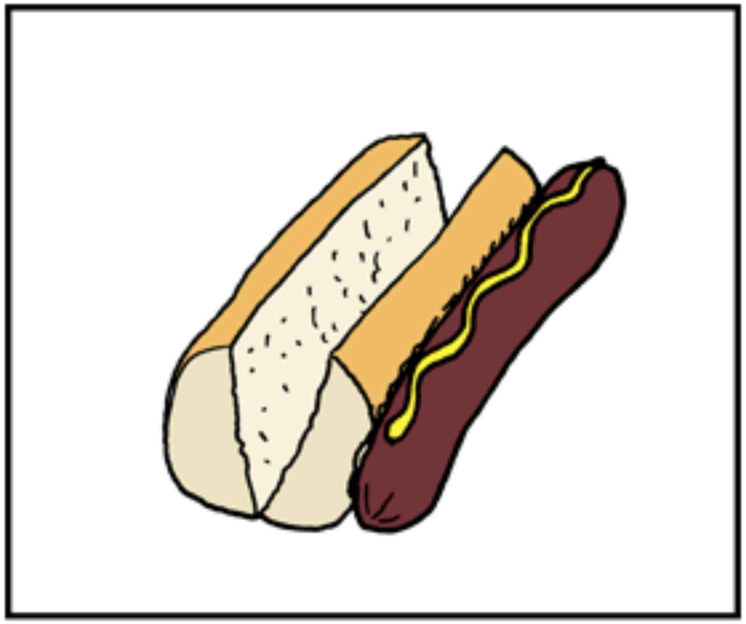


REVERSE POLISH SAUSAGE

|< < PREV RANDOM NEXT > >|

PERMANENT LINK TO THIS COMIC: HTTP://XKCD.COM/645/
IMAGE URL (FOR HOTLINKING/EMBEDDING): HTTP://IMGS.XKCD.COM/COMICS/RPS.PNG

