

CS 277, Data Mining

Web Data Analysis

Padhraic Smyth

Department of Computer Science

Bren School of Information and Computer Sciences

University of California, Irvine

Web Mining

Web = a potentially enormous “data set” for data mining

Multiple aspects of “Web mining”

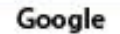
1. Web Content
e.g., categorizing Web pages based on their text content
2. Web Connectivity/Link Analysis
e.g., characterizing distributions on path lengths between pages
e.g., determining importance of pages from graph structure
3. Web Usage
e.g., understanding user behavior from Web and search logs
4. Web Advertising
e.g., algorithms for optimizing which ads to show which users

All are interconnected/interdependent

- E.g., Google (and most search engines) use both content and connectivity

Different Aspects of User Data on the Web

| Type | Content | How Collected |
|---------------------|---|--|
| Navigation data | URLs, time-stamps, etc | Web logs |
| Search query data | Text strings | Web logs |
| Transaction data | Item purchased, credit card, home address, etc | Server-side database |
| Registration data | Name, address, demographics | Server-side database |
| Cookie files | Information to link user to session | File “dropped” by Web server on client machine |
| User-generated text | Blogs, tweets, product reviews, customer emails | Server-side database or Web crawling |
| Social network data | Who is connected to who | Social networking sites |
| Location data | Location (x,y) at time t | IP address or GPS |



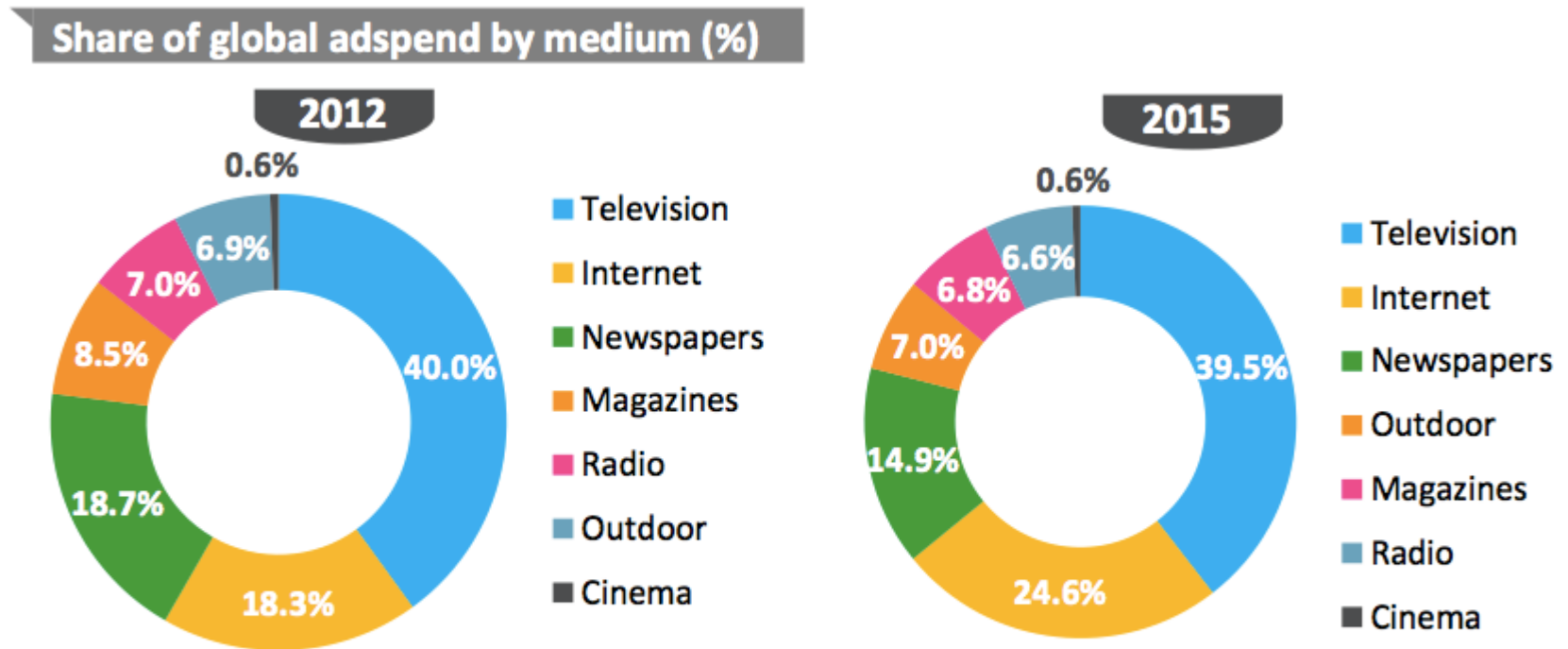
Who has this Data?

