

CSE 312

Foundations of Computing II

Lecture 23: Finish distinct elements; Markov Chains + application

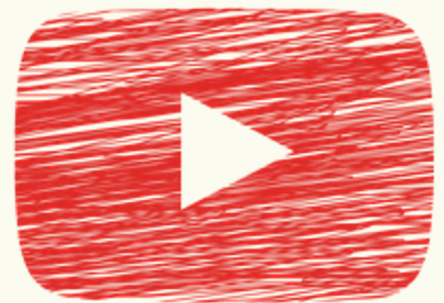
Today: Counting distinct elements

32, 12, 14, 32, 7, 12, 32, 7, 32, 12, 4

Application

You are the content manager at YouTube, and you are trying to figure out the **distinct** view count for a video. How do we do that?

Note: A person can view their favorite videos several times, but they only count as 1 **distinct** view!



Other applications

- IP packet streams: How many distinct IP addresses or IP flows (source+destination IP, port, protocol)
 - Anomaly detection, traffic monitoring
- Search: How many distinct search queries on Google on a certain topic yesterday
- Web services: how many distinct users (cookies) searched/browsed a certain term/item
 - Advertising, marketing trends, etc.

Counting distinct elements

32, 12, 14, 32, 7, 12, 32, 7, 32, 12, 4

N = # of IDs in the stream = 11, m = # of distinct IDs in the stream = 5

Want to compute number of **distinct** IDs in the stream.

How to do this without storing all the elements?

Detour – Min of I.I.D. Uniforms

If $Y_1, \dots, Y_m \sim \text{Unif}(0,1)$ (iid) where do we expect the points to end up?

$$\text{In general, } \mathbb{E}[\min(Y_1, \dots, Y_m)] = \frac{1}{m+1}$$

$$\mathbb{E}[\min(Y_1)] = \frac{1}{1+1} = \frac{1}{2}$$

$m = 1$



$$\mathbb{E}[\min(Y_1, Y_2)] = \frac{1}{2+1} = \frac{1}{3}$$

$m = 2$



$$\mathbb{E}[\min(Y_1, \dots, Y_4)] = \frac{1}{4+1} = \frac{1}{5}$$

$m = 4$



Back to counting distinct elements

32, 12, 14, 32, 7, 12, 32, 7, 32, 12, 4

N = # of IDs in the stream = 11, m = # of distinct IDs in the stream = 5

Want to compute number of **distinct** IDs in the stream.

How to do this without storing all the elements?

Distinct Elements – Hashing into $[0, 1]$

Hash function $h: U \rightarrow [0,1]$

Assumption: For all $x \in U$, $h(x) \sim \text{Unif}(0,1)$ and mutually independent

32, 12, 14, 32, 7, 12, 32, 7



$h(32)$, $h(12)$, $h(14)$, $h(32)$, $h(7)$, $h(12)$, $h(32)$, $h(7)$

Distinct Elements – Hashing into $[0, 1]$

Hash function $h: U \rightarrow [0,1]$

Assumption: For all $x \in U$, $h(x) \sim \text{Unif}(0,1)$ and mutually independent

32, 12, 14, 32, 7, 12, 32, 7
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

$h(32), h(12), h(14), h(32), h(7), h(12), h(32), h(7)$

$M=4$ distinct elements

→ 4 i.i.d. RVs $h(32), h(12), h(14), h(7) \sim \text{Unif}(0,1)$

$$\rightarrow \mathbb{E}[\min\{h(32), h(12), h(14), h(7)\}] = \frac{1}{4+1} = \frac{1}{5}$$

Distinct Elements – Hashing into $[0, 1]$

Hash function $h: U \rightarrow [0,1]$

Assumption: For all $x \in U$, $h(x) \sim \text{Unif}(0,1)$ and mutually independent

x_1, x_2, \dots, x_N contains m distinct elements



$h(x_1), h(x_2), \dots, h(x_N)$ contains m i.i.d. rvs $\sim \text{Unif}(0,1)$

and $N - m$ repeats



$$\mathbb{E}[\min\{h(x_1), \dots, h(x_N)\}] = \frac{1}{m + 1}$$

A super duper clever idea!!!!

$$\mathbb{E}[\min\{h(x_1), \dots, h(x_N)\}] = \frac{1}{m+1}$$

$$\text{So } m = \frac{1}{\mathbb{E}[\min\{h(x_1), \dots, h(x_N)\}]} - 1$$



What if $\min\{h(x_1), \dots, h(x_N)\}$ is $\approx \mathbb{E}[\min\{h(x_1), \dots, h(x_N)\}]$?

The MinHash Algorithm – Idea

$$m = \frac{1}{\mathbb{E}[\min\{h(x_1), \dots, h(x_N)\}]} - 1$$

1. Compute $\text{val} = \min\{h(x_1), \dots, h(x_N)\}$
2. Assume that $\text{val} \approx \mathbb{E}[\min\{h(x_1), \dots, h(x_N)\}]$
3. Output as estimate for m : $\text{round}\left(\frac{1}{\text{val}} - 1\right)$



The MinHash Algorithm – Implementation

Algorithm **MinHash**(x_1, x_2, \dots, x_N)

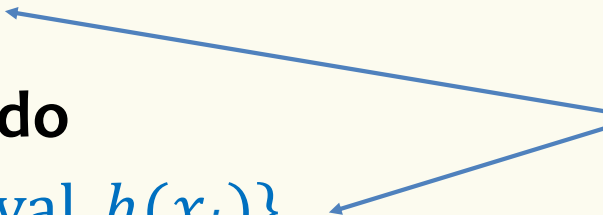
$val \leftarrow \infty$

for $i = 1$ **to** N **do**

$val \leftarrow \min\{val, h(x_i)\}$

return $\text{round}\left(\frac{1}{val} - 1\right)$

Memory cost = just remember val
(with sufficient precision)



MinHash Example

Stream: 13, 25, 19, 25, 19, 19

Hashes: 0.51, 0.26, 0.79, 0.26, 0.79, 0.79

1. Compute $\text{val} = \min\{h(x_1), \dots, h(x_N)\}$
2. Assume that $\text{val} \approx \mathbb{E}[\min\{h(x_1), \dots, h(x_N)\}]$
3. Output $\text{round}\left(\frac{1}{\text{val}} - 1\right)$

**What does
MinHash return?**

MinHash Example II

Stream: 11, 34, 89, 11, 89, 23

Hashes: 0.5, 0.21, 0.94, 0.5, 0.94, 0.1

Output is $\frac{1}{0.1} - 1 = 9$

Clearly, not a very good answer!

