Anomalous and Significant Subgraph Detection in Attributed Networks

Part II: Dynamic 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

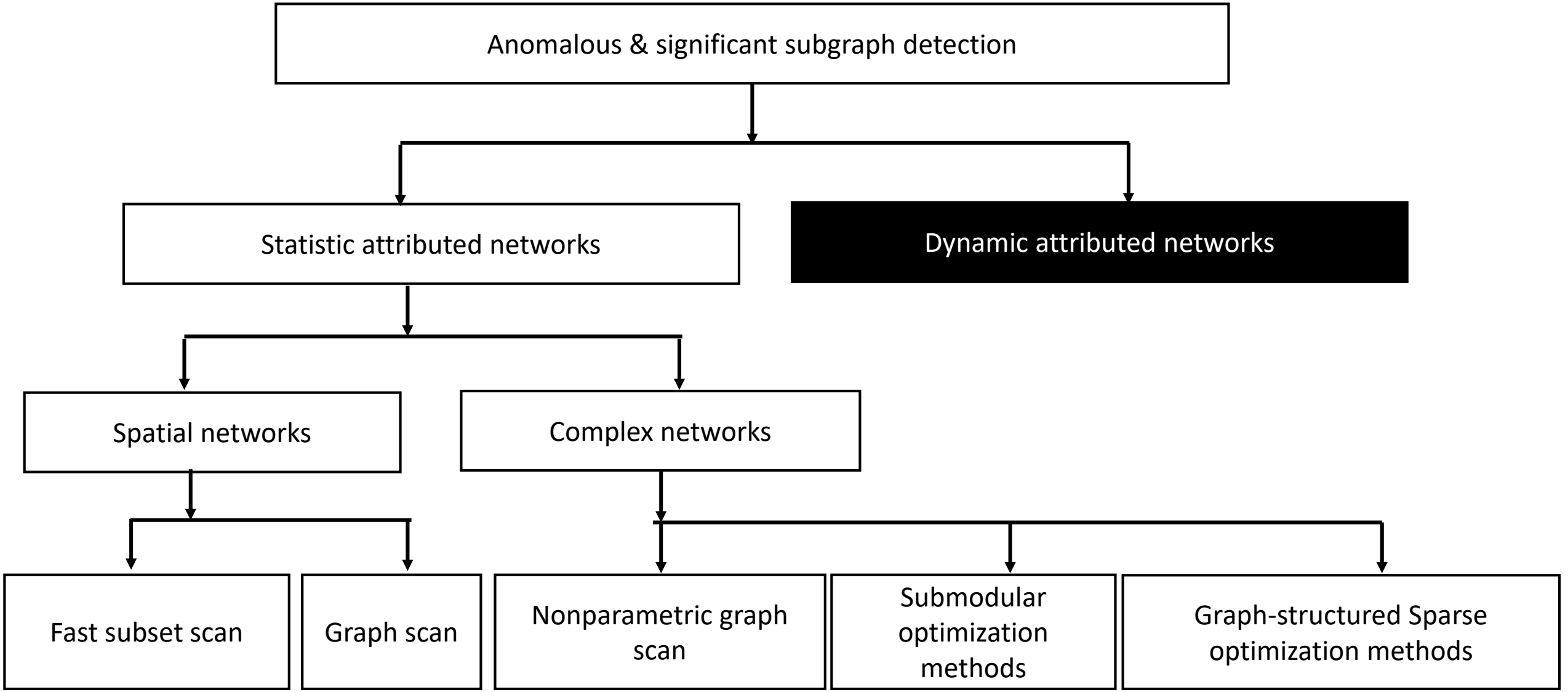
[2] H.J. Heinz III College
Carnegie Mellon University

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

Roadmap

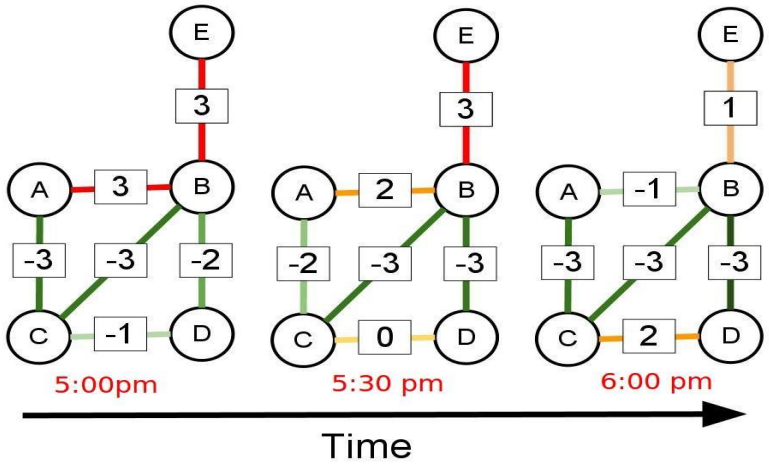
- Introduction & Motivation
- Part 1: Subgraph Detection in Statistic Attributed Networks
- **Part 2: Subgraph Detection in Dynamic Attributed Networks**
- Future Directions

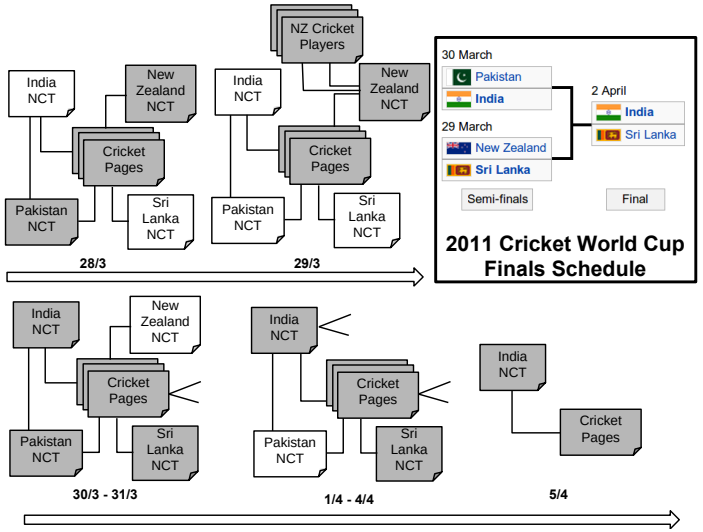
Taxonomy


Anomalous & significant subgraph detection
Statistic 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

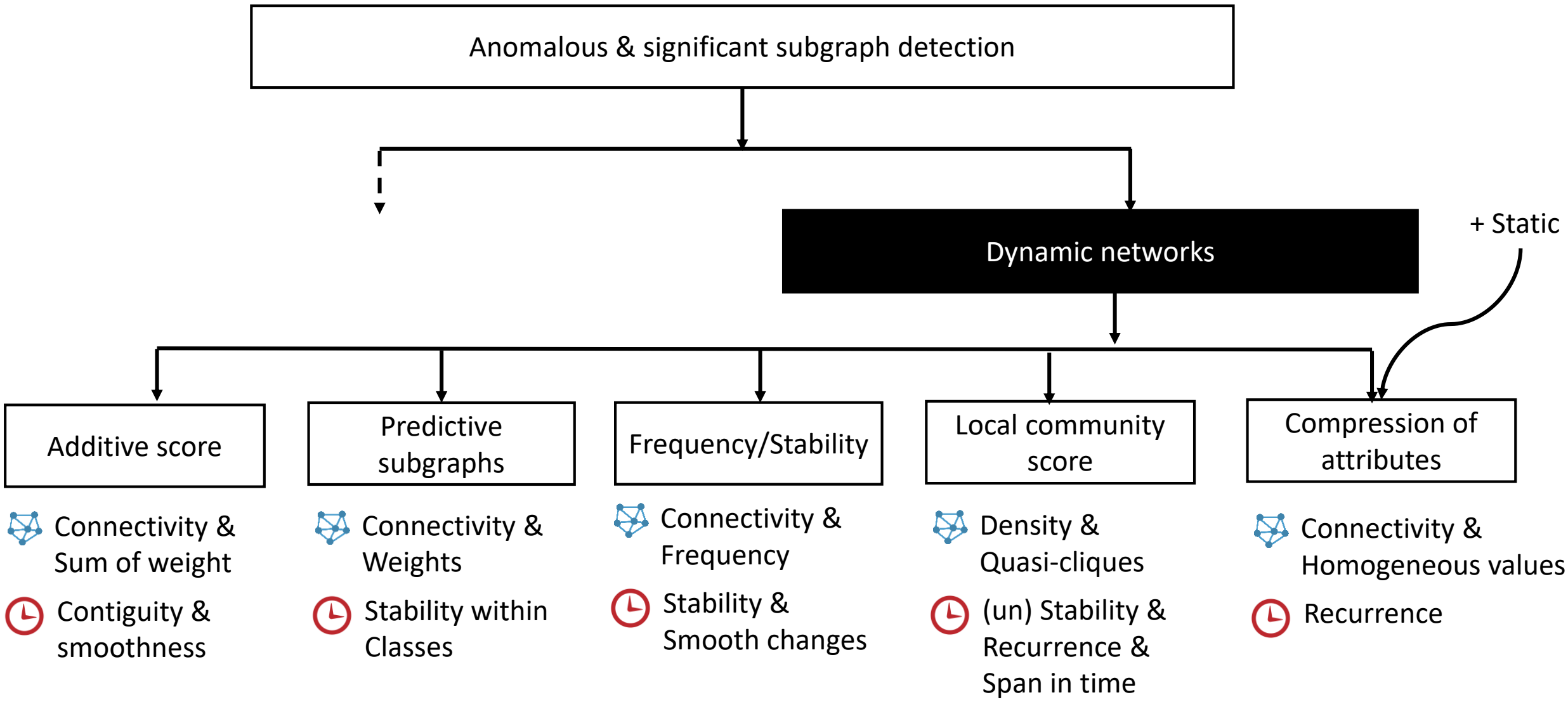
Specifics of the dynamic setting

- Graph and temporal (multi-network) dimensions
  - **Graph:** Connectivity/Density/Weight
  - **Time:** Contiguity/Recurrence/Stability
- Baseline: Independent snapshot detection and then connect in time
  - (-) Do not consider all slices at a time
  - (-) Heuristic post-processing across time
- Focus for this part: **detection of general subgraphs**
- What we will **NOT** address
  - Global network properties over time
  - Individual node/edge properties over time
  - Evolutionary clustering: partitions the full network as opposed to focus on specific subgraphs
  - Static structure: e.g. community detection

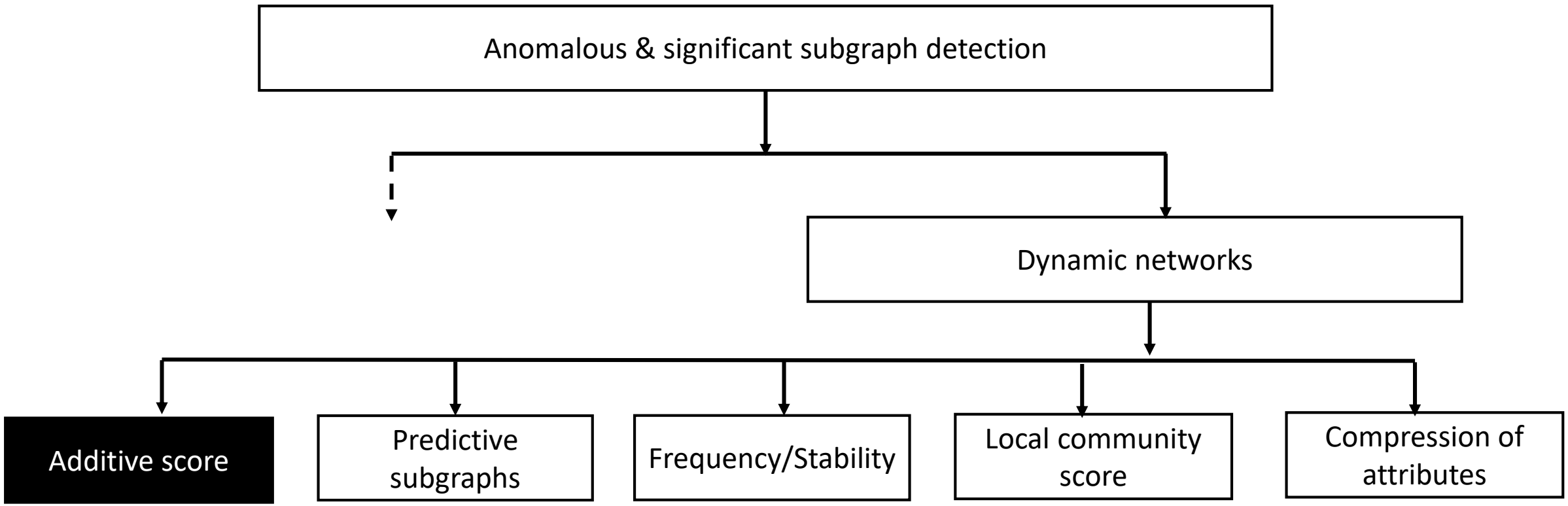




Taxonomy


Anomalous & significant subgraph detection
Dynamic networks
+ Static
Additive score
Predictive subgraphs
Frequency/Stability
Local community score
Compression of attributes
Connectivity & Sum of weight
Contiguity & smoothness
Connectivity & Weights
Stability within Classes
Connectivity & Frequency
Stability & Smooth changes
Density & Quasi-cliques
(un) Stability & Recurrence & Span in time
Connectivity & Homogeneous values
Recurrence

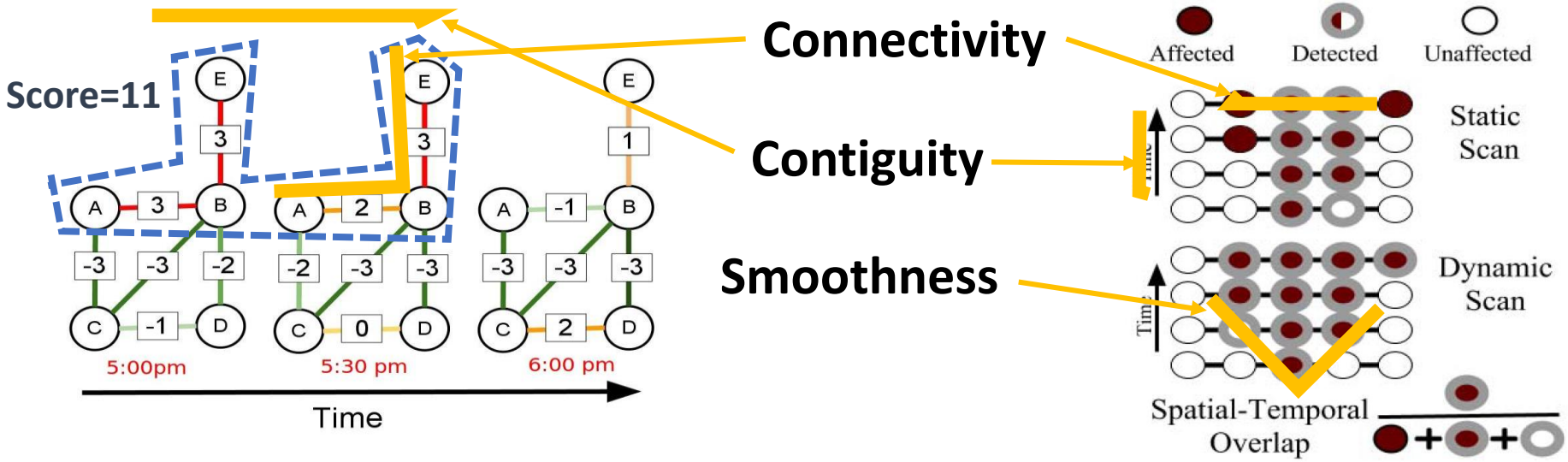
Taxonomy



Connectivity & Sum of weight

Contiguity & smoothness

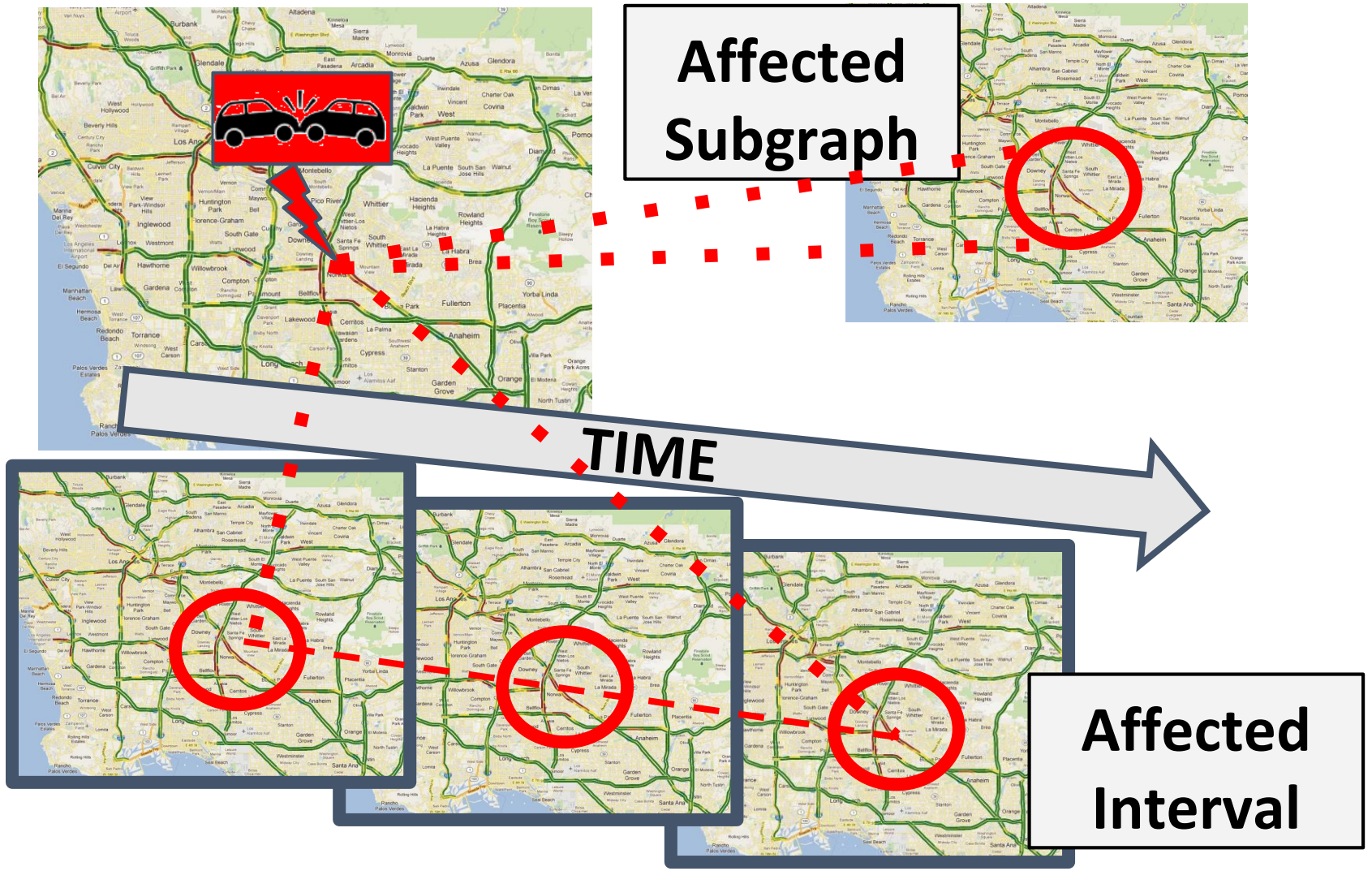
Additive attribute score



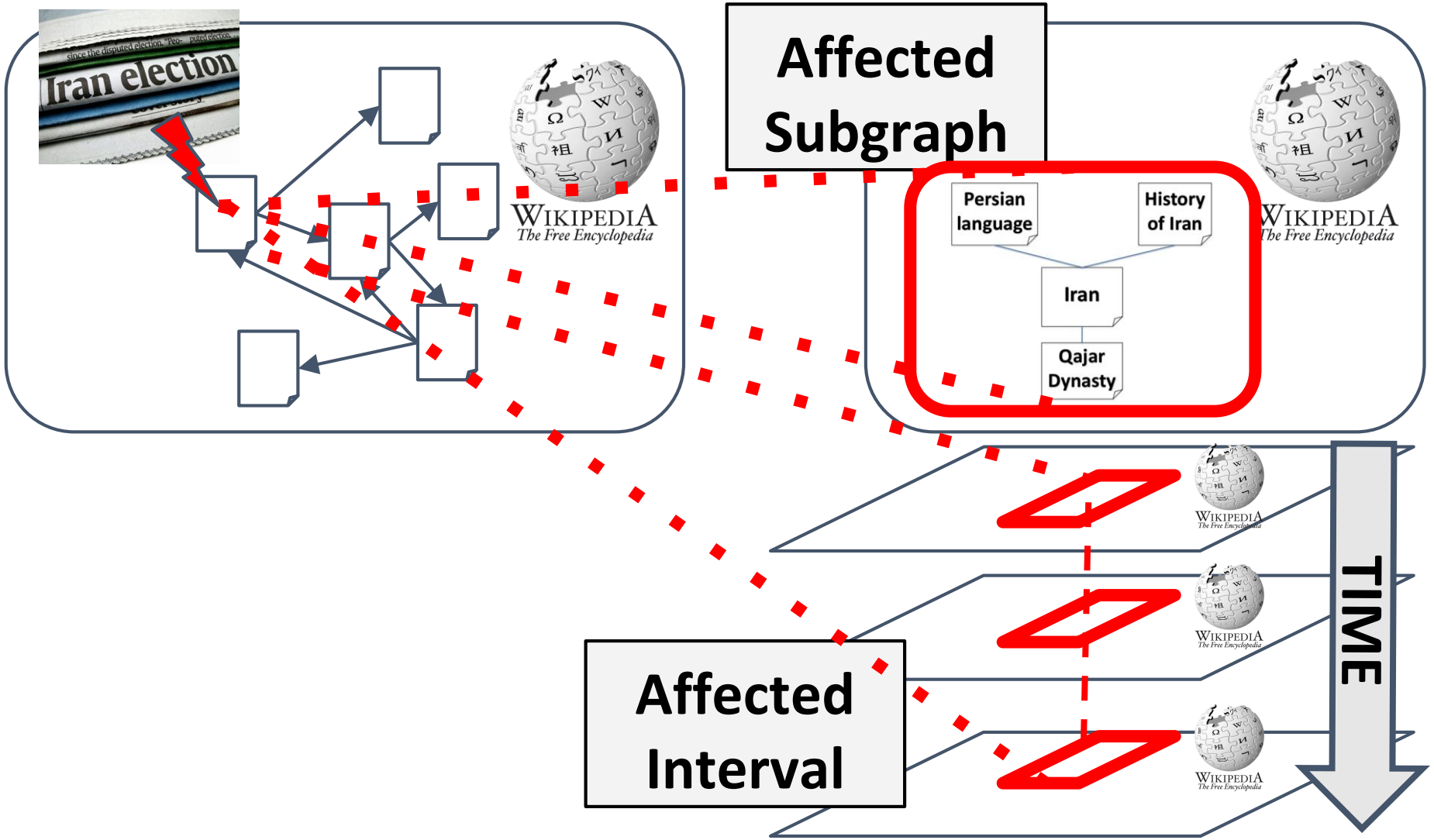
- Signed weights on elements
- Score = sum of weights
- Subgraphs of interest:
  - stable or smoothly evolving
  - connected

