Math 2270-2 Spring 2012
Computer Lab 5: Web Search

The original version of this lab was written by Professor Chris Cashen in his Postdoc years at Utah. More figures and longer explanations were added in April 2012. Cashen's original Sample Network is used for the calculation and the answer set for the lab remains unchanged.
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 $\lambda_1 > \lambda_2 > \cdots > \lambda_n$. Denote the n eigenpairs by $(\lambda_1, \vec{v}_1), (\lambda_2, \vec{v}_2), \ldots, (\lambda_n, \vec{v}_n)$.
The n vectors $\vec{v}_1, \ldots, \vec{v}_n$ are independent and span $\mathcal{R}^n$, therefore for each $\vec{w} \in \mathcal{R}^n$ there are unique constants $c_1, \ldots, c_n$ such that

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

We verify that $B\vec{v}_1 = \lambda_1\vec{v}_1$ implies

$$B^2\vec{v}_1 = B(B\vec{v}_1) = B\left(\lambda_1\vec{v}_1\right) = \lambda_1 B\left(\vec{v}_1\right) = \lambda_1^2\vec{v}_1.$$

Similarly, $B^3\vec{v}_1 = \lambda_1^3\vec{v}_1$, and in general $B^k\vec{v}_1 = \lambda_1^k\vec{v}_1$. The same holds for any other eigenpair. Then

$$\begin{aligned} B^k\vec{w} &= c_1\lambda_1 B^k\vec{v}_1 + \cdots + c_n\lambda_n B^k\vec{v}_n \\ &= c_1\lambda_1^k\vec{v}_1 + \cdots + c_n\lambda_n^k\vec{v}_n \\ &= \lambda_1^k\left(c_1\vec{v}_1 + \left(\tfrac{\lambda_2}{\lambda_1}\right)^k c_2\vec{v}_1 + \cdots + \left(\tfrac{\lambda_n}{\lambda_1}\right)^k c_n\vec{v}_n\right) \\ &\approx \lambda_1^k c_1\vec{v}_1 \quad \text{as } k \to \infty. \end{aligned}$$

In other words, the sequence $\vec{w}, B\vec{w}, B^2\vec{w}, B^3\vec{w}, \cdots$ approaches multiples of the leading eigenvector $\vec{v}_1$. If $\vec{v}$ is $B^k\vec{w}$ for large k, then we expect $B\vec{v} \approx \lambda_1\vec{v}$. Take norms across this equation and solve for the lead eigenvalue:

$$\lambda_1 \approx \frac{\|B\vec{v}\|}{\|\vec{v}\|}.$$

We have found an approximate eigenpair $(\lambda_1, \vec{v})$, obtained without trying to compute roots of the characteristic polynomial of B.

Web Search Engine

Section 6.7 of Strang's textbook uses this approach to model an internet search engine. Let $s_1, \ldots, s_n$ be a list of web sites. Let $L = (l_{ij})$ be the incidence matrix that records links between the web sites: $l_{ij} = 1$ if site

s_i links to site s_j, and $l_{i\,j} = 0$ otherwise. By swapping rows and columns, the $i\,j$ entry of the transpose L^T is 1 if s_j links to s_i and zero otherwise.
There are two ways that a web site can be important, and therefore worthy of a top listing in the search engine.