Not unlikely: $P(h(x) < 0.1) = 0.1$

The MinHash Algorithm – Problem

Algorithm **MinHash**(x_1, x_2, \dots, x_N)

$\text{val} \leftarrow \infty$

for $i = 1$ **to** N **do**

$\text{val} \leftarrow \min\{\text{val}, h(x_i)\}$

return $\text{round}\left(\frac{1}{\text{val}} - 1\right)$

$\text{val} = \min\{h(x_1), \dots, h(x_N)\}$

$$\mathbb{E}[\text{val}] = \frac{1}{m+1}$$

Problem: val is not $\mathbb{E}[\text{val}]$!
How far is val from $\mathbb{E}[\text{val}]$?

$$\text{Var}(\text{val}) \approx \frac{1}{(m+1)^2}$$

How can we reduce the variance?

Idea: Repetition to reduce variance!

Use k independent hash functions h^1, h^2, \dots, h^k

$$\text{val}_1 = \min\{h^1(x_1), \dots, h^1(x_N)\}$$

$$\text{val}_2 = \min\{h^2(x_1), \dots, h^2(x_N)\}$$

...

$$\text{val}_k = \min\{h^k(x_1), \dots, h^k(x_N)\}$$

$$\text{val} \leftarrow \frac{1}{k} \sum_{i=1}^k \text{val}_i$$

Output as estimate

$$\text{for } m: \quad \text{round}\left(\frac{1}{\text{val}} - 1\right)$$



How can we reduce the variance?

Idea: Repetition to reduce variance!

Use k independent hash functions h^1, h^2, \dots, h^k

Algorithm MinHash (x_1, x_2, \dots, x_N)

$val_1, \dots, val_k \leftarrow \infty$

for $i = 1$ **to** N **do**

for $j = 1$ **to** k **do** $val_j \leftarrow \min\{val_j, h^j(x_i)\}$

$val \leftarrow \frac{1}{k} \sum_{i=1}^k val_i$

return $\text{round}\left(\frac{1}{val} - 1\right)$



$$\text{Var}(val) = \frac{1}{k} \frac{1}{(m+1)^2}$$

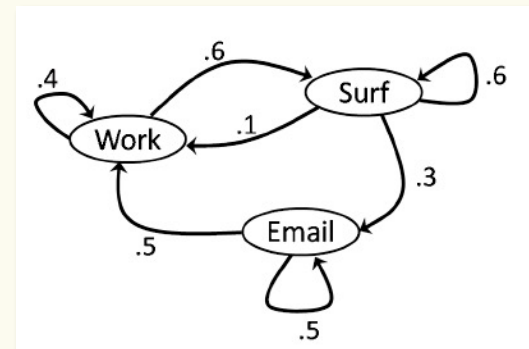
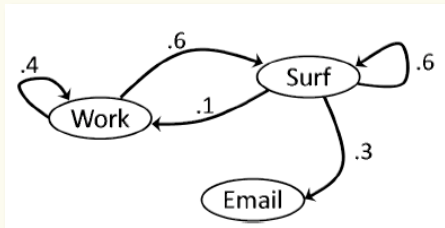
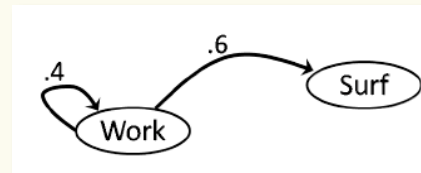
Agenda

- Markov Chains ◀
- Application: PageRank

A typical day in my life....



time $t = 0$



A typical day in my life

How do we interpret this diagram?

At each time step t

– I can be in one of 3 **states**

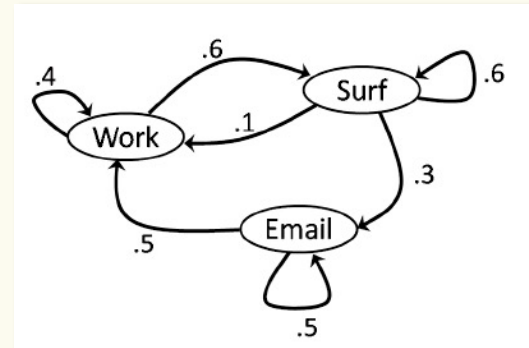
- Work, Surf, Email

– If I am in some state s at time t

- the **labels of out-edges** of s **give the probabilities** of my moving to each of the states at time $t + 1$ (as well as staying the same)

– so **labels on out-edges sum to 1**

e.g. If I am in **Email**, there is a 50-50 chance I will be in each of **Work** or **Email** at the next time step, but I will never be in state **Surf** in the next step.



This kind of random process is called a **Markov Chain**

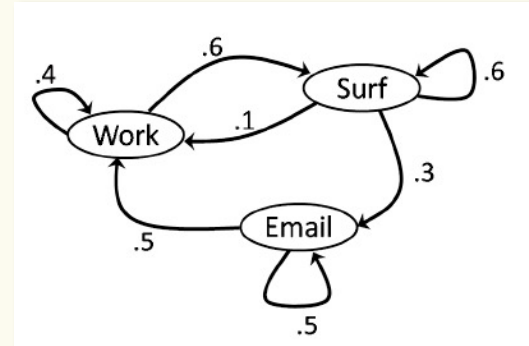
This diagram looks vaguely familiar if you took CSE 311 ...