- Every Web site has server side navigation and query data
- Search engines, social network sites
 - Google, Microsoft, Yahoo!, Facebook
- Content Publishers
 - Newspapers, magazines, portals
- Retailers
 - Amazon, eBay, Target, Experian
- Data aggregators/advertising networks
 - Doubleclick, BlueKai
- Web analytics
 - Nielsen, ComScore

The Company Perspective: Learning about the User

- Companies would like to integrate all of this data
 - create a rich and dynamic picture of each user
 - Analogy: think of the local village store owner and his/her regular customers
- Challenges:
 - Accessing the data:
 - Retailer generally only sees their own Website data
 - Google, Facebook, etc
 - Legal restrictions
 - Privacy laws prohibit widespread sharing of data
 - Technical challenges
 - Is “J. Smith” registered at Target the same as “John Smith” who has a Facebook account, or jsmith@gmail.com?
 - Is John Smith, 24 Main Street, Maitland likely to be the same person as John Smith with an IP address in Maitland?

Predicted Increase in Internet Share of Advertising



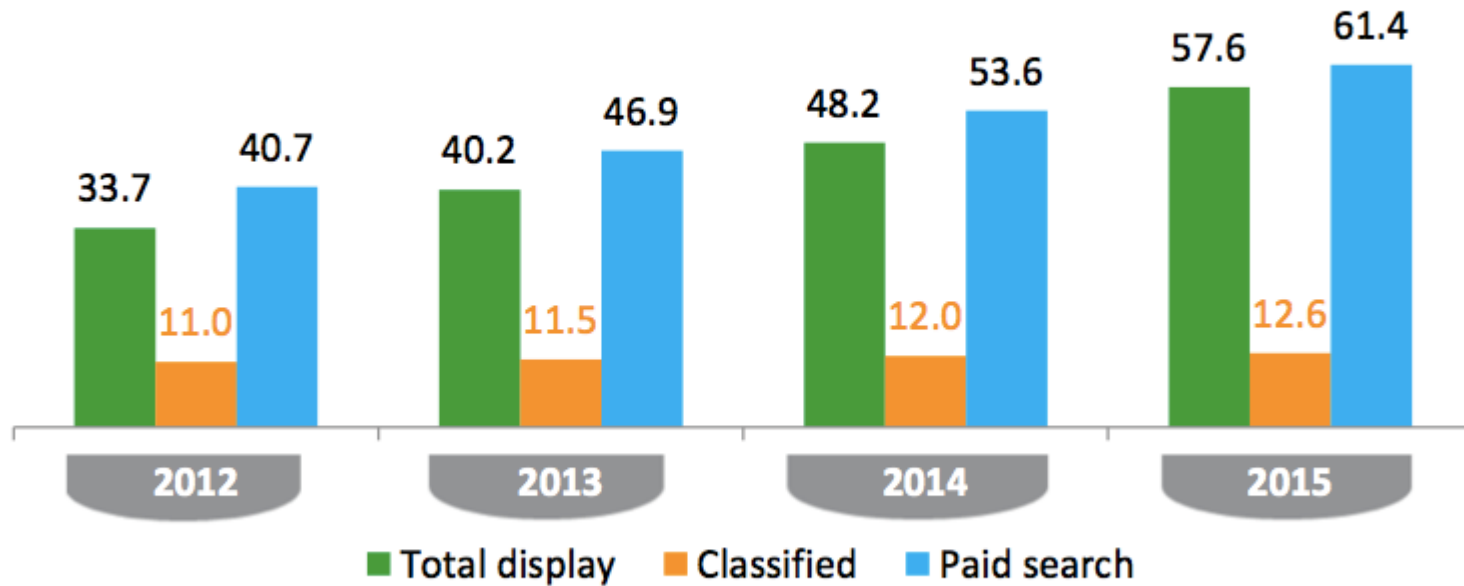
Source: ZenithOptimedia

Total Annual Expenditure on Advertising worldwide = \$500 billion

From Techcrunch.com, Sept 30, 2013

Trends in Spending on Internet Advertising

Internet adspend by type 2012-2015 (US\$bn)



