Lecture 19: Numerical Linear Algebra, PageRank cont'd

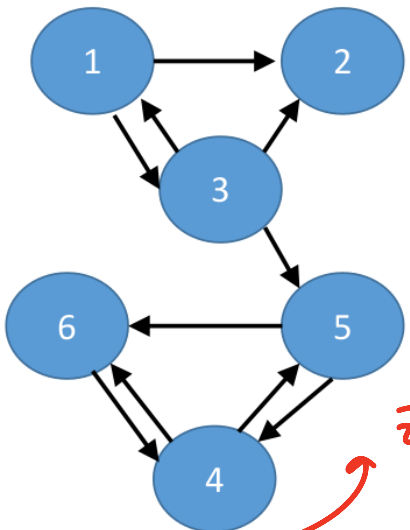
Leili Rafiee Sevyeri

Based on lecture notes by me and many previous CS370 instructors

Winter 2024

Cheriton School of Computer Science
University of Waterloo

Page Rank example (Notes Ex. 7.4)



$$P = \begin{bmatrix} 0 & 0 & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 1 \\ 0 & 0 & \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$$

$$d = [0, 1, 0, 0, 0, 0].$$

(Page 2 is a dead end!)

zero column

dead end

Random Surfing
(but gets stuck in dead ends)

$$G = \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \\ \begin{array}{cccccc} +\downarrow & +\downarrow & +\downarrow & +\downarrow & +\downarrow & +\downarrow \\ 2 & 0 & 3 & 2 & 2 & 1 \end{array} \end{array}$$

Page Rank example

$$\frac{1}{6} e d^T = \begin{bmatrix} 0 & 1/6 & 0 & 0 & 0 & 0 \\ \vdots & 1/6 & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1/6 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$P' = P + \frac{1}{6} e d^T = \begin{bmatrix} 0 & \frac{1}{6} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{6} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{6} & 0 & 0 & \frac{1}{2} & 1 \\ 0 & \frac{1}{6} & \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{6} & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$$

$$M = \begin{bmatrix} \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{6} & \frac{1}{6}\alpha+\frac{1}{6} & \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{6}-\frac{1}{6}\alpha \\ \frac{1}{3}\alpha+\frac{1}{6} & \frac{1}{6} & \frac{1}{6}\alpha+\frac{1}{6} & \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{6}-\frac{1}{6}\alpha \\ \frac{1}{3}\alpha+\frac{1}{6} & \frac{1}{6} & \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{6}-\frac{1}{6}\alpha \\ \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{6} & \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{3}\alpha+\frac{1}{6} & \frac{5}{6}\alpha+\frac{1}{6} \\ \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{6} & \frac{1}{6}\alpha+\frac{1}{6} & \frac{1}{3}\alpha+\frac{1}{6} & \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{6}-\frac{1}{6}\alpha \\ \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{6} & \frac{1}{6}-\frac{1}{6}\alpha & \frac{1}{3}\alpha+\frac{1}{6} & \frac{1}{3}\alpha+\frac{1}{6} & \frac{1}{6}-\frac{1}{6}\alpha \end{bmatrix}$$

Add Teleportation
out of Dead Ends
(fills in empty cols)

Add Occasional Random
Teleportation to Also Escape Cycles

$$M = \alpha P' + (1-\alpha)\frac{1}{R} e e^T$$

Page Rank example – Final Google matrix

For $\alpha = 0.85$, we have:

$$M = \begin{bmatrix} \frac{1}{40} & \frac{1}{6} & \frac{37}{120} & \frac{1}{40} & \frac{1}{40} & \frac{1}{40} \\ \frac{9}{20} & \frac{1}{6} & \frac{37}{120} & \frac{1}{40} & \frac{1}{40} & \frac{1}{40} \\ \frac{9}{20} & \frac{1}{6} & \frac{1}{40} & \frac{1}{40} & \frac{1}{40} & \frac{1}{40} \\ \frac{1}{40} & \frac{1}{6} & \frac{1}{40} & \frac{1}{40} & \frac{9}{20} & \frac{7}{8} \\ \frac{1}{40} & \frac{1}{6} & \frac{37}{120} & \frac{9}{20} & \frac{1}{40} & \frac{1}{40} \\ \frac{1}{40} & \frac{1}{6} & \frac{1}{40} & \frac{9}{20} & \frac{9}{20} & \frac{1}{40} \end{bmatrix}.$$

The sum of each column is 1.

