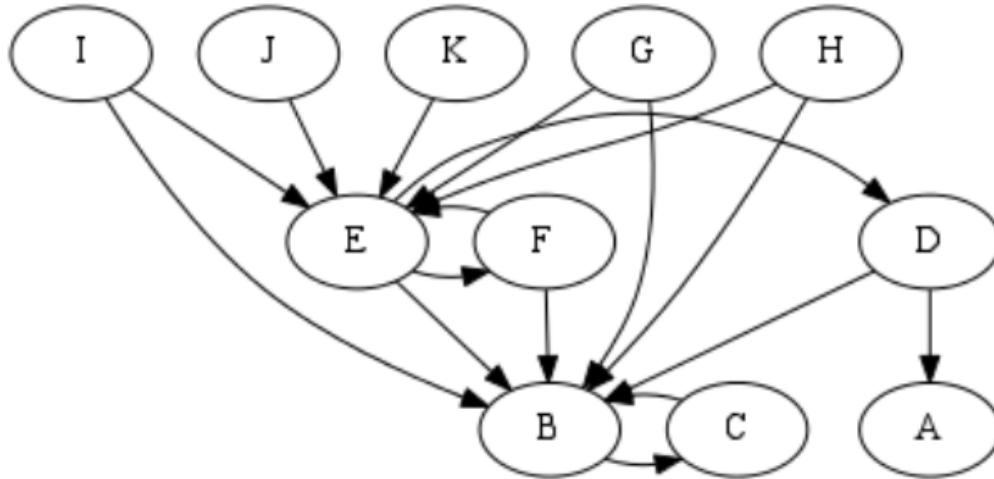
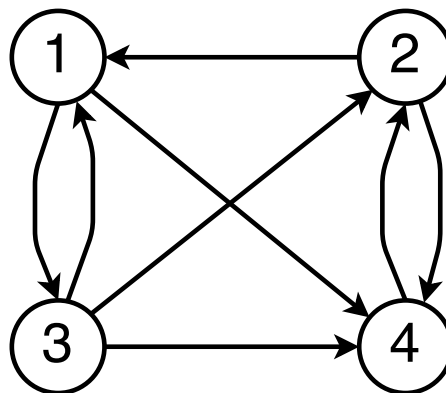


In class, we saw this graph, with each node representing people who are “following” each other on Twitter:



Our goal is to rank these people by their importance in the Twittersverse. We decided to do this by imagining that we start at a random node in the graph, and then we travel through the graph by following edges at random. The most “important” nodes are the ones we end up visiting most often. We simulated a random walk through the graph by randomly picking among torn up bits of paper to tell us where to go next, and then we recorded how many times we ended up at each node. It seems like there must be an easier way...

Thinking about probability



For this example, we're going to pretend the graph above represents our "internet", with four nodes representing web pages. In class, when we kept track of how often we ended up at each node in a random walk through the graph, what we were really doing was trying to estimate the **probability** that we'd end up visiting each node at some point. Intuitively, if we ended up at one node more often than the others, that tells us it has a higher probability of being visited, and is therefore more important. Here we're going to try to figure out the **actual value of these probabilities**, so we don't have to estimate it with a random walk.

A question we might ask could be: "If I walk randomly through this graph, what percentage of the time would I end up at page 1?" (Or 2, 3, 4?) But then, of course, we have to ask: how long do you need to walk? And does it matter where you start?

As it turns out, a question that gives you the same answer asymptotically – and is actually easier to answer – is: "After infinitely many steps through this graph, what's the probability that I'll end up at a given page?" At first glance, this doesn't seem like an easy question at all. But it turns out that, if we calculate the probability of ending up at each page after 1 step through the graph, and after 2 steps, 3 steps, ..., and keep going for long enough, it becomes pretty clear what the "correct" answer is. Once we formulate our problem more mathematically, we'll see why.

Power iterations

Define a vector

$$\vec{P}^{(k)} := \begin{pmatrix} p_1^{(k)} \\ p_2^{(k)} \\ p_3^{(k)} \\ p_4^{(k)} \end{pmatrix}$$

that denotes the probability of ending up at each of the four pages after k steps through the graph. So, for example, the probability that I'll end up at page 1 after taking 10 steps through the graph is given by $p_1^{(10)}$.

