This lab was written by Professor Chris Cashen in his Postdoc years at Utah. Some minor display, maple code display, and font size changes were made. It remains exactly the same as Chris Cashen wrote.

Submit your project on the due date in class as a worksheet print. Submit study group efforts as one worksheet print with multiple names.

See Section 6.7 of the text for problem background.

1 Introduction

Suppose $B$ is an $n \times n$ matrix with real eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_n$ and corresponding eigenvectors $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n$. For any $\vec{w} \in \mathbb{R}^n$ we may write $\vec{w} = c_1 \vec{v}_1 + \cdots + c_n \vec{v}_n$. Then $B^k \vec{w} = c_1 \lambda_1^k \vec{v}_1 + \cdots + c_n \lambda_n^k \vec{v}_n$. As $k$ gets large, $\lambda_1^k$ becomes much larger than the other $\lambda_i^k$, so, assuming $c_i \neq 0$, $B^k \vec{w} \approx c_1 \lambda_1^k \vec{v}_1$. In other words, the orbit of $\vec{w}$, which is the sequence $\vec{w}, B\vec{w}, B^2\vec{w}, B^3\vec{w}, \ldots$ approaches multiples of the leading eigenvector $\vec{v}_1$. Conversely, we can find $\vec{v}_1$ by following an orbit $\vec{w}, B\vec{w}, B^2\vec{w}, B^3\vec{w}, \ldots$. Let $\vec{v}_1$ be 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 of the text 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 determine how important a site is. We use an iterative method to determine the site rankings.

Let $a_i$ be 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 $\vec{a} = (a_i)$ and $\vec{h} = (h_i)$. 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 1 consisting of all 1’s. Thus, $\vec{h}^0 = L \vec{1}$. Similarly, $\vec{a}^0 = L^T \vec{1}$, which counts incoming links.

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

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

\[
\begin{align*}
\vec{h}^{i+1} &= L\vec{a}^i \\
\vec{a}^{i+1} &= L^T \vec{h}^i
\end{align*}
\]

Notice:

\[
\vec{h}^4 = L\vec{a}^3 = (LL^T)\vec{h}^2 = (LL^T)L\vec{a}^1 = (LL^T)^2 \vec{h}^0
\]
Similarly, $\vec{h}^{2k} = (LL^T)^k \vec{h}^0$ and $\vec{a}^{2k} = (L^T L)^k \vec{a}^0$. The rankings we want are $\vec{h}$ equals the leading eigenvector of $LL^T$ and $\vec{a}$ equals the leading eigenvector of $L^T L$.

## 2 A Sample Network

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

```maple
interface(displayprecision=2):with(LinearAlgebra):
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);` and `map(evalf,A);`.

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