CS 512 Assignment 1

Chia-Hao Hsieh (chsieh17)

Question 1: Network Measures and Models

It is important to study the measures of a network, such as centrality of nodes in a network and model the structure of a network using a good abstract model.

1. What are the differences and relationships among the following measures: (1) degree centrality, (2) Eigenvector centrality, (3) PageRank, and (4) HITS? What are the key differences between PageRank and HITS algorithms?

Degree centrality measures the degree of nodes. The higher degree a node has, the more important it is.

Eigenvector centrality measures prestige of nodes. Like degree centrality, nodes with higher degree get higher prestige. In addition, high prestige neighbor nodes contribute more to prestige of a node.

PageRank measures the possibility of a random node walker lands at a node. At any node, the random node walker model randomly jumps to an adjacent node, or in a small possibility, jumps to any other node.

HITS computes two values for a page: authority value and hub value. Authority value is similar to PageRank or prestige while hub value is based on how many highly-ranked nodes it points to. By computing these two values iteratively, HITS improves its result.

PageRank only considers the fact that a good page is cited by many pages while HITS also takes into account that a good page cites many other good pages as well.

2. Researchers have been modeling social and/or information networks using several models. What are the relationships and differences among the following three models: (1) Edös-Rényi random graph model, (2) Watts-Strogatz small world model, and (3) Barabasi-Albert scale-free network model? Show analytically the preferential attachment method in the Barabasi-Albert scale-free network model generates the networks whose node degree distribution follows the power-law distribution.

ER model generates a random graph by starting with a bunch of nodes and randomly adding edges between them. Every possible edge happens independently with a constant probability. ER graphs

are sharply concentrated but not heavy-tailed. They have low clustering coefficient. Their degree distribution converges to a Poisson distribution.

Watts-Strogatz small world model (the beta model) produces graphs with small world properties, including short average path lengths and high clustering.

ER graphs have small diameter while the beta model graphs have small diameter and high clustering. However, the beta model still fails to produce scale-free networks, whose degree distribution follow a power law.

Barabasi-Albert scale-free network model (BA model) produce scale-free networks by preferential attachment:

Let N(k, t) be the number of nodes with degree k at time t.

A new node is added at each time-step, so we have the total number of nodes N = t.

The total number of links is mt.

The average degree is 2m since each link contributes to the degree of two nodes.

Each new node arrives with degree m.

The probability that the new node will link to a degree-k node is:

$$\Pi(k) = \frac{k}{\sum_j k_j} = \frac{k}{2mt} \cdot \tag{5.31}$$

where 2mt is the total degrees of all nodes.

After a new node is added, the expected number of new links connected to degree k nodes is:

$$\frac{k}{2mt} \times N p_k(t) \times m = \frac{k}{2} p_k(t), \tag{5.32}$$

where the degree distribution $p_k(t) = N(k, t) / N$.

So $Np_k(t)$ is just $N(k, t)$, the number of nodes with degree k at time t.

With a new link added, these k-degree nodes are now k+1-degree nodes. That is, we have:

$$\frac{k}{2} p_k(t) \tag{5.33}$$

number of k-degree nodes less than before.

Similarly, there are also some (k-1)-degree nodes are now k-degree:

$$\frac{k-1}{2} p_{k-1}(t). \tag{5.34}$$

So we now know the expected number of k-degree nodes after new node addition:

$$(N+1)p_k(t+1) = Np_k(t) + \frac{k-1}{2} p_{k-1}(t) - \frac{k}{2} p_k(t). \tag{5.35}$$

This equation applies to all k > m.

For k=m, we have a different equation:

$$(N+1)p_m(t+1) = Np_m(t) + 1 - \frac{m}{2} p_m(t) \ . \tag{5.36}$$

because there is only one newly-added m-degree node, that is, the new node itself.

For 0<k<m, the number of k-degree nodes is always 0 since each node arrives with m degrees.

Then we can rewrite those two equations as the following:

Eq. 5.35 and 5.36 are the starting point of the recursive process that provides p_k. Let us use the fact that we are looking for a stationary degree distribution, supported by numerical simulations Fig. 5.6. This means that in the $N = t \to \infty$ limit, $p_k(\infty) = p_k$. Using this we can write the l.h.s. of Eq. 5.35 and 5.36 as $(N+1)p_k(t+1) - Np_k(t) \to Np_k(\infty) + p_k(\infty) - Np_k(\infty) = p_k(\infty) = p_k$ $(N+1)\, p_{mk}(t+1) - Np_m(t) \to p_m$. Therefore the rate equations Eq. 5.35 and 5.36 take the form:

$$p_k = \frac{k-1}{k+2} p_{k-1} \qquad k > m \tag{5.37}$$

$$p_m = \frac{2}{m+2} \tag{5.38}$$

Note that Fig. 5.37 can be rewritten as

$$p_{k+1} = \frac{k}{k+3} p_k \tag{5.39}$$

via a $k \to k+1$ variable change.

