Anomaly Detection For Social Networks

Prof. Shou-de Lin

CSIE/GINM, NTU
Introduction on Anomaly Detection

- Anomalous events occur relatively infrequently.
- Also referred to as outliers, exceptions, peculiarities, surprise, etc.
- However, when they do occur, the consequences can be quite dramatic (often in a negative sense).
  - Cyber intrusions
  - Credit card fraud
  - Terrorists

“Mining needle in a haystack.”

2009/11/17  SNA09, Shou-de Lin
Outliers in 2-D Data

- $N_1$ and $N_2$ are regions of normal behaviours
- Points $o_1$ and $o_2$ are anomalies
- Points in region $O_3$ might also be anomalies
Challenges on Outlier Detection

• Defining a representative normal region is challenging
• The boundary between normal and outlying behaviour is often not precise
• The exact notion of an outlier is different for different application domains
• Availability of labelled data for training/validation
• Data might contain noise
• Normal behaviour keeps evolving
Sample Data

- Most common form of the data handled by anomaly detection techniques is **Record Data**
  - Univariate or Multivariate
  - Nature of attributes
    - Binary, Categorical, Continuous, Hybrid

```
<table>
<thead>
<tr>
<th>Tid</th>
<th>SrcIP</th>
<th>Start time</th>
<th>Dest IP</th>
<th>Dest Port</th>
<th>Number of bytes</th>
<th>Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>206.135.38.95</td>
<td>11:07:20</td>
<td>160.94.179.223</td>
<td>139</td>
<td>192</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>206.163.37.95</td>
<td>11:13:56</td>
<td>160.94.179.219</td>
<td>139</td>
<td>195</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>206.163.37.95</td>
<td>11:14:29</td>
<td>160.94.179.217</td>
<td>139</td>
<td>180</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>206.163.37.95</td>
<td>11:14:30</td>
<td>160.94.179.255</td>
<td>139</td>
<td>199</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>206.163.37.95</td>
<td>11:14:32</td>
<td>160.94.179.254</td>
<td>139</td>
<td>19</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>206.163.37.95</td>
<td>11:14:35</td>
<td>160.94.179.253</td>
<td>139</td>
<td>177</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>206.163.37.95</td>
<td>11:14:36</td>
<td>160.94.179.252</td>
<td>139</td>
<td>172</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>206.163.37.95</td>
<td>11:14:38</td>
<td>160.94.179.251</td>
<td>139</td>
<td>285</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>206.163.37.95</td>
<td>11:14:41</td>
<td>160.94.179.250</td>
<td>139</td>
<td>195</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>206.163.37.95</td>
<td>11:14:44</td>
<td>160.94.179.249</td>
<td>139</td>
<td>163</td>
<td>Yes</td>
</tr>
</tbody>
</table>
```
Input Data – *Complex Data Types*

- Relationship among data instances
  - Nothing (normal vector data)
  - Temporal (Sequential)
  - Spatial
  - **Graph** or Network (what we focus in SNA )
Anomaly Detection on Graph Data

• D. J. Cook and L. B. Holder, Graph-Based Data Mining, IEEE Intelligent Systems, 2000. (SUBDUE)
• Cabel C. Noble, Diane J. Cook. Graph-based Anomaly Detection. KDD 2005.
Overview

• Introduce to Graph-based Anomalies
• **Prerequisite: how to obtain frequent subgraphs?**
  – SUBDUE: A Graph Pattern Mining Algorithm
• KDD05 Method
  – Finding the anonymous substructure from a network
  – Comparing which network is more abnormal
    • SUBDUE-based approach
    • Conditional-Entropy based approach
• ICDM06 Method
What is SUBDUE

- An algorithm for detecting repetitive patterns (substructures) within graphs
• To find a graph pattern that best compresses the graph data

Consider this graph

Consider this pattern to compress the graph
SUBDUE (2/5)

• SUBDUE starts by finding unique vertices in the input graph
• Each of these substructures is extended by one edge in every possible way
• Substitute the original graph using the substructures.