PageRank So Far

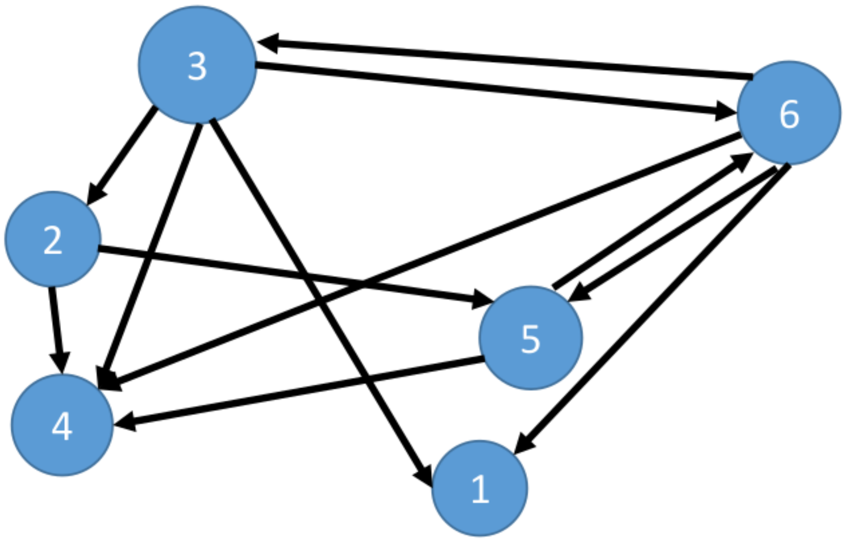
- Introduced the simple "random surfer" model for ranking web pages
- Began describing the random surfing process with a "Google matrix" of transition *probabilities*.

Next up:

- Look at properties of the matrix M.
- Explore how it's used in the "actual" PageRank algorithm
- Review eigenvalues/vectors

Example Review

Construct the google matrix $M = \alpha\left(P + \frac{1}{R}ed^T\right) + (1-\alpha)\frac{1}{R}ee^T$ for the small web shown here, using $\alpha = \frac{1}{2}$, and $R = 6$ pages.



Recall:

$$P_{ij} = \begin{cases} \dfrac{1}{\deg(j)}, & \text{if link } j \rightarrow i \text{ exists} \\ 0, & \text{otherwise} \end{cases}$$

$$d_i = \begin{cases} 1, & \text{if } deg(i) = 0 \\ 0, & \text{otherwise} \end{cases}$$

$$e = [1,1,1,\ldots,1]^T$$

Solution

$$G = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

$$\begin{matrix} + & + & + & + & + & + \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 2 & 4 & 0 & 2 & 4 \end{matrix}$$

$$P = \begin{bmatrix} 0 & 0 & 1/4 & 0 & 0 & 1/4 \\ 0 & 0 & 1/4 & 6 & 0 & 0 \\ 0 & 0 & 0 & \vdots & 0 & 1/4 \\ 0 & 1/2 & 1/4 & \vdots & 1/2 & 1/4 \\ 0 & 1/2 & 0 & \vdots & 0 & 1/4 \\ 0 & 0 & 1/4 & 0 & 1/2 & 0 \end{bmatrix}$$

