Focused Clustering and Outlier Detection in Large Attributed Graphs
Focused Clustering and Outlier Detection in Large Attributed Graphs

- Many real-world graphs have attributes associated with the nodes, in addition to their connectivity information.

- A protein-protein interaction network (interaction relations and gene expressions associated with the proteins)

- Social networks (friendship relations as well as user attributes such as interests and demographics)

- Attributed graph: nodes, edges, feature vectors associated with the nodes

- Graph clustering and graph outlier detection have been studied extensively on plain graphs. Recently, algorithms have been extended to graphs with attributes as often observed in the real-world.

- All of these techniques fail to incorporate the user preference into graph mining.

- Previous methods use all the given attributes or they perform an unsupervised feature selection
Focused Clustering and Outlier Detection in Large Attributed Graphs

• A marketing manager interested in selling cosmetics aim to find communities in a large social network with certain age, gender, and income-level.

• new method: cluster and outlier detection based on user preference (focused clustering), users might not be concerned with all but a few available attributes.

• introduce a novel user-oriented approach for mining attributed graphs

• The key aspect of approach is to infer user preference by the focus attributes through a set of user-provided exemplar nodes.

• FocusCO algorithm identifies the focus, extracts focused clusters and detects outliers.
Focused Clustering and Outlier Detection in Large Attributed Graphs

Figure 1: Example graph with two focused clusters and one focused outlier.
Focused Clustering and Outlier Detection in Large Attributed Graphs

- Given a large attributed graph $G(V, E, F)$ with $|V| = n$ nodes and $|E| = m$ edges, each node is associated with $|F| = d$ attributes (features), **extract from $G$ only the clusters pertaining to a user $u$’s interest**

- the user provides a small set $C_{ex}$ of exemplar nodes

- calculate the weights (relevance) of attributes that “define” the nodes in $C_{ex}$, i.e., **the weights of attributes that make the nodes as similar as possible**.

- large weights for only a few attributes (e.g., degree and location) which we call the focus attributes
• **Given** a large graph $G(V, E, F)$ with node attributes, and a set of exemplar nodes $C_{ex}$ of user $u$’s interest;

• **Infer** attribute weights of relevance/importance,

• **Extract** focused clusters $C$ that are (1) dense in graph structure, and (2) coherent in heavy focus attributes,

• **Detect** focused outliers $O$, i.e. nodes that deviate from their cluster members in some focus attributes.
Focused Clustering and Outlier Detection in Large Attributed Graphs

- Inferring Attribute Relevance:
  - identify the relevance weights of node attributes that make the exemplar nodes similar to each other.
  - the distance between two nodes with feature vectors $f_i$ and $f_j$ is $(f_i f_j)^T A (f_i f_j)$
  - optimization objective

$$
\min_A \sum_{(i,j) \in P_S} (f_i - f_j)^T A (f_i - f_j)
- \gamma \log \left( \sum_{(i,j) \in P_D} \sqrt{(f_i - f_j)^T A (f_i - f_j)} \right)
$$
Focused Clustering and Outlier Detection in Large Attributed Graphs

- Focused Cluster Extraction: 
  \[ w(i, j) = \frac{1}{1 + \sqrt{(f_i - f_j)^T \text{diag}(\beta)(f_i - f_j)}} \]

- first identify good candidate nodes that potentially belong to such clusters, and then to expand around those candidate nodes to find the clusters.

  - first re-weigh the edges \( E \) by the weighted similarity of their end nodes
  - induce \( G \) on the edges with notably large weights (core set)

  - Next, we expand around each core by carefully choosing new nodes to include in the cluster and continue expanding until there exist no more nodes that increase the quality of the cluster (several measures of cluster quality)

  - The node additions are based on a best-improving search strategy. While being cautious in which node to add (the best one at every step), our decisions are greedy.
Focused Clustering and Outlier Detection in Large Attributed Graphs

- Focused Outlier Detection:

- a node that belongs to a focused cluster structurally (having many edges to its members), but that deviates from its members in some focus attributes significantly is an outlier
Focused Clustering and Outlier Detection in Large Attributed Graphs

- To evaluate focused clustering quality, we use the Normalized Mutual Information (NMI), a metric for computing clustering accuracy.
- FocusCO remains to all the other approaches in the face of irrelevant attributes for the clustering task.