Using **Minimum Description Length (MDL)** to evaluate the quality of the substitution:

\[ M(S, G) = DL(G | S) + DL(S) \]

- \(G\): the entire graph, \(S\): the substructure
- \(DL(G | S)\): the description length of \(G\) after compressing it using \(S\)
- \(DL(S)\): the description length of the substructure

\[ i.e., \]

2009/11/17 SNA09, Shou-de Lin
SUBDUE (3/5)

- Compress the graph and evaluate the quality

Size(original Graph) = Size(Substructure) + Size(Input Graph Compressed by Substructure)

\[
\begin{align*}
\#\text{vertex}=2, \#\text{edges}=1 & \quad \text{Size(Input Graph)} = \#\text{vertices} + \#\text{edges} = 3 \\
\#\text{vertex}=12, \#\text{edges}=12 & \quad \text{Size(Input Graph)} = \#\text{vertices} + \#\text{edges} = 24 \\
\#\text{vertex}=8, \#\text{edges}=8 & \quad \text{Size(Input Graph)} = \#\text{vertices} + \#\text{edges} = 16
\end{align*}
\]

\[
\therefore \text{Value} = \frac{24}{3+16} = 1.26
\]

2009/11/17  SNA09, Shou-de Lin
After evaluating the substructures, only the top k best substructures are retained. In this example, assume k=1

Value = \frac{24}{(3+16)} = 1.26

Value = \frac{24}{(3+20)} = 1.04
SUBDUE (5/5)

• The top substructure, which is retained, is again expanded by one edge in every possible way. And then these substructures are evaluated in a similar manner.

Value=$\frac{24}{14+5}=1.26$

Value=$\frac{24}{8+5}=1.84$

• Keep going on until the value stops increasing to find the best compressing substructure.
Overview

• Introduce to Graph-based Anomalies
• Prerequisite: how to obtain frequent subgraphs?
  – SUBDUE: A Graph Pattern Mining Algorithm

• KDD05 Method
  – Finding the anonymous substructure from a network
  – Comparing which network is more abnormal
    • SUBDUE-based approach
    • Conditional-Entropy based approach

• ICDM06 Method
Anomalous Substructure Detection

• The frequent patterns produce low values of the MDL quantity $M(S,G) = DL(G|S) + DL(S)$

• An anomaly can be thought of as the opposite of a frequent pattern (i.e., infrequently)
  – Just find substructures producing high $M(S,G)$

• However, very small (e.g., a single vertex) and very large ($S=G$) substructures will have relatively high $M$ values.
A Better Heuristic

• A better heuristic is defined

\[ F_2(S, G) = \text{Size}(S) \times \text{Frequency}(S, G) \]

  – \text{Size}(S): the number of vertices in S
  – \text{Frequency}(S, G): the frequency that S appears in G

• A substructure is \textit{anomalous} \textbf{if it produce low} \( F_2(S, G) \)

• How about the previous issues?
  – \textbf{Largest} ones will not be found since \text{Size}(S) is high
  – \textbf{Smallest} ones will only be considered anomalous if they do not appear very often
An Example

- The most anomalous substructure is D
  - Its F2 value = 1*1 = 1
- Several 2-vertex substructures have F2 value = 2*1 = 2
  - E.g., A → C, D → A
- The least anomalous substructure is the entire graph
  - Its F2 value = 9*1 = 9
Overview

• Introduce to Graph-based Anomalies
• Prerequisite: how to obtain frequent subgraphs?
  – SUBDUE: A Graph Pattern Mining Algorithm
• KDD05 Method
  – Finding the anonymous substructure from a network
  – Comparing which network is more abnormal
    • SUBDUE-based approach
    • Conditional-Entropy based approach
• ICDM06 Method
Anomaly Among Networks

• Given several networks, we want to learn which one is more abnormal.
• Idea: abnormal networks are **harder to compress**
• Recall SUBDUE
  – Run multiple iterations on a graph
  – After each iteration, the graph is compressed using the discovered substructure
  – The next iteration operates on the newly-compressed graph
• Basic idea
  – The frequent substructure will be discovered in the **first several iterations**, while later ones will become less and less valuable (i.e., less common)
  – Assuming SUBDUE **halts** once the graph contains no substructure with more than one instance
Finding Anomalous Networks

• Anomalous subgraphs tend to experience less compression than other subgraphs, since they contain few common patterns.