Link Analysis

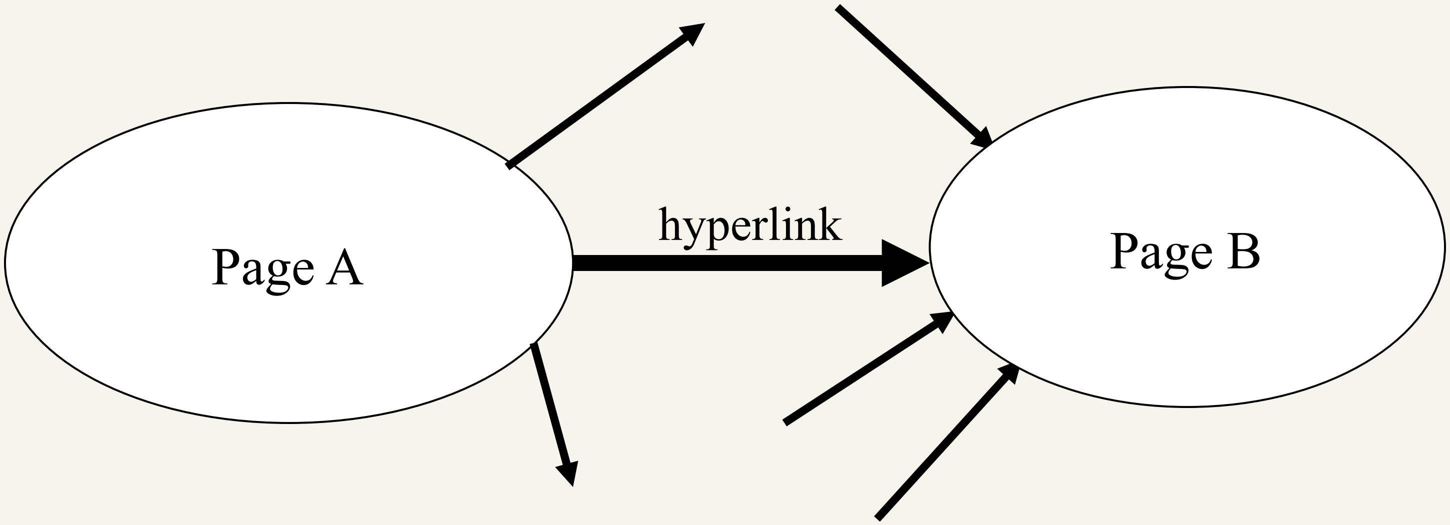
David Kauchak
cs160
Fall 2009

adapted from:
http://www.stanford.edu/class/cs276/handouts/lecture15-linkanalysis.ppt
http://webcourse.cs.technion.ac.il/236522/Spring2007/ho/WCFiles/Tutorial05.ppt

Administrative

- Course feedback
  - homeworks
- Midterm review today
- Assign 4 out soon

The Web as a Directed Graph

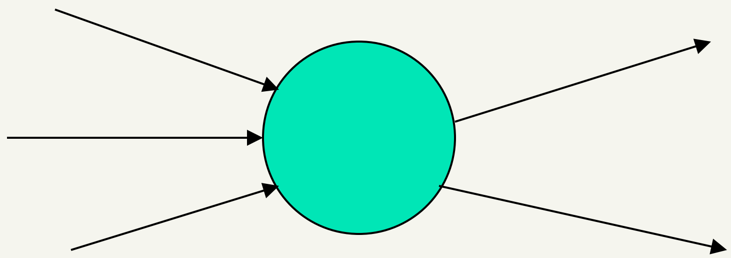


A hyperlink between pages denotes author perceived relevance AND importance

How can we use this information?

Query-independent ordering

- First generation: using link counts as simple measures of popularity
- Two basic suggestions:
  - Undirected popularity:
    - Each page gets a score = the number of in-links plus the number of out-links (3+2=5)
  - Directed popularity:
    - Score of a page = number of its in-links (3)



problems?

What is pagerank?

- The random surfer model
- Imagine a user surfing the web randomly using a web browser
- The pagerank score of a page is the probability that that user will visit a given page

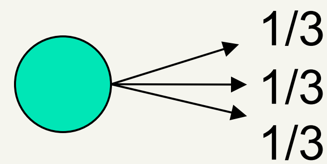
http://images.clipartof.com/small/7872-Clipart-Picture-Of-A-World-Earth-Globe-Mascot-Cartoon-Character-Surfing-On-A-Blue-And-Yellow-Surfboard.jpg

Random surfer model

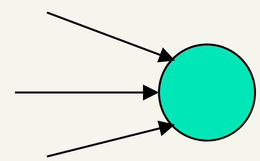
- We want to model the behavior of a “random” user interfacing the web through a browser
- Model is independent of content (i.e. just graph structure)
- What types of behavior should we model and how?
  - Where to start
  - Following links on a page
  - Typing in a url (bookmarks)
  - What happens if we get a page with no outlinks
  - Back button on browser

Random surfer model

- Start at a random page
- Go out of the current page along one of the links on that page, equiprobably



