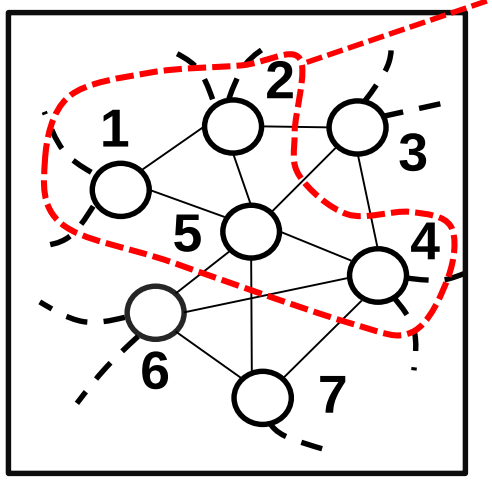
Constraint is defined based on network topology.

$$\max_{S,R} F(S,R)$$

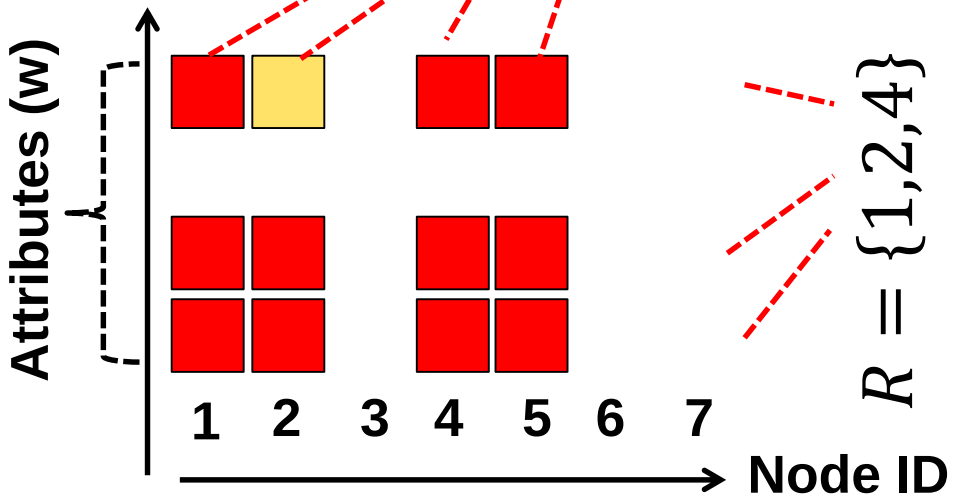
$s.t.$ S satisfies a predefined topological constraint

Subgraph detection: definition

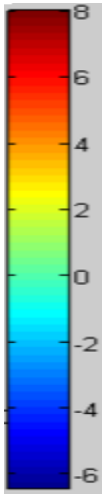
- Multivariate static networks



$S = \{1,2,4,5\}$



$R = \{1,2,4\}$



Constraint is defined based on network topology.

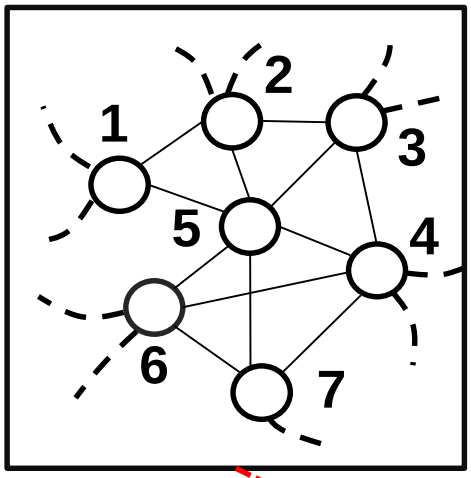
$$\max_{S,R} F(S,R)$$

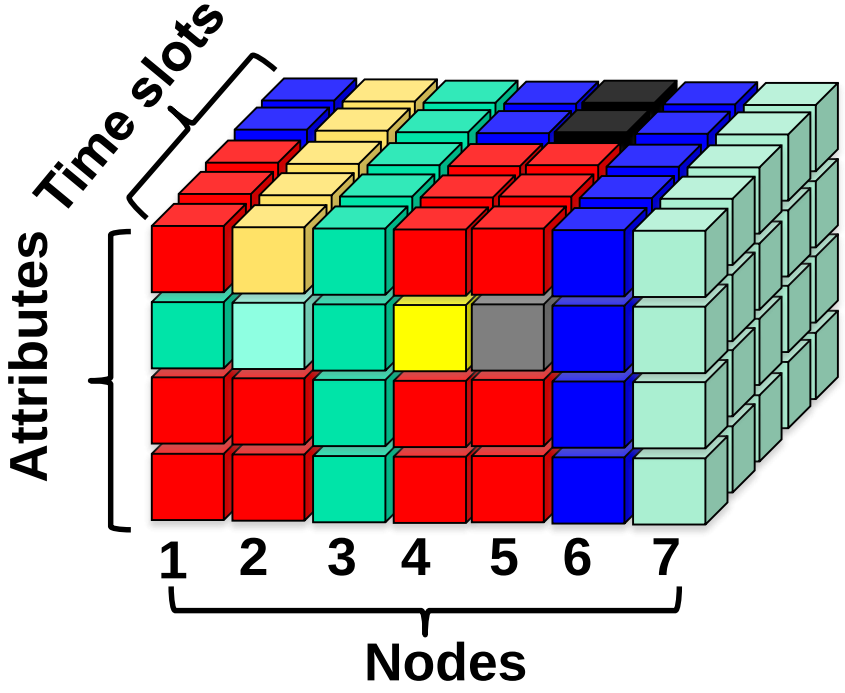
$s.t.$ S satisfies a predefined topological constraint

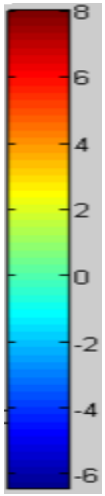
Subgraph detection: definition

- Multivariate dynamic networks

Network topology





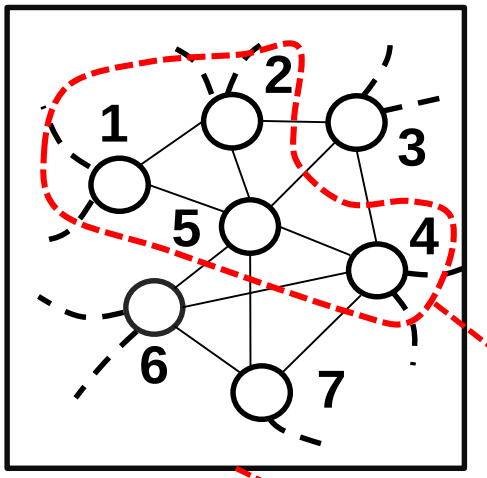


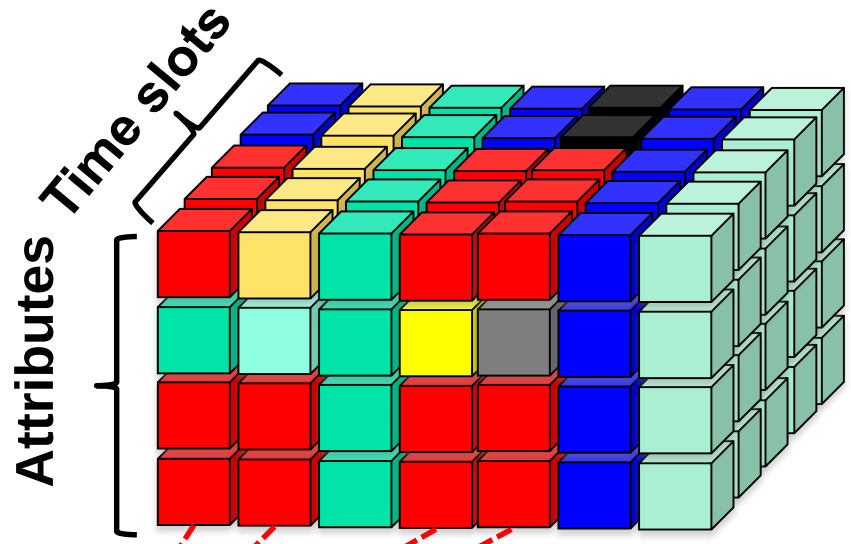
Constraint is defined based on network topology.

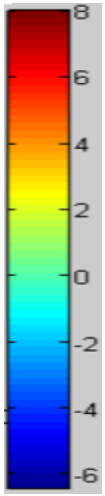
$$\max_{S,R,W} F(S,R,W)$$
$$s.t.\ S \text{ satisfies a predefined topological constraint}$$

Subgraph detection: definition

Network topology







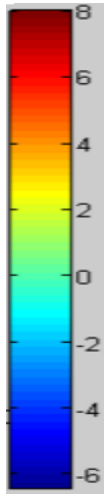
$S = \{1,2,4,5\}$

Constraint is defined based on network topology.

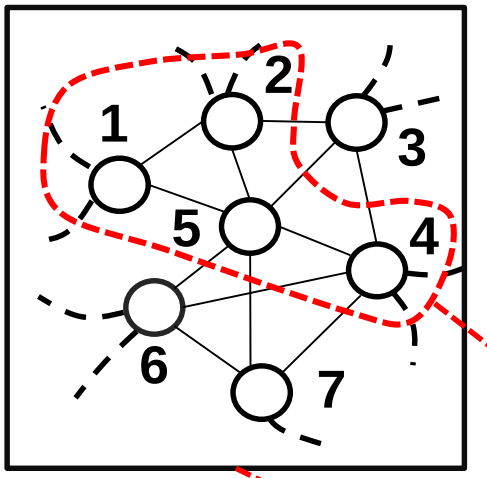
$$\max_{S,R,W} F(S,R,W)$$

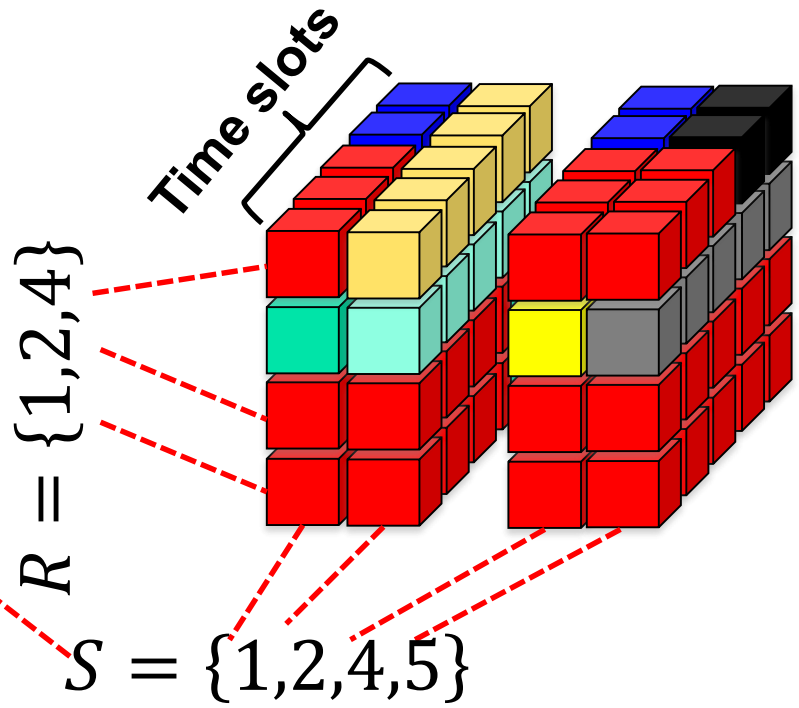
$s.t.$ S satisfies a predefined topological constraint

Subgraph detection: definition



Network topology



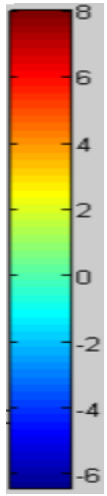


Constraint is defined based on network topology.

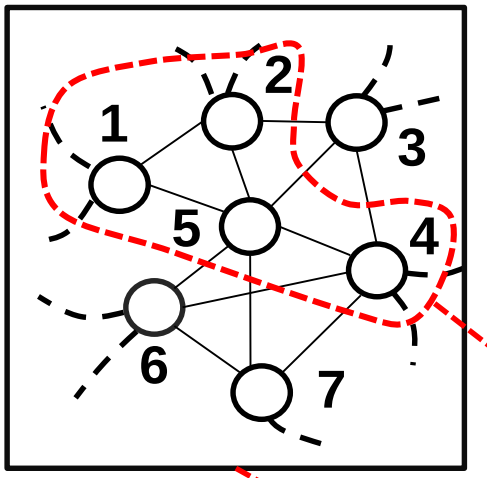
$$\max_{S,R,W} F(S,R,W)$$

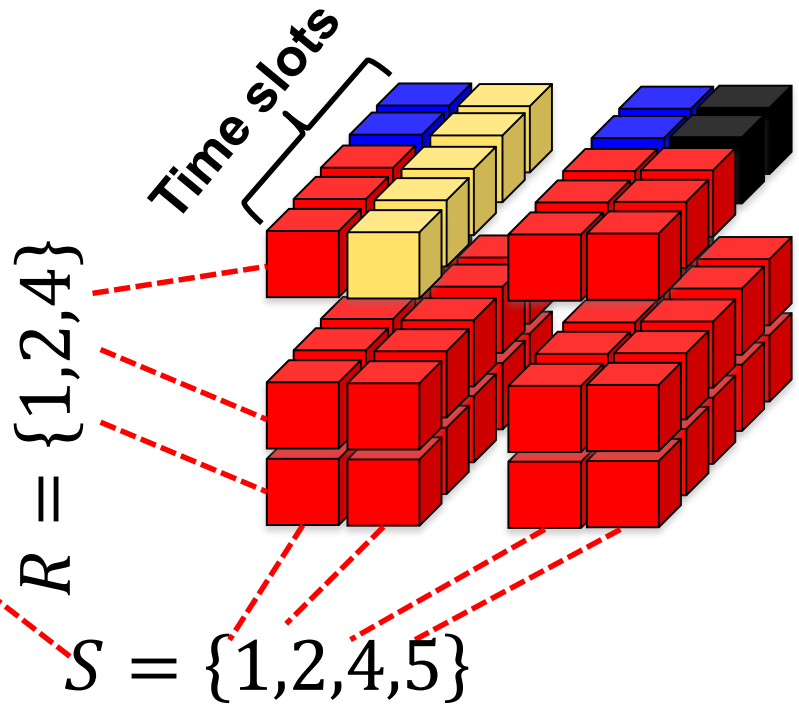
$s.t.$ S satisfies a predefined topological constraint

Subgraph detection: definition



Network topology





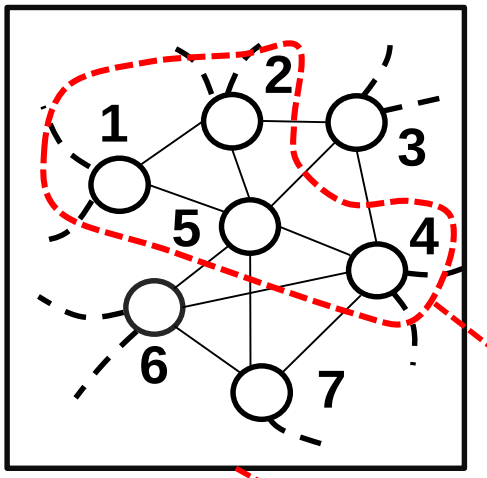
Constraint is defined based on network topology.

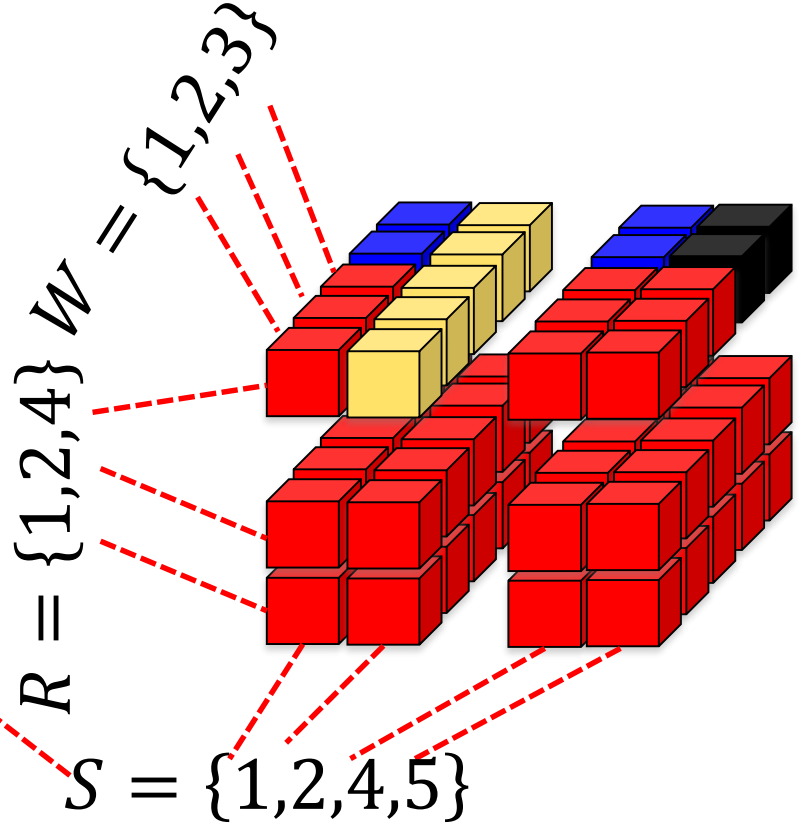
$$\max_{S,R,W} F(S,R,W)$$

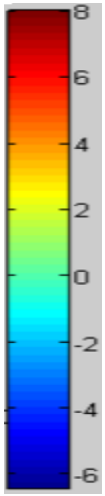
$s.t.$ S satisfies a predefined topological constraint

Subgraph detection: definition

Network topology



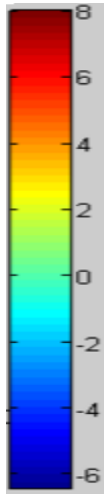




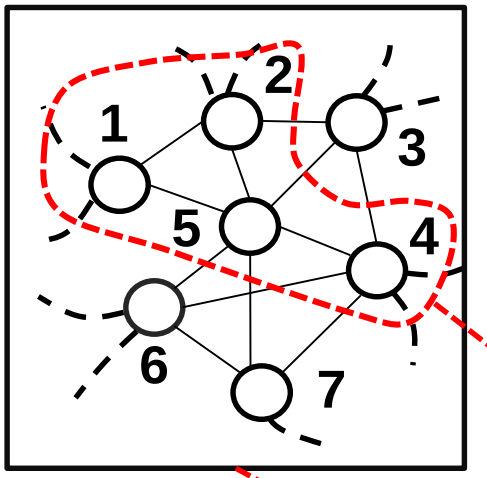
Constraint is defined based on network topology.

$$\max_{S,R,W} F(S,R,W)$$
$$s.t.\ S \text{ satisfies a predefined topological constraint}$$

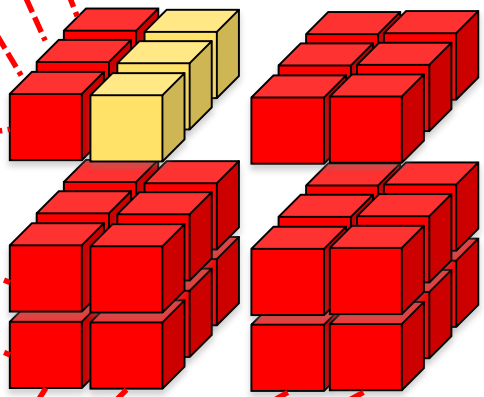
Subgraph detection: definition



Network topology



$R = \{1,2,4\}$ $W = \{1,2,3\}$



$S = \{1,2,4,5\}$

Constraint is defined based on network topology.

$$\max_{S,R,W} F(S,R,W)$$

$s.t.$ S satisfies a predefined topological constraint

Score function & constraints

- Score functions
  - Parametric scan statistics
    - Kulldorff's statistic, Expectation-based statistic
  - Nonparametric scan statistics
    - Higher Criticism (HC) statistic, Berk-Jones's statistic
  - Network design based functions
    - Prize Collecting Steiner Tree (PCST) objective
- Topological constraints
  - Regular shapes, such as circles and rectangles.
  - Connectivity (the focus of this tutorial)
  - Compactness

Computational Challenges

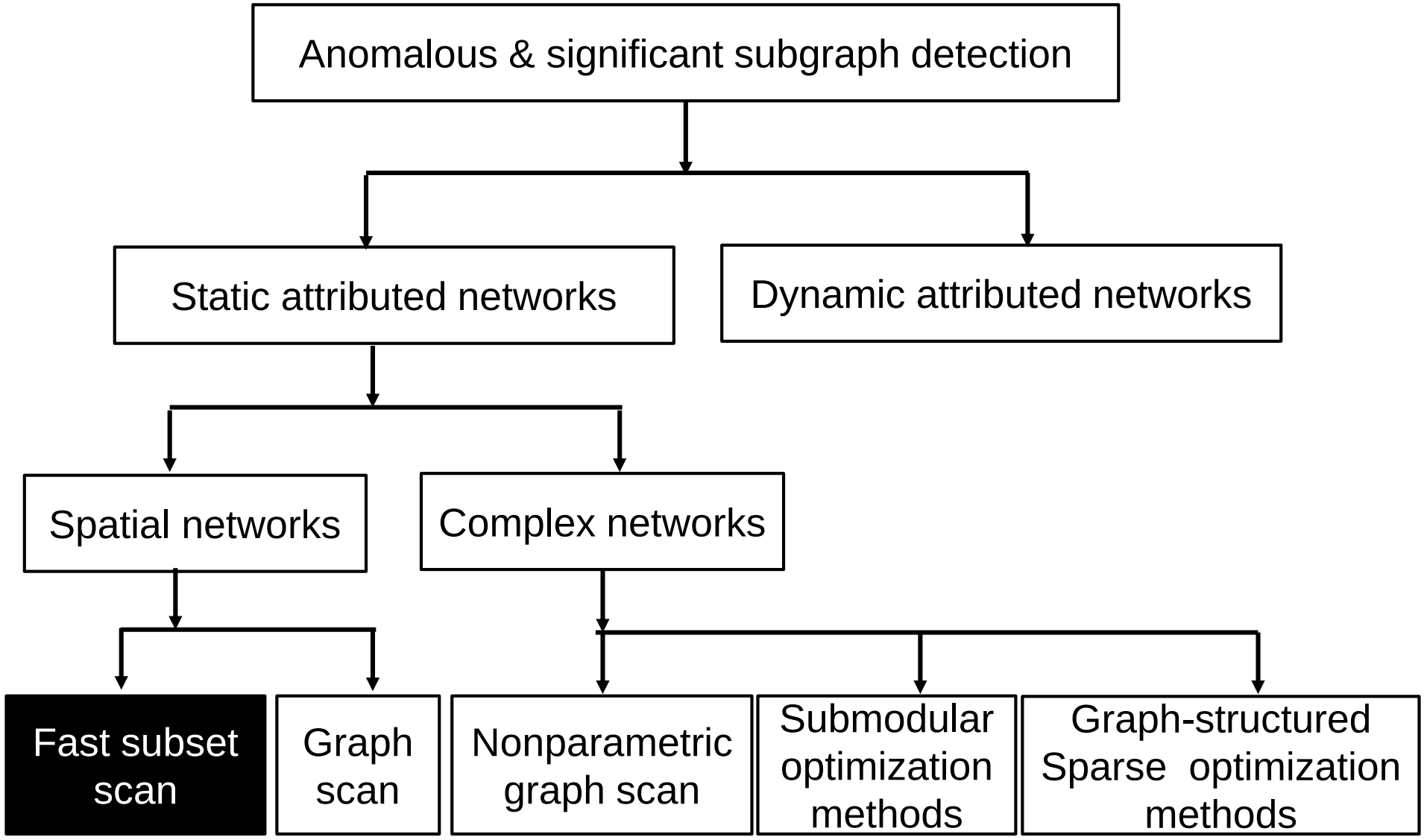
- Exponentially many possible subsets, $O(2^N \cdot 2^M)$, where N and M refer to the total numbers of nodes and attributes, respectively: computationally infeasible for naïve search.
- Given a score function and a topological constraint (e.g. connectivity) predefined by a user, how we can identify the highest scoring subgraphs efficiently and effectively?

Comparisons with related topics

- The unique aspect of this tutorial is that the focus is on detection of subgraph patterns that optimize certain structural and attribute properties (or constraints) in large attributed networks.
- In comparison, most relevant tutorials were focused on analysis of graph-level or node-level patterns in networks without attributes.
- Community detection and node embedding methods will not be reviewed in this tutorial.

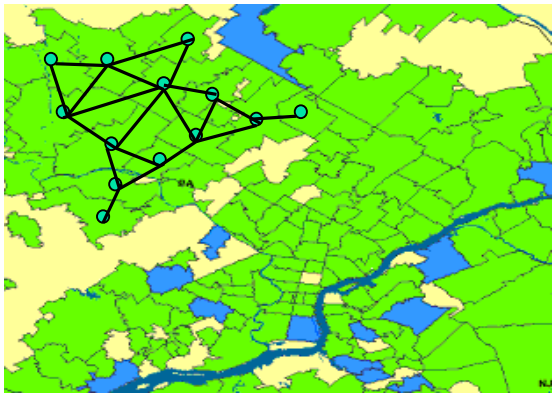
Part 1: Subgraph Detection in Static Attributed Networks

Taxonomy


Anomalous & significant subgraph detection
Static attributed networks
Dynamic attributed networks
Spatial networks
Complex networks
Fast subset scan
Graph scan
Nonparametric graph scan
Submodular optimization methods
Graph-structured Sparse optimization methods

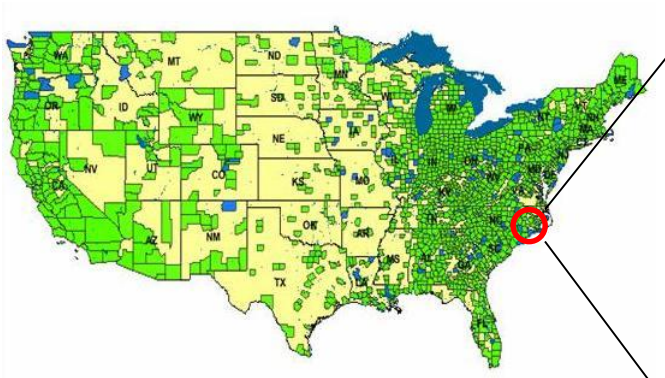
Detection in Spatial Networks

- Each graph node corresponds to the centroid of a small area (e.g., zip code or census tract), with corresponding lat/long coordinates.
- Edges are defined by spatial adjacency between areas.

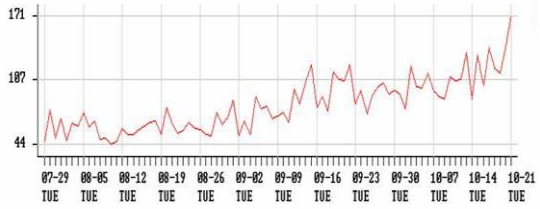


- Some quantities (e.g., number of crimes or disease cases) are monitored for each area → attributes of that node.
- Goal: find connected subgraph with collectively anomalous attribute values.
- Graph sizes tend to be relatively small (hundreds-thousands) but still far too large for exhaustive search over subgraphs.