Markov chains are a special kind of *probabilistic (finite) automaton*

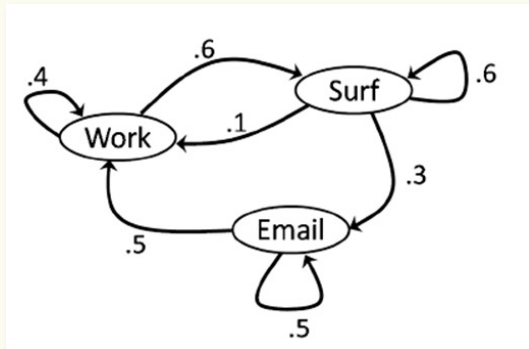
The diagrams look a bit like those of Deterministic Finite Automata (DFAs) you saw in 311 except that...

- There are no input symbols on the edges
 - Think of there being only one kind of input symbol “another tick of the clock” so no need to mark it on the edge
- They have multiple out-edges like an NFA, except that they come with probabilities

But just like DFAs, the only thing they remember about the past is the state they are currently in.



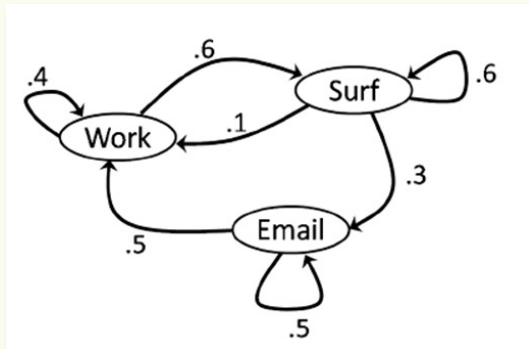
Many interesting questions about Markov Chains



Given: In state *Work* at time $t = 0$

1. What is the probability that I am in state s at time 1?
2. What is the probability that I am in state s at time 2?
3. What is the probability that I am in state s at some time t far in the future?

Many interesting questions about Markov Chains



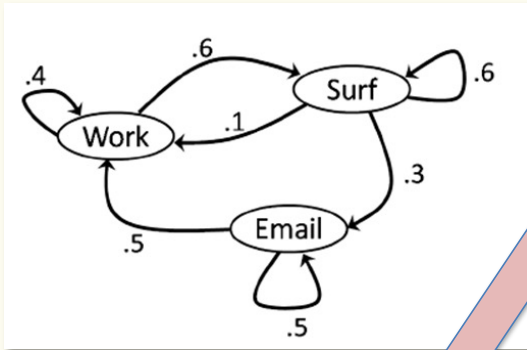
1. What is the probability that I am in state s at time 1?
2. What is the probability that I am in state s at time 2?

Define variable $X^{(t)}$ to be state I am in at time t

Given: In state **Work** at time $t = 0$

t	0	1	2
$P(X^{(t)} = \text{Work})$	1		
$P(X^{(t)} = \text{Surf})$	0		
$P(X^{(t)} = \text{Email})$	0		

An organized way to understand the distribution of $X^{(t)}$

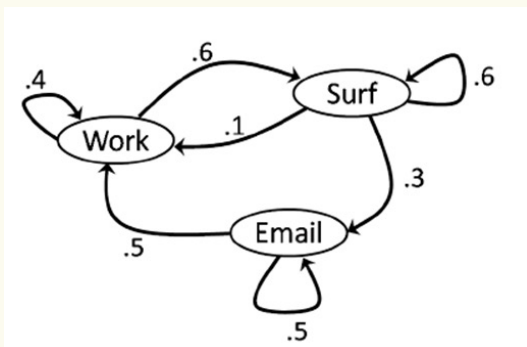


Write as a tuple $(q_W^{(t)}, q_S^{(t)}, q_E^{(t)})$ a.k.a. a row vector:

$$\mathbf{q}^{(t)} = [q_W^{(t)}, q_S^{(t)}, q_E^{(t)}]$$

t	0	1	2
$q_W^{(t)} = P(X^{(t)} = \text{Work})$	1	0.4	$= 0.4 \cdot 0.4 + 0.6 \cdot 0.1 = 0.16 + 0.06 = \mathbf{0.22}$
$q_S^{(t)} = P(X^{(t)} = \text{Surf})$	0	0.6	$= 0.4 \cdot 0.6 + 0.6 \cdot 0.6 = 0.24 + 0.36 = \mathbf{0.60}$
$q_E^{(t)} = P(X^{(t)} = \text{Email})$	0	0	$= 0.4 \cdot 0 + 0.6 \cdot 0.3 = 0 + 0.18 = \mathbf{0.18}$

Describe evolution of $q^{(t)}$ using the “transition probability matrix



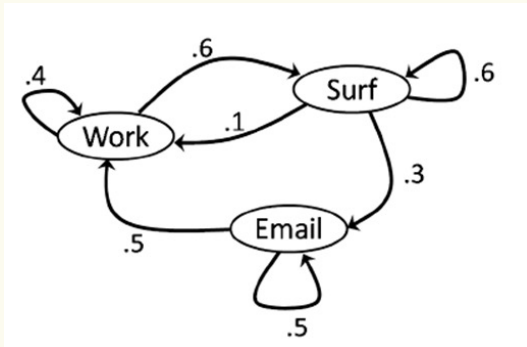
$$\mathbf{M} = \begin{bmatrix} p_{WW} & p_{WS} & p_{WE} \\ p_{SW} & p_{SS} & p_{SE} \\ p_{EW} & p_{ES} & p_{EE} \end{bmatrix} = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0 & 0.5 \end{bmatrix}$$

$$p_{WW} = P(X^{(t+1)} = \text{Work} \mid X^{(t)} = \text{Work})$$

$$p_{SE} = P(X^{(t+1)} = \text{Email} \mid X^{(t)} = \text{Surf})$$

etc

An organized way to understand the distribution of $X^{(t)}$



$$[q_W^{(t+1)}, q_S^{(t+1)}, q_E^{(t+1)}] = [q_W^{(t)}, q_S^{(t)}, q_E^{(t)}] \begin{matrix} M \\ \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0 & 0.5 \end{bmatrix} \end{matrix}$$