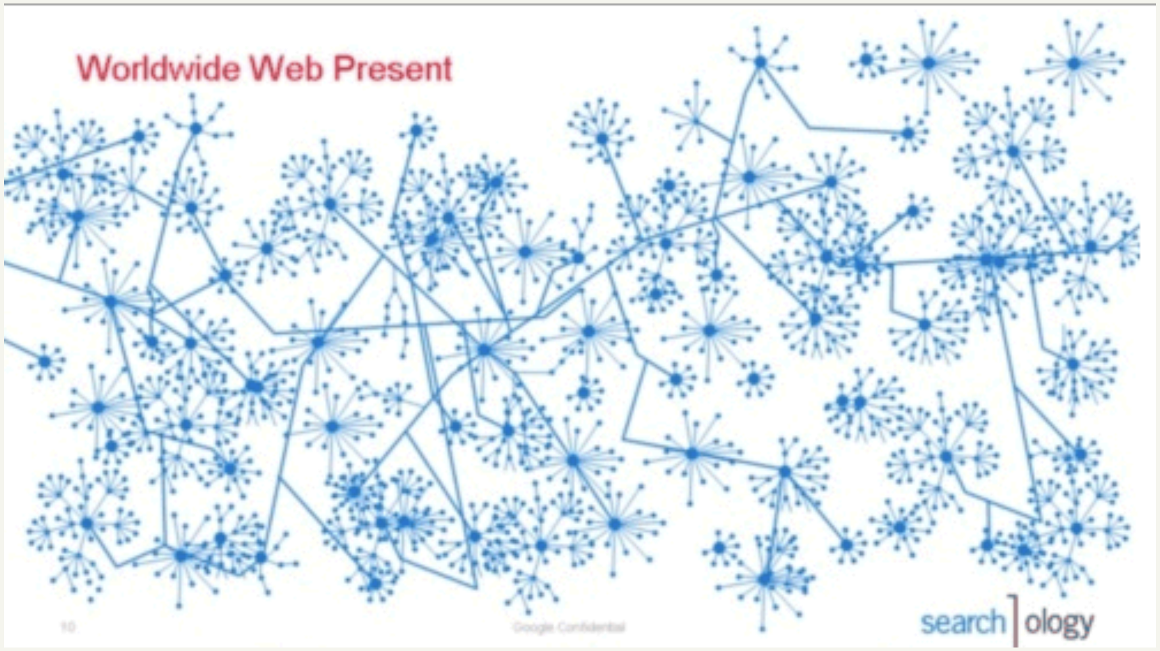
- "Teleporting"
  - If a page has no outlinks always jump to random page



  - With some fixed probability, randomly jump to any other page, otherwise follow links

The questions…

- Given a graph and a teleporting probability, we have some probability of visiting every page
- What is that probability for each page in the graph?



http://3.bp.blogspot.com/_ZaGO7GjCqAI/Rkyo5uCmBdI/AAAAAAAACLo/zsHdSlKc-q4/s640/searchology-web-graph.png

Markov process

- A markov process is defined by:
  - $x_1, x_2, \ldots, x_n$ a set of states
  - An *n* by *n* transition matrix describing the probability of transitioning to state x_j given that you're in state x_i

$$\begin{array}{c|c} & x_1 \quad x_2 \quad \cdots \quad x_n \\ \hline \begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array} & \begin{bmatrix} \\ x_{ij}: p(x_j|x_i) \\ \\ \end{bmatrix} \end{array}$$

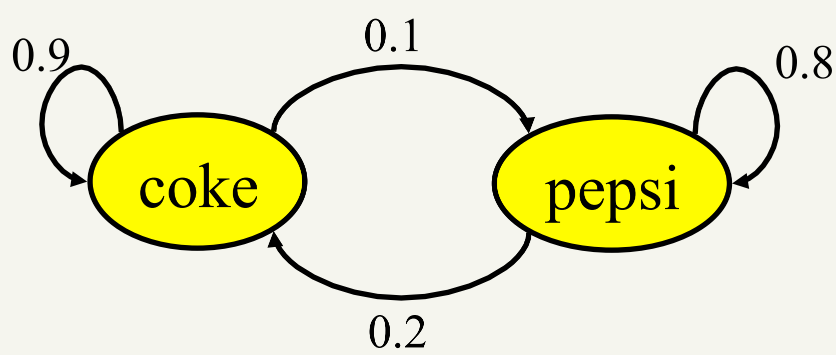
rows must sum to 1

Markov Process
Coke vs. Pepsi Example

- Given that a person's last cola purchase was Coke, there is a 90% chance that his next cola purchase will also be Coke.
- If a person's last cola purchase was Pepsi, there is an 80% chance that his next cola purchase will also be Pepsi.

transition matrix:

$$\begin{array}{cc} & \begin{array}{cc} \text{coke} & \text{pepsi} \end{array} \\ \begin{array}{c} \text{coke} \\ \text{pepsi} \end{array} & \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} \end{array}$$



