Link Analysis in Graphs: PageRank

Link Analysis

Graphs

Recall definitions from Discrete math and graph theory.

Graph. A **graph** G is a structure $\langle V, E \rangle$, where

- $V = \{v_1, \ldots, v_n\}$ is a finite set of vertices or nodes;
- $E = \{(v, w) | v, w \in V\}$, is a set of *pairs of vertices* called edges.

Undirected and directed graph. In a **directed graph**, an edge $e = (v, w)$ is interpreted as a *connection from v to w* but **not** a *connection from w to v*.

In an **undirected graph**, an edge $e = (v, w)$ is interpreted as a connection **between** v and w.

Representations. Graphs can be represented in a number of ways:

- **Set notation.** A representation of a graph that follows the definition above. **Example.** $G = \langle \{A, B, C, D, E\}, \{(A, B), (A, C), (A, E), (B, C), (B, E), (C, D)\} \rangle$.
- **Graphical representation.** A graph can be represented as a drawing. Each **node** is drawn as a *point* or *circle* on a plane, and each **edge** is a *line* connecting the representations of its two vertices. To draw a **directed graph**, **arrows** are added to the edge lines to point from the first vertex in the edge to the second.

  Example. Figure 1 shows the graphical representations of G in the cases when G is directed and undirected.

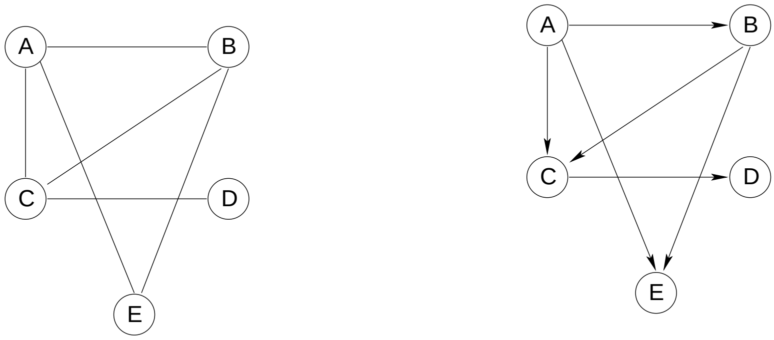

Figure 1: Undirected (left) and directed (right) graphs.

- **Matrix.** A graph can be represented as an **adjacency matrix** M_G whose rows and columns are vertices. If edge $(v_i, v_j) \in E$, $M_G[i, j] = 1$, otherwise, $M_G[i, j] = 0$. Undirected graphs have symmetrical adjacency matrices (or, alternatively, only uppre diagonal portions of those matrices are considered). Matrices for directed graphs need not be symmetric.

  Example. The adjacency matrices for graph G in undirected and directed cases:

  Undirected G:

G	A	B	C	D	E
A	—	1	1	0	1
B	1	—	1	0	1
C	1	1	—	1	0
D	0	0	1	—	0
E	1	1	0	0	—

  Directed G:

G	A	B	C	D	E
A	—	1	1	0	1
B	0	—	1	0	1
C	0	0	—	1	0
D	0	0	0	—	0
E	0	0	0	0	—

- **Lists.** A graph can be represented by an associative array of **adjacency lists**. The domain of the array A_G is V. For $v \in V$, $A_G[v]$ lists all $w \in V$, such that $(v, w) \in E$.

  Example. The adjacency lists for the undirected and directed versions of graph G are shown below:

  Undirected G:

A:	B,C,E
B:	A,C,E
C:	A,B,D
D	C
E	A,B

  Directed G:

A:	B,C,E
B:	C,E
C:	D
D	
E	

Labeled Graphs. A **labeled graph** G is a graph $G = \langle V, E\rangle$, where $E = \{(v, w, l)\}$, where $v, w \in V$ are vertices connected by the edge and l is a *label*. The domain for the set of possible labels is usually specified up-front.

Egde labels can be used to specify the *length of a connection*, *cost to traverse the edge*, *type on edge* and many other properites.

Graphs can have additional *edge* and *vertex* labels.

Properties of Graphs.

Path. A path in a graph $G = \langle V, E\rangle$ is a sequence $p = e_1, e_2, \ldots e_s$ of edges, $e_1 = (w_1, w'_1), \ldots, e_s = (w_s, w'_s)$, such that $w'_1 = w_2, w'_2 = w_3, \ldots w'_{s-1} = w_s$. In *undirected graphs* p is called a *path between* w_1 *and* w'_w. In *directed graphs* p is called a *path* ***from*** w_1 ***to*** w'_s.

Connected Graphs. A graph $G = \langle E, G\rangle$ is called **connected** iff for any pair $v_i, v_j \in V$, there exists a path p between v_i and v_j (or, from v_i to v_j).

Shortest path. The **length** of a path p in a graph G is *the number of edges in it*.

