Sampling and Summarization for Social Networks

PAKDD 2013 Tutorial
Shou-De Lin*, Mi-Yen Yeh#, and Cheng-Te Li*

* Computer Science and Information Engineering, National Taiwan University
  # Institute of Information Science, Academic Sinica

sdlin@csie.ntu.edu.tw, miyen@iis.sinica.edu.tw, d98944005@csie.ntu.edu.tw

Tutorial slides can be downloaded here: http://mslab.csie.ntu.edu.tw/tut-pakdd13/
About This Tutorial

• It is a two-hour tutorial for PAKDD2013 on social network sampling and summarization
  – We do not anticipate to cover *everything relevant* to this topic.
  – We will highlight the trend, categorize different types of strategies, and describe some ongoing works of us

• Agenda
  – Introduction + Sampling + Q/A (45+10 min)
  – Summarization + conclusion + Q/A (45+10 min)
Big Social Network ➞ Billions of different types of nodes and links

What can be mined from this picture?
Motivation

- Sometimes the full networks are not completely observed in advance
- Even they are, loading everything into memory for further analysis might not be feasible
- Even it is feasible, generating some simple statistics (e.g. average path length, diameter) can take a long time, not to mention more complicated ones (e.g. counting the occurrence of certain pattern)
An Example on Facebook

- 1+ Billion users
- Avg: 130 friends each node

It costs >1TB memory to simply save the raw graph data (without attributes, labels nor content)

This can cause problems for information extraction, processing, and analysis

Two possible solutions: **Sampling and Summarization**
Sampling Versus Summarization

• **Sampling**
  – Assume the information of nodes/links become known only after they are sampled
  – Require certain sampling strategy to explore/expand the network gradually
  – Goal: gradually identify a small set of representative nodes and links of a social network, usually given little prior information about this network

• **Summarization**
  – The entire social network is known in prior
  – Goal: condense the social network as much as possible without losing too much information
Homogeneous VS Heterogeneous Social Networks

• **Homogeneous** → **Single Relational Network**
  – *Single* object type & Link type

• **Heterogeneous** → **Multi-Relational Network**
  – *Multiple* object type & Link type

• Example
  – Homogeneous
  – Heterogeneous

Lin et al., Sampling and Summarization for Social Networks, PAKDD 2013 tutorial
Sampling for Social Networks
Sampling Social Networks

• Assume that the detailed information of a node can only be seen after it is sampled
  – Entire social network is not known in advance

• Goal
  – Sample (i.e. gradually observe nodes and links) a sub-network that represents the whole network
    • To preserve certain properties of the original network

Lin et al., Sampling and Summarization for Social Networks, PAKDD 2013 tutorial
Evaluating the Sampling Quality

• How to measure the quality of the sampling algorithm?

• A sampling algorithm is effective if
  – The sampled social network can preserve certain network properties
  – Using the sampled network to perform an ultimate task (e.g. centrality analysis, link prediction, etc), one can produce similar results as if this task were performed on the fully observed network
  – The sample sub-network is small
Properties Preserved (1/3)

• **Homogeneous Static Social Network**
  – In/Out Degree Distribution
  – Path Length Distribution
  – Clustering Coefficient Distribution
  – Eigenvalues
  – Weakly/Strongly Connected Component Size Distribution
  – Community Structure
  – Etc..
Properties Preserved (2/3)

• Homogeneous **Dynamic** Social Networks
  (Graphs are time-evolving)
  
  – Densification Power Law
    • Number of edges vs. number of nodes over time
  
  – Shrinking diameter
    • Observed that shrinks and stabilizes over time
  
  – Average clustering coefficient over time
  
  – Largest singular value of graph adjacency matrix over time
  
  – Etc...
Properties Preserved (3/3)

- Heterogeneous Social Network
  - Note type Distribution
  - Intra-link and Inter-link type Distribution
  - Higher-order types connection
Evaluation Metrics

• Whether certain properties are preserved
  – For single value properties (E.g. clustering coefficient, average path length), one can measure whether this value is preserved
  – For distributional properties (E.g. degree distribution, component size distribution), one can compute the distance between two distributions (e.g. KL divergence)

• Whether certain end-task can be performed similarly
  – Performing a certain task using the sampled network, and check whether the results are similar to those when the full network is used
Sampling for **Homogeneous** Social Networks
Three Main Strategies