Source: ZenithOptimedia

From Techcrunch.com, Sept 30, 2013

Link Analysis and the PageRank Algorithm

The Web Graph

- Graph $G = (V, E)$
 - V = set of all Web pages, let $n = |V|$
 - E = set of all hyperlinks
- Number of nodes ?
 - Difficult to estimate, > 10 billion?
 - Crawling the Web is highly non-trivial
- Number of edges?

Graph is sparse, i.e.,
mean number of outlinks per page is a small constant and not $O(n)$

The Web Graph

- The Web graph is inherently dynamic
 - nodes and edges are continually appearing and disappearing
- Research on general properties of the Web graph
 - What is the distribution of the number of in-links and out-links?
 - What is the distribution of number of pages per site?
 - Typically power-laws for many of these distributions
 - How far apart are 2 randomly selected pages on the Web?
 - What is the “average distance” between 2 random pages?
 - And so on...

An Aside: Social Networks

- Social networks = graphs
 - V = set of “actors” (e.g., students in a class)
 - E = set of interactions (e.g., collaborations)
 - Typically small graphs, e.g., $n = |V| = 10$ or 50
- Long history of social network analysis (e.g. at UCI)
- Quantitative data analysis techniques that can automatically extract “structure” or information from graphs
 - E.g., who is the most important “actor” in a network?
 - E.g., are there clusters in the network?
- Comprehensive reference:
 - S. Wasserman and K. Faust, Social Network Analysis, Cambridge University Press, 1994.

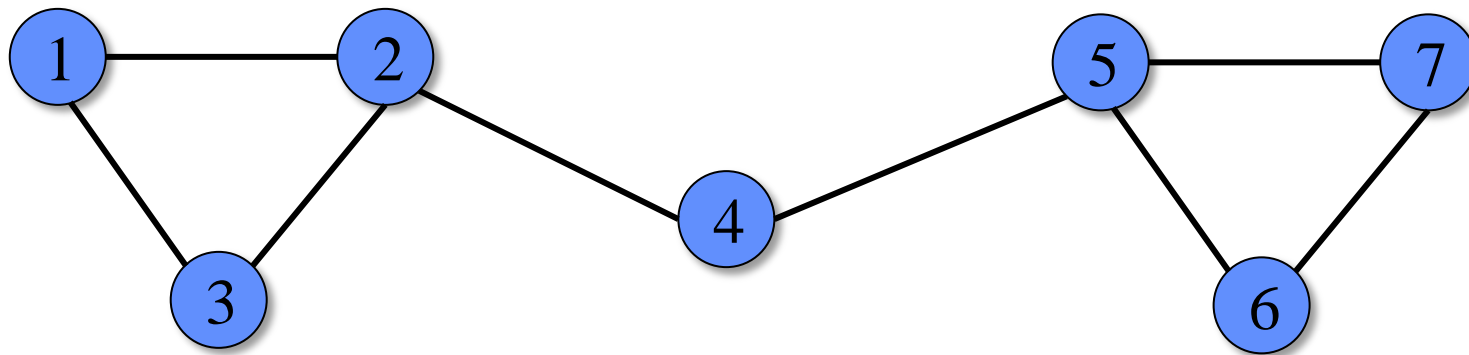
Node Importance in Networks

- General idea is that some nodes are more important than others in terms of the structure of the graph
 - “importance” is also referred to as “centrality” in the social network literature
- In a directed graph, “in-degree” may be a useful indicator of importance
 - e.g., for a citation network among authors (or papers)
 - in-degree is the number of citations => “importance”
 - However, “in-degree” is too simple in practice in that it implicitly assumes that all edges are of equal importance
 - In the next few slides we will develop a more powerful recursive approach

Other Notions of Node Importance or Centrality

Betweenness Centrality:

The fraction of shortest-paths (between all pairs of nodes in a network) that a node is part of



In this simple example, Node 4 would clearly have a much higher betweenness centrality score compared to any of the other nodes.