Multivariate event detection



Spatial time series data from spatial locations s_i (e.g. zip codes)



Time series of counts $c_{i,m}{}^t$ for each zip code s_i for each data stream d_m.

Outbreak detection

d_1 = respiratory ED

d_2 = constitutional ED

d_3 = OTC cough/cold

d_4 = OTC anti-fever

(etc.)

Main goals:

Detect any emerging events.

Pinpoint the affected subset of locations and time duration.

Characterize the event by identifying the affected streams.

Compare hypotheses:

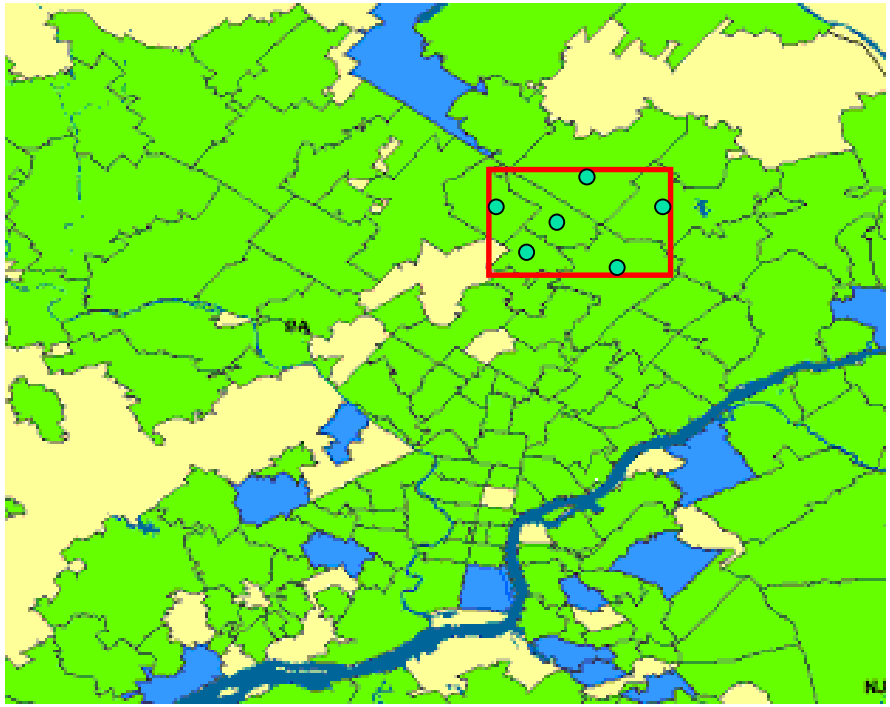
$H_1(D, S, W)$

D = subset of streams
S = subset of locations
W = time duration

vs. H_0: no events occurring

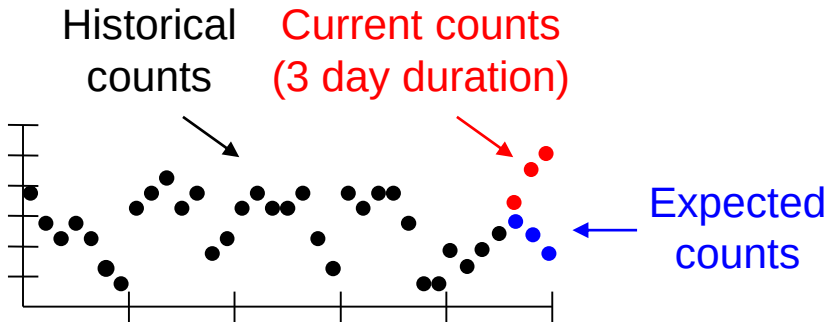
Expectation-based scan statistics

(Kulldorff, 1997; Neill and Moore, 2005)



We search for spatial regions (subsets of locations) where the recently observed counts for some subset of streams are significantly higher than expected.

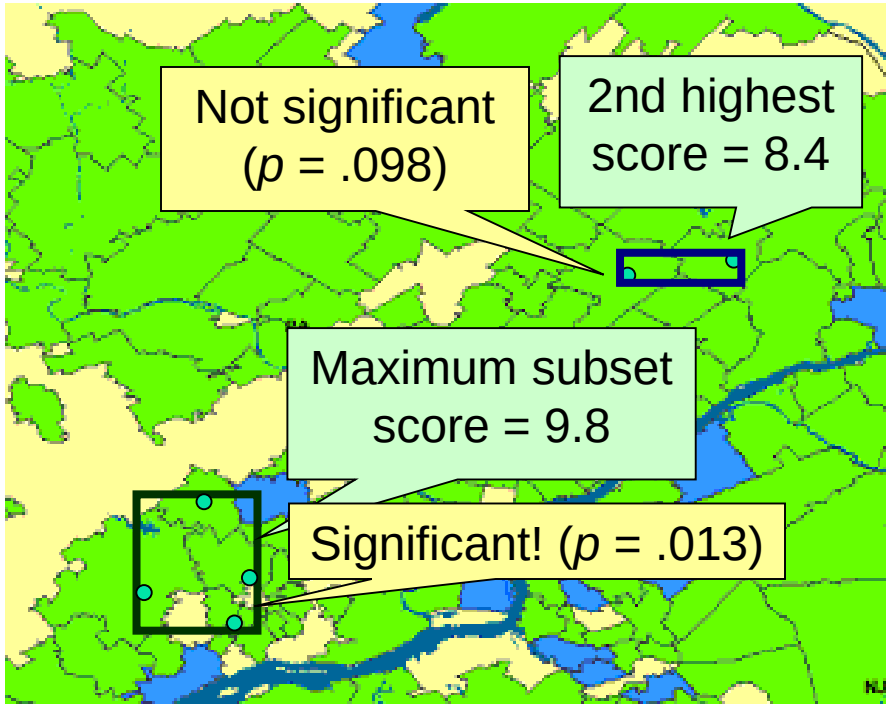
We perform **time series analysis** to compute expected counts ("baselines") for each location and stream for each recent day.



We then compare the actual and expected counts for each subset (D, S, W) under consideration.

Expectation-based scan statistics

(Kulldorff, 1997; Neill and Moore, 2005)

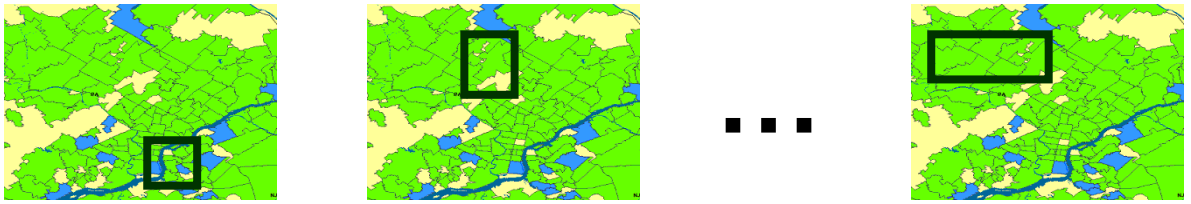


We find the subsets with highest values of a likelihood ratio statistic, and compute the *p*-value of each subset by randomization testing.

$$F(D,S,W) = \frac{\Pr(\text{Data} \mid H_1(D,S,W))}{\Pr(\text{Data} \mid H_0)}$$

<u>To compute p-value</u>
Compare subset score to maximum subset scores of simulated datasets under H_0.

$F_1^* = 2.4$ $F_2^* = 9.1$ ... $F_{999}^* = 7.0$



Which regions to search?

Typical approach: "spatial scan" (Kulldorff, 1997)

Each search region S is a **sub-region** of space.

- Choose some region shape (e.g. circles, rectangles) and consider all regions of that shape and varying size.
- Low power for true events that do not correspond well to the chosen set of search regions (e.g. irregular shapes).

Our approach: "subset scan" (Neill, 2012)

Each search region S is a **subset** of locations.

- Find the highest scoring subset, subject to some constraints (e.g. spatial proximity, connectivity).
- For multivariate, also optimize over subsets of streams.
- Exponentially many possible subsets, $O(2^N \times 2^M)$: computationally infeasible for naïve search.

Fast subset scan

- In certain cases, we can optimize F(S) over the exponentially many subsets of the data, while evaluating only O(N) rather than $O(2^N)$ subsets.
- Many commonly used scan statistics have the property of linear-time subset scanning:
  - Just sort the data records (spatial locations, etc.) from highest to lowest priority according to some function…
  - … then search over groups consisting of the top-k highest priority records, for k = 1..N.

The highest scoring subset is **guaranteed** to be one of these!

Sample result: we can find the **most anomalous** subset of Allegheny County zip codes in 0.03 sec vs. 10^{24} years.

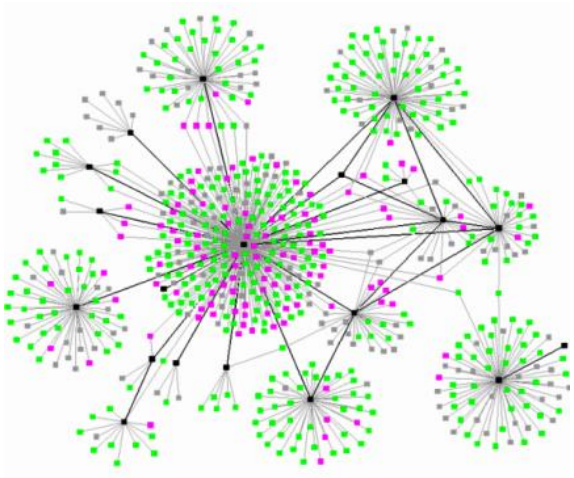
Fast subset scan with spatial proximity constraints

- Maximize a likelihood ratio statistic over all subsets of the "local neighborhoods" consisting of a center location s_i and its k-1 nearest neighbors, for a fixed neighborhood size k.
- Naïve search requires $O(N \cdot 2^k)$ time and is computationally infeasible for $k > 25$.
- For each center, we search over all subsets of its local neighborhood in O(k) time using LTSS, thus requiring a total time of O(Nk) + O(N log N) for sorting the locations.
- In Neill (2012), we show that this approach dramatically improves the timeliness and accuracy of outbreak detection for irregularly-shaped disease clusters.

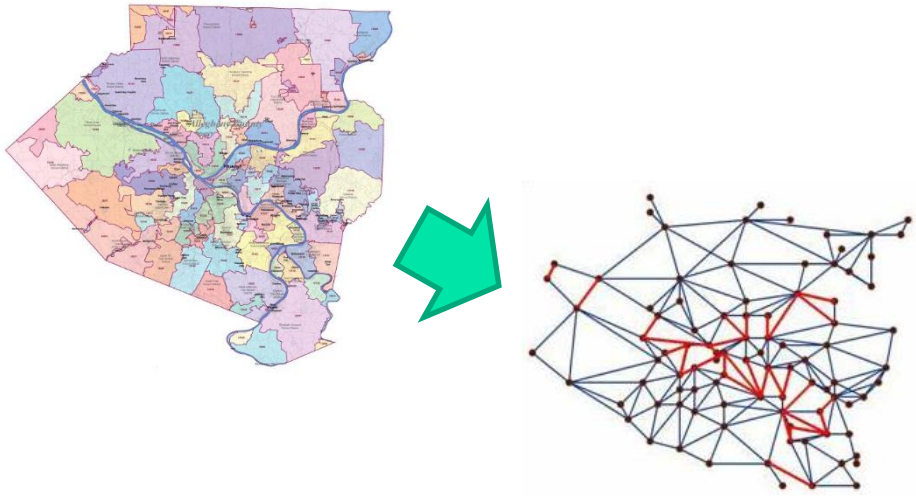
Incorporating connectivity constraints

Proximity-constrained subset scans may return a disconnected subset of the data.

In some cases this may be undesirable, or we might have non-spatial data so proximity constraints cannot be used.

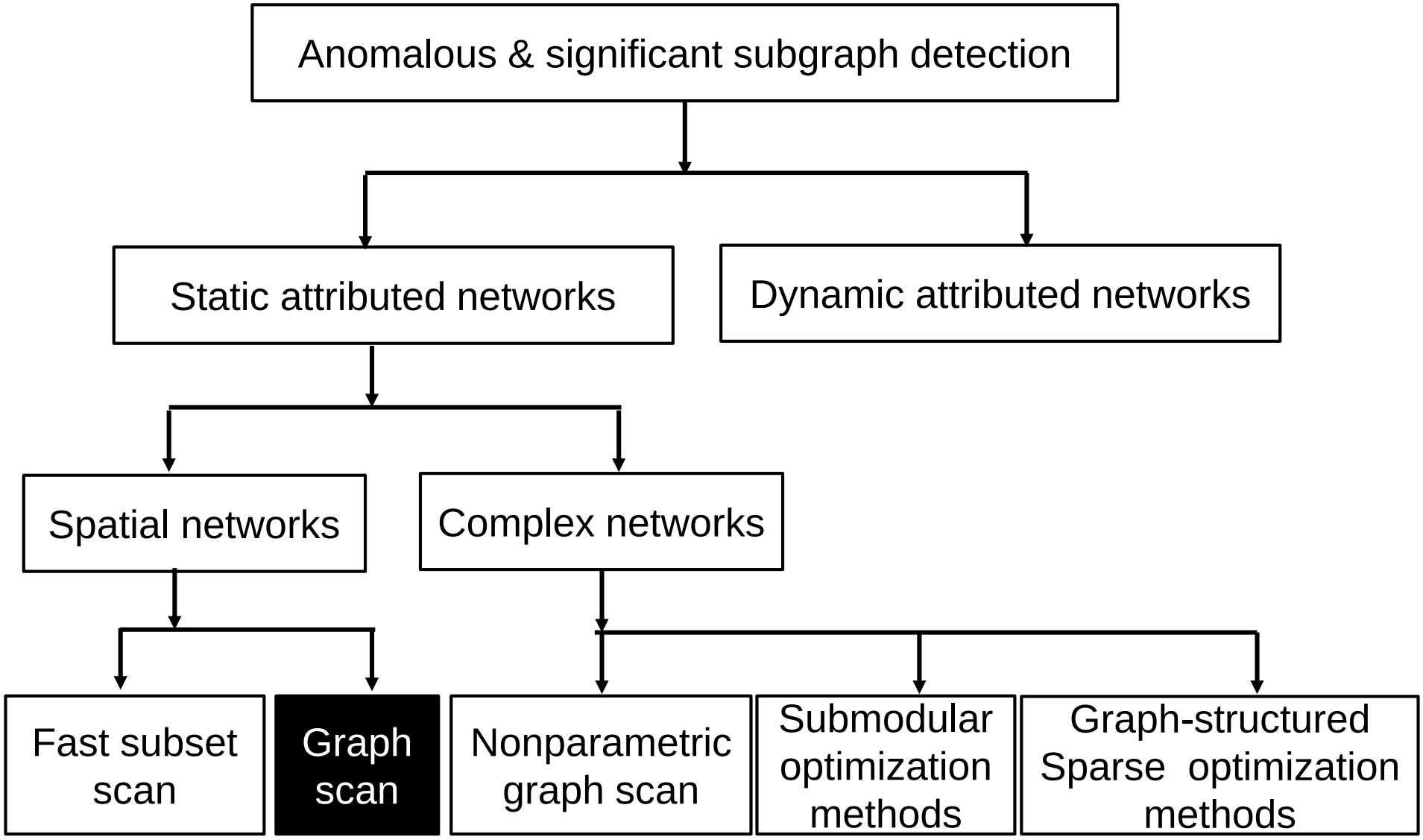


Example: tracking disease spread from person-to-person contact.



Example: identifying a **connected** subset of zip codes (Allegheny County, PA)

Taxonomy


Anomalous & significant subgraph detection
Static attributed networks
Dynamic attributed networks
Spatial networks
Complex networks
Fast subset scan
Graph scan
Nonparametric graph scan
Submodular optimization methods
Graph-structured Sparse optimization methods

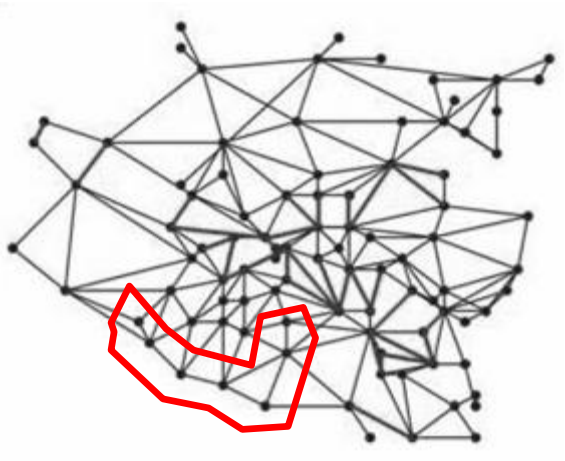
Incorporating connectivity constraints

Proximity-constrained subset scans may return a disconnected subset of the data.

In some cases this may be undesirable, or we might have non-spatial data so proximity constraints cannot be used.

Our **GraphScan** algorithm* can efficiently and exactly identify the highest-scoring connected subgraph:

- Can incorporate multiple data streams
- With or without proximity constraints
- Graphs with several hundred nodes



We can use the LTSS property to rule out subgraphs that are provably suboptimal, dramatically reducing our search space.

*Speakman, McFowland, Neill. Scalable detection of anomalous patterns with connectivity constraints. *J Comput Graph Stat* 24(4): 1014-1033, 2015.

Incorporating connectivity constraints

We represent groups of subsets as strings of 0's, 1's, and ?'s.

Assume that the graph nodes are sorted from highest priority to lowest priority.

Priority Ranking	1	2	3	4	5	6
Bit String	1	0	0	1	?	?

The above bit string represents four possible subsets: {1,4}, {1,4,5}, {1,4,6}, and {1,4,5,6}.

LTSS property <u>without</u> connectivity constraints:
"If node x ∈ S and node y ∉ S, for x > y,
then subset S cannot be optimal."

We can use the LTSS property to rule out subgraphs that are provably suboptimal, dramatically reducing our search space.

Incorporating connectivity constraints

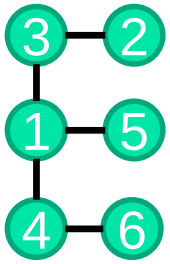
We represent groups of subsets as strings of 0's, 1's, and ?'s.

Assume that the graph nodes are sorted from highest priority to lowest priority.

Priority Ranking	1	2	3	4	5	6
Bit String	1	0	0	1	?	?

The above bit string represents four possible subsets: {1,4}, {1,4,5}, {1,4,6}, and {1,4,5,6}.

LTSS property with connectivity constraints:
"If node x ∈ S and node y ∉ S, for x > y,
and S \ {x} and S U {y} are both connected,
then subset S cannot be optimal."



We can use the LTSS property to rule out subgraphs that are provably suboptimal, dramatically reducing our search space.

Incorporating connectivity constraints

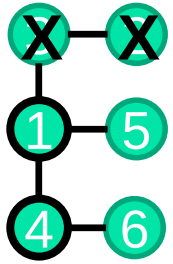
We represent groups of subsets as strings of 0's, 1's, and ?'s.

Assume that the graph nodes are sorted from highest priority to lowest priority.

Priority Ranking	1	2	3	4	5	6
Bit String	1	0	0	1	?	?

The above bit string represents four possible subsets: {1,4}, {1,4,5}, {1,4,6}, and {1,4,5,6}.

LTSS property with connectivity constraints:
"If node x ∈ S and node y ∉ S, for x > y,
and S \ {x} and S U {y} are both connected,
then subset S cannot be optimal."



suboptimal

We can use the LTSS property to rule out subgraphs that are provably suboptimal, dramatically reducing our search space.

Incorporating connectivity constraints

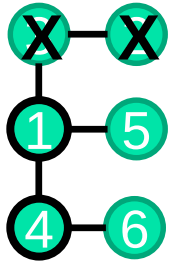
We represent groups of subsets as strings of 0's, 1's, and ?'s.

Assume that the graph nodes are sorted from highest priority to lowest priority.

Priority Ranking	1	2	3	4	5	6
Bit String	1	0	0	1	?	?

The above bit string represents four possible subsets: {1,4}, {1,4,5}, {1,4,6}, and {1,4,5,6}.

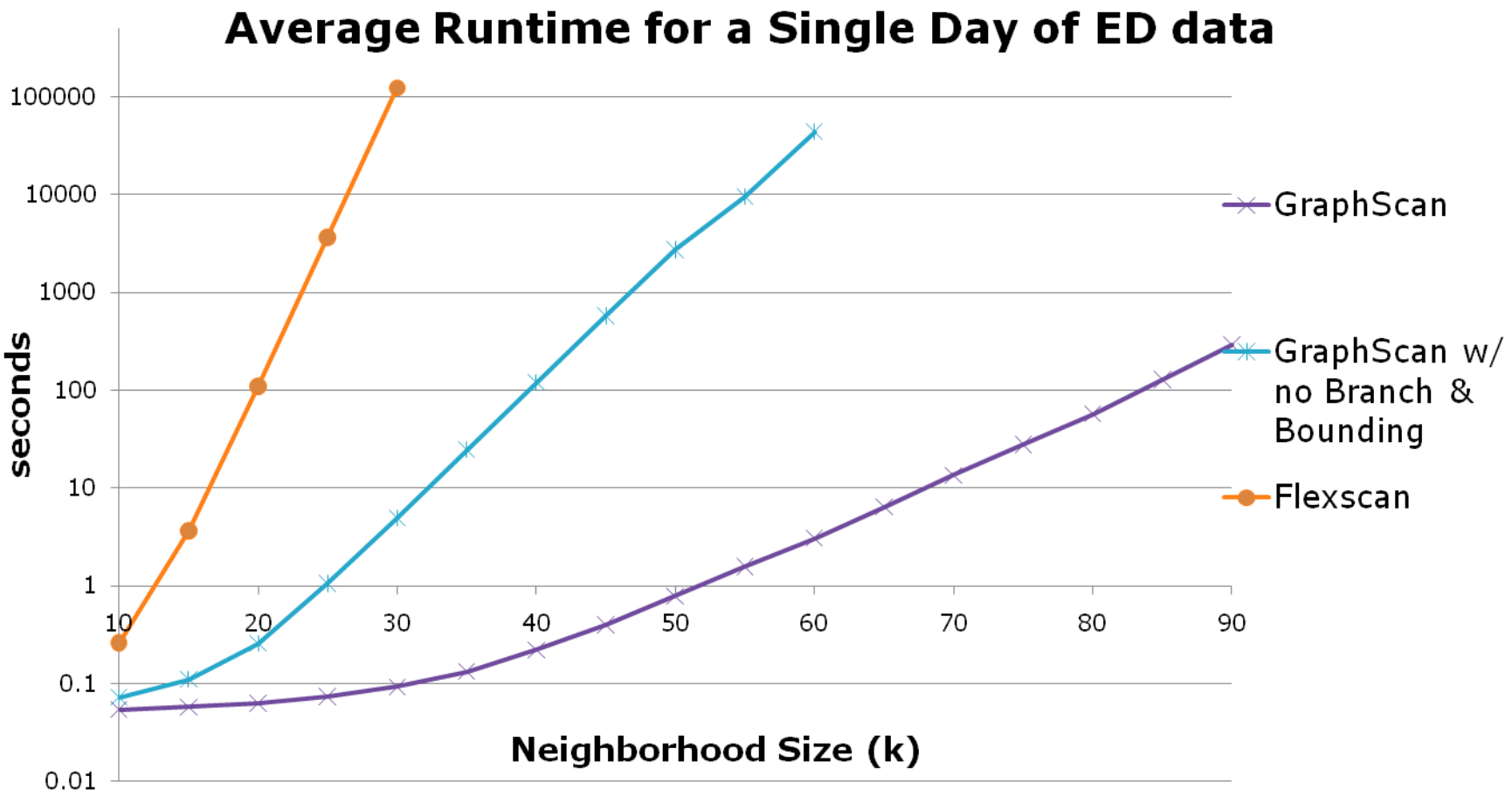
LTSS property with connectivity constraints:
"If node x ∈ S and node y ∉ S, for x > y,
and S \ {x} and S U {y} are both connected,
then subset S cannot be optimal."



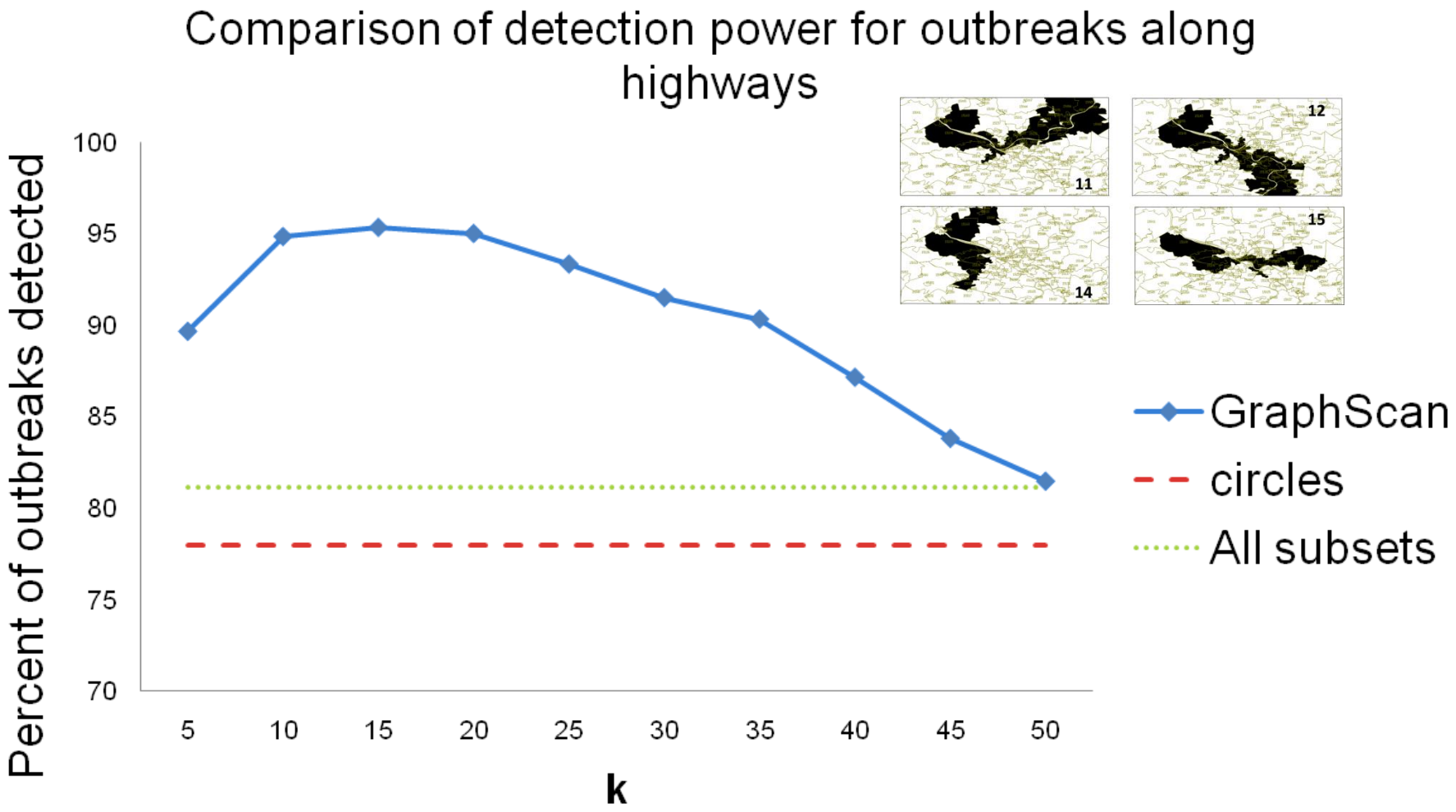
suboptimal

Additional speedups can be gained by **branch-and-bounding**: we use the unconstrained subset score as an upper bound on the connected subgraph score, and rule out subsets which cannot be higher-scoring than the best subset found so far.