• Node Selection
• Edge Selection
• Sampling by Exploration
  – Random Walk
  – Graph Search
  – Chain-Referral Sampling

Seeds (i.e., ego)

Lin et al., Sampling and Summarization for Social Networks, PAKDD 2013 tutorial
Node Selection

• **Random Node Sampling**
  – Uniformly select a set of nodes

• **Degree-based Sampling** [Adamic’01]
  – the probability of a node being selected is proportional to its degree (assuming known)

• **PageRank-based Sampling** [Leskovec’06]
  – the probability of a node being selected is proportional to its PageRank value (assuming known)
Edge Selection

• Random Edge (RE) Sampling
  – Uniformly select edges at random, and then include the associated nodes

• Random Node-Edge (RNE) Sampling
  – Uniformly select a node, then uniformly select an edge incident to it

• Hybrid Sampling [Leskovec’06]
  – With probability $p$ perform RE sampling, with probability $1-p$ perform RNE sampling
Edge Selection (cont.)

• **Induced Edge Sampling** [Ahmed’12]
  – Step 1: Uniformly select edges (and consequently nodes) for several rounds
  – Step 2: Add edges that exist between sampled nodes

• **Frontier Sampling** [Ribeiro’10]
  – Step 0: Randomly select a set of nodes L as seeds
  – Step 1: Select a seed u from L using degree-based sampling
  – Step 2: Select an edge of u, (u, v), uniformly
  – Step 3: Replace u by v in L and add (u, v) to the sequence of sampled edges
  – * Repeat Step 1 to 3
Sampling by Exploration

• **Random Walk** [Gjoka’10]
  – The next-hop node is chosen uniformly among the neighbors of the current node

• **Random Walk with Restart** [Leskovec’06]
  – Uniformly select a random node and perform a random walk with restarts

• **Random Jump** [Ribeiro’10]
  – Same as random walk but with a probability $p$ we jump to any node in the network

• **Forest Fire** [Leskovec’06]
  – Choose a node $u$ uniformly
  – Generate a random number $z$ and select $z$ out links of $u$ that are not yet visited
  – Apply this step recursively for all newly added nodes
Sampling by Exploration (cont.)

- Ego-Centric Exploration (ECE) Sampling
  - Similar to random walk, but each neighbor has probability to be selected
  - Multiple ECE (starting with multiple seeds)

- Depth-First / Breadth-First Search [Krishnamurthy’05]
  - Keep visiting neighbors of earliest / most recently visited nodes

- Sample Edge Count [Maiya’11]
  - Move to neighbor with the highest degree, and keep going

- Expansion Sampling [Maiya’11]
  - Construct a sample with the maximal expansion. Select the neighbor v based on $\arg \max_{v \in N(S)} |N(\{v\}) - (N(S) \cup S)|$

  S: the set of sampled nodes, N(S): the 1st neighbor set of S
Example: Expansion Sampling

|N({A})| = 4

|N({E}) – N({A}) U {A}| = |{F, G, H}| = 3

|N({D}) – N({A}) U {A}| = |{F}| = 1
Drawback of Random Walk: Degree Bias!

- Real average node degree $\sim 94$, Sampled average node degree $\sim 338$
- Solution: modify the transition probability :

$$P_{v,w} = \begin{cases} 
\frac{1}{k_v} \cdot \min(1, \frac{k_v}{k_w}) & \text{if } w \text{ is a neighbor of } v \\
1 - \sum_{y \neq v} P_{v,y} & \text{if } w = v \\
0 & \text{otherwise}
\end{cases}$$
Metropolis Graph Sampling [Hubler’08]

• Step 1: Initially pick one subgraph sample \( S \) with \( n' \) nodes randomly

• Step 2: Iterate the following steps until convergence

  2.1: Remove one node from \( S \)

  2.2: Randomly add a new node to \( S \rightarrow S' \)

  2.3: Compute the likelihood ratio

\[
a = \frac{\rho^*(S')}{\rho^*(S)}
\]

  - if \( a \geq 1 \): accept transition: \( S := S' \)
  - if \( a < 1 \): accept transition: \( S := S' \) with probability \( a \)
  - reject transition: \( S := S' \) with probability \( 1 - a \)

  \( \rho^*(S) \) measures the similarity of a certain property between the sample \( S \) and the original network \( G \)