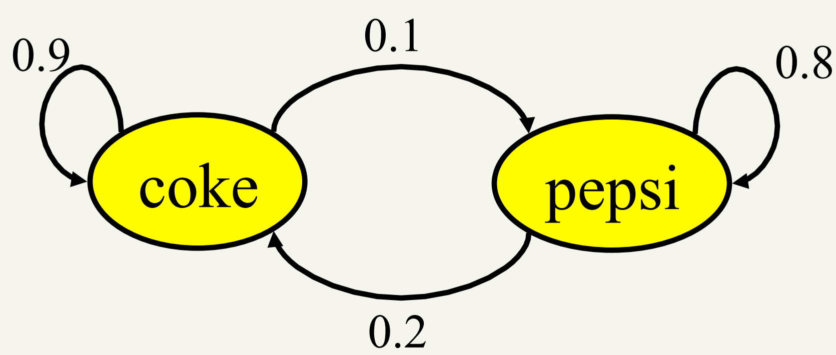
Markov Process

Coke vs. Pepsi Example (cont)

Given that a person is currently a Pepsi purchaser, what is the probability that he will purchase Coke two purchases from now?

transition matrix:

$$
\begin{array}{cc}
 & \begin{array}{cc} \text{coke} & \text{pepsi} \end{array} \\
\begin{array}{c} \text{coke} \\ \text{pepsi} \end{array} & \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}
\end{array}
$$



Markov Process

Coke vs. Pepsi Example (cont)

Given that a person is currently a Pepsi purchaser, what is the probability that he will purchase Coke two purchases from now?

$$P = \begin{array}{cc} \text{coke} & \text{pepsi} \\ \left[\begin{array}{cc} 0.9 & 0.1 \\ 0.2 & 0.8 \end{array}\right] \end{array} \begin{array}{cc} \text{coke} & \text{pepsi} \\ \left[\begin{array}{cc} 0.9 & 0.1 \\ 0.2 & 0.8 \end{array}\right] \end{array} = \begin{array}{ccc} \text{coke} & \text{pepsi} & \\ \left[\begin{array}{cc} 0.83 & 0.17 \\ 0.34 & 0.66 \end{array}\right] & & \begin{array}{c} \text{coke} \\ \text{pepsi} \end{array} \end{array}$$

Pepsi → ?   ? → Coke

Markov Process

Coke vs. Pepsi Example (cont)

Given that a person is currently a Coke purchaser, what is the probability that he will purchase Pepsi **three** purchases from now?

$$P^3 = \begin{array}{c} \begin{array}{cc} \text{coke} & \text{pepsi} \end{array} \\ \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{cc} \text{coke} & \text{pepsi} \end{array} \\ \begin{bmatrix} 0.83 & 0.17 \\ 0.34 & 0.66 \end{bmatrix} \end{array} = \begin{array}{cc} \begin{array}{cc} \text{coke} & \text{pepsi} \end{array} & \\ \begin{bmatrix} 0.781 & 0.219 \\ 0.438 & 0.562 \end{bmatrix} & \begin{array}{c} \text{coke} \\ \text{pepsi} \end{array} \end{array}$$

Steady state

- In general, we can calculate the probability after *n* purchases as P^n
- We might also ask the question, what is the probability of being in state coke or pepsi?
- This is described by the steady state distribution of a markov process
  - Note, this is a distribution over *states* not state transitions
- How might we obtain this?

Steady state

- We have some initial **vector** describing the probabilities of each starting state
  - For example, coke drinker: x = [1, 0]

$$\begin{array}{cccc} & \text{coke} \quad \text{pepsi} & \text{coke} \quad \text{pepsi} & \text{coke} \quad \text{pepsi} \\ xP^3 = & \begin{bmatrix} 1 & 0 \end{bmatrix} & \begin{bmatrix} 0.781 & 0.219 \\ 0.438 & 0.562 \end{bmatrix} & = \begin{bmatrix} 0.781 & 0.219 \end{bmatrix} \end{array}$$

  - We could have said a person that drinks coke 80% of the time, x = [0.80, 0.20]

$$xP^3 = \begin{bmatrix} .8 & .2 \end{bmatrix} \begin{bmatrix} 0.781 & 0.219 \\ 0.438 & 0.562 \end{bmatrix} = \begin{bmatrix} 0.712 & 0.288 \end{bmatrix}$$

