Query-Based Outlier Detection in Heterogeneous Information Networks

Jonathan Kuck\textsuperscript{1}, Honglei Zhuang\textsuperscript{1}, Xifeng Yan\textsuperscript{2}, Hasan Cam\textsuperscript{3}, Jiawei Han\textsuperscript{1}

\textsuperscript{1}University of Illinois at Urbana-Champaign
\textsuperscript{2}University of California at Santa Barbara
\textsuperscript{3}US Army Research Lab
• **Heterogeneous information networks** are networks composed of multi-typed, interconnected vertices and links

• **Outlier detection** aims to find vertices that deviate significantly from other vertices

• However, outliers in such a heterogeneous network could be defined in many different ways
  – E.g. Finding outlier in authors of this paper
• **Heterogeneous information networks** are networks composed of multi-typed, interconnected vertices and links.
• **Outlier detection** aims to find vertices that deviate significantly from other vertices.
• However, outliers in such a heterogeneous network could be defined in many different ways.
  – E.g. Finding outlier in authors of this paper:
  
  
  **Jonathan**
  
  **Honglei**
  
  **Xifeng**
  
  **Hasan**
  
  **Jiawei**
• **Heterogeneous information networks** are networks composed of multi-typed, interconnected vertices and links
• **Outlier detection** aims to find vertices that deviate significantly from other vertices
• However, outliers in such a heterogeneous network could be defined in many different ways
  – E.g. Finding outlier in authors of this paper

![Jonathan Honglei Xifeng Hasan Jiawei](image)

• As users have their own intuition about which kind of outliers they are interested in...

*Allow users to specify queries for outlier detection*
Research Challenges

• How do users *interact* with the system to specify their queries?

• How do we define a *general* outlierness measure for different queries?

• How to *efficiently* find outliers for different queries?
Outline

• Basic Concepts and Notations
• Outlier Query Language
• NetOut: Outlierness Measure
• Processing Outlier Queries
• Experimental Results
• Summary
Basic Concepts and Notations

• Heterogeneous Information Network
  – An information network with multiple types of vertices \( G = (V, E, \phi, T) \) where \( V \) is the set of vertices; \( E \) is the set of edges; \( T \) is the set of types, and \( \phi : V \mapsto T \) assigns each vertex a type.
Basic Concepts and Notations

• **Meta-Path**
  – An ordered sequence of vertex types, denoted as $P$
  – E.g. $P = (\text{Author, Paper, Venue})$

• **Meta-Path Instantiation**
  – Use $\pi_P (v_i, v_j)$ to denote paths between two vertices $v_i, v_j$
    with the types that align with the given meta-path $P$
  – E.g. $\pi_P (\text{Zoe, KDD})$ is shown below in red solid lines

![Network schema](image1)

![An instantiated network](image2)
Basic Concepts and Notations

• Neighborhood
  – Based on a meta-path $P$, defined as $N_P(v_i) = \{v_j \mid \pi_P(v_i, v_j) \neq \emptyset\}$
  – E.g. $N_P(\text{Zoe}) = \{\text{KDD}, \text{ICDE}\}$

• Neighbor Vector
  – Based on a meta-path $P$, the neighbor vector $\sigma_P(v_i)$ describes how many paths there are from $v_i$ to each of its neighbors
  – E.g. $\sigma_P(\text{Zoe}) = (\text{KDD}:3, \text{ICDE}:2)$
Formalization of Outlier Queries

• Given a heterogeneous information network
• A query can be formalized as

\[ Q = (S_c, S_r, P, w) \]

• where
  – \( S_c \subset V \) is a set of (same-typed) candidate vertices
    • Specifying from where outliers are chosen from
  – \( S_r \subset V \) is a set of (same-typed) reference vertices
    • Specifying a sample of normal vertices
    • Optional. By default it equals to the candidate set
  – \( P, w \) define a weighted set of meta-paths
    • Specifying how two vertices are compared
Outlier Query Language

