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**
• Conclusion and 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 to dynamic setting

• Combine the graph and temporal (multi-network) dimension
  • Connectivity
  • Contiguity

• Combination of exhaustive static solutions are disadvantaged
  • Do not consider all slices at a time
  • Agreement of results as a heuristic post-processing

• We will focus on subgraph problems
  • E.g. evolutionary clustering does not fall in this rubric
Taxonomy

Anomalous & significant subgraph detection

Dynamic networks

Additive score
Predictive subgraphs
Frequency/Stability
Local community score
Compression of attributes

+ Static
Taxonomy

Anomalous & significant subgraph detection

Dynamic networks

Additive score

Predictive subgraphs

Frequency/Stability

Local community score

Compression of attributes
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

Affected Subgraph

Affected Interval

TIME

Dynamic processes:
- fact search

- Persian language
- History of Iran
- Iran
- Qajar Dynasty
Mining heavy dynamic subgraphs

[Boqdanov 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 (t² intervals)
Baseline filter-and-verify approach

- **Discard “bad” intervals**
  - UB on score $O(|E|)$
  - Discard interval if UB < global LB
- **TopDown:** Evaluate remaining intervals
  - Transform to PCST-Tree and solve exactly
  - Faster than existing PCST heuristics \cite{Johnson00}
- **Challenge:** Quadratic filtering cost $O(t^2 |E|)$
  - Need to consider every interval
  - Traffic: 1 month, every 5 min \( t=8640 \)
- **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 $[j,k]$, $i \leq j \leq k$
- Minimum overlap $\alpha = (j-i+1)/(k-i+1)$: The length ratio of shortest and longest interval
- If $0 < \alpha < 1$, then $\#\text{groups} = O(t \log(t))$
How to Filter Whole Groups?

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

\begin{align*}
\text{Time} & \quad i \quad j \quad k \\
\begin{array}{c}
\includegraphics[width=0.5\textwidth]{g.png}
\end{array}
\end{align*}
Time to build index $O(t |E|)$, Bottom-up
Compose one $\hat{G}$ in $O(\log(t) |E|)$
Compute all $\hat{G}$ in $O(t \log^2(t) |E|)$ vs $O(t^2 |E|)$
MEDEN: Prune Intervals

Size

Starting Time

Verify
An order of magnitude speedup

Scalability

No Filtering

Filtering, No Grouping

Almost all intervals filtered as groups
Identify All Significant Regions

[Mongiovi et al. SDM’13]

• Separate the time from graph dimension
• Large-neighborhood Local Search
• Informed seed generation
• Handle overlap
So far: stable solution in time

- No change in participating nodes/edges
- Processes in real networks might shift and change in size
- Idea: incorporate smoothness
  - As a constraint
  - As 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 $\alpha$ edges/nodes change at a time
  - Similar smoothness criteria in evolutionary clustering
    \cite{Chakrabarti06,Chi07}
- **Max-score $\alpha$–smooth** subgraph: NP hard even on trees (reduction from 3SAT)
  - 0-smooth is polynomial on trees
Overview of Solution

- **Dynamic Network**
- **No-loss Filtering**
- **Accurate Search**
- **No-loss Filtering**
- **Smooth Process**

Reduce the instance size

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

No Loss of Accuracy

Candidate search

UB(v,t) < LB

Spot

∞-smooth candidates

“Smooth” each candidate according to α
High-Quality Search: Spot&Smooth

Spot ∞-smooth candidates

“Smooth” each candidate according to α

Best paths connecting positive components

\( \alpha = 1 \)

t=1

t=2

remove
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 constraints: selection of a subset at $t$ depends on 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
**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
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
DIPS: Graph regularization+sparsity

\[ \text{[Dang et al. ICDM’15]} \]

\[ S_i: \text{network instance} \quad l_i: \text{network label} \]

\[ S_1, S_2, S_3, \ldots, S_n \]

Similar networks with similar labels

Similar network with dissimilar labels

Meta graph defined on Si’s

Learn a low dimensional embedding subspace \( Y \)

Discriminative substructures for global network property prediction

Network regularization

Sparness regularization

\[ \mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T, \ldots, \mathbf{v}_n^T \]

Fitting error

Prediction

Explanation

(Preserve network similarity and network discriminatory)
Notation

• Let $DS = \{S_1, S_2, \dots, S_n\}$ be a dataset of network instances
• Let $S_i = \{V_i, E_i, F\}$ where
  $V_i = \{v_1, v_2, \dots, v_{m_i}\}$: a set of nodes
  $E_i \subseteq V_i \times V_i$: a set of undirected edges
  $F$: node labeling function
  $\ell_i$: global state/property of $S_i$
• Let $S = \{V, E, W\}$ be a network aggregated from all $S_i$'s
  
  $V = V_1 \cup V_2 \cdots \cup V_i$, $E \subseteq V \times V$
  $W(p,q) = n^{-1} \times \sum E_i(p,q)$
• Find substructures that highly predict
  global network labels
Subspace Learning