Evaluation: run times



Evaluation: detection power



Extensions of GraphScan

What if we want to allow for events which spread dynamically over the (static) graph structure?

Based on a new variant of the LTSS property[1], we can search for dynamic patterns while enforcing soft constraints on **temporal consistency.**

We have applied this method for accurate detection, tracking, and source-tracing of contaminants spreading through a water distribution network.[2]

What if the underlying graph structure is unknown?

We can accurately **learn** the graph structure from unlabeled outbreak data, and use the learned structure for detection.

Often, the learned graph enables even faster detection of events than the true graph![3]

[1]Speakman, Somanchi, McFowland, and Neill. Penalized fast subset scanning. *J Comput Graph Stat* 25(2): 382-404, 2016.

[2]Speakman, Zhang, Neill. Dynamic pattern detection with temporal consistency and connectivity constraints. *Proc. ICDM 2013.*

[3]Somanchi and Neill, submitted.

Variants of GraphScan

Previous exact approaches are very slow…

FlexScan (Tango & Takahashi, 2005): exhaustive search over connected subgraphs within each spatial neighborhood, infeasible for k > 25.

Contiguous Max-LLR model (Murray et al., 2014): requires solving many mixed integer linear programs, exponentially many in worst case.

… but a variety of heuristic approaches exist.

Duczmal et al.: simulated annealing, genetic algorithms

Assuncao et al.: spanning trees

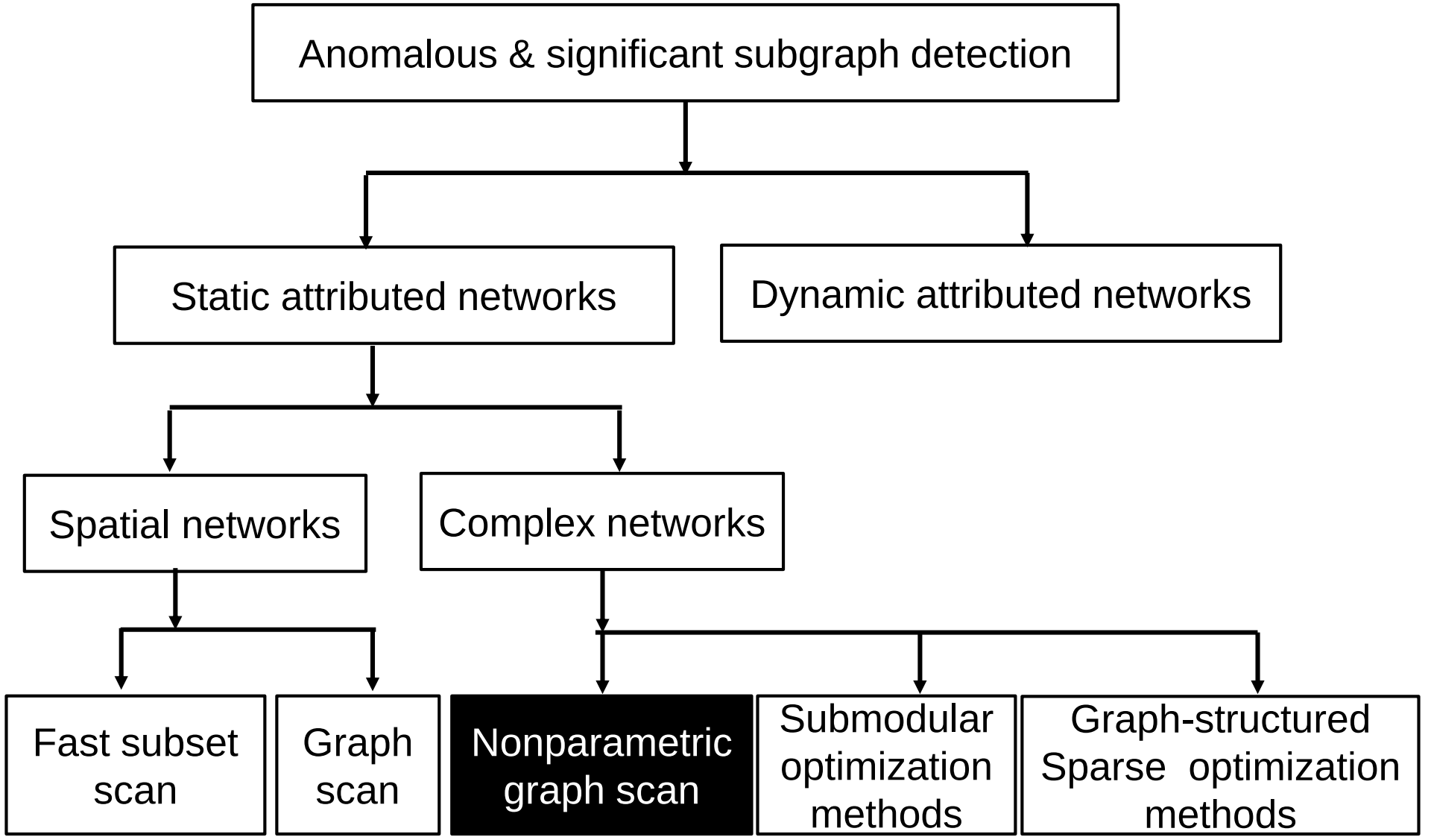
Chen and Neill: greedy growth

Speakman et al.: additive GraphScan

1) Construct conditionally additive score function.

2) Optimizing F(S) reduces to maximum weight connected subgraph problem.

Taxonomy


Anomalous & significant subgraph detection
Static attributed networks
Dynamic attributed networks
Spatial networks
Complex networks
Fast subset scan
Graph scan
Nonparametric graph scan
Submodular optimization methods
Graph-structured Sparse optimization methods

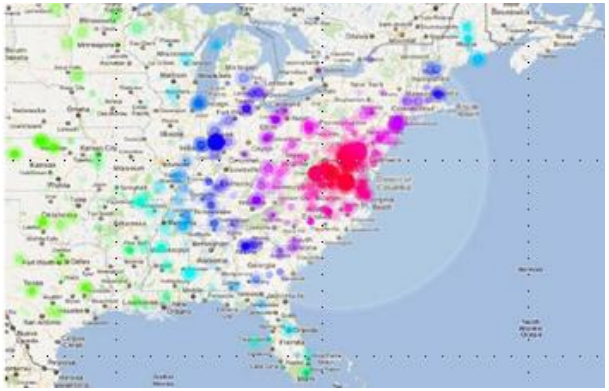
Event Detection from Social Media

(Chen and Neill, KDD 2014)

Protest in Mexico, 7/14/2012

2012 Washington D.C. Traffic

Tweet Map for 2011 VA Earthquake



Social media is a real-time "sensor" of large-scale population behavior, and can be used for early detection of emerging events...

… but it is very complex, noisy, and subject to biases.

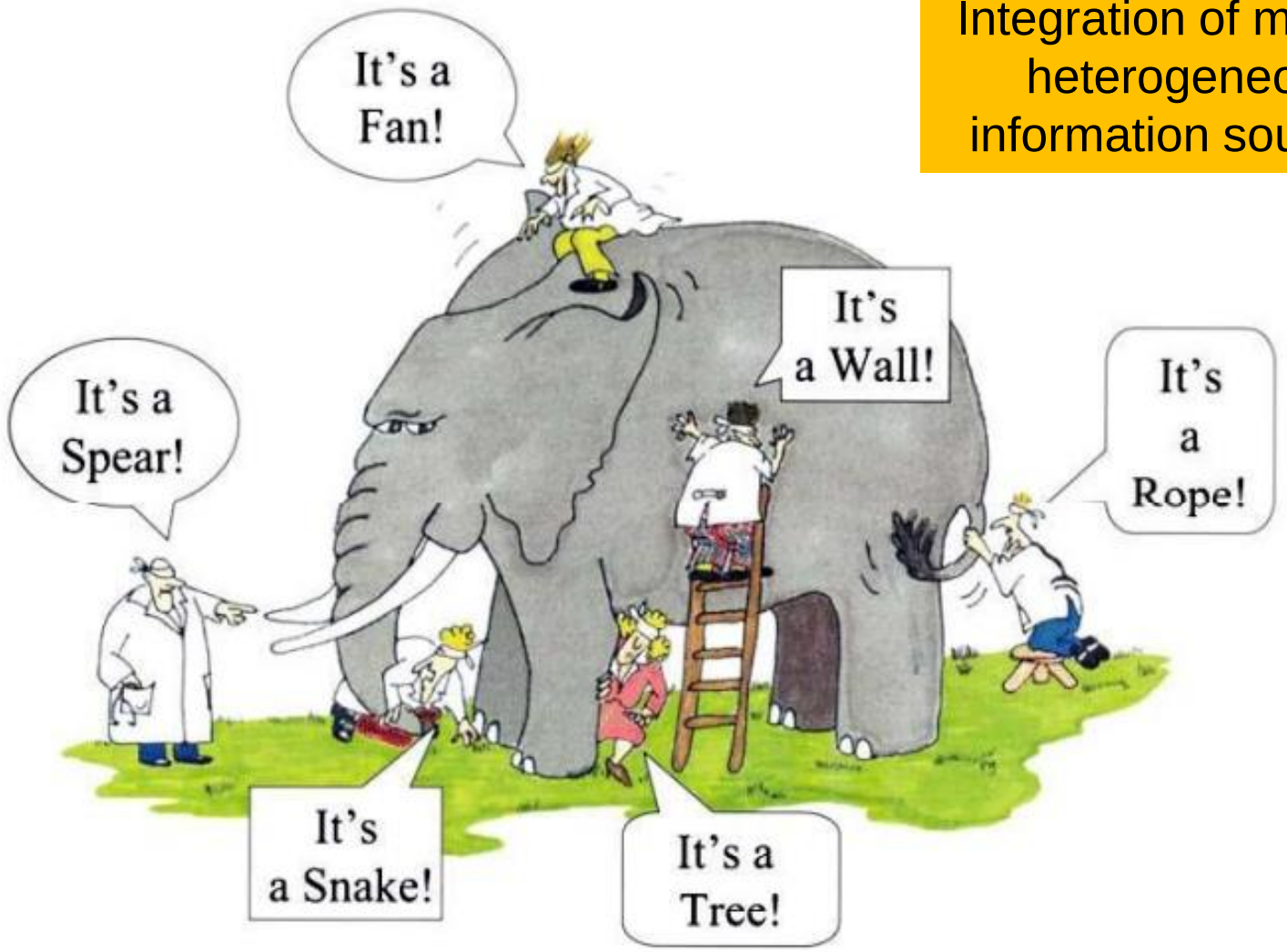
We have developed a new event detection methodology:

"Non-Parametric Heterogeneous Graph Scan" (NPHGS)

Applied to: civil unrest prediction, rare disease outbreak detection, and early detection of human rights events.

Technical Challenges

Integration of multiple heterogeneous information sources!



Technical Challenges

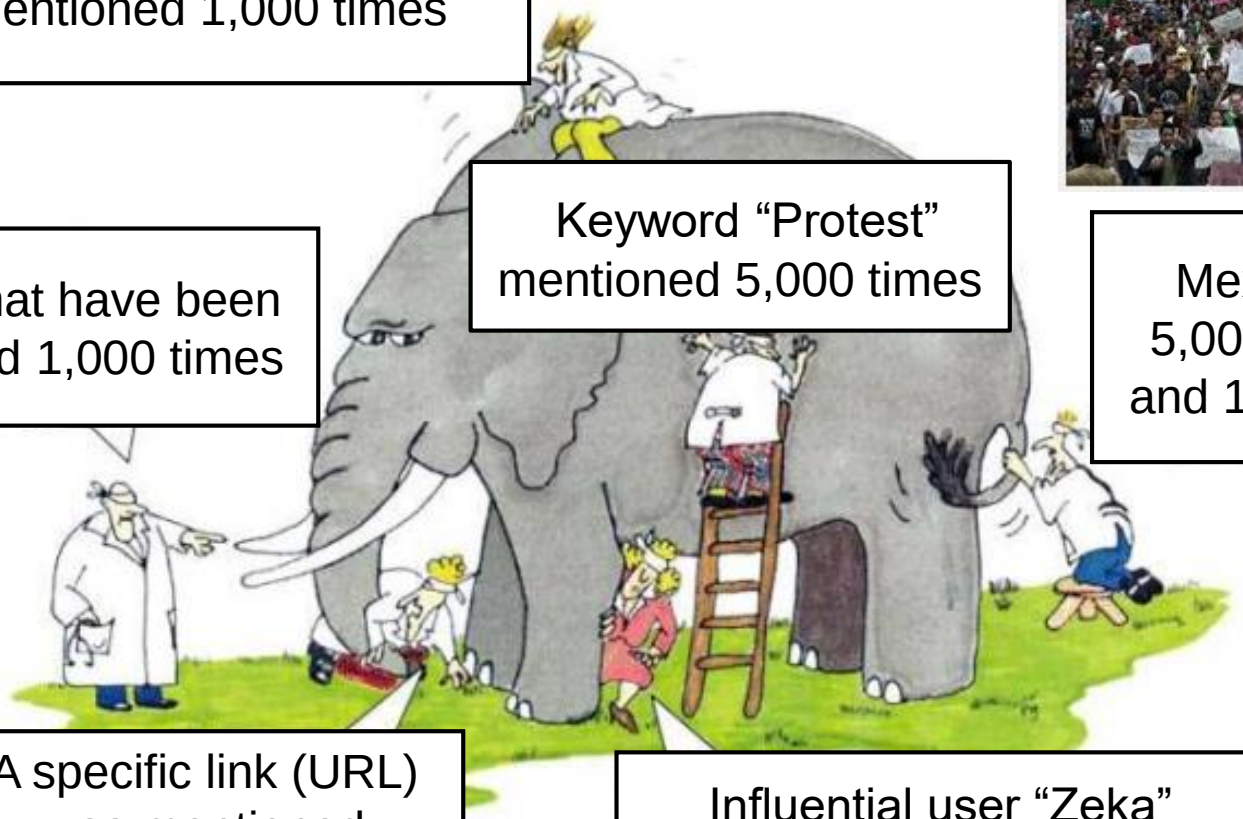
One week before Mexico's 2012 presidential election:

Hashtag "#Megamarch" mentioned 1,000 times

Keyword "Protest" mentioned 5,000 times

Tweets that have been re-tweeted 1,000 times

Mexico City has 5,000 active users and 100,000 tweets



A specific link (URL) was mentioned 866 times

Influential user "Zeka" posted 10 tweets

Technical Challenges

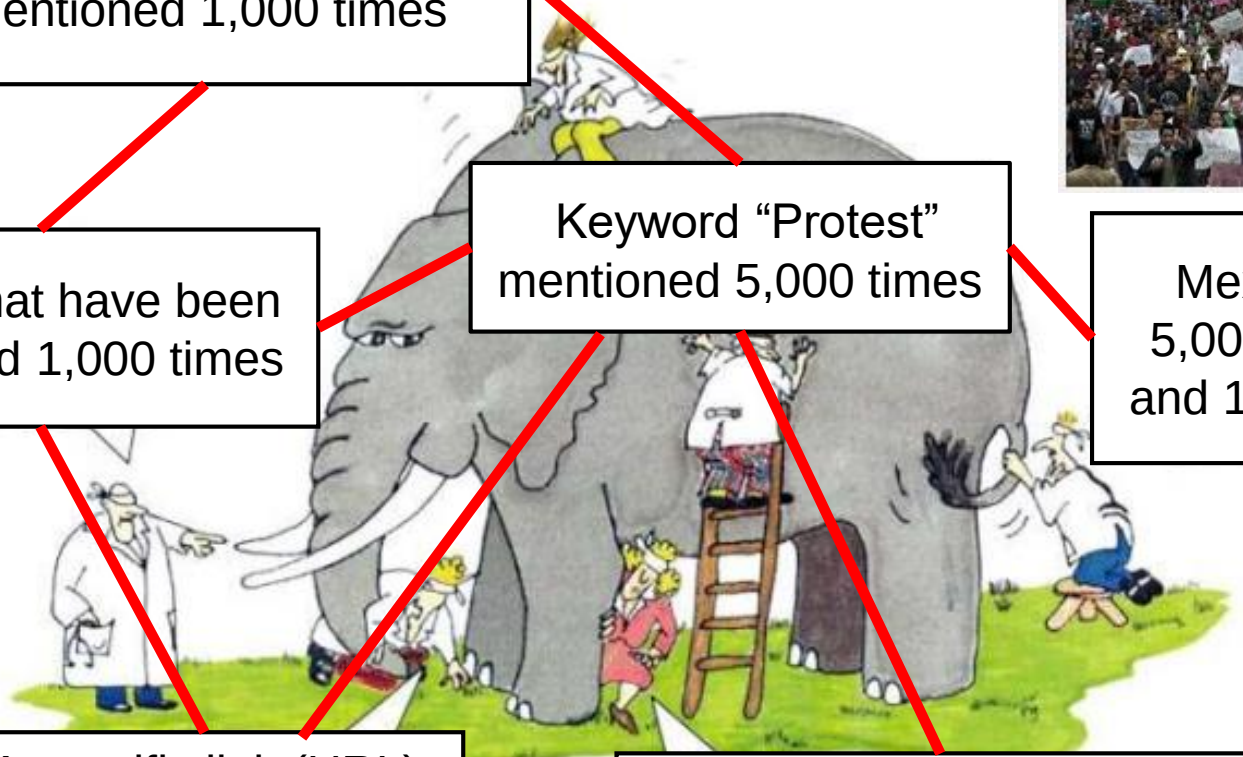
One week before Mexico's 2012 presidential election:

Hashtag "#Megamarch" mentioned 1,000 times

Keyword "Protest" mentioned 5,000 times

Tweets that have been re-tweeted 1,000 times

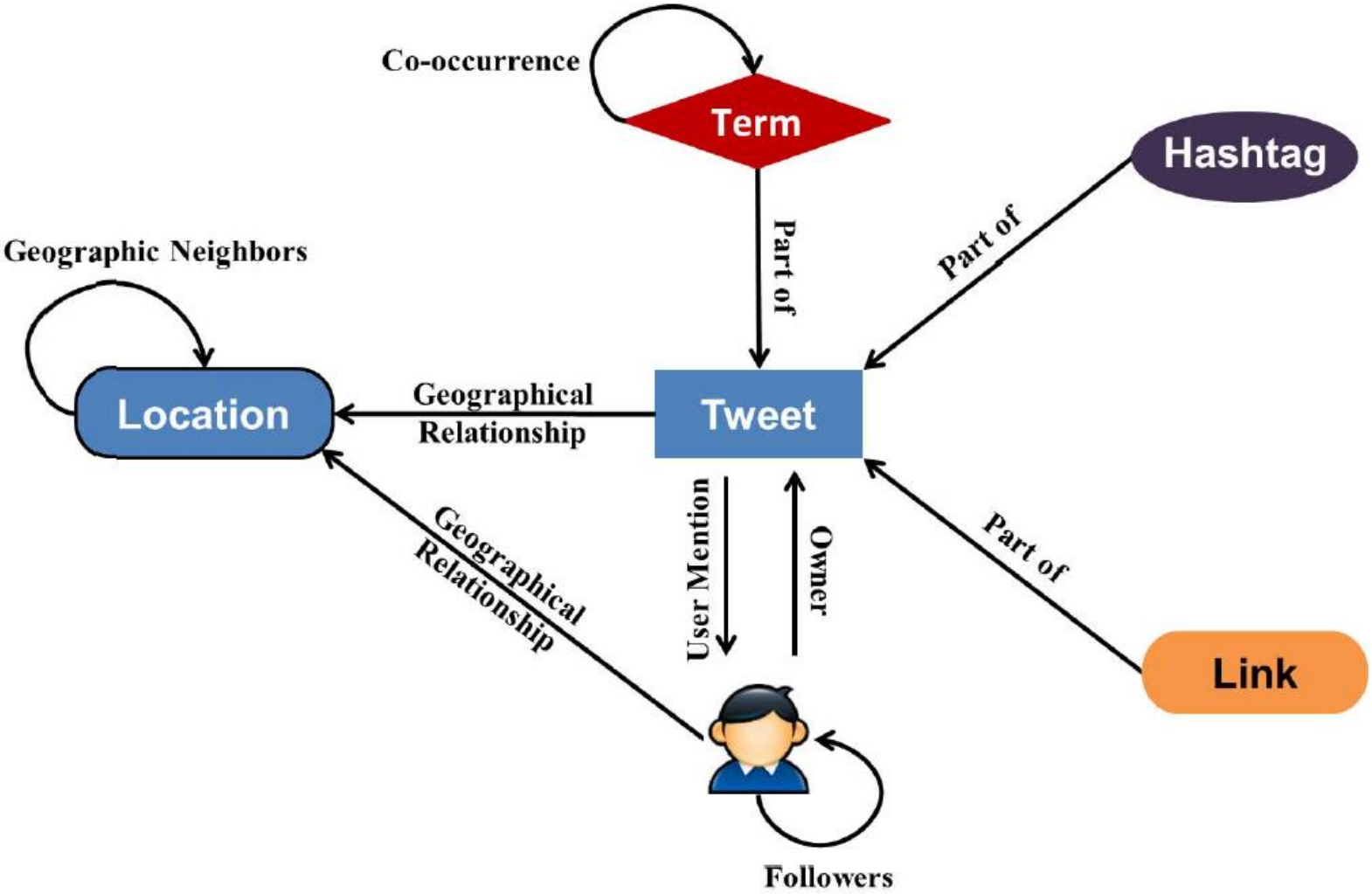
Mexico City has 5,000 active users and 100,000 tweets



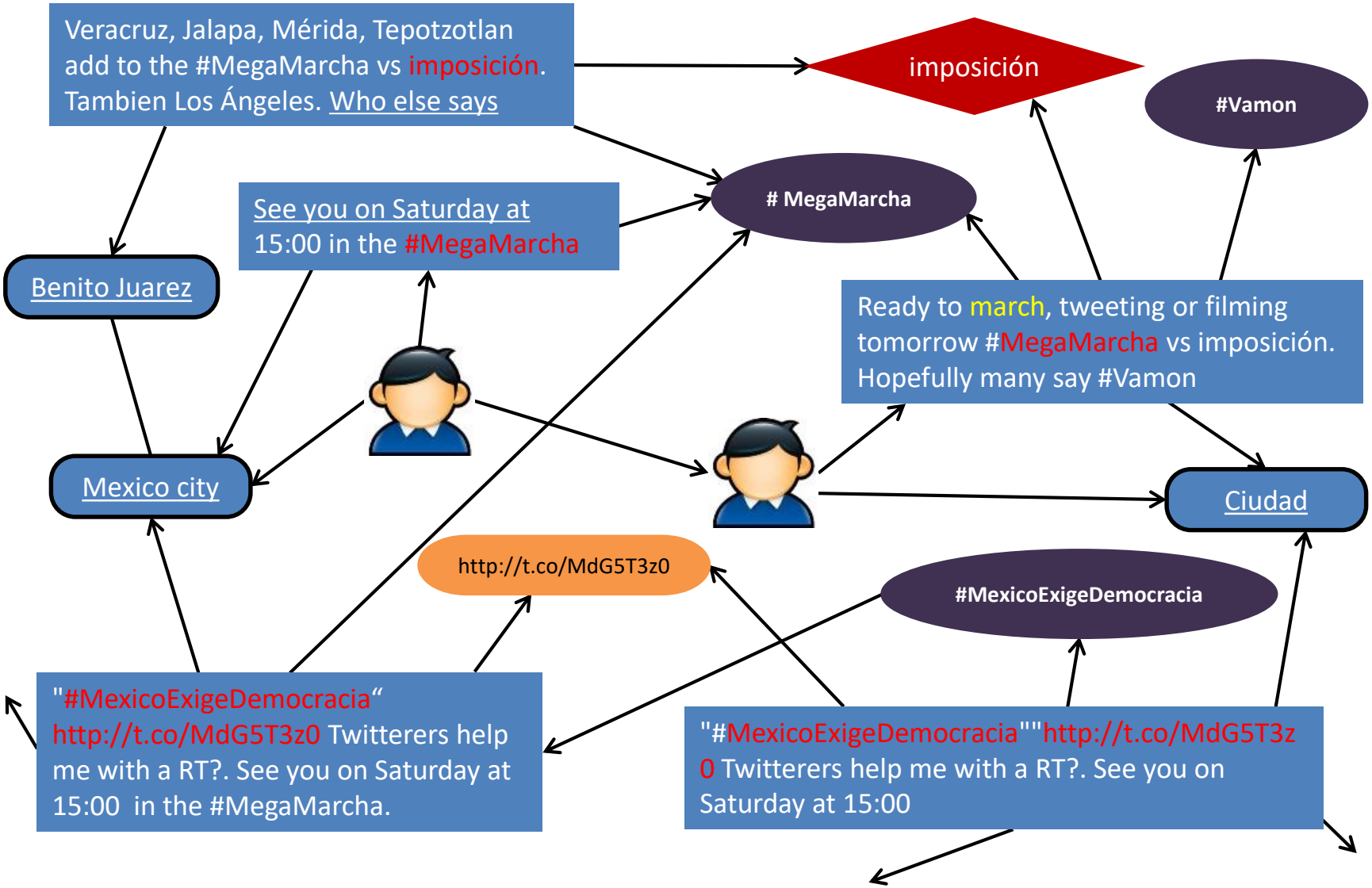
A specific link (URL) was mentioned 866 times