*Figure 2: NMI vs. attribute size $|F|$. Results averaged over 100 runs, bars depict 25-75%.*
In (a), we increase the cluster size where we create clusters of the same size in the graph. In (b), we allow the graph to contain variable size clusters and increase the variance of the cluster sizes, by randomly drawing them from increasing ranges. Finally in (c), we increase the number of unfocused clusters in the graph, while keeping the number of focused clusters fixed.

Figure 3: NMI clustering quality results on synthetic graphs, for FocusCO, CODA with three different $\lambda$ parameter settings, and METIS on unweighted graphs; (a) for changing cluster sizes (all clusters have the same size), (b) for changing cluster size variance (graph has variable size clusters), and (c) for increasing number of unfocused clusters. FocusCO performs the best in all scenarios across a wide range of settings. Symbols depict the mean over 100 runs, bars depict 25-75%.
• one types of focused clusters in Disney (Disney is an Amazon co-purchase graph of Disney movies):

✦ User wants to understand how the popularity of a movie influences its community. The user decides that the product’s popularity is related to the features Number_of_reviews and Sales_rank, and so chooses a few products which have similar values in those attributes. FocusCO then uses this exemplar set \( C \) to learn an attribute weighting \( \eta \). This reflects the user’s intent, and has also captured another dimension correlated with those attributes; Number_of_different_authors
The first focused cluster (blue) reflects traditional Disney classics such as Robinhood. Its outlier is a sequel (An Extremely Goofy Movie) that is much less popular than the other classics in the cluster. The second community (green) focuses on popular older Disney movies, and has outliers such as American Legends and again the Goofy sequel, that are much less popular. The third cluster (orange) overlaps with the first focused cluster. It is a subset of the classic Disney movie of the larger cluster that were predominantly starred by animals (e.g., The Rescuers).
• PolBlogs, where a user seeks to understand the difference between blog content written by different liberal bloggers during Iraq war controversy in 2005.

• Using several liberal blogs as an example set

• The focused community outlier for this group is David Sirota, a well-connected liberal blogger who did not explicitly mention Waas in the dataset

Figure 9: A focused cluster of liberal blogs in PolBlogs with a focus on Iraq war debate. Outlier David Sirota does not mention Waas in his posts.
Event detection in activity networks
Event detection in activity networks

- Detecting events is a fundamental problem in data mining and numerous methods have been applied to a variety of scenarios, including time series and data streams, networks.

- Our goal is to identify parts of the network with unusual high activity confined in a small space or in a dense part of the graph.

- Given a time instance $t$, each node $v \in V$ in the network keeps a value $w_t(v)$ with the measurement value for the monitored activity. The objective of our approach is to detect an event happening in the network at time instance $t$.

- We aim at finding a subset of network nodes $S$, such that all nodes in $S$ are close to each other and they all have high levels of activity.

- In this paper, we assume that the activity values $w(v)$ are provided as input to our problem.
Event detection in activity networks

• Statistical methods:

✦ statistical approach defines a null hypothesis, therefore, it assumes an underlying distribution over the data. Instead, our approach is formalized through an optimization function, and no such assumptions are needed.

✦ usually these approaches assume that the shape is predefined (e.g., a circle or an axis-parallel rectangle)

✦ all these approaches assume that there exists an underlying Euclidean geometry on the space. For several of the applications we are interested, such as social networks, such Euclidean geometry is not present,

• Event detection in social media:

✦ we offer a graph-theoretic formulation to the problem, which can be applied to any graph, not just geography-induced graphs
Event detection in activity networks

• we formalize event detection as a problem of finding a subgraph $S \subseteq V$ in a graph $G = (V,E,w)$ with node weights $w(v)$, for each $v \in V$.

• Our goal is to find a subset of vertices $S \subseteq V$ that has large total weight according to the weight function $w$, and it is sufficiently compact, that is, the vertices in $S$ are close to each other according to the distance function $d$.