• Define the anomaly measure as

\[ A = 1 - \frac{1}{n} \sum_{i=1}^{n} (n - i + 1) * c_i \]

  – \( n \): the number of iterations
  – \( c_i \): the percentage of the graph that is compressed away on the \( i^{th} \) iteration

\[ \frac{DL_{i-1}(G) - DL_i(G)}{DL_0(G)} \]

  – \( DL_j(G) \): the description length of the subgraph after \( j \) iterations.

The higher \( A \) the more anomalous.
Analysis

\[ A = 1 - \frac{1}{n} \sum_{i=1}^{n} (n - i + 1) \cdot c_i \]

- All network begin with an A-value=1 (i.e., completely anomalous)
  - The values drop off as portions of the subgraphs are compressed away iterations after iterations
  - \( c_i \) varies from 0 to 1
    - 0: the subgraph was not changed on the \( i^{th} \) iteration
    - 1: the entire subgraph was compressed away
  - \( (n-i+1) \): varies from \( n \) to 1 as \( i \) increases, it causes \( A \) to drop off more sharply for compressions that occur early on
  - \( (1/n) \): guarantees the final value in \([0, 1]\)
An Example

• Which of these three networks are more abnormal?

• In the 1st iteration, \( \text{A} \rightarrow \text{B} \) is ranked the highest, and will be used to compress the entire graph.

• If this is the only iteration under consideration, the third network is regarded as the most anomalous since it will not be compressed as all.
Overview

• Introduce to Graph-based Anomalies
• Prerequisite: how to obtain frequent subgraphs?
  – SUBDUE: A Graph Pattern Mining Algorithm
• KDD05 Method
  – Finding the anonymous substructure from a network
  – Comparing which network is more abnormal
    • SUBDUE-based approach
    • Conditional-Entropy based approach
• ICDM06 Method
Anomaly as Predictability

• The more predictable, the less abnormal

• Here the Conditional Substructure Entropy are used to measure the abnormality of a graph
Conditional Entropy

- **Conditional Entropy** measure the amount of information needed to describe an event, given that some other event is known to have occurred
  - $X$: the set of possible events
  - $Y$: the set of **prior** events (one of which is known to have occurred)

$$H(X \mid Y) = - \sum_{y \in Y} \sum_{x \in X} P(y) \cdot P(x \mid y) \cdot \log(P(x \mid y))$$

- $P(y)$: the probability that event $y$ occurred
- $P(x \mid y)$: the probability of event $x$ given event $y$
Conditional Substructure Entropy

• Given an **arbitrary n-vertex substructure**, how many bits are needed to describe its **surroundings**?
  – Surrounding: the extensions to the substructure

• **Notations**
  – Y: contains all n-vertex substructures in G
  – X: contains all extensions of the substructures in Y
  – For a given substructure \( y \in Y \),
    \( P(y) \) is the number of instances of \( y \) in G, divided by the total number of instances of all n-vertex substructures
  – For a particular substructures \( x \in X, y \in Y \),
    \( P(x|y) \) is the percentage of instances of \( y \) that extend to an instance of \( x \)
Conditional Substructure Entropy

- Here we need to modify the $H(X|Y)$
  - Because being able to predict the “absent” of instances is as important as being able to predict the existence of them.
  - We want to measure the bits needed to describe \textbf{which events occurred and which ones did not occur}

\[
H(X \mid Y) = \sum_{y \in Y} \sum_{x \in X} P(y) \left( P(x \mid y) \log(P(x \mid y)) + ((1 - P(x \mid y)) \log(1 - P(x \mid y)) \right)
\]
An Example