- Sensor readings
- Score = spatio-temporal scan statistic
- Subgraphs:
  - stable or smoothly evolving
  - connected

Dynamic processes: traffic jams

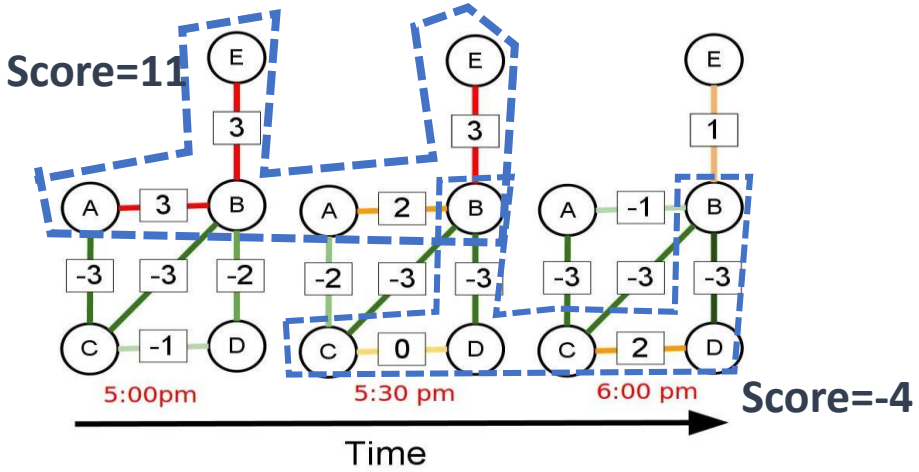


Dynamic processes: fact search



Mining heavy dynamic subgraphs

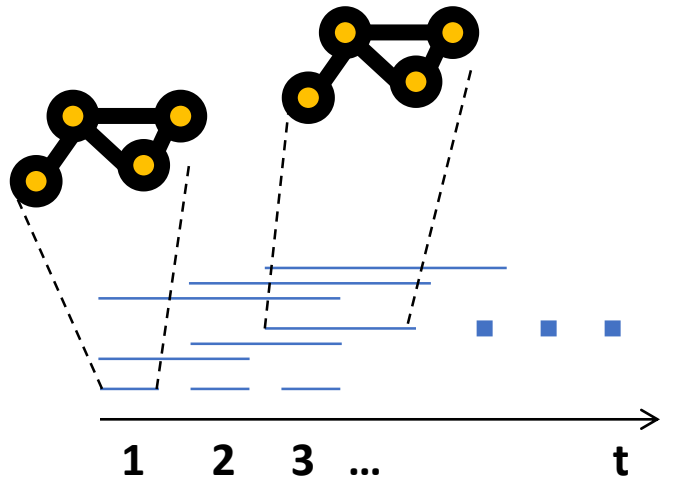
[Bogdanov et al. ICDM'11]

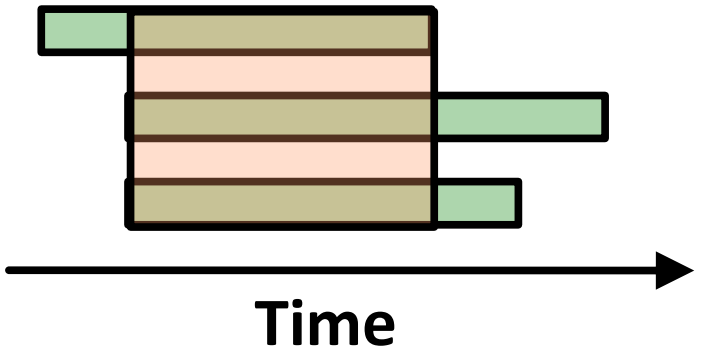


- Node/edge scores change over time, structure fixed
- Positive scores correspond to anomaly/event of interest
- Dynamic subgraph: (subgraph, interval)
- **Heaviest Dynamic Subgraph** of maximum score
  - NP-hard (Single slice: **HS ⟷ PCST**)
  - Large search space**: (connected subgraphs) x (t^2 intervals)**

Baseline filter-and-verify approach

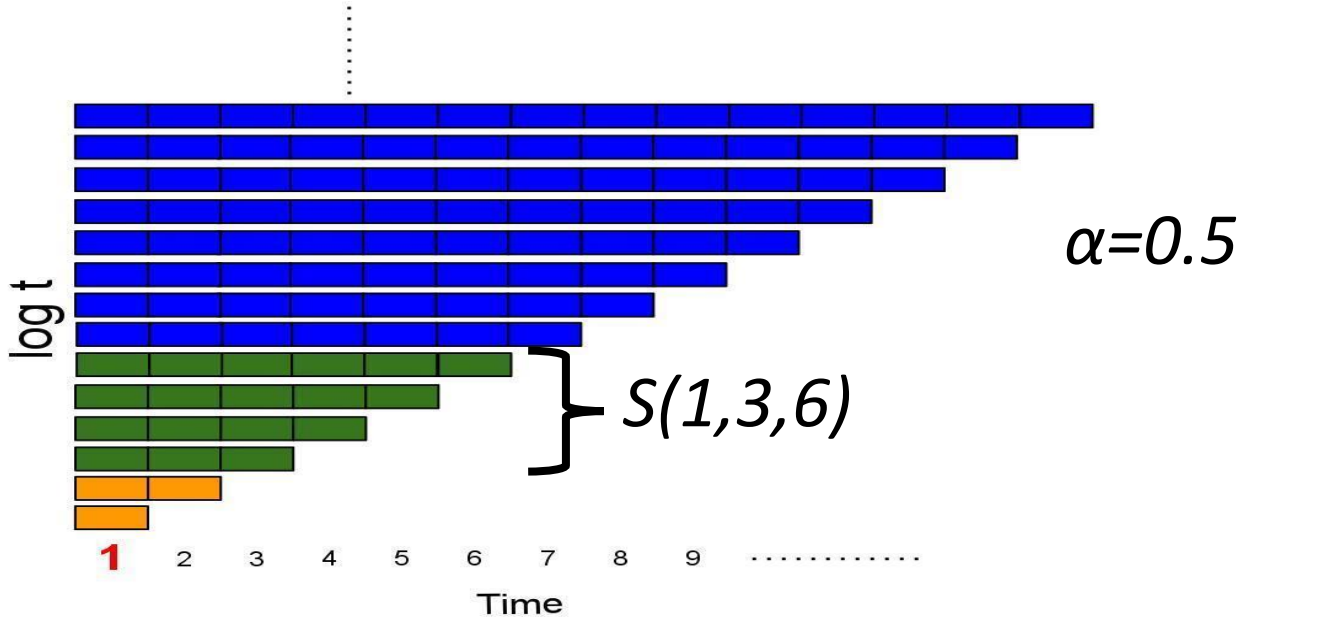
- Discard “bad” intervals
  - UB on score O(|E|)
    - Relax connectivity constraints to obtain UB
  - Discard intervals if UB < global LB
- To obtain a LB and evaluate intervals
  - Transform to PCST-Tree and solve exactly
  - Faster than existing PCST heuristics *[Johnson’00]*
- Challenge: Quadratic filtering cost O($\mathbf{t^2}$ |E|)
  - Need to consider every interval
  - Traffic: 1 month@5 min -> **75 mil. aggregated graphs**
- **Idea:** Filter at a coarser temporal resolution
  - Combine overlapping intervals into groups
  - High overlap -> similar solutions
  - Filter whole groups at a time





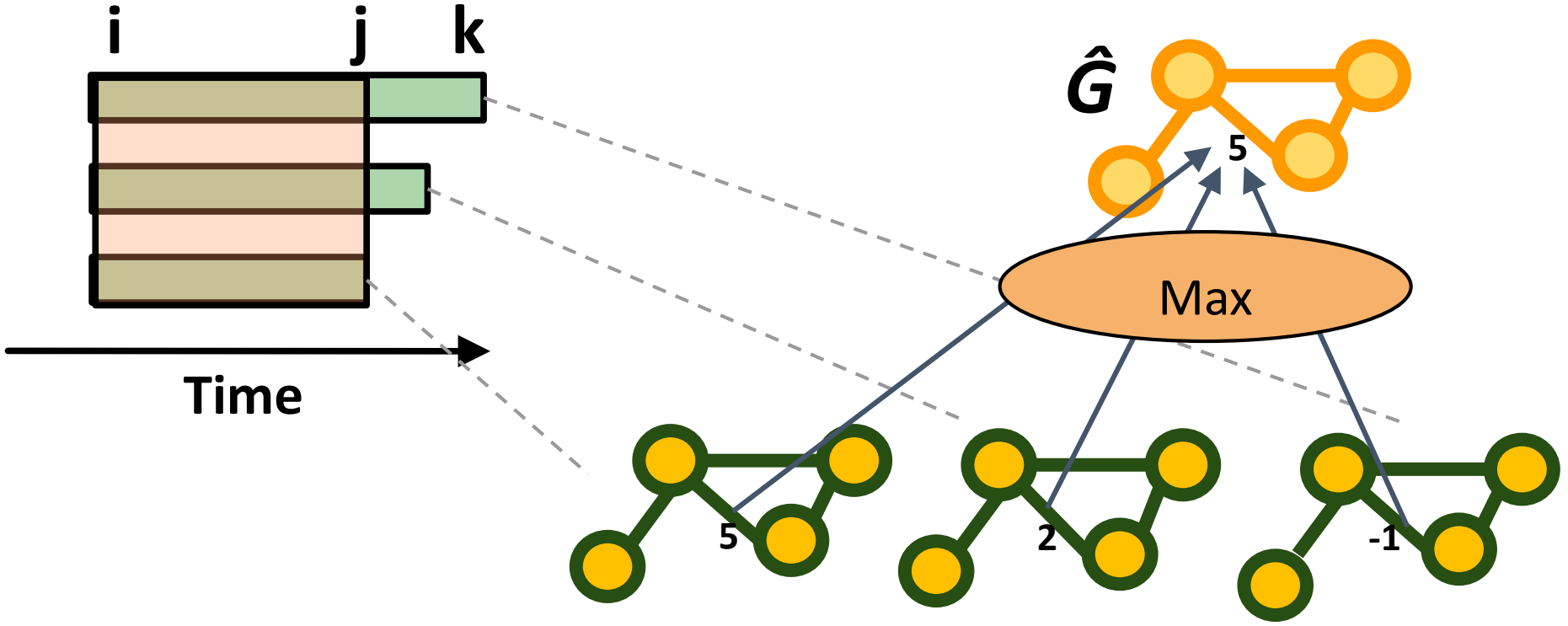
Left-Aligned Interval Grouping

- Left-aligned group *S(i,j,k)*: intervals that start at time *i* and end at time *[j,k], i≤j≤k*
- Minimum overlap α=(j-i+1)/(k-i+1): The length ratio of shortest and longest interval
- *If* ***0<α<1****, then* ***#groups = O(t log(t))***

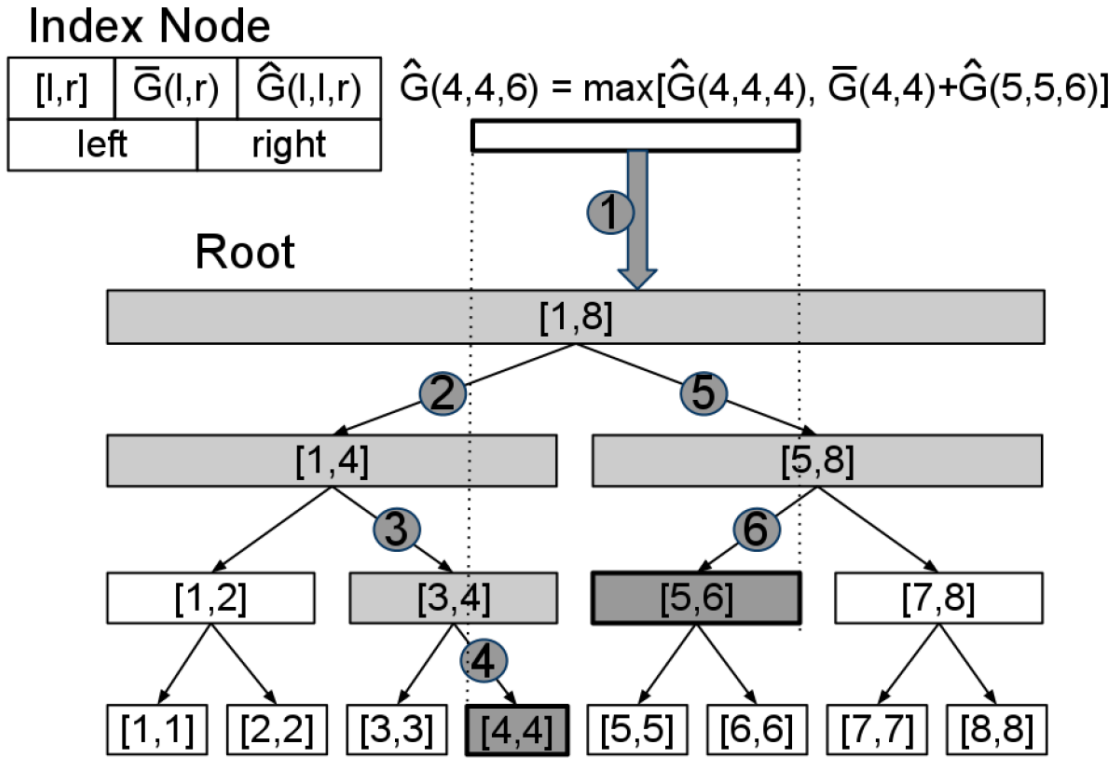


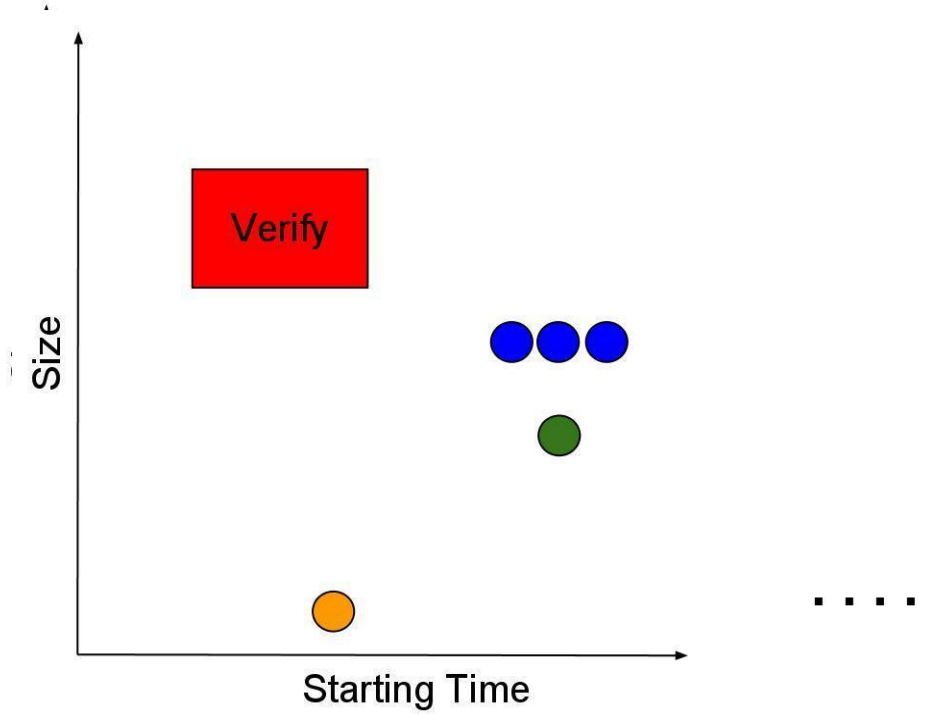
How to Filter Whole Groups?

- Dominating graph ***Ĝ(i,j,k)*** for a group ***S(i,j,k)***
  - Weights: maximum in any grouped interval
  - HS($\hat{G}(i,j,k)$) ≥ $HS(G(i,l))$, $j \le l \le k$



Filter in $O(t\log^2(t)|E|)$



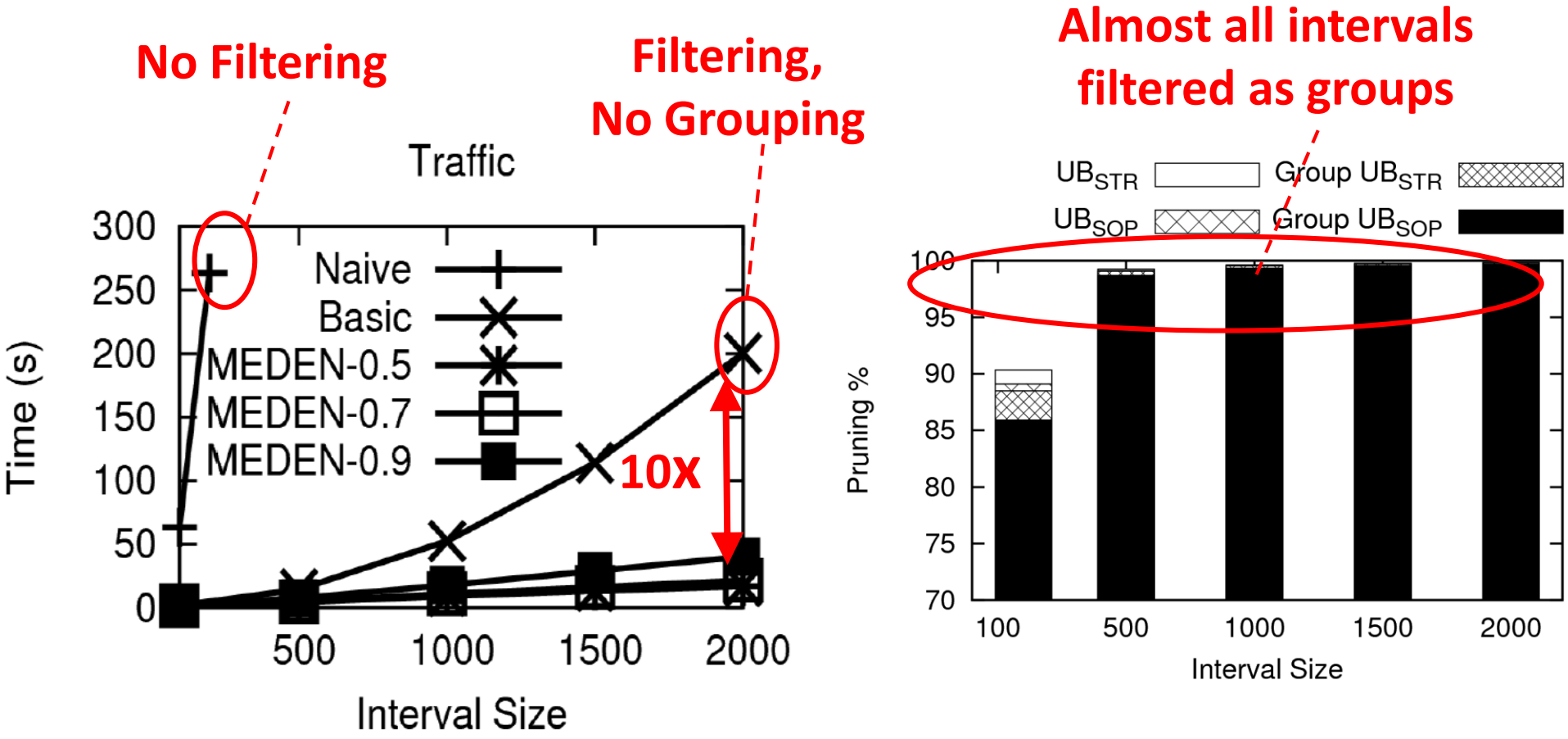


Time to build index **O(t|E|)**, Bottom-up

Compose one ***Ĝ*** *in* **O(log(t)|E|)**

Compute all ***Ĝ*** *in* **O(t log²(t) |E|)** **vs O(t²|E|)**

Scalability

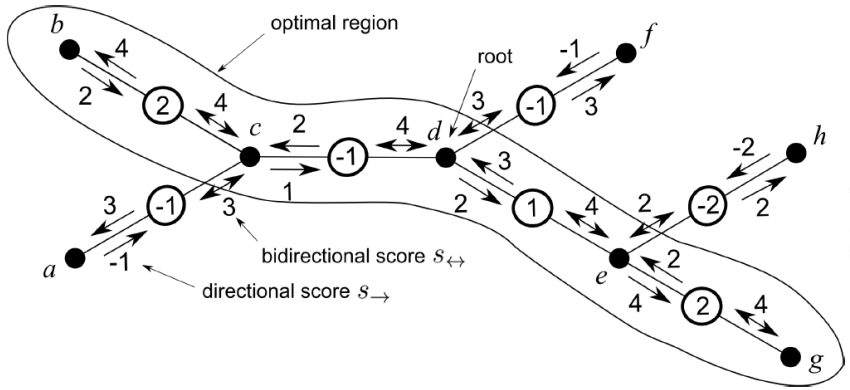


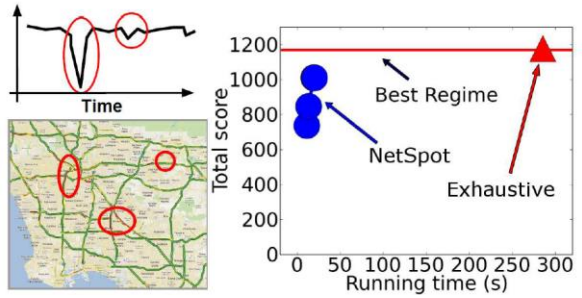
An order of magnitude speedup

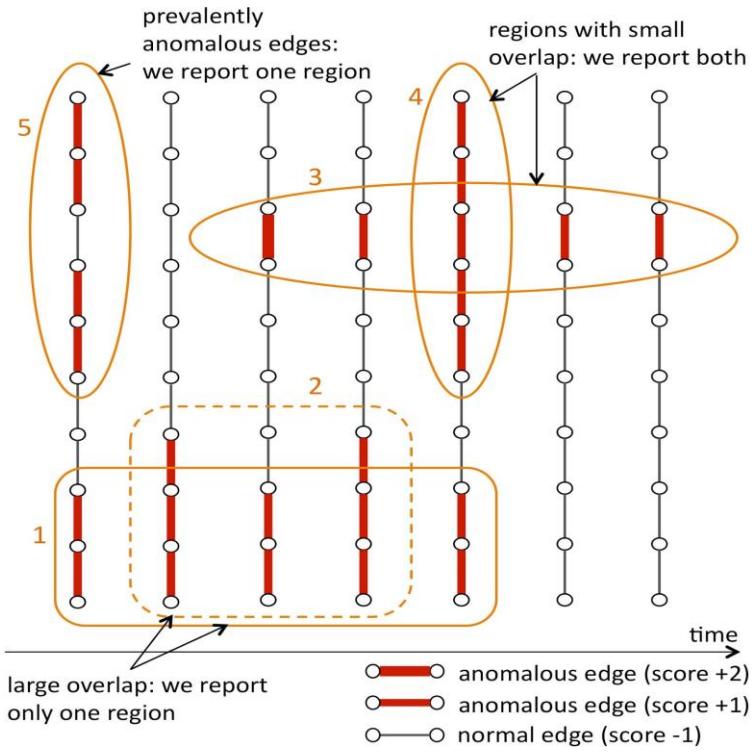
Identify All Significant Regions

[Mongiovi et al. SDM'13]

