In-link count

If many pages point to page P, then P is presumably a good page.

However, some links are more important than others:

1. Links from a different domain count more than links from same domain.
2. Nature of anchor (e.g. font, boldface, position?) may indicate judgment of relative importance.
3. A link from an important page is more important than a link from an unimportant page. This is circular, but the circularity can be resolved:

Page Rank Algorithm

Suppose that each page P has an importance I(P) computed as follows: First, every page has an inherent importance E (a constant) just because it is a web page. Second, if page P has importance I(P) then P contributes an indirect importance F*I(P) that is shared among the pages that P points to. (F is another constant). That is, let $O_P$ be the number of outlinks from P. Then if there is a link from P to Q, P contributes $F*I(P)/O_P$ “units of importance” to Q.

(What happens if P has no outlinks, so that $O_P = 0$? This actually turns out to create trouble for our model. For the time being, we will assume that every page has at least one outlink, and we will return to this problem below.)

We therefore have the following equation: for every page Q, if Q has in-links from $P_1$ ... $P_m$,

$$I(Q) = E + F \sum \frac{I(P)}{O_{P_i}}$$

This looks circular, but it is just a set of linear equations in the quantities I(P). Let N be the total number of pages on the Web (or in our index.) We have N equations (one for each value of Q on the left) in N unknowns (the values I(P) on the right), so that, at least looks promising.

We now make the following observation. Suppose we write down all the above equations for all the different pages Q on the Web. Now we add up all the left hand sides and all right hand sides. On the left we have the sum of I(Q) over all Q on the web; call this sum S. On the right we have N occurrences of E and for every page P, $O_P$ occurrences of $F*I(P)/O_P$. Therefore, over all the equations, we have for every page P a total of $F*I(P)$, and these add up to $F*S$. (Note the importance here of our assumption that $O_P > 0$.) Therefore, we have

$$S = NE + FS \text{ so } F = 1 - NE/S.$$  (2) 

Since the quantities E,F,N,S are all positive, it follows that $F < 1, E < S/N$.

For example suppose we have pages U,V,W,X,Y,Z linked as shown in the figure

Note that X and Y each have three in-links but two of these are from the "unimportant" pages U and W, Z has two in-links and V has one, but these are from "important" pages. Let E = 0.05 and let F=0.7. We get the following equations (for simplicity, I use the page name rather than I(page name)):
U = 0.05
V = 0.05 + 0.7 * Z
W = 0.05
X = 0.05 + 0.7*(U/2+V/2+W/2)
Y = 0.05 + 0.7*(U/2+V/2+W/2)
Z = 0.05 + 0.7*(X+Y)

The solution to these is U = 0.05, V = 0.256, W = 0.05, X = 0.175, Y = 0.175, Z = 0.295

**Markov model**

A useful way to think about the above model is as follows: Imagine someone who is browsing the web for a very long time. Each time he reads a page P, he decides where to go next using the following procedure:

1. Flip a weighted coin, that comes up heads with probability e and tails with probability (1-e).
2. If coin was heads, he picks a page at random in the Web and goes there.
3. If coin was tails, he picks an outlink from P at random and follows it. (Again, we assume that every page has at least one outlink.)

The browser does this for eons (posit that the web stays constant all that time) and we keep track of where he has been. At the end of this we state that the importance of each page is the fraction of the time that he has spent on that page while browsing. It is easy to show that "importance", thus defined, satisfies the equations above, where E=e/N and F=(1-e). Or, equivalently, we can ask "What is the probability that after an eon of browsing, the browser is on page P," and that probability is equal to the "importance".
Computing importance

Equation (1) can be used iteratively to converge on the steady state for the importance \( I(P) \):

**Algorithm PageRank:**

\[
\text{for (all pages } P) \quad I[P] \leftarrow 1/N \quad /* \text{start with a uniform distribution } */ \\
\text{repeat } \{ \text{for (all pages } P) \quad J[P] \leftarrow E + F \ast \sum Q \text{ links to } P \ I[Q]/O_Q; \ \\
\quad \text{if } |\vec{J} - \vec{I}| < \epsilon \text{ then exitloop; } \\
\quad I \leftarrow J \}
\]

return \( \vec{J} \)

**Dependence on \( e \) (the probability of “tails”)**

For \( e = 1 \), just a uniform distribution (uniformly random transition at each step).

For \( e \) close to 1, the distribution is nearly uniform. The order of Page Rank is just number of in-links.

At \( e = 0 \), things go kerflooey. There still exists a stable distribution, but there may be more than one, and the system may not converge to it. The above algorithms may not converge.

At \( e \) close to 0, the system is unstable (i.e. small changes to structure make large change to solution.) The above algorithms converge only slowly.

Experiments of Page et al. used \( e=0.15 \).

**Pages with no outlinks**

**Solution 1:** Say that every page has a self-loop (an outlink to itself.) The problem is that this rewards pages with no outlinks.

**Solution 2:** (from Page et al.)

Step 1: Prune all pages with no outlinks, then prune all pages which had only onlinks to the pages you just pruned, and keep on doing this until all pages have an outlink.

Step 2: Compute PageRank over this set;

Step 3: Reinstate the pages you pruned and let importance propagate forward without changing the values of the importances calculated in (2).