Influential user "Zeka" posted 10 tweets

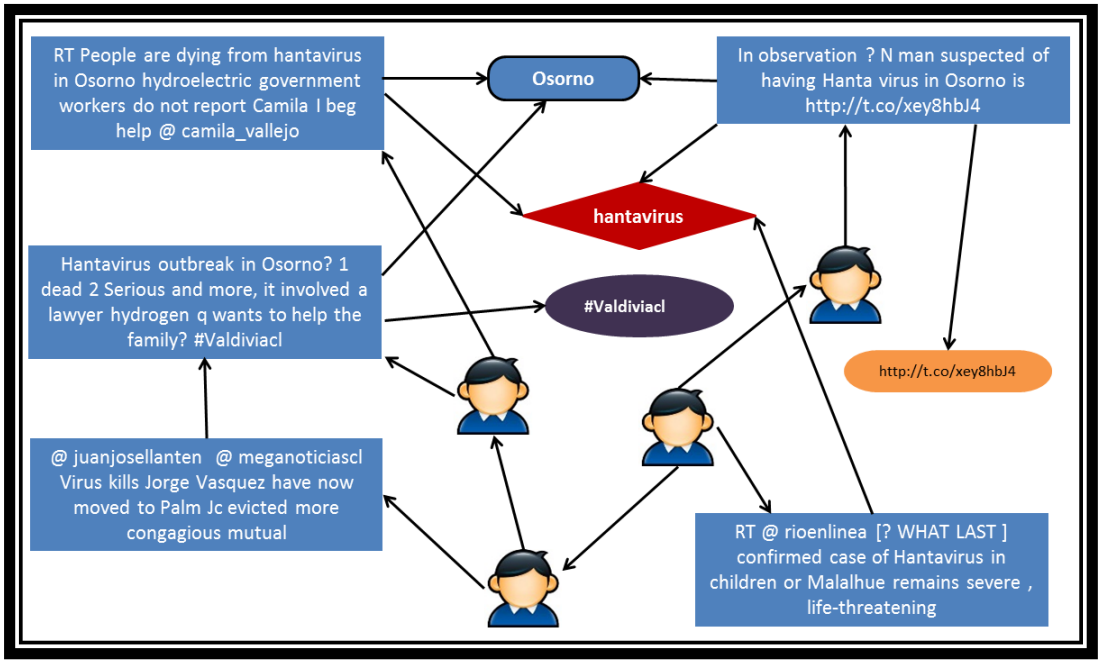
Twitter Heterogeneous Network

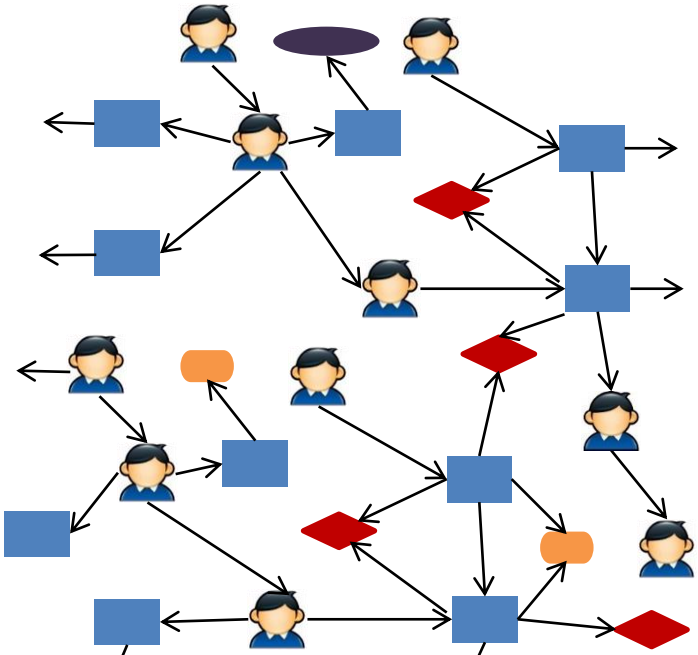


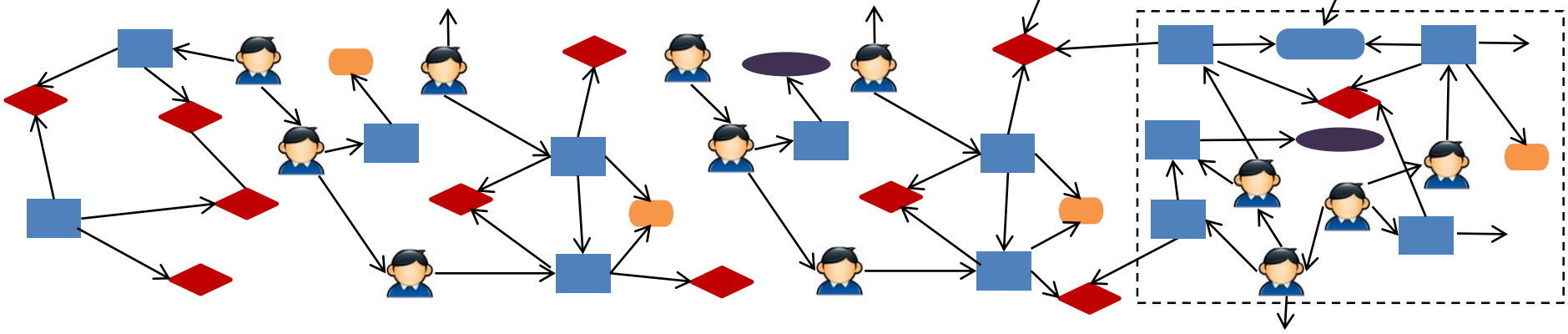
Twitter Heterogeneous Network



Twitter Heterogeneous Network







Nonparametric Heterogeneous Graph Scan

(Chen and Neill, KDD 2014)

1) We model the heterogeneous social network as a **sensor network**.

   Each node senses its local neighborhood, computes multiple features, and reports the overall degree of anomalousness.

2) We compute an **empirical p-value** for each node:

   - Uniform on [0,1] under the null hypothesis of no events.
   - We search for subgraphs of the network with a higher than expected number of low (significant) empirical p-values.

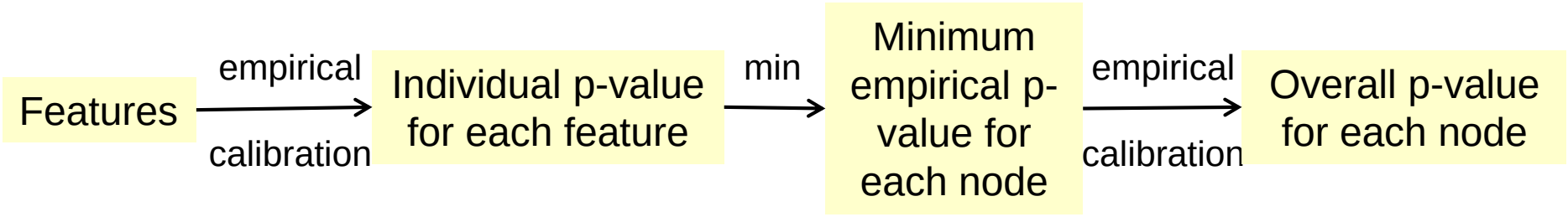
3) We can scale up to very large heterogeneous networks:

   - Heuristic approach: **iterative subgraph expansion** ("greedy growth" to subset of neighbors on each iteration).
   - We can efficiently find the best subset of neighbors, ensuring that the subset remains connected, at each step.

Sensor network modeling

Each node reports an empirical p-value measuring the current level of anomalousness for each time interval (hour or day).

Object Type	Features
User	# tweets, # retweets, # followers, #followees, #mentioned_by, #replied_by, diffusion graph depth, diffusion graph size
Tweet	Klout, sentiment, replied_by_graph_size, reply_graph_size, retweet_graph_size, retweet_graph_depth
City, State, Country	# tweets, # active users
Term	# tweets
Link	# tweets
Hashtag	# tweets



Nonparametric scan statistics

Number of nodes in S with p-values ≤α.

Subgraph

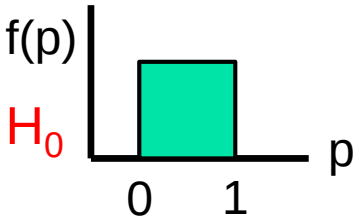
$$F(S) = \max_{\alpha \le \alpha_{max}} F_\alpha(S) = \max_{\alpha \le \alpha_{max}} \phi(\alpha, N_\alpha(S), N(S))$$

Significance level

Number of nodes in S

Berk-Jones (BJ) statistic:

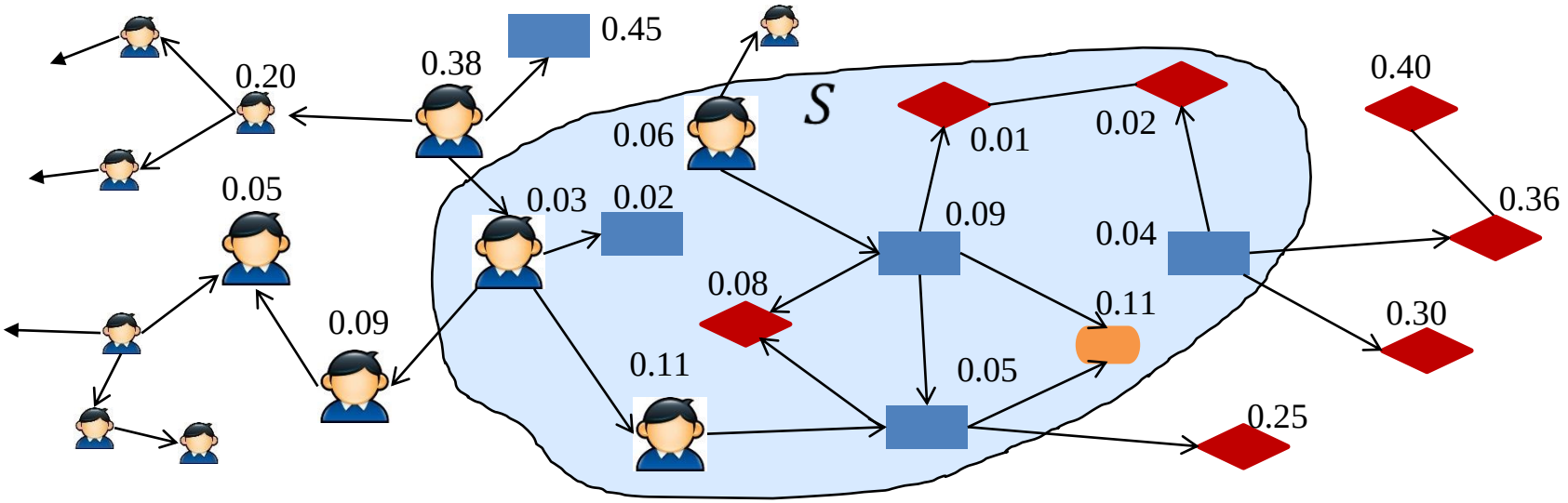
$$\phi_{BJ}(\alpha, N_\alpha(S), N(S)) = N(S) K\left(\frac{N_\alpha}{N}, \alpha\right)$$



Kullback-Liebler divergence:

$$K(x, y) = x \log\left(\frac{x}{y}\right) + (1 - x) \log\left(\frac{1 - x}{1 - y}\right)$$

Nonparametric graph scanning



$$S^{\star} = \underset{S \in V:\, S \text{ is connected}}{\operatorname{argmax}} F(S)$$

We propose an approximate algorithm with time cost O(|V| log |V|).

NPHGS evaluation- civil unrest

Country	# of tweets	News source*
Argentina	29,000,000	Clarín; La Nación; Infobae
Chile	14,000,000	La Tercera; Las Últimas Notícias; El Mercurio
Colombia	22,000,000	El Espectador; El Tiempo; El Colombiano
Ecuador	6,900,000	El Universo; El Comercio; Hoy

Gold standard dataset: 918 civil unrest events between July and December 2012.

Example of a gold standard event label:

PROVINCE = “El Loa” COUNTRY = “Chile”
DATE = “2012-05-18” LINK = “http://www.pressenza.com/2012/05/...”
DESCRIPTION = “A large-scale march was staged by inhabitants of the northern city of Calama, considered the mining capital of Chile, who demanded the allocation of more resources to copper mining cities”

We compared the detection performance of our NPHGS approach to homogeneous graph scan methods and to a variety of state-of-the-art methods previously proposed for Twitter event detection.

NPHGS results- civil unrest

Method	FPR (FP/Day)	TPR (Forecasting)	TPR (Forecasting & Detection)	Lead Time (Days)	Lag Time (Days)	Run Time (Hours)
ST Burst Detection	0.65	0.07	0.42	1.10	4.57	30.1
Graph Partition	0.29	0.03	0.15	0.59	6.13	18.9
Earthquake	0.04	0.06	0.17	0.49	5.95	18.9
RW Event	0.10	0.22	0.25	0.93	5.83	16.3
Geo Topic Modeling	0.09	0.06	0.08	0.01	6.94	9.7
NPHGS (FPR=.05)	0.05	0.15	0.23	0.65	5.65	38.4
NPHGS (FPR=.10)	0.10	0.31	0.38	1.94	4.49	38.4
NPHGS (FPR= .15)	0.15	0.37	0.42	2.28	4.17	38.4
NPHGS (FPR=.20)	0.20	0.39	0.46	2.36	3.98	38.4

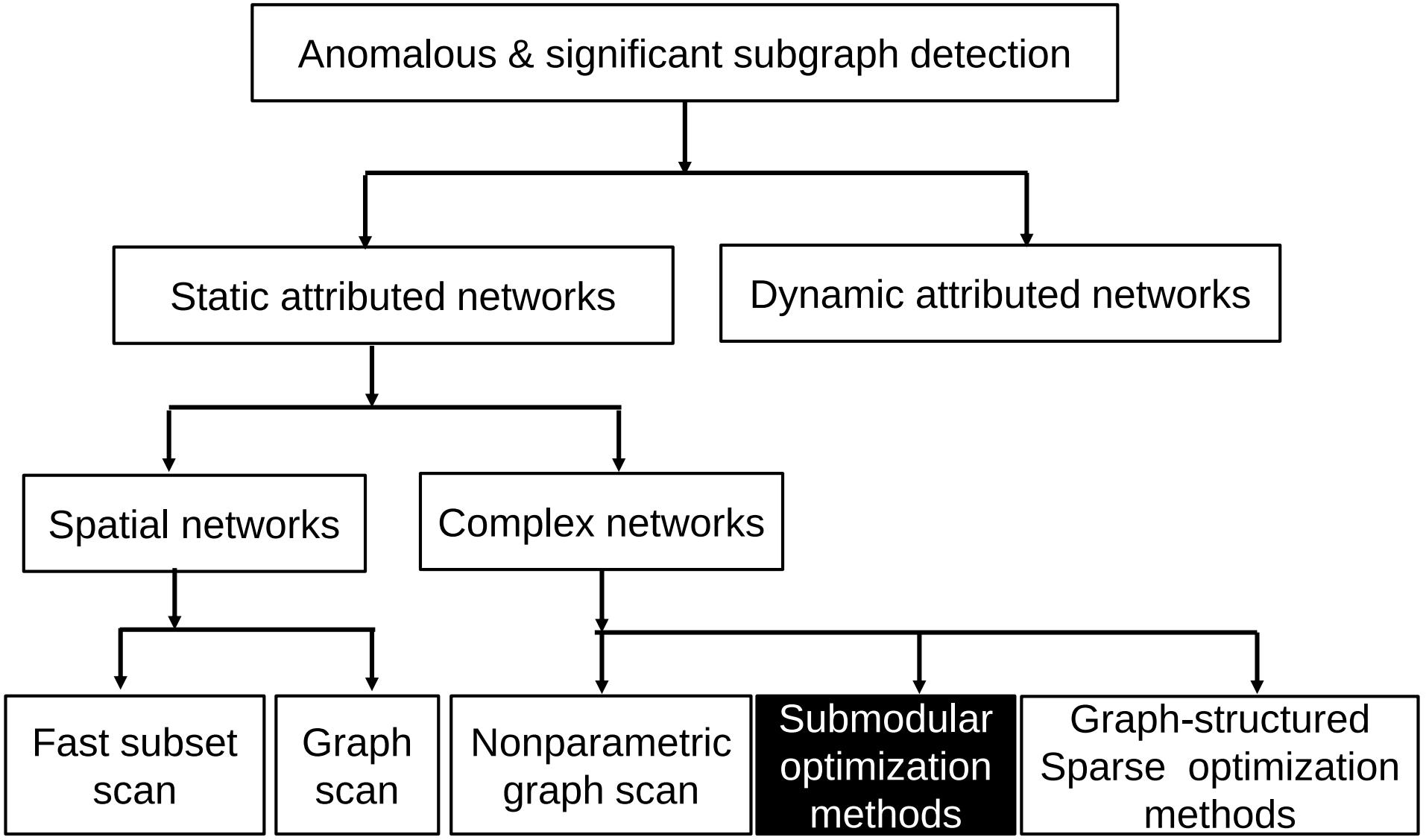
Table 3: Comparison between NPHGS and Existing Methods on the civil unrest datasets

NPHGS outperforms existing representative techniques for both event detection and forecasting, increasing **detection power**, **forecasting accuracy**, and **forecasting lead time** while reducing **time to detection**.

Similar improvements in performance were observed on a second task:

Early detection of rare disease outbreaks, using gold standard data about 17 hantavirus outbreaks from the Chilean Ministry of Health.

Taxonomy


Anomalous & significant subgraph detection
Static attributed networks
Dynamic attributed networks
Spatial networks
Complex networks
Fast subset scan
Graph scan
Nonparametric graph scan
Submodular optimization methods
Graph-structured Sparse optimization methods

Subgraph detection via submodular optimization

(Rozenshtein et al., KDD 2014)

- A class of subgraph detection problems can be framed as a general submodular (but not monotone) maximization problem:

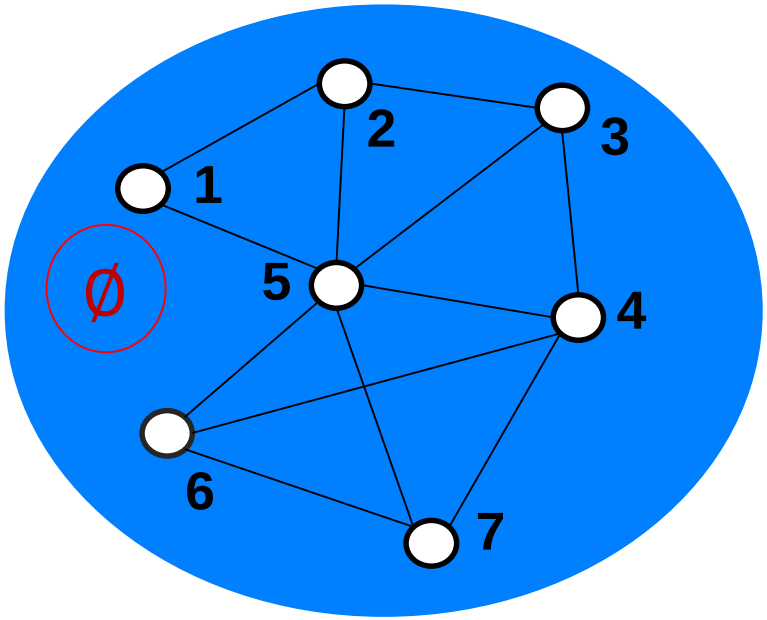
$$\max_S F(S) + l \times D(S)$$

A submodular score function that characterizes the level of anomalousness of the subset of nodes S.

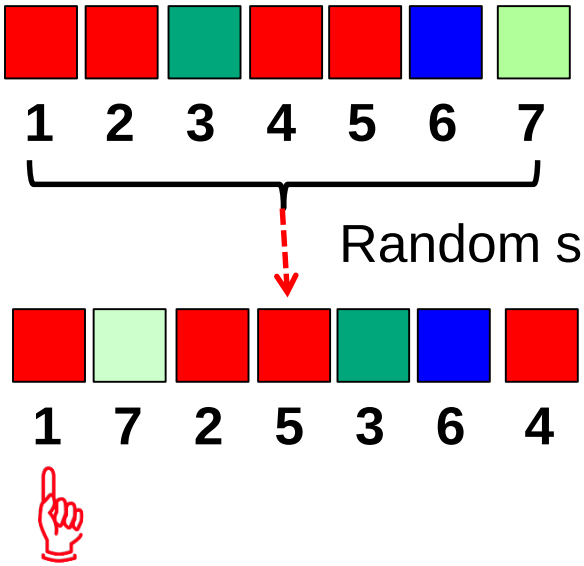
A submodular compactness function that gives a higher score if the subset of nodes S is more compact.

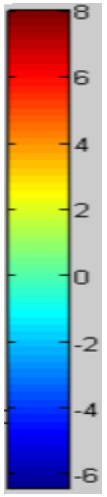
½-approximation for submodular maximization (Buchbinder et al., 2012)

Network topology $\mathbb{G} = (\mathbb{V}, \mathbb{E})$



Node-level attributes





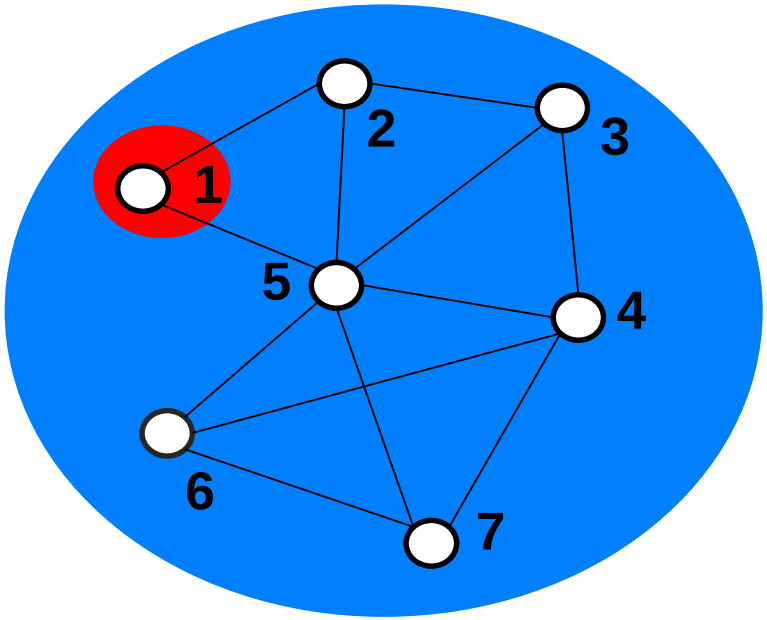
Initialize $A = \emptyset, B = \text{everything}$

In each step, grow A or shrink B

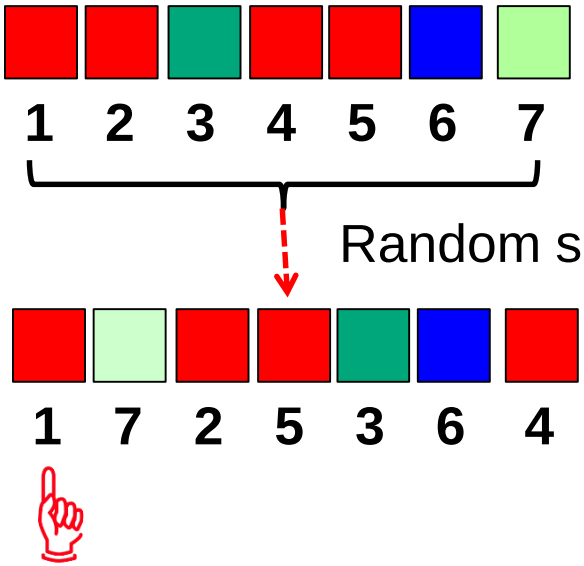
Invariant: $A \subseteq B$

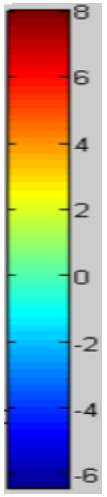
½-approximation for submodular maximization (Buchbinder et al., 2012)

Network topology $\mathbb{G} = (\mathbb{V}, \mathbb{E})$



Node-level attributes





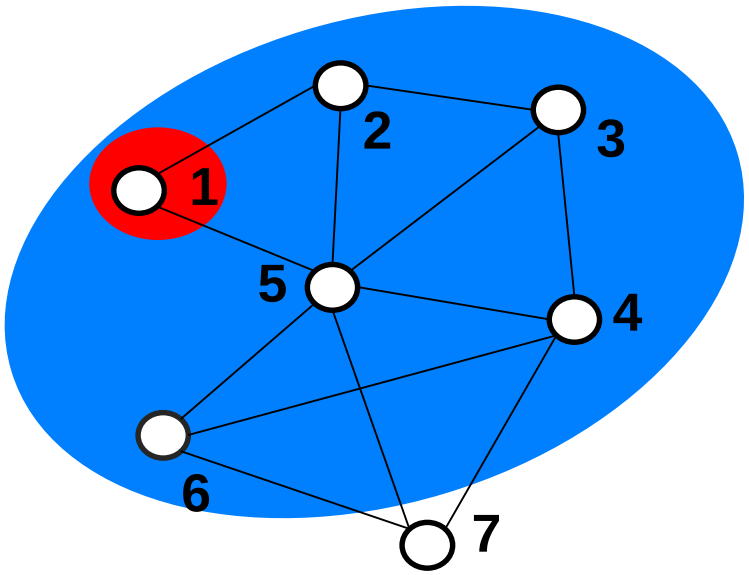
Initialize $A = \emptyset, B =$ everything

In each step, grow A or shrink B

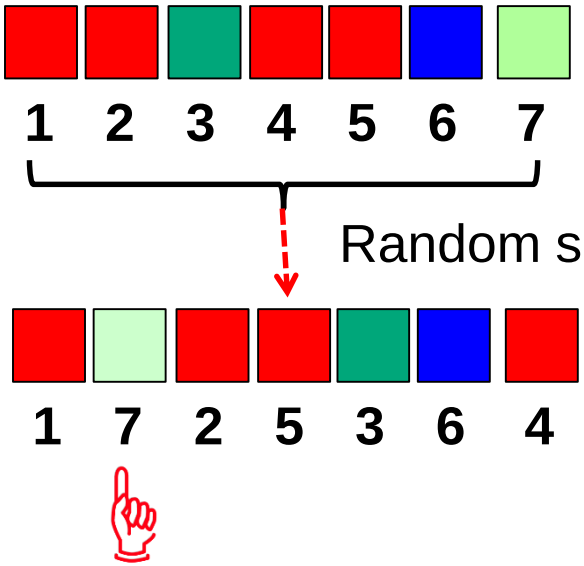
Invariant: $A \subseteq B$

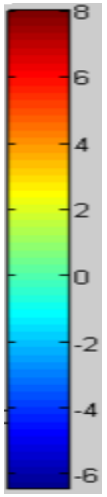
½-approximation for submodular maximization (Buchbinder et al., 2012)

Network topology $\mathbb{G} = (\mathbb{V}, \mathbb{E})$



Node-level attributes





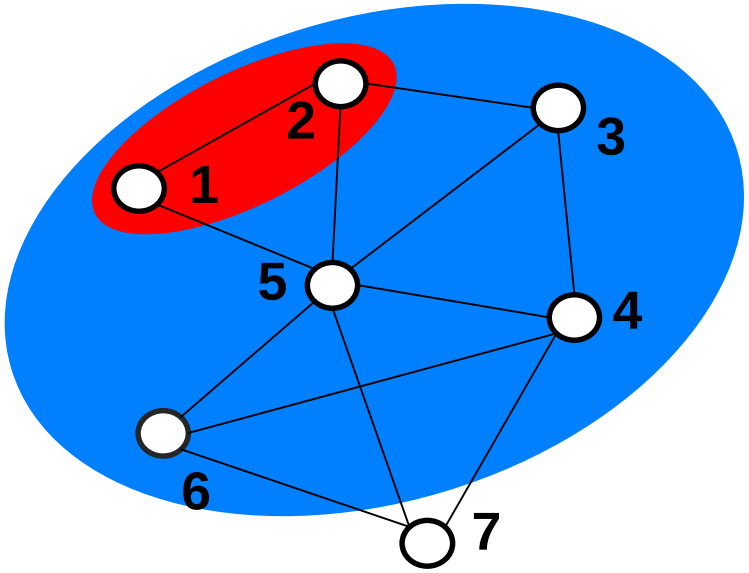
Initialize $A = \emptyset, B = \text{everything}$