1. Construct two meta-graphs network instances representing
   - similar networks & similar global properties
     - meta-graph $G^+$ and its adjacency matrix $K^+$
   - similar networks & dissimilar global properties
     - meta-graph $G^-$ and its adjacency matrix $K^-
   - use simple notion of similarity such as cosine

\[
K^+_{ij} = \begin{cases} 
\frac{v_i \cdot v_j}{\|v_i\|\|v_j\|}, & \text{if } \ell_i = \ell_j, S_j \in kNN(S_i) \text{ or } S_i \in kNN(S_j) \\
0, & \text{otherwise}
\end{cases}
\]
2. Project instance $S_i$ to $y_i$ while optimizing the objective functions for similarity and dis-similarity:

\[
\text{minimize } \sum_i \sum_j \|y_i - y_j\|^2 K_{ij}^+ \\
= \sum_i \sum_j (\|y_i\|^2 + \|y_j\|^2 - 2y_i^T y_j)K_{ij}^+ \\
= 2Y^T (D^+ - K^+) Y = 2\text{tr}(Y^T L^+ Y)
\]

\[
\text{maximize } \sum_i \sum_j \|y_i - y_j\|^2 K_{ij}^- \\
= 2Y^T (D^- - K^-) Y = 2\text{tr}(Y^T L^- Y)
\]

Combined function: $Y = \arg \max_Y \left\{ \text{tr} \left( Y^T (L^- - \beta L^+) Y \right) \right\}$

subject to $Y^T D^+ Y = I$

With solution: $LY = ADY$
Sparse Subnetworks

Projection matrix $Y$

- each row embeds a network instance
- each column defines a discriminating dimension

3. Add constraints based on topology and sparsity

$$\arg\min_U \sum_i \sum_j \|Y - V^T U\|_F + \lambda_2 \sum_{u \in U} u^T Cu + \lambda_1 \sum_{u \in U} |u|$$

- each column in $V$ encodes the node attributes of a network instance
- $U$ selects and combines nodes to form embedding subspace
Sparse Subnetworks

\[
\arg\min_U \sum_i \sum_j \|Y - V^T U\|_F^2 + \lambda_2 \sum_{u \in U} u^T C u + \lambda_1 \sum_{u \in U} |u|
\]

- C is the Laplacian matrix for the aggregated network S
  \[
  C_{pq} = C_{qp} = \begin{cases} 
  \sum_q W(p, q) & \text{if } v_p = v_q \\
  -W(p, q) & \text{if } v_p, v_q \text{ are connected} \\
  0 & \text{otherwise}
  \end{cases}
  \]

- Impose network “soft” constraint on the solution
Learning sparse subnetwork structure

- Non-smooth and strictly convex
- Apply coordinate gradient descent method

\[ u_i = \begin{cases} 
\frac{(a_2 + \lambda_2)}{a_1} & \text{if } a_2 < -\lambda_1 \\
\frac{(a_2 - \lambda_2)}{a_1} & \text{if } a_2 > +\lambda_1 \\
0 & \text{if } a_2 \in [-\lambda_1;+\lambda_1]
\end{cases} \]

\[ a_1 = 2 \sum_i v_{ii}^2 + 2\lambda_2 C_{ii} \]

\[ a_2 = 2 \sum_i v_{ii} (v_i - \bar{u}^T \bar{v}_i) - 2\lambda_2 \bar{u}^T \bar{C}_i \]

- Final node selection:

\[ u = (u_1, \ldots, u_m)^T \quad \text{with} \quad u_p = \max_i |u_{i,p}| \]
Results - Quality

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

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

Fig1. Prediction accuracy

**Image Networks**

- 1920 nodes
- 11172 edges
- 156 network samples
  (sunglass vs naked-eye)
Sparse Subnetworks

Brain Networks

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

Gene Expression Networks

Embryonic development

- 1321 genes
- 5277 edges
- 35 network samples

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
Frequent and stable structure

- Goal: find frequently occurring/conserved subnetworks
  - Communication threads
  - Interactions that persist in time
- Score is related to a property of the structure over time
- Need constraints to avoid trivial solutions
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?
Temporal relationships, graphs and subgraphs

- Two adjacent edges are **temporally related** if the difference in their timestamps is less than $\Delta T$
- Two nodes are **temporally connected** if there exists a sequence of adjacent edges $P = \{e_1, \ldots, e_m\}$ connecting them and $\forall e_i, e_{i+1} \in P$, $e_i$ and $e_{i+1}$ are temporally related
- **Temporally connected graph**: a subset of temporally connected nodes

\[ \Delta T = 2 \]
Temporally-isomorphic subgraphs

- Isomorphic subgraphs with the same sequence of information flow

- Problem: Given
  1. Dynamic Interaction Network
  2. Temporal connectivity threshold $\Delta T$
  3. Support threshold $\tau$

Mine:
All **connected, temporally isomorphic subgraphs** with support above $\tau$