Let's assume that we start our walk through our 4-page internet by starting randomly at any given page. This means that, before we've taken any steps through the graph, our probability vector is

$$\vec{P}^{(0)} = \begin{pmatrix} p_1^{(0)} \\ p_2^{(0)} \\ p_3^{(0)} \\ p_4^{(0)} \end{pmatrix} = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}.$$

Now let's calculate $\vec{P}^{(1)}$, the probability distribution after taking a single step:

- **Probability of landing at page 1** ($p_1^{(1)}$): to get to page 1 after one step, that means we either started at page 2 and then transitioned to 1, or we started at 3 and then moved to 1. From 2, the probability of going to 1 is $1/2$ (since we could go either to 1 or 4); and from 3, the probability of going to 1 is $1/3$. This means that $p_1^{(1)} = 1/2 \times (\text{the probability that we were at 2}) + 1/3 \times (\text{the probability that we were at 3}) = 1/2 \cdot p_2^{(0)} + 1/3 \cdot p_3^{(0)} = 0.2083$.

- **Probability of landing at page 2** ($p_2^{(1)}$): to get to page 2 after one step, that means we started at 3 and went to 2, or we started at 4 and went to 2. The probability of going from 3 to 2 is $1/3$, and the probability of going from 4 to 2 is 1 (since there's only a single outgoing edge from 4). So, $p_2^{(1)} = 1/3 \cdot p_3^{(0)} + 1 \cdot p_4^{(0)} = 0.3333$.
- **Probability of landing at page 3** ($p_3^{(1)}$): the only way to get to page 3 is from page 1 – from page 1, you will go to page 3 half of the time. Thus, $p_3^{(1)} = 1/2 \cdot p_1^{(0)} = 0.1250$.
- **Probability of landing at page 4** ($p_4^{(1)}$): we can get to page 4 from page 1 with probability $1/2$, from page 2 with probability $1/2$, or page 3 with probability $1/3$. Hence $p_4^{(1)} = 1/2 \cdot p_1^{(0)} + 1/2 \cdot p_2^{(0)} + 1/3 \cdot p_3^{(0)} = 0.3333$.

We can use these same steps for the general case of computing $\vec{P}^{(k+1)}$ based on $\vec{P}^{(k)}$. Specifically, we can write:

$$\vec{P}^{(k+1)} := \begin{pmatrix} p_1^{(k+1)} \\ p_2^{(k+1)} \\ p_3^{(k+1)} \\ p_4^{(k+1)} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}p_2^{(k)} + \frac{1}{3}p_3^{(k)} \\ \frac{1}{3}p_3^{(k)} + p_4^{(k)} \\ \frac{1}{2}p_1^{(k)} \\ \frac{1}{2}p_1^{(k)} + \frac{1}{2}p_2^{(k)} + \frac{1}{3}p_3^{(k)} \end{pmatrix}.$$

This is a linear system, which we can write in matrix form as:

$$\begin{aligned} \vec{P}^{(k+1)} &= \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} & 1 \\ \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{3} & 0 \end{pmatrix} \vec{P}^{(k)} \\ &=: \mathbf{M} \vec{P}^{(k)}. \end{aligned}$$

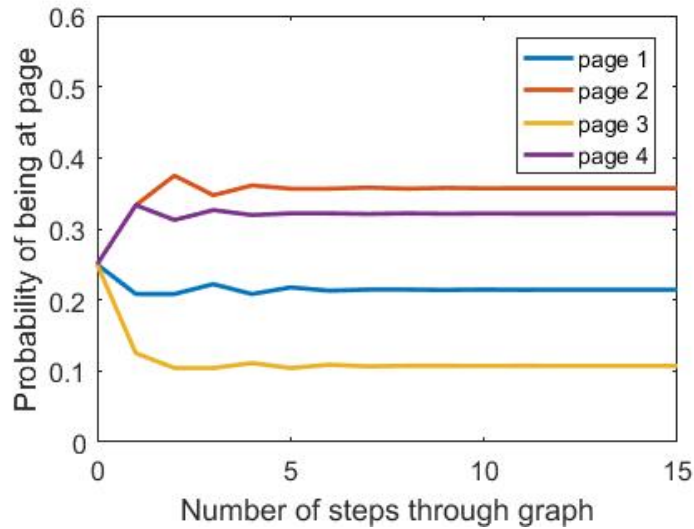
We should note a few important properties of the transition matrix \mathbf{M} , which will be true of any transition matrix for this kind of PageRank problem:

- All columns of \mathbf{M} sum to 1 – this implies that \mathbf{M} is *column-stochastic*, which means it has all kinds of nice properties.
- All nonzero entries in a given column have the same value (e.g., you can't have $1/3$ and $1/2$ in the same column).
- If there's no link from page i to page j , then $\mathbf{M}_{j,i} = 0$ (the subscript j, i means the i th column of the j th row).