Vector-matrix
multiplication

$$q_W^{(t)} = P(X^{(t)} = \text{Work})$$

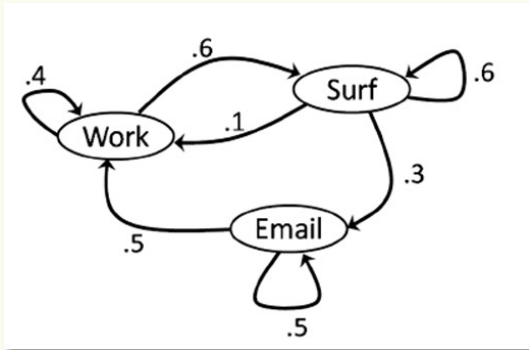
$$q_S^{(t)} = P(X^{(t)} = \text{Surf})$$

$$q_E^{(t)} = P(X^{(t)} = \text{Email})$$

Write $\mathbf{q}^{(t)} = [q_W^{(t)}, q_S^{(t)}, q_E^{(t)}]$

Then for all $t \geq 0$, $\mathbf{q}^{(t+1)} = \mathbf{q}^{(t)} \mathbf{M}$

An organized way to understand the distribution of $X^{(t)}$



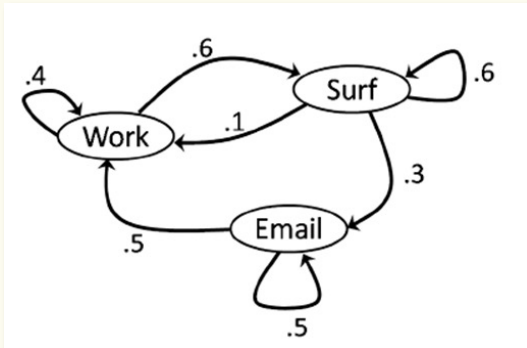
$$[q_W^{(t+1)}, q_S^{(t+1)}, q_E^{(t+1)}] = [q_W^{(t)}, q_S^{(t)}, q_E^{(t)}] \begin{matrix} M \\ \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0 & 0.5 \end{bmatrix} \end{matrix}$$

Write $\mathbf{q}^{(t)} = [q_W^{(t)}, q_S^{(t)}, q_E^{(t)}]$ Then for all $t \geq 0$, $\mathbf{q}^{(t+1)} = \mathbf{q}^{(t)} \mathbf{M}$

So $\mathbf{q}^{(1)} = \mathbf{q}^{(0)} \mathbf{M}$

$$\mathbf{q}^{(2)} = \mathbf{q}^{(1)} \mathbf{M} = (\mathbf{q}^{(0)} \mathbf{M}) \mathbf{M} = \mathbf{q}^{(0)} \mathbf{M}^2$$

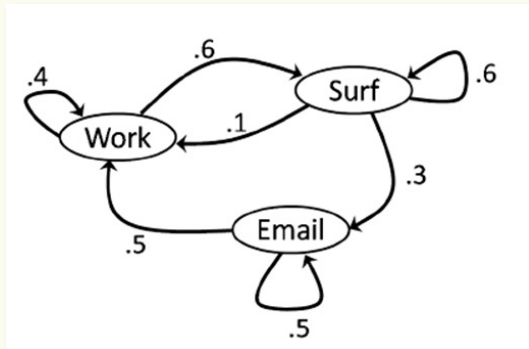
By induction ... we can derive



$$M = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0 & 0.5 \end{bmatrix}$$

$$q^{(t)} = q^{(0)} M^t \text{ for all } t \geq 0$$

Many interesting questions about Markov Chains



Given: In state **Work** at time $t = 0$

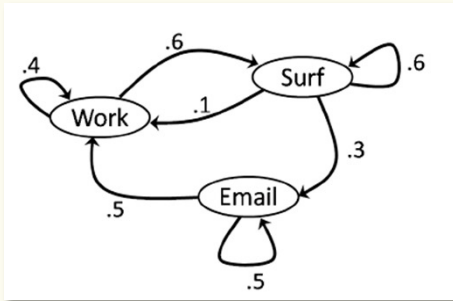
1. What is the probability that I am in state s at time 1?
2. What is the probability that I am in state s at time 2?
3. What is the probability that I am in state s at some time t far in the future?

$$\mathbf{q}^{(t)} = \mathbf{q}^{(0)} \mathbf{M}^t \text{ for all } t \geq 0$$

What does $\mathbf{q}^{(t)}$ look like for really big t ?

$$q^{(t)} = q^{(0)} M^t \text{ for all } t \geq 0$$

M^t as t grows



M

$$\begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0 & 0.5 \end{bmatrix}$$

M^2

	W	S	E
W	.22	.6	.18
S	.25	.42	.33
E	.45	.3	.25

M^3

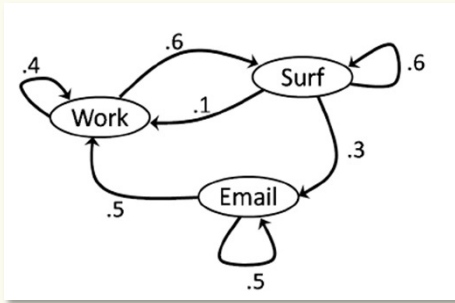
	W	S	E
W	.238	.492	.270
S	.307	.402	.291
E	.335	.450	.215

M^{10}

	W	S	E
W	.2940	.4413	.2648
S	.2942	.4411	.2648
E	.2942	.4413	.2648

$$q^{(t)} = q^{(0)} M^t \text{ for all } t \geq 0$$

M^t as t grows



$$M = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0 & 0.5 \end{bmatrix}$$

$$M^2 = \begin{matrix} & W & S & E \\ W & \begin{pmatrix} .22 & .6 & .18 \end{pmatrix} \\ S & \begin{pmatrix} .25 & .42 & .33 \end{pmatrix} \\ E & \begin{pmatrix} .45 & .3 & .25 \end{pmatrix} \end{matrix}$$

$$M^3 = \begin{matrix} & W & S & E \\ W & \begin{pmatrix} .238 & .492 & .270 \end{pmatrix} \\ S & \begin{pmatrix} .307 & .402 & .291 \end{pmatrix} \\ E & \begin{pmatrix} .335 & .450 & .215 \end{pmatrix} \end{matrix}$$