• General Formulation

\[
\text{FIND OUTLIERS FROM } \text{author} \{ \text{“C. Faloutsos”} \}.\text{paper.author}
\]

\[
\text{COMPARE TO } \text{venue} \{ \text{“KDD”} \}.\text{paper.author}
\]

\[
\text{JUDGED BY } \text{author.}\text{paper.venue}
\]

\[
\text{TOP 10;}
\]
Outlier Query Language

• General Formulation

FIND OUTLIERS

FROM author{“C. Faloutsos”}.paper.author

Specifying the candidate set

\[ S_c = N_P (\text{C. Faloutsos}) \quad P = (\text{Author, Paper, Author}) \]

COMPARE TO venue {“KDD”}.paper.author

JUDGED BY author.paper.venue

TOP 10;
Outlier Query Language

• General Formulation

FIND OUTLIERS
FROM author{"C. Faloutsos"}.paper.author
COMPARE TO venue {"KDD"}.paper.author
JUDGED BY author.paper.venue
TOP 10;

\[ S_r = N_P(KDD) \quad P = (\text{Venue, Paper, Author}) \]
Outlier Query Language

• General Formulation

FIND OUTLIERS
FROM author{"C. Faloutsos"}.paper.author
COMPARE TO venue{"KDD"}.paper.author
JUDGED BY author.paper.venue

Specify the candidate set
Specify the reference set

Compare vertices by neighbor vector \( \sigma_P(v_i) \)
where \( P = (\text{Author, Paper, Venue}) \)

TOP 10;
Outlier Query Language

• General Formulation

FIND OUTLIERS

FROM author{"C. Faloutsos"}.paper.author

COMPARE TO venue {"KDD"}.paper.author

JUDGED BY author.paper.venue

TOP 10;

Only return top 10 outliers
Outlier Query Language (cont’)

• Allow complicated judging standards
  – E.g. multiple weighted neighbor vectors

\[
\text{JUDGED BY } \text{author.paper.venue: 0.6, author.paper.author: 0.4}
\]

• Allow operations on sets
  – E.g. select by conditions

\[
\text{FROM } \text{venue \{"KDD"\}.paper.author AS A}
\text{WHERE COUNT (A.paper) > 10}
\]
NetOut: A General Network-Based Outlierness Measure

- Given a meta-path $P$ (in **JUDGED BY** clause)
- Define the *connectivity* between two vertices
  \[
  \kappa(v_i, v_j) = \left| \pi_{(pp^{-1})}(v_i, v_j) \right|
  \]
  - The more paths between them, the more similar they are
- Define *relative connectivity*
  - Compare the connectivity to self-connectivity
    \[
    \tilde{\kappa}(v_i, v_j) = \frac{\left| \pi_{(pp^{-1})}(v_i, v_j) \right|}{\left| \pi_{(pp^{-1})}(v_i, v_i) \right|}
    \]
    - Use self-connectivity as a expected connectivity to measure whether two vertices are unexpectedly connected or unexpectedly disconnected
- **Outlier Measure: NetOut**
  \[
  \Omega(v_i) = \sum_{v_j \in S_r} \tilde{\kappa}(v_i, v_j)
  \]
Comparing NetOut to Other Measures

A toy example.

<table>
<thead>
<tr>
<th>Neighbor Vector</th>
<th>VLDB</th>
<th>KDD</th>
<th>STOC</th>
<th>SIGGRAPH</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Reference Author(s) × 100</em></td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Sarah</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Rob</td>
<td>0</td>
<td>1</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Lucy</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td><em>Joe</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td><em>Emma</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30</td>
</tr>
</tbody>
</table>

Outlier measure comparison. The lower value, the more likely to be an outlier.

<table>
<thead>
<tr>
<th></th>
<th>NetOut</th>
<th>PathSim[1]</th>
<th>CosSim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sarah</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Rob</td>
<td>6.24</td>
<td>9.97</td>
<td>12.43</td>
</tr>
<tr>
<td>Lucy</td>
<td>31.11</td>
<td>32.79</td>
<td>32.83</td>
</tr>
<tr>
<td>Joe</td>
<td>50.00</td>
<td>1.94</td>
<td>7.04</td>
</tr>
<tr>
<td>Emma</td>
<td>3.33</td>
<td>5.44</td>
<td>7.04</td>
</tr>
</tbody>
</table>

Joe is not necessarily an interesting outlier, as those papers might simply be noise.