With this matrix handy, we can calculate the probability of landing at each page after an arbitrary number of steps quite easily. If we have our starting distribution $\vec{P}^{(0)}$, the probability distribution after k steps will be

$$\vec{P}^{(k)} = \mathbf{M}^k \vec{P}^{(0)},$$

which you can calculate in MATLAB, Octave, Wolfram Alpha, etc. (You could also do it by hand, but I don't recommend it!) Essentially, we take a probability vector and multiply it repeatedly by the matrix \mathbf{M} : these are called *power iterations*. Going back to our 4-page webgraph, let's try



calculating $\vec{P}^{(k)}$ for successively higher k . The plot above shows what happens to the probability distributions as we multiply them repeatedly by \mathbf{M} , starting with a uniform initial distribution (each page has probability 0.25).

Notice anything surprising here? After about 8 steps through the graph, the probability distribution levels out! So now we can see how we might answer our earlier question: “After infinitely many steps through this graph, what’s the probability that I’ll end up at a given page?” Our probability distributions will, at some point, converge to a stationary (i.e., unchanging) probability distribution – and this is what we refer to when we talk about PageRank being “the probability of visiting a given page.”

For our example here, the stationary probability distribution – which we’ll call \vec{P} , without a superscript – is:

$$\vec{P} = \begin{pmatrix} 0.214 \\ 0.357 \\ 0.107 \\ 0.321 \end{pmatrix}.$$

This vector represents our long-term probability of visiting each page: this is the *PageRank* of each of the pages. So page 1 has rank 0.214, page 2 has rank 0.357, page 3 has rank 0.107, and page 4 has rank 0.321.

Eigenvectors

In the last section, we computed the PageRanks of a set of webpages by multiplying a starting probability vector by a matrix over and over again until the probabilities stopped changing. This is faster and more accurate than when we traversed graphs at random with the help of little pieces of paper, but it turns out there’s (yet another) way.

How do we know if our k th probability vector $\vec{P}^{(k)}$ is actually the final, stationary probability vector \vec{P} ? Well, we know it's stationary if it doesn't change at the next iteration, i.e. if:

$$\vec{P}^{(k)} = \vec{P}^{(k+1)} = \mathbf{M}\vec{P}^{(k)}.$$

So our final vector \vec{P} has the property that

$$\vec{P} = \mathbf{M}\vec{P}.$$

This means that \vec{P} is an **eigenvector** of \mathbf{M} with corresponding eigenvalue 1. Do you remember earlier when I mentioned that \mathbf{M} is column-stochastic, and column-stochastic matrices have some nice properties? Well, one of these is that they always have an eigenvalue equal to 1 – and if all components of our webgraph are connected (as is the case in our example), there is only a **single** eigenvalue equal to 1. This means we can compute a unique PageRank vector \vec{P} .

Let's try it on our 4×4 matrix \mathbf{M} from the last example. Below I've included the commands to input at the command line in MATLAB or Octave (an open-source MATLAB substitute), in case you don't have a method for computing eigenvectors. Black font is console output from MATLAB, and red font is what I typed in.

```
>> M = [0 1/2 1/3 0; 0 0 1/3 1; 1/2 0 0 0; 1/2 1/2 1/3 0]
```

```
M =
```

```
0          0.5000    0.3333         0
0           0        0.3333    1.0000
0.5000         0         0         0
0.5000    0.5000    0.3333         0
```

```
>> [eigVecs, eigVals] = eig(M)
```

```
eigVecs =
```

```
0.3991 + 0.0000i  -0.0000 + 0.0000i  -0.2887 - 0.5000i  -0.2887 + 0.5000i
0.6652 + 0.0000i   0.5345 + 0.0000i   0.5774 + 0.0000i   0.5774 + 0.0000i
0.1996 + 0.0000i  -0.8018 + 0.0000i  -0.0000 + 0.5000i  -0.0000 - 0.5000i
0.5987 + 0.0000i   0.2673 + 0.0000i  -0.2887 - 0.0000i  -0.2887 + 0.0000i
```

```
eigVals =
```

```
1.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i  -0.5000 + 0.2887i   0.0000 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.5000 - 0.2887i
```

Calling `[eigVecs, eigVals] = eig(M)` returns the four eigenvectors as columns of `eigVecs`, and `eigVals` is a matrix with the corresponding eigenvalues given on the diagonal.