- For the set $Y$, let the substructure size of $n=2$
  - Then $Y$ contains:
    - $A \rightarrow B \rightarrow C \rightarrow B \rightarrow C \rightarrow A$

- $X$ will contain all extensions of substructures in $Y$:
  - $A \rightarrow B \rightarrow C \rightarrow B \rightarrow C \rightarrow A$
  - $B \rightarrow C \rightarrow A \rightarrow C \rightarrow A \rightarrow B$

- Suppose that $y$ is $B \rightarrow C$ and that $x$ is $B \rightarrow C \rightarrow B$, then $P(x|y) = 1/2$

$$H(X|Y) = \sum_{y \in Y} \sum_{x \in X} P(y) \left\{ (P(x|y) \log(P(x|y)) + ((1 - P(x|y))) \log(1 - P(x|y))) \right\}$$

2009/11/17 SNA09, Shou-de Lin
Overview

• Introduce to Graph-based Anomalies

• Prerequisite: how to obtain frequent subgraphs?
  – SUBDUE: A Graph Pattern Mining Algorithm

• KDD05 Method
  – Finding the anonymous substructure from a network
  – Comparing which network is more abnormal
    • SUBDUE-based approach
    • Conditional-Entropy based approach

• ICDM06 Method
Graph-based Anomalies

• Definition
  – A graph structure $S'$ is **anomalous** if it is **not isomorphic** to the graph’s normative substructure $S$, but is **isomorphic to $S$ within $X\%$**
    – $X$: the percentage of vertices and edges that need to be changed in order for $S'$ to be isomorphic to $S$
      • $X$ is usually small

• Intuition
  – If a person or entity is attempting to commit fraud, they will do all they can to hide their illicit behavior
  – To the end, they convey their actions close to legitimate actions, but not identical
Types of Graph Anomaly

• **Insertion** category
  – A vertex exists that is unexpected
  – An edge exists that is unexpected

• **Modification** category
  – The label on a vertex is different than was expected
  – The label on an edge is different than was expected

• **Deletion** category
  – An expected vertex is absent
  – An expected edge between two vertices is absent
Detect Abnormal Instance of *Modification*

• Basic Idea
  – Use the Minimum Description Length to discover the best substructure in a graph
  – And then subsequently examines all of the instances of that substructure that look similar to that pattern
An Example

GBAD-MDL output:
D is the anomaly
The context of the anomaly are also presented
Detect Abnormal Instance of *Insertion*

• Basic idea
  – Also use the MDL to discover the best substructure
  – Instead of examining all instances for similarity
  – Examine all extensions to the normative patterns
  – And look for extensions with the lowest probability
An Example

- After one iteration, the best substructure is
- On the second iteration, this is compressed to a single vertex, extensions are evaluated
- The result is
Detect Abnormal Instance of *Deletion*

- Basic idea
  - Again use MDL to discover the best substructure
  - Examine all of the instances of parent substructures that are missing various edges and vertices
  - An anomalous value is associated with the parent instances to represent the cost of transformation
  - The instance with the *lowest* cost of transformation is considered the anomaly
An Example

Normative pattern:

Anomalous instance:
DHS Insight Project: Cargo Data

- Shipment data from PIERS (Port Import Export Reporting Service)
- Only North American imports (U.S., Puerto Rico, Canada)
- 65,535 records (shipments)
- Information categories:
  - General
  - Commodity codes
  - Countries and ports
  - U.S. company names and locations
  - Foreign shipper names and locations
  - Notification party names and locations
  - Shipping line, vessel and packaging
  - Container
  - Weight and shipment
  - Financial
Anomaly Detection in Cargo Data

- Marijuana seized at port on Florida [U.S. Customers Service 2000].
- Smuggler did not disclose some financial information, and ship traversed extra port.
- GBAD-P discovers the extra traversed port; GBAD-MPS discovers the missing financial information.