• As a set weight function we consider simply the sum of all the weights in the set, that is,

$$W(S) = \sum_{v \in S} w(v).$$
Event detection in activity networks

• For measuring the total distance of a set $S$ we consider two options:

• The first option is to sum the distances of all pairs of vertices in $S$, This type of objective function is suitable for events that are concentrated in a small area in a round-shaped area, for instance a football game

$$D_{AP}(S) = \frac{1}{2} \sum_{v \in S} \sum_{u \in S} d(u,v).$$
Event detection in activity networks

• But events might have different shapes, such as a parade, a street concert, a firework show, and so on; for such events, we need a distance function that does not penalize long distances, as long as points in between are also active. This leads to using the **minimum Steiner tree** of the subgraph induced by the set of vertices S. We denote this total-distance measure by \( D_T(S) \).

\[
D_T(S) = \min_{T \in \mathcal{T}(G[S])} \sum_{(u,v) \in T} d(u,v),
\]

• given a set V of points (vertices), interconnect them by a network (graph) of shortest length, where the length is the sum of the lengths of all edges. In the Steiner tree problem, extra intermediate vertices and edges may be added to the graph in order to reduce the length of the spanning tree.
Event detection in activity networks

- Subsets of vertices $S \subseteq V$ that correspond to meaningful events need to have large weight value $W(S)$ and small total distance value $D(S)$. Consider a linear combination of the two measures into a single objective with a normalization coefficient.

- Find a subset of vertices $S \subseteq V$ that maximizes the objective function

$$Q(S) = \lambda W(S) - D(S).$$
Event detection in activity networks

- EventAllPairs+

\[
Q_{AP}^+(S) = Q_{AP}(S) + D_{AP}(V) \\
= \lambda W(S) - D_{AP}(S) + D_{AP}(V).
\]

- **Greedy algorithms**: the search strategy can be viewed as running two greedy algorithms simultaneously.

- The one greedy process starts from the empty set \(X_0 = \emptyset\); and grows it to optimize \(f\), while the other greedy process starts from the ground set \(Y_0 = V\) and shrinks it to optimize \(-f\). The algorithm guarantees that the growing set \(X\) is always included into the shrinking set \(Y\).

- The algorithm stops when the sets are equal
Event detection in activity networks

- **MaxCut** formulation: For a graph, a maximum cut is a cut whose size is at least the size of any other cut.

- \(((s, t)\text{-MaxCut})\). Given a graph G and two vertices s and t, partition the vertices of G into two sets A and B such that \(s \in A\) and \(t \in B\) and the total weight of cross edges is maximized.

- Assume that we are given a parameter \(\alpha\). We then construct a new graph, H by adding two special vertices s and t into G. We connect s to each \(v \in V\) with some weight and We connect t to each \(v \in V\) with some another weight. an A,B cut of H such that \(s \in A\) and \(t \in B\)
Event detection in activity networks

- minimizing objective function (EventTree):

\[ Q^+_T(S) = \lambda W(V) - \lambda W(S) + D_T(S) = \lambda W(V \setminus S) + D_T(S). \]

- find a set S so that the tree cost \( D_T(S) \) and the (scaled by \( \lambda \)) weight of the vertices not included in S is minimized. This problem is known as the prize-collecting Steiner-tree (PCST) problem. The term “prize collecting” comes from thinking of the weights on the vertices of the graph as prizes and the goal is to find a tree that minimizes the tree cost and the total value of prizes not spanned by it.

- The objective is to select a subset of the nodes such that the cost of the tree to connect them plus the prizes of the nodes not in the tree is minimized.
Event detection in activity networks

- City-sensor data. We use public-bike sharing data from three systems: Barcelona Bicing, Minneapolis Nice Ride and Washington D.C. Capital bikeshare. In each case, a network node corresponds to a bike station. The activity level of a bike station is the fraction of bikes in the station with respect to the full capacity of the station.

- Social-media data. We obtain a dataset of geolocated twitter messages for 100 cities. For our experiments we focus on the cities of New York and Los Angeles. For each city we collapse the locations of the input tweets to centroids, based on proximity, and we use those centroids as network nodes. The activity level of each centroid is the number of tweets during a day.
GLAD: Group Anomaly Detection in Social Media Analysis
GLAD: Group Anomaly Detection in Social Media Analysis

- While people enjoy the openness and convenience of social media, many malicious behaviors, such as bullying, terrorist attack planning, and fraud information dissemination, can happen. Therefore, it is extremely important that we can detect these abnormal activities as accurately and early as possible to prevent disasters and attacks.

- In reality, anomaly may not only appear as an individual point, but also as a group. For instance, a group of people collude to create false product reviews or threat campaign in social media platforms; in large organizations, malfunctioning teams or insider groups closely coordinate with each other to achieve a malicious goal.

- By definition, anomaly detection aims to find “an observation that deviates so much from other observations.