A **shortest path** between two vertices v and w is a path that starts in v and ends in w with the smallest length (number of edges in it).

Complete graphs. A graph $G = \langle V, E\rangle$ is **complete** iff for all vertices $v, w \in V$, $(v, w) \in E$.

Vertex degrees. Let $G = \langle V, E\rangle$ be an *undirected* graph. The **degree of a node** $v \in V$ in G is defined as

$$degree(v) = |\{(v, v') \in E | v' \in V\}|,$$

i.e., it is the *number of edges that connect* v *to other vertices in the graph.*

Let $G = \langle V, E\rangle$ be a *directed* graph. The **in-degree** of a node $v \in V$ in G is defined as

$$in - degree(v) = |\{(v', v) \in E | v' \in V\}|,$$

i.e. the *number of edges in G that end at* v.
The **out-degree** of a node $v \in V$ in G is defined as

$$out - degree(v) = |\{(v, v') \in E | v' \in V\}|,$$

i.e., it is the *number of edges that start in* v.

Graphs and Social Networks

Social entity. A **social entity** is a *community*, *organization* or *setting* involving a collection of *interacting actors*.

Actors. In different social entities actors may be:

- Humans. (e.g., employees in a company).
- Groups of humans (e.g., sports teams).
- Legal or political entities (e.g., companies or states).
- Inanimate ojects (e.g., individual computers).
- Virtual objects (e.g., web pages or files).

Social Network. A **social network** of a *social entity* is a structure documenting *interactions between actors within the entity*.

Typically, social networks are represented as graphs. A **social network graph** $SN = \langle V, E \rangle$ is constructed as follows:

- V is the set of **actors of the social entity**.
- E is the set of **interactions** between the actors in the entity. I.e., $(v, w) \in E$ iff, actors v and w have an interaction that is tracked by the social network.

Interactions can be **symmetric**, in which case SN is an undirected graph, or **assymetric**, in which case SN is a directed graph.

Examples. Examples of social networks:

- Business interactions. Email exchanges between employees of a company.
- Social interactions. "Friendship" relationship on facebook.
- Academic interactions. Citation of a paper by another paper. Co-authoring of papers by researchers.
- Relationship interactions. Kinship relationships between people.
- Kevin Bacon game. Actors having roles in the same movie.
- Web page interactions. Links from one web page to another.

PageRank via Web Traversal

Web Search specifics. Compared to "traditional" Information Retrieval, ***web search*** has the following properties:

- Huge document collection. (world wide web is the biggest document collection).
- No "golden set". Web is unobservable, hence, we cannot find the sets of all relevant documents for the queries.
- Only few links visited. Only the top 20-40-100 links are of any importance. Users rarely venture beyond in search of relevant web pages.
- **Web pages are linked!** Can this be used to improve search?
- Web page owners are not trustworthy. *Search engine spamming* and (somewhat less horrible) *search engine optimization* attempt to circumvent the results of web search on certain queries.

Prestige. Idea: a *good web search engine* must combine discovery of pages that contain all/most query terms with **robust ranking**, which **promotes important, high-quality, reliable pages** to the top. **Prestige** is a measure of **web page importance**.

PageRank. PageRank is a procedure for computing the **prestige** of each web pages in a collection.

There is a number of definitions/derivations for the PageRank computation. To illustrate how it works, we will use the more simple definition.

Web as a graph. We treat World Wide Web as a **social network**, where **individual web pages** (urls) are nodes, or actors, and **hypertext links** between them are interactions.

More formally, consider the directed acyclic graph $G_{WWW} = \{V, E\}$. The set V of vertices is the list of individual web pages (urls). An edge $(v, w) \in E$ iff the web page v has in its body an *anchor tag* `<a href="URL">` where `URL` is the URL of the web page w.

Given a web page $i \in V$, The set $I(i)$ of **in-links** is the set of all edges $e \in E$, such that $e = (v, i)$ for some $v \in V$.

Given a web page $i \in V$, the set $O(i)$ of **out-links** is the set of all edges $e \in E$, such that $e = (i, v)$ for some $v \in V$.

Note: often, only the in-links and out-links from web pages located on a ***different site*** *are included in I(i) and O(i).*

Surfing the web. PageRank is a way of modeling the behavior of a web surfer in a single browser window. In particular, PageRank models the following traversal:

PageRank Traversal:

1. The user starts surfing the web from some, *randomly selected page* from V^a.
2. On each step, the user observes some web page i. *With probability* $d \in (0, 1)$ (s)he chooses to click on any of the links available on the page (assuming the page has at least one out-link).
3. Each link found on the page i *can be selected with the same probability.*
4. *With probability* $1 - d$ the user gets tired of surfing the web by following links and instead goes directly *to a randomly selected web page from the collection* V.
5. If a web page has no out-links, the user simply goes *to a randomly selected web page from the collection* V.