Starting from Eq.. 5.38, we can compute p_{m+1}, p_{m+2}, p_{m+3} using Eq.5.39:

$$\begin{aligned} p_{m+1} &= \frac{m}{m+3} p_m = \frac{2m}{(m+2)(m+3)} \\ p_{m+2} &= \frac{m+1}{m+4} p_{m+1} = \frac{2m(m+1)}{(m+2)(m+3)(m+4)} \\ p_{m+3} &= \frac{m+2}{m+5} p_{m+2} = \frac{2m(m+1)}{(m+3)(m+4)(m+5)} \end{aligned} \tag{5.40}$$

By observing these ps, we can find a pattern for p_k:

$$p_k = \frac{2m(m+1)}{k(k+1)(k+2)}, \tag{5.41}$$

For large k,

$$p_k \sim k^{-3}. \tag{5.46}$$

All the above pictures are taken from:
http://barabasilab.neu.edu/networksciencebook/download/network_science_december_ch5_2013.pdf

Question 2: Clustering in Heterogeneous Information Networks

A heterogeneous information network consists of objects and links of heterogeneous types and thus can be used to represent sophisticated relationships among objects and their links in information networks.

1. RankClus clusters heterogeneous information networks by integrating ranking and clustering in the clustering process.

(1) Why does such an integration lead to better quality of clustering than SimRank-based clustering?

With RankClus, ranking and clustering can mutually enhance each other. The higher ranking a node gets within a cluster, more likely it plays an important role in that cluster, and vice versa.

SimRank, on the other hand, is relatively simple and doesn't consider different importance of each node in the network.

(2) Why is RankClus more efficient than SimRank?

With SimRank, it computes similarity scores for all possible pairs of any two nodes. Its time complexity is $O(V^2)$.

On the other hand, RankClus just compares distance with all cluster center for each iteration.

According to the paper *RankClus: Integrating Clustering with Ranking for Heterogeneous Information Network Analysis*:

"At each iteration, the time complexity of RankClus is comprised of three parts: ranking part, mixture model estimation part and clustering adjustment part. … Overall, the time complexity is $O(t(t_1|E|+t_2(K|E|+ K +mK) +mK^2))$, where t is the iteration number of the whole algorithm and t_2 is the iteration number of the mixture model. If the network is a sparse network, the time is almost linear with the number of objects. "

The time complexity of RankClus is almost linear when the network is sparse.

2. When clustering a heterogeneous information network, different meta-paths carry different semantic meanings and thus lead to different clustering results.

(1) Describe three different meta-paths in the DBLP network, and explain the semantic meanings of them. And (2) describe what clustering results you expect when performing clustering with these meta-paths.

A − P − A means that two authors directly collaborate with each other on some paper. Authors that collaborate together will be in the same cluster.

A − P − V − P − A means that two authors publish on the same venue. Authors that work on similar topics and have similar reputation will be in the same cluster.

A − P → P ← P − A means two authors co-cite another paper on each of their papers. Authors that cite the same papers will be in the same cluster.

(3) design a mechanism so that a user may give his/her guidance to generate clusters with desired features.

Ask a user to provide some examples for his/her desired result.

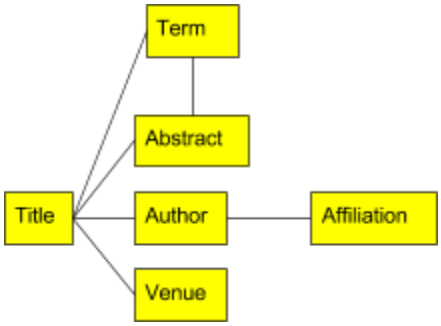
We then analyze these examples, and find meta-paths that can classify nodes into the same (or at least similar) results.

With these meta-paths, we can get the desired result.

Question 3: Classification of Heterogeneous Information Networks

Structures of a heterogeneous information network often carries critical information for classification. Take PubMed (http://en.wikipedia.org/wiki/PubMed) as an example. Discuss the following:

1. Describe how to construct a heterogeneous information network from the PubMed database, and give several example meta-paths in such a network and their semantic meanings.



Author – Title – Author means that two authors directly collaborate with each other on a paper.

Author – Title – Abstract – Title – Author means that two authors work on similar topic.

2. Discuss what will be good examples for classification of such a heterogeneous information network, and how you would modify the RankClass algorithm or propose a new meta-path-based algorithm to perform effective classification over the PubMed information network.

To learn the effect of certain medicines, we can classify papers by medicines they discuss.

First, collect all medicines mentioned by all papers. These are the classes.

For each medicine, initialize each paper a ranking scores based on the how many times the medicine is mentioned in the title and abstract. Normalize the score based on the total time the medicines get mentioned in all papers.

Iteratively update ranking scores by looking at the ranking of neighbors until the scores converge. We then classify each paper by its highest ranking score among all classes.