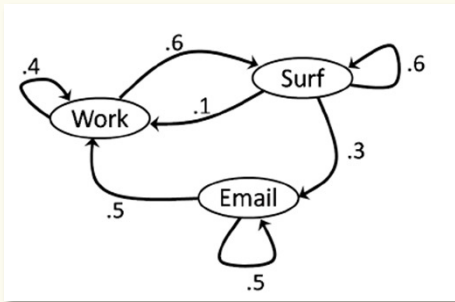
$$M^{10} = \begin{matrix} & W & S & E \\ W & \begin{pmatrix} .2940 & .4413 & .2648 \end{pmatrix} \\ S & \begin{pmatrix} .2942 & .4411 & .2648 \end{pmatrix} \\ E & \begin{pmatrix} .2942 & .4413 & .2648 \end{pmatrix} \end{matrix}$$

$$M^{30} = \begin{matrix} & W & S & E \\ W & \begin{pmatrix} .29411764705 & .44117647059 & .26470588235 \end{pmatrix} \\ S & \begin{pmatrix} .29411764706 & .44117647058 & .26470588235 \end{pmatrix} \\ E & \begin{pmatrix} .29411764706 & .44117647059 & .26470588235 \end{pmatrix} \end{matrix}$$

$$M^{60} = \begin{matrix} & W & S & E \\ W & \begin{pmatrix} .294117647058823 & .441176470588235 & .264705882352941 \end{pmatrix} \\ S & \begin{pmatrix} .294117647068823 & .441176470588235 & .264705882352941 \end{pmatrix} \\ E & \begin{pmatrix} .294117647068823 & .441176470588235 & .264705882352941 \end{pmatrix} \end{matrix}$$

What does this say about $q^{(t)}$

M^t as t grows



$$q^{(60)} = q^{(0)} M^{60}$$

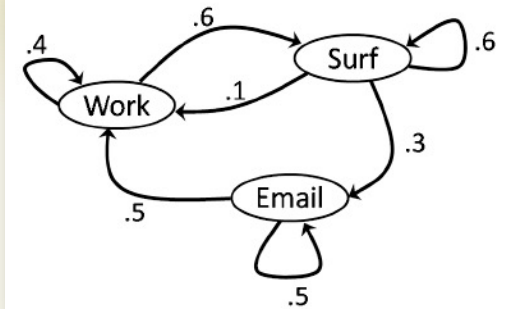
$$[q_W^{(0)}, q_S^{(0)}, q_E^{(0)}] \cdot \begin{pmatrix} .294117647058823 & .441176470588235 & .264705882352941 \\ .294117647068823 & .441176470588235 & .264705882352941 \\ .294117647068823 & .441176470588235 & .264705882352941 \end{pmatrix} = [q_W^{(60)}, q_S^{(60)}, q_E^{(60)}]$$

- In the long run, the starting state doesn't really matter!!
- So in a long stretch of time, chance I'm surfing the web is:

- Suppose that we believe that the distribution on states converges to some fixed probability vector $\boldsymbol{\pi} = (\pi_W, \pi_S, \pi_E)$
- Can we figure out what $\boldsymbol{\pi}$ is just by looking at M ?

Observation

If $q^{(t+1)} = q^{(t)}$ then it will never change again!



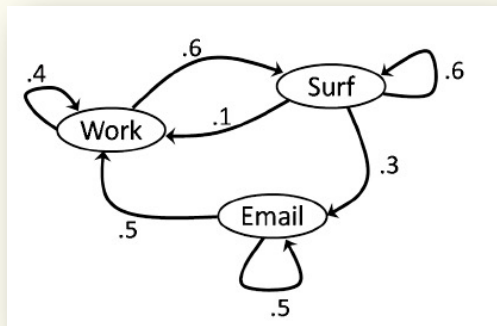
Called a **stationary distribution** and has a special name

$$\boldsymbol{\pi} = (\pi_W, \pi_S, \pi_E)$$

Solution to $\boldsymbol{\pi} = \boldsymbol{\pi} M$

Solving for Stationary Distribution

$$(\pi_W, \pi_S, \pi_E) \begin{pmatrix} 0.4 & 0.6 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0 & 0.5 \end{pmatrix} = (\pi_W, \pi_S, \pi_E)$$



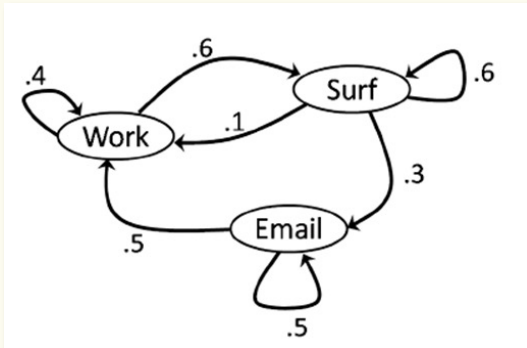
$$\pi_W \cdot 0.4 + \pi_S \cdot 0.1 + \pi_E \cdot 0.5 = \pi_W$$

$$\pi_W \cdot 0.6 + \pi_S \cdot 0.6 + \pi_E \cdot 0 = \pi_S$$

$$\pi_W \cdot 0 + \pi_S \cdot 0.3 + \pi_E \cdot 0.5 = \pi_E$$

$$\pi_W + \pi_S + \pi_E = 1$$

Computing the Stationary Distribution



$$[\pi_W, \pi_S, \pi_E] \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0 & 0.5 \end{bmatrix} = [\pi_W, \pi_S, \pi_E]$$

Solve system of equations:

Stationary Distribution satisfies

- $\boldsymbol{\pi} = \boldsymbol{\pi M}$, where $\boldsymbol{\pi} = (\pi_W, \pi_S, \pi_E)$
- $\pi_W + \pi_S + \pi_E = 1$