- Alternate between optimization in the temporal and graph domains
  - Fix a subgraph: compute the MS time subsequence
  - Fix a time interval: compute the Heaviest Subgraph
- Seeds: Heaviest subgraph, Max. Subsequence
  - Consider an augmented MST of every snapshot
  - Optimal edge-rooted scores for every edge in O(|E|)
  - Optimal edge-rooted interval
- Allow overlap

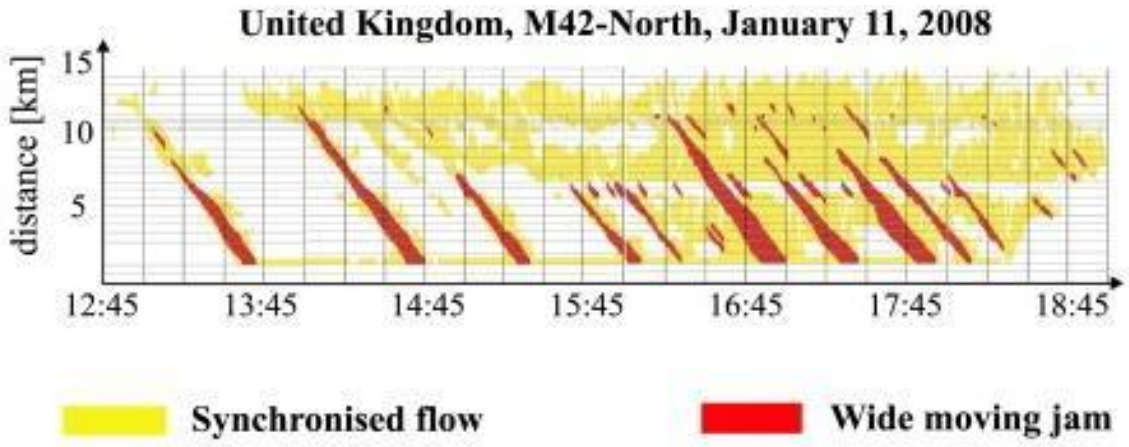


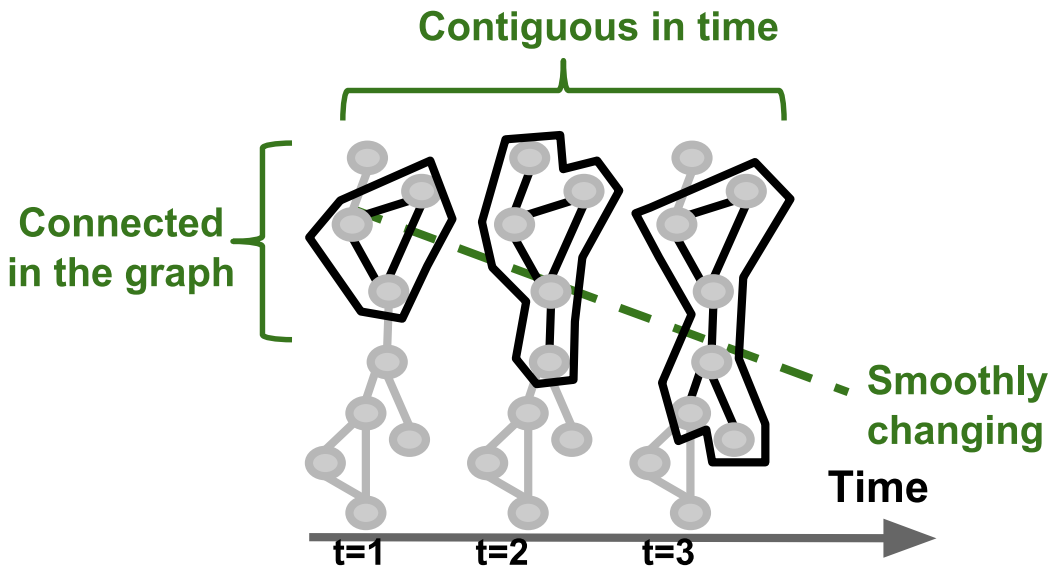




So far: “stable” solutions across time

- No change in participating nodes/edges
- Processes in real networks might shift and change in size
- Idea: incorporate smoothness
  - **As a constraint**: not too many nodes/edges change
  - **As a penalty**

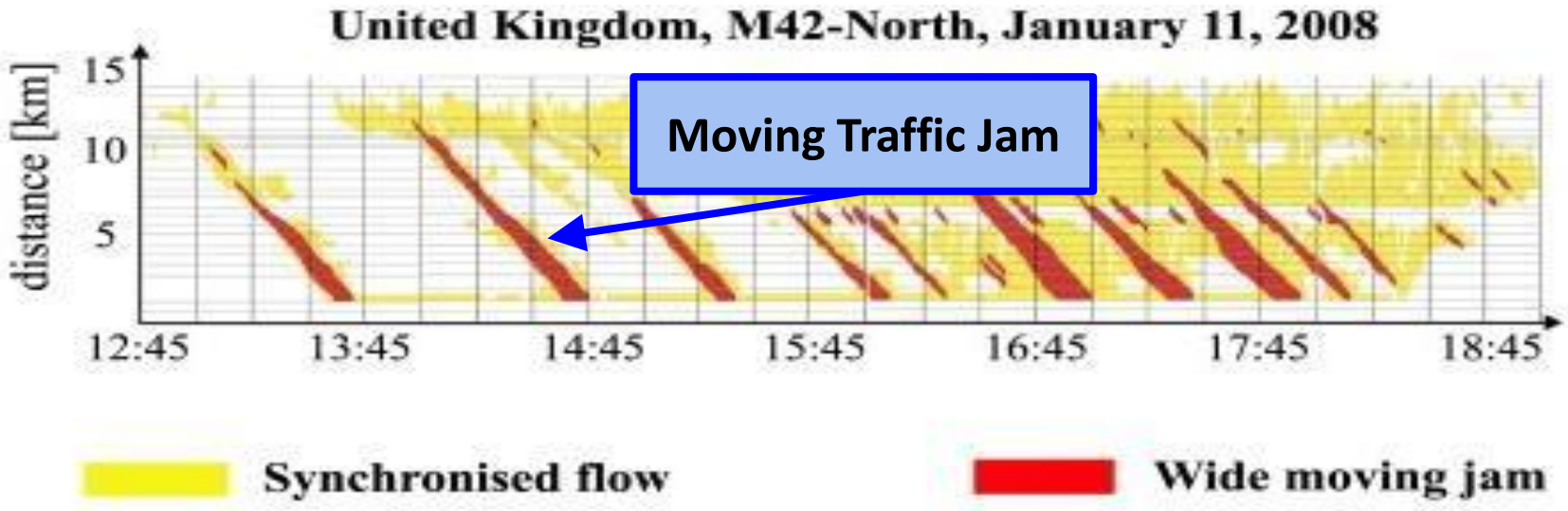




Smooth Network Processes

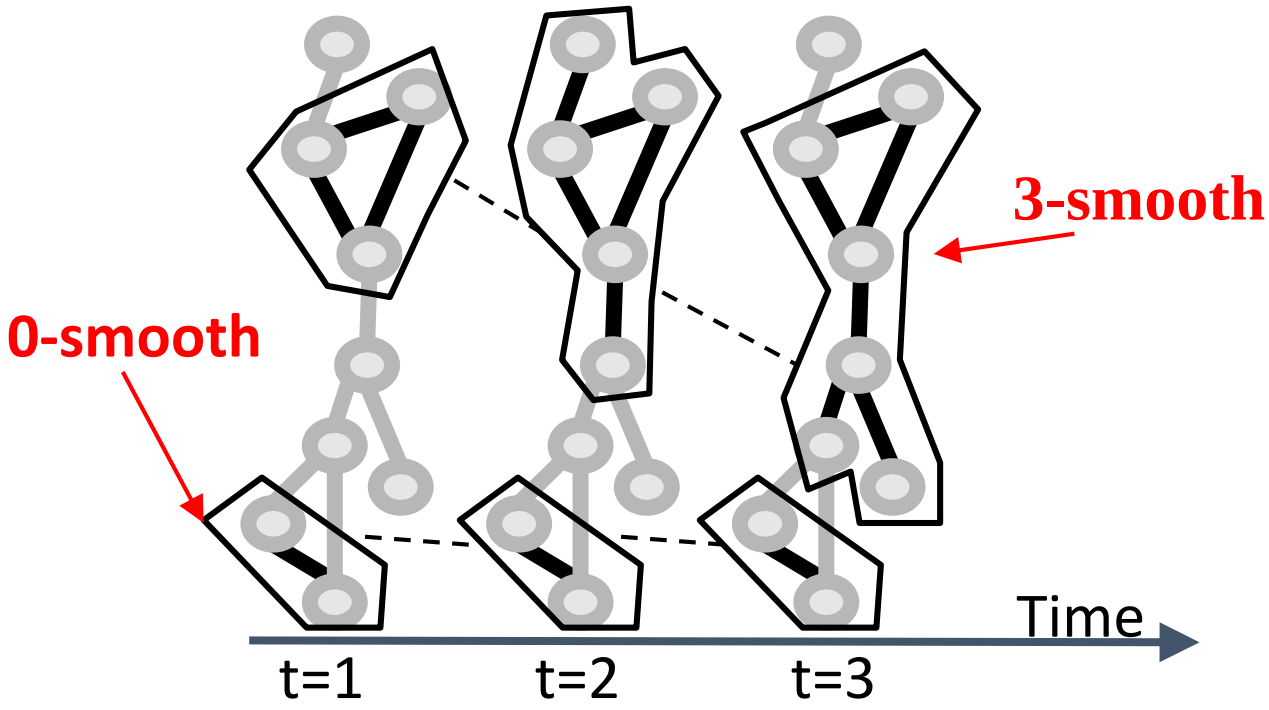
[Mongiovi et al. ICDM'13]

- Processes may evolve in the network
  - Traffic congestions grow, shift and shrink
  - Attention in information networks shifts with current events
- Allow the subgraph of interest to change smoothly in time (smoothness constraint)

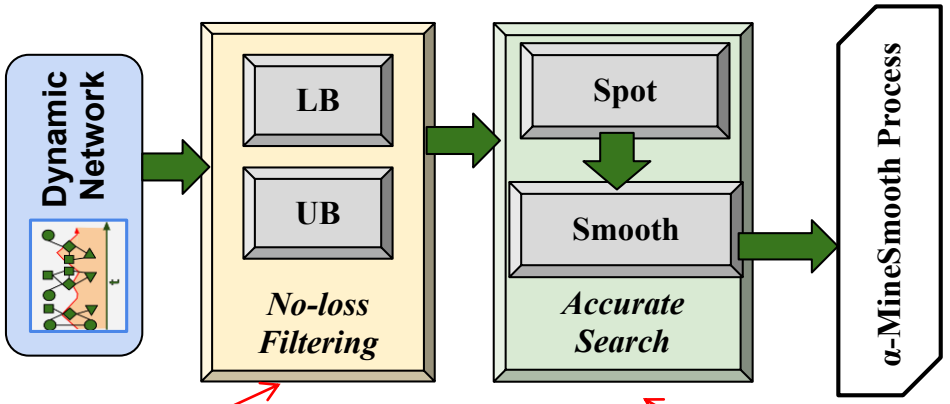


Smooth Temporal Subgraphs

- **Connected** in the graph structure
- **Contiguous** in time (no negative score time slices)
- **Smooth**: at most **α** edges/nodes change at a time
  - Similar smoothness criteria in evolutionary clustering *[Chakrabarti'06 ,Chi'07]*
- **Max-score α–smooth** subgraph**:** NP hard even on trees (reduction from 3SAT)
  - 0-smooth is polynomial on trees



Overview of Solution

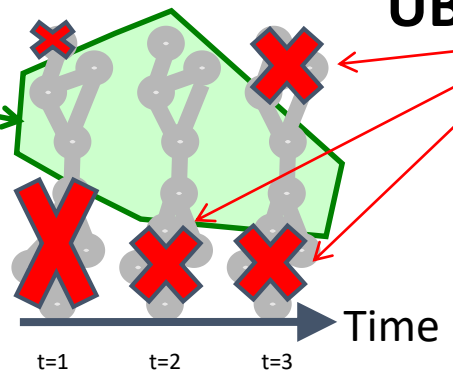


Reduce the instance size

Candidate search

UB(v,t) < LB

Remaining small and fragmented Instance (~5% of original)



Spot
∞-smooth candidates

"Smooth"
each candidate according to α

No Loss of Accuracy

Bounds in a single slice and across time

- Bottom
  - Tr
  - **Pc**
    d(
  - **UI**
  - **d(**
    of
- Grow
  d(C) ≥
- *Theore*
  solutio
  groups

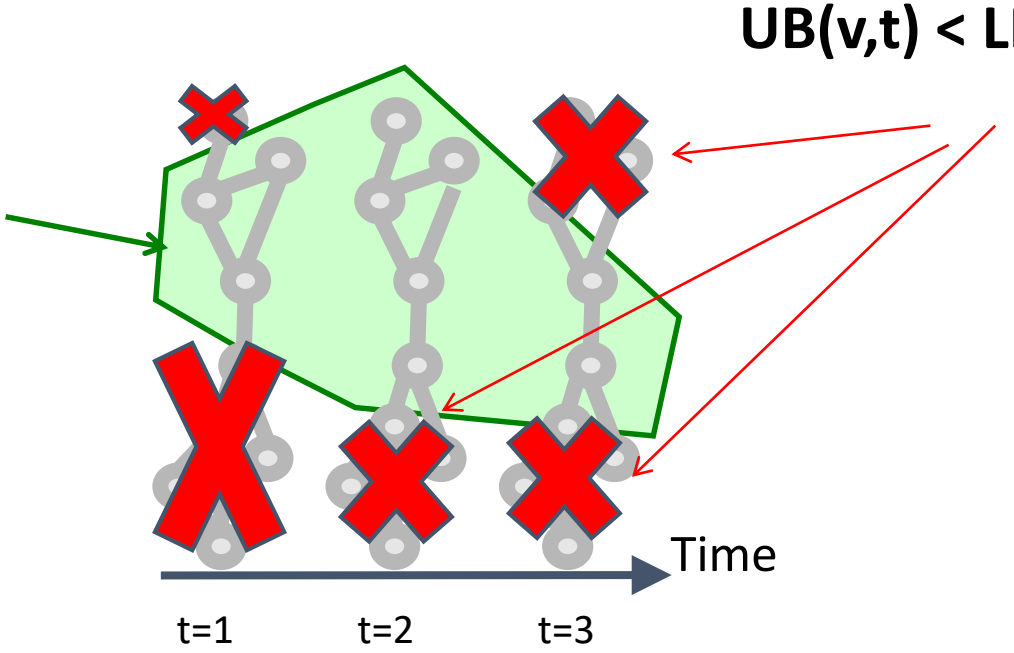
B in time

ard DAGs in

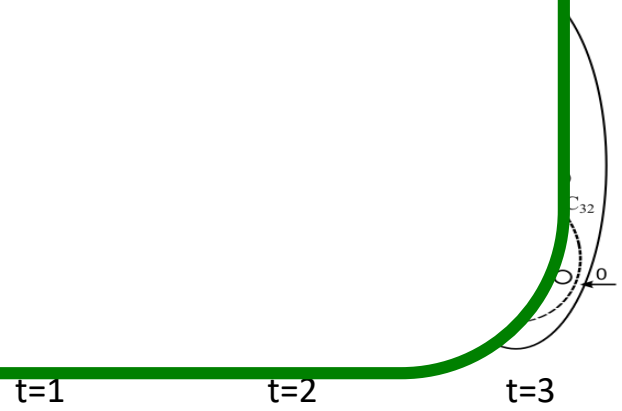
(DP)

$+ Ub_b$

Remaining small and fragmented Instance (~5% of original)



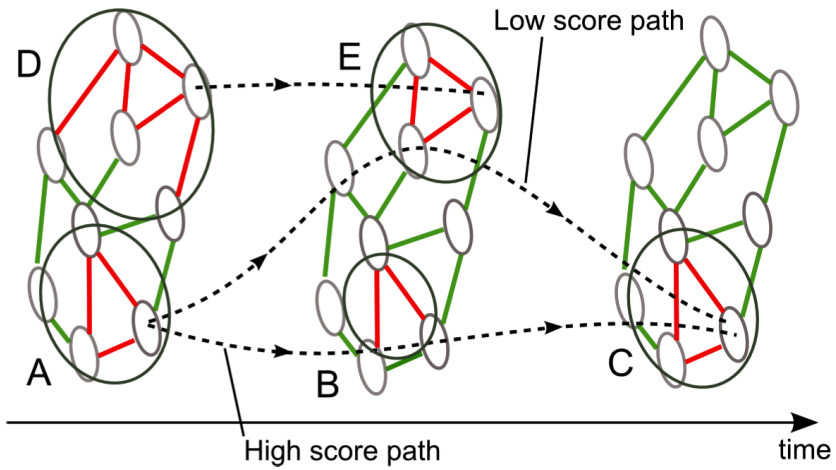
No Loss of Accuracy



Complexity: O(|DAG|)