$$d = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T$$

$$e = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T$$

$R = 6, \quad \alpha = 1/2.$

$$M = \alpha\left(P + \frac{1}{R} e d^T\right) + (1-\alpha)\frac{1}{R} e e^T$$

$$\alpha\left(P + \frac{1}{R} e d^T\right) = \frac{1}{2}\begin{bmatrix} 1/6 & 0 & 1/4 & 1/6 & 0 & 1/4 \\ 1/6 & 0 & 1/4 & 1/6 & 0 & 0 \\ 1/6 & 0 & 0 & 1/6 & 0 & 1/4 \\ 1/6 & 1/2 & 1/4 & 1/6 & 1/2 & 1/4 \\ 1/6 & 1/2 & 0 & 1/6 & 0 & 1/4 \\ 1/6 & 0 & 1/4 & 1/6 & 1/2 & 0 \end{bmatrix}$$

$$\frac{(1-\alpha)}{R} e e^T = \frac{1}{2}\left(\frac{1}{6}\right)\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$$

i.e., $\frac{1}{12}$ everywhere!

$$M = \begin{bmatrix} 1/6 & 1/12 & 5/24 & 1/6 & 1/12 & 5/24 \\ \vdots & 1/12 & 5/24 & \vdots & 1/12 & 1/12 \\ \vdots & 1/12 & 1/12 & \vdots & 1/12 & 5/24 \\ \vdots & 1/3 & 5/24 & \vdots & 1/3 & 5/24 \\ \vdots & 1/3 & 1/12 & \vdots & 1/12 & 5/24 \\ 1/6 & 1/12 & 5/24 & 1/6 & 1/3 & 1/12 \end{bmatrix}$$

observe that Column sum to 1 (individually).

Some Useful Properties of M

The entries of M satisfy $0 \leq M_{ij} \leq 1$.

Each column of M sums to 1.

$$\sum_{i=1}^{R} M_{ij} = 1$$

Interpretation: if we are on a webpage, probability of being on some webpage after a transition is 1. (i.e., we can't just disappear).

Markov Transition Matrices

The google matrix M is an example of a Markov matrix.

We define a Markov matrix Q by the two properties we just saw:

$$0 \leq Q_{ij} \leq 1$$

and

$$\sum_{i=1}^{R} Q_{ij} = 1$$

Probability Vector

Now, define a **probability vector** as a vector q such that

$$0 \leq q_i \leq 1$$

and

$$\sum_{i=1}^{R} q_i = 1$$

If the "surfer" starts at a random page with equal probabilities, this can be represented by a probability vector, where $p_i = \frac{1}{R}$.

If a surfer starts at page 1, then the corresponding probability vector is $[1, 0, 0, \ldots, 0]$.

Evolving The Probability Vector

superscript on p indicates number of transitions taken.

Now we have:

- the probability vector describing the **initial state**, p^0.
- a Markov matrix M describing the **transition probabilities** among pages.

Their product Mp^0 tells us the probabilities of our surfer being at each page after **one transition**.

$$p^1 = Mp^0$$

Likewise, for any step n, next step probabilities are, $p^{n+1} = Mp^n$.

Evolving The Probability Vector: Example #1

$p^0 = [1, 0, 0, 0]^T$. (We're *definitely* starting on page 1.)

If we had a google/transition matrix $M = \begin{bmatrix} 1/3 & 0 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 \\ 1/3 & 1/2 & 0 & 0 \\ 1/3 & 0 & 1 & 1/2 \end{bmatrix}$

Then after one step, what is $p^1 = Mp^0$? And what does it mean?

$$Mp^0 = p^1 = \left[\frac{1}{3}, 0, \frac{1}{3}, \frac{1}{3}\right]^T$$

Approximately 33% chance of being at page 1, 3, or 4 after one step, starting from page 1.

Evolving The Probability Vector: Example #2

$p^0 = \left[\frac{1}{2}, 0, \frac{1}{2}, 0\right]^T$. (We're on page 1 or 3 with probability 0.5 each.)