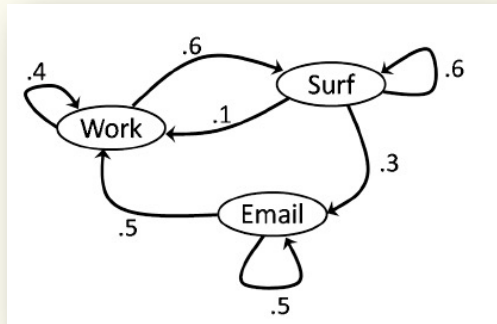
$$\Rightarrow \pi_W = \frac{10}{34}, \pi_S = \frac{15}{34}, \pi_E = \frac{9}{34}$$

$$\left\{ \begin{array}{l} 0.4 \cdot \pi_W + 0.1 \cdot \pi_S + 0.5 \cdot \pi_E = \pi_W \\ 0.6 \cdot \pi_W + 0.6 \cdot \pi_S = \pi_S \\ 0.3 \cdot \pi_S + 0.5 \cdot \pi_E = \pi_E \end{array} \right.$$

$$\pi_W + \pi_S + \pi_E = 1$$

Solving for Stationary Distribution

$$(\pi_W, \pi_S, \pi_E) \begin{pmatrix} 0.4 & 0.6 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0 & 0.5 \end{pmatrix} = (\pi_W, \pi_S, \pi_E)$$



$$\pi_W \cdot 0.4 + \pi_S \cdot 0.1 + \pi_E \cdot 0.5 = \pi_W$$

$$\pi_W \cdot 0.6 + \pi_S \cdot 0.6 + \pi_E \cdot 0 = \pi_S$$

$$\pi_W \cdot 0 + \pi_S \cdot 0.3 + \pi_E \cdot 0.5 = \pi_E$$

$$\pi_W + \pi_S + \pi_E = 1$$

$$\Rightarrow \pi_W = \frac{10}{34}, \pi_S = \frac{15}{34}, \pi_E = \frac{9}{34}$$

As $t \rightarrow \infty$, $q^{(t)} \rightarrow \pi$ no matter what distribution $q^{(0)}$ is !!

Markov Chains recap

- A set of n **states** $\{1, 2, 3, \dots, n\}$
- The state at time t is denoted by $X^{(t)}$
- A square **transition matrix** M , dimension $n \times n$

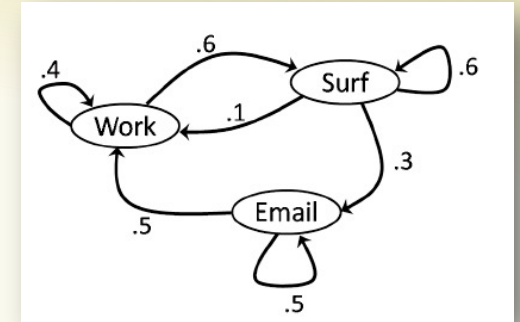
$$M_{ij} = P(X^{(t+1)} = j \mid X^{(t)} = i)$$

$$\begin{pmatrix} 0.4 & 0.6 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0 & 0.5 \end{pmatrix}$$

- M^t_{ij} = Pr (in state j after t steps | start in state i).
- Nice Markov chains are not sensitive to initial distribution of states. $M^t \rightarrow W$, where all rows in W are the same probability vector π
- A **stationary distribution** π is the solution to:

$$\pi = \pi M, \text{ normalized so that } \sum_{i \in [n]} \pi_i = 1$$

$$M^{60} \begin{matrix} & W & S & E \end{matrix} \begin{matrix} W \\ S \\ E \end{matrix} \begin{pmatrix} .294117647058823 & .441176470588235 & .264705882352941 \\ .294117647068823 & .441176470588235 & .264705882352941 \\ .294117647068823 & .441176470588235 & .264705882352941 \end{pmatrix}$$



The Fundamental Theorem of Markov Chains

Theorem. Any nice* Markov chain has a unique stationary distribution π .

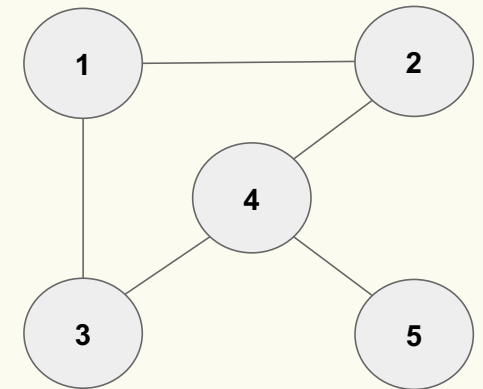
Moreover, as $t \rightarrow \infty$, for all i, j , $\lim_{t \rightarrow \infty} M_{ij}^t = \pi_j$

**aperiodic and irreducible: these concepts are beyond us but they turn out to cover a very large class of Markov chains of practical importance.*

Another Example: Random Walks

Suppose we start at node 1, and at each step transition to a neighboring node with equal probability.

This is called a “random walk” on this graph.



Agenda

- Markov Chains
- Application: PageRank ◀

PageRank: Some History

The year was 1997

- Bill Clinton in the White House
- Deep Blue beat world chess champion (Kasparov)

The Internet was not like it was today. Finding stuff was hard!

- In Nov 1997, only one of the top 4 search engines actually found itself when you searched for it

The Problem

Search engines worked by matching words in your queries to documents.

Not bad in theory, but in practice there are lots of documents that match a query.

- Search for ‘Bill Clinton’, top result is ‘Bill Clinton Joke of the Day’
- Susceptible to spammers and advertisers

The Fix: Ranking Results

- Start by doing filtering to relevant documents (with decent textual match).
- Then **rank** the results based on some measure of ‘quality’ or ‘authority’.

Key question: How to define ‘quality’ or ‘authority’?