**Solution 3:** (From Langville and Meyer) If there are no outlinks, you jump randomly (i.e. with \( 1/N \) probability to each known page.) Algorithm 1 can now be modified as shown:

**Algorithm PageRank2:**

\[
N \leftarrow \text{total number of pages;} \\
\text{for (all pages } P) \quad I[P] \leftarrow 1/N \quad /* \text{start with a uniform distribution } */ \\
\text{repeat } \{ \\
\quad Z \leftarrow \sum Q|Q \text{ has no outlinks} \quad I[Q]; \\
\quad \text{for (all pages } P) \quad J[P] \leftarrow E + Z/N + F \ast \sum Q \text{ links to } P \ I[Q]/O_Q \\
\quad \text{if } |\vec{J} - \vec{I}| < \epsilon \text{ then exitloop; } \\
\quad I \leftarrow J \\
\} \\
\text{return } \vec{J}
\]
Non-uniform “source of importance”

It is not necessary for the “inherent” importance of all pages \( P \) to be the same value \( E \); one can have any distribution \( E(P) \), representing some other evaluation of inherent importance. For example, one could have \( E(P) \) ranked higher for pages on a .edu site, or for the Yahoo home page, or for your own page, etc. The algorithms above work exactly the same; just change \( E \) to \( E(P) \).

Non-uniform outlinks

Or there may be a reason to think that some outlinks are better than others (e.g. font, font size, links to a different domain are more important than links within a domain.) You can assign \( W(Q,P) \) be the weight of the link from \( Q \) to \( P \) however you want; the only constraints are that the weights are non-negative and that the weights of the outlinks out of \( Q \) add up to 1. Then just replace \( 1/O_Q \) in the algorithm by \( W(Q,P) \).

HITS model (Kleinberg)

Observation: The Web is not actually important pages pointing to other important pages. Rather, the good pages in any domain tend to be either authorities with good content, or hubs with good links. A hub has a lot of outlinks to authorities; an authority has a lot of in-links from hubs. Authorities tend not to point to authorities, either because they are in competition, or because that’s not what they’re concerned with. We can perhaps improve importance evaluation by taking this structure into account.

Also, we can try to use this to solve the problem of “non-self-descriptive” important Web pages. For example, one would want the query “Japanese auto manufacturer” to return the home pages for the major Japanese auto manufacturer, but in fact as of 9/17/07 the top 100 results returned by Google does not contain a single such home page, and only one page (#32) that is in the web site of a Japanese auto manufacturer. However, if we can find good hubs with those words, they probably tend to point to these home pages.

HITS Markov Model

\[
C \leftarrow \text{top K pages for query, as returned by search engine} \\
\text{/* K is a fixed parameter. */} \\
O \leftarrow \text{follow all outlinks from C.} \\
I \leftarrow \text{follow at most D inlinks from C; /* D is another fixed parameter.} \\
\text{There is an inherent limitations on the number of outlinks from any page,} \\
\text{but there are pages with vast numbers of inlinks. */} \\
PAGES \leftarrow C \cup O \cup I; \\
M \leftarrow \text{the following Markov model on PAGES:} \\
\text{\{} \\
\text{TURN } \leftarrow \text{TRUE; } \\
P \leftarrow \text{arbitrary starting point; } \\
\text{repeat } \{ \\
\text{randomChoice } \{ \\
\text{with probability e: } \{ \\
P \leftarrow \text{random choice in PAGES; } \\
\text{TURN } \leftarrow \text{randomly either TRUE or FALSE; } \} \\
\text{with probability } (1-e): \{ \\
\}
\} 
\]
if (TURN)
    then { choose a random outlink from P to Q;
            P ← Q;
            mark this as an “authority” visit to P;
        }
    else { choose a random inlink into P from Q;
            P ← Q;
            mark this as an “hub” visit to P;
        }  
    TURN ← not TURN;
}  

Find the stable distribution for this Markov model as in previous section. The “hub” value of page P is then the frequency with which P is labelled a hub and its “authority” value is the frequency with which it is labelled an authority.

Thus, in this Markov model, when you are not jumping randomly, you are alternating moving across an outlink (presumably from a hub to an authority) and moving across an inlink (presumably from an authority to a hub.) You thus have two vectors L(P), the probability that a random hub visit is P and M(P), the probability that a random authority visit is P. Let A be the transition matrix when TURN = TRUE and let B be the transition matrix when TURN = FALSE; it is easily shown that B = transpose of A. Then M = AL and L=BM so M = (AB)M. and you solve this equation in the same way as PageRank equations.

(This is actually the rational reconstruction of the model due to Ng et al., which they show to be more stable than Kleinberg’s original proposal. There are a number of alternative formulations in the literature.)

Problem: Topic drift. Algorithm drifts toward tightly connected regions of Web, which may not be exactly what was asked about. Example: Search on “jaguar” converged to a collection of sites about Cincinatti. Explanation: Large number of articles in the Cincinatti Enquirer about the Jacksonville Jaguars football team, all link to same Cincinatti Enquirer service pages. Various partial solution studied, none wholly successful. (See Borodin et al. http://www10.org/cdrom/papers/314/ for discussion, examples)