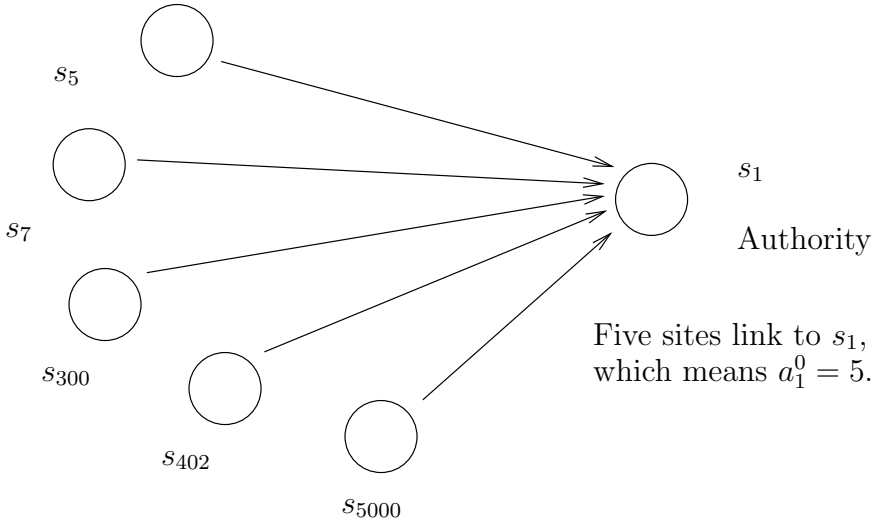
authority. A site s is an *authority* if many other sites link to s.
hub site. A site s is a *hub* if s links to many other sites.

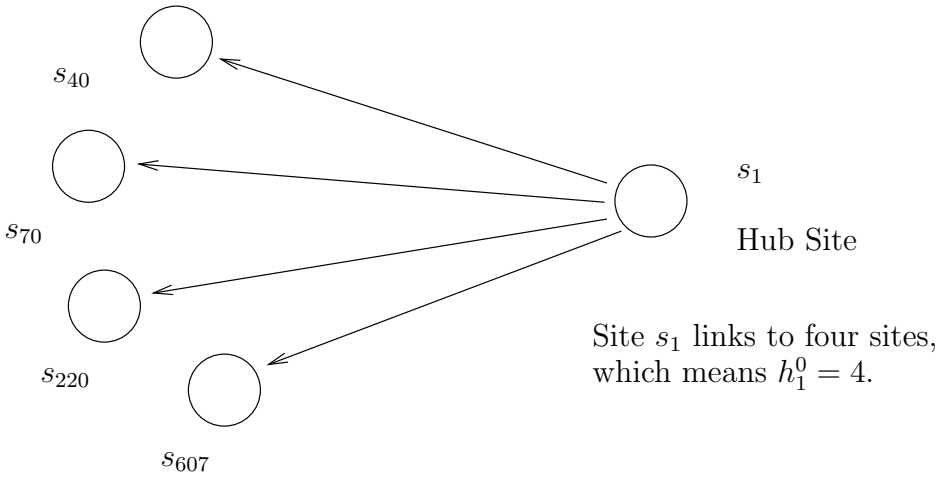
Ranking a Web Site

Initially, the ranking of a web site s is a simple count, defined as follows.

$$\begin{array}{rcl} a_i^0 & = & \text{the count of the number of sites } s \text{ that link to site } s_i \\ & = & \text{sum of row } i \text{ elements in incidence matrix } L^T, \\ h_i^0 & = & \text{the count of the number of sites } s \text{ to which } s_i \text{ links} \\ & = & \text{sum of row } i \text{ elements in incidence matrix } L. \end{array}$$

Then a_i^0 is the authority ranking of site s_i and h_i^0 is the hub ranking of site s_i.





Define vectors

$$\vec{a}^0 = \begin{pmatrix} a_1^0 \\ a_2^0 \\ a_3^0 \\ \vdots \end{pmatrix}, \quad \vec{h}^0 = \begin{pmatrix} h_1^0 \\ h_2^0 \\ h_3^0 \\ \vdots \end{pmatrix}.$$

The number of links and also the quality of the links determines the importance of a site. We use an iterative method to determine the site rankings, denoted

$$\begin{aligned} \textbf{Authority:} \quad & \vec{a}^0, \vec{a}^1, \vec{a}^2, \ldots, \\ \textbf{hub:} \quad & \vec{h}^0, \vec{h}^1, \vec{h}^2, \ldots. \end{aligned}$$

