Web Structure Mining (and Associates)

Overview

Terminology:

- Link Analysis: analysis of graph structures.
- Web Structure Mining: analysis of the web graph.
- Social Network Analysis: analysis graphs representing relationships between humans (social networks).

Overview:

- Basics of social network analysis.
  - Node Centrality.
  - Node Prestige.
- Co-citation and Bibliographic Coupling.
- HITS algorithm (hubs and authorities)
- PageRank algorithm (*see separate handout*)
- Community Discovery.

Basics of Social Network Analysis

From textbook:

> Social network [analysis] is the study of social entities (...called **actors**), and their interactions and relationships.

Social network: a **graph** representing interactions and/or relationships between different actors.

Interactions/Relationships:

- Sent email to (asymmetric);
- Is a boss of (asymetric);
- Is a friend of (symetric);
- Works on the same project with (symmetric);
- Co-wrote a paper with (symmetric);
- Referenced work by (assymetric);
- ...

Social networks can be both directed and undirected depending on the type of the interaction/relationship they represent.

Edges in social networks are called links or ties

Notation. Let $I = \{i_1, \ldots i_n\}$ be a collection of actors. A social network is a graph $G = \rangle I, E \langle$, where E is the collection of links between pairs of actors (either directed or undirected).

Given an actor $i \in I$:

$d(i)$ denotes the degree of i in G;
$d_i(i)$ denotes the in-degree of i in G (if G is directed);
$d_o(i)$ denotes the out-degree of i in G (if G is directed);

We note, that for a social network G with n actors, the maximum possible degree of an actor can be $n-1$.

Given two actors i and j, as $d(i,j)$ we denote the length of the shortest path from i to j (the number of links on the path), if such a path exists.

Centrality

A **central actor** is an actor with many ties.

Degree Centrality

This way of computing centrality looks at the degree of the actor's node in the network.

Undirected Graph. For undirected social network G, degree centrality of an actor i, denoted $C_D(i)$ is:

$$C_D(i) = \frac{d(i)}{n-1}.$$

Undirected Graph. For directed social network G, degree centrality of an actor i, denoted $C_D(i)$ is:

$$C_D(i) = \frac{d_o(i)}{n-1}.$$

(i.e., the degree centrality is based only on out-links).

Closeness Centrality

This way of computing centrality looks at how close an actor is to other actors in the network.

For both **directed** and **undirected** graphs, closeness centrality of an actor i, denoted $C_C(i)$ is defined as:

$$C_C(i) = \frac{n-1}{\sum_{j \in I} d(i,j)}.$$

$0 < C_C(i) \leq 1$, as minimal distance of $d(i,j)$ for all $j \neq i$ is 1, thus, the minimal value for $\sum_{j \in I} d(i,j)$ is $n-1$.

To compute closeness centrality for all actors, the graph must be connected.

Betweenness Centrality

This way of computing centrality measures how often an actor appears on the interaction paths between two other actors.

Undirected graphs. Given two actors j and k, let p_{jk} denote the number of shortest paths between j and k. Let $i \neq j$ and $i \neq k$ be a third actor. We denote as $p_{jk}(i)$ the number of shortest paths between j and k that pass through i.

Then, the betweenness centrality of an actor i, denoted $C_B(i)$ is defined as

$$C_B(i) = \sum_{j \in I, j \neq i} \sum_{k \in I, k \neq j, k \neq i} \frac{p_{jk}(i)}{p_{jk})}.$$

$C_B(i)$ ranges from 0 to $\frac{(n-1)(n-2)}{2}$ (the number of pairs of actors who are not i).

$C_B(i)$ can be normalized:

$$C_B(i) = \frac{2 \sum_{j \in I, j \neq i} \sum_{k \in I, k \neq j, k \neq i} \frac{p_{jk}(i)}{p_{jk})}}{(n-1)(n-2)}.$$

Directed graph.

$$C_B(i) = \frac{\sum_{j \in I, j \neq i} \sum_{k \in I, k \neq j, k \neq i} \frac{p_{jk}(i)}{p_{jk})}}{(n-1)(n-2)}.$$

(because the graph is directed (x,y) and (y,x) are now different edges).

Prestige

An actor has high prestige if the actor is an object of extensive ties **as a recipient**.

Prestige is defined on **directed graphs**.

Degree Prestige

Degree prestige of actor i in the social network, denoted $P_D(i)$ is defined as: $P_D(i) = \frac{d_i(i)}{n-1}$.

Degree prestige is dual to the degree closeness of an actor: the former uses the incoming links, the latter — the outgoing links.

Proximity Prestige

This measure generalizes the degree prestige by considering not only the actors directly adjacent to actor i.

An **influence domain** of an actor i, denoted I_i is the set of actors which can reach i through the social network G.

The average distance from an actor $j \in I_i$ to i is computed as

$$d(\bar{j}, i) = \frac{sum_{j\in I_i} d(j,i)}{|I_i|}.$$

The **proximity prestige** of an actor i, denoted $P_P(i)$ is defined as

$$P_P(i) = \frac{\frac{|I_i|}{n-1}}{\frac{\sum_{j\in I_i} d(j,i)}{|I_i|}} = \frac{|I_i^2|}{(n-1)\sum_{j\in i} d(j,i)}.$$

Here, $\frac{|I_i|}{n-1}$ is the fraction of actors in I that can reach i.

$P_P(i)$ ranges from 0 to 1.

Rank Prestige

All previous measures assume that all other actors have the same influence on prestige of a given actor.