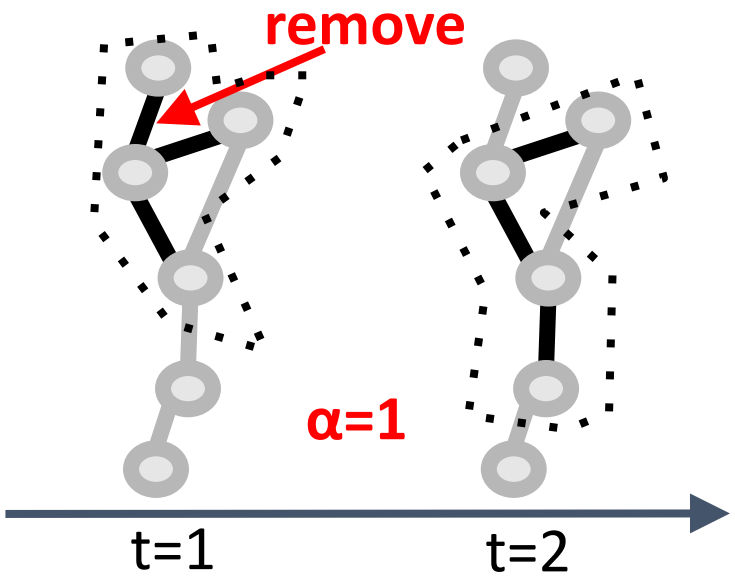
High-Quality Search: Spot&Smooth

Spot ∞-smooth candidates

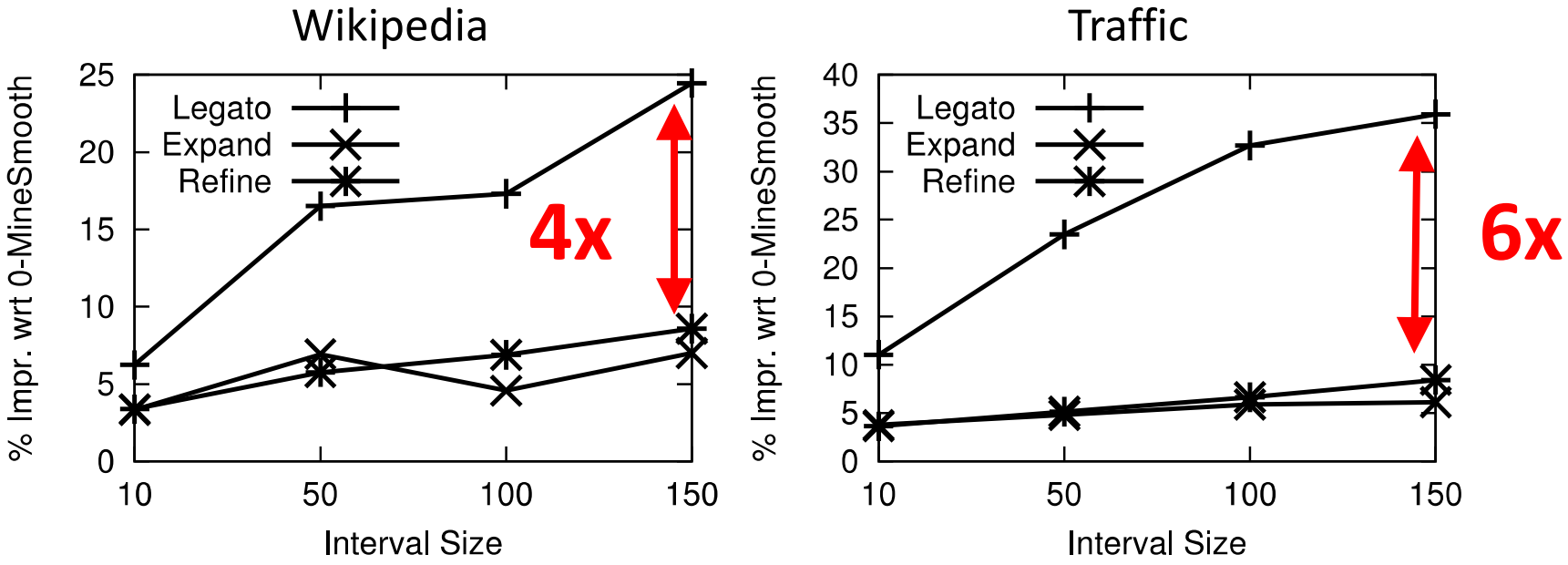


Best paths connecting positive components

"Smooth" each candidate according to α



Quality: Solution Score Improvement

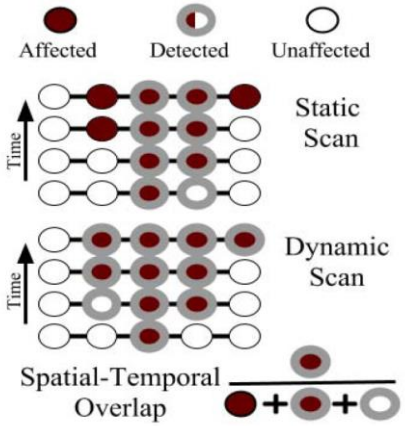


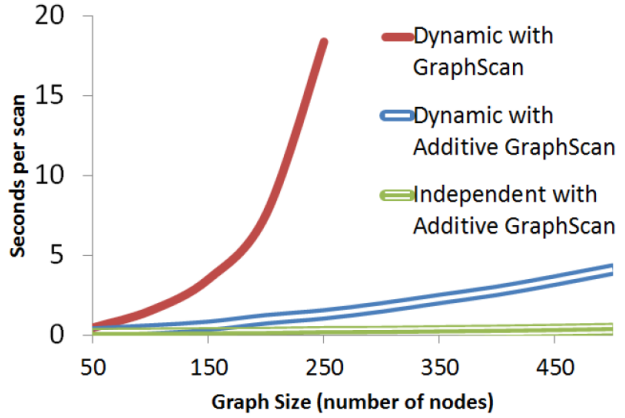
- **Higher-score temporal subgraphs**
- **95% of instance pruned**
- **Infeasible without bound-based pruning**

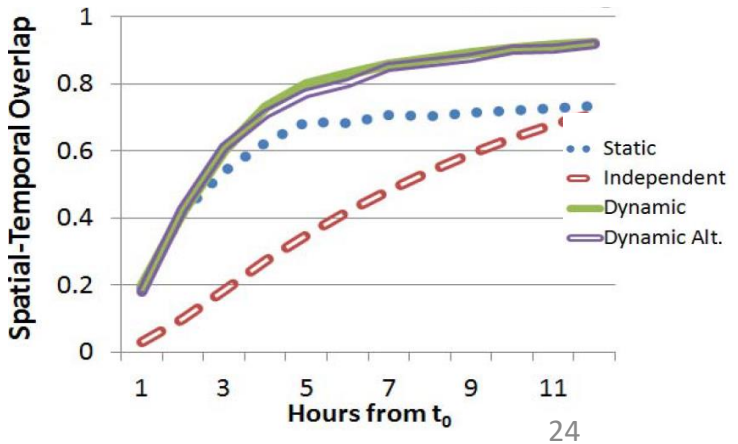
Dynamic GraphScan

[Speakman et al. ICDM'13]

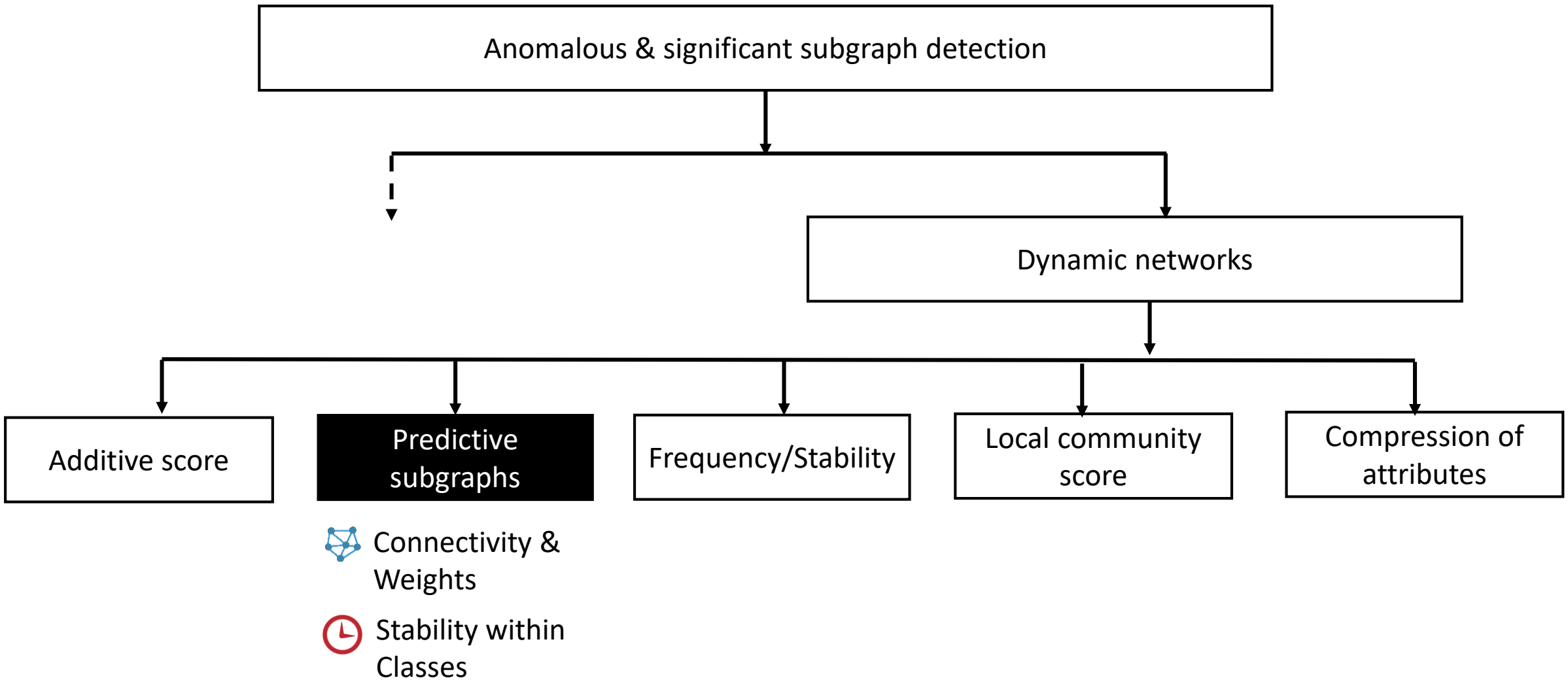
- Setting: binary uncertain node sensors
- Log-likelihood **Expectation-based binomial (EBB)** scan statistic
  - Additive set function even after incorporating consistency constraints
  - Consistency soft constraints: selection of a subset at *t* affects those at *t-1* and *t+1*
- Efficient solution: Dynamic Additive GraphScan
  - Enforces connectivity
  - Prefers smooth (consistency) subgraph selection
  - Scales with the instance size
  - Retains good quality





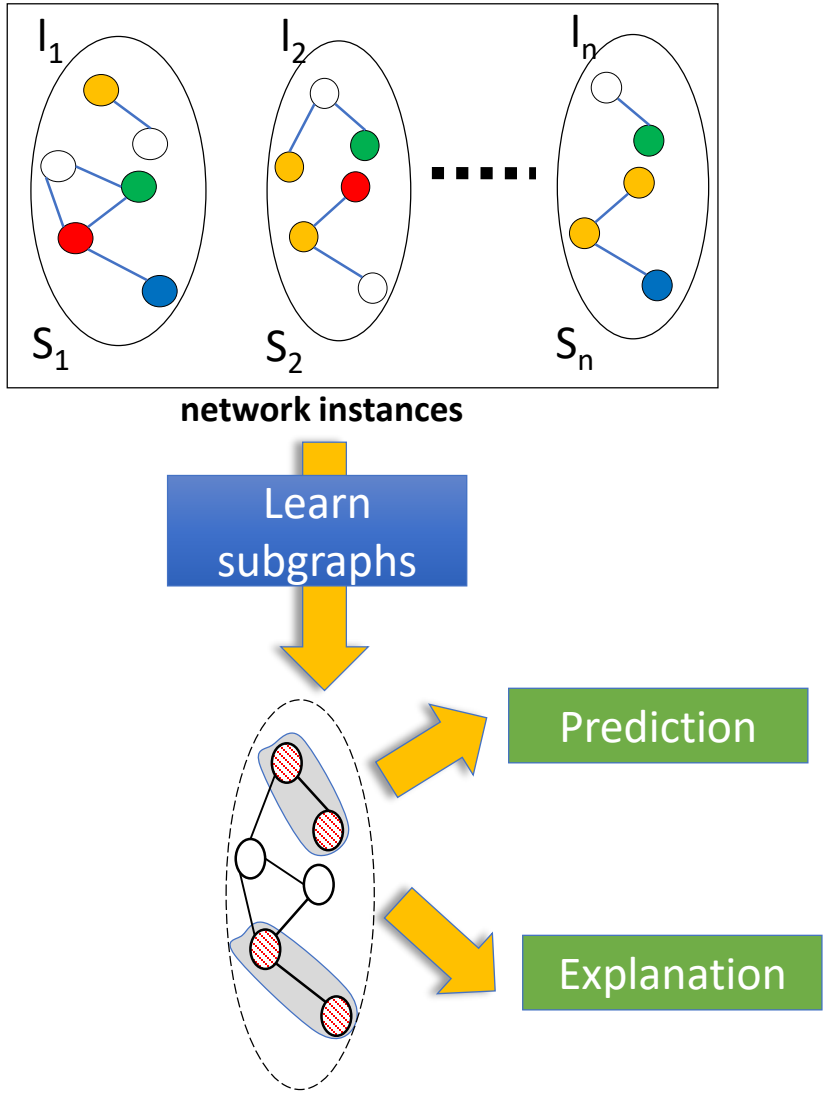


Taxonomy


Anomalous & significant subgraph detection
Dynamic networks
Additive score
Predictive subgraphs
Frequency/Stability
Local community score
Compression of attributes
Connectivity & Weights
Stability within Classes

Subgraphs predictive of global network state

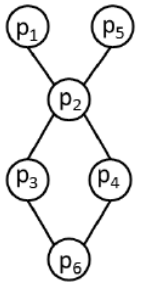
- **Input:** A set of network instances
  - Global network labels
  - Real-valued node/edge attributes
- **Goal:** Learn subgraphs
  - Connected
  - Predictive (separate instances)
- Challenges
  - "Wide" data: much fewer instances than the number of nodes/edges
  - Enforcing connectivity



MINDS: Network-constrained decision trees

[Ranu et al. KDD'13]

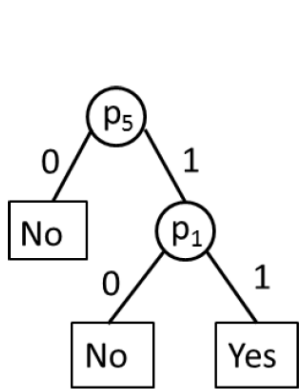
- Idea: Learn decision trees based on node attributes, while ensuring connectivity
- Large search space: all connected subgraphs
  - MCMC sampling of trees
  - Add nodes with probability proportional to their information gain (IG) increase
  - Allow for removal of nodes
  - Greedy DT construction in subgraph
- (+) Scalable and accurate
  - Compared to unconstrained SVM and Greedy
- (-) Sampling-based: non-deterministic



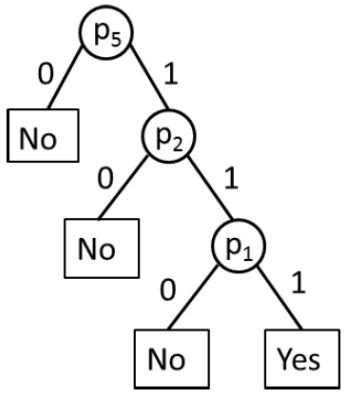
Network Structure

Human ID	p_1	p_2	p_3	p_4	p_5	p_6	Cancer
1	1	0	1	0	0	1	No
2	1	1	0	0	1	0	Yes
3	1	1	0	1	1	0	Yes
4	1	1	1	0	0	0	No
5	1	1	1	1	1	1	Yes
6	0	0	0	0	0	0	No
7	0	1	0	0	1	0	No
8	0	1	0	1	1	0	No

Local and Global states of the network



Unconstrained



Network-constrained

Discriminative Subgraph Discovery (SNL)

[Dang et al. PKDD'14]

- Learn a node projection in a low-dimensional space
  - Same-label instances are "close" and different-label instances "far"
  - The projection is "smooth" w.r.t. network structure
- Solution: spectral graph theory
- (+) Scalable and accurate
  - Compared to unconstrained SVM and Greedy
- (-) Does not enforce sparsity in solution – need to threshold **u** as post-processing

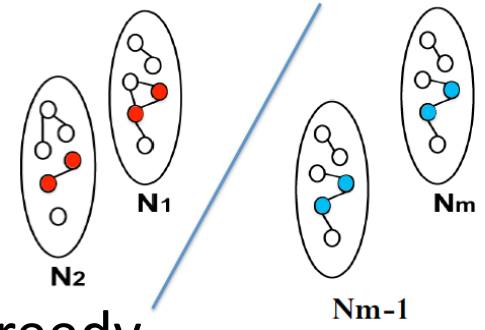
$$\mathbf{u}^* = \arg\max_{\mathbf{u}} \left\{ tr\left(\mathbf{u}^T \mathbf{V} (\mathbf{L}^- - \mathbf{L}^+) \mathbf{V}^T \mathbf{u}\right) \right\}$$
$$\text{subject to } \mathbf{u}^T \mathbf{C} \mathbf{u} \leq t$$
$$\text{and } \mathbf{u}^T \mathbf{V} \mathbf{D}^+ \mathbf{V}^T \mathbf{u} = 1$$

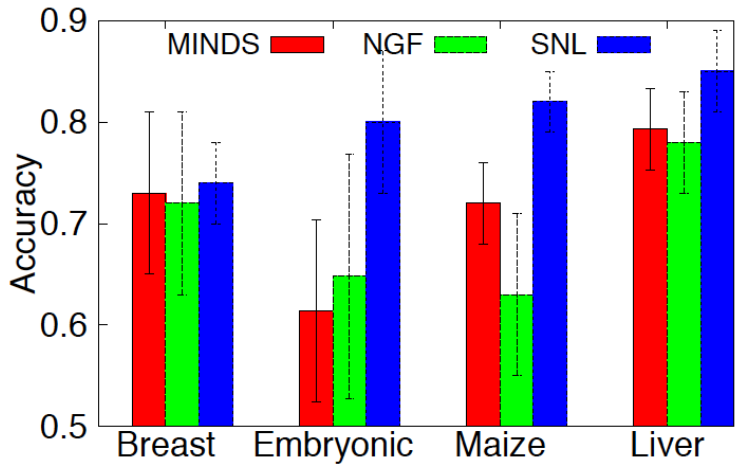
Maps network nodes to real values (1D case)

Network structure regularization

Ensure Uniqueness

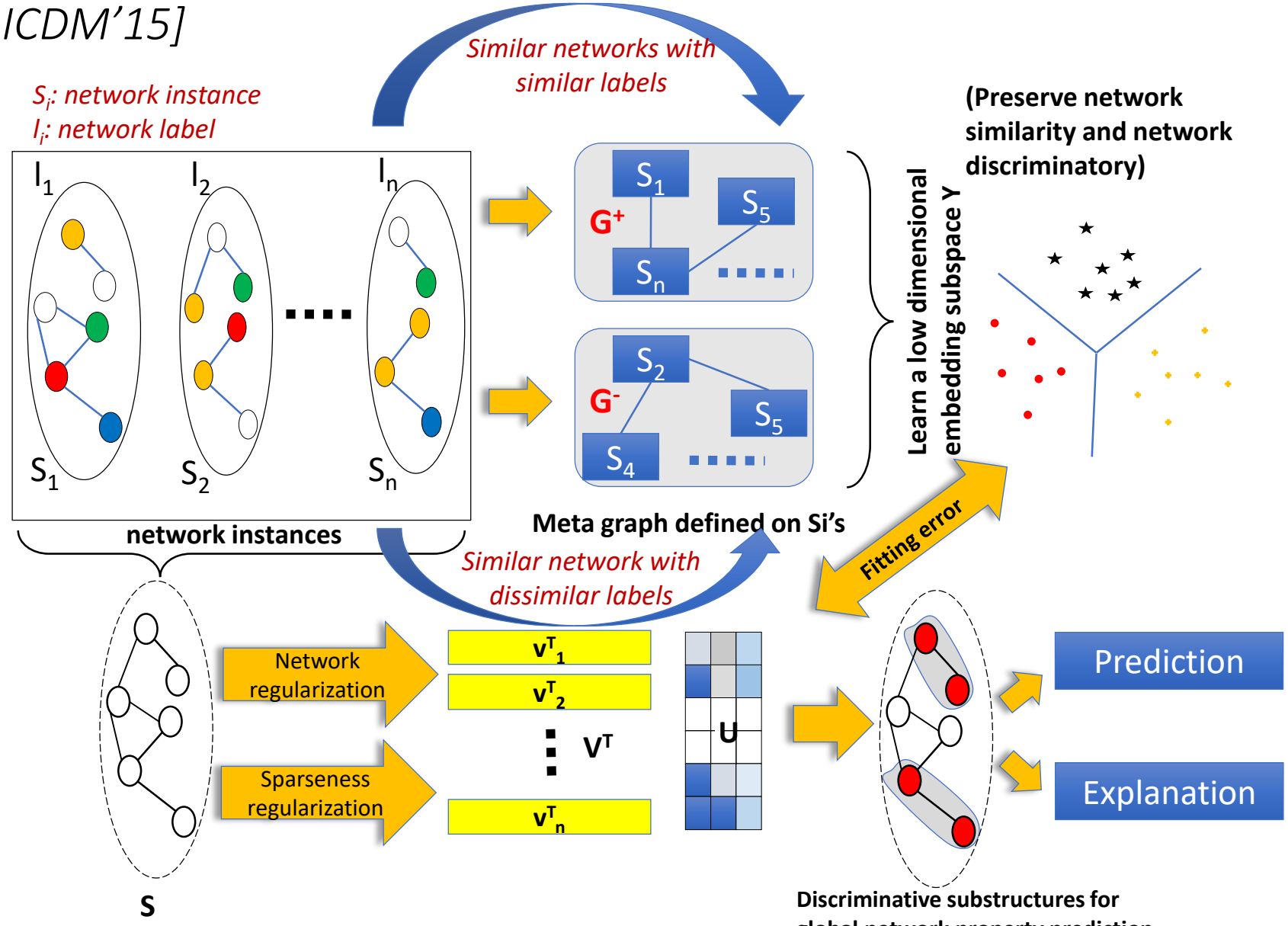
Laplacians of meta-graphs



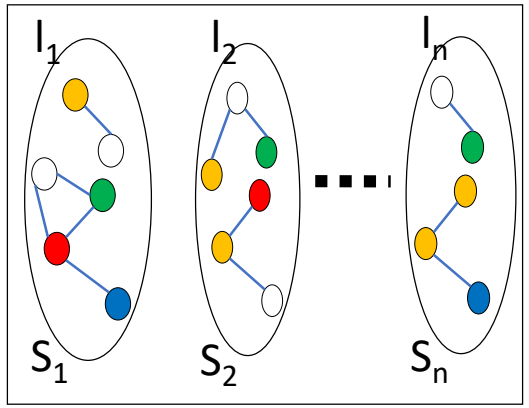


DIPS: Graph regularization+sparsity

[Dang et al. ICDM'15]



Discriminative subspace

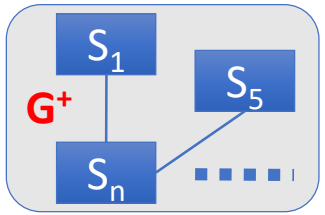


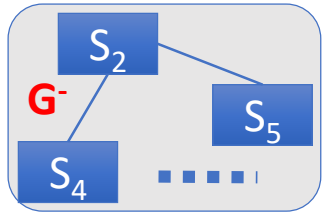
Input: a set of labeled network instances and a similarity function

$$S_i = \{V_i, E_i, F\}$$

Construct two kNN similarity graphs among network instances

- G^+ and its weighted adjacency matrix K^+: similarity among instances of *different labels*
- G^- and its adjacency matrix K^-: similarity among same-label instances





Use simple notion of similarity: Cosine

Goal: Learn a low-dimensional space that increases similarity in G^+ and decrease that in G^-