[a] PageRank actually allows to relax this condition and start from some page, randomly selected from a predefined collection of pages: a (typically small) subset of the entire web page collection.

PageRank defined. The **PageRank** of a page $i \in V$ is the **probability of eventually reaching page** i **via the traversal procedure** outlined above[1].

Deriving PageRank

Let $p(i)$ be the *probability of reaching web page* i (i.e., the PageRank of page i). Let $I(i) = \{j_1, \ldots j_s\}$ be the set of all web pages which link **to** i. Let the probabilities of reaching each of those pages be $p(j_1), \ldots, p(j_s)$ respectively. Also, let $O(j_k)$ be the set of *all outbound edges from* j_k.

Assumption: All web pages in V have at least one out-link.

- Suppose we have reached page j_1. From that page, with probability d, we elect to follow on of the links. j_1 has $|O(j_1)|$ out-links on it, so, **with probability**
$$p(i|j_1, \text{follow links}) = \frac{1}{|O(j_1)|}$$
we can reach web page i. Since $p(\text{follow links}) = d$, we obtain:
$$p(i|j_1) = d \cdot \frac{1}{|O(j_1)|}.$$

- Similar reasoning for all other $j \in I(i)$ yields
$$p(i|j_k) = d \cdot \frac{1}{|O(j_k)|}.$$

- We can reach page i in one of only two ways:
  1. By following a link from one of $j_1, \ldots, j_s$.
  2. By randomly selecting i when the user chooses to jump (i.e. not follow a link from a current page).

- We obtain the following formula for computing the probability $p(i)$:
$$p(i) = (1-d) \cdot \frac{1}{|V|} + (p(i|j_1) \cdot p(j_1) + \ldots + p(i|j_s) \cdot p(j_s)).$$
Figure 2 illustrates how these probabilities are computed. From here, substituting $p(i|j_k)$ we obtain:
$$p(i) = (1-d) \cdot \frac{1}{|V|} + d \cdot \sum_{k=1}^{s} \frac{1}{|O_{j_k}|} \cdot p(j_k).$$

Thus,

$$pageRank(i) = (1-d) \cdot \frac{1}{|V|} + d \cdot \sum_{k=1}^{s} \frac{1}{|O_{j_k}|} \cdot pageRank(j_k). \tag{1}$$

Note, that this is a **recursive definition.**

Computing PageRank

From formula (1), we see that in order to compute PageRank of a page, we need to know the PageRank of its ”ancestors”. A standard way to model such computation is to perform it iteratively.

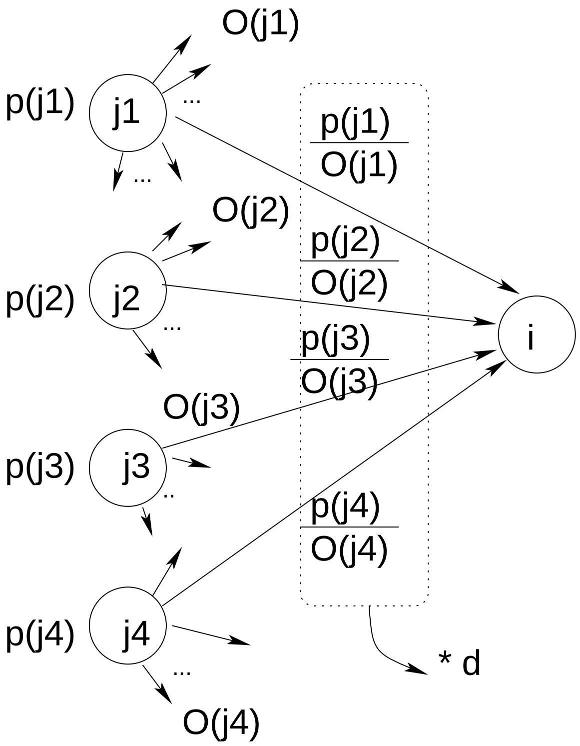

Figure 2: Computing the probability of reaching a web page.

PageRank via iterative process. The traditional iterative algorithm for PageRank uses the following iterative procedure:

$$pageRank^0(i) = \frac{1}{|V|} \quad \text{for all} i \in V \tag{2}$$

$$pageRank^r(i) = (1-d) \cdot \frac{1}{|V|} + d \cdot \sum_{k=1}^{s} \frac{1}{|O_{j_k}|} \cdot pageRank^{r-1}(j_k). \tag{3}$$

$$\textbf{Stop when}: \left(\sum_{i \in V} (pageRank^r(i) - pageRank^{r-1}(i)) \right) < \epsilon \tag{4}$$

References

[1] Page, Lawrence; Brin, Sergey; Motwani, Rajeev and Winograd, Terry (1998). The PageRank citation ranking: Bringing order to the Web. *Technical Report, Department of Computer Science, Stanford University*. http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf