Computing for Data Sciences

Lecture 18 – 19

GRAPHS

RANDOM GRAPHS AND ITS REPRESENTATION

A graph is an ordered pair G = (V, E) comprising of:

1. A set V of vertices, nodes or points where V = {v1,….,vn}
2. A set E of edges, arcs or lines where $E \subseteq V^2$ which are 2-element subsets of V (i.e., an edge is related with two vertices, and the relation is represented as an unordered pair of the vertices with respect to the particular edge).

The vertices belonging to an edge are called the ends or end vertices of the edge. A vertex may exist in a graph and not belong to an edge. V and E are usually taken to be finite. The order of a graph is |V|, its number of vertices. The size of a graph is |E|, its number of edges. Graphs are used to represent many real life applications: Graphs are used to represent networks. The networks may include paths in a city or telephone network or circuit network. Graphs are also used in social networks like LinkedIn, Facebook. For example, in Facebook, each person is represented with a vertex (or node). Each node is a structure and contains information like person id, name, gender and locale.

In modern texts in graph theory, unless stated otherwise, graph means "undirected simple finite graph".

- **Undirected graph:** An undirected graph is a graph in which edges have no orientation. The edge (x, y) is identical to the edge (y, x), i.e., they are not ordered pairs. The maximum number of edges in an undirected graph without a loop is n(n − 1)/2.

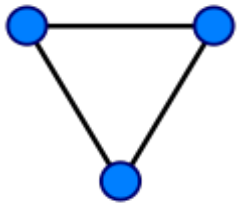

Figure 1: A simple undirected graph with three vertices and three edges.

- The diagram below is a graphical **Vertice Edge representation** of the following graph:

  V = {1, 2, 3, 4, 5, 6};

  E = {{1, 2}, {1, 5}, {2, 3}, {2, 5}, {3, 4}, {4, 5}, {4, 6}}.

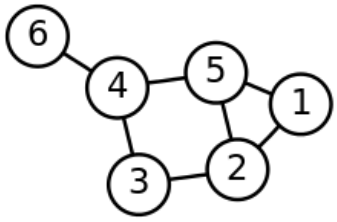

Figure 2: A graph with six nodes

Another common representation is an **Adjacency matrix**, which is a two-dimensional array, where A_{ij} is non-zero when there is an edge $(v_i, v_j) \in E$. An adjacency matrix is a means of representing which vertices (or nodes) of a graph are adjacent to which other vertices. With an adjacency matrix, we can find out whether an edge is present in constant time, by just looking up the corresponding entry in the matrix. For example, if the adjacency matrix is named graph, then we can query whether edge (i,j) is in the graph by looking at graph[i][j].

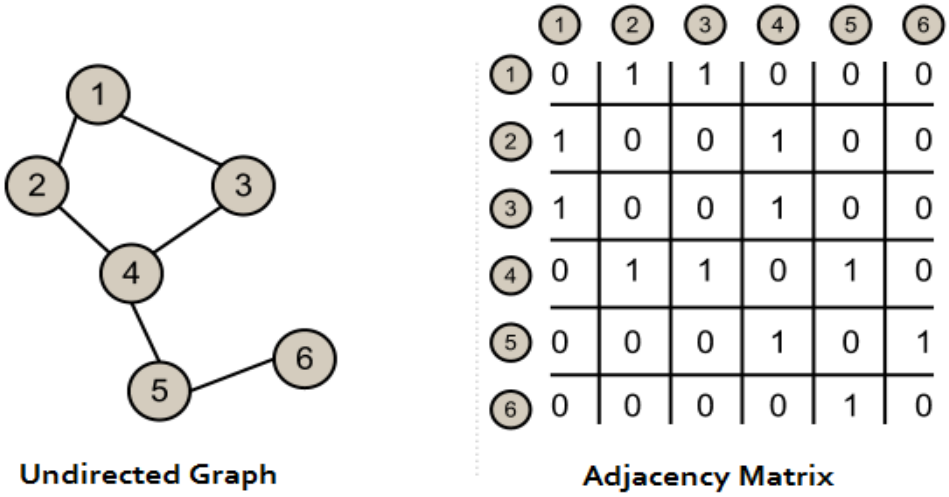

Figure 3: Adjacency Matrix Representation of the undirected graph

So what's the disadvantage of an adjacency matrix?

First, it takes $\Theta(V^2)$ space, even if the graph is sparse: relatively few edges. In other words, many graphs are sparse in the sense that most of the possible edges between pairs of vertices do not exist.

Second, if you want to find out which vertices are adjacent to a given vertex *i*, you have to look at all |V| entries in row *i*, even if only a small number of vertices are adjacent to vertex *i*. A symmetric sparse matrix arises as the adjacency matrix of an undirected graph. In such cases, a good way to store a sparse matrix is by using Vertical Edge Representation.

PAGE RANK

PageRank is a function that assigns a real number to each page in the Web. PageRank is an algorithm used by Google Search to rank websites in their search engine results. PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

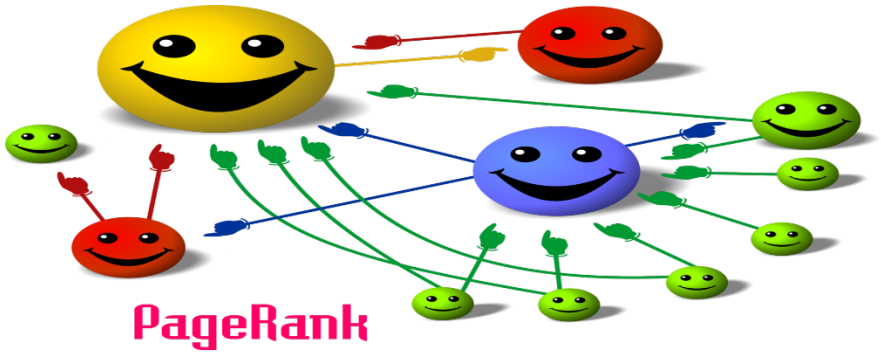

Figure 4: The size of each face is proportional to the total size of the other faces which are pointing to it.