Emma is obviously an outlier and should be assigned a lower value.

Comparison on Real Data Set

- Apply on network constructed from DBLP data set
- Find outliers among Christos Faloutsos’ coauthors, in terms of their publishing venues

<table>
<thead>
<tr>
<th>Method</th>
<th>$\Omega_{NetOut}$ Name</th>
<th>$\Omega$-value</th>
<th>$\Omega_{PathSim}$ Name</th>
<th>$\Omega$-value</th>
<th>$\Omega_{CosSim}$ Name</th>
<th>$\Omega$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Adam Wright</td>
<td>2.54</td>
<td>Wenyao Ho</td>
<td>1.07</td>
<td>John Chien-Han Tseng</td>
<td>0.0022</td>
</tr>
<tr>
<td>2</td>
<td>Philip Koopman</td>
<td>2.55</td>
<td>Fernanda Balem</td>
<td>1.12</td>
<td>Fernanda Balem</td>
<td>0.0038</td>
</tr>
<tr>
<td>3</td>
<td>Nicholas D. Sidiroopoulos</td>
<td>3.29</td>
<td>Rebecca B. Buchheit</td>
<td>1.31</td>
<td>Guoqiang Shan</td>
<td>0.0046</td>
</tr>
<tr>
<td>4</td>
<td>Katia P. Sycara</td>
<td>3.64</td>
<td>John Chien-Han Tseng</td>
<td>1.41</td>
<td>Wenyao Ho</td>
<td>0.0066</td>
</tr>
<tr>
<td>5</td>
<td>David S. Doermann</td>
<td>3.65</td>
<td>Chi-Dong Chen</td>
<td>1.47</td>
<td>Chi-Dong Chen</td>
<td>0.0077</td>
</tr>
</tbody>
</table>

authors with very few paper; uninteresting outliers
Query Processing

• The calculation of NetOut can be written as

\[
\Omega(v_i; Q) = \sum_{v_j \in S_r} \frac{K(v_i, v_j)}{K(v_i, v_i)} = \sum_{v_j \in S_r} \frac{\langle \sigma(v_i), \sigma(v_j) \rangle}{\langle \sigma(v_i), \sigma(v_i) \rangle} = \frac{1}{\|\sigma(v_i)\|_2^2} \left\langle \sigma(v_i), \sum_{v_j \in S_r} \sigma(v_j) \right\rangle
\]
Query Processing

• The calculation of NetOut can be written as

\[
\Omega(v_i; Q) = \sum_{v_j \in S_r} \frac{K(v_i, v_j)}{K(v_i, v_i)}
\]

\[= \sum_{v_j \in S_r} \frac{\langle \sigma(v_i), \sigma(v_j) \rangle}{\langle \sigma(v_i), \sigma(v_i) \rangle}
\]

\[= \frac{1}{\|\sigma(v_i)\|^2} \left\langle \sigma(v_i), \sum_{v_j \in S_r} \sigma(v_j) \right\rangle
\]

\[O(|S_c|), \text{ only calculated once}
\]

• Time complexity: \(O(|S_c| + |S_r|)\)
Pre-Materialization of Meta-Path

• We observe...
  – Calculation of outlierness only has time complexity of $O(|S_c| + |S_r|)$
  – Retrieving $\sigma(v_i)$ for each vertex is much more time consuming

• Pre-materialization
  – Pre-store the materialization of all the length-2 meta-paths
Selective Pre-Materialization

• Storing materialization of all length-2 meta-paths can be space-consuming

• Selective Pre-materialization
  – From a given set of training queries, take vertices that frequently appear in the candidate sets (e.g. those appear in more than 10% of queries)
  – Pre-materialize all the length-2 meta-paths for these frequently appearing vertices
Experimental Setup