Subspace Learning + Sparse Subnetworks

Project instance S_i to y_i while optimizing the objective functions for similarity and dis-similarity:

$$\begin{aligned}\text{minimize} \sum_i \sum_j \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 \mathbf{K}_{ij}^+ \\ = \sum_i \sum_j (\|\mathbf{y}_i\|_2^2 + \|\mathbf{y}_j\|_2^2 - 2\mathbf{y}_i^T \mathbf{y}_j)\mathbf{K}_{ij}^+ \\ = 2\mathbf{Y}^\mathbf{T}(\mathbf{D}^+ - \mathbf{K}^+)\mathbf{Y} = \mathbf{2tr}(\mathbf{Y}^\mathbf{T}\mathbf{L}^+\mathbf{Y})\end{aligned}$$

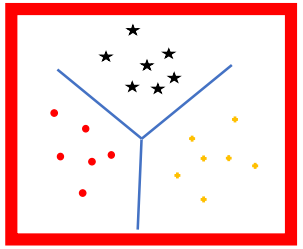
$$\begin{aligned}\text{maximize} \sum_i \sum_j \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 \mathbf{K}_{ij}^- \\ = 2\mathbf{Y}^\mathbf{T}(\mathbf{D}^- - \mathbf{K}^-)\mathbf{Y} = \mathbf{2tr}(\mathbf{Y}^\mathbf{T}\mathbf{L}^-\mathbf{Y})\end{aligned}$$

Combined:

$$\mathbf{Y} = \arg\max_{\mathbf{Y}} \left\{ \mathbf{tr}\left(\mathbf{Y}^\mathbf{T}(\mathbf{L}^- - \beta\mathbf{L}^+)\mathbf{Y}\right)\right\} \quad \text{subject to} \quad \mathbf{Y}^\mathbf{T}\mathbf{D}^+\mathbf{Y} = \mathbf{I}$$

Solution:

$$\mathbf{LY} = \mathbf{\Lambda DY}$$



fitting error; sparsify nodes; regularizing network

$$\arg\min_{\mathbf{U}} \|\mathbf{Y} - \mathbf{V}^\mathbf{T}\mathbf{U}\|_F^2 + \lambda_2 \sum_{\mathbf{u}\in\mathbf{U}} \mathbf{u}^\mathbf{T}\mathbf{C}\mathbf{u} + \lambda_1 \sum_{\mathbf{u}\in\mathbf{U}} |\mathbf{u}|$$

- Add constraints
  - network topology
  - sparsity
- Enforce selection of connected subgraphs
- U selects and combines nodes to form embedding subspace
- Non-smooth but convex -> GD

$$\mathbf{C}_{pq} = \mathbf{C}_{qp} = \begin{cases} \sum_q W(p,q) & \text{if } v_p \equiv v_q \\ -W(p,q) & \text{if } v_p,\ v_q \text{ are connected} \\ 0 & \text{otherwise} \end{cases}$$

Results - Quality

- **DIPS** (Discovering Predictive Substructure)
- NGF (tree/node): random forest
- MINDS (tree/node): MCMC + D. Trees
- SVD+SVM

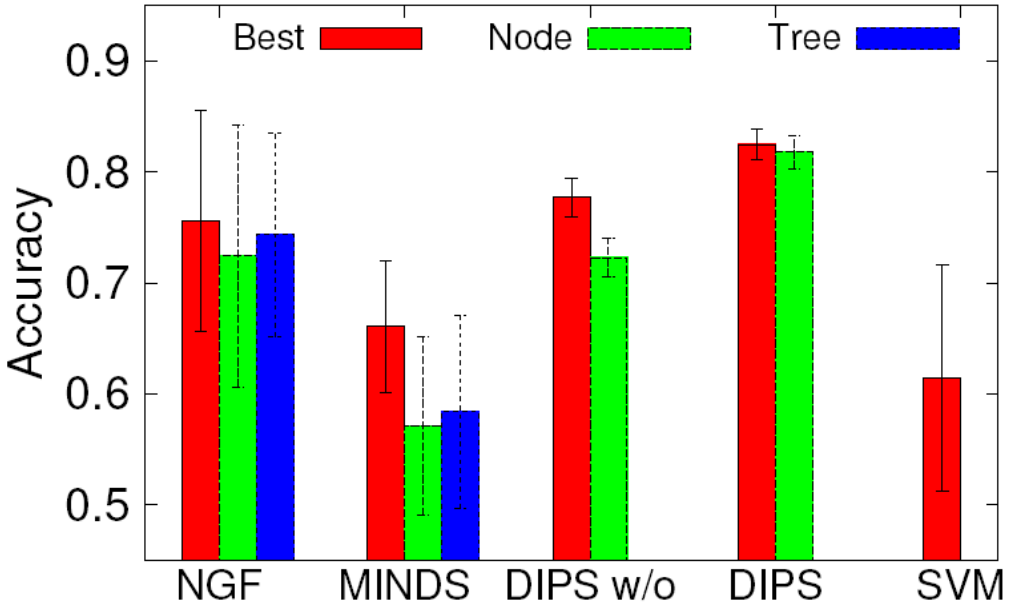
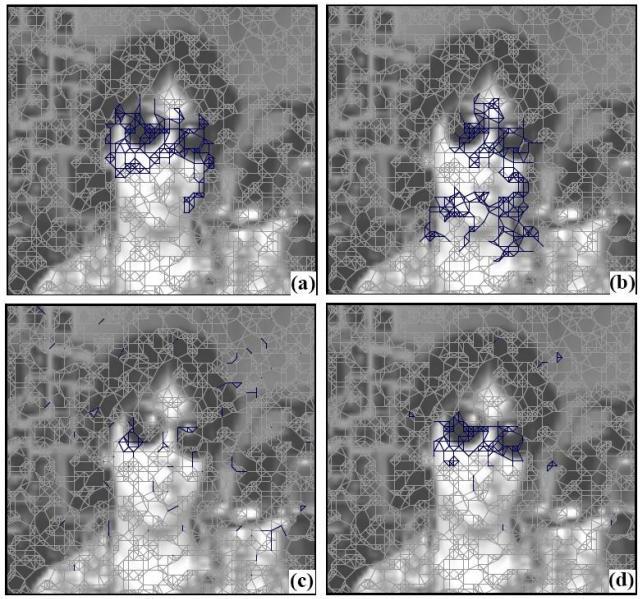


Fig1. Prediction accuracy

Image Networks

- 1920 nodes
- 11172 edges
- 156 network samples

(sunglass vs naked-eye)

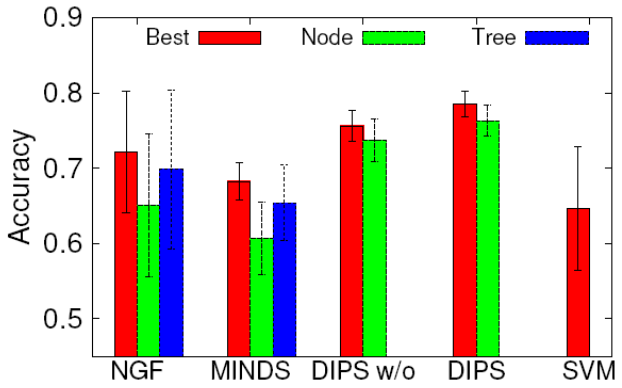


Subnetworks selected by
(a) NGF (b) MINDS
(c) DIPS(w/o) (d) DIPS

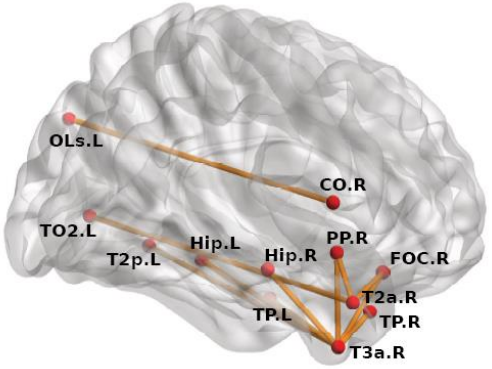
Sparse Subnetworks

Brain Networks

- 2948 nodes (on average)
- 191750 edges
- 173 network samples



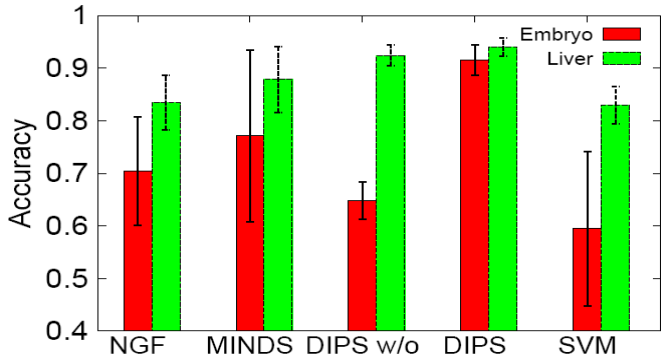
Predictive substructures largely reside in temporal lobe (TP.L, TP.R, T2a.R, T2p.L, T3a.R) and hippocampus (Hip.L/Hip.R),

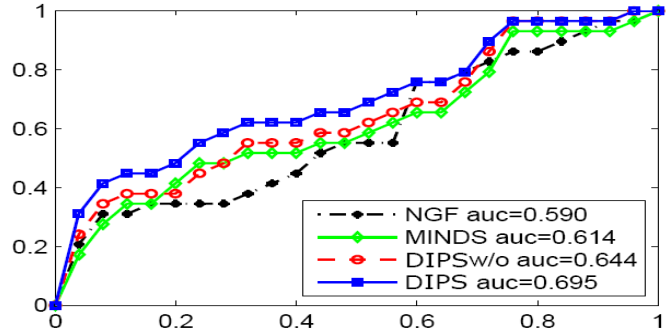


Gene Expression Networks

Embryonic development

- 1321 genes
- 5277 edges
- 35 network samples

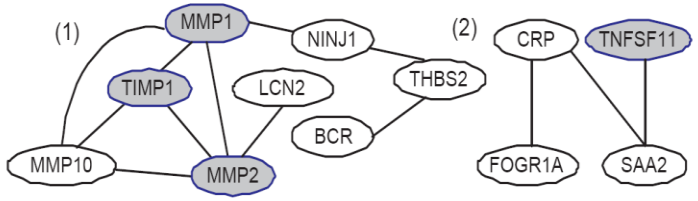




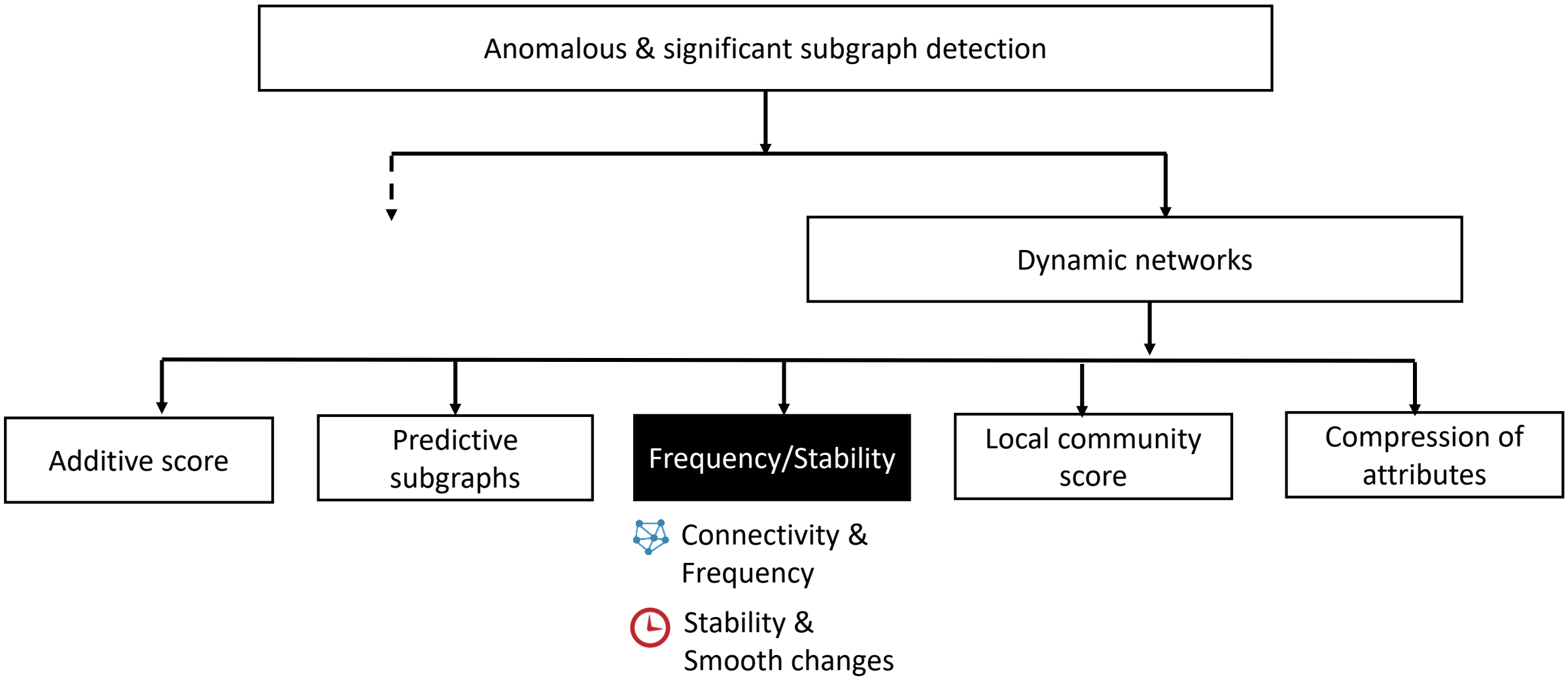
ground truth genes collected from literature

Liver metastasis

- 7383 genes
- 251916 edges
- 123 network samples

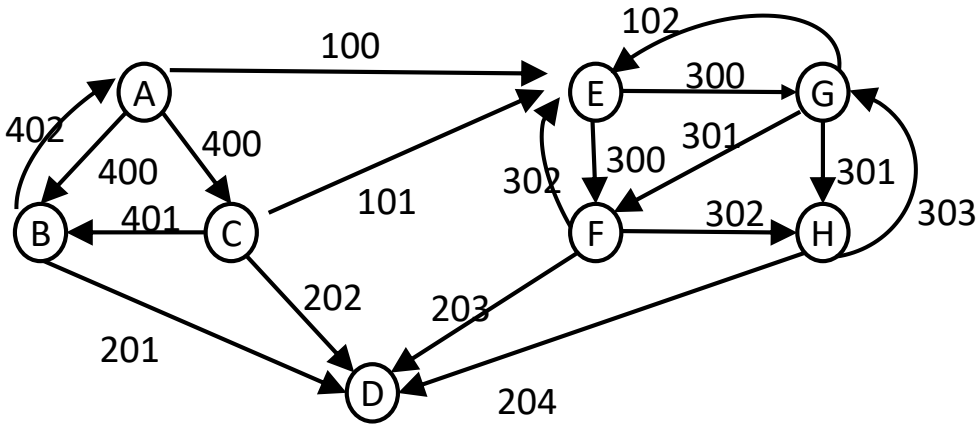


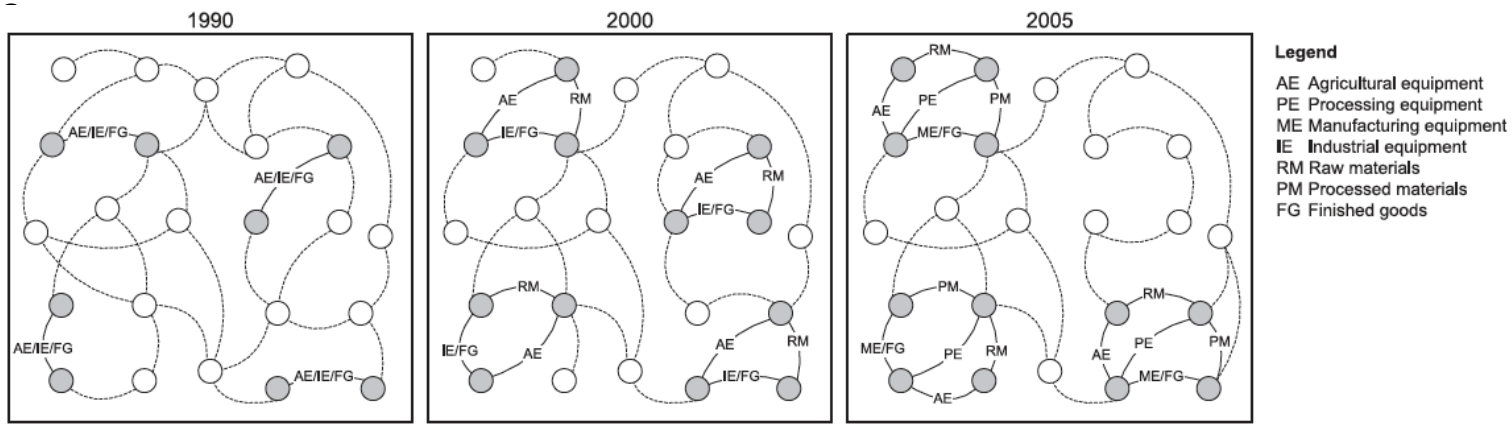
Taxonomy


Anomalous & significant subgraph detection
Dynamic networks
Additive score
Predictive subgraphs
Frequency/Stability
Local community score
Compression of attributes
Connectivity & Frequency
Stability & Smooth changes

Frequent and stable structure

- **Goal**: find frequently occurring/conserved subnetworks
  - Frequent dynamic subgraphs
  - Communication threads
  - Interactions that persist in time
- Score is related to a property o the structure over time
- Need constraints to avoid trivial solutions
  - Frequency

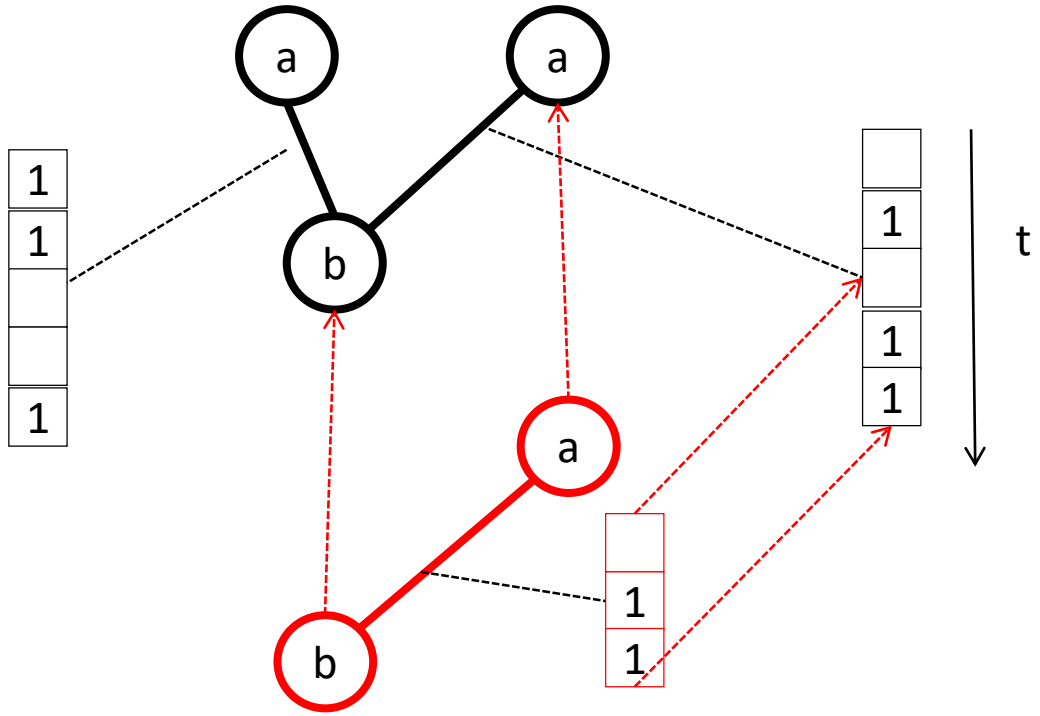




Frequent Dynamic Subgraphs

[Borgwardt et al. ICDM'06]

- Generalize frequent subgraphs to the dynamic setting
- Edges are associated with an existence vector across time
- Dynamic subgraph embedding
  - Isomorphic structure and matching labels
  - Matching temporal edge pattern (activation times) for a specific duration
  - Embeddings do not overlap in time/graph
- Solution main ideas
  - Start with frequent (label-wise) edges
  - Ensure edge pattern match (prefix trees)
  - Extend GREW *[Kuramochi et al. ICDM'04]* to handle edge activations

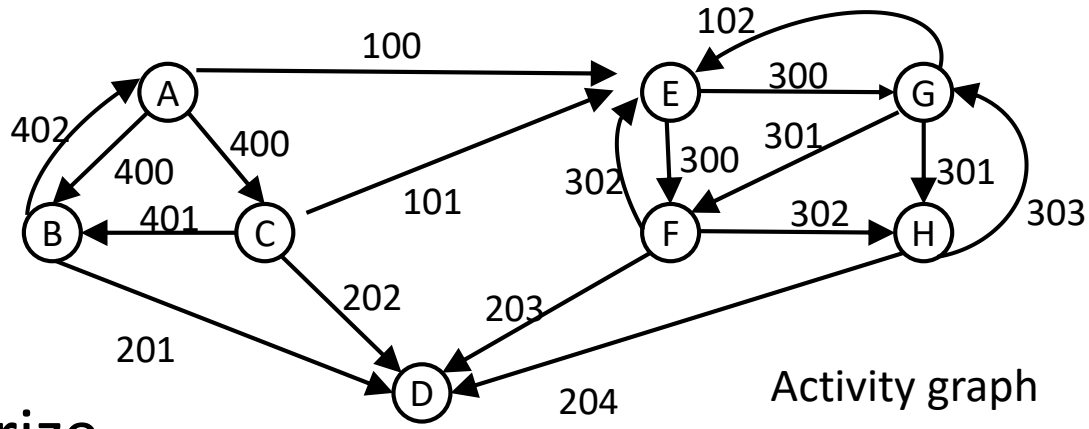


asynchronous dynamic subgraph

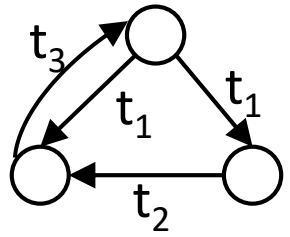
Recurring communication motifs

[Gurukar et al. SIGMOD'15]

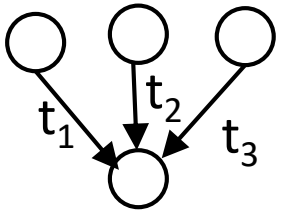
- Motif: A ***recurring subgraph*** that has a ***unique sequence of interactions***
  - Recurring subgraph: Support of 3
  - Unique sequence: $t_1 < t_2 < t_3$
- Communication/Activity motifs characterize social networks
  - Do they vary across social networks?
  - … with time of day/ user experience, etc.
- How to mine most conserved motifs efficiently?