- We are interested in finding the groups which exhibit a pattern that does not conform to the majority of other groups.
GLAD: Group Anomaly Detection in Social Media Analysis

- three major challenges in group anomaly detection:

- Two forms of data coexist in social media: one is the point-wise data, which characterize the features of an individual person. The other is pair-wise relational data, which describe the properties of social ties

- Group anomaly is usually more subtle than individual anomaly. At the individual level, the activities might appear to be normal. Therefore, existing anomaly detection algorithms usually fail when the anomaly is related to a group rather than individuals

- developing a method that can be easily generalized to dynamic setting is critical to anomaly detection in evolving social media data
GLAD: Group Anomaly Detection in Social Media Analysis

- All groups share the same set of roles but possibly with different role mixture rates. Normal groups follow the same pattern with respect to their role mixture rates, but the anomalous group has a role mixture rate that deviates from the normal pattern.

- We identify the group membership and the role for each individual and define the group anomaly with respect to the role mixture rate. We develop a hierarchical Bayes model: the GLAD model, for detecting the group anomaly.

- In practice, we first identify the normal mixture rates. Then for each learned group, we evaluate the likelihood of its observations being generated with the normal mixture rates. The lower the likelihood value is, the more anomalous the group would be.
GLAD: Group Anomaly Detection in Social Media Analysis

- For the dynamic d-GLAD model, we emphasize on the temporal aspect of the data and detect the change of the role mixture rate within the groups.

- detect the groups whose mixture rates change drastically from the previous time stamps. Compared with GLAD, we not only need to decide whether a group is anomaly or not, but also need to specify when the group appears anomalous.

- Both GLAD and d-GLAD build upon the notion of role mixture rate, which essentially requires a precise inference of both the group membership and role identity for each individual in the group.
GLAD: Group Anomaly Detection in Social Media Analysis

- GLAD achieves the highest detection accuracy. It is also more robust over 10 random runs.

- d-GLAD achieves the lowest false positive rate
GLAD: Group Anomaly Detection in Social Media Analysis

- Real World Datasets

- Scientific Publications: We create a dataset from a pre-processed DBLP dataset. The dataset consists of conference papers from 20 conferences of four major areas: database (DB), data mining (DM), information retrieval (IR) and artificial intelligence (AI).

- We set up the group anomaly detection scenario as follows: we randomly sample groups of papers from KDD and treat them as normal groups. Then we sample groups of papers from the other conferences (e.g., CVPR, ICML, SIGMOD) and inject them into KDD papers as group anomalies. If the two papers have at least one common author, we add a link between them.
GLAD: Group Anomaly Detection in Social Media Analysis

Table 3: Group Anomaly Accuracy of GLAD and four baselines on DBLP publications. With KDD papers treated as normal groups and other conferences are treated as group anomalies respectively.

<table>
<thead>
<tr>
<th>Methods</th>
<th>GLAD</th>
<th>Graph-LDA</th>
<th>Graph-MGM</th>
<th>MMSB-LDA</th>
<th>MMSB-MGM</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBLP: KDD/CVPR</td>
<td>0.4167</td>
<td>0.3333</td>
<td>0.3333</td>
<td>0.2500</td>
<td>0.2500</td>
</tr>
<tr>
<td>DBLP: KDD/ICML</td>
<td>0.2500</td>
<td>0.0833</td>
<td>0.0833</td>
<td>0.1667</td>
<td>0.1667</td>
</tr>
<tr>
<td>DBLP: KDD/SIGMOD</td>
<td>0.2875</td>
<td>0.0750</td>
<td>0.0500</td>
<td>0.1625</td>
<td>0.1625</td>
</tr>
<tr>
<td>DBLP: KDD/CIKM</td>
<td>0.4500</td>
<td>0.4000</td>
<td>0.3625</td>
<td>0.2625</td>
<td>0.2625</td>
</tr>
<tr>
<td>DBLP: KDD/EDBT</td>
<td>0.2625</td>
<td>0.0500</td>
<td>0.0875</td>
<td>0.2000</td>
<td>0.2000</td>
</tr>
</tbody>
</table>
GLAD: Group Anomaly Detection in Social Media Analysis

- US Senate Voting

Figure 6: Common votes graph with party labels inferred by GLAD for 100 senators on the aggregated network. Compared with ground truth, two outliers are highlighted due to their anomalous voting behavior.