Don't be alarmed by the complex numbers! The only eigenpair we're interested in is the first one – that's the one with eigenvalue 1. This gives you the eigenvector $[0.3991, 0.6652, 0.1996, 0.5987]^T$. This isn't a valid probability distribution, since the elements don't sum to 1. To get our stationary probability \vec{P} , we need to scale the vector so they do sum to 1. Doing so gives us

$$\vec{P} = \begin{pmatrix} 0.214 \\ 0.357 \\ 0.107 \\ 0.321 \end{pmatrix},$$

which is exactly what we got with the power iterations!

Problems

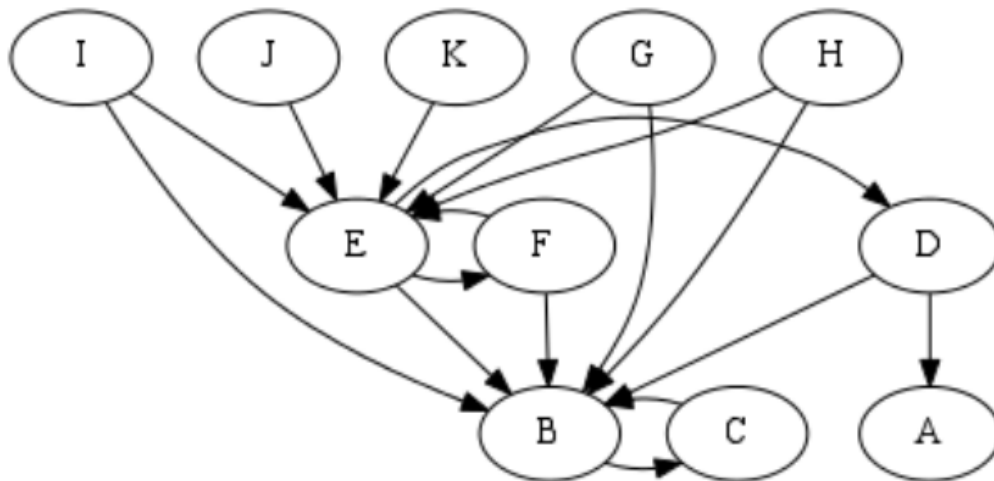
Here are some problems that you can complete for bonus marks. You can receive:

- Two bonus points for fully correct solutions to 1(a)-(e). You will receive one bonus point for fully correct solutions to 1(a)-1(d) only, OR for substantially but not entirely correct solutions to 1(a)-1(e).
- One bonus point for question 2.

No part marks will be given. You do not need to attempt all problems to submit your answers. Once you have answers, you can email them to me (sbrad@cs.ubc.ca) any time before the final exam (date TBD).

Question 1: The Twitterverse revisited

With a bit more information, you can calculate the PageRanks of the graph we looked at in class.



- (a) We'll start with a slightly simplified version of what we did in class. Suppose this is how we traverse the graph:

Repeat:

- Pick a random person q (from A-K) to start on.
- If q follows people, choose the next person at random from among the people q follows; if q doesn't follow anyone, choose a random person from A-K.

Write the formulas for computing $\vec{P}^{(k+1)}$ based on the values of $\vec{P}^{(k)}$. This will be just like we did in the example in the "Power iterations" section, but you need to take into account that because A doesn't follow anyone, from node A we could end up at any of the other nodes (including back at A) with equal probability.