If we have same matrix $M = \begin{bmatrix} 1/3 & 0 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 \\ 1/3 & 1/2 & 0 & 0 \\ 1/3 & 0 & 1 & 1/2 \end{bmatrix}$,

then after one step, what is p^1?

$$p^1 = Mp^0 = \begin{bmatrix} 1/3 & 0 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 \\ 1/3 & 1/2 & 0 & 0 \\ 1/3 & 0 & 1 & 1/2 \end{bmatrix} \begin{bmatrix} 1/2 \\ 0 \\ 1/2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/6 \\ 0 \\ 1/6 \\ 4/6 \end{bmatrix} \to 1$$

Sum of the column $= 1$

$0 \leq p_i^1 \leq 1$, So, p^1 is a probability vector.

Preserving a Probability Vector

$0 \leq p_i^{n+1} \leq 1$

If p^n is a probability vector, is $p^{n+1} = Mp^n$ also a probability vector?

(ie. Do we have: $0 \leq p_i^{n+1} \leq 1$ and $\sum_i p_i^{n+1} = 1$?)

Yes! First, why non-negative?

We have $p_i^{n+1} \geq 0$, since it is just sums & products of probabilities ≥ 0.

Preserving a Probability Vector

$p^{n+1} = Mp^n$, where M is a Markov matrix.

Reordering summation $\sum_i M_{ij} = 1$

We can also show $\sum_i p_i^{n+1} = 1$, as follows:

$$\sum_i p_i^{n+1} = \sum_i \sum_j M_{ij} p_j^n = \sum_j \left(p_j^n \underbrace{\sum_i M_{ij}}_{1} \right) = \sum_j p_j^n = 1$$

Column (M) = 1

since p^n was a probability vector

To be a probability vector, also need $p_i^{n+1} \leq 1$? Why is this true?

by def'n of matrix/vector multiply

This sum is also 1, as p^n is a probability vector.

$\Longrightarrow p^{n+1}$ is a probability vector.

Page Rank idea

Finally, Page Rank asks:

With what **probability** does our surfer end up at each page after **many** steps, starting from $p^0 = \frac{1}{R}e$?

After k steps we are in

$$\underbrace{M \ldots (M(}_{k}Mp^0)) = M^k p^0 \text{ state}$$

i.e., What is $p^\infty = \lim_{k \to \infty} (M)^k p^0$?

Exponent k (not step index)

Higher probability in p^∞ vector implies greater importance.

Then we can rank the pages by this importance measure.

Page Rank algorithm summary

1. Given a graph of a network, compute a corresponding Google transition (Markov) matrix...

$$M = \alpha\left(P + \frac{1}{R}ed^T\right) + (1-\alpha)\frac{1}{R}ee^T$$

2. Repeatedly evolve a probability vector p^i via $p^{n+1} = Mp^n$ towards a steady state, approximating a "random surfer".

3. The site with the highest probability of being visited is considered most important/influential.

Starting from $p^0 = \frac{1}{R}e$, repeated multiplication by M gives a sequence of probability vectors, eventually settling down.

$$\underset{p^0}{\begin{bmatrix} 0.16667 \\ 0.16667 \\ 0.16667 \\ 0.16667 \\ 0.16667 \\ 0.16667 \end{bmatrix}}, \underset{p^1 = Mp^0}{\begin{bmatrix} 0.09583 \\ 0.16666 \\ 0.11944 \\ 0.26111 \\ 0.16666 \\ 0.19027 \end{bmatrix}}, \underset{p^2 = M^2p^0}{\begin{bmatrix} 0.08245 \\ 0.12318 \\ 0.08934 \\ 0.28118 \\ 0.19342 \\ 0.23041 \end{bmatrix}}, \underset{p^3 = M^3p^0}{\begin{bmatrix} 0.06776 \\ 0.10280 \\ 0.07749 \\ 0.32051 \\ 0.18726 \\ 0.24415 \end{bmatrix}}, \ldots, \underset{p^{10} = M^{10}p^0}{\begin{bmatrix} 0.05205 \\ 0.07428 \\ 0.05782 \\ 0.34797 \\ 0.19975 \\ 0.26810 \end{bmatrix}}$$

Page #	Ranking
1	6
2	4
3	5
4	1
5	3
6	2

For earlier example, the pages are ranked as: $4, 6, 5, 2, 3, 1$ based on these final probabilities.