Question 4 (Programming Required): Similarity Search in Heterogeneous Information Network

This task is to write a program to take a heterogeneous network and evaluate similarity queries based on user-provided meta-paths. The data input is a heterogeneous information network of academic publications, with 4 types: author, venue, paper and term.

- **[Data Set (Heterogeneous Version): dblp_4area.zip]** In the dataset we have 4 node files named after its corresponding node type and one relation file:
  - author.txt: contains all the researchers, first column is id, second column is researcher name
  - venue.txt: contains 20 representative conferences in 4 areas: data mining, database, information retrieval and machine learning. First column is id, second column is conference name. Venues in different years are combined into the same entity (e.g., SIGMOD'05 and SIGMOD'06 both refer to entity SIGMOD)
  - paper.txt: contains the papers published in these 20 conferences before 2011
  - term.txt: contains all the non-stopword unigrams extracted from paper titles
  - relation.txt, used to store the undirected relations between entities. First column contains an ID of paper, second column contains an ID of the other three types

  [Data Set (Homogeneous Version): APVPA_net, APTPA_net] These two files are the relation files of two compressed homogeneous network using meta-path: APVPA and APTPA, respectively. Both network only contains the relation between authors. Please use these two networks for Personalized Page-Rank.

The goal is to evaluate similarity queries using three similarity measures: PathSim and Personalized Page-Rank. The query input is a researcher's name (e.g., Jiawei Han), the output is the top-10 most similar researchers.

Implementation tips can be found in **hw1tips.pdf.**

- Sub-Task 1. Output the top-10 ranked results (i.e., similar researchers) for two authors: "Christos Faloutsos" and "AnHai Doan", using PathSim and Personalized Page-Rank as measures respectively, taking APVPA (author-paper-venue-paper-author) as meta-path
- Sub-Task 2. Output the top-10 ranked results for two researchers: "Xifeng Yan" and "Jamie Callan", using PathSim and Personalized Page-Rank as measures respectively, but taking APTPA (author-paper-term-paper-author) as meta-path

[Caveats]
1) You can use any programming language you want to write the algorithm. In particular, C/C++, Java, Python, Matlab, R are preferred.
2) Please put the results in the submitted pdf.

3) Source code needs to be uploaded with the pdf and compressed into a single zip.
4) It's encouraged to have decoupled functions/classes for different measures to reduce grading workload.

The top similar authors to Christos Faloutsos using APVPA with PathSim are:

Christos Faloutsos
Jiawei Han
Rakesh Agrawal
Hans-Peter Kriegel
Jian Pei
Raghu Ramakrishnan
H. V. Jagadish
Nick Koudas
Hector Garcia-Molina
Divesh Srivastava

The top similar authors to AnHai Doan using APVPA with PathSim are:

AnHai Doan
Jignesh M. Patel
Xuemin Lin
Balakrishna R. Iyer
Jun Yang
Mohamed F. Mokbel
Jayant R. Haritsa
Walid G. Aref
Ming-Chien Shan
Richard T. Snodgrass

The top similar authors to Xifeng Yan using APTPA with PathSim are:

Xifeng Yan
Hong Cheng
Mohammed Javeed Zaki
Jianyong Wang
Ke Wang
Srinivasan Parthasarathy
Jiong Yang
Wynne Hsu
Jian Pei
Anthony K. H. Tung

The top similar authors to Jamie Callan using APTPA with PathSim are:

Jamie Callan
Chris Buckley
Naphtali Rishe
Vijay V. Raghavan
Jie Lu
Alan F. Smeaton
Gerard Salton
Kalervo Järvelin
Nicholas J. Belkin
Prasenjit Mitra

The top similar authors to Christos Faloutsos using APVPA with P-PageRank are:

Christos Faloutsos
C. Lee Giles
Jiawei Han
Huan Liu
Wei Fan
Deng Cai
Philip S. Yu
Wei Zhang
Yong Yu
Ji-Rong Wen

The top similar authors to AnHai Doan using APVPA with P-PageRank are:

AnHai Doan
C. Lee Giles
Jiawei Han
Huan Liu
Philip S. Yu
Yong Yu
Ji-Rong Wen
Craig A. Knoblock
Qiang Yang
Wei Fan

The top similar authors to Xifeng Yan using APVPA with P-PageRank are:

Xifeng Yan

Michail Vlachos
Dale Schuurmans
Rakesh Agrawal
Rajeev Motwani
Padhraic Smyth
Ming-Syan Chen
Divesh Srivastava
Mong-Li Lee
Shlomo Zilberstein

The top similar authors to Jamie Callan using APVPA with P-PageRank are:

Jamie Callan
Michail Vlachos
Dale Schuurmans
Rakesh Agrawal
Rajeev Motwani
Padhraic Smyth
Ming-Syan Chen
Divesh Srivastava
Mong-Li Lee
Shlomo Zilberstein