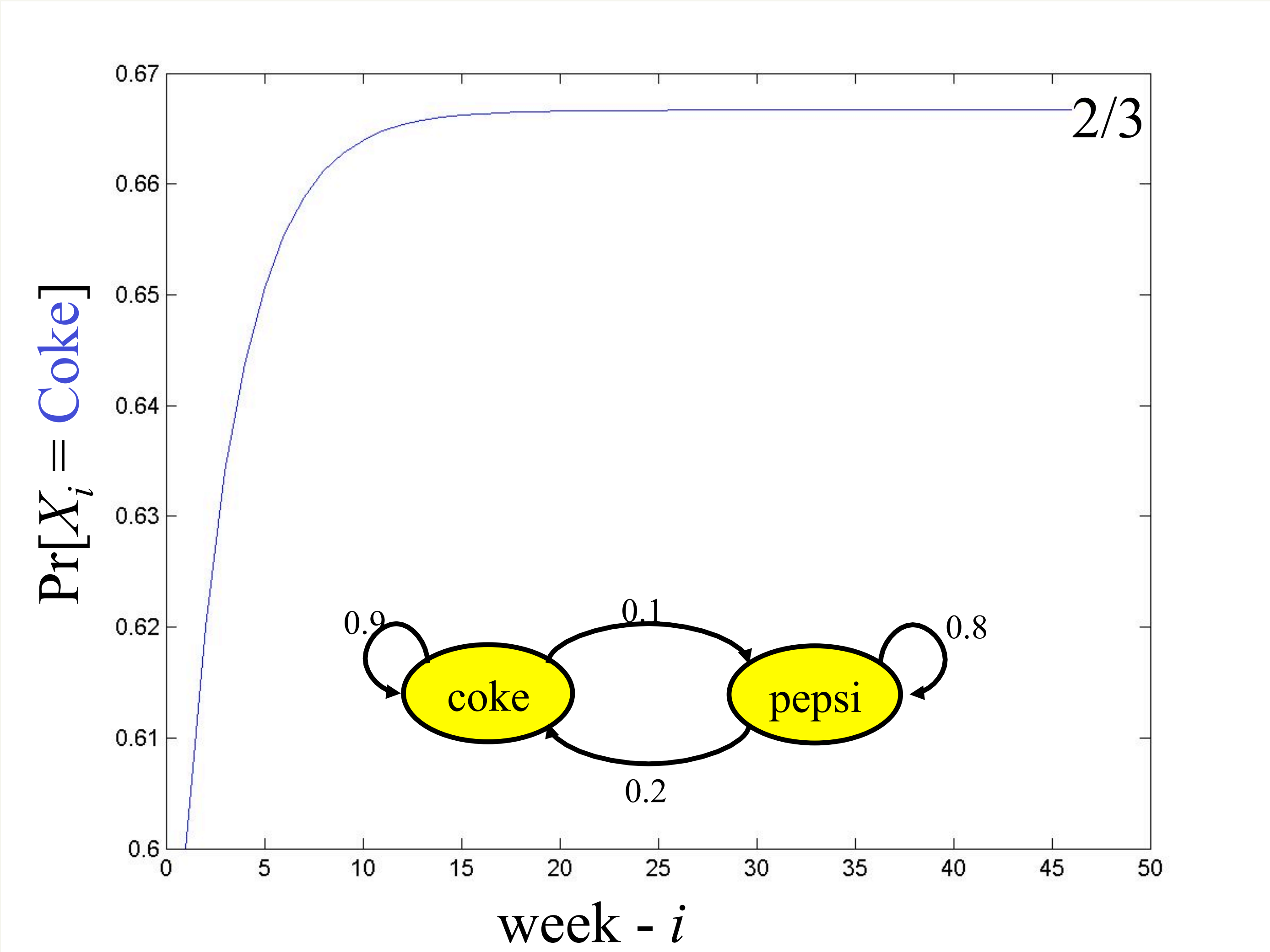
Steady state

- Most common
  - start with some initial x
  - xP, xP^2, xP^3, xP^4, …
  - For many processes, this will eventually settle