Questions to Ponder...

- Do we actually know if it will settle (*converge*) to a fixed final result?
- If yes, then how long will it take? Roughly how many *iterations* are needed before we can stop?
- Can we implement this *efficiently* (e.g. for very large networks?)

Making Page Rank Efficient

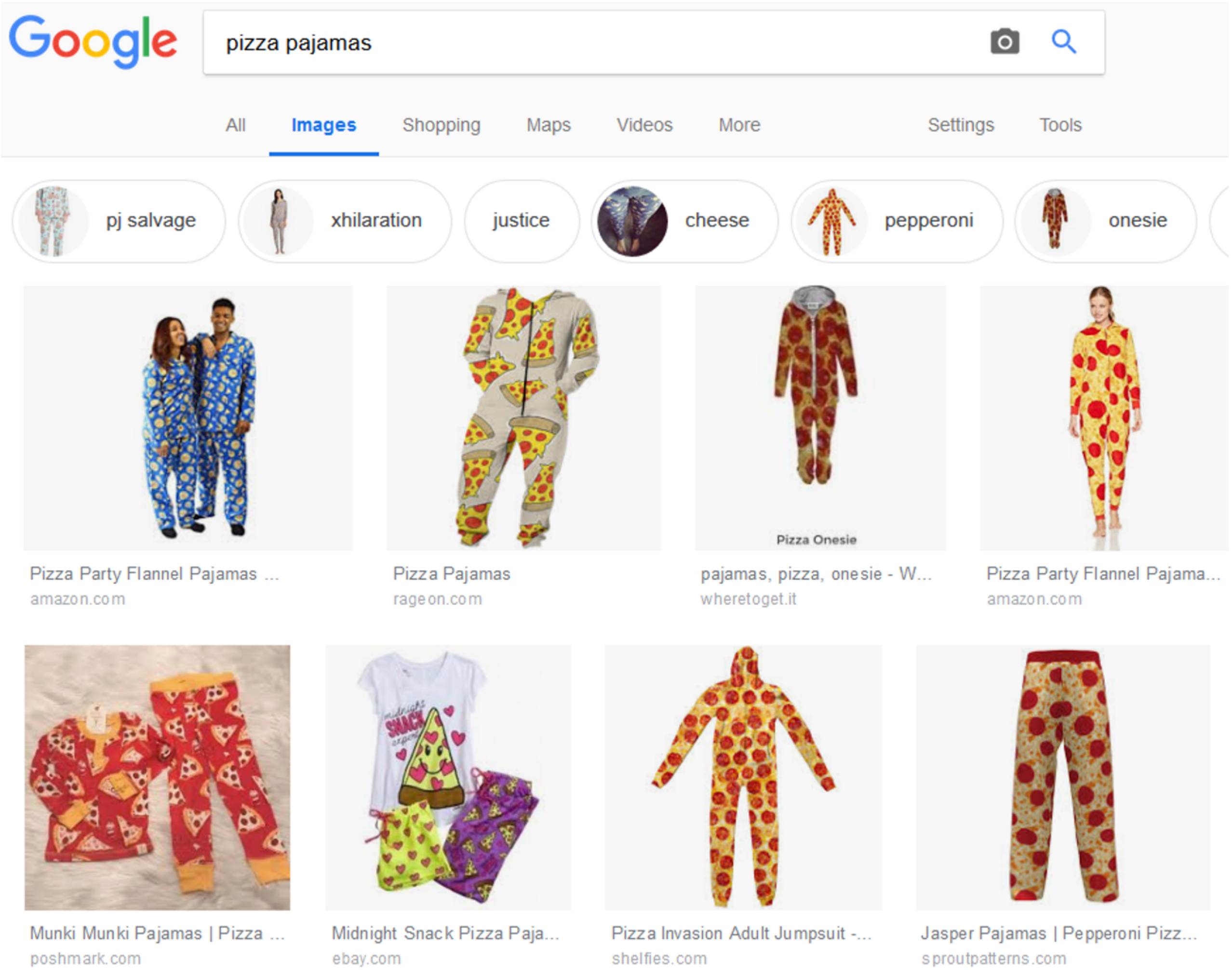
A naïve implementation of Page Rank involves repeatedly multiplying massive matrices with dimensions $> 1\text{billion} \times 1\text{billion}$.