In each step, grow A or shrink B

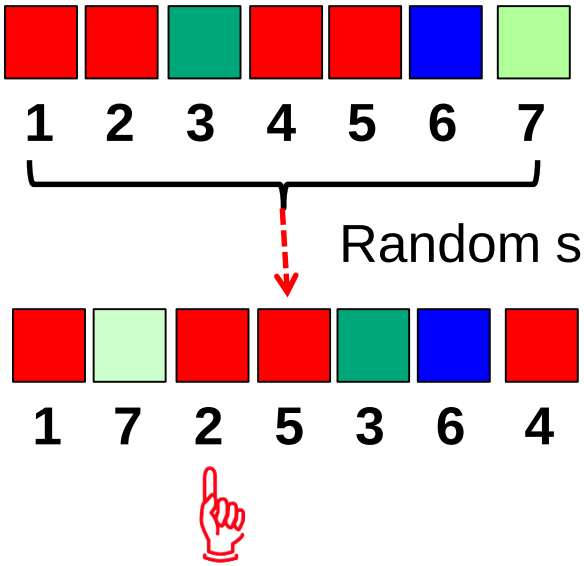
Invariant: $A \subseteq B$

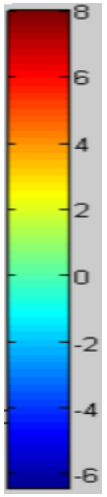
½-approximation for submodular maximization (Buchbinder et al., 2012)

Network topology $\mathbb{G} = (\mathbb{V}, \mathbb{E})$



Node-level attributes





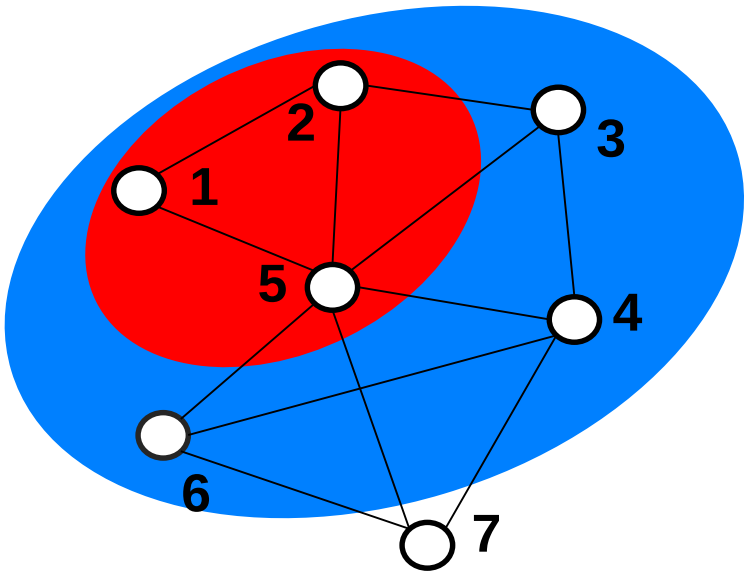
Initialize $A = \emptyset, B =$ everything

In each step, grow A or shrink B

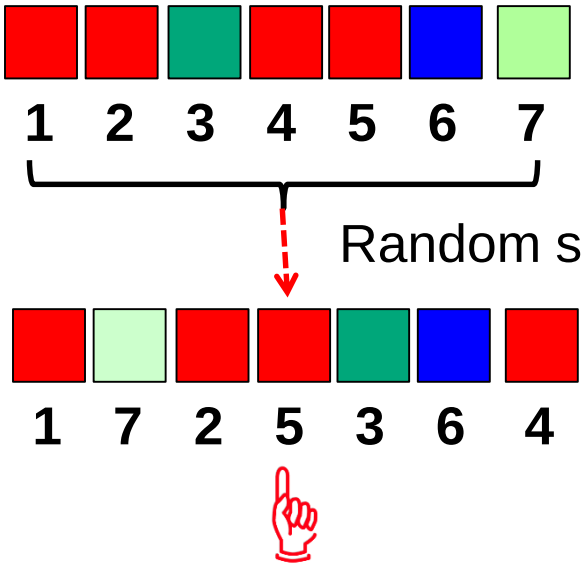
Invariant: $A \subseteq B$

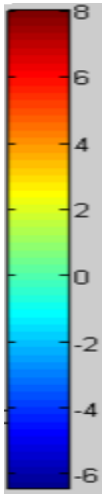
½-approximation for submodular maximization (Buchbinder et al., 2012)

Network topology $\mathbb{G} = (\mathbb{V}, \mathbb{E})$



Node-level attributes





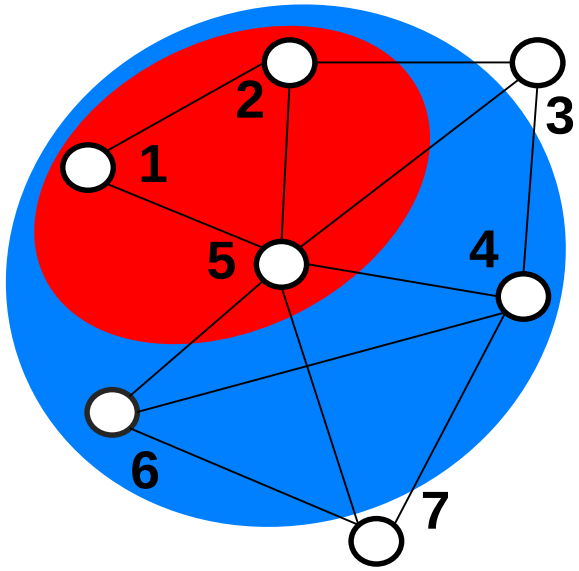
Initialize $A = \emptyset, B =$ everything

In each step, grow A or shrink B

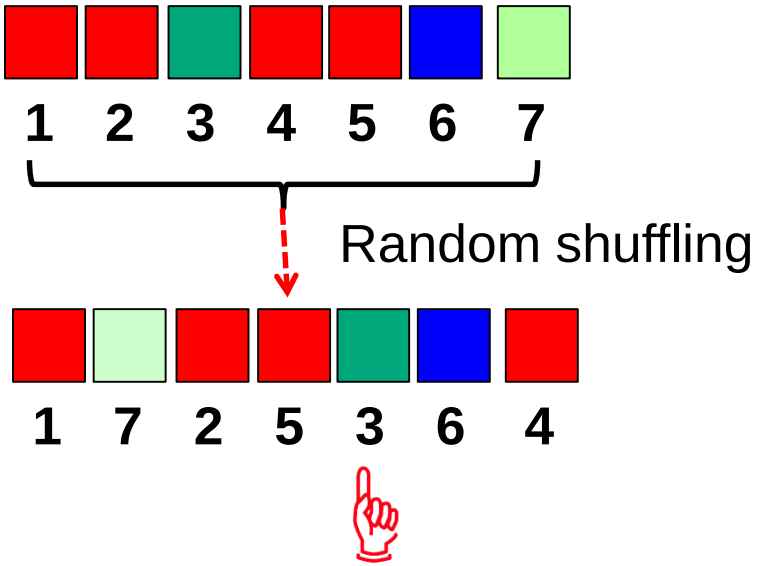
Invariant: $A \subseteq B$

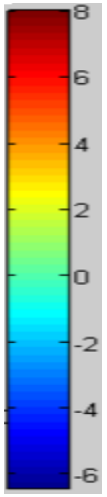
½-approximation for submodular maximization (Buchbinder et al., 2012)

Network topology $\mathbb{G} = (\mathbb{V}, \mathbb{E})$



Node-level attributes





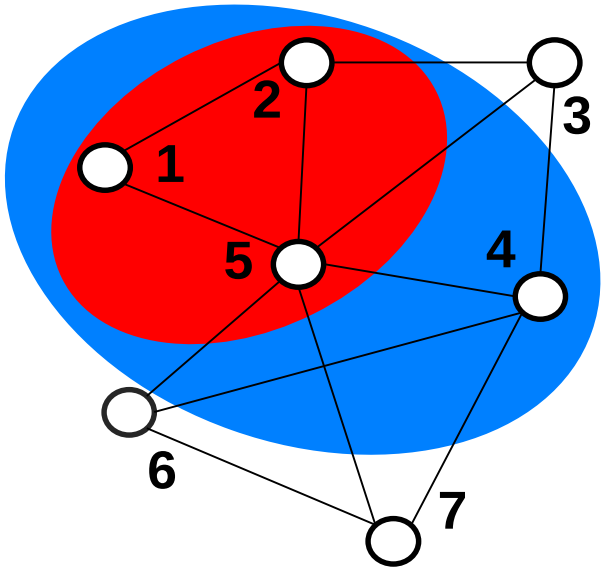
Initialize $A = \emptyset, B = \text{everything}$

In each step, grow A or shrink B

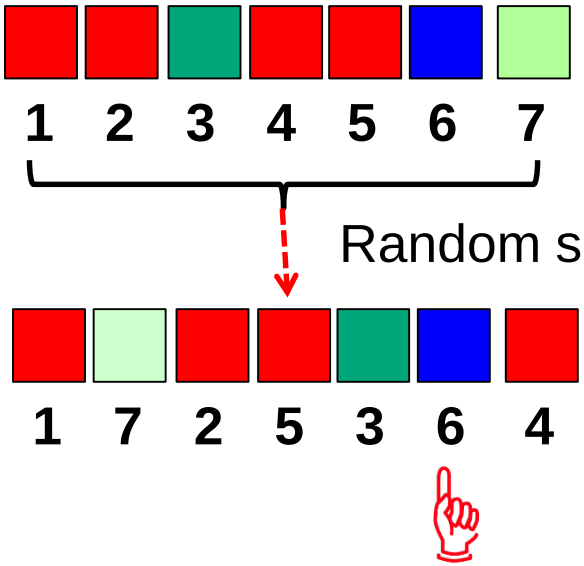
Invariant: $A \subseteq B$

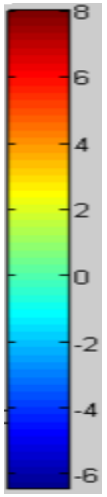
½-approximation for submodular maximization (Buchbinder et al., 2012)

Network topology $\mathbb{G} = (\mathbb{V}, \mathbb{E})$



Node-level attributes





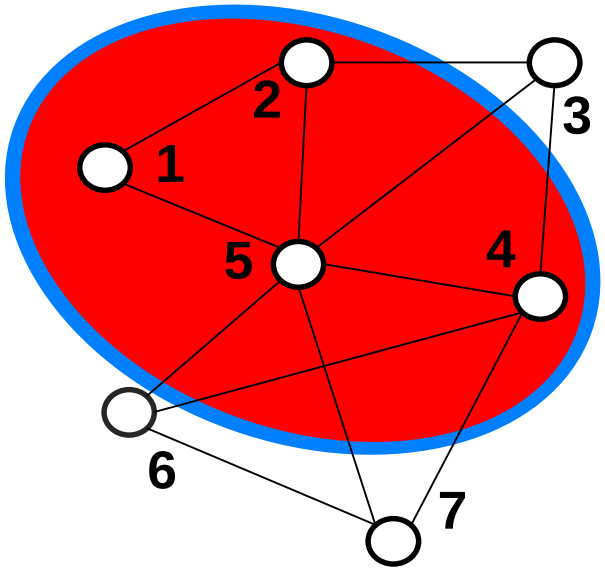
Initialize $A = \emptyset, B =$ everything

In each step, grow A or shrink B

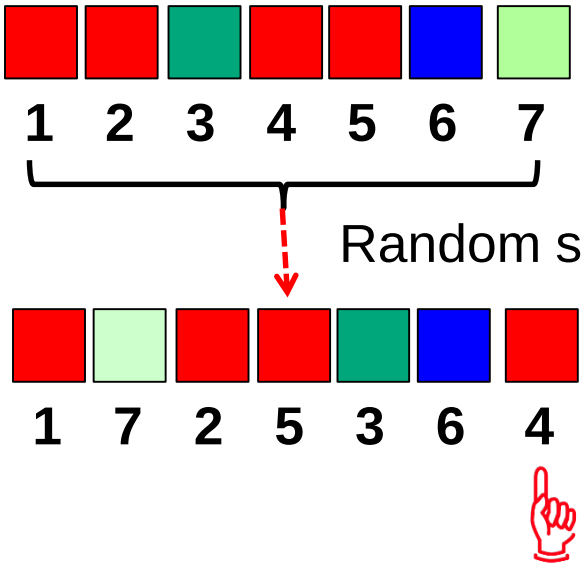
Invariant: $A \subseteq B$

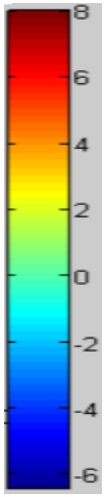
½-approximation for submodular maximization (Buchbinder et al., 2012)

Network topology $\mathbb{G} = (\mathbb{V}, \mathbb{E})$



Node-level attributes





S={1,2,4,5} ⟵

Initialize $A = \emptyset, B =$ everything

In each step, grow A or shrink B

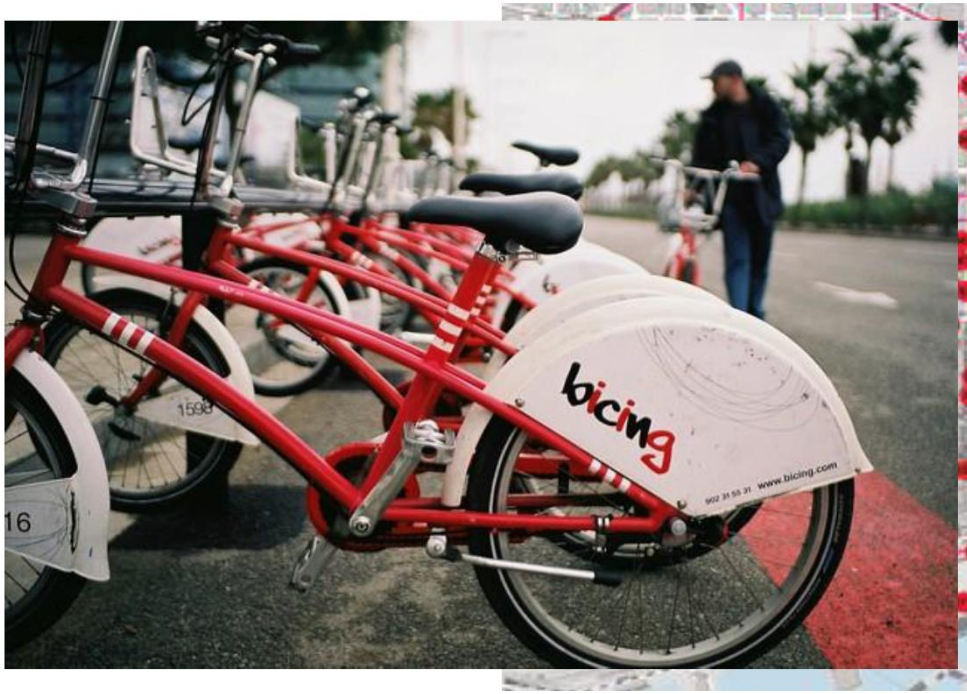
Invariant: $A \subseteq B$

When $A = B$, we return A (or B) as the final subset of nodes.

(Rozenshtein et al., KDD 2014)

Case studies: event detection

- Bicing sensor networks

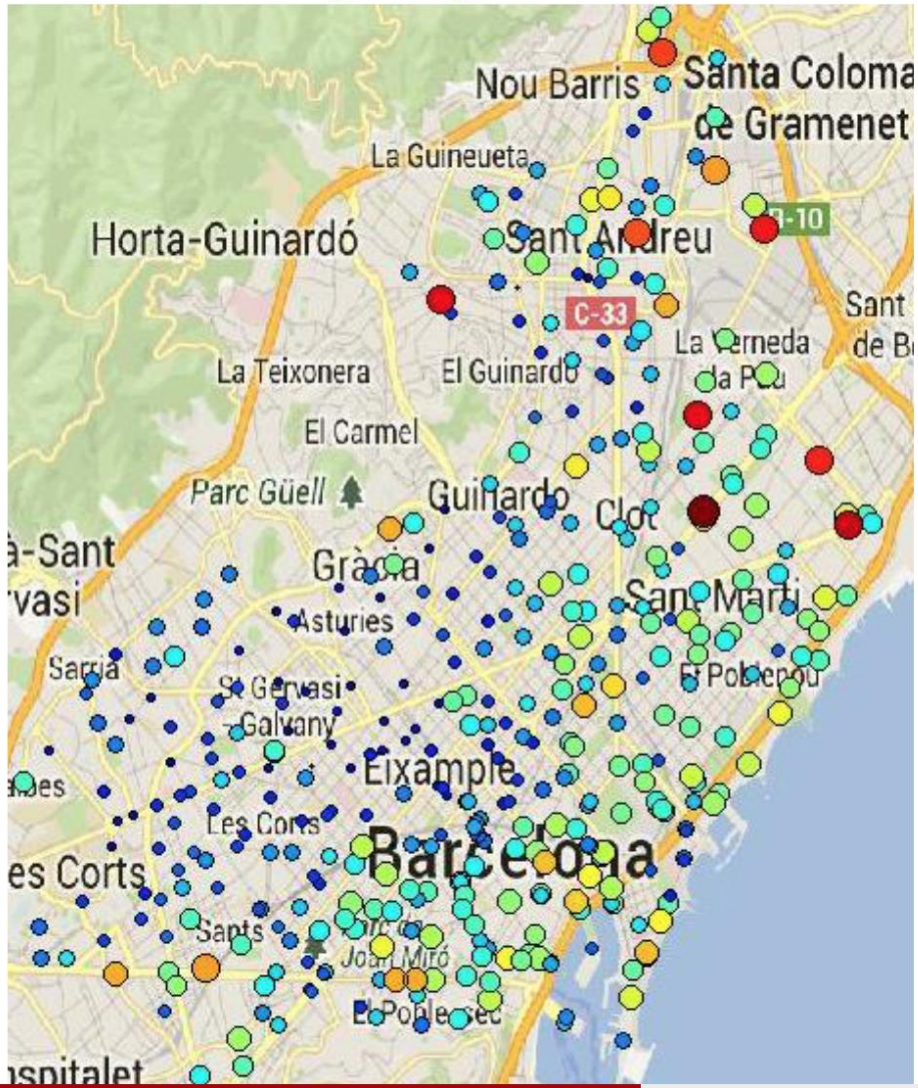


(Rozenshtein et al., KDD 2014)

Case studies: event detection

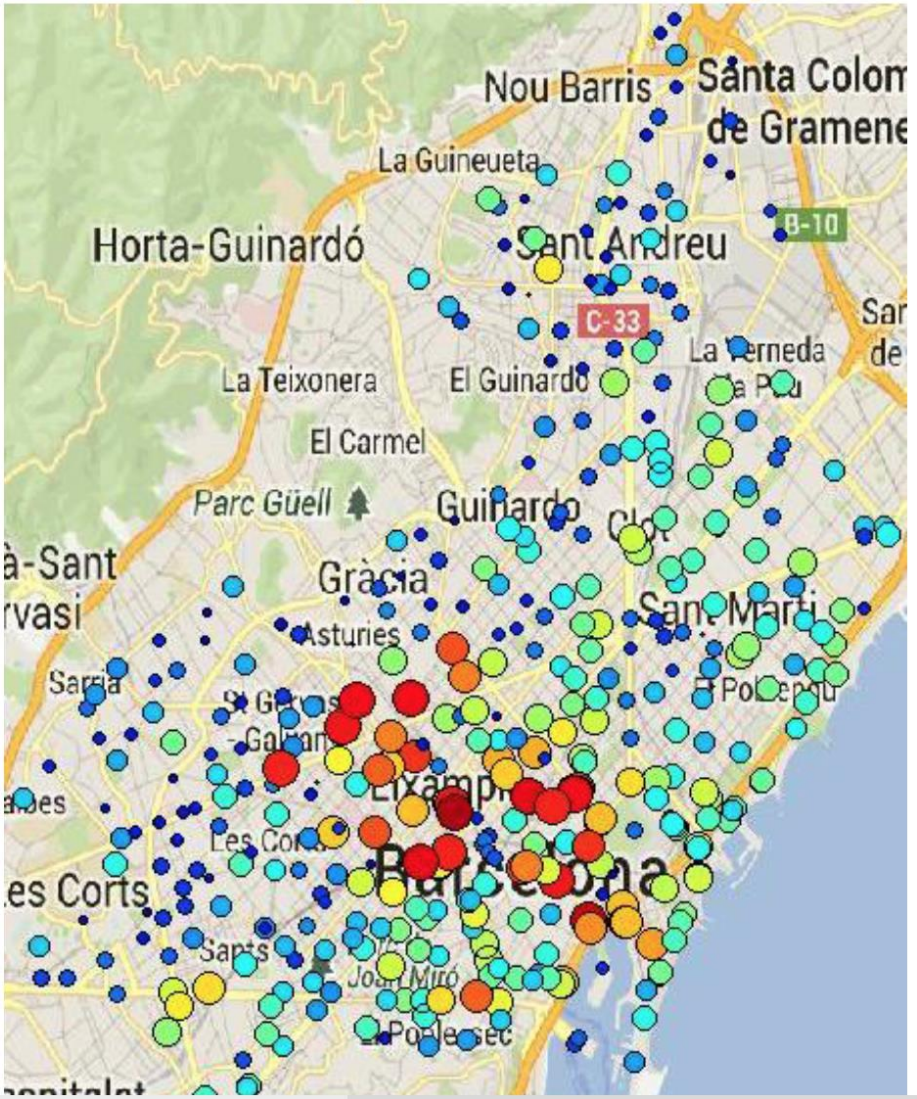
15.11.2012

ordinary day, no events



11.09.2012

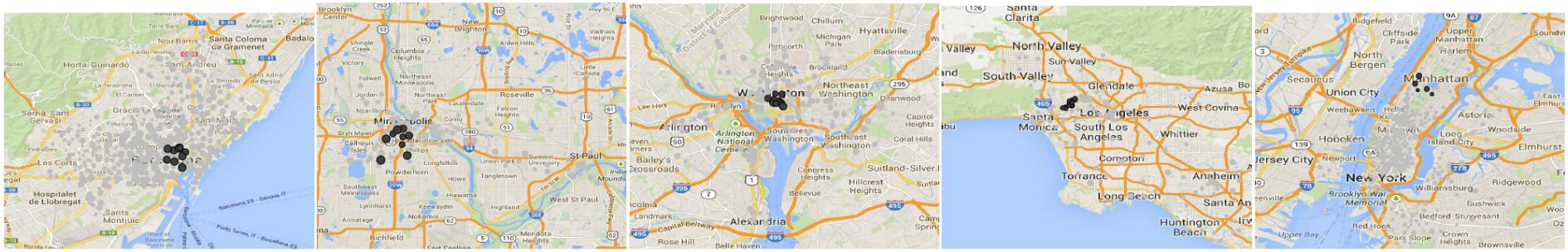
Catalunya national day



(Rozenshtein et al., KDD 2014)

Case studies: event detection

- Events discovered with bicing data



(a) Barcelona: 11.09.12 National Day of Catalonia

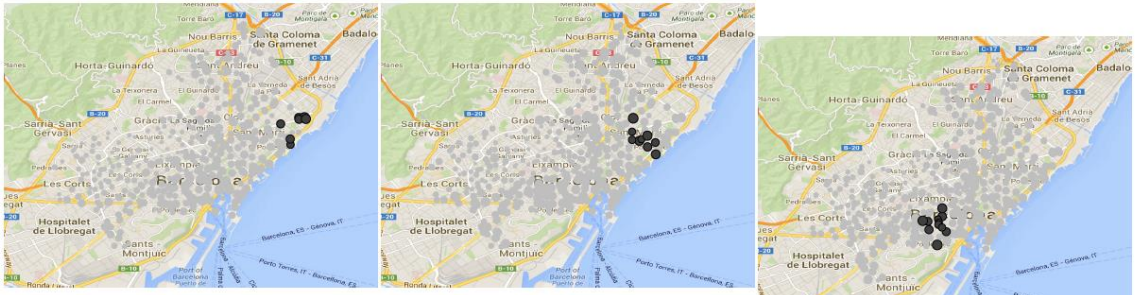
(b) Minneapolis: 4.07.12 Independence Day

(c) Washington, DC: 27.05.13 Memorial Day

(d) Los Angeles: 31.05.10 Memorial Day

(e) New York: 6.09.10 Labor Day

Figure 4: Public holiday city-events discovered using the SDP algorithm.

