Department of
Electronic & Computer Engineering

ELEC6910Q – Analytics and Systems for Social Media and Big Data Applications
Lecture 6

Prof. James She
james.she@ust.hk
Announcements

1. Midterm Project (due Oct. 20) – show what learnt with your ideas
   • Details will be discussed.

2. Guest Speaker from NextDigital next week
   • “Analytics and Systems for Social Media and Big Data Applications”

3. Marks of your tutorials and activities. Check online & approach your TAs by tomorrow if any question.
For this dataset, item-item CF performs better than user-user CF
Selected Works from T3

- $k$ increases \(\rightarrow\) recommendation performance increases and converges
- What if $k > 100$? keep converging?
Last lecture

Recall Collaborative Filtering
- People who purchased A also purchased B
  - Different from k-NN:
    - Recommendations from opinions of users not similar (in terms of profile) to you
  - User-based
    - Recommend things that were purchased or viewed by users who are similar to you
  - Item-based
    - Recommend things that are similar to the items that you have viewed/purchased before

Communities in Social Media
1. 2 types of groups in social media:
   - Explicit groups: formed by user subscriptions
   - Implicit groups: implicitly formed by social interactions
2. Social interactions indicate communities:
   - Messaging
   - Tagging
   - etc.
3. Through community detection
   - Not all site have the joining function or user don't use it
   - Group changes dynamically
Outcome of this lecture

1. Recall – $k$-NN and Cluster Coefficient

2. Community Detection

3. Applications of Community Detections
Recall $k$-NN and more thinking

1. most simplest non-parametric machine learning

2. classification: a sample is classified by a majority vote of its neighbors;

3. regression: the output is some property value of a sample averaged from its $k$ nearest neighbors.
Recall *k*-NN and more thinking: Re-think

**Distance-Weighted k-NN**

Might want to weight nearer neighbors more heavily...

\[
\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i}
\]

where

\[
w_i \equiv \frac{1}{d(x_q, x_i)^2}
\]

and \(d(x_q, x_i)\) is distance between \(x_q\) and \(x_i\)
Recall - Local Measurement

Focusing on neighboring nodes and edges

- Focusing on the node

- The local clustering coefficient of a node in a graph quantifies how close its neighbors are to being a clique

- The local clustering coefficient is defined as:

\[
C_i = \frac{2|\{e_{jk}\}|}{k_i(k_i - 1)}
\]

- \(C_i\) is the local clustering coefficient of node \(i\), \(|\{e_{jk}\}|\) is the # of edge exist within the neighbor of node \(i\), \(k_i\) is the degree of node \(i\).

Note: \(j\) and \(k\) are any two neighbors of node \(i\) that are connected.
Recall - Local Measurement

Local clustering coefficient - example

- Calculate the local clustering coefficient of the blue node with
  \[ C_i = \frac{2|\{e_{jk}\}|}{k_i(k_i - 1)} \]

- Maximum of 3 connections among the neighbor

- In top, \( C_i = \frac{2*3}{3*2} = 1 \)

- In middle, \( C_i = \frac{2*1}{3*2} = \frac{1}{3} \)

- In bottom, \( C_i = \frac{2*0}{3*2} = 0 \)

\[ c = 1 \]
\[ c = \frac{1}{3} \]
\[ c = 0 \]
Outcome of this lecture

1. Recall – $k$-NN Cluster Coefficient

2. Community Detection

3. Applications of Community Detections
Visualization of Community

For better understanding of your data
Coloring the communities in Visualization

Given knowledge of nodes interests from dataset
Identifying Community

Additional knowledge

e.g., Consider connectedness of friends in the figure.

x and y have both three friends in a big community

1. x’s friends are independent and y’s friends are all connected

2. who’s friends are more likely to join the same community?

Why do we care?
• Information argument [Granovetter 1973]
  • unconnected friends give independent support

• Social capital argument [Coleman 1988]
  • safety / trust advantage in having friends who know each other
Identifying Community

Subjective definition

Each component is a community

Definition is subjective
Community Detection
Community Detection
How to find the community

- Criteria vary depending on the tasks
- Roughly, community detection methods can be divided into 4 categories (not exclusive):
  - **Node-Centric Community**
    - Each node in a group satisfies certain properties
  - **Group-Centric Community**
    - Consider the connections within a group as a whole. The group has to satisfy certain properties without zooming into node-level
  - **Network-Centric Community**
    - Partition the whole network into several disjoint sets
  - **Hierarchy-Centric Community**
    - Construct a hierarchical structure of communities
Node-Centric Community Detection

Focus on properties of nodes

• Nodes satisfy different properties
e.g., Complete Mutuality
  • **Clique:** a maximum complete subgraph in which all nodes are adjacent to each other

Nodes 5, 6, 7 and 8 form a clique

• NP-hard to find the maximum clique in a network
• Straightforward implementation to find cliques is very expensive in time complexity