Activity graph



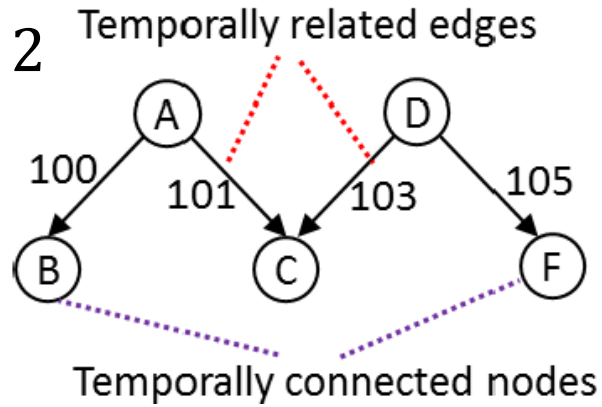
Activity motif

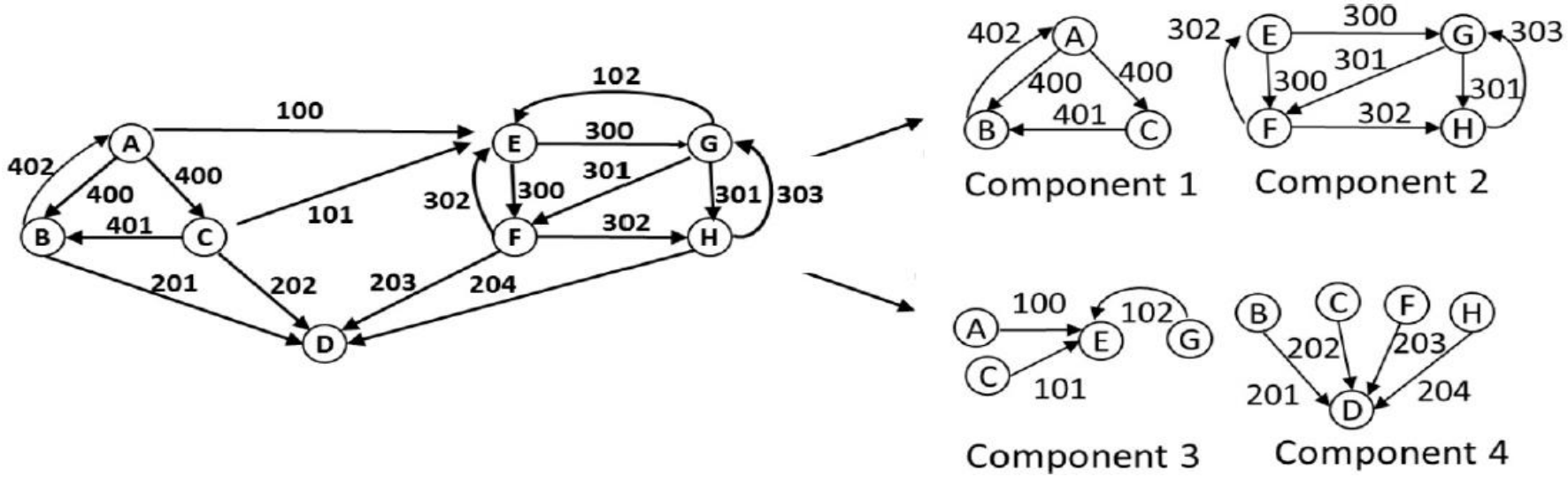


Activity motif

Temporal relationships, graphs and subgraphs

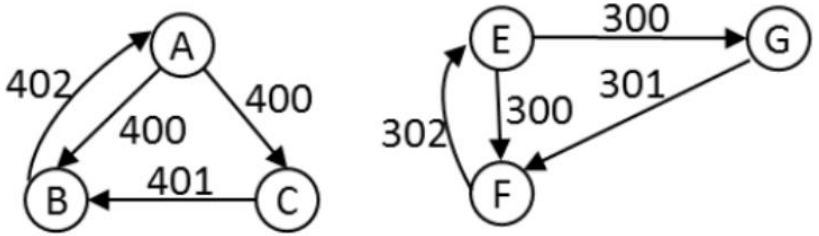
- Two adjacent edges are ***temporally related*** if the difference in their timestamps is less than $\Delta T = 2$
- Two nodes are ***temporally connected*** if their exists a sequence of adjacent edges connecting them and and are temporally related
- ***Temporally connected graph***: a subset of temporally connected nodes





Temporally-isomorphic subgraphs

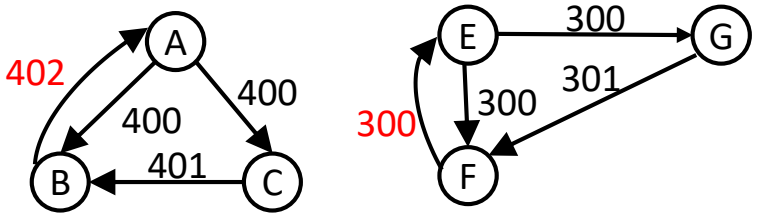
- Isomorphic subgraphs with the ***same sequence of information flow***

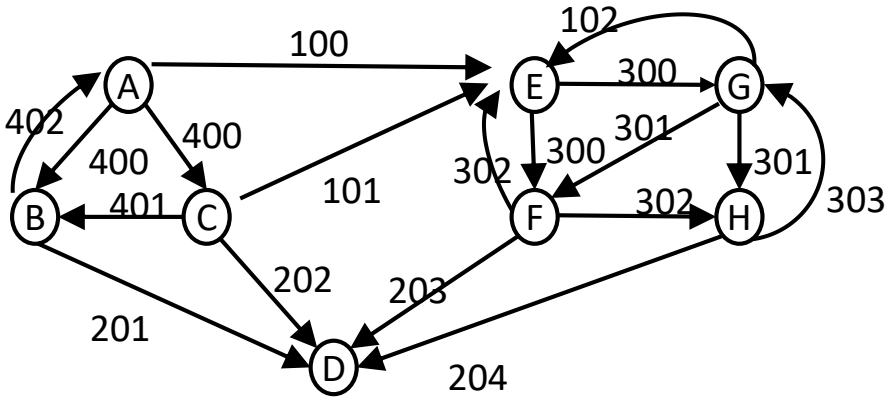


- Problem: Given
  1. Dynamic Interaction Network
  2. Temporal connectivity threshold ΔT
  3. Support threshold τ

Mine:
All ***connected***, **temporally isomorphic subgraphs** with ***support above τ***

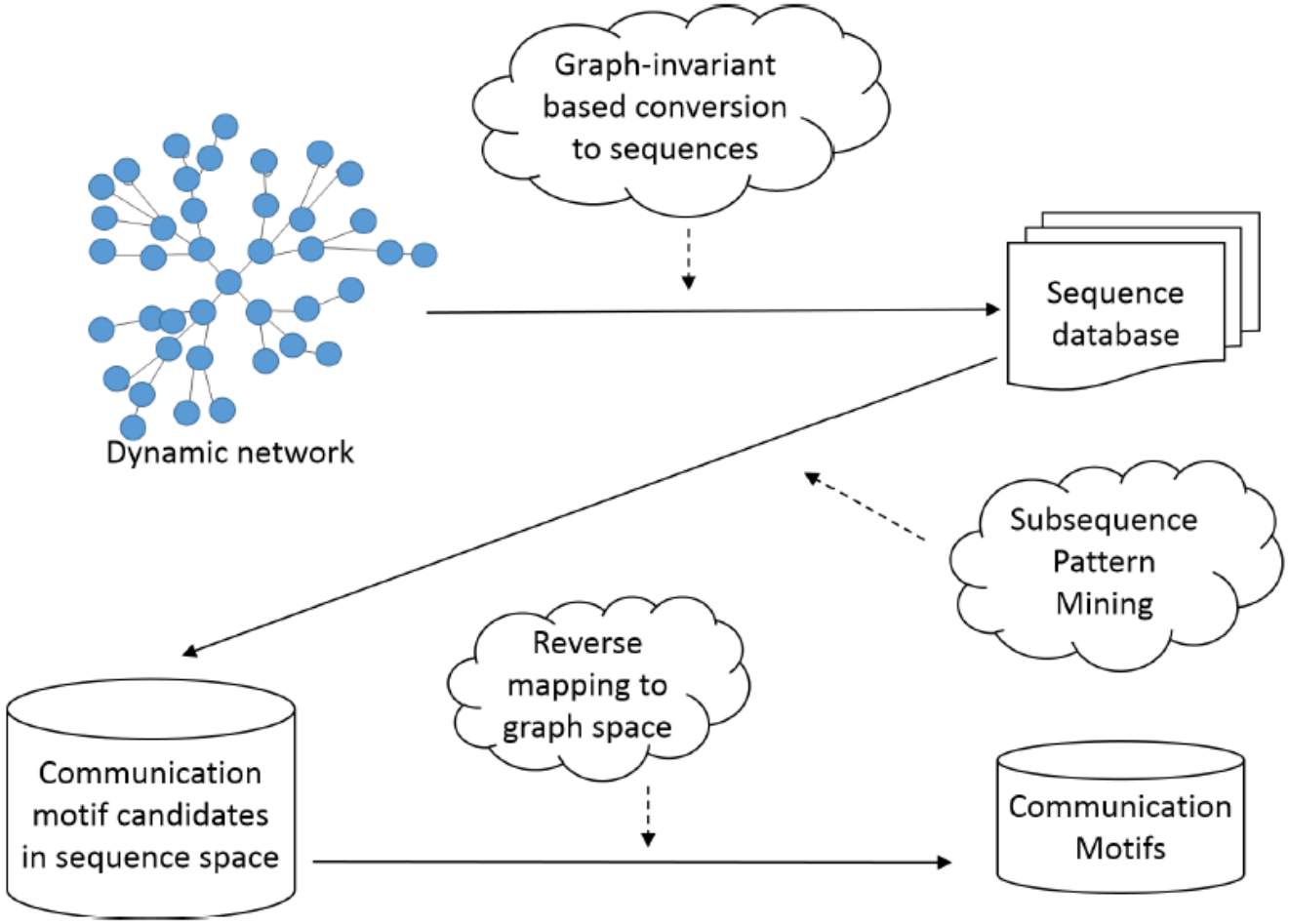
But, NOT temporally isomorphic:



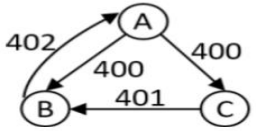


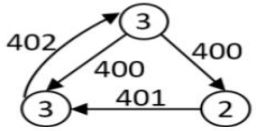
At $\tau = 3, \Delta T = 1$

COMMIT: Pipeline and main ideas



Encoding sequences



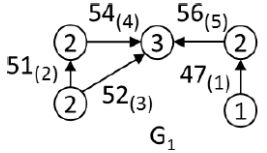


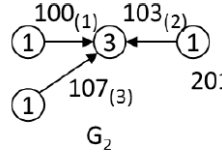
(3,2)(3,3)(2,3)(3,3)

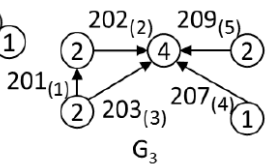
Replace node labels with degrees

Encode edges in sequence

Sequence growth







SID	Sequence
1	(1,2), (2,2), (2,3), (2,3), (2,3)
2	(1,3), (1,3), (1,3)
3	(2,2), (2,4), (2,4), (1,4), (1,4)

Support Set of (1,3)
(1, < 3 >)
(1, < 4 >)
(1, < 5 >)
(2, < 1 >)
(2, < 2 >)
(2, < 3 >)
(3, < 2 >)
(3, < 3 >)
(3, < 4 >)
(3, < 5 >)

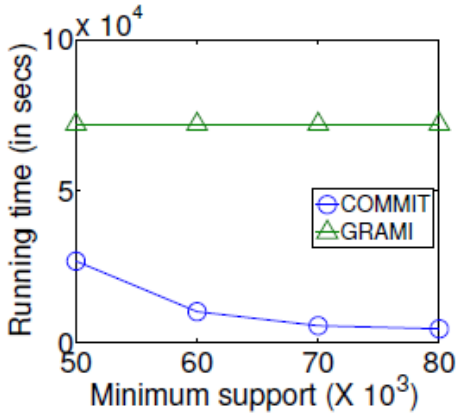
Subsequence growth from (1,3) to (1,3), (1,3),(1,3)

Support Set of (1,3), (1,3)
(1, < 3, 4 >)
(1, < 4, 5 >)
(2, < 1, 2 >)
(2, < 2, 3 >)
(3, < 2, 3 >)
(3, < 3, 4 >)
(3, < 3, 5 >)

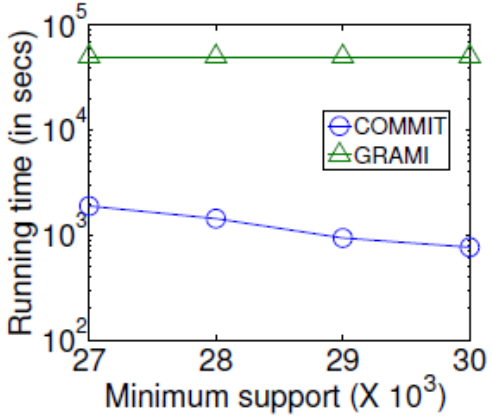
Support Set of (1,3), (1,3), 1,3)
(1, < 3, 4, 5 >)
(2, < 1, 2, 3 >)
(3, < 2, 3, 4 >)
(3, < 3, 4, 5 >)

Motifs and scalability

- GRAMI *[Kalnis et al., VLDB'14]* ***crashes in*** around ***16 hours***
  - General frequent subgraph miner in a single large graph
- COMMIT ***finishes in*** around ***15 minutes***



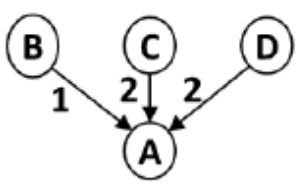
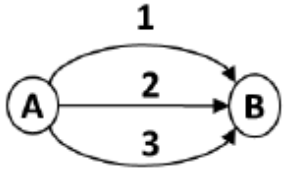
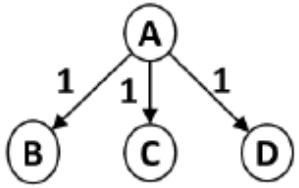
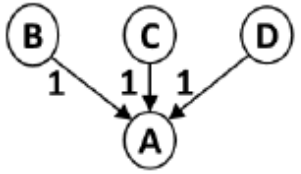
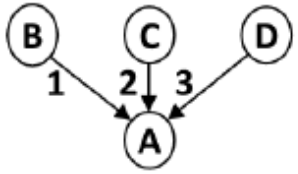
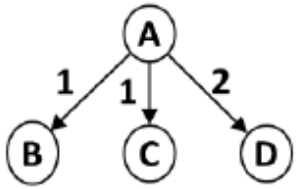
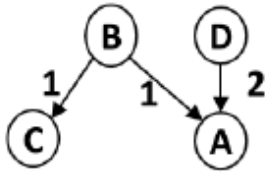
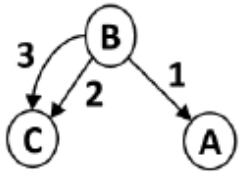
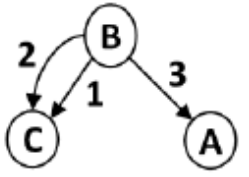
(a) Twitter



(b) Facebook

Further questions:
- How to set ΔT?
- Isomorphic structure too strict for real life. Relax?

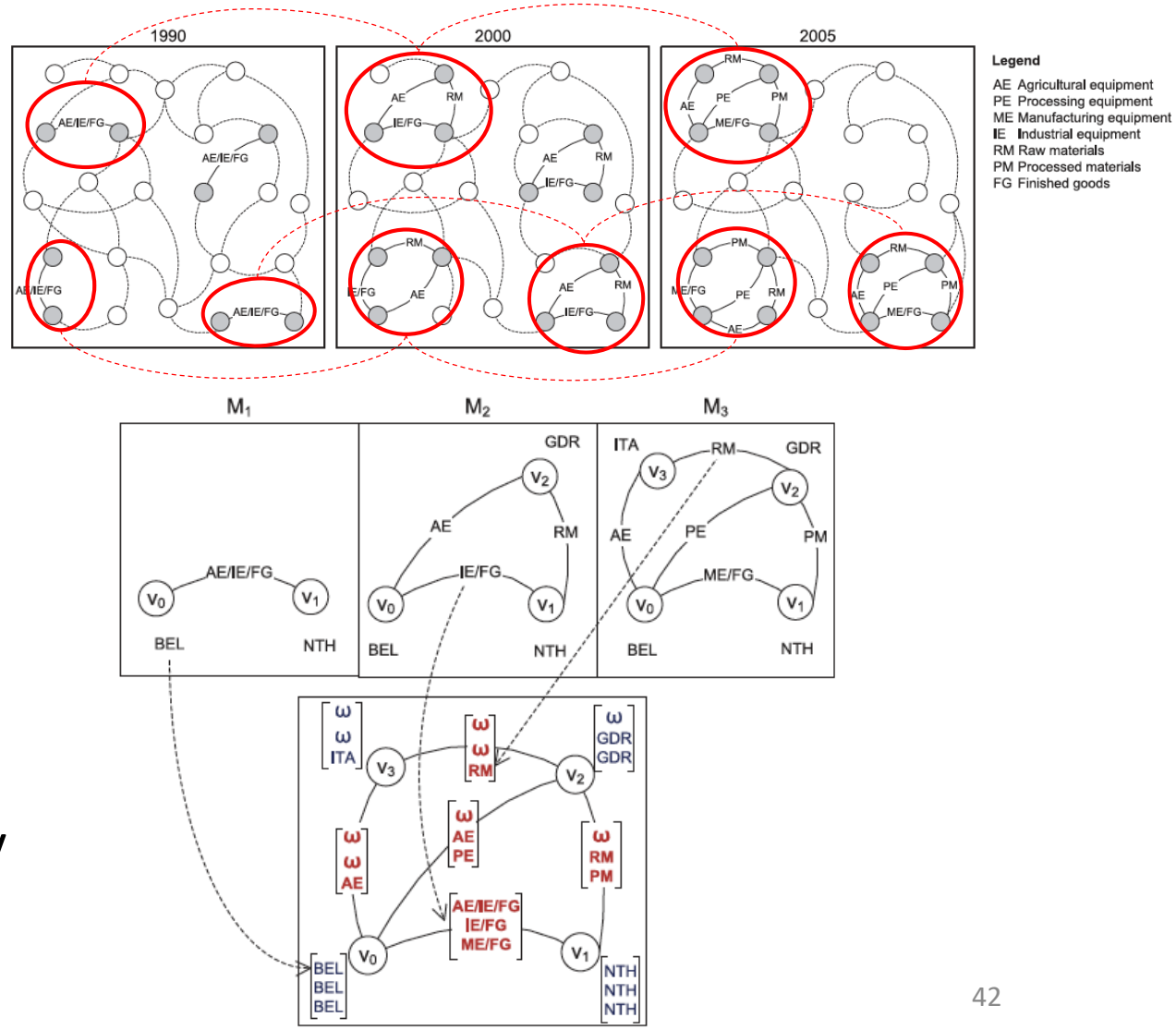
Mined motifs

Twitter Mentions	Facebook Wall Posts	Enron Email
		
		
		

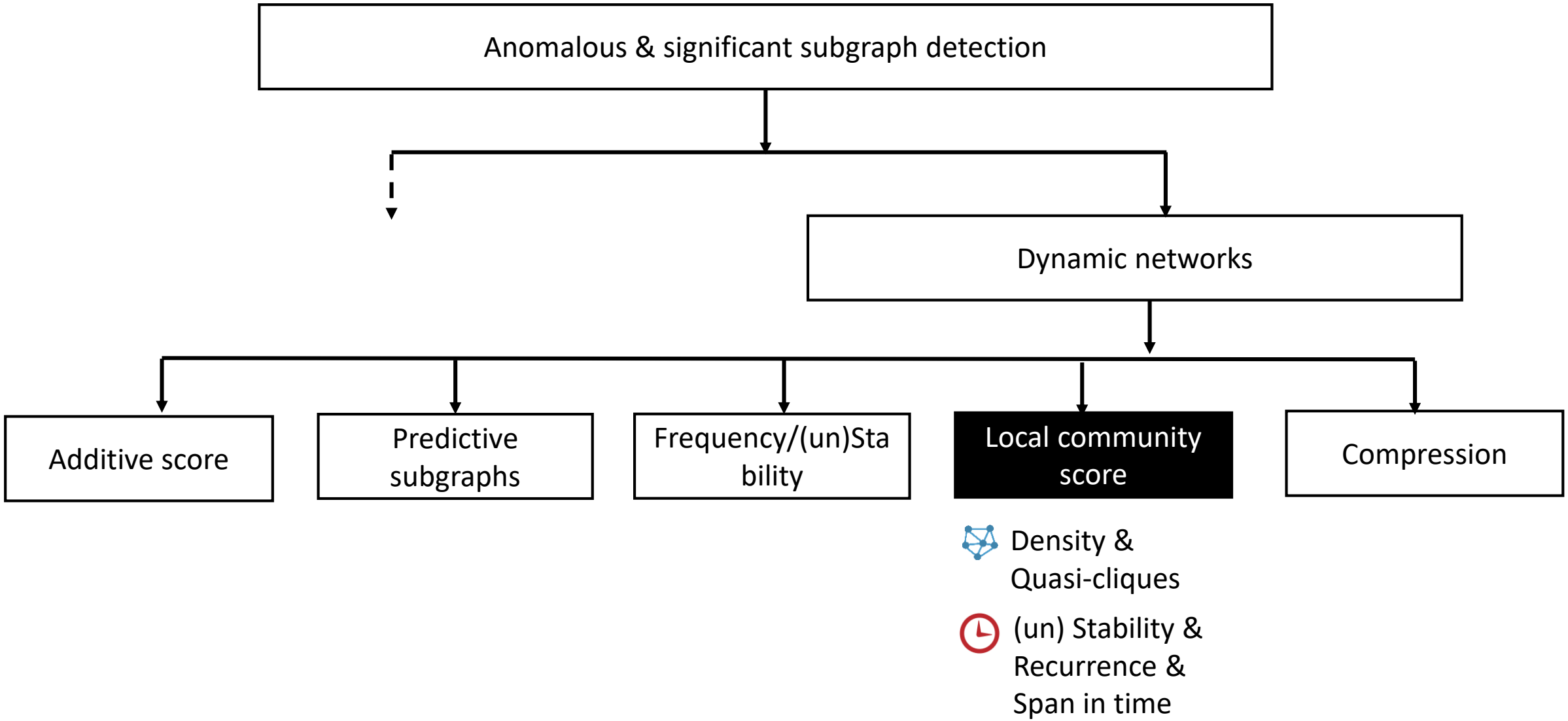
Conserved and Coevolving Relational Motifs