• Be derived approximately using Simulated Annealing
Sampling for Heterogeneous Social Networks
Sampling on Heterogeneous Social Networks

• Heterogeneous Social Networks (HSN)
  – A graph $G=\langle V, E \rangle$ has $n$ nodes $(v_1, v_2, ..., v_n)$, $m$ directed edges $(e_1, ..., e_m)$ and $k$ different types
  – Each node/edge belongs to a type
    • Given a finite set $L = \{L_1, ..., L_k\}$ denoting $k$ types

• Sampling methods for HSN
  – Multi-graph sampling [Gjoka’10]
  – Type-distribution preserving sampling (Li’11)
  – Relational-profile preserving sampling (Yang’13)
Multigraph Sampling

- Random walk sampling on the union multiple graph to avoid stopping on the disconnected graph.
Sampling Heterogeneous Social Networks

• Sampling methods for HSN
  – Multi-graph sampling [Gjoka’10]
  – Type-distribution preserving sampling (Li’11)
  – Relational-profile preserving sampling (Yang’13)
Node Type Distribution Preserving Sampling

• Given a graph $G$ and a sampled subgraph $G_S$
• The node type distribution of $G_S$ is expected to be the same as $G$, i.e., $d(Dist(G_S), Dist(G)) = 0$
  – $d()$ denotes the difference between two distributions

Original Network

Sampled Network

$(9:6) = (3:2)$
Connection-type Preserving Sampling

• Heterogeneous Connection
  – For an edge $E[v_i,v_j]$
  – **Intra**-connection edge: $\text{Type}(v_i) = \text{Type}(v_j)$
  – **Inter**-connection edge: $\text{Type}(v_i) \neq \text{Type}(v_j)$

• **Intra**-Relationship preserving
  – The ratio of the intra-connection should be preserved, that is:
    
    $$d(\text{IR}(G_S),\text{IR}(G)) = 0$$
  
  – If the intra-relationship is preserved, the **inter**-relationship is also preserved

![Diagram showing original and sampled networks with ratio 6:4 = 3:2]
Respondent-driven Sampling

• First proposed in social science [Heck’99] to solve the hidden population in surveying.
• Two Main Phases:
  Snowball sampling ➔ Finding steady-state in Recruitment matrix

![Diagram showing respondents, limited coupon, and transition matrix with steady-state vector.]

Transition Matrix

\[
\begin{array}{ccc}
S_{11} & S_{12} & S_{13} \\
S_{21} & S_{22} & S_{23} \\
S_{31} & S_{32} & S_{33} \\
\end{array}
\]

N-step transition

steady-state vector

\[
\begin{array}{ccc}
P_1 & P_2 & P_3 \\
\end{array}
\]
Comparing Different Sampling algorithms

- Respondent-driven Sampling does a good job with small node size, but saturate to mediocre afterwards
- Random node sampling performs poorly in the beginning, but reaches the best results after sufficient amount of nodes are sampled.
Heterogeneous Social Networks

• Sampling methods for HSN
  – Multi-graph sampling [Gjoka’10]
  – Type-distribution preserving sampling (Li’11)
  – Relational-profile preserving sampling (Yang’13)
Relational Profile Preserving Sampling

- Node-type/intra-type preservation considers the **semantics** of nodes, but not the **structure** of networks
- Propose the **Relational Profile** to consider semantic and structure all together
  - Capture the dependency between each **Node Type (NT)** and **Edge Type (ET)** of a **directed** Heterogeneous Network
  - Consists of 4 Relational Matrices
    - **Conditional probabilities** $P(T_j|T_i)$ (e.g. $P(\text{LT}=\text{cites}|\text{NT}=\text{paper})$)
    - Node to node, node to edge, edge to node, edge to edge
Example of Relational Profile (RP)

