Heterogeneous Subgraph Features for Information Networks

Andreas Spitz, Diego Costa, Kai Chen, Jan Greulich, Johanna Geiß, Stefan Wiesberg, and Michael Gertz

June 10, 2018 — GRADES-NDA, Houston, Texas, USA

Heidelberg University, Germany
Database Systems Research Group
Many information networks are heterogeneous

- Scientific publication networks
- Knowledge bases
- Metabolic networks
- ...
Many information networks are heterogeneous

- Scientific publication networks
- Knowledge bases
- Metabolic networks
- ...

How do you learn in heterogeneous networks?
Many information networks are heterogeneous

- Scientific publication networks
- Knowledge bases
- Metabolic networks
- ...

How do you learn in heterogeneous networks?

- With features, of course
Many information networks are heterogeneous

- Scientific publication networks
- Knowledge bases
- Metabolic networks
- ...

How do you learn in heterogeneous networks?

- With features, of course
- But how do you get the features?
Problems of Established Feature Extraction Approaches

**Classic features:**

- Require domain knowledge
- Are time-consuming to engineer
- Require metadata that may not be available

Neural node embeddings:

- Sample neighbourhoods through random walks
- Require extensive parameter tuning

Alternative idea:

use labeled subgraph counts as features
Problems of Established Feature Extraction Approaches

Classic features:

- Require domain knowledge
- Are time-consuming to engineer
- Require metadata that may not be available

Neural node embeddings:

- Sample neighbourhoods through random walks
- Require extensive parameter tuning
Problems of Established Feature Extraction Approaches

Classic features:

- Require domain knowledge
- Are time-consuming to engineer
- Require metadata that may not be available

Neural node embeddings:

- Sample neighbourhoods through random walks
- Require extensive parameter tuning

**Alternative idea:** use labeled subgraph counts as features
Heterogeneous Subgraph Features
Labeled subgraphs around a node:

- Encode neighbourhood information
- Are extremely diverse in heterogeneous networks
Motivation: Heterogeneous Subgraph Features

Labeled subgraphs around a node:
- Encode neighbourhood information
- Are extremely diverse in heterogeneous networks

Conjecture:
The subgraph neighbourhood of a node is representative of its function and label.
Motivation: Heterogeneous Subgraph Features

Labeled subgraphs around a node:

- Encode neighbourhood information
- Are extremely diverse in heterogeneous networks

Conjecture:
The subgraph neighbourhood of a node is representative of its function and label.
Motivation: Heterogeneous Subgraph Features

Labeled subgraphs around a node:
- Encode neighbourhood information
- Are extremely diverse in heterogeneous networks

Conjecture:
The subgraph neighbourhood of a node is representative of its function and label.

\[ \text{count (S , )} = 1 \]
Motivation: Heterogeneous Subgraph Features

Labeled subgraphs around a node:
- Encode neighbourhood information
- Are extremely diverse in heterogeneous networks

**Conjecture:**
The subgraph neighbourhood of a node is representative of its function and label.

$$\text{count (S, } \begin{array}{c} \text{node 1} \\ \text{node 2} \end{array} \text{) } = 1$$

$$\text{count (S, } \begin{array}{c} \text{node 3} \\ \text{node 4} \end{array} \text{) } = 2$$
**Problem:** depending on the iteration order, the nodes of structurally identical subgraphs may be visited in different order.
Core approach:

- Explore the local neighbourhood around each node
- Represent subgraphs by their characteristic string
- Count subgraphs by hashing the characteristic string
- Use the counts of subgraphs as node features
Heterogeneous Subgraph Encoding

Core approach:
- Explore the local neighbourhood around each node
- Represent subgraphs by their characteristic string
- Count subgraphs by hashing the characteristic string
- Use the counts of subgraphs as node features

Characteristic string construction:
- Encode each node as a block
- Blocks start with the node label
- Subsequent entries denote neighbours of all given labels
- Blocks are sorted lexicographically
Heterogeneous degree sequences:

- Are a **pseudo**-canonical encoding
- May result in colliding encodings

\[
z210y101y100y100x021x011
\]
Encoding Collisions

Heterogeneous degree sequences:

- Are a **pseudo**-canonical encoding
- May result in colliding encodings

Encoding collisions:

- Can only be enumerated (no closed formula)
- Depend on the network structure and the labels
- Have negligible frequency in practice

\[
\text{z210y101y100x021x011}
\]
Heuristic for Hub Mitigation

Real-world networks have:

- Skewed degree distributions
- Highly connected nodes (hubs)
Heuristic for Hub Mitigation

Real-world networks have:

▶ Skewed degree distributions
▶ Highly connected nodes (hubs)

Due to hubs:

▶ Feature extraction time is strongly increased
▶ Random walks retrieve non-local information
Real-world networks have:

- Skewed degree distributions
- Highly connected nodes (hubs)

Due to hubs:

- Feature extraction time is strongly increased
- Random walks retrieve non-local information

**Intuition:** Do not explore beyond nodes with degree $> d_{max}$. 
Evaluation: Label Prediction
Label Prediction: Task Definition

Given:
- Heterogeneous network
- Some nodes with missing labels

Predict:
- Missing node labels

Formal approach:
- Model as a classification task using logistic regression
- Evaluate with $F_1$-score
Label Prediction: Task Definition

Given:
- Heterogeneous network
- Some nodes with missing labels

Predict:
- Missing node labels

Formal approach:
- Model as a classification task using logistic regression
- Evaluate with $F_1$-score
Label Prediction: Data Sets

Movie network (IMDB):
- Star-shaped structure around movies
- Low edge density

Scientific publication network (MAG):
- Intermediate structure
- Papers form the core component

Entity cooccurrence network (LOAD):
- Cooccurrences of named entities in text
- Strongly connected structure
- High edge density
Feature Engineering and Extraction