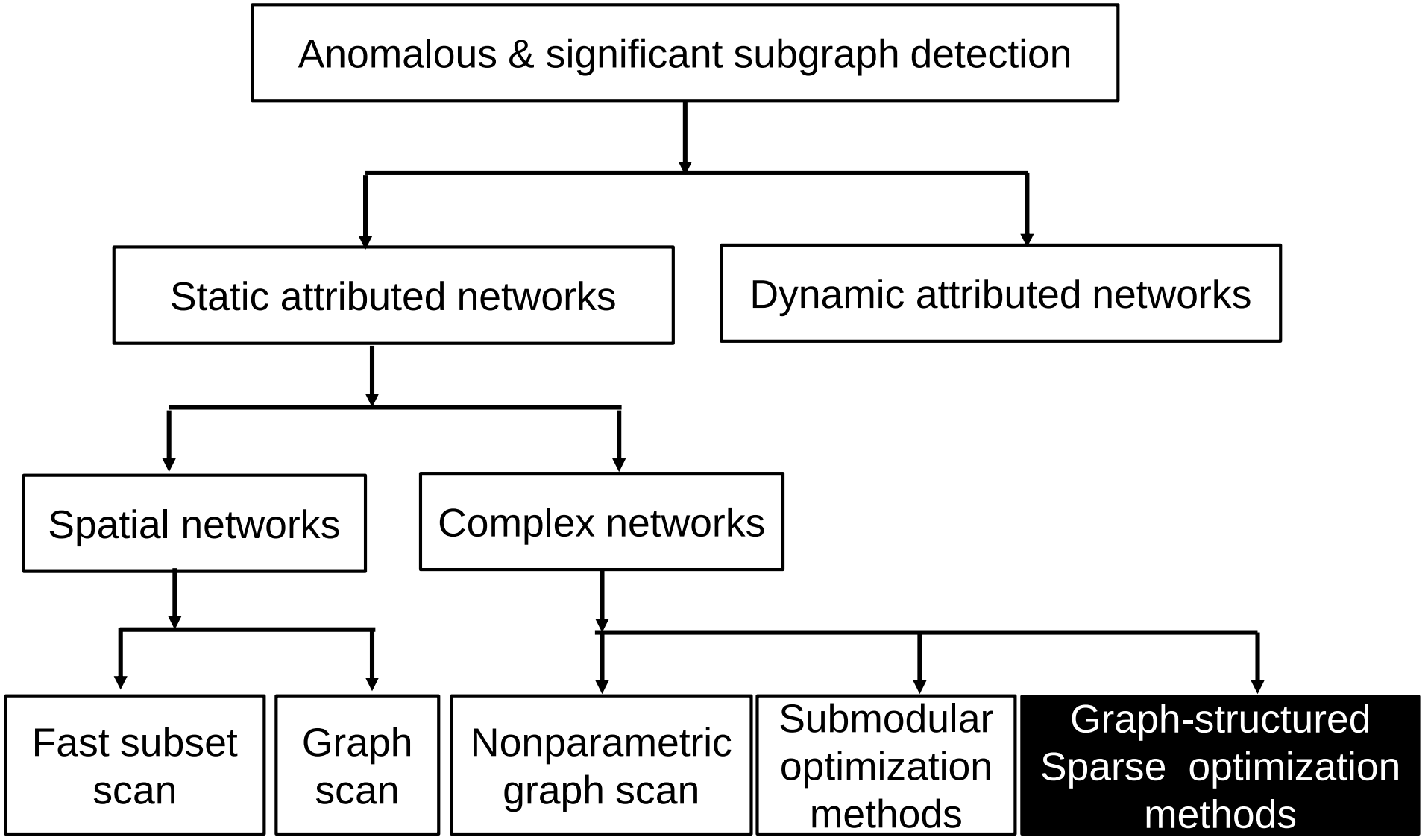


(a) 01.06.12 Primavera sound music festival

(b) 18.09.12 festival of the Poblenou neighborhood

(c) 31.10.12 Halloween

Taxonomy


Anomalous & significant subgraph detection
Static attributed networks
Dynamic attributed networks
Spatial networks
Complex networks
Fast subset scan
Graph scan
Nonparametric graph scan
Submodular optimization methods
Graph-structured Sparse optimization methods

Graph structured sparse optimization

The problem of subgraph detection

$$\max_{S \subseteq V} F(S) \quad s.t.\ S \text{ satisfies a predefined topological constraint.}$$

can be reformulated as

$$\max_{\boldsymbol{y} \subseteq \{0,1\}^n} f(\boldsymbol{y}) \quad s.t.\ \mathrm{supp}(\boldsymbol{y}) \text{ satisfies a predefined topological constraint.}$$

where $\mathrm{supp}(\boldsymbol{y}) = \{i \mid y_i > 0\}$ and S can be identified as

$$S = \mathrm{supp}(\boldsymbol{y}), \text{ and } f(\boldsymbol{y}) = F(S)$$

Graph structured sparse optimization

- This approach solves the relaxed problem

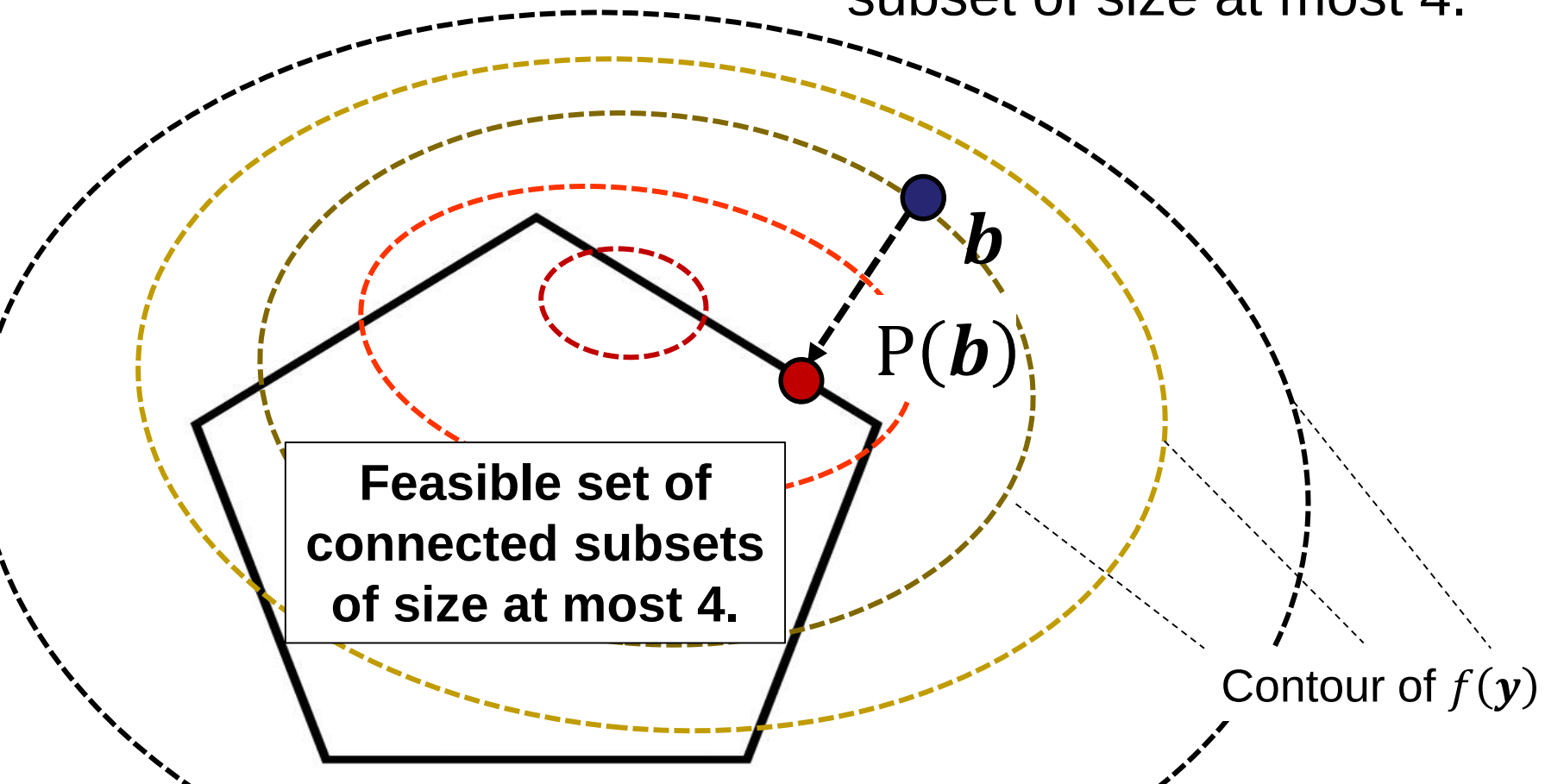
$$\max_{\boldsymbol{y}\subseteq[0,1]^n} f(\boldsymbol{y}) \quad s.t.\ \text{supp}(\boldsymbol{y}) \text{ satisfies a predefined topological constraint.}$$

- Three novel sparse optimization algorithms
  - Graph-structured iterative hard thresholding (Graph-IHT). (Zhou and Chen, ICDM, 2016)
  - Graph-structured gradient hard thresholding Pursuit (Graph-GHTP). (Zhou and Chen, ICDM, 2016)
  - Graph-structured matching pursuit (Graph-MP) (Chen and Zhou, IJCAI, 2016)

Interpretation of projection oracle

- A projection oracle $\mathrm{P}(\boldsymbol{b})$ is defined as

$$\mathrm{P}(\boldsymbol{b}) = \arg\min_{\boldsymbol{y}\in\mathbb{R}^n} \|\boldsymbol{y}-\boldsymbol{b}\|_2^2 \quad s.t. \quad \mathrm{supp}(\boldsymbol{y}) \text{ is a connected subset of size at most 4.}$$

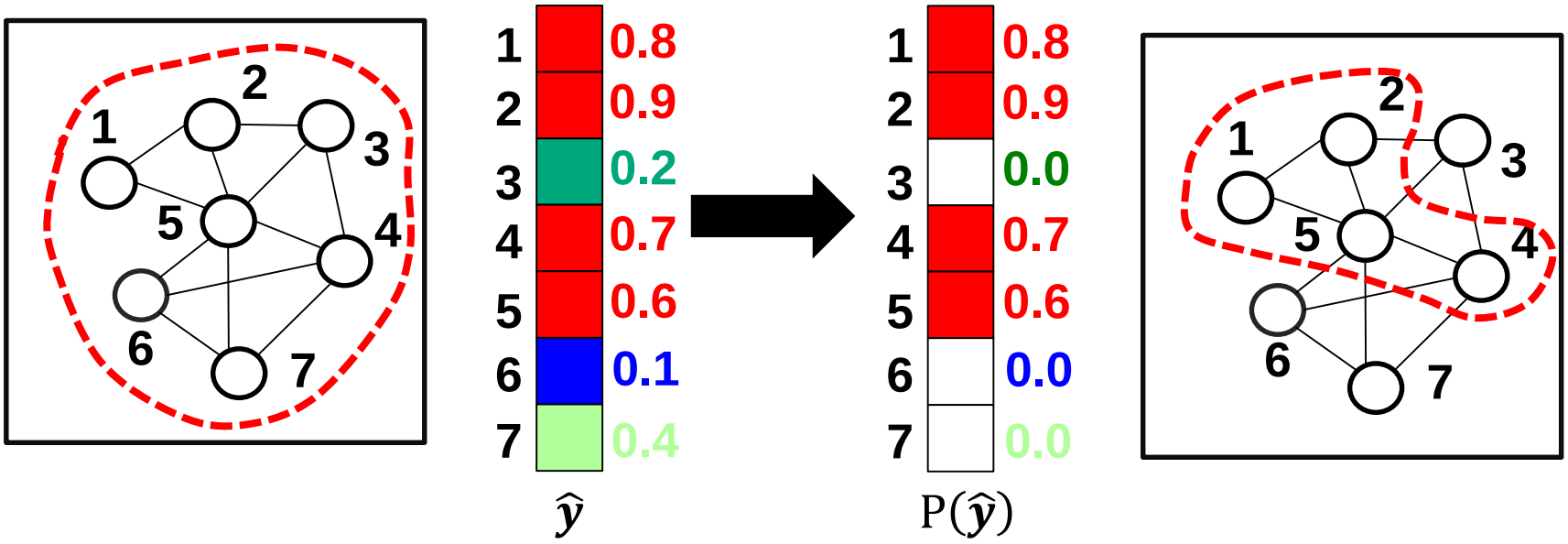


Interpretation of projection oracle

- A projection oracle $\mathrm{P}(\boldsymbol{b})$ is defined as

$$\mathrm{P}(\boldsymbol{b}) = \arg\min_{\boldsymbol{y}\in\mathbb{R}^n} \|\boldsymbol{y}-\boldsymbol{b}\|_2^2 \quad s.t. \quad \mathrm{supp}(\boldsymbol{y}) \text{ is a connected subset of size at most 4.}$$

Network topology $\mathbb{G} = (\mathbb{V}, \mathbb{E})$



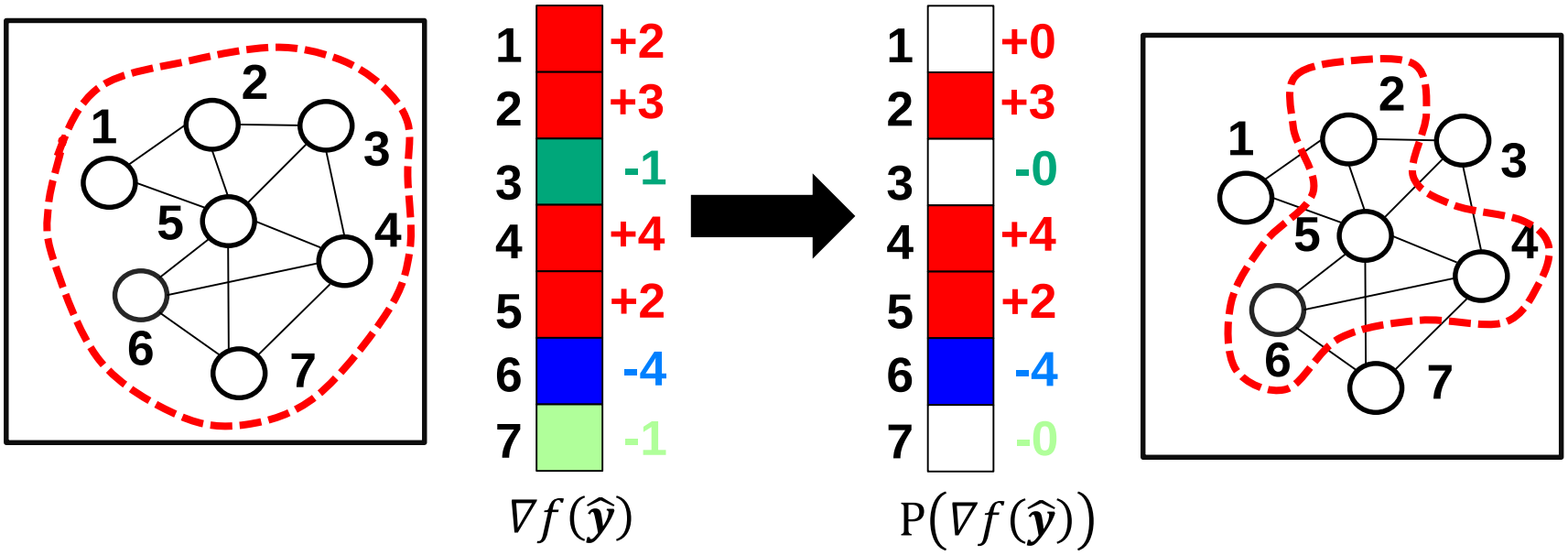
Projection of relaxed vector $\hat{y}$

Interpretation of projection oracle

- A projection oracle $\mathrm{P}(\boldsymbol{b})$ is defined as

$$\mathrm{P}(\boldsymbol{b}) = \arg\min_{\boldsymbol{y}\in\mathbb{R}^n} \|\boldsymbol{y}-\boldsymbol{b}\|_2^2 \quad s.t. \quad \mathrm{supp}(\boldsymbol{y}) \text{ is a connected subset of size at most 4.}$$

Network topology $\mathbb{G} = (\mathbb{V}, \mathbb{E})$



Projection of a gradient $\nabla f(\hat{\boldsymbol{y}})$.

Description of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)

Algorithm : GRAPH-IHT

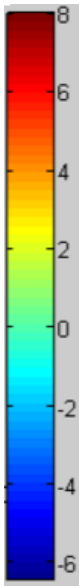
1 **Input:** Instance $\mathbb{G}$;
2 **Output:** The subset S;
3 $i \leftarrow 0$, $\mathbf{y}^i \leftarrow$ an initial vector;
4 **repeat**
5 $\quad \mathbf{b} \leftarrow \mathbf{y}^i + \eta \cdot \mathrm{P}(\nabla f(\mathbf{y}^i))$;
6 $\quad \mathbf{y}^{i+1} \leftarrow \mathrm{P}(\mathbf{b})$
7 $\quad i = i + 1$;
8 **until** *halting condition holds* ;
9 $S = \mathrm{supp}(\mathbf{y}^{i+1})$;
10 **return** S

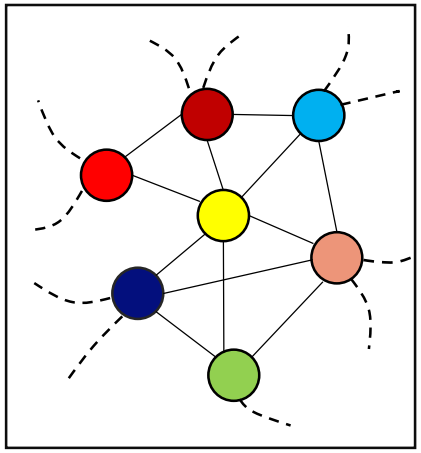
Projection on the gradient $\nabla f(\boldsymbol{y}^i)$

Projection on an intermediate solution $\boldsymbol{b}$

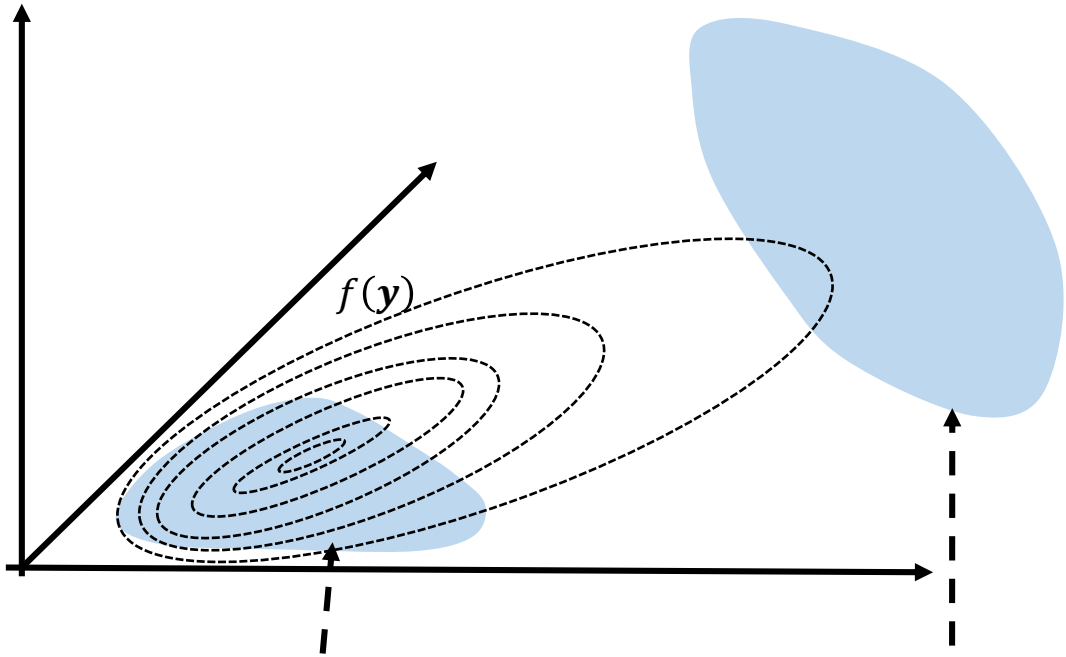
Illustration of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)





Network instance $\mathbb{G}$

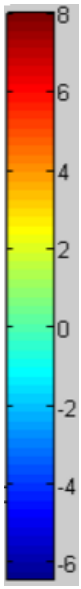


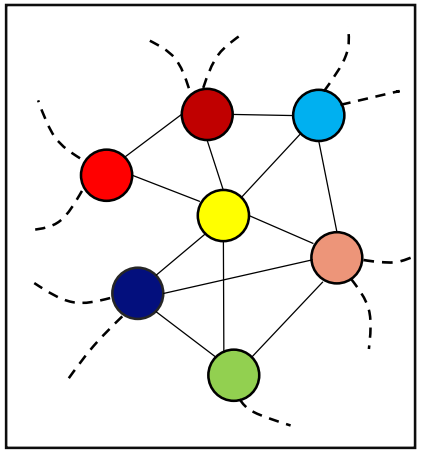
$$\mathcal{M}(\mathbb{G}, k = 5) = \{\boldsymbol{y} \mid \boldsymbol{y} \in [0,1]^n, \text{supp}(\boldsymbol{y}) \in \mathbb{M}(\mathbb{G}, k = 5)\}$$

$\mathbb{M}(\mathbb{G}, k = 5)$ represents the space of connected subsets of size at most 5.

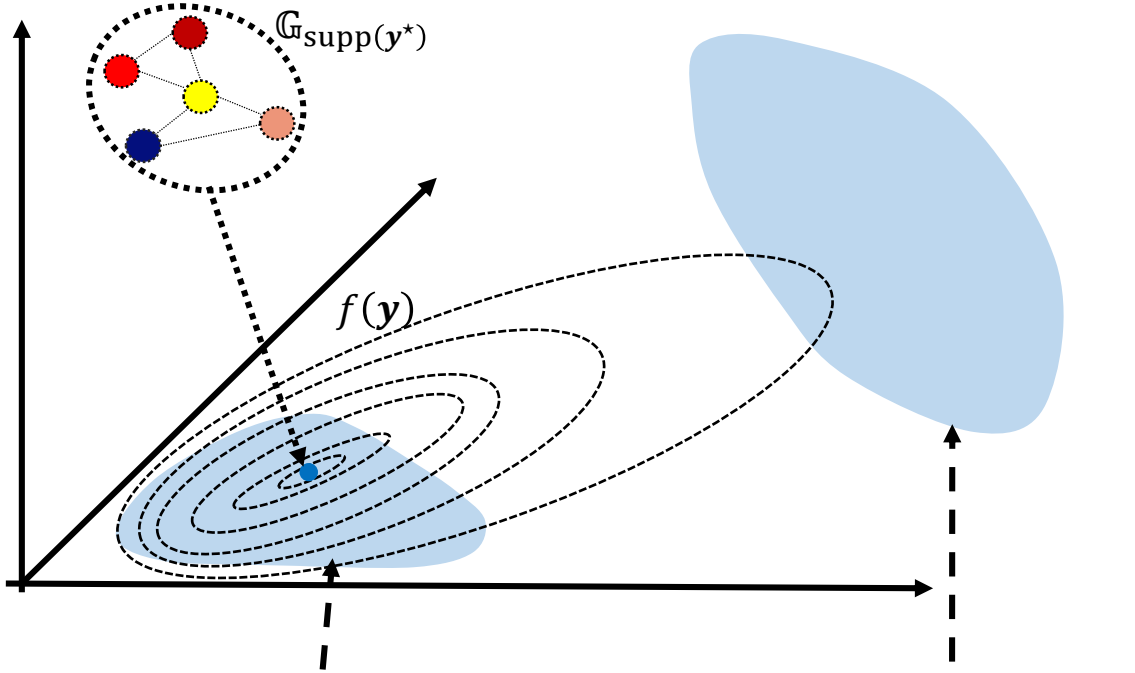
Illustration of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)





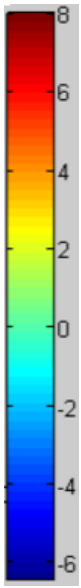
Network instance $\mathbb{G}$

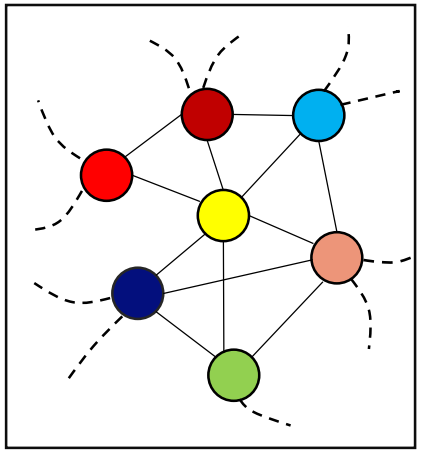


$$\mathcal{M}(\mathbb{G}, k=5) = \{y \mid y \in [0,1]^n, \text{supp}(y) \in \mathbb{M}(\mathbb{G}, k=5)\}$$

Illustration of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)





Network instance $\mathbb{G}$

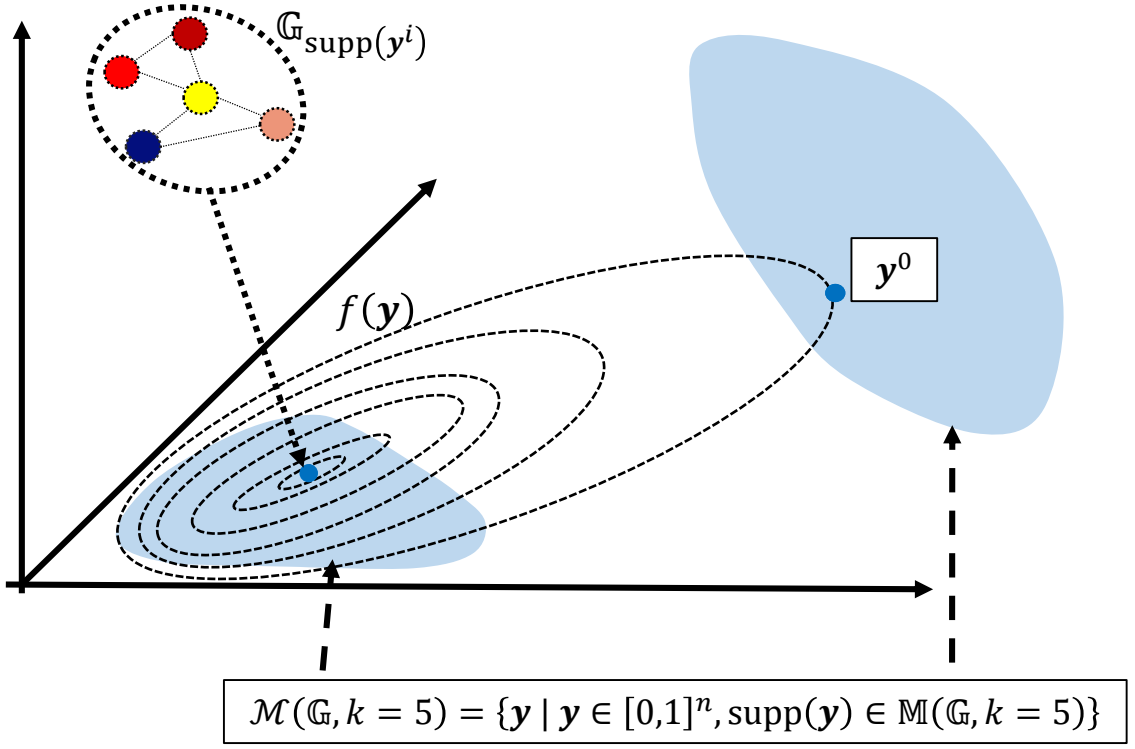


Illustration of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)

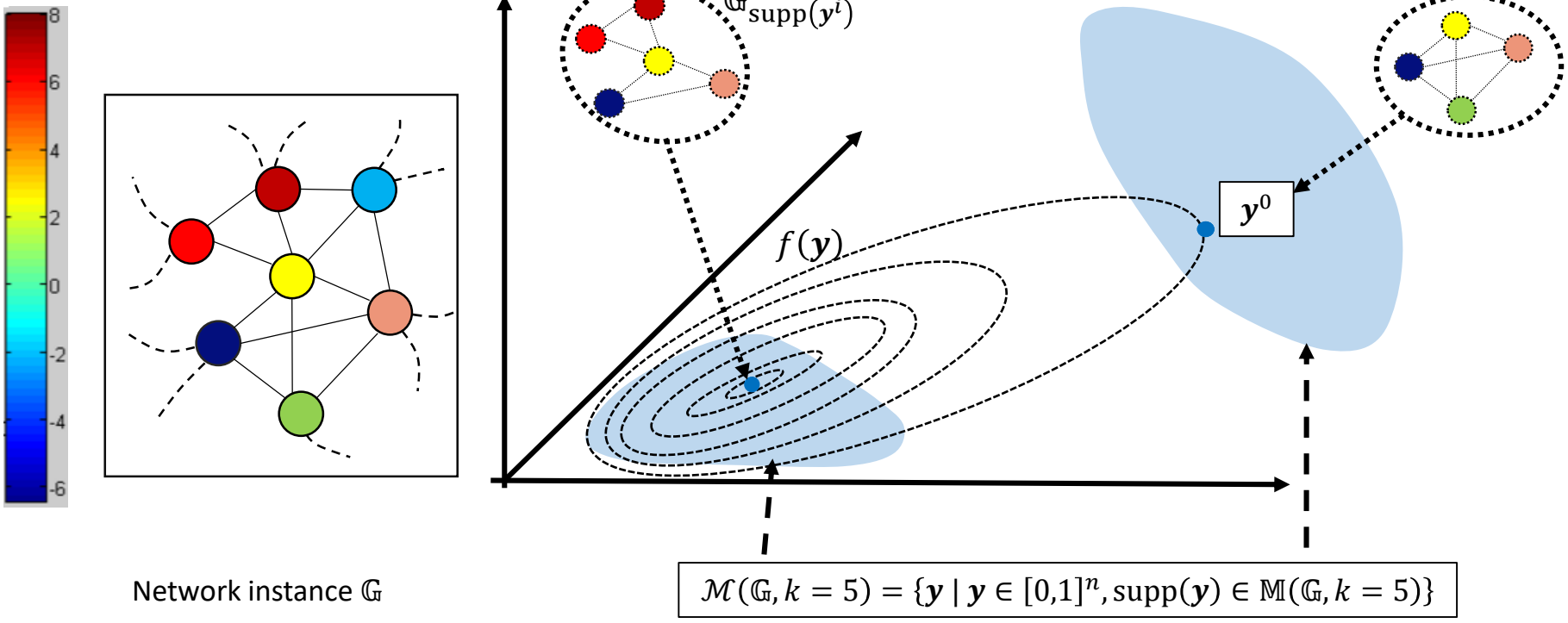


Illustration of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)