The initial authority rankings and the initial hub rankings are computed using the incidence matrix L.

$$\begin{aligned} \vec{a}^0 &= L^T\vec{1} \\ \vec{h}^0 &= L\vec{1} \\ \vec{1} &= \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}. \end{aligned}$$

This works because $\textbf{row}(L,i)\vec{1}$ = sum of the elements in row i of L. This counts outgoing links from a hub, while $\textbf{row}(L^T,i)\vec{1}$ = sum of elements in row i of L^T is the count of the number of incoming links to an authority.

The initial authority ranking of a the site s_i is given by the i–th component of $\vec{a}^0$. However, this ranking only counts links, but does not consider link quality. A site is a better 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 $\vec{h}^0$. Instead of adding up all the incoming links to s_i, instead we weight them by $\vec{h}^0$. This is why we define the next iteration to rank an authority by the equation $\vec{a}^1 = L^T\vec{h}^0$.

Similarly, hubs are more important if they have many links to good authorities, so we define the next iteration to rank a hub by the equation $\vec{h}^1 = L\vec{a}^0$.

We continue in this way to improve the rankings based on the rankings of the previous step and define by iteration the hub and authority rankings by the equations

$$\begin{aligned} \vec{h}^{i+1} &= L\vec{a}^i \\ \vec{a}^{i+1} &= L^T\vec{h}^i \end{aligned}$$

Compute new equations for the iterates as follows.

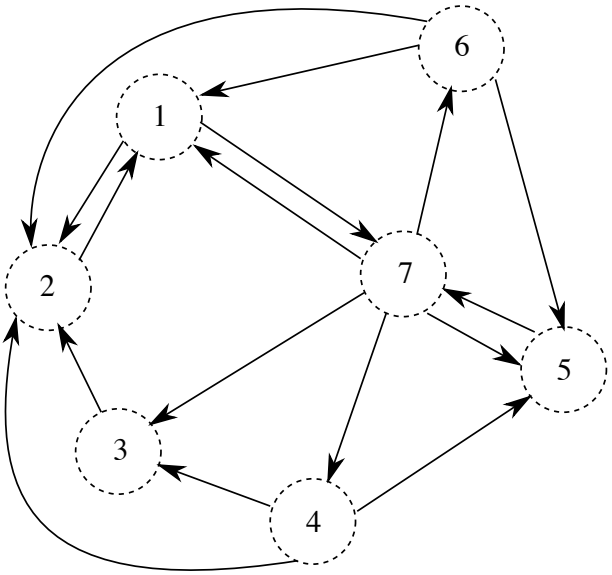
$$\begin{aligned} \vec{h}^4 &= L\vec{a}^3 \\ &= (LL^T)\vec{h}^2 \\ &= (LL^T)L\vec{a}^1 \\ &= (LL^T)^2\vec{h}^0, \\ &\quad \text{and in general} \\ \vec{h}^{2k} &= (LL^T)^k\vec{h}^0. \end{aligned}$$

Similarly

$$\begin{aligned} \vec{a}^4 &= L^T\vec{h}^3 \\ &= (L^TL)\vec{a}^2 \\ &= (L^TL)L^T\vec{h}^1 \\ &= (L^TL)^2\vec{a}^0, \\ &\quad \text{and in general} \\ \vec{a}^{2k} &= (L^TL)^k\vec{a}^0. \end{aligned}$$

The rankings we want are:

$\vec{h}$ equals the leading eigenvector of LL^T and
$\vec{a}$ equals the leading eigenvector of L^TL.