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>A</th>
<th>C</th>
<th>J</th>
<th>c</th>
<th>p</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.44</td>
<td>0.22</td>
<td>0.22</td>
<td>0.11</td>
<td>0.44</td>
<td>0.33</td>
<td>0.22</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>0.5</td>
<td>0.33</td>
<td>0.17</td>
<td></td>
<td>0.66</td>
<td>0.33</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>0.5</td>
<td>0.5</td>
<td></td>
<td></td>
<td>0.6</td>
<td>0.4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>A</th>
<th>C</th>
<th>J</th>
<th>c</th>
<th>p</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.182</td>
<td>0.364</td>
<td>0.091</td>
<td>0.273</td>
<td>0.182</td>
<td>0.364</td>
<td>0.364</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>p</td>
<td>0.5</td>
<td>0.125</td>
<td>0.375</td>
<td></td>
<td>0.17</td>
<td>0.5</td>
<td>0.33</td>
</tr>
<tr>
<td>a</td>
<td>0.5</td>
<td>0.5</td>
<td></td>
<td></td>
<td>0.22</td>
<td>0.33</td>
<td>0.44</td>
</tr>
</tbody>
</table>

Lin et al., Sampling and Summarization for Social Networks, PAKDD 2013 tutorial

13/05/02
Challenge: How to approximate RP when the true RP is unknown

• We propose Exploration by Expectation Sampling
• Aim to preserve the unknown relational profile while adding new sample node
  1. Randomly choose a starting node and the corresponding edges
  2. Based on current RP, select a next node from all 1 degree neighbor
  3. Add the new node and all its edges
  4. Update RP of the sub-sampled graph
  5. Repeat step 3, 4 & 5 until the converge of RP

• Which node should be selected?
  – Select the node whose inclusion can potentially lead to the largest change to the existing RP
    • Use the partially observed RP to generate the ‘expected amount of change’ of each node as its score
    • Weighted sampling based on the score
Relational Profile Sampling (RPS)

Idea: Sample to increase the diversity

\[ D(v, G_s) = \text{estimated change of RP given sampling } v \text{ on the current graph } G_s \]
\[ = E[\Delta_p(G_s, G_s+v) | G_s], \text{ where } \Delta_p = \text{RMSE}_{RP} \]
which can be calculated as

\[ \sum_{t \in NT} P(type(v) = t|G_s) \Delta RP(G_s, G_s + v) \]

Exploiting the existing RP, \( P(\text{type}(v) = t | G_s) \) can be obtained using the observed types of \( v \)'s neighbors

\[ \prod_{i \in N(v)} \left( \frac{RP(type(i)|type(v) = t)P(type(v) = t)}{Z} \right) \]

\( P(\text{type} | \text{type}) \) can be obtained from the existing RP

Goal: maximize expected property (Relational Profile distribution) change
Evaluation

• Datasets: 3 real-life large scale social networks
• Baselines:
  – Random Walk Sampling (RW)
  – Degree-based sampling (HDS)
• Evaluation I (Property Preservation): see how well the sampled network approximates two properties of the full network
• Evaluation II (Prediction): training a prediction model using the sampled network to infer out-of-sampled network status:
  – **Node Type Prediction**: Predict the type of unseen nodes in the network using a sub-sampled network
  – **Missing Relations Prediction**: Recover/predict the missing links
• Features:
  • $f_{\text{deg}} = (\text{in/out deg; avg in/out deg of neighbors})$
  • $f_{\text{topo}} = (\text{Common Neighbors; Jaccard’s Coefficient; etc})$
  • $f_{\text{nt}} = P(\text{type}(v)|G_S)$
  • $f_{\text{RPnode}} = \frac{\text{#type}(v)t\forall v \in N(n)}{|N(n)|}$
  • $f_{\text{RPpath}} = \prod_{i \in N(n)} \frac{1}{Z} RP(\text{type}(i)|\text{type}(v) = t)P(\text{type}(v) = t) \sum_{p \in \text{Path}(s,t)} \prod_{(p_1,p_2) \in p} P(\text{type}(p_2)|\text{type}(p_1))$
Experiments (Property Preservation)

- RP (RMSE)

- Weighted PageRank

Preserving relative node weights propagated throughout entire network
Experiments (Prediction)

• We show Academic Network for brevity.
Task-driven Network Sampling

• Sampling Community Structure [Maiya’10][Satuluri’11]

• Sampling Network Backbone for Influence Maximization [Mathioudakis’11]

• Sampling High Centrality Individuals [Maiya’10]

• Sampling Personalized PageRank Values [Vattani’11]

• Sampling Network for Link/Label Prediction [Ahmed’12]
Short Summary

- Why sampling a social network?
  - the full network (e.g. Facebook) cannot be fully observed
  - crawling can be costly in terms of resource and time consumption (therefore a smart sampling strategy is needed)