I'll get you started by writing the formula for $p_F^{(k+1)}$: to get to node F at step $k+1$, that means that we either went from node E at the previous step (which happens with probability $1/3$), or we landed at node A at the previous step and then chose node F at random for the next (which has probability $1/11$). So, the formula for $p_F^{(k+1)}$ is:

$$p_F^{(k+1)} = \frac{1}{11}p_A^{(k)} + \frac{1}{3}p_E^{(k)}.$$

- (b) Write down the transition matrix \mathbf{M} for this problem. Remember that \mathbf{M} needs to be column-stochastic, meaning the columns must all sum to 1.
- (c) Calculate the eigenvalues of \mathbf{M} . If you did part (b) correctly, there should be exactly one eigenvalue equal to 1. What's the corresponding eigenvector?
- (d) Based on the eigenvector you got from (c), what are the PageRanks? Explain your result. Does this strike you as a useful answer? Why or why not?
- (e) If you got answered (d), you should hopefully understand why, in class, we used this slightly different traversal method:

Repeat:

- Pick a random person q from A-K to start on.
- Choose a random number from 1 to 6.
- If your random number is less than 6:
 - If q follows people, choose the next person at random among the people q follows; if q doesn't follow anyone, choose a random person from A-K.
- If your random number is equal to 6:
 - Begin at a random person q from A-K.

This is a different problem called *PageRank with damping*: at each step, we have a probability d (in this case $d = 1/6$) of starting at a new random page. The probability d is called a *damping factor*. With damping, our formulas for $\vec{P}^{(k)}$ are going to look a bit different. To repeat the example of $p_F^{(k+1)}$: we can get to node F the "usual" way from E or A (after picking a random number less than 6), or it can be our new starting node after we pick a random number equal to 6.

The probability that we pick a random start at step k and then end up at F at step $k + 1$ is going to be $\frac{1}{6} \times \frac{1}{11}$ (that's the probability of a random start, times the probability that we then choose node F). The probability of getting to F directly from E or A is $\frac{5}{6} \left(\frac{1}{11}p_A^{(k)} + \frac{1}{3}p_E^{(k)} \right)$ – in other words, it's the same transition probability we had before, but multiplied by $5/6$. That's because, if we're at node A or node E, we now have probability $1/6$ of starting somewhere at random, instead of taking one of the usual paths from the node. Adding these together gives us

$$p_F^{(k+1)} = \frac{1}{66} + \frac{5}{6} \left(\frac{1}{11}p_A^{(k)} + \frac{1}{3}p_E^{(k)} \right).$$

If we do this for every node we can see that, in the general case, the update rule for a PageRank problem with damping factor d is

$$\vec{P}^{(k+1)} = \begin{pmatrix} d/N \\ d/N \\ \vdots \\ d/N \end{pmatrix} + (1 - d)\mathbf{M}\vec{P}^{(k)},$$

where N is the number of pages and \mathbf{M} is the original transition matrix.

Compute the PageRanks of our graph for the damped problem. This means calculating the stationary probability distribution \vec{P} that satisfies:

$$\vec{P} = \begin{pmatrix} d/N \\ d/N \\ \vdots \\ d/N \end{pmatrix} + (1 - d)\mathbf{M}\vec{P}.$$

Question 2: Power iterations vs. eigenvector calculations

Warning: this question requires some linear algebra knowledge.

In the 4×4 example we looked at earlier, we initially computed the PageRanks by starting with all pages having a uniform probability of 0.25, and repeatedly multiplying this initial probability vector by \mathbf{M} . We saw that if we did this enough times, we eventually got to the same answer that we got with the eigenvector computation (i.e., the eigenvector with eigenvalue 1). But would this have happened for **any** initial probability vector $\vec{P}^{(0)}$?

Assume that the vector entries are all non-negative, and they all sum up to 1 – in other words, that $\vec{P}^{(0)}$ is a valid probability distribution. Either prove that the sequence $\vec{P}^{(k)} = \mathbf{M}^k \vec{P}^{(0)}$ converges to the eigenvector \vec{P} (with eigenvalue 1) for any such $\vec{P}^{(0)}$, or provide a counterexample. You only need to prove/disprove the claim for the particular 4×4 matrix \mathbf{M} that we looked at.

Hint: Suppose \mathbf{A} is a matrix with dominant eigenvector \vec{X}_1 (this means the eigenvector corresponding to the largest-magnitude eigenvalue). For a vector \vec{v}_0 , the sequence $\mathbf{A}^k \vec{v}_0$ converges to \vec{X}_1 provided that \vec{v}_0 has some component in the direction of \vec{X}_1 . You can use the eigenvalues and eigenvectors we computed for \mathbf{M} to help you answer.