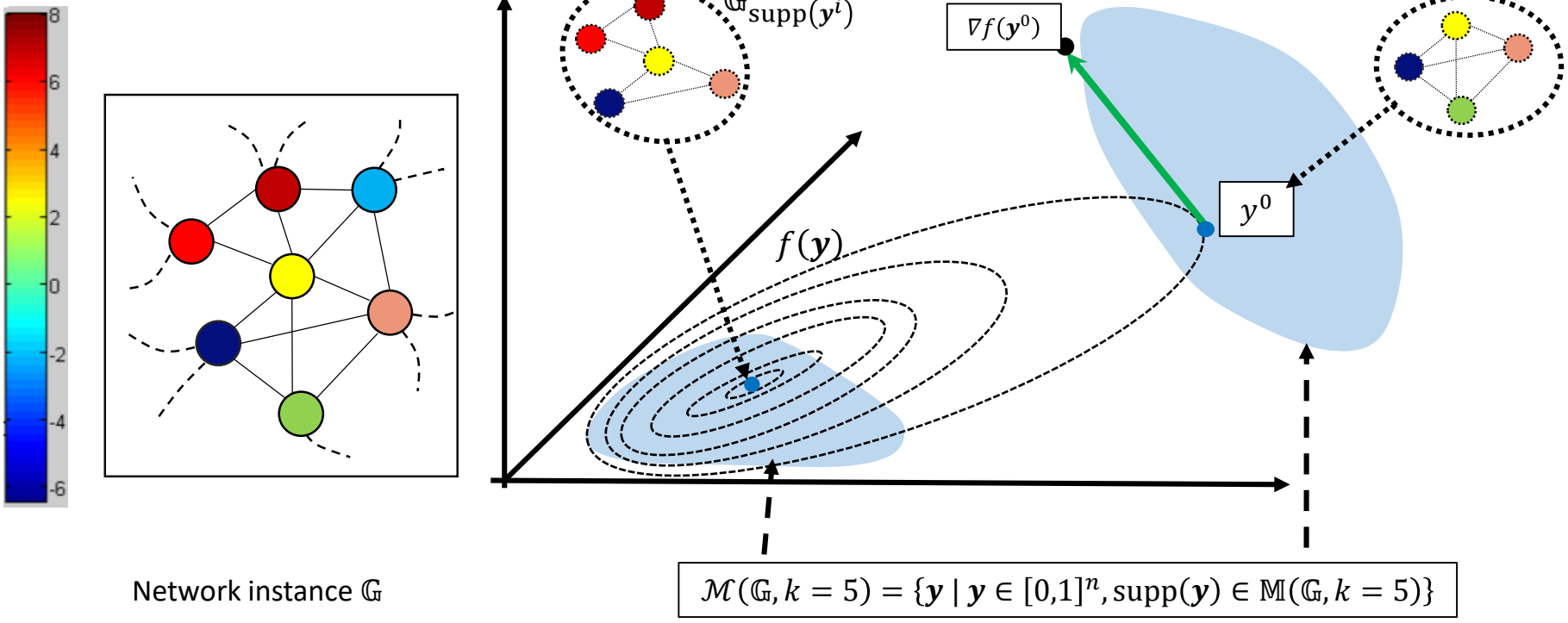


Illustration of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)

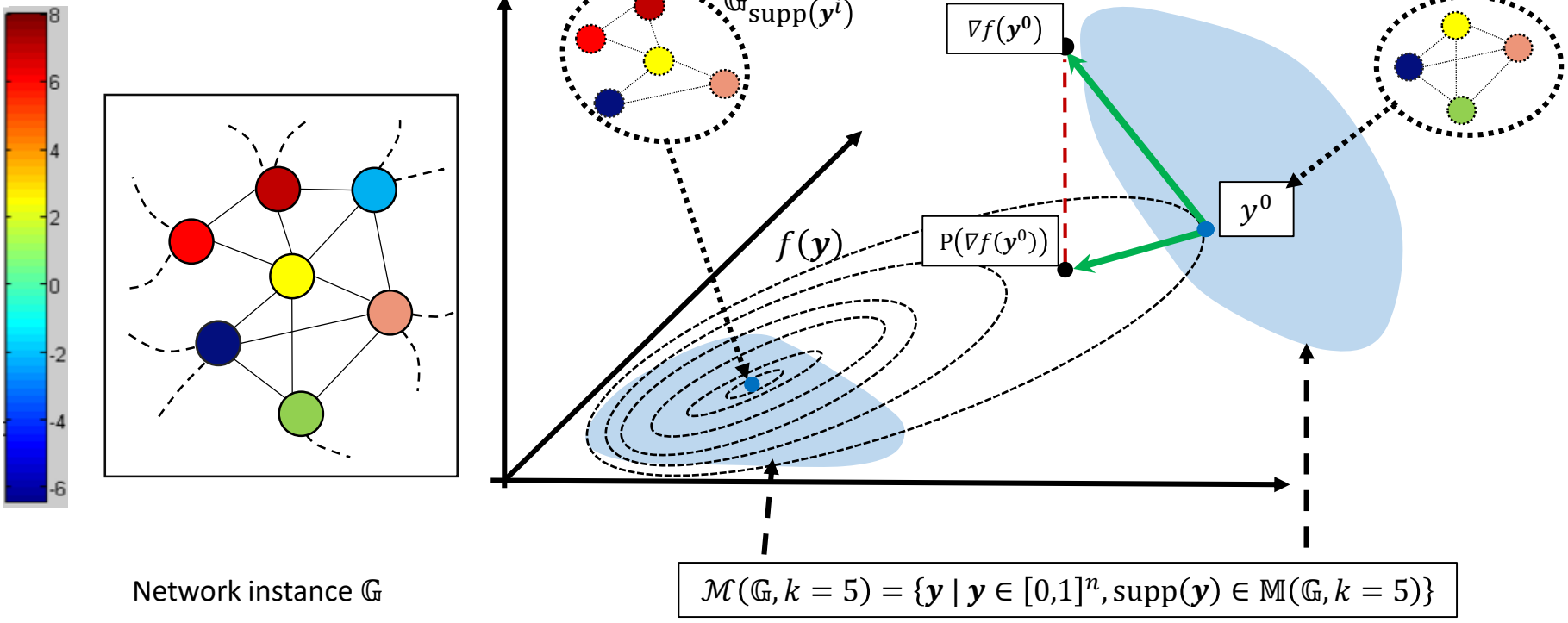


Illustration of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)

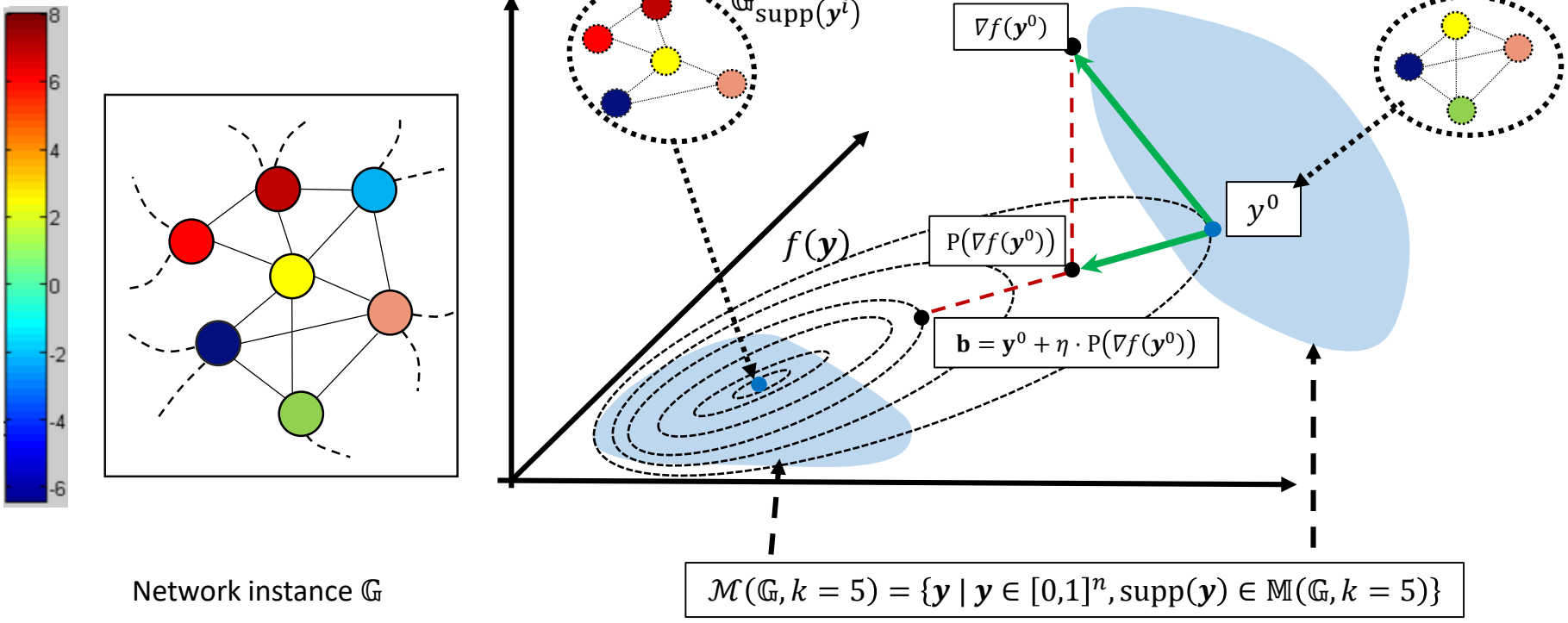


Illustration of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)

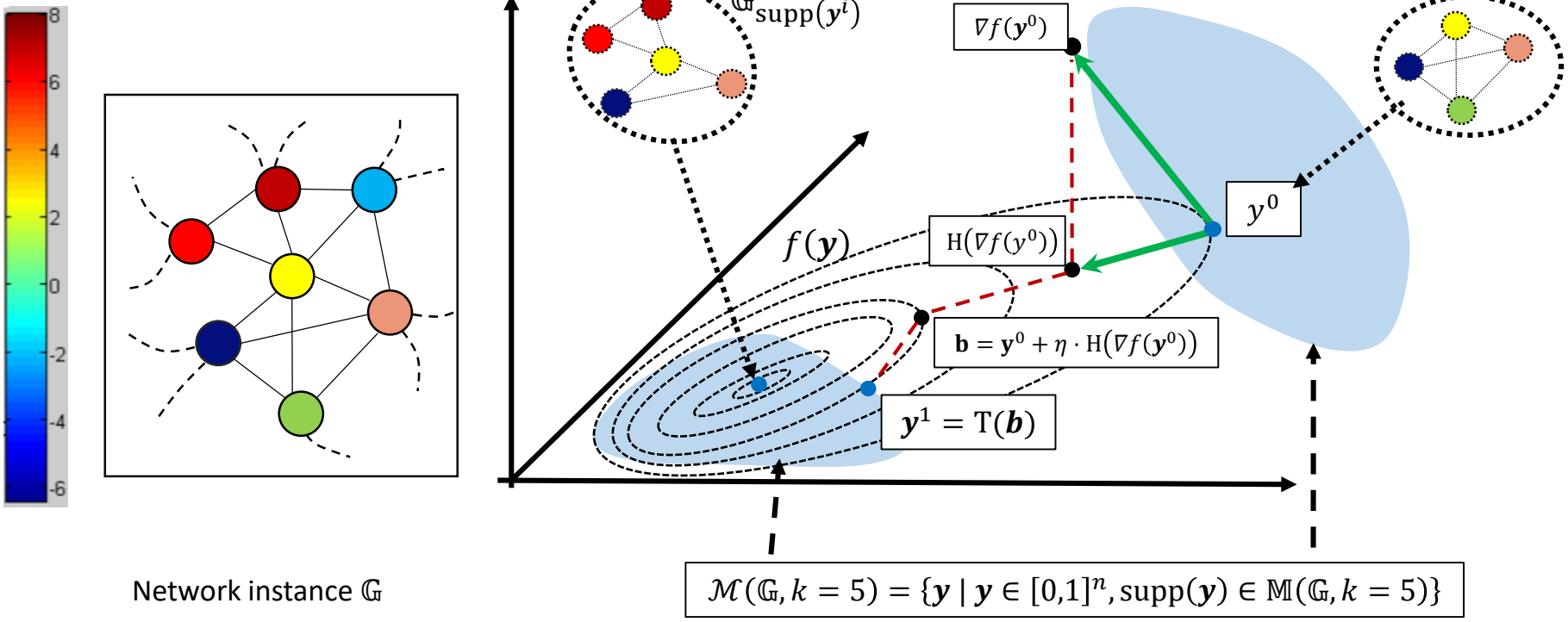
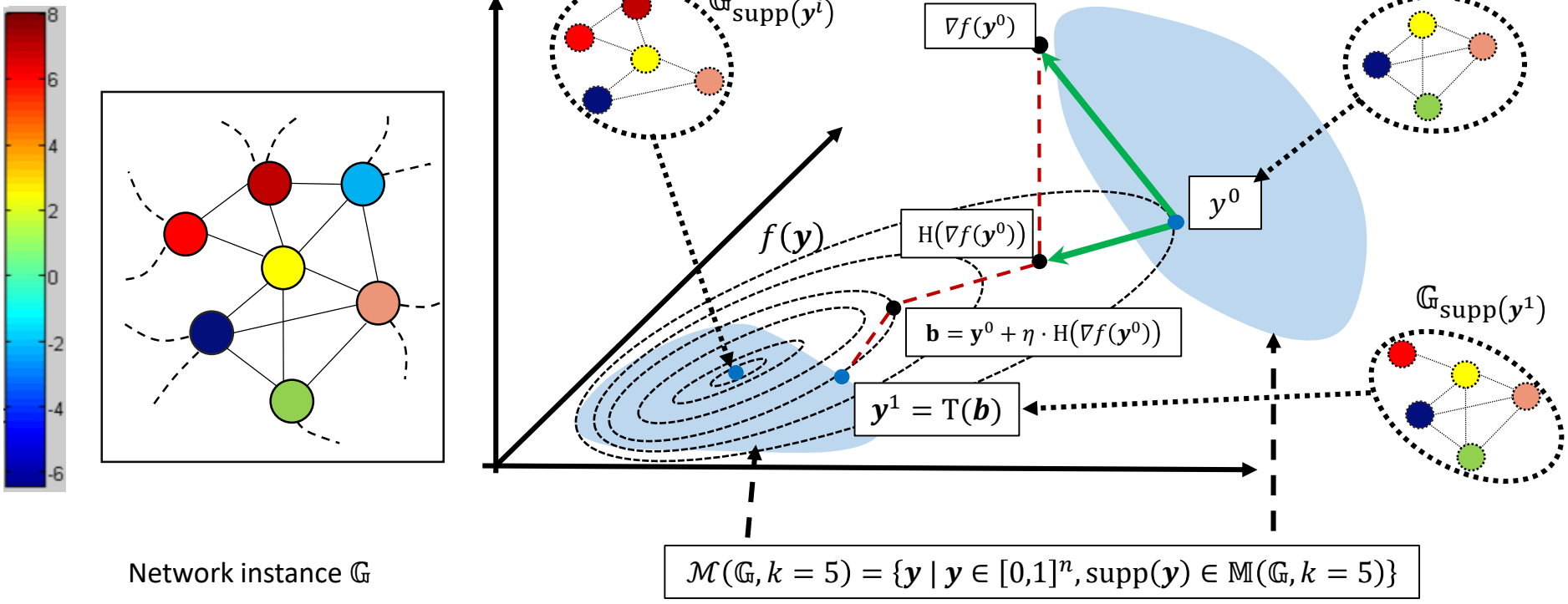


Illustration of the Graph-IHT algorithm

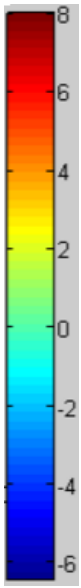
(Zhou and Chen, ICDM, 2016)

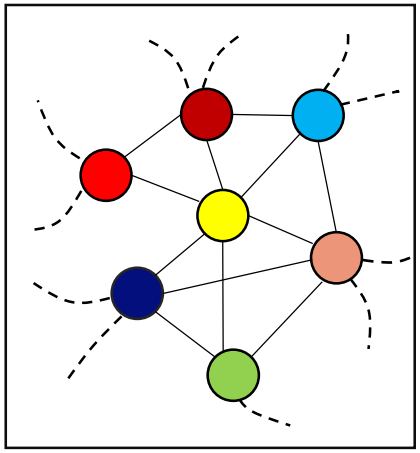


Network instance $\mathbb{G}$

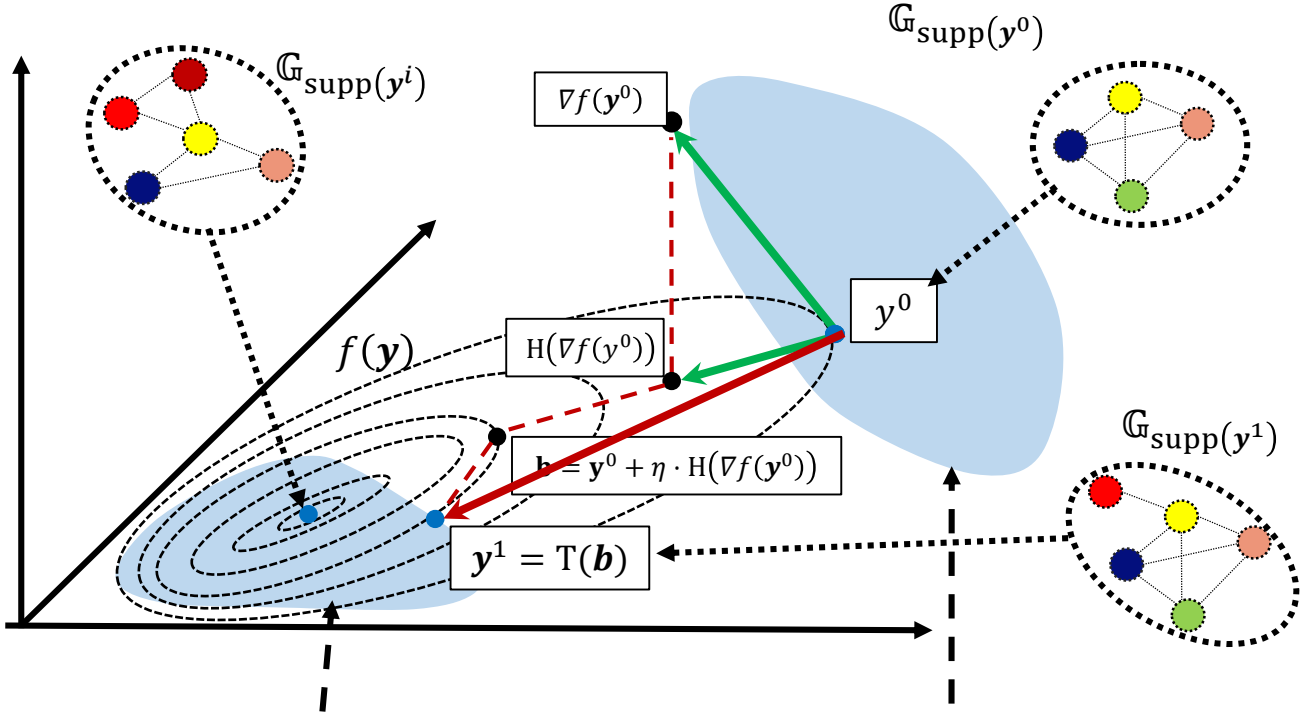
Illustration of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)





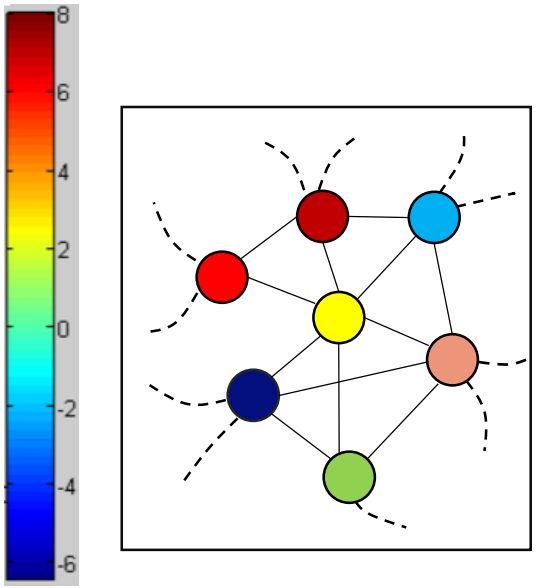
Network instance $\mathbb{G}$



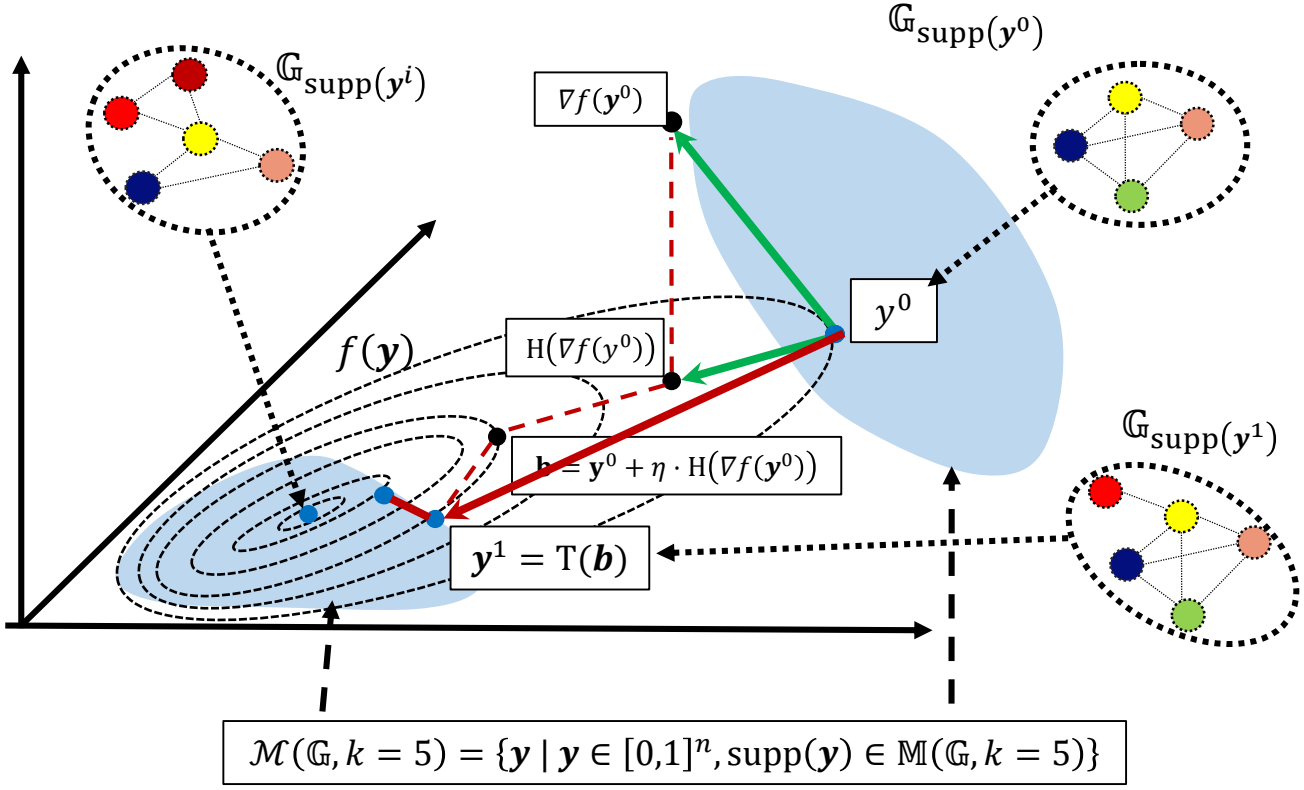
$$\mathcal{M}(\mathbb{G}, k = 5) = \{\boldsymbol{y} \mid \boldsymbol{y} \in [0,1]^n, \text{supp}(\boldsymbol{y}) \in \mathbb{M}(\mathbb{G}, k = 5)\}$$

Illustration of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)



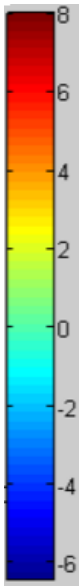
Network instance $\mathbb{G}$

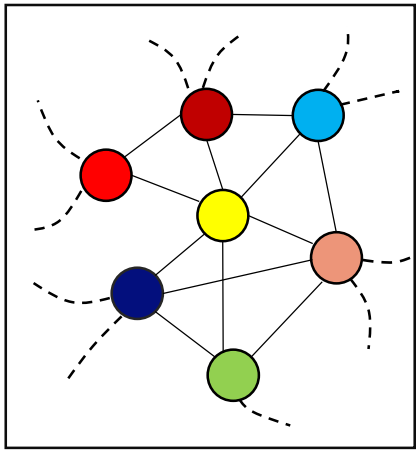


$$\mathcal{M}(\mathbb{G}, k = 5) = \{\boldsymbol{y} \mid \boldsymbol{y} \in [0,1]^n, \text{supp}(\boldsymbol{y}) \in \mathbb{M}(\mathbb{G}, k = 5)\}$$

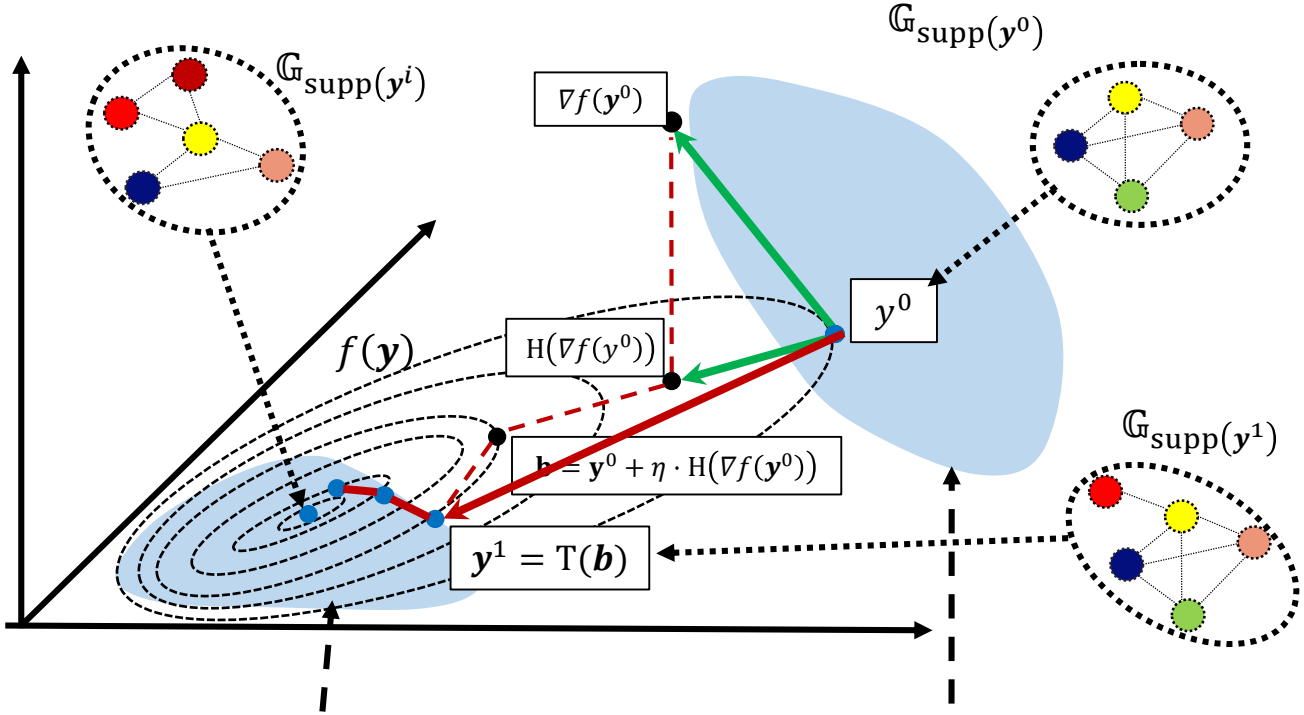
Illustration of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)