Time complexity is an issue: $O(n^3)$ time in general for all-pairs shortest paths

This can be reduced to $O(n |E|)$ in sparse graphs (Brandes' method).

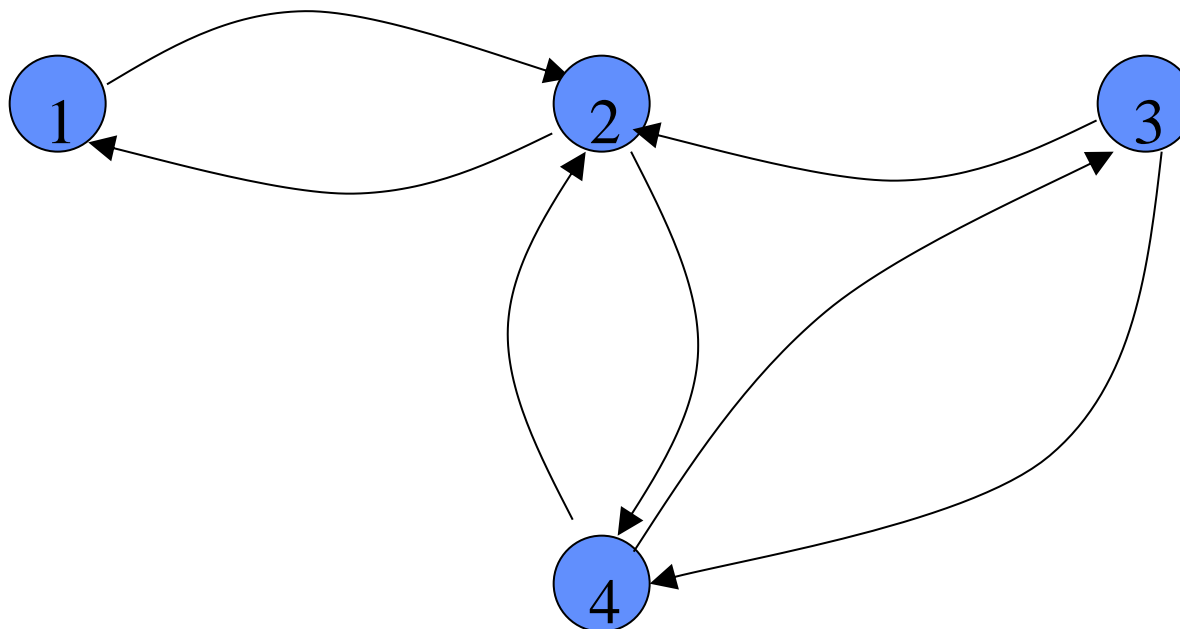
Recursive Notions of Node Importance in Directed Graphs

- w_{ij} = weight of link from node i to node j
 - assume $\sum_j w_{ij} = 1$ and weights are non-negative
 - e.g., default choice: $w_{ij} = 1/\text{outdegree}(i)$
 - more outlinks => less importance attached to each
- Define r_j = importance of node j in a directed graph (n = number of nodes)