• **Data Set**
  – DBLP Data Set
    • 2,244,018 publications, 1,274,360 authors
    • Construct the network according to the schema above
    • Publishing venue information is included
    • Terms are extracted from publication titles

• **Query Sets**
  – Design three different types of queries to find different kinds of outliers
  – Generate 10,000 random queries for each type of queries as a query set
Case Study

• With different queries, the outlier measure is able to capture different outliers accordingly
  – E.g. Finding outliers in Christos Faloutsos’ coauthors

Outlier results by comparing publishing venues

\[ S_c = S_r = \text{author(“Christos Faloutsos”).paper.author} \]
\[ P = \text{author.paper.venue} \]

<table>
<thead>
<tr>
<th>Ranking</th>
<th>Name</th>
<th>(\Omega)-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Adam Wright</td>
<td>2.54</td>
</tr>
<tr>
<td>2</td>
<td>Philip Koopman</td>
<td>2.55</td>
</tr>
<tr>
<td>3</td>
<td>Nicholas D. Sidiropoulos</td>
<td>3.29</td>
</tr>
<tr>
<td>4</td>
<td>Katia P. Sycara</td>
<td>3.64</td>
</tr>
<tr>
<td>5</td>
<td>David S. Doerrmann</td>
<td>3.65</td>
</tr>
<tr>
<td>6</td>
<td>Asim Smailagic</td>
<td>3.69</td>
</tr>
<tr>
<td>7</td>
<td>John Chien-Han Tseng</td>
<td>4.00</td>
</tr>
<tr>
<td>8</td>
<td>Daniel P. Siewiorek</td>
<td>4.22</td>
</tr>
<tr>
<td>9</td>
<td>Jessica K. Hodgins</td>
<td>4.52</td>
</tr>
<tr>
<td>10</td>
<td>Dimitris N. Metaxas</td>
<td>4.57</td>
</tr>
</tbody>
</table>

Outlier results by comparing coauthor communities

\[ S_c = S_r = \text{author(“Christos Faloutsos”).paper.author} \]
\[ P = \text{author.paper.author} \]

<table>
<thead>
<tr>
<th>Ranking</th>
<th>Name</th>
<th>(\Omega)-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dimitris N. Metaxas</td>
<td>1.06</td>
</tr>
<tr>
<td>2</td>
<td>Bin Zhang</td>
<td>1.06</td>
</tr>
<tr>
<td>3</td>
<td>Hui Zhang</td>
<td>1.07</td>
</tr>
<tr>
<td>4</td>
<td>Lionel M. Ni</td>
<td>1.07</td>
</tr>
<tr>
<td>5</td>
<td>Bin Liu</td>
<td>1.08</td>
</tr>
<tr>
<td>6</td>
<td>Joel H. Saltz</td>
<td>1.08</td>
</tr>
<tr>
<td>7</td>
<td>Yang Wang</td>
<td>1.08</td>
</tr>
<tr>
<td>8</td>
<td>Hao Wang</td>
<td>1.08</td>
</tr>
<tr>
<td>9</td>
<td>Ee-Peng Lim</td>
<td>1.12</td>
</tr>
<tr>
<td>10</td>
<td>Katia P. Sycara</td>
<td>1.13</td>
</tr>
</tbody>
</table>
Efficiency Study

• Comparing strategies
  – **Baseline**: No pre-materialization
  – **Pre-Materialization (PM)**: All length-2 meta-path instantiations are pre-computed and indexed
  – **Selective Pre-Materialization (SPM)**: Only a subset of instantiations with relative frequency larger than 0.01 are indexed
Parameter Studies for SPM

- Different thresholds for selective pre-materialization
Summary

• Propose a framework for query-based outlier detection in heterogeneous information networks
• Formalize the definition of a query and provide a query language for users to interact with the system
• Design a general outlierness measure NetOut which can effectively find interesting outliers
• Present an efficient implementation to process such queries
Thank you
26-03-2015

• Introduction
• Outlier Query Language
• NetOut Measure
• Query Processing
• Experimental Results