THE RANDOM SURFER MODEL

The random surfer visits a web page with a certain probability which derives from the page's PageRank. The probability that the random surfer clicks on one link is solely given by the number of links on that page. This is why one page's PageRank is not completely passed on to a page it links to, but is divided by the number of links on the page.

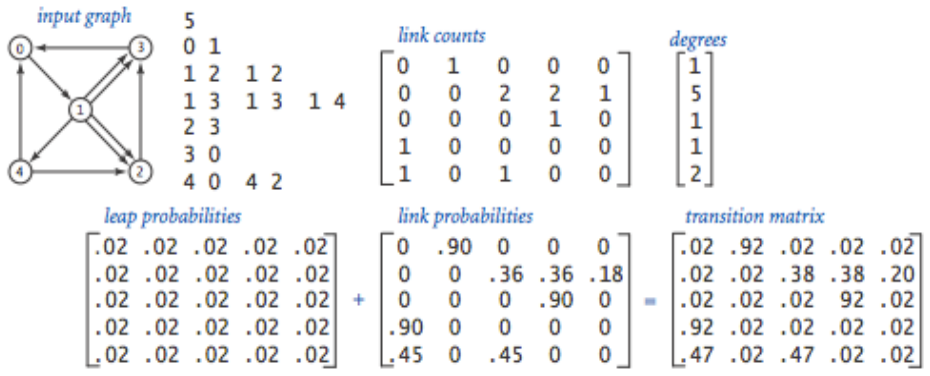

Figure 5: Transition Matrix Calculation

Transition matrix is M.

Suppose we start a random surfer at any of the n pages of the Web with equal probability. Then the initial vector vo will have 1/n for each component. If M is the transition matrix of the Web, then after one step, the distribution of the surfer will be MV_0, after two steps it will be $M(MV_0) = M^2V_0$, and so on.

It is known that the distribution of the surfer approaches a limiting distribution v that satisfies v = Mv, provided two conditions are met:

1. The graph is strongly connected; that is, it is possible to get from any node to any other node.
2. There are no dead ends: nodes that have no arcs out.

STRUCTURES IN THE WEB:

There was a large strongly connected component (SCC), but there were several other portions that were almost as large.

1. The in-component, consisting of pages that could reach the SCC by following links, but were not reachable from the SCC.
2. The out-component, consisting of pages reachable from the SCC but unable to reach the SCC.
3. Tendrils, which are of two types. Some tendrils consist of pages reachable from the in-component but not able to reach the in-component. The other tendrils can reach the out-component, but are not reachable from the out-component.

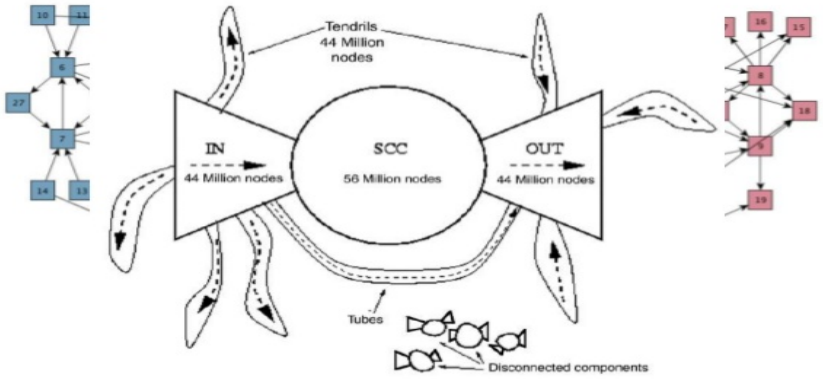

Figure 6: "Bow-Tie" Model of the web

In addition, there were small numbers of pages found either in:

a) Tubes, which are pages reachable from the in-component and able to reach the out-component, but unable to reach the SCC or be reached from the SCC.
b) Isolated components that are unreachable from the large components (the SCC, in- and out-components) and unable to reach those components.

DEAD ENDS:

If we allow dead ends, the transition matrix of the Web is no longer stochastic, since some of the columns will sum to 0 rather than 1. A matrix whose column sums are at most 1 is called substochastic. If we compute M_iV for increasing powers of a substochastic matrix M, then some or all of the components of the vector go to 0. That is, importance "drains out" of the Web, and we get no information about the relative importance of pages.

SPIDER TRAPS:

A spider trap is a set of nodes with no dead ends but no arcs out. These structures can appear intentionally or unintentionally on the Web, and they cause the PageRank calculation to place all the PageRank within the spider traps. We modify the calculation of PageRank by allowing each random surfer a small probability of teleporting to a random page, rather than following an out-link from their current page. The iterative step, where we compute a new vector estimate of PageRanks v' from the current PageRank estimate v and the transition matrix M is

$v' = \beta Mv + (1 - \beta)e/n$

Where β is a chosen constant, usually in the range 0.8 to 0.9, e is a vector of all 1's with the appropriate number of components, and n is the number of nodes in the Web graph. The term βMv represents the case where, with probability β, the random surfer decides to follow an out-link from their present page. The term $(1-\beta)e/n$ is a vector each of whose components has value $(1-\beta)/n$ and represents the introduction, with probability $1 - \beta$, of a new random surfer at a random page.