[Ahmed et al. ICDM'11, TKDD'15]

- CRMs: Recurring node sets whose relations change in a consistent manner over time
  - Frequent (number of embeddings)
  - Non-overlapping embeddings
  - Relations change across snapshots
  - Connected
  - Each snapshot covers a sufficient fraction of all involved nodes (different from freq. subgraphs)
- Solution: Grow patterns around frequent (anchor) edges
  - Use prefixScan [Pei et al.'11] for anchors
  - Use canonical labels to avoid redundancy
  - Prune by frequency constraints

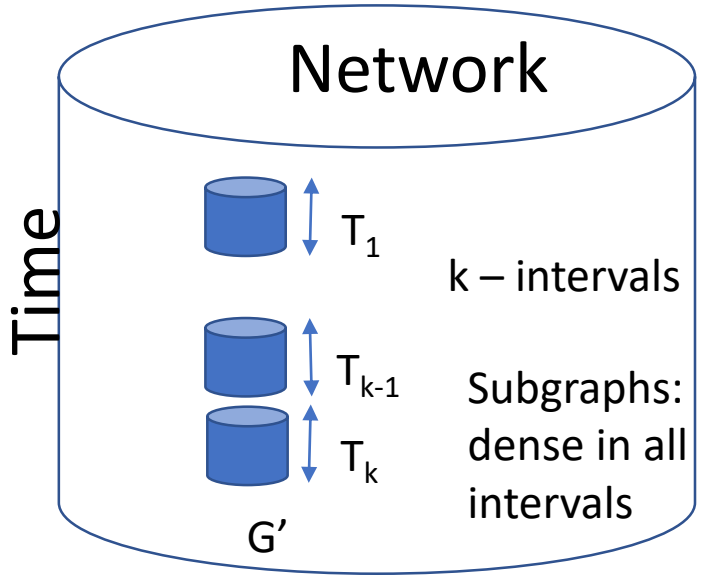


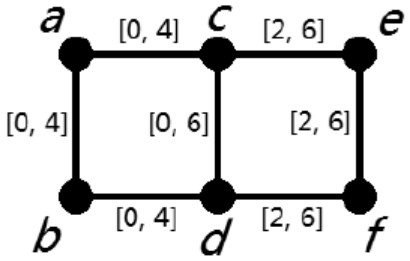
Taxonomy

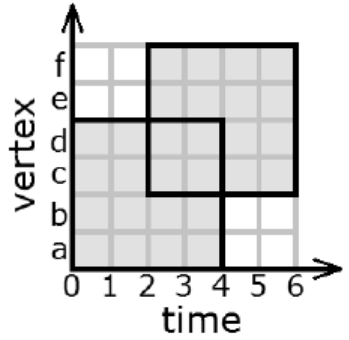

Anomalous & significant subgraph detection
Dynamic networks
Additive score
Predictive subgraphs
Frequency/(un)Stability
Local community score
Compression
Density & Quasi-cliques
(un) Stability & Recurrence & Span in time

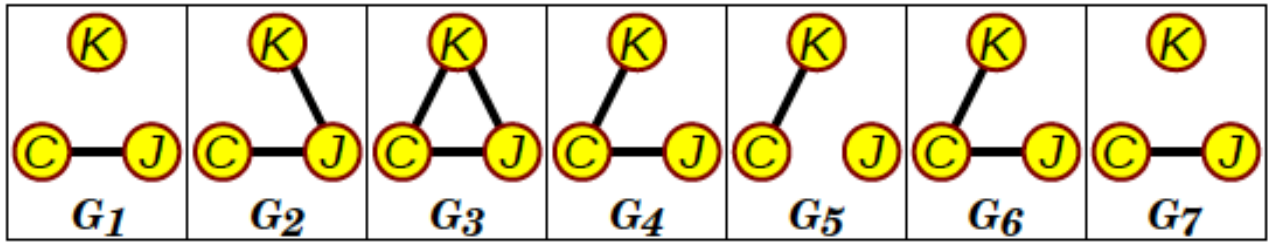
Dynamic communities

- **Score**: community goodness over time
  - Density
  - Quasi-clique
  - “Graphlet” distributions
- Local community scores
- Community is the same set of nodes over an interval of time or a subset of the network instances



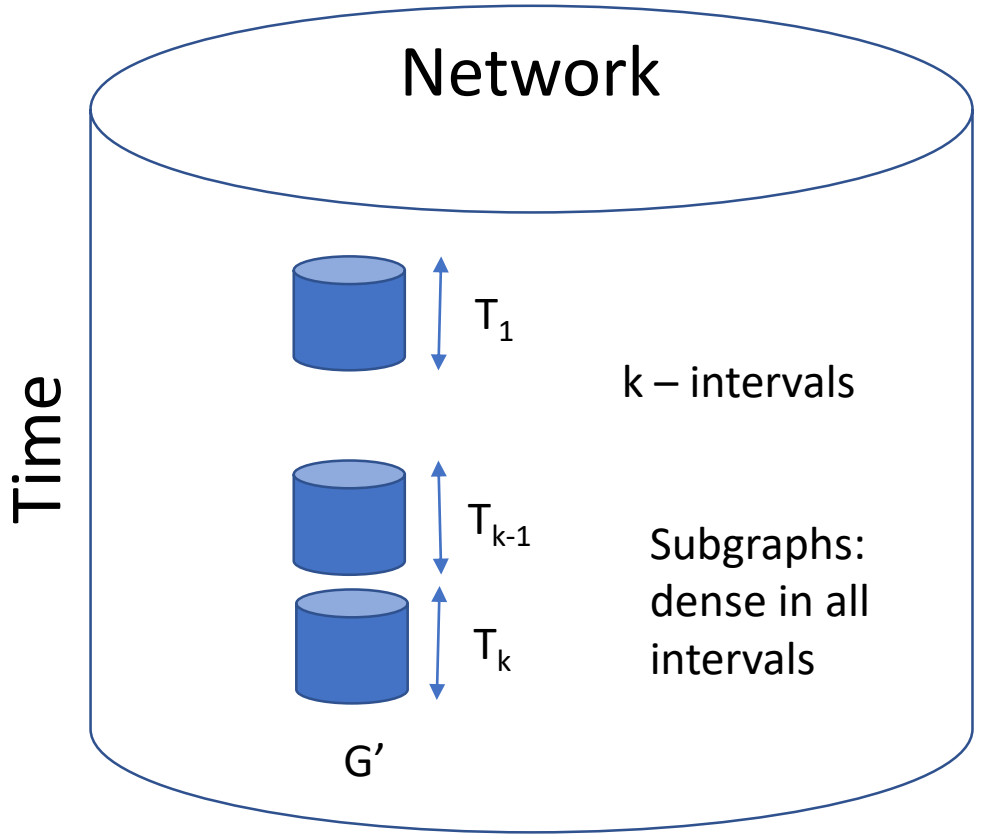






Dynamic Communities in Interaction Networks
[Rozenshtein et al. PKDD'14]

- Edges are interactions in time
- Dense temporal subgraph
  - Density: 2|E'|/|V'| in interval I
- **Problem:** Given the number of occurrences (intervals) **k**, and budget on the timespan **$B>=T_1+\ldots+T_k$** , find the subset of nodes V' that maximize the density within those intervals
- NP-hard (red. from Vertex Cover)



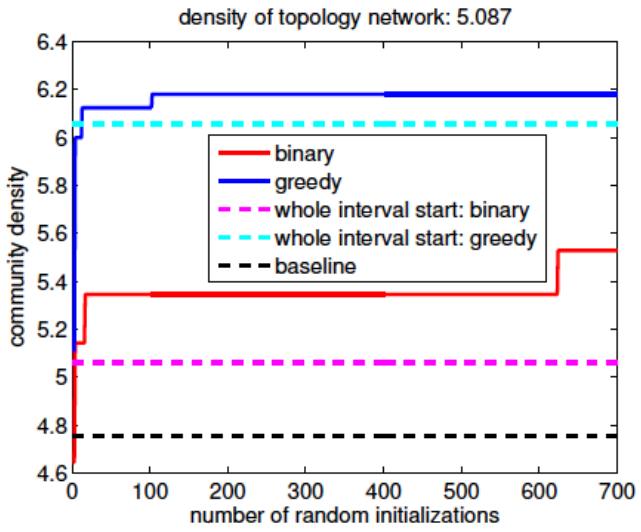
Solutions

- Alternate between subproblems
  - (1) Optimal intervals T for fixed V’
    - Maximum-coverage with multiple budgets (MCMB), NP-hard
    - **Greedy:** one interval at a time with best improvement
    - **Binary:** Relax by using only the span increase as a proxy
  - (2) Optimal set of nodes V’ for fixed T
    - Densest subgraph on aggregated graph
- Randomly initialize with a candidate solution

GREEDY MCMB:

$$\frac{q(W, \mathcal{T} \cup R, G) - q(W, \mathcal{T}, G)}{\max(x, y)}$$

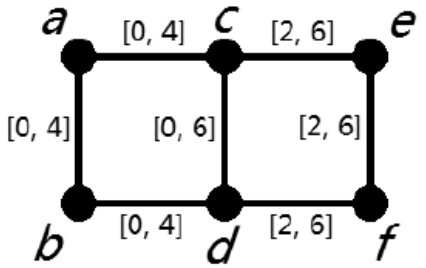
$$x = \frac{1}{K - |\mathcal{T}|} \text{ and } y = \frac{\text{span}(R)}{B - \text{span}(\mathcal{T})}$$

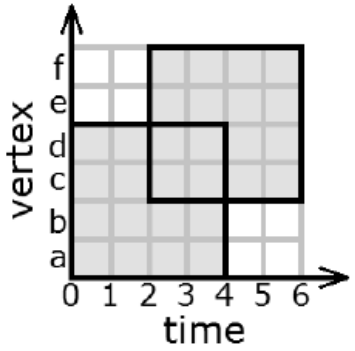


Students, $B = 7$ days, $K = 7$

Diversified Temporal Quasi-Cliques
[Yang et al. KDD'16]

- Setting: Edges last over an interval
- **Dense temporal subgraph** (V',E',I')
  - y-quasi-clique: the temporal degree exceeds a y-fraction of all included nodes V' over the interval of interest I'
  - constraint on size ("larger than")
  - constraint on time span ("longer than")
- **Coverage**: the nodes over time that the temporal graph includes
  - submodular
- **Problem**: k dense temporal subgraph that maximize coverage (diversity)





(a,b,c,d, [0-4]) and **(c,d,e,f, [2-6])**

Both 0.6-Quasi-clisues

span: 4

size:4

Coverage:28

Solutions

- Baseline:
  - find quasi-cliques at individual timestamps
  - Greedily maximize coverage (submodular)
  - (-) does not consider temporal evol.
- Solution:
  - Prune vertices and edges
  - Divide-and-conquer on remainder
  - Divide: candidates including **v** and those excluding **v**, for every node **v**
- Optimal search prioritization for scalability

- Pruning:
  - Degree-and-duration
  - Distance-based
  - Pattern size
  - Vertex-based
  - Diversity-based

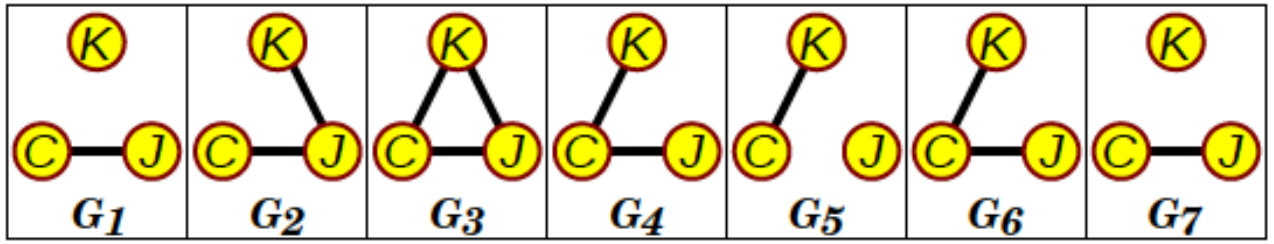
Table 3: Running time (in sec)

Dataset	Baseline	Complete	Quick
DBLP coauthor	>864,000	198	198
Enron employees	>864,000	>864,000	2,417
Facebook wall posts	>864,000	560	560
Linux kernel mails	>864,000	7,293	958
Slashdot threads	>864,000	3,697	2,438
Wikipedia simple-En	1,656	1	1

Unstable Communities in Network Ensembles

[Rahman et al. SDM'16]

- UC: Sets of nodes forming unstable "configurations"
- Measure in terms of subgraph distributions
- **Subgraph divergence:** relative entropy of the observed subgraphs compared to uniform
- **Problem**: Find all maximal size-scaled subgraphs of divergence exceeding a threshold T
  - NP-hard
  - Anti-monotonicity -> Apriori-style solution

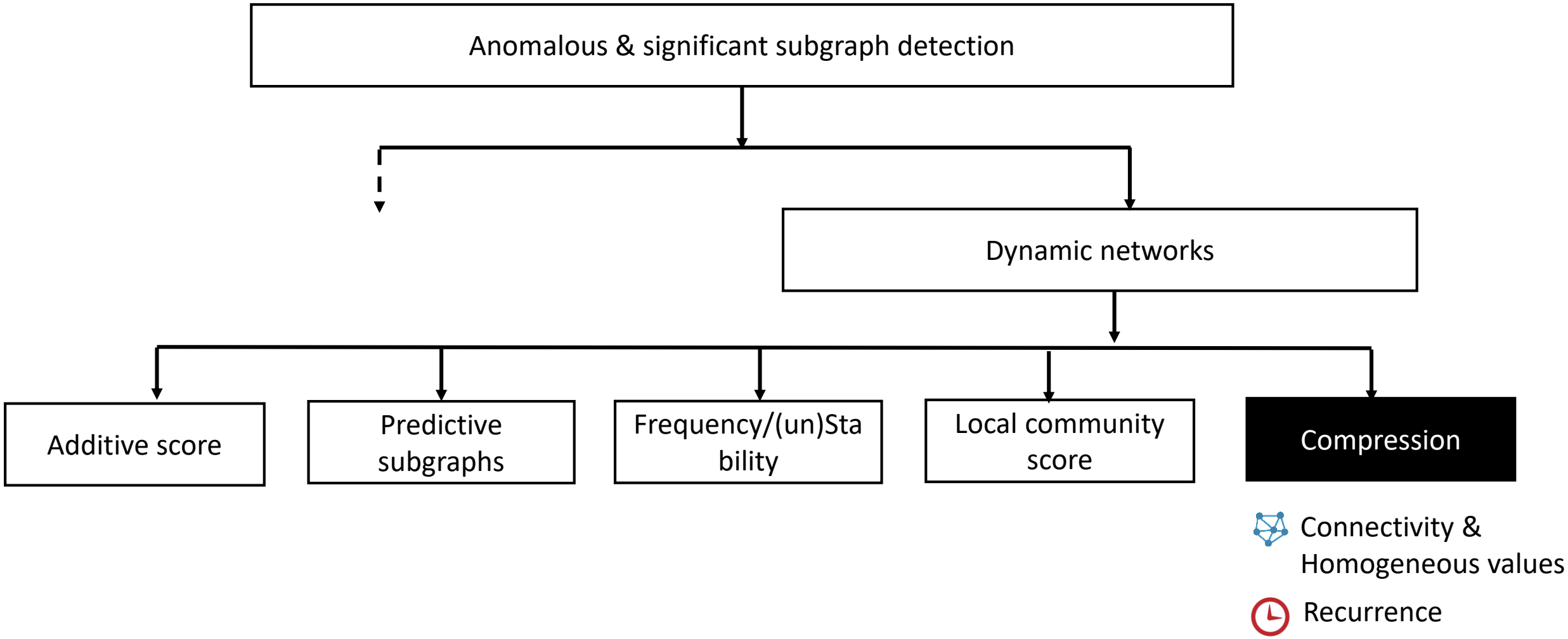


$$\Pr(X = G) = \frac{\psi_{\mathcal{G}}(G)}{|\mathcal{G}|} \qquad \Pr(Y = G) = \frac{1}{2^{\binom{k}{2}}}$$

$$S_{\mathcal{G}}(U) = \sum_{\substack{G \in \mathcal{P}(U) \\ \psi_{\mathcal{G}}(G) \neq 0}} \Pr(X = G) \log_2 \left(\frac{\Pr(X = G)}{\Pr(Y = G)} \right)$$

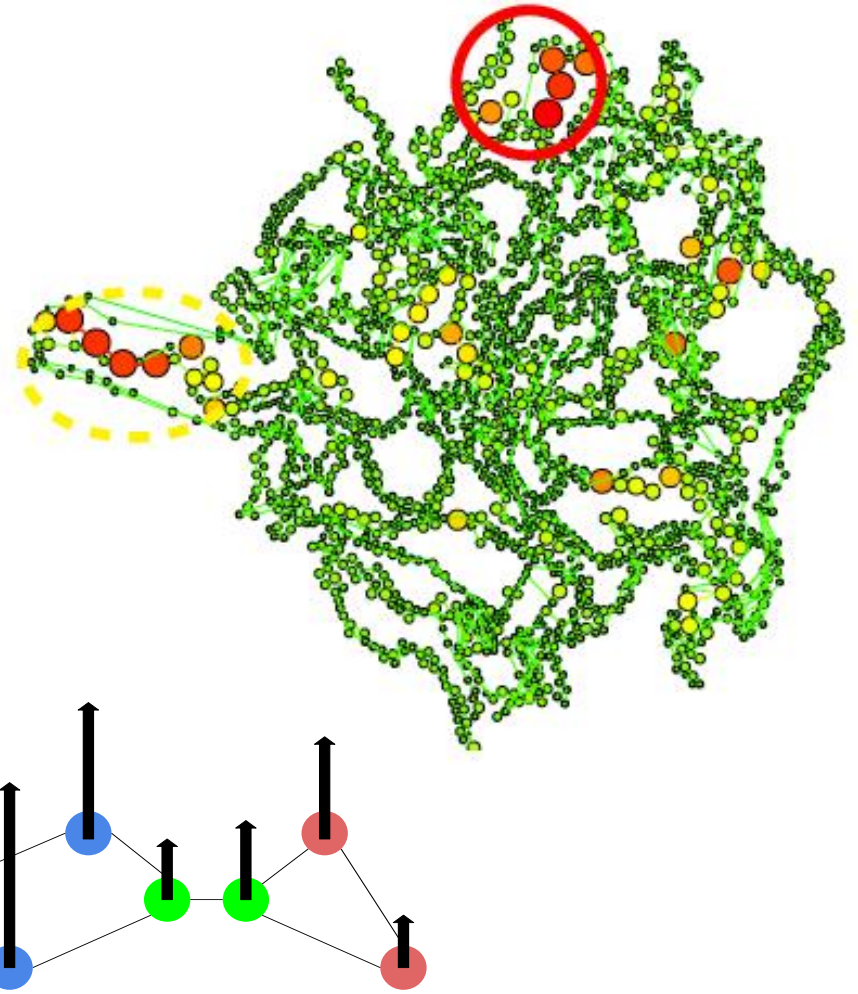
$$= \binom{k}{2} + \sum_{\substack{G \in \mathcal{P}(U) \\ \psi_{\mathcal{G}}(G) \neq 0}} \frac{\psi_{\mathcal{G}}(G)}{|\mathcal{G}|} \log_2 \frac{\psi_{\mathcal{G}}(G)}{|\mathcal{G}|}$$

Taxonomy


Anomalous & significant subgraph detection
Dynamic networks
Additive score
Predictive subgraphs
Frequency/(un)Stability
Local community score
Compression
Connectivity & Homogeneous values
Recurrence

A network with node values: compress values

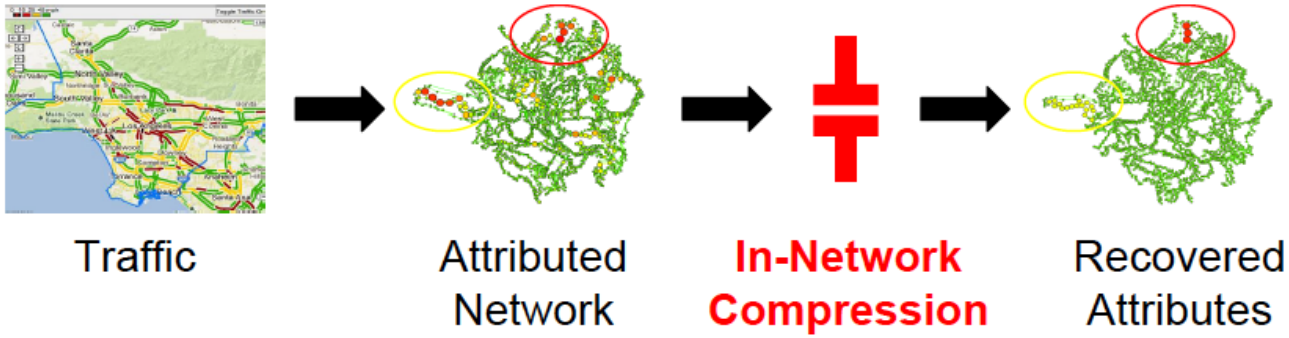
- How to detect and summarize regions that “stand out” (modified by local processes)?
- **Score:** compression quality
- Different from structure-based compression
  - The structure is viewed as the space in which the data (attributes) is embedded
  - Ideas from *Graph Signal Processing*
- *Applications:*
  - *Concise representation in dynamic/multi-network cases*
  - *Detect anomalous-value subgraphs*