**But, NOT** temporally isomorphic:

At $\tau = 3$, $\Delta T = 1$
COMMIT: Pipeline and main ideas

Encoding sequences

Replace node labels with degrees

Encode edges in sequence

Sequence growth

Communication motif candidates in sequence space

Dynamic network

Graph-invariant based conversion to sequences

Reverse mapping to graph space

Sequence database

Subsequence Pattern Mining

Communication Motifs

Sequence growth from \{1,3\} to \{1,3\}, \{1,3\}, \{1,3\}

Support Set of \{(1,3), (1,3)\}

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)

(1, 3, 4, 5)
Motifs and scalability

- GRAMI [Kalnis et al., VLDB’14] inevitably crashes in around 16 hours.
- COMMIT finishes in around 15 minutes.

Further questions:
- How do I know $\Delta T$?
- Isomorphic structure too strict for real life. Relax?

Mined motifs

<table>
<thead>
<tr>
<th>Twitter Mentions</th>
<th>Facebook Wall Posts</th>
<th>Enron Email</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Twitter Mentions" /></td>
<td><img src="image" alt="Facebook Wall Posts" /></td>
<td><img src="image" alt="Enron Email" /></td>
</tr>
</tbody>
</table>

(a) Twitter  
(b) Facebook
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
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 \(B \geq T_1 + \ldots + T_k\), 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:**

$$
q(W, T \cup R, G) - q(W, T, G)
$$

$$\max(x, y)
$$

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

![Graph showing community density vs. number of random initializations]
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 \(I'\)

• Coverage: the nodes over time that the temporal graph includes

• 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-Quasiclisses Coverage: 28
Solutions

• Baseline:
  • find quasi-cliques at individual timestamps
  • Greedily maximize coverage (submodular)
  • (-) does not take advantage of time

• 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 Based Pruning
  • Distance Based Pruning
  • Pattern Size Based Pruning
  • Vertex Based Pruning
  • Diversity Based Pruning

<table>
<thead>
<tr>
<th>Table 3: Running time (in sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>DBLP coauthor</td>
</tr>
<tr>
<td>Enron employees</td>
</tr>
<tr>
<td>Facebook wall posts</td>
</tr>
<tr>
<td>Linux kernel mails</td>
</tr>
<tr>
<td>Slashdot threads</td>
</tr>
<tr>
<td>Wikipedia simple-En</td>
</tr>
</tbody>
</table>
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 divergence exceeding a threshold $T$
  • NP-hard
  • Anti-monotonicity -> Apriori-style
Taxonomy

Anomalous & significant subgraph detection

Dynamic networks

Additive score

Predictive subgraphs

Frequency/(un)Stability

Local community score

Compression
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 seen as the space in which the data (attributes) is embedded
  • Related to *Graph Signal Processing*
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 $X \setminus P$
• 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 $\Psi(B_P) = \sum_{v \in B_P} |w(v) - \mu(X)|$

We show that: $\mathbb{E}[\mu(B_P)] = \mu(P)$

SSE Reduction:

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

$$...$$

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

Bounding reduction:

Given:
- Sample $S$ (uniform/biased)
- Confidence parameter $\delta$
- Slice $s$

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

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

where:

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

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

$$\epsilon = \sqrt{\frac{-\theta^2}{2||S_P||} \log \left( \frac{\delta}{2} \right)}$$

where:

$$\theta = \max_{u,v \in X} |w(u) - w(v)|, S_P = S \cap P$$

Figure: $\phi(s) = (2.44 - 1.4)^2 \frac{5.9}{4} = 14.22$
Importance sampling

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

DBLP topics

- 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]

- 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 critical
  - *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 $x^TCSCx$
  - That is also sparse: $x^TLx$
- After relaxation ($x$: real values) and substitution -> spectral clustering
- Speed up from cubic complexity
  - Chebyshev polynomials: $O(pmn)$ \[ \text{[HVG11]} \]
  - Power method: $O(tn^2)$

\[
x^* = \min_{x \in [-1, 1]^n} \frac{x^TCSCx}{x^TCx + \beta x^TLx}
\]

Laplacian matrix of $G$: $L = D - A$
Laplacian of a complete graph: $C = nI - 1_{n \times n}$
Signal matrix: $S_{u,v} = (W(u) - W(v))^2$

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

\[
y^* = \min_y \frac{y^TM_2y}{y^Ty}
\]
Results

• 100x speed up compared to exact, similar quality

• Superior performance to graph wavelet baselines
Part II: References


• 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


Part II: References


• Sayan Ranu, Minh Hoang, and Ambuj Singh. Mining discriminative subgraphs from global-state networks. SIGKDD, 2013.


• Arleí Silva, Petko Bogdanov, and Ambuj K Singh. Hierarchical in-network attribute compression via importance sampling. ICDE, 2015.

• Arleí Silva, Xuan-Hong Dang, Prithwish Basu, Ambuj K Singh, and Ananthram Swami. Graph wavelets via sparse cuts. SIGKDD, 2016