<table>
<thead>
<tr>
<th></th>
<th>Homogeneous SN</th>
<th>Heterogeneous SN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node and Edge Selection</td>
<td>[Leskovec’06] [Adamic’01] [Ahmed’12] [Ribeiro’10]</td>
<td>[Kurant’12]</td>
</tr>
<tr>
<td>Sampling by Exploration</td>
<td>[Krishnamurthy’05] [Leskovec’06] [Hubler’08] [Gjoka’10] [Ribeiro’10] [Maiya’11] [Kurant’11]</td>
<td>[Gjoka’11] [Li’11] [Kurant’12] [Yang’13]</td>
</tr>
<tr>
<td>Task-driven Sampling</td>
<td>[Maiya’10] [Satuluri’11] [Mathioudakis’11] [Vattani’11] [Ahmed’12]</td>
<td></td>
</tr>
</tbody>
</table>
The 2nd part of this tutorial:

Social Network Summarization
Goals of Social Network Summarization

• Find a condensed representation of a given social network to
  – produce a succinct overview of the social network,
  – enable efficient storage,
  – facilitate efficient mining / query processing
Beyond Graph Summarization

• The goal is to summarize not only the structure or topology information such as:
  – Neighbor set / adjacency
  – Reachability
  – Connectivity

• but also the semantic info such as:
  • Attributes of an entity
  • Relationships of entity-entity, entity-community, community-community.
Issues for Summarization

• Purpose
  – Are there certain properties to preserve? Do we summarize the social network to facilitate certain query processing in the end?

• Precision of the summary
  – Lossless: can recover the exact original graph
  – Lossy: cannot fully recover, usually for a better compression ratio

• Evaluation
  – Space saving: Reduction of # node/edge, total data size in bytes, number of nonempty blocks in adjacency matrix, bit per edges, etc.
  – Efficiency: time for summarization and query processing on the summaries.
  – Quality: reconstruction errors, entropy, interestingness, query errors (degree, centrality, connectivity), etc.
Main Approaches for Summarization

• Aggregation based
  – Creating a summary graph with supernodes and superedges.
  – For efficient storage, analysis, and visualization.

• Abstraction
  – Extracting a subgraph given certain criteria for abstraction and various visualizations.

• Compression based
  – Encoding the network in a space-efficient way based on the structure information.

• Application-oriented
Web/Graph -> Social Network
- from the homogeneous to the heterogeneous structure
Main Directions

• Aggregation based
• Abstraction
• Compression based
• Application-oriented
Aggregation based on Node/Link Structures

• The basic idea:
  – Merge similar nodes into a supernode.
  – Add a superedge between two supernodes conditionally.
  – E.g., complete bipartite graph and cliques

• What if a subgraph is not complete?
  – E.g.,

Supernode graph + edge corrections!
Aggregation based on Node/Link Structures (cont’d)

• S-node representation for Web graphs [Raghavan’03]:
  – Partition web pages
    • URL split (domain) + Cluster split (adjacency list of out-links)
  – Supernode graph: a node represents a partition and a link between two partitions if there exists any link between two pages, one from each partition.
  – Positive/Negative superedge graphs: used to annotate the actual linkage between web pages.
  – Lossless representation.

• A two-part representation $R(S,C)$ is proposed [Navlakha’08]:
  – Graph summary $S$: an aggregated graph. Merge nodes with more common neighbors.
  – Edge corrections $C$: to be used while recovering the original graph.
  – Both lossless/lossy methods are proposed.
Edge Aggregations by Frequent Patterns

- Leverage pattern mining to compress the Web graph, which supports community discovery and random access [Buehrer’08].
- Two phases:
  - Clustering phase: nodes with similar out-links are grouped together.
  - Mining phase (for each cluster): mine virtual nodes to aggregate edges
- A lossless and more compact structure

<table>
<thead>
<tr>
<th>Vertex Id</th>
<th>Outlink List</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>1,2,3,5,6,10,12,15</td>
</tr>
<tr>
<td>55</td>
<td>1,2,3,5</td>
</tr>
<tr>
<td>102</td>
<td>1,2,3,20</td>
</tr>
<tr>
<td>204</td>
<td>1,7,8,9</td>
</tr>
<tr>
<td>13</td>
<td>1,2,3,8</td>
</tr>
<tr>
<td>64</td>
<td>1,2,3,5,6,10,12,15</td>
</tr>
<tr>
<td>43</td>
<td>1,2,3,5,6,10,22,31</td>
</tr>
<tr>
<td>431</td>
<td>1,2,3,5,6,10,21,31,67</td>
</tr>
</tbody>
</table>
Aggregation by Node Attributes and Relations

- SNAP [Tian’08]: Summarizing by Grouping Nodes on Attributes and Pairwise Relationships.