2 A Sample Network

Above is a model of a network representing several web sites.
Problem 1. Compute the incidence matrix L from the diagram.
Please verify this matrix.
For example, `2 --> 1`, `6 --> 1`, `7 --> 1`, which implies `col(L,1)=Vector([0,1,0,0,0,1,1])`.

```
L := <
<0, 1, 0, 0, 0, 1, 1>|
<1, 0, 1, 1, 0, 1, 0>|
<0, 0, 0, 1, 0, 0, 1>|
<0, 0, 0, 0, 0, 0, 1>|
<0, 0, 0, 1, 0, 1, 1>|
<0, 0, 0, 0, 0, 0, 1>|
<1, 0, 0, 0, 1, 0, 0>>;
```

Problem 2. Compute the rankings as follows:

```
interface(displayprecision = 2,rtablesize=25): with(LinearAlgebra):
# Compute the iterates a[0] to a[19] and h[0] to h[19].
fix:=x->Normalize(x,inplace):
Ones:=Vector(7,1):
h[0] := L.Ones; fix(h[0]);
a[0] := Transpose(L).Ones; fix(a[0]);
h[1] := L.a[0]; fix(h[1]);
a[1] := Transpose(L).h[0]; fix(a[1]);
for i from 2 to 19 do  h[i] := L.Transpose(L).h[i-2]; fix(h[i]); end do:
for i from 2 to 19 do  a[i] := Transpose(L).L.a[i-2]; fix(a[i]); end do:
# Do 10 iterations to find approximate eigenvectors for L^TL and LL^T.
H := <h[0] | h[1]>: for i from 2 to 9 do H := <H | h[i]>; end do:
A := <a[0] | a[1]>: for i from 2 to 9 do 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. This is done by dividing the vector v by $c = \max(v_1, \ldots, v_7)$.

Observe how the rankings change at each step by considering the matrices H and A. These matrices will contain fractions. It is easier to interpret H and A by using decimals instead of fractions:

```
A7by10:=evalf(A);
H7by10:=evalf(H);
```

The last column of each matrix is an eigenvector approximation.

Problem 3. Compute approximate eigenvectors $\vec{h}$ and $\vec{a}$ from these matrices, using $\vec{h} = \mathbf{col}(A, 10)$ and $\vec{a} = \mathbf{col}(A, 10)$.

```
# Approximate eigenvectors are the last columns of the two matrices A,H.
v1:=Column (A,10): v2:=Column (H,10): evalf(v1),evalf(v2);
```

Problem 4. Calculate the length of $LL^T\vec{h}$ and the length of $\vec{h}$, then divide to estimate the leading eigenvalue λ_2 of LL^T. **Repeat** for $L^TL\vec{a}$ and $\vec{a}$ to estimate the leading eigenvalue λ_1 of L^TL. This idea comes from the eigenpair relation

$$A\vec{v} = \lambda\vec{v}$$

which implies for $\lambda > 0$ the equation

$$\lambda = \frac{\|A\vec{v}\|}{\|\vec{v}\|}.$$

```
# Compute eigenvalue lambda2 for v2.
ans2:=L.Transpose(L).v2;
lambda2:=Norm(ans2)/Norm(v2);
# Compute eigenvalue lambda1 for v1.
ans1:=Transpose(L).L.v1;
lambda1:=Norm(ans1)/Norm(v1);
```

Problem 5. Verify that $\vec{h}$ is approximately an eigenvector for LL^T by computing $LL^T\vec{h} - \lambda_2\vec{h}$. It should be nearly the zero vector. **Repeat** to verify $\vec{a}$ is approximately an eigenvector of L^TL by computing $L^TL\vec{a} - \lambda_1\vec{a}$, which again should be nearly the zero vector.

```
# Verify eigenpair (lambda2,v2)
evalf(ans2-lambda2*v2);
# eigenpair (lambda2,v2) verified, if it is nearly zero.
# Verify eigenpair (lambda1,v1)
evalf(ans1-lambda1*v1);
# eigenpair (lambda1,v1) verified, if it is nearly zero.
```

Problem 6. Is the site with the most incoming links the best authority? Is the site with the most outgoing links the best hub?