Math 2270 - Computer Lab 5 : Web Search

Dylan Zwick

Fall 2012

In today's lab we're going to explore the mathematics of search engines, and how they relate to calculating eigenvalues and eigenvectors of very, very large matrices.

The Math

Suppose B is an $n \times n$ matrix with real eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_n$ and corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$. For any $\mathbf{w} \in \mathbb{R}^n$ we may write

$$\mathbf{w} = c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n.$$

Then

$$B^k\mathbf{w} = c_1\lambda_1^k\mathbf{v}_1 + \cdots + c_n\lambda_n^k\mathbf{v}_n.$$

As k gets large, λ_1^k becomes much larger than the other λ_i^k so, assuming $c_1 \neq 0$,

$$B^k\mathbf{w} \approx c_1\lambda_1^k\mathbf{v}_1.$$

In other words, the orbit of $\mathbf{w}$, which is the sequence $\mathbf{w}, B\mathbf{w}, B^2\mathbf{w}, B^3\mathbf{w}, \ldots$ approaches multiples of the leading eigenvector $\mathbf{v}_1$. Conversely, we can find $\mathbf{v}_1$ by following an orbit $\mathbf{w}, B\mathbf{w}, B^2\mathbf{w}, \ldots$. Let $\mathbf{v}_1$ by a vector pointing in the direction that the orbit is tending. If n is large this approach is easier than trying to compute roots of the characteristic polynomial of B.

In section 6.7 this approach is used to model an internet search engine. Let $s_1, \ldots, s_n$ be a list of websites. Let $L = (l_{ij})$ be the matrix that records links between the websites: $l_{ij} = 1$ if site s_i links to site s_j, and $l_{ij} = 0$ otherwise. (Note that the ij entry of L^T is 1 if s_j links to s_i.)

There are two ways that a website can be important, and therefore worthy of a top listing in our search engine. A site can be an *authority*, a site to which many other sites link, or it can be a *hub*, a site that has links to many other sites. However, in each case it is not just the number, but also the quality of the links that determines how important a site is. We use an iterative method to determine the site rankings.

Let a_i by the ranking of site s_i as an authority, and let h_i be the ranking of site s_i as a hub. We collect these into n-dimensional vectors $\mathbf{a} = (a_i)$ and $\mathbf{h} = (h_i)$. Approximate these rankings as follows. For a first approximation let a_i^0 be the number of sites that link to s_i, and let h_i^0 be the number of sites to which s_i links. We can compute these with the matrix L. The number of sites to which s_i links is the sum of row i of L, which can be found by taking the dot product of row i with the vector $\mathbf{1}$ consisting of all 1's. Thus, $\mathbf{h}^0 = L\mathbf{1}$. Similarly, $\mathbf{a}^0 = L^T\mathbf{1}$, which counts incoming links.

The initial authority ranking of the site s_i is given by the i-th component of $\mathbf{a}^0$. However, this ranking only counts links, but does not consider their quality. A site is more of an authority if the incoming links are from important hubs, rather than from random sites. We improve the authority rankings by taking into account the hub ranking $\mathbf{h}^0$. Instead of adding up all the incoming links to s_i we weigh them by $\mathbf{h}^0$. Let $\mathbf{a}^1 = L^T\mathbf{h}^0$. Similarly, hubs are more important if they have many links to good authorities, so let $\mathbf{h}^1 = L\mathbf{a}^0$.

We continue in this way to improve the rankings based on the rankings of the previous step:

$$\mathbf{h}^{i+1} = L\mathbf{a}^i$$
$$\mathbf{a}^{i+1} = L^T\mathbf{h}^i$$

Notice:

$$\mathbf{h}^4 = L\mathbf{a}^3 = (LL^T)\mathbf{h}^2 = (LL^T)L\mathbf{a}^1 = (LL^T)^2\mathbf{h}^0$$

Similarly, $\mathbf{h}^{2k} = (LL^T)^k\mathbf{h}^0$ and $\mathbf{a}^{2k} = (L^TL)^k\mathbf{a}^0$. The rankings we want are **h** equals the leading eigenvector of LL^T and **a** equals the leading eigenvector of L^TL.

A Sample Network

$$interface(displayprecision = 2) : with(LinearAlgebra):$$

Below is a model of a network. Compute the link matrix L. Compute the rankings as follows:

$h[0] = L.Vector(7, 1) : Normalize(h[0], inplace) : h[1] := L.L^+.Vector(7, 1) : Normalize(h[1], inplace) : H :=< h[0]|h[1] >$:

$a[0] := L^+.Vector(7, 1) : Normalize(a[0], inplace) : a[1] := L^+.L.Vector(7, 1) : Normalize(a[1], inplace) : A :=< a[0]|a[1] >$:

for i from 2 to 6 do $h[i] := L.L^+.h[i-2] : Normalize(h[i], inplace) : H :=< H|h[i] >$: $a[i] := L^+.L.a[i-2] : Normalize(a[i], inplace) : A :=< A|a[i] >$ **end do**:

(The command *Normalize(v, inplace)* takes a vector v and replaces it by a vector in the same direction whose largest component is 1. You could use *Normalize(v,inplace,Euclidean)* to replace v by a unit vector.)

Observe how the rankings change at each step by considering the matrices H and A. These matrices will contain fractions, and will be easier to interpret by evaluating these as decimals using *evalf*:

$$map(evalf, H); map(evalf, A);$$

Estimate the eigenvectors **h** and **a** from these approximations. Is the site with the most incoming links the best authority? Is the site with the most outgoing links the best hub?

Verify that **h** is approximately an eigenvector of LL^T by computing $LL^T\mathbf{h}$ and normalizing. Compare the length of **h** to the length of $LL^T\mathbf{h}$ to estimate the leading eigenvalue. Do the same for **a**.