- User specify a node-attribute set and a relation-type set, the system returns an (A,R)-compatible grouping.
  - E.g., A={gender, department}, R={friends, classmates}

- Lossless.

Each student in group $G_1$ has at least a friend and a classmate in group $G_2$. 
Aggregation by Node Attributes and Relations (cont’d)

- *k*-SNAP [Tian’08]: relaxes the homogeneity requirement for the relationships and allows users to control (drill-down, roll-up) the sizes of the summaries.
  - *k* is the user-specified number of grouping nodes.
  - Not requiring that every node participates in a group relationship.
  - Lossy
Aggregation by Node Attributes and Relations (cont’d)

• [Zhang’10] Improves two limitations of k-SNAP in practice
  – Limitation 1: Only handles categorical node attribute
    • Sol: Provide cutoffs to categorize numerical attributes
  – Limitation 2: The search space is too large for manually identifying interesting summaries
    • Sol: An interestingness measure is introduced to evaluate the interestingness of a summary
Main Directions

• Aggregation based
• **Abstraction**
• Compression based
• Application-oriented
Visual Analysis of Large Heterogeneous Social Networks

- **OntoVis** [Shen’06] is a visual analysis tool for heterogeneous social network based on the given **ontology** graph
  - **Semantic abstraction**: generate an induced graph of node types selected by users
  - **Structural abstraction**: remove one-degree nodes and duplicate paths for reducing visual complexity
  - **Importance filtering**: using statistics such as node degree, dispersion, and disparity per type to determine the important node types for emphasizing.
Visual Analysis of Large Heterogeneous Social Networks (cont’d)

[Shen’06]

Ontology graph of the movie Dataset from the UCI KDD Archive

Importance filtering on “node type disparity”
Node size: disparity of connected types
# on edge: frequencies of links between two types

Visual Analysis of Large Heterogeneous Social Networks (cont’d)

Ontology graph of the movie
Dataset from the UCI KDD Archive¹

Semantic abstraction on “role-actor” relationships
Red nodes: role
Blue nodes: actor

Egocentric Abstraction on Heterogeneous Social Networks

- Construct the abstracted graph of an ego node for a heterogeneous social network
  - Identify each unique $k$-step linear combination of relations as a feature
  - Counting the frequency of each unique feature
  - Several criteria are introduced to decide which features are important for the ego
    - E.g., abstraction by showing only rare/frequent features.

[Li’09]
Main Directions

- Aggregation based
- Abstraction
- **Compression based**
- Application-oriented
Ordering-based Compression

• URLs of web pages have two features:
  – **Similarity**: pages that are proximal in the lexicographic ordering tend to have similar sets of neighbors.
  – **locality**: many links are intra-domain, and therefore likely to point to pages nearby in the lexicographic ordering.

• By leveraging the *lexicographical order*, the BV scheme [Boldi’04] needs only 3bits per edge to encode the Web graphs.

• However, nodes in social networks have no natural orders.
Ordering for Nodes in Social Networks

• The **shingle ordering** based on Jaccard coefficient to find locality in social networks [Chierichetti’09].

\[
M_\sigma(A) = \sigma^{-1}(\min_{a \in A} \{\sigma(a)\})
\]

\[
\Pr[M_\sigma(A) = M_\sigma(B)] = \frac{|A \cap B|}{|A \cup B|}
\]

\[
= J(A, B).
\]

• If two nodes share a lot of common out-neighbors, with high probability they will be close to each other in a shingle-based ordering.
Neighbor Query Friendly Compression of Social Networks

- A novel Eulerian data structure using multi-position linearizations (MP) of directed graphs is proposed to compress social networks while both out/in neighbor queries can be answered in sublinear time [Maserrat’10].

Original graph G

Compose an Eulerian path by duplicating nodes

v(1) = v_8, v(2) = v_7, and so on.