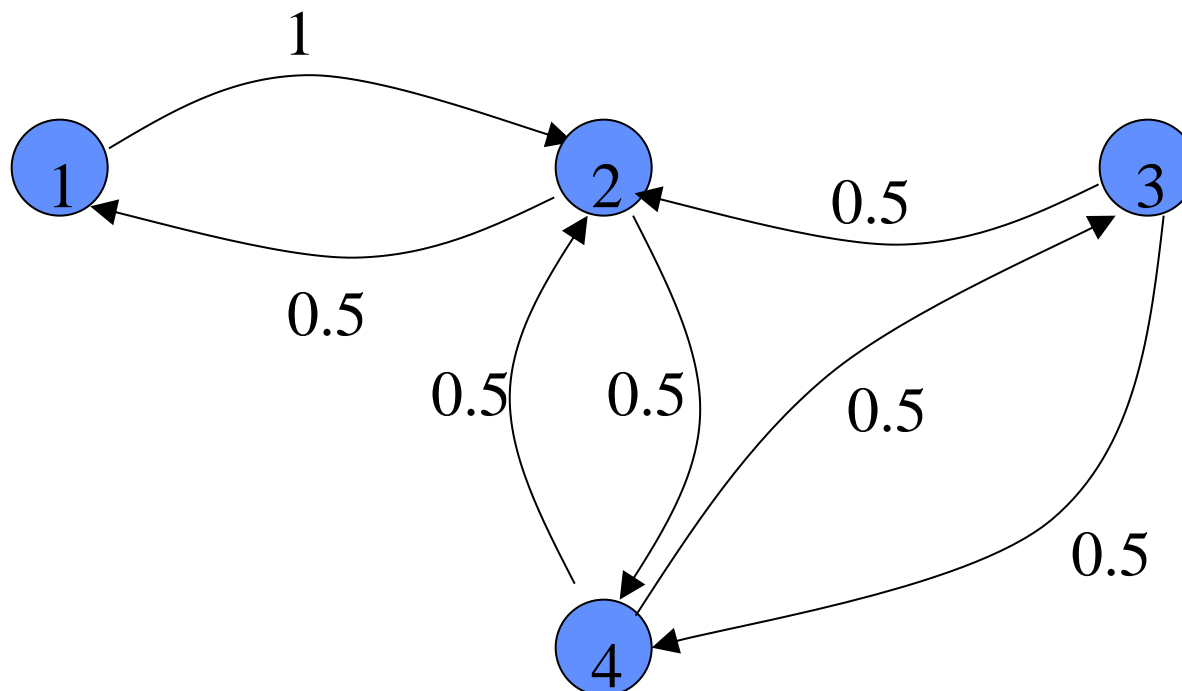
$$r_j = \sum_i w_{ij} r_i \quad i, j = 1, \dots, n$$

- Importance of a node is a weighted sum of the importance of nodes that point to it
 - Makes intuitive sense
 - Leads to a set of recursive linear equations