How can we apply/implement this in a way that is computationally feasible?

We'll exploit (1) precomputation and (2) sparsity.

First step: Precomputation

The ranking vector p^∞ can be precomputed once and stored, independent of any specific query. To later search for a keyword, e.g., "pizza pajamas", Google finds **only** the subset of pages matching the keyword(s), and ranks those by their values in the **(precomputed)** p^∞.

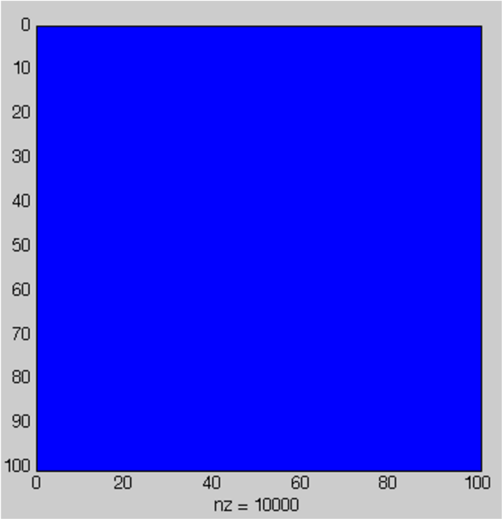


Second Step: Sparsity

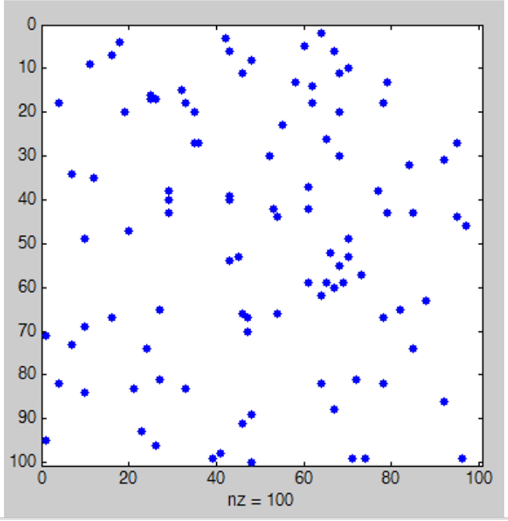
In numerical linear algebra, we often deal with two kinds of matrices.

Dense: Most or all entries are **non-zero**. Store in an $N \times N$ array, manipulate "normally".

Sparse: Most entries are **zero**. Use a "sparse" data structure to save space (and time). Prefer algorithms that avoid "destroying" sparsity (i.e., filling in zero entries).



Non-zero entries (blue) in a dense matrix.



Non-zeros in a sparse matrix.

Sparse Matrix-Vector multiplication

Multiplying a sparse matrix with a vector can be done efficiently!

Only non-zero matrix entries are ever accessed/used.

6x5 multiplication

7 "

$$\begin{pmatrix} \boxed{2} & \square & \square & \boxed{1} & \square \\ \square & \square & \square & \square & -2 \\ \square & \square & 1 & \square & \square \\ \square & 3 & \square & \square & \square \\ \square & \square & \square & \square & \square \\ 1 & \square & \square & \square & 1 \end{pmatrix} \begin{pmatrix} \boxed{1} \\ 2 \\ 3 \\ \boxed{4} \\ 5 \end{pmatrix} = \begin{pmatrix} \boxed{2 \cdot 1} + \boxed{1 \cdot 4} \\ -2 \cdot 5 \\ 1 \cdot 3 \\ 3 \cdot 2 \\ 0 \\ 1 \cdot 1 + 1 \cdot 5 \end{pmatrix} = \begin{pmatrix} 6 \\ -10 \\ 3 \\ 6 \\ 0 \\ 6 \end{pmatrix}$$

Note that the little squares are representing zeros here.