Enter two groups:

- Jon Kleinberg (professor at Cornell)
- Larry Page and Sergey Brin (Ph.D. students at Stanford)

Both groups had the same brilliant idea

Larry Page and Sergey Brin (Ph.D. students at Stanford)

- Took the idea and founded Google, making billions



Jon Kleinberg (professor at Cornell)

- MacArthur genius prize, Nevanlinna Prize and many other academic honors

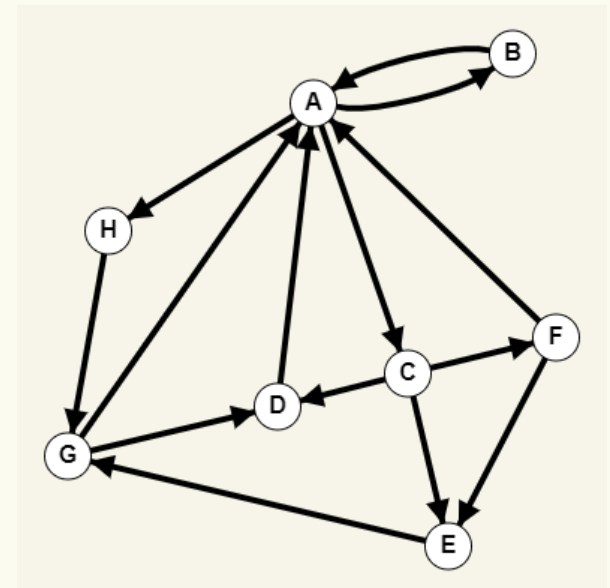


PageRank - Idea

Take into account the directed graph structure of the web.

Use **hyperlink analysis** to compute what pages are high quality or have high authority.

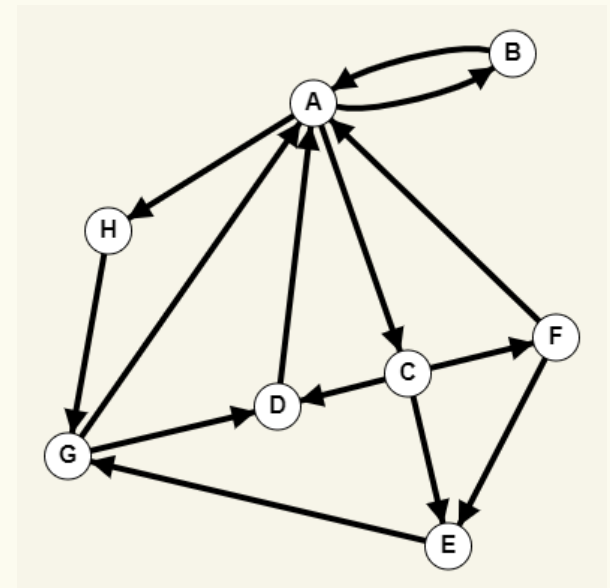
Trust the Internet itself to define what is useful via its links.



PageRank - Idea

Idea 1: Think of each link as a citation
“vote of quality”

Rank pages by in-degree?



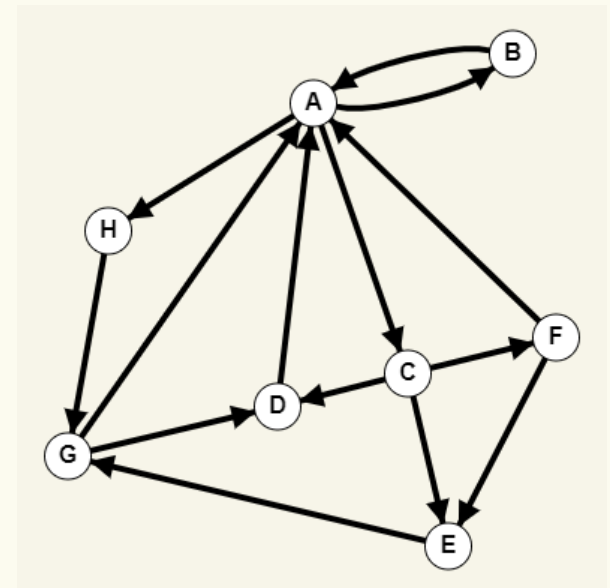
PageRank - Idea

Idea 1: Think of each link as a citation
“vote of quality”

Rank pages by in-degree?

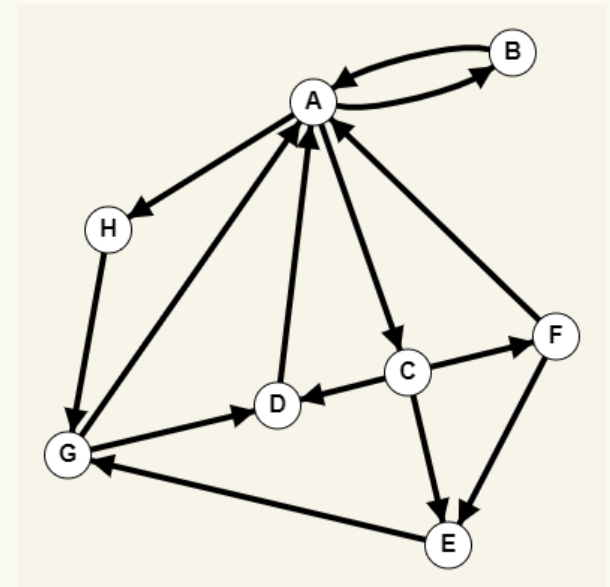
Problems:

- Spamming
- Not all links created equal
- Some linkers are not discriminating



PageRank - Idea

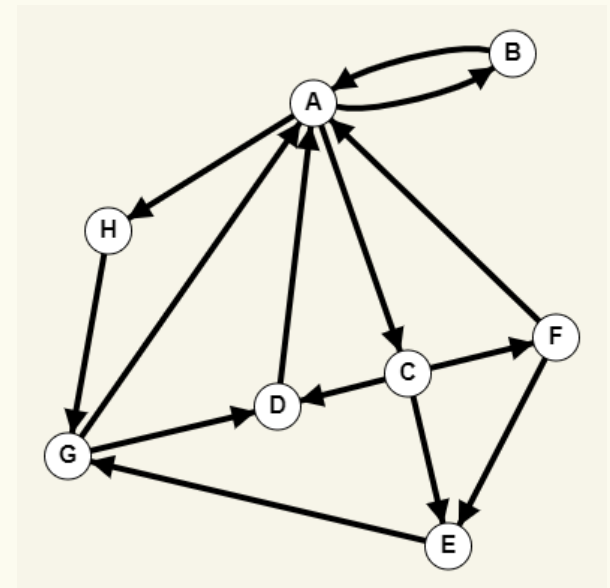
Idea 2 : Perhaps we should weight the links somehow and then use the weights of the in-links to rank pages