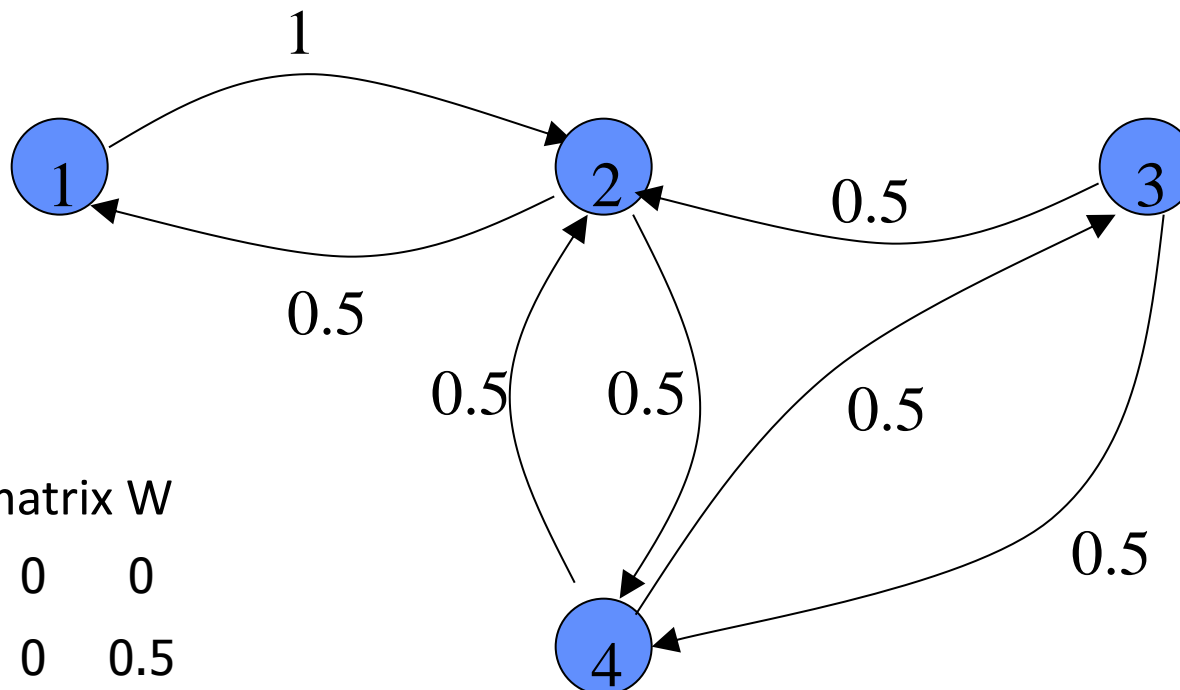
Simple Example



Simple Example



Simple Example



Weight matrix W

| | | | |
|-----|-----|-----|-----|
| 0 | 1 | 0 | 0 |
| 0.5 | 0 | 0 | 0.5 |
| 0 | 0.5 | 0 | 0.5 |
| 0 | 0.5 | 0.5 | 0 |

Each row in W represents the set of outgoing weights and sums to 1

Matrix-Vector form

- Recall r_j = importance of node j

$$r_j = \sum_i w_{ij} r_i \quad i, j = 1, \dots, n$$

$$\text{e.g., } r_2 = 1 r_1 + 0 r_2 + 0.5 r_3 + 0.5 r_4$$

= dot product of r vector with column 2 of W

Weight matrix W

| | | | |
|-----|-----|-----|-----|
| 0 | 1 | 0 | 0 |
| 0.5 | 0 | 0 | 0.5 |
| 0 | 0.5 | 0 | 0.5 |
| 0 | 0.5 | 0.5 | 0 |

Let \underline{r} = $n \times 1$ vector of importance values for the n nodes

Let W = $n \times n$ matrix of link weights

=> we can rewrite the importance equations as

$$\underline{r} = W^T \underline{r}$$

Eigenvector Formulation

Reformulate our problem in matrix-vector terms

Solve the importance equations for unknown \underline{r} , with known W

$$\underline{r} = W^T \underline{r}$$

We recognize this as a standard eigenvalue problem, i.e.,

$$A \underline{r} = \lambda \underline{r} \quad (\text{where } A = W^T)$$

with λ = an eigenvalue with value 1 (assuming there is such an eigenvalue)
and \underline{r} = the eigenvector corresponding to $\lambda = 1$

Eigenvector Formulation

Need to solve for \underline{r} in the following equation:

$$(W^T - \lambda I) \underline{r} = 0$$

Note: W is a stochastic matrix, i.e., rows are non-negative and sum to 1

Results from linear algebra tell us that:

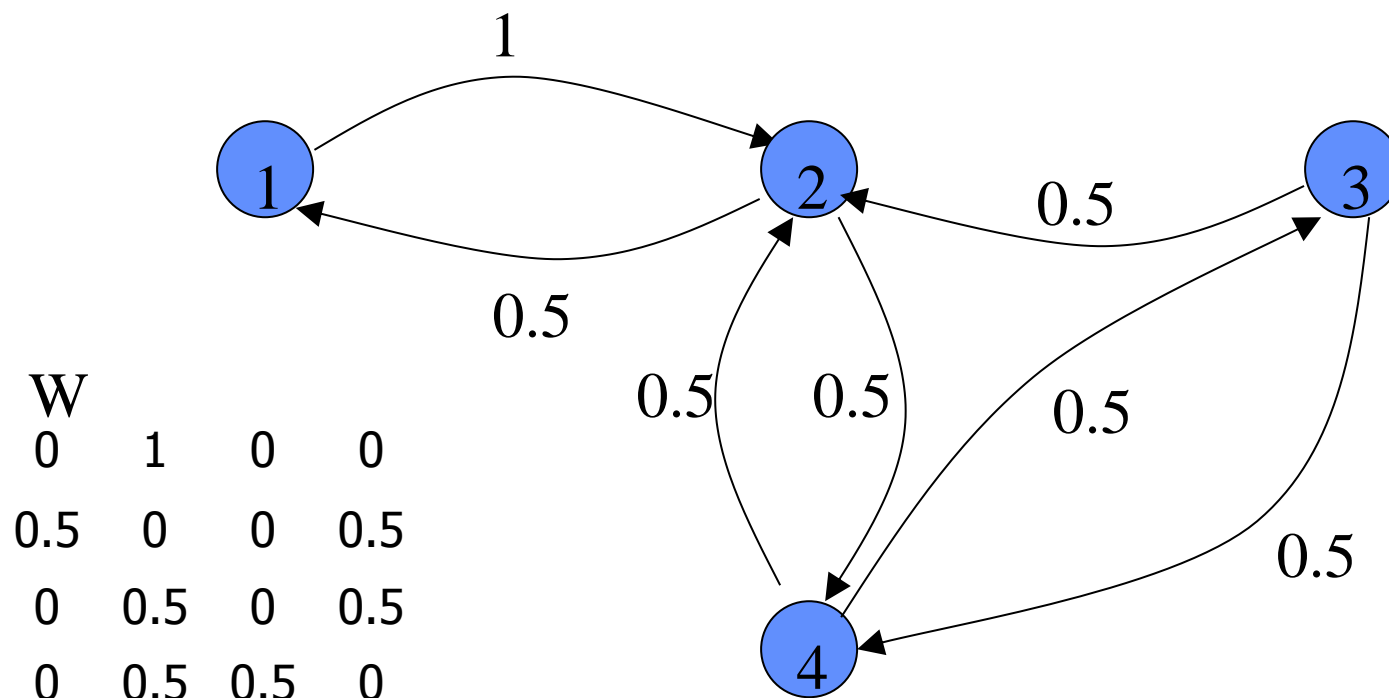
- (a) Since W is a stochastic matrix, W and W^T have the same eigenvectors/eigenvalues
- (b) The largest of these eigenvalues λ is always 1
- (c) the vector \underline{r} corresponds to the eigenvector corresponding to the largest eigenvalue of W (or W^T)

Solution for the Simple Example

Solving for the eigenvector of W we get

$$\underline{r} = [0.2 \quad 0.4 \quad 0.133 \quad 0.2667]$$

Results are quite intuitive, e.g., 2 is “most important”



PageRank Algorithm: Applying this idea to the Web

1. Crawl the Web to get nodes (pages) and links (hyperlinks)
[highly non-trivial problem!]
2. Weights from each page = $1/(\text{\# of outlinks})$
3. Solve for the eigenvector \underline{r} (for $\lambda = 1$) of the weight matrix

Computational Problem:

- Solving an eigenvector equation scales as $O(n^3)$
- For the entire Web graph $n > 10$ billion (!!)
- So direct solution is not feasible

Can use the power method (iterative)

$$\underline{r}^{(k+1)} = W^T \underline{r}^{(k)} \quad \text{for } k=1,2,\dots$$

Power Method for solving for \underline{r}

$$\underline{r}^{(k+1)} = W^T \underline{r}^{(k)}$$

Define a suitable starting vector $\underline{r}^{(1)}$

e.g., all entries $1/n$, or all entries = $\text{indegree}(\text{node})/|E|$, etc

Each iteration is matrix-vector multiplication $\Rightarrow O(n^2)$

- problematic?

no: since W is highly sparse each iteration is effectively $O(n)$
(Web pages have limited outdegree)

For sparse W , the iterations typically converge quite quickly:

- rate of convergence depends on the “spectral gap”