Rank Prestige is a way of computing prestige of an actor via the prestige of its neighbors. More specifically, **rank prestige** of an actor i, denoted $P_R(i)$, is defined as

$$P_R(i) = \sum_{j\in I} A_{ij} \cdot P_R(j).$$

Here A_{ij} is a matrix representing the edges of the graph G: $A_{ij} = 1$ if $(i,j) \in E$, and $A_{ij} = 0$ otherwise (i.e., the adjacency matrix).

Note, this is a **recursive formula**.

Let $\mathbf{P} = (P_R(i_1), \ldots, P_R(i_n))$ be the vector of all rank prestige values. Then,

$$\mathbf{P} = A^T\mathbf{P}.$$

Solutions to this equation are **eigenvectors** of A.

Rank prestige is used in both **PageRank** and **HITS** algorithms.

Co-citation and Bibliographic Coupling

Citation analysis considers the social network graph where nodes are **academic papers** or **other information sources**, and links are **references** between them. (note, that the WWW graph falls under this definition.)

Co-citation

Definition. Paper k **co-cites** papers i and j if the citation graph G has (k, i) and (k, j) links.

Co-citation as a similarity measure. If two papers are **co-cited** on numerous occasions, they are probably similar. The degree of similarity between papers i and j can be quantified as the **co-citation coefficient**, denoted C_{ij} and computed as follows:

$$C_{ij} = \sum_{k \in I} A_{ki} A_{kj}.$$

Bibliographic Coupling

Bibliographic coupling is the dual notion to that of co-citation.

Definition. Papers i and j are **bibliographically coupled** if there is a third paper k that is cited in both i and j: i.e., if (i, k) and (j, k) are links in G.

Bibliographic coupling as a similarity measure. The more papers two papers i and j cite, the more likely they are to be similar to each other. This can be quantified as a **bibliographic coupling coefficient**, denoted B_{ij} in the following manner:

$$B_{ij} = \sum_{k \in I} A_{ik} A_{jk}.$$

PageRank

See separate handout.

Hypertext Induced Topic Search (HITS)

HITS, or Hypertext Induced Topic Search algorithm is an Information Retrieval algorithm[1] that given a query retrieves and ranks a number of web pages that are supposed to contain information about the query.

HITS consists of three parts:

1. **Information Retrieval.** HITS outsources the IR business. Given a query q, it passes it to a proper web search engine and obtains a number of top pages retrieved by the search engine.
2. **Growth.** HITS adds a number of other pages, by harvesting links from the pages retrieved on stage 1.
3. **Ranking.** HITS ranks the expanded set of pages by computing two scores for each page:
   - Authority score: a measure of the *prestige* of the page;
   - Hub score: a measure of the *centrality* of the page.

   The algorithm then returns a list of pages with the highest authority score and a separate list of pages with the highest hub score.

More formally:

1. **Step 1. Information Retrieval.**
   (a) Given an input query q, pass it to a web search engine.
   (b) Retrieve the list W of 200 top pages found by the web search engine.
   We refer to W as the **root set** for query q
2. **Step 2. Growth.**
   (a) For each $w \in W$, add up to k (usually $k = 50$) pages that w links to.
   (b) The new set of pages, consisting of W and all newly added pages is denoted S and referred to as the **base set** for query q.
3. **Step 3. Ranking.**
   (a) Construct a directed graph $G = \langle S, E \rangle$ out of nodes in the base set. E contains all the links between the pages in S. Denote as L the adjacency matrix for G.
   (b) Compute the **authority** and **hub** scores (denoted $a(i)$ and $h(i)$) of each page $i \in S$ using the following recursive relations:

$$a(i) = \sum_{(j,i) \in E} h(j)$$
$$h(i) = \sum_{(i,j) \in E} a(i)$$

[1] It is an IR algorithm by virtue of what its input and output are. However, it "our-sources" the actual retrieval of information.

```
Algorithm HITS-Iterate(G)
begin
  a_0 = (1, 1, ..., 1);
  h_0 = (1, 1, ..., 1);
  k := 1;
  repeat
     a_k := L^T L a_{k-1};
     h_k := L L^T h_{k-1};
     a_k := a_k / ||a_k||;
     h_k := h_k / ||h_k||;
 until ||a_k - a_{k-1}|| < ε_1  and  ||h_k - h_{k-1}|| < ε_2;
  return  a_k, h_k;
end
```

Figure 1: Power iteration version of the ranking part of the HITS algorithm

From Recurrence to Iteration

The authority and hub scores form a mutually recursive relationship. Let $S = \{i_1, \ldots i_n\}$. Let $\mathbf{a} = (a(i_1), \ldots, a(i_n))$ and $\mathbf{h} = (h(i_1), \ldots h(i_n)$.

We can connect these two vectors as follows:

$$\mathbf{a} = L^T\mathbf{h}$$

$$\mathbf{h} = L\mathbf{a}$$

Using substitution, we can obtain the following self-recurrence relations:

$$\mathbf{a} = L^TL\mathbf{a}$$

$$\mathbf{h} = LL^T\mathbf{h}$$

These, in turn, may be transformed into an iterative procedures:

$$\mathbf{a_k} = L^TL\mathbf{a_{k-1}}$$

$$\mathbf{h_k} = LL^T\mathbf{h_{k-1}}$$

Traditionally, $\mathbf{a_0} = \mathbf{h_0} = (1, 1, \ldots 1)$.

Figure 1 shows the pseudocode of the iterative part of the HITS algorithm which uses normalization on each step to guarantee convergence. Normalization ensures that on each step

$$\sum_{i \in S} a(i) = 1$$

$$\sum_{i \in S} h(i) = 1$$