Second Step: Exploiting Sparsity

To implement Page Rank efficiently, it is crucial to exploit sparsity.

Sadly, our google matrix M was **fully dense**. No zeros at all!

A dense matrix-vector multiply with $1,000,000,000^2$ entries is slooooooooooooooooooO00O...O00OOooooOO000OOOoooooow.

The trick: Use linear algebra manipulations to perform the main iteration

$$p^{n+1} = Mp^n$$

without ever creating/storing M!

Exploiting Sparsity in M

dense column in "dead end"

We have $M = \alpha\left(P + \frac{1}{R} e d^T\right) + \frac{(1-\alpha)}{R} e e^T$

sparse, not all pages are linked together

fully dense

Consider computing

$$M p^n = \underset{(1)}{\alpha P p^n} + \underset{(2)}{\frac{\alpha}{R} e d^T p^n} + \underset{(3)}{\frac{(1-\alpha)}{R} e e^T p^n}$$

Output p^{n+1} is a vector, and a sum of 3 vectors:

(1) is a sparse matrix-vector multiply. It can be done efficiently.

(3) involves $ee^T p^n = e(e^T p^n)$ which requires the "dot-product" $e^T p^n$.

This is just 1.

End of Lecture $\underline{\underline{19}}$, computation is not done yet!

Algorithm

Given this efficient/sparse iteration, loop until the max change in probability vector per step is small (< tol) – easy!

Page Rank Algorithm

$$
\begin{aligned}
&\mathbf{p}^0 = \mathbf{e}/R \\
&\text{For } k = 1, \ldots, \text{ until converged} \\
&\qquad \mathbf{p}^k = M\mathbf{p}^{k-1} \\
&\qquad \text{If } \max_i |[\mathbf{p}^k]_i - [\mathbf{p}^{k-1}]_i| < tol \text{ then quit} \\
&\text{EndFor}
\end{aligned}
\tag{7.7.1}
$$

Google Search: Other Factors

Page Rank can be "tweaked" to incorporate other (commercial?) factors.

Replace standard teleportation $\frac{1-\alpha}{R}ee^T$ with $(1-\alpha)\nu e^T$, where a special probability vector ν places extra weight on whatever sites you like.

In modern search engines, many factors besides pure link-based ranking can come into play.

(Hence, Search Engine Optimization (SEO) is a lucrative business.)

Convergence of Page Rank

Remaining questions:

- How can we be sure that Page Rank will ever "settle down" to a fixed probability vector?
- If it does, how many iterations will it take?

We will need some additional facts about Markov matrices, involving **eigenvalues** and **eigenvectors**.

Review: Eigenvalues and Eigenvectors

Recall from linear algebra:

An eigenvalue λ and corresponding eigenvector $\mathbf{x}$ of a matrix Q are a scalar and non-zero vector, respectively, which satisfy

$$Q\mathbf{x} = \lambda\mathbf{x}.$$

Review: Eigenvalues and Eigenvectors

Equivalently, this can be written

$$Q\mathbf{x} = \lambda \mathbf{I}\mathbf{x}$$

where I is the identity matrix.

Rearranging gives

$$(\lambda I - Q)\mathbf{x} = \mathbf{0}$$

which implies that the matrix $\lambda I - Q$ must be *singular* for λ and $\mathbf{x}$ to be an eigenvalue/eigenvector pair, since we want $\mathbf{x} \neq \mathbf{0}$.

Quick Review: Eigenvalues and Eigenvectors

A *singular* matrix A satisfies $\det A = 0$.

Thus to find the eigenvalues λ of Q, we can solve the **characteristic polynomial** given by

$$\det(\lambda I - Q) = 0$$

Example

Find the eigenvalues/eigenvectors of $\begin{bmatrix} 2 & 2 \\ 5 & -1 \end{bmatrix}$