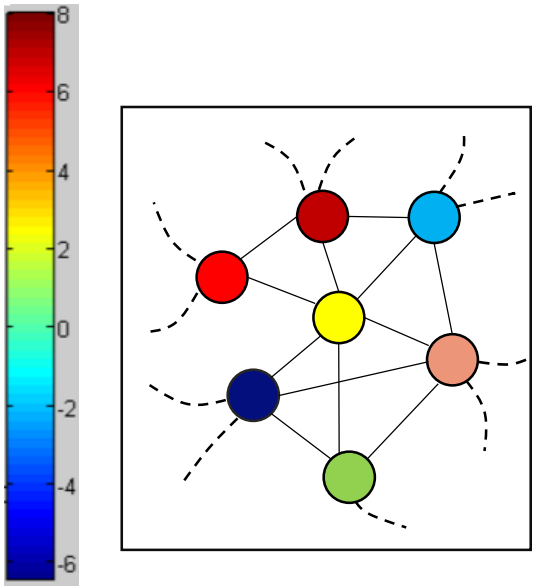
Network instance $\mathbb{G}$



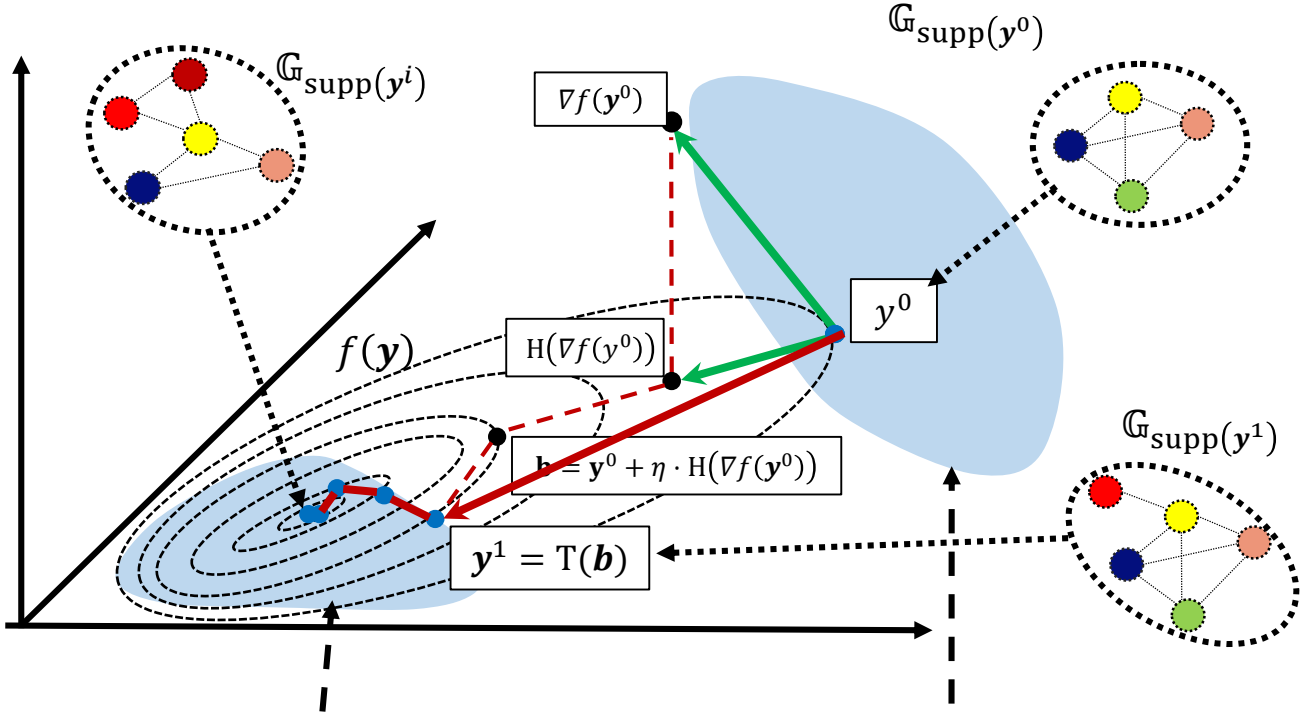
$$\mathcal{M}(\mathbb{G}, k = 5) = \{\boldsymbol{y} \mid \boldsymbol{y} \in [0,1]^n, \text{supp}(\boldsymbol{y}) \in \mathbb{M}(\mathbb{G}, k = 5)\}$$

Illustration of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)



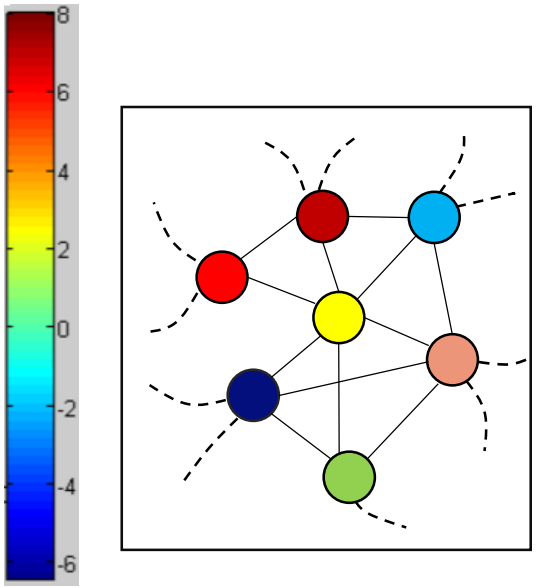
Network instance $\mathbb{G}$



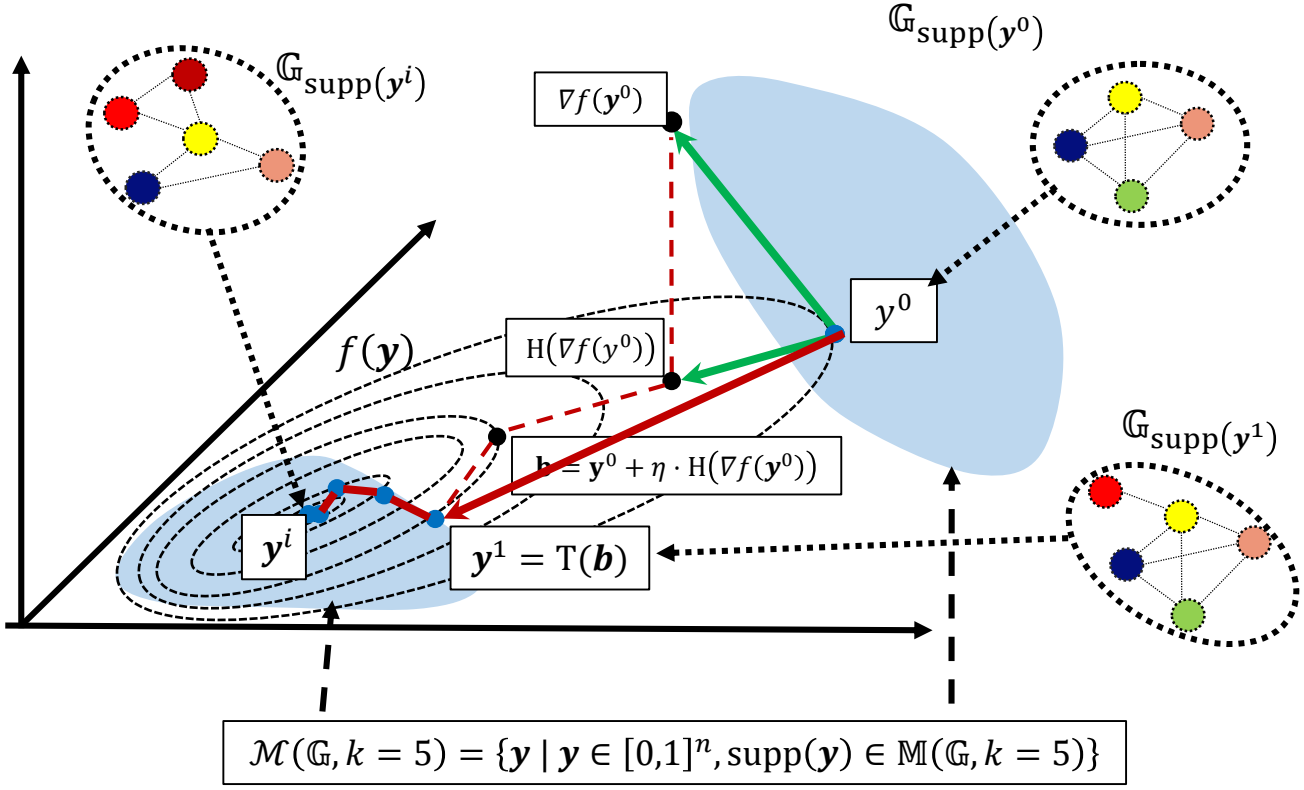
$$\mathcal{M}(\mathbb{G}, k = 5) = \{\boldsymbol{y} \mid \boldsymbol{y} \in [0,1]^n, \text{supp}(\boldsymbol{y}) \in \mathbb{M}(\mathbb{G}, k = 5)\}$$

Illustration of the Graph-IHT algorithm

(Zhou and Chen, ICDM, 2016)



Network instance $\mathbb{G}$



Theoretical Guarantees

(Zhou and Chen, ICDM, 2016)

- The proposed algorithms have the following nice theoretical properties
  - Nearly-linear time complexity.
  - Let $\boldsymbol{y}^\star$ be the optimal solution of the relaxed problem. Under practical assumptions, we have the tight error bound

$$\left\|\boldsymbol{y}^\star - \boldsymbol{y}^i\right\|_2 \leq c \cdot \|\nabla_I f(\boldsymbol{y}^\star)\|_2$$

    where
    - c is a constant value, and
    - $I = \arg\max_S \|\nabla_S f(\boldsymbol{y}^\star)\|_2$ s.t. S satisfies the predefined topological constraint.

Experiments

- Four real datasets for anomalous subgraph detection

Dataset	# of Nodes	# of Edges	# of snapshots
BWSN	12,527	14831	hourly: 8
CiteHepPh	11,895	76,284	yearly: 11
RoadTraffic	1,723	5,301	per-15-min: 68×304
ChicagoCrime	46,357	168,020	yearly: 15

Comparison on scores of the identified subgraphs

	BWSN				CitHepPh			
	Kulldorff	EMS	EBP	Run Time	Kulldorff	EMS	EBP	Run Time
GRAPH-GHTP	**1097.15**	**21.56**	**79.71**	**165.86**	16296.40	**337.90**	**9342.94**	155.74
GraphLaplacian	474.96	14.89	49.91	55315.94	2585.44	202.38	2305.05	22424.24
EventTree	834.59	20.25	32.13	441.74	**16738.43**	335.34	9061.56	**124.28**
DepthFirstGraphScan	735.85	20.41	79.30	5929.00	9531.19	260.06	5561.66	12183.88
NPHGS	541.13	16.90	58.59	256.91	11965.14	326.23	9098.22	175.08

	Traffic		ChicagoCrime			
	EMS	Run Time	Kulldorff	EMS	EBP	Run Time
GRAPH-GHTP	**20.45**	22.25	**6386.08**	**5.45**	**5172.54**	3177.60
GraphLaplacian	5.40	291.75	-	-	-	-
EventTree	12.40	5.02	4388.42	4.91	3965.96	**226.50**
DepthFirstGraphScan	8.13	47.73	1123.49	2.56	1094.21	12133.50
NPHGS	6.28	**0.22**	966.70	2.43	948.23	701.40

Review of other methods

- Scalable anomaly ranking of attributed neighborhoods (Perozzi and Akoglu, SDM, 2016)
  - Rank a predefined set of neighborhoods (subgraphs) based on internal connectivity, boundary, and node-level attributes in quadratic time in the neighborhood size.
- Focused cluster or subgraph outlier detection (Perozzi et al., KDD, 2016)
  - Given an initial set of nodes provided by a user
  - Step 1: Identify a subset of attributes that the given nodes agree on (called “focus attributes”)
  - Step 2: Find densely connected subgraphs that also agree on these attributes (called “focused clusters”)

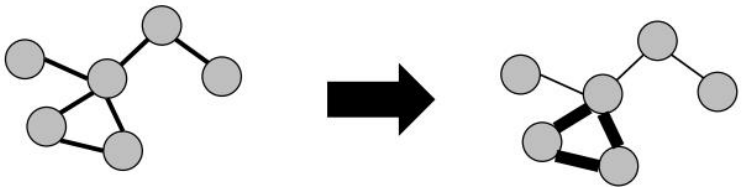
Focused subgraph outlier detection

(Perozzi et al., KDD, 2016)

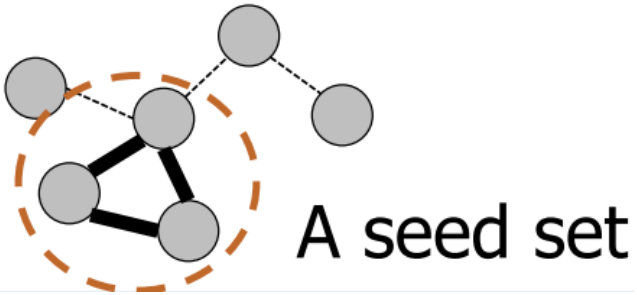
- Finding nodes to cluster around

for each $(i,j) \in E$ **do**

$$w(i,j) = 1/(1+\sqrt{(\boldsymbol{f_i}-\boldsymbol{f_j})^T diag(\boldsymbol{\beta})(\boldsymbol{f_i}-\boldsymbol{f_j})})$$



- Highly weighted edges are reserved
- The connected components are considered as seeds

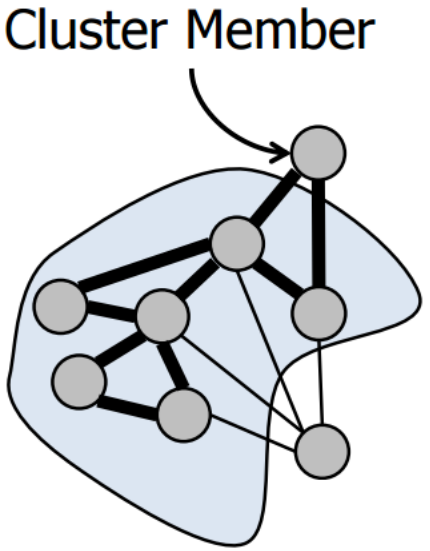


Focused subgraph outlier detection

(Perozzi et al., KDD, 2016)

1. Clustering objective: subgraph conductance weighted by focus

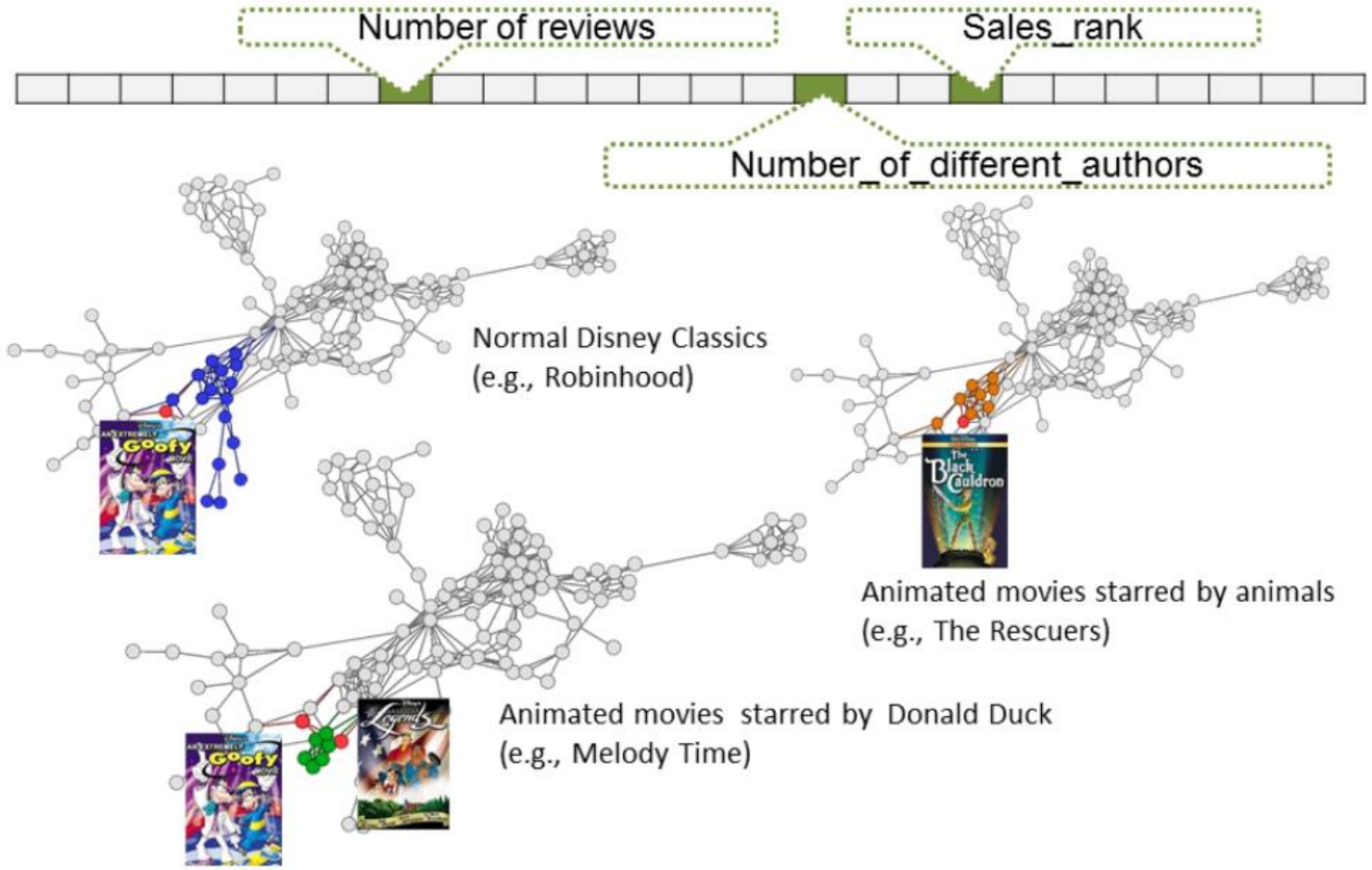
$$F(S) = \frac{\text{WeightedOutDegree}(S)}{\text{WeightedDensity}(S)}$$



2. At each edge in subgraph expansion
    1. Examine boundary nodes
    2. Add node with the best marginal gain

Disney: amazon co-purchase network

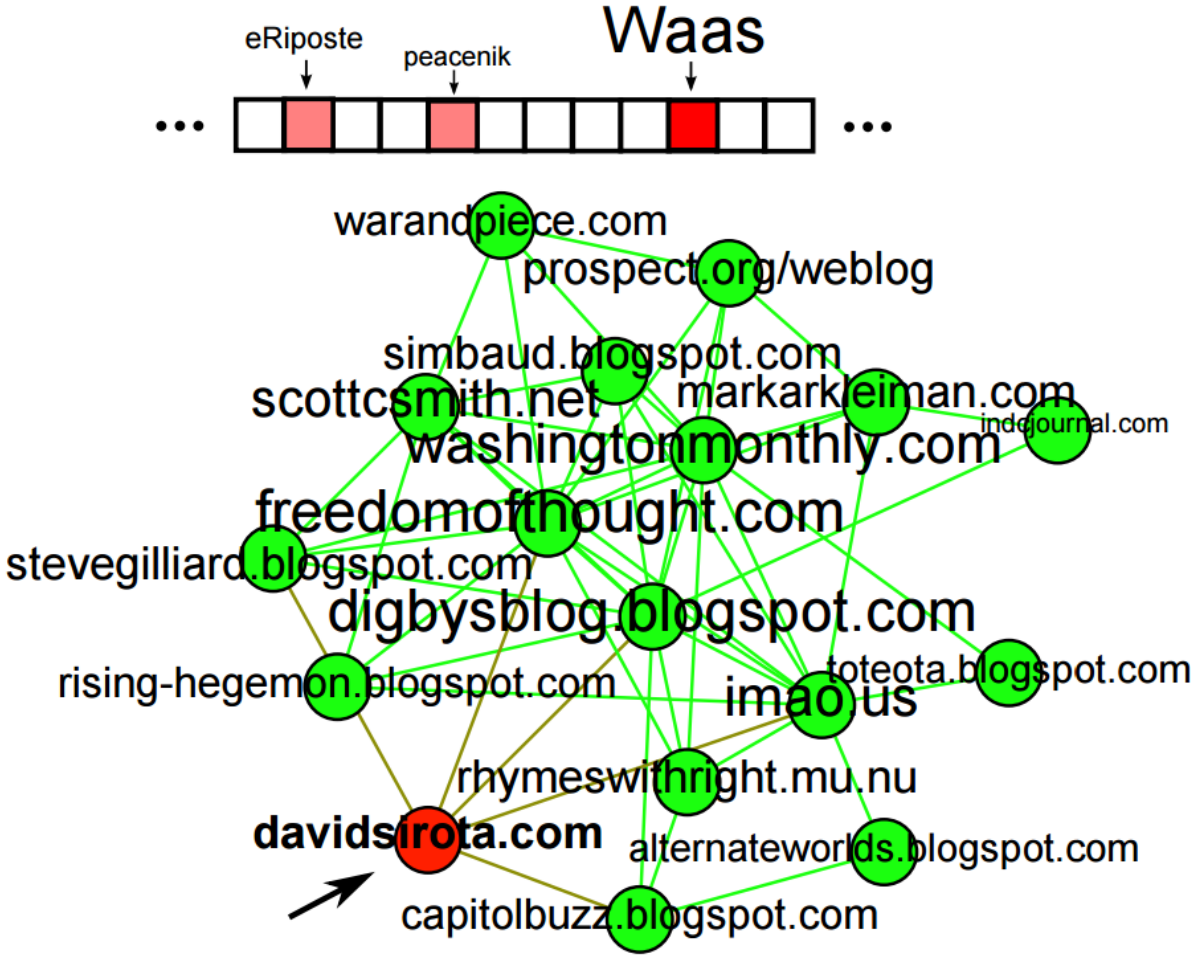
(Perozzi et al., KDD, 2016)



The detected subgraphs focus on attributes related to popularity (sales rank, number of reviews, etc)

Political blogs citation network

(Perozzi et al., KDD, 2016)



A focused cluster of liberal blogs in Pol-Blogs with a focus on Iraq ware debate

Part I: References

- Kulldorff, M. (1997). A spatial scan statistic. Communications in Statistics-Theory and methods, 26(6), 1481-1496.
- Neill, D. B., & Moore, A. W. (2005, August). Anomalous spatial cluster detection. In Proceedings of the KDD 2005 Workshop on Data Mining Methods for Anomaly Detection.
- Neill, D. B. (2012). Fast subset scan for spatial pattern detection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 74(2), 337-360.
- Speakman, S., McFowland III, E., & Neill, D. B. (2015). Scalable detection of anomalous patterns with connectivity constraints. Journal of Computational and Graphical Statistics, 24(4), 1014-1033.
- Speakman, S., Somanchi, S., McFowland III, E., & Neill, D. B. (2016). Penalized fast subset scanning. Journal of Computational and Graphical Statistics, 25(2), 382-404.
- Speakman, S., Zhang, Y., & Neill, D. B. (2013, December). Dynamic pattern detection with temporal consistency and connectivity constraints. In 2013 IEEE 13th International Conference on Data Mining (pp. 697-706). IEEE.

Part I: References

- Chen, F., & Neill, D. B. (2014, August). Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1166-1175). ACM.
- Chen, F., & Neill, D. B. (2015). Human rights event detection from heterogeneous social media graphs. Big Data, 3(1), 34-40.
- Rozenshtein, P., Anagnostopoulos, A., Gionis, A., & Tatti, N. (2014, August). Event detection in activity networks. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining(pp. 1176-1185). ACM.
- Chen, F., & Zhou, B. (2016). A Generalized Matching Pursuit Approach for Graph-Structured Sparsity. In Proc. IJCAI (pp. 1389-1395).
- Zhou, B., & Chen, F. (2016). Graph-Structured Sparse Optimization for Connected Subgraph Detection. In Proc. ICDM (to appear).
- Buchbinder, N., Feldman, M., Naor, J. S., & Schwartz, R. (2012, October). A Tight Linear Time (1/2)-Approximation for Unconstrained Submodular Maximization. In *Proc.* FOCS (pp. 649-658).

Part I: References

- Neill, D. B., McFowland, E., & Zheng, H. (2013). Fast subset scan for multivariate event detection. Statistics in medicine, 32(13), 2185-2208.
- Neill, D. B., & Cooper, G. F. (2010). A multivariate Bayesian scan statistic for early event detection and characterization. Machine learning, 79(3), 261-282.
- Perozzi, B., & Akoglu, L. (2015). Scalable anomaly ranking of attributed neighborhoods. In Proc. SDM, 207-215.
- Perozzi, B., Akoglu, L., Iglesias Sánchez, P., & Müller, E. (2014). Focused clustering and outlier detection in large attributed graphs. In Proc. KDD, 1346-1355.
- Akoglu, L., Tong, H., & Koutra, D. (2015). Graph based anomaly detection and description: a survey. Data Mining and Knowledge Discovery, 29(3), 626-688.
- Bindu, P. V., & Thilagam, P. S. (2016). Mining social networks for anomalies: Methods and challenges. Journal of Network and Computer Applications, 68, 213-229.

Part I: References

- Kuo, T. W., Lin, K. C. J., & Tsai, M. J. (2015). Maximizing submodular set function with connectivity constraint: Theory and application to networks. IEEE/ACM Transactions on Networking (TON), 23(2), 533-546.
- Hegde, C., Indyk, P., & Schmidt, L. (2015). A nearly-linear time framework for graph-structured sparsity. In Proceedings of the 32nd International Conference on Machine Learning (ICML-15) (pp. 928-937).
- Chuang, H. Y., Lee, E., Liu, Y. T., Lee, D., & Ideker, T. (2007). Network-based classification of breast cancer metastasis. Molecular systems biology, 3(1), 140.
- de Oliveira, D. P., Neill, D. B., Garrett Jr, J. H., & Soibelman, L. (2010). Detection of patterns in water distribution pipe breakage using spatial scan statistics for point events in a physical network. Journal of Computing in Civil Engineering, 25(1), 21-30.

5 minutes break: Q/A