Markov Process

Coke vs. Pepsi Example (cont)

Simulation:



Back to pagerank

- Can we use this to solve our random surfer problem?
  - States are web pages
  - Transitions matrix is the probability of transitioning to page A given at page B
  - "Teleport" operation makes sure that we can get to any page from any other page

- Matrix is much bigger, but same approach is taken…
  - P, P^2, P^3, P^4, …

Pagerank summary

- Preprocessing:
  - Given a graph of links, build matrix **P**
  - From it compute **steady state** of each state
  - An entry is a number between 0 and 1: the pagerank of a page
- Query processing:
  - Retrieve pages meeting query
  - Integrate pagerank score with other scoring (e.g. tf-idf)
  - Rank pages by this combined score

The reality

- Pagerank is used in google, but so are many other clever heuristics

Pagerank: Issues and Variants

- How realistic is the random surfer model?
  - Modeling the back button
  - Surfer behavior sharply skewed towards short paths
  - Search engines, bookmarks & directories make jumps non-random
- Note that all of these just vary how we create our initial transition probability matrix

Biased surfer models

- Random teleport to any page is not very reasonable
- Biased Surfer Models
  - Weight edge traversal probabilities based on match with topic/query (non-uniform edge selection)
  - Bias jumps to pages on topic (e.g., based on personal bookmarks & categories of interest)

Topic Specific Pagerank

- Conceptually, we use a random surfer who teleports, with say 10% probability, using the following rule:
  - Selects a category based on a query & user-specific distribution over the categories
  - Teleport to a page uniformly at random within the chosen category
- What is the challenge?

Topic Specific Pagerank

- Ideas?
- **Offline**:Compute pageranks for *individual* categories
  - Query independent as before
  - Each page has multiple pagerank scores – one for each category, with teleportation only to that category
- **Online**: Distribution of weights over categories computed by query context classification
  - Generate a dynamic pagerank score for each page - weighted sum of category-specific pageranks

Spamming pagerank

Other link analysis

- Pagerank is not the only link analysis method
  - Many improvements/variations of pagerank
  - Hubs and authorities

Midterm review: general notes

- We’ve covered a lot of material
  - Anything from lecture, readings, homeworks and assignments is fair game
  - Today’s material NOT on the midterm
- T/F, short answer, short workthrough problems
- Some questions like homework, but also many “conceptual” questions
- Won’t need calculator
- 1hr and 15 mins goes by fast
- Grading for the class

Midterm review

- indexes
  - representation
  - skip pointers
  - why we need an index

- boolean index
  - merge operation
  - query optimization
  - phrase queries (query proximity)

Midterm review

- index construction
  - implementing efficiently
  - sort-based
  - spimi
  - distributed index contruction
    - map reduce
  - dealing with data that refreshes frequently

Midterm review

- index compression
  - dictionary compression
    - variable width entries
    - blocking
    - front-coding
  - postings list compression
    - gaps
    - gap encoding/compression

Midterm review

- documents in the index
- text preprocessing
  - tokenization
  - text normalization
  - stop lists
  - java regex
- computer hardware basics
- data set analysis
  - statistics
  - heaps’ law
  - zipf's law

Midterm review

- ranked retrieval
  - vector space representation and retrieval
  - representing documents and queries as vectors
  - cosine similarity measure
  - normalization/reweighting techniques
  - calculating similarities from index
  - Speeding up ranking calculations
    - approximate top K approaches (e.g. champion lists)
    - cluster pruning

Midterm review

- Evaluation
  - precision
  - recall
  - F1
  - 11-point precision
  - MAP
  - Kappa statistic
  - A/B testing

Midterm review

- snippet/summary generation
- spelling correction
  - edit distance
  - word n-grams
  - jaccard coefficient
- relevance feedback

Midterm review

- web
  - basic web search engine
  - spam
  - estimating the size of the web (or the size of a search engine's index)
  - duplicate detection at web scale
  - crawling