Inching towards PageRank

1. Web page has high quality if it's linked to by lots of high quality pages
2. A page is high quality if it links to lots of high quality pages

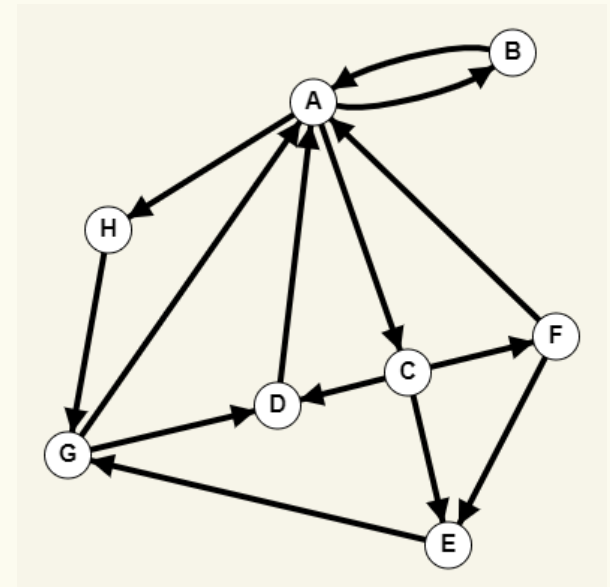
That's a recursive definition!



Inching towards PageRank



- If web page x has d outgoing links, one of which goes to y , this contributes $1/d$ to the importance of y
- But $1/d$ of what?
We want to take into account the importance of x too...
... so it actually contributes $1/d$ of the importance of x



This gives the following equations

Idea: Use the transition matrix M defined by a *random walk* on the web to compute quality of webpages.

Namely: Find q such that $qM = q$ **Seem familiar?**



This gives the following equations

Idea: Use the transition matrix M defined by a *random walk* on the web to compute quality of webpages.

Namely: Find q such that $qM = q$ **Seem familiar?**



This is the stationary distribution for the Markov chain defined by a random web surfer

- Starts at some node (webpage) and randomly follows a link to another.
- Use stationary distribution of her surfing patterns after a long time as notion of quality

Issues with PageRank

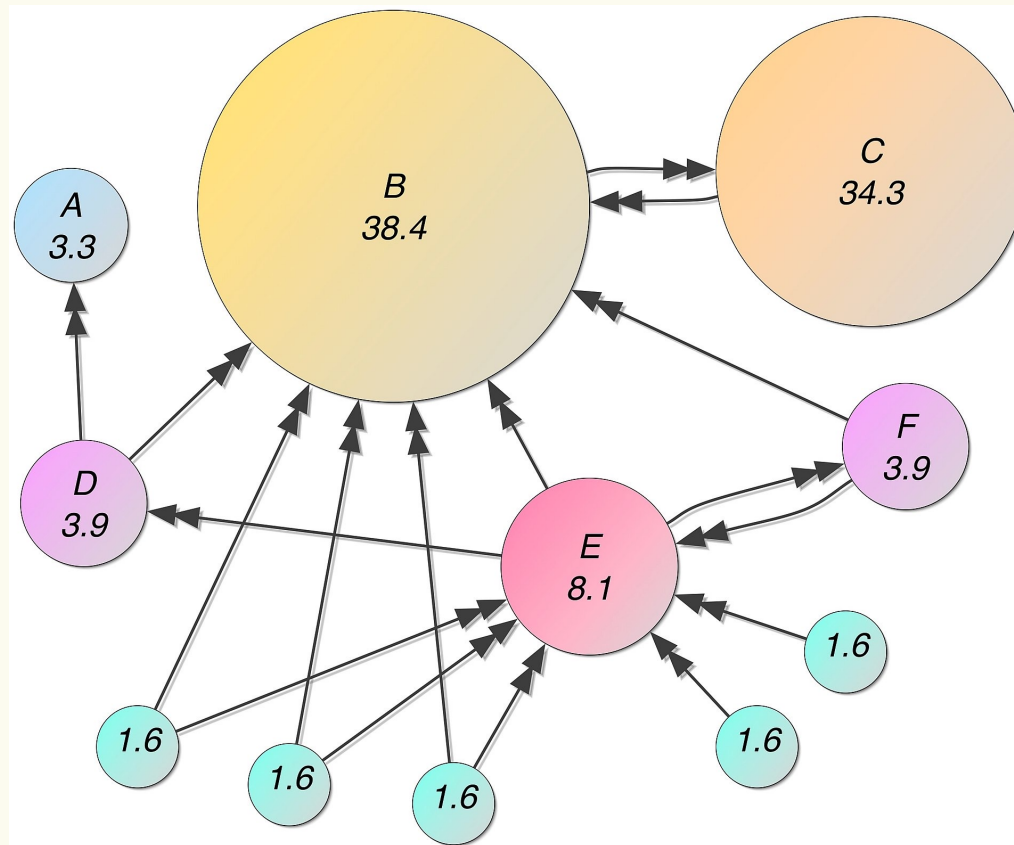
- How to handle dangling nodes (dead ends that don't link to anything) ?
- How to handle Rank sinks – group of pages that only link to each other ?

Both solutions can be solved by “teleportation”

Final PageRank Algorithm

1. Make a Markov Chain with one state for each webpage on the Internet with the transition probabilities $M_{ij} = \frac{1}{outdeg(i)}$.
2. Use a modified random walk. At each point in time if the surfer is at some webpage i :
 - If i has outlinks:
 - With probability p , take a step to one of the neighbors of i (equally likely)
 - With probability $1 - p$, “teleport” to a uniformly random page in the whole Internet.
 - Otherwise, always “teleport”
3. Compute stationary distribution π of this perturbed Markov chain.
4. Define the PageRank of a webpage i as the stationary probability π_i .
5. Find all pages with decent textual match to search and then order those pages by PageRank!

PageRank - Example



It Gets More Complicated

While this basic algorithm was the defining idea that launched Google on their path to success, this is far from the end to optimizing search

Nowadays, Google and other web search engines have a LOT more secret sauce to rank pages, most of which they don't reveal 1) for competitive advantage and 2) to avoid gaming of their algorithms.