MP_1-linearization of G
(Min. distance among all pairs of (u,v) is 1)

- Local information: 2 bits to encode if (v(i-1), v(i)) and (v(i), v(i+1)) exists in E(G).
- Pointers: next appearance of v(i).
Neighbor Query Friendly Compression of Social Networks (cont’d)

Neighbor query of $v$ in $O(\sum_{u \in N_v} \text{deg}(u) \log |V(G)|)$

The upper bound of bits used for encoding a graph is asymptotically about $\frac{1}{2} \log(|V(G)|)$, which is the number of bit used for encoding an edge by baseline.

* Similar ideas are also used for lossy compression to preserve communities in social networks [Maserrat’12].
Other Graph/Network Compressions

• Community-based (hubs and spokes) Compression [Kang’10]
• Mix clusterings and orders for Compressing Social Networks [Boldi’11]
• Encoding based on the newly defined structural entropy for Erdös-Rényi graphs [Choi’12].
Main Directions

- Aggregation based
- Abstraction
- Compression based
- Application-oriented
Application-Oriented Summarization

• Summarization for query-answering and pattern mining
  – Adjacency, degree, centrality [LeFevre’10]
  – Connectivity [Zhou’10][Toivonen’11]
  – Graph pattern mining/search [Chen’09][Kang’10][Fan’12]
• Graph management system [Kang’11]
  and so on.
REMARKS
Social Network Summarization Overview

Accuracy

Lossless

Lossy

Network

Homogeneous

Heterogeneous

Aggregation-based

Abstraction

Compression

Application-oriented

Strategy
## Summarization Categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Homogeneous</th>
<th>Heterogeneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregation-based</td>
<td>[Raghavan’03][Navlakha’08] [Buehrer’08]</td>
<td>[Tian’08] [Zhang’10][Liu’11]</td>
</tr>
<tr>
<td>Abstraction</td>
<td></td>
<td>[Shen’06][Li’09]</td>
</tr>
<tr>
<td>Compression</td>
<td>[Chierichetti’09][Maserrat’10] [Maserrat’12] [Kang’10][Choi’12]</td>
<td></td>
</tr>
<tr>
<td>Application-oriented</td>
<td>[Zhou’10][LeFevre’10] [Toivonen’11][Kang’11]</td>
<td>[Chen’09][Fan’12]</td>
</tr>
</tbody>
</table>

**Summarization Strategies:** _Lossless / Lossy_
Opportunities for Future Research

• Advanced techniques to sample/summarize more complex graph structures
  – E.g. location-based social networks, diffusion networks, dynamic social networks

• For task-driven sampling and summarization: A general framework across tasks are still missing

• Sampling/Summarization given noisy data

• Standard evaluation metrics and benchmark data are highly demanding.

• Many others…
Final Remarks

• Sampling and summarization have immediate practical value in big data era
  – Allow data miners to perform advanced mining tasks in large graphs
  – Enable scalable storage and querying
  – Facilitate the development of real-world applications

• Existing works are rich, but by no means complete to handle every aspect of the problem.
Acknowledgements

• This tutorial is partially sponsored by National Science Council, National Taiwan University and Intel Corporation under Grants NSC101-2911-I-002-001, NSC101-2628-E-002-028-MY2 and NTU102R7501

• Special thanks to Shu-Ming Hsu @ Academia Sinica for his inputs
Reference – Homogeneous Sampling

Reference – Heterogeneous Sampling

- J.-Y. Li and M.-Y. Yeh. On Sampling Type Distribution from Heterogeneous Social Networks. In PAKDD 2011.
- Cheng-Lun Yang, Perng-Hwa Kung, Chun-An Chen, Shou-De Lin Semantically Sampling in Heterogeneous Social Networks in WWW 2013
Reference – Task-driven Sampling

• V. Satuluri, S. Parthasarathy, and Y. Ruan. Local Graph Sparsification for Scalable Clustering. In SIGMOD 2011.
References: Aggregation-based Summarization

References: Abstraction-based Summarization


• C.-T. Li and S.-D. Lin. Egocentric Information Abstraction for Heterogeneous Social Networks, In Proc. of International Conference on Advances in Social Network Analysis and Mining (ASONAM’09), 2009.
References: Compression-based Summarization

References: Application-oriented Summarization