Subgraph features:

- Maximum number of edges: 5
- No exploration beyond 10% of highest degree nodes
- Masked starting node label

Embedded features:

- DeepWalk
- LINE
- node2vec
## Extraction Runtime Estimation (seconds per node)

<table>
<thead>
<tr>
<th></th>
<th>subgraph features</th>
<th>node2vec</th>
<th>DeepWalk</th>
<th>LINE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>75%</td>
<td>90%</td>
<td>95%</td>
</tr>
<tr>
<td>LOAD</td>
<td>32.1</td>
<td>19.6</td>
<td>29.7</td>
<td>53.0</td>
</tr>
<tr>
<td>IMDB</td>
<td>2.6</td>
<td>1.7</td>
<td>3.0</td>
<td>6.7</td>
</tr>
<tr>
<td>MAG</td>
<td>25.2</td>
<td>10.4</td>
<td>11.0</td>
<td>19.5</td>
</tr>
</tbody>
</table>

Percentages denote nodes for which the extraction finished in at most the shown time.
Evaluation Results (Training Size)

- Subgraph
- node2vec
- DeepWalk
- LINE

F1 score for MAG, LOAD, and IMDB datasets.
Evaluation Results (Missing Labels)

The diagrams show the F1 score for different models (Subgraph, node2vec, DeepWalk, LINE) across different subgraph percentages (0%, 15%, 30%, 45%, 60%, 75%) for three datasets: MAG, LOAD, and IMDB.
Evaluation: Institution Ranking
Institution Ranking: Task Definition

Given:

- Scientific publication network
- A range of years
- A set of conferences

Predict ranking of institutions:

- For upcoming conferences
- By accepted papers
- For the next conference

*KDDCup 2016.* [https://kddcup2016.azurewebsites.net](https://kddcup2016.azurewebsites.net)
Institution Ranking: Task Definition

Given:
- Scientific publication network
- A range of years
- A set of conferences

Predict ranking of institutions:
- For upcoming conferences
- By accepted papers
- For the next conference

Formal approach:
- Model as a regression task for the institution relevance score
- Evaluate with normalized discounted cumulative gain (NDCG20)

*KDDCup 2016.* [https://kddcup2016.azurewebsites.net](https://kddcup2016.azurewebsites.net)
Institution Ranking: Data Set

Subset of the Microsoft Academic Graph:

- Institutions $I$
- Authors $A$
- Papers $P$
- Publication data from 2011 - 2016

Data preparation:

- Focus on 5 conferences
  KDD, FSE, ICML, MM, MOBICOM
- Use citations to a depth of 3
Feature Types and Extraction

Classic features (manually engineered):

▶ Previous relevance scores, publication counts, etc. (8)
▶ Linguistic features (32)

Subgraph features:

▶ Maximum number of edges: 5
▶ No maximum degree exploration limit

Embedded features:

▶ DeepWalk
▶ LINE
▶ node2vec
NDCG Scores for Institution Ranking

- Random Forest
- Linear Regression
- Decision Tree
- Bayesian Ridge
- Classic
- Subgraphs
- Combined
- node2vec
- DeepWalk
- LINE

Graphs showing NDCG scores for various methods across different conferences:

- KDD
- FSE
- ICML
- MM
- MOBICOM
## Average NDCG Scores for Institution Ranking

<table>
<thead>
<tr>
<th></th>
<th>LinRegr</th>
<th>DecTree</th>
<th>RanForest</th>
<th>BayRidge</th>
</tr>
</thead>
<tbody>
<tr>
<td>classic</td>
<td>0.65</td>
<td>0.58</td>
<td>0.64</td>
<td>0.51</td>
</tr>
<tr>
<td>subgraph</td>
<td>0.58</td>
<td>0.51</td>
<td><strong>0.68</strong></td>
<td>0.65</td>
</tr>
<tr>
<td>combined</td>
<td>0.62</td>
<td>0.46</td>
<td><strong>0.68</strong></td>
<td>0.60</td>
</tr>
<tr>
<td>node2vec</td>
<td>0.18</td>
<td>0.19</td>
<td>0.39</td>
<td>0.27</td>
</tr>
<tr>
<td>DeepWalk</td>
<td>0.14</td>
<td>0.17</td>
<td>0.25</td>
<td>0.18</td>
</tr>
<tr>
<td>LINE</td>
<td>0.17</td>
<td>0.23</td>
<td>0.56</td>
<td>0.23</td>
</tr>
</tbody>
</table>
Feature Importance Analysis (Random Forest)

KDD

ICML

FSE

MM

MobiCom

0.178

0.067

0.089

0.048

0.137

0.032

0.064

0.063

0.048

0.046
Summary & Resources
Summary

Heterogeneous subgraph features:

- Extracted by local exploration and enumeration
- Avoid isomorphism test by encoding degree sequences
Summary

Heterogeneous subgraph features:

- Extracted by local exploration and enumeration
- Avoid isomorphism test by encoding degree sequences

In comparison to classic features:

- Similar performance
- Require no domain knowledge for extraction
- No engineering process necessary
Summary

Heterogeneous subgraph features:

- Extracted by local exploration and enumeration
- Avoid isomorphism test by encoding degree sequences

In comparison to classic features:

- Similar performance
- Require no domain knowledge for extraction
- No engineering process necessary

In comparison to embedded features:

- Better predictive performance
- Longer extraction time
The implementation is available online:

- C++ (core extraction routines)
- Python (wrapper)

https://dbs.ifi.uni-heidelberg.de/resources/hsgf/
The implementation is available online:

- C++ (core extraction routines)
- Python (wrapper)

https://dbs.ifi.uni-heidelberg.de/resources/hsgf/