• Commonly used in traditional social network analysis
• Here, we discuss some representative ones
Node-Centric Community Detection

Clique Percolation Method to Find/Use Cliques

- CPM is such a method to find overlapping communities
  - Input
    - A parameter k, and a network
  - Procedure
    - Find out all cliques of size k in a given network
    - Construct a clique graph. Two cliques are adjacent if they share k-1 nodes
    - Each connected components in the clique graph form a community
Node-Centric Community Detection

CPM example

1. Find all cliques with size $k$ (3 in this case)

2. Connect cliques if they share $k-1$ (2 in this case) nodes

Communities:

$\{1, 2, 3, 4\}$
$\{4, 5, 6, 7, 8\}$

3. Form community with connected component

Cliques of size 3:

$\{1, 2, 3\}$, $\{1, 3, 4\}$, $\{4, 5, 6\}$,
$\{5, 6, 7\}$, $\{5, 6, 8\}$, $\{5, 7, 8\}$,
$\{6, 7, 8\}$
Hierarchy-Centric Community Detection
Divisive Hierarchical Clustering (DHC) [Girvan and Newman 2002]

1. For every edge, compute its betweenness
2. Remove the edge with the highest betweenness
3. Re-compute the edge betweenness
4. Repeat until no more edge exists or until specified number of clusters produced
Hierarchy-Centric Community Detection

DHC Example

Recompute the betweenness until no more edge

Highest betweenness edge, removed in first step

If the specified number of cluster is two, operation stopped
Hierarchy-Centric Community Detection

DHC Example

First hierarchy:
After removing 9 edges with highest betweenness centrality (a), the graph is partitioned into 2 disjoint sets (b).

Second:
Treat each isolated cluster as a single graph and repeat steps for 1st hierarchy.
Hierarchy-Centric Community Detection
Another DHC Example

Divisive clustering on Edge-Betweenness

- Progressively remove edges with the highest betweenness
  - Remove e(2,4), e(3, 5)
  - Remove e(4,6), e(5,6)
  - Remove e(1,2), e(2,3), e(3,1)
Group-centric Community Detection

• Consider the connections within a group that satisfies certain properties

• Goal: verify if the group is an community

• Approach:
  • Density-based groups
Group-centric Community Detection
Density-based Groups

- It is acceptable for some nodes to have low connectivity

- The whole group satisfies a certain condition
  - e.g., the group density $> \gamma$ (threshold)
  - $\gamma$ – dense quasi-clique: $\frac{|E_s|}{|V_s|(|V_s|-1)/2} > \gamma$ where

| $|E_s|$ | # of edges in the subgraph |
| $|V_s|$ | # of vertices in the subgraph |
Group-centric Community Detection
An example

• Given the sampled subgraph and $\gamma = 0.6$
  • Subgraph 1, calculate $\frac{|E_s|}{|V_s||V_s|/(|V_s|-1)/2}$
    • $\frac{7}{5(5-1)/2} = 0.7 > \gamma$
  • Subgraph 2, calculate $\frac{|E_s|}{|V_s||V_s|/(|V_s|-1)/2}$
    • $\frac{5}{5(5-1)/2} = 0.5 < \gamma$
  • Subgraph 3, calculate $\frac{|E_s|}{|V_s||V_s|/(|V_s|-1)/2}$
    • $\frac{5}{4(4-1)/2} = 0.833 > \gamma$

• Which groups will be a community?

• Threshold can be learned from known community
In-class Activities 6
2 person, 10 mins

- Use \( \frac{|E_s|}{|V_s|(|V_s|-1)/2} \) – dense quasi-clique:
  to prove which vertices are likely to be in a community

\(|E_s|\): # of edges in the subgraph
\(|V_s|\): # of vertices in the subgraph
5 minutes break
Mid-term project (individual, Oct 20 due)

1. 4-page only project paper (in **IEEE conference paper format**).
   - **Topic** (your idea related to the course, or our suggestions online)
   - **Datasets** (from online, or collection if you can)
   - **Data processing** (from lecture + tutorial)
   - **Methodologies for justifications** (from lecture)
   - **Conclusions** (from what you learnt)

2. Deadline: 23:59, Oct. 20 (2 weeks from now)

3. Hints: Using interesting datasets, e.g., [https://snap.stanford.edu/data/](https://snap.stanford.edu/data/)

4. Details about scoring: [http://course.ee.ust.hk/elec6910q/project/ScoreRubric.pdf](http://course.ee.ust.hk/elec6910q/project/ScoreRubric.pdf)
Mid-term project

1. **Topic (your own idea or pick from our suggestions)**
   - the topic you selected should be ORIGINAL, and "cool, interesting or timely".
   - clearly state what or why it is motivated to work on your topic, and possible results (claims, discoveries, conclusions, etc.)