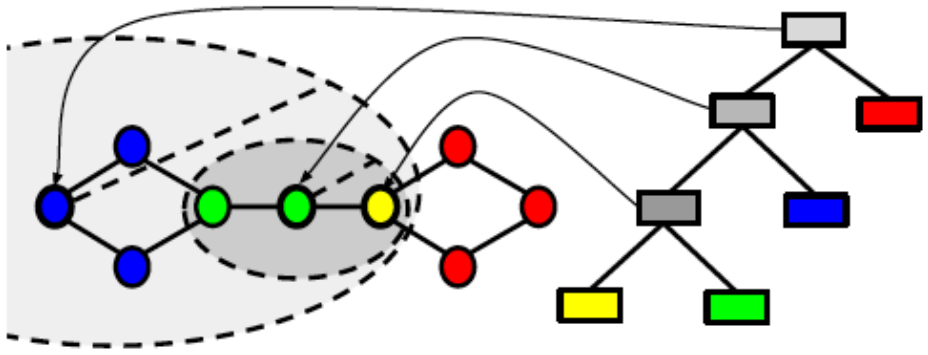


In-network attribute compression

[Silva et al. ICDE'15]

- Lossy compression of node/edge states
- Fixed budget (e.g. bits)
- Network structure is known

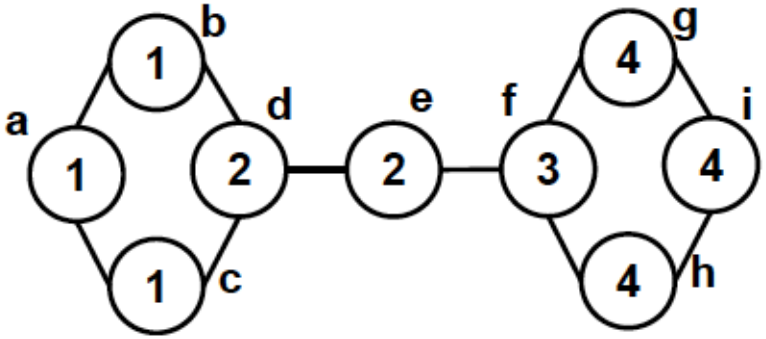




- Extract homogeneous center+radius regions
- Repeat recursively
- Values in leaves represented as their averages

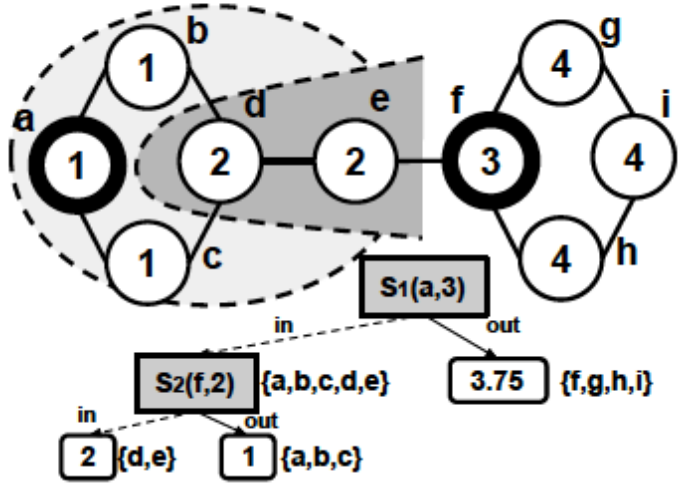
Slice trees and challenges

- Slice S(c,r,X)
  - c: center; r: radius; X: a subset of all nodes V
  - A slice partitions X in P {p: d(c,u)<r} and X \ P
- SliceTree(G,k)
  - A binary tree that encodes k recursive slices in G
  - Compress each leaf as mean value
- Optimal SliceTree(G,k) is NP-hard (Red. from vertex cover)



Advantages and Challenges

- Better compression than existing work
  - Haar trees: hierarchical clustering + Haar wavelets *[Gavish et al. 10]*
  - Graph Fourier *[Zarni et al. 00, Shuman et al. 13]*
- Compact basis (regions/partitions) representation
- Finding the optimal partitions is NP-hard



Greedy SliceTree using importance sampling

1. Identify the slice of max error reduction (O(|V||E|)-> sampling)
2. Include it into the SliceTree
3. Repeat k times (Budget)

Biased sample

Importance sample B:

$$p(v) = \frac{|w(v) - \mu(X)|}{\lambda}$$

Unbiasing $\mu(B_P)$:

$$\mu(B_P) = \frac{1}{\Psi(B_P)} \sum_{v \in B_P} \frac{\lambda}{|w(v) - \mu(X)|} w(v)$$

where $B_P = B \cap P$ and $\Psi(B_P) = \sum_{v \in B_P} \frac{\lambda}{|w(v) - \mu(X)|}$

$$\mathbb{E}[\mu(B_P)] = \mu(P)$$

$$SSE(Y) = \sum_{v \in Y} (w(v) - \mu(Y))$$

SSE Reduction:

$$\phi(s) = SSE(X) - SSE(P) - SSE(X \setminus P)$$

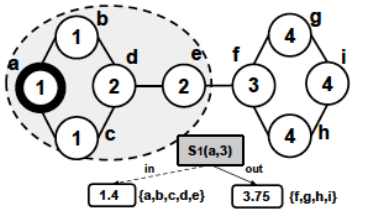
$$\dots$$

$$= (\mu(X) - \mu(P))^2 \frac{||P||||X||}{||X \setminus P||}$$

Bounding reduction:

Given:

- Sample S (uniform/biased)
- Confidence parameter δ
- Slice s



$$\phi(s) = (2.44 - 1.4)^2 5.9/4 = 14.22$$

$\phi(s)$ **can be bounded as:**

$$Pr[\phi(s) < \max\{\phi_1(s), \phi_2(s)\}] > 1 - \delta$$

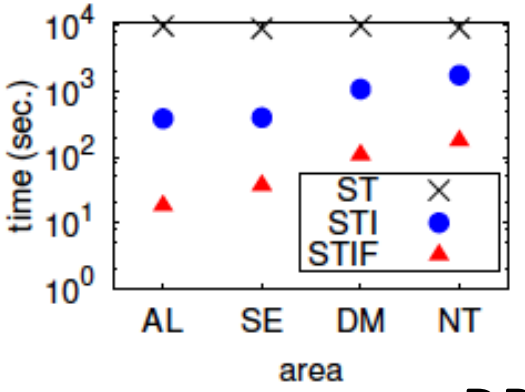
where:

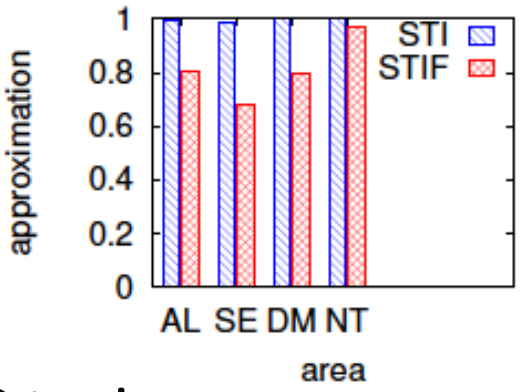
$$\phi_1(s) = (\mu(X) - \mu(S_P) - \epsilon)^2 \frac{||P||||X||}{||X \setminus P||}$$

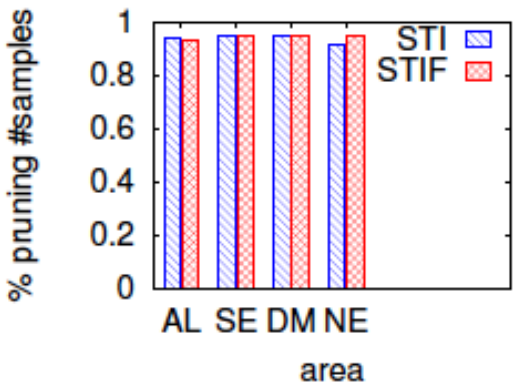
$$\phi_2(s) = (\mu(X) - \mu(S_P) + \epsilon)^2 \frac{||P||||X||}{||X \setminus P||}$$

$$\epsilon = \sqrt{\frac{-\theta^2}{2||S_P||} \log(\frac{\delta}{2})}, \theta = \max_{u,v \in X} |w(u) - w(v)|, S_P = S \cap P$$

Importance sampling



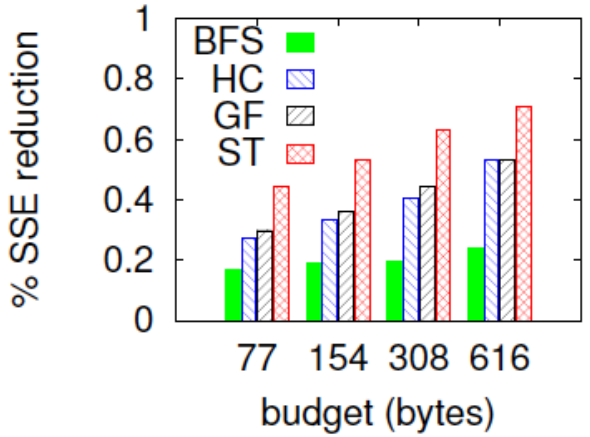




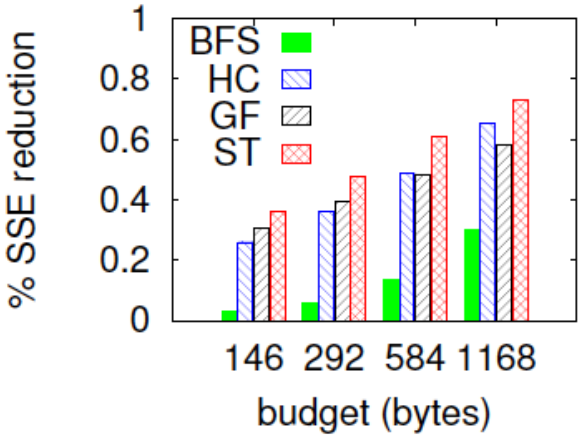
DBLP topics

- 500x faster execution than naïve ST
- Small error + control over it

- 80% reduction of SSE error with small number of bits (compared to full data)
- Exploits smoothness of attributes in the network



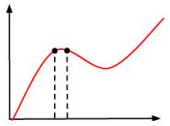
Traffic

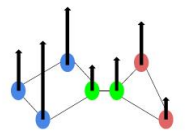


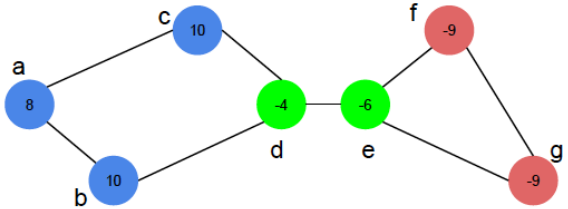
Gene expression

Graph wavelets via sparse cuts

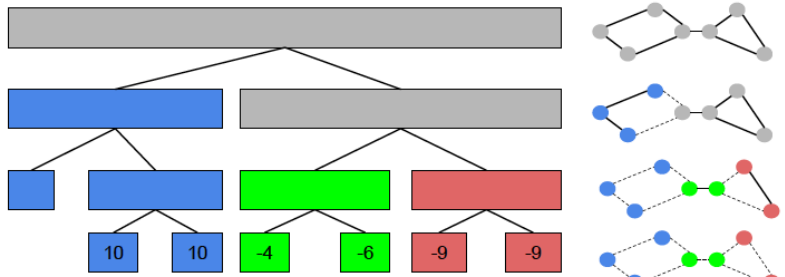
[Silva et al. KDD'16]

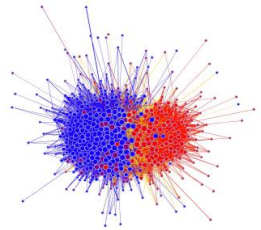




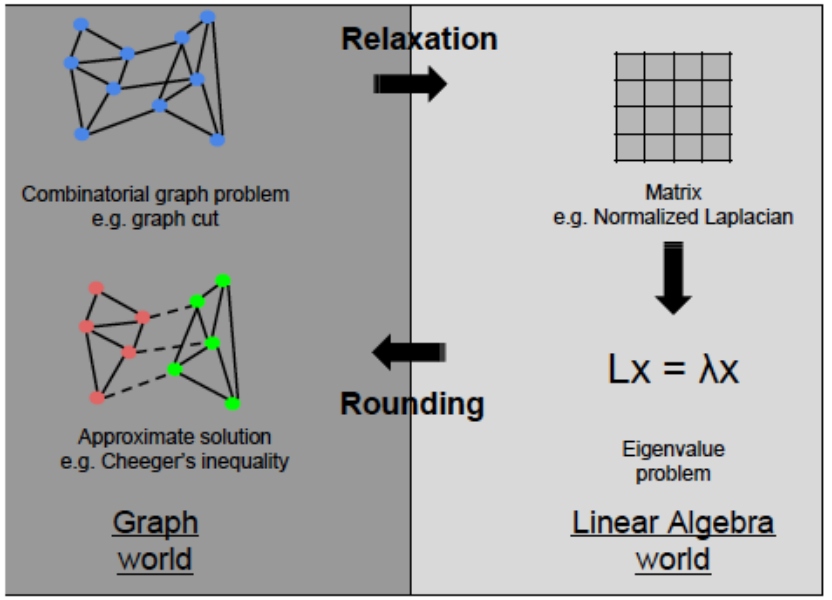


Partitioning





- Idea: Learn a graph wavelet basis
  - Sparse cut
  - Smooth signal
- Signal processing on graphs (predefined basis)
  - Graph Fourier basis
  - Diffusion wavelets
  - **Partitioning** wavelets
- Partitioning wavelets
  - Advantageous, but partition crytical
  - *How to incorporate the signal in learning the basis (partitions)?*



Graph-regularized signal smoothness

- Learn a partition (+1/-1) vector x*
  - Minimize the square signal difference within the partition $x^T CSCx$
  - That is also sparse: $x^T Lx$
- After relaxation (x: real values) and substitution -> spectral clustering
- Ideas to speed up the solution from cubic complexity
  - Chebyshev polynomials
  - Power method

$$\mathbf{x}* = \min_{\mathbf{x} \in [-1,1]^n} \frac{\mathbf{x}^T CSC\mathbf{x}}{\mathbf{x}^T C\mathbf{x} + \beta \mathbf{x}^T L\mathbf{x}}$$

Laplacian matrix of G: $L = D - A$

Laplacian of a complete graph: $C = n\mathbf{I} - \mathbf{1}_{n \times n}$

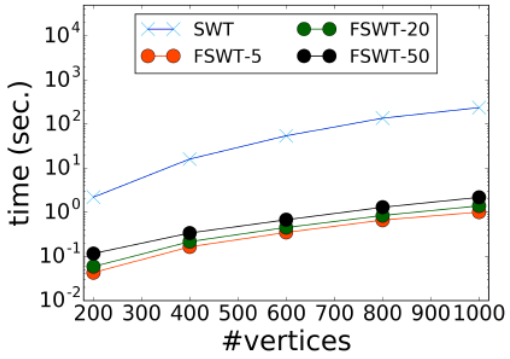
Signal matrix: $S_{u,v} = (W(u) - W(v))^2$

Using substitution $\mathbf{x} = ((C + \beta L)^+)^{\frac{1}{2}} \mathbf{y}$

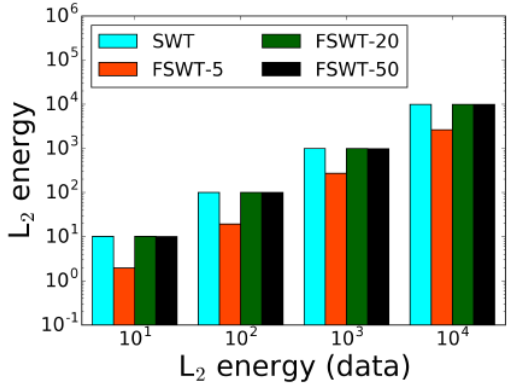
$$\mathbf{y}* = \min_{\mathbf{y}} \frac{\mathbf{y}^T M\mathbf{y}}{\mathbf{y}^T\mathbf{y}}$$

Results

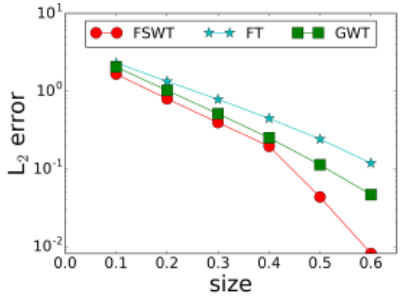
- 100x speed-up compared to exact, similar quality
- Superior performance to graph wavelet baselines



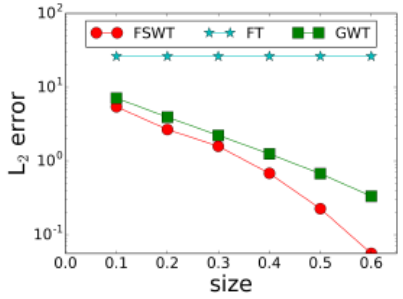
Scalability



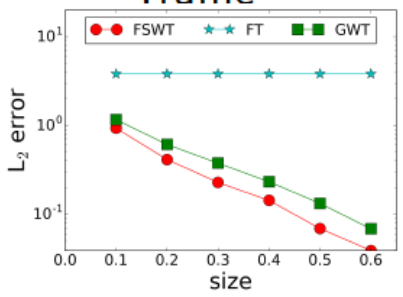
Approximation



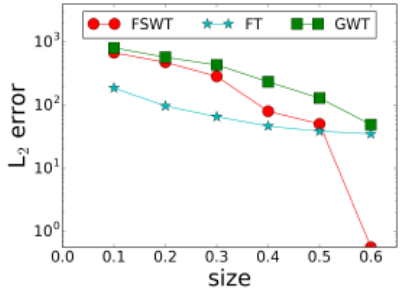
Traffic



Human



Wiki



Blogs

Dynamic setting: looking ahead …

- Simpler, but still applicable formulations
  - Need constraints to avoid trivial solutions
  - Time and graph dimensions are not naturally comparable
- Better understanding of the solution space
  - Bounds
  - Indexing
  - Sampling with guarantees
- Parallel/Distributed techniques

Additive score	Predictive subgraphs	Frequency/ (un)Stability	Local community score	Compression
• Control for overlap • Dyn. structure	• Better kernels • Advanced ML	• Simpler/Unified formulations	• Other community measures: cuts, modularity…	• Structure+attributes • Temporal

Acknowledgements

- F. Chen would like to acknowledge support from NSF Grant IIS-1441479
- P. Bogdanov would like to acknowledge support from NSF Grant DMR-1309410 (Subaward: 75595)
- D. Neill would like to acknowledge support from NSF Grants IIS-0953330, NSF IIS-0916345, NSF IIS-0911032 and funding from the John D. and Catherine T. MacArthur Foundation
- A. Singh would like to acknowledge support from Army Research Lab NS-CTA W911NF-09-2-0053 and NSF Grant III-1219254

Part II: References

- Rezwan Ahmed and George Karypis. Algorithms for mining the evolution of conserved relational states in dynamic networks, ICDM, 2011
- Rezwan Ahmed and George Karypis, Algorithms for mining the coevolving relational motifs in dynamic networks, ACM Transactions on Knowledge Discovery from Data (TKDD), 2015
- Petko Bogdanov, Misael Mongiovi, and Ambuj K Singh, Mining heavy subgraphs in time-evolving networks. ICDM, 2011
- Xuan Hong Dang, Ambuj K Singh, Petko Bogdanov, Hongyuan You, and Bayyuan Hsu, Discriminative subnetworks with regularized spectral learning for global-state network data. PKDD, 2014.
- Xuan Hong Dang, Hongyuan You, Petko Bogdanov, and Ambuj K Singh. Learning predictive substructures with regularization for network data. ICDM, 2015
- Saket Gurukar, Sayan Ranu, and Balaraman Ravindran. Commit: A scalable approach to mining communication motifs from dynamic networks. SIGMOD, 2015.
- Misael Mongiovi, Petko Bogdanov, Razvan Ranca, Ambuj K Singh, Evangelos E Papalexakis, and Christos Faloutsos. Netspot: Spotting signicant anomalous regions on dynamic networks, SDM, 2013.

Part II: References

- Misael Mongiovi, Petko Bogdanov, and Ambuj K Singh. Mining evolving network processes. ICDM, 2013.
- Ahsanur Rahman, Steve TK Jan, Hyunju Kim, B Aditya Prakash, and TM Murali. Unstable communities in network ensembles, SDM, 2016.
- Sayan Ranu, Minh Hoang, and Ambuj Singh. Mining discriminative subgraphs from global-state networks. SIGKDD, 2013.
- Polina Rozenshtein, Nikolaj Tatti, and Aristides Gionis. Discovering dynamic communities in interaction networks. PKDD, 2014.
- Arlei Silva, Petko Bogdanov, and Ambuj K Singh. Hierarchical in-network attribute compression via importance sampling. ICDE, 2015.
- Arlei Silva, Xuan-Hong Dang, Prithwish Basu, Ambuj K Singh, and Ananthram Swami. Graph wavelets via sparse cuts. SIGKDD, 2016
- 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.

Thank you!

Feng Chen
Petko Bogdanov

Daniel B. Neill

Ambuj K. Singh

?

Questions

IEEE BigData