2. **Datasets (Either one)**
   - Download from online (e.g.: https://snap.stanford.edu/data/)
   - Provided by us

3. **Data processing / Methodologies**
   - apply what learnt (data visualizations + social network analysis techniques) to proceed the datasets to justify your results

4. **Conclusions**
   - a simple paragraph to summarize your results (≤ 3 points) with your justifications
Suggested topic 1

1. Topic: Movie Recommendation

2. Datasets: MovieLens Dataset
   • http://grouplens.org/datasets/movielens/

3. Data processing / Methodologies
   • Use Collaborative Filtering as baseline for comparison
   • Propose your own method for recommendation
   • Compare with state-of-the-art results to know how good you are

4. Reference:
Suggested topic 2

1. Topic: Community Detection

2. Datasets: Digg.com
   • http://course.ee.ust.hk/elec6910q/material/data.mat

3. Data processing / Methodologies
   • Implement at least 2 community detection methods
   • Utilize the detected communities for recommendation
   • Use K-NN as baseline for comparison

4. Reference:
Suggested topic 3

1. Topic: Recommending friendships with User-Shared Images on Social Media

2. Datasets: Fotolog (please contact Ming(cpming@ust.hk))

3. Data processing / Methodologies
   • compute user similarity based on shared images
   • rank the similarity and recommend friendships
   • evaluate the recommended friendship with the ground truth
     • i.e., % of recommended friendships actually exist

4. Reference:
Network-Centric Community Detection

- Network-centric criterion needs to consider the connections within a network globally
- Goal: partition nodes of a network into disjoint sets
- Approaches:
  - Clustering based on vertex similarity
  - Latent space models
  - Block model approximation
  - Spectral clustering
  - Modularity maximization
Network-Centric Community Detection Clustering Based on vertex similarity

- Vertex similarity is defined by how similar their interaction patterns are.

- 2 nodes (vertexes) are structurally equivalent if connect to the same nodes:
  - e.g., nodes 8 and 9 are structurally equivalent (connects to 1, 6, 13)
  - groups are defined over equivalent nodes

- In practice, use vector similarity:
  - structurally equivalent too strict as rarely occur in a large
  - e.g., cosine similarity, Jaccard similarity and k-mean
Network-Centric Community Detection
Clustering based on vertex similarity

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a vector

structurally equivalent

Jaccard Similarity:

\[
J(A, B) = \frac{|A \cap B|}{|A \cup B|}.
\]

\[
J(5, 8) = \frac{|\{6\}|}{|\{1, 2, 6, 13\}|} = 1/4
\]

\[
J(5, 9) = \frac{|\{6\}|}{|\{1, 2, 6, 13\}|} = 1/4
\]
Network-Centric Community Detection
Clustering based on vertex similarity

• For practical use with huge networks:
  – Consider the connections as features
  – Use Cosine or Jaccard similarity to compute vertex similarity
  – Apply classical k-means clustering Algorithm

• K-means:

**Algorithm 1** Basic K-means Algorithm.

1: Select $K$ points as the initial centroids.
2: repeat
3: Form $K$ clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: until The centroids don’t change
Network-Centric Community Detection

**K-means**

**Initialize** $m_i, i = 1, \ldots, k$ (e.g., $k$ randomly selected $x^{(\ell)}$)

**Repeat**

For all $x^{(\ell)} \in X$

$$b_i^{(\ell)} = \begin{cases} 1 & \text{if } i = \text{arg min}_j \| x^{(\ell)} - m_j \| \\ 0 & \text{otherwise} \end{cases}$$

For all $m_i, i = 1, \ldots, k$

$$m_i = \frac{\sum_\ell b_i^{(\ell)} x^{(\ell)}}{\sum_\ell b_i^{(\ell)}}$$ *Find the new mean $m_i$ with $x^{(\ell)}$ in group $i$*

**Until** $m_i$ converge.
Network-Centric Community Detection

$K$-means
Recall- Network-Centric Community Detection Clustering Based on vertex similarity

- Vertex similarity is defined by how similar their interaction patterns are.

- 2 nodes (vertexes) are structurally equivalent if connect to the same nodes:
  - e.g., nodes 8 and 9 are structurally equivalent (connects to 1, 6, 13)
  - groups are defined over equivalent nodes.

- In practice, use vector similarity:
  - structurally equivalent too strict as rarely occur in a large
  - e.g., cosine similarity, Jaccard similarity and k-mean.
Recall- Network-Centric Community Detection
Clustering based on vertex similarity

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>a vector</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>structurally equivalent</td>
<td>8</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Jaccard Similarity:

\[ J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \]

\[ J(5, 8) = \frac{|\{6\}|}{|\{1, 2, 6, 13\}|} = \frac{1}{4} \]

\[ J(5, 9) = \frac{|\{6\}|}{|\{1, 2, 6, 13\}|} = \frac{1}{4} \]

\[ J(8, 9) = \frac{|\{1, 6, 13\}|}{|\{1, 6, 13\}|} = 1 \]
Recall- Network-Centric Community Detection

K-means
-End of Lecture 6 –

Questions / Comments?