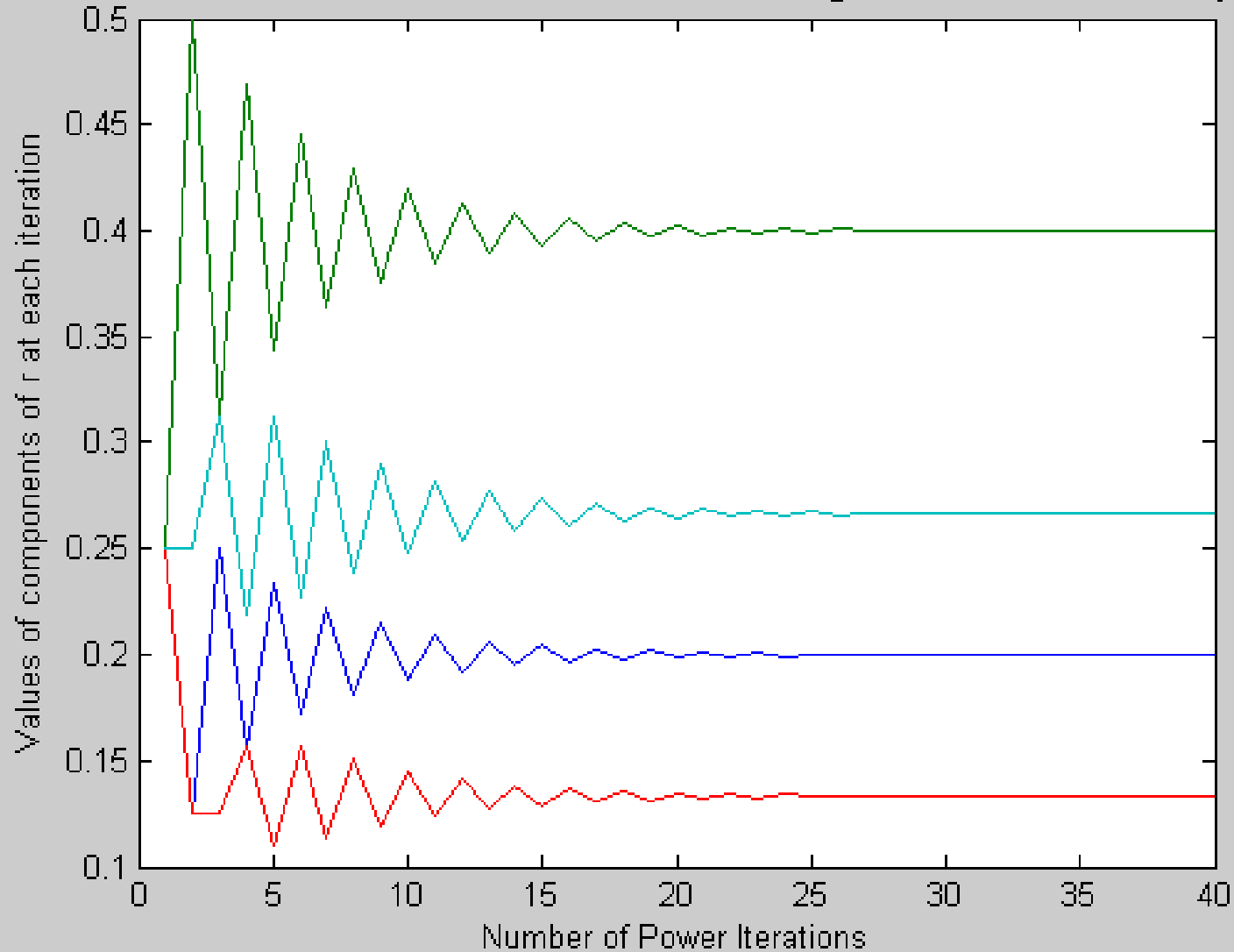
- > how quickly does $\text{error}(k) = (\lambda_2 / \lambda_1)^k$ go to 0 as a function of k ?

- > if $|\lambda_2|$ is close to 1 ($= \lambda_1$) then convergence is slow

- empirically: Web graph with 300 million pages

- > 50 iterations to convergence (Brin and Page, 1998)

Illustration of Power Iteration Convergence for W Example



Basic Principles of Markov Chains

Discrete-time finite-state first-order Markov chain, K states

Transition matrix $A = K \times K$ matrix

- Entry $a_{ij} = P(\text{state}_t = j \mid \text{state}_{t-1} = i)$, $i, j = 1, \dots, K$
- Rows sum to 1 (since $\sum_j P(\text{state}_t = j \mid \text{state}_{t-1} = i) = 1$)
- Note that $P(\text{state} \mid \dots)$ only depends on state_{t-1}

P_0 = initial state probability = $P(\text{state}_0 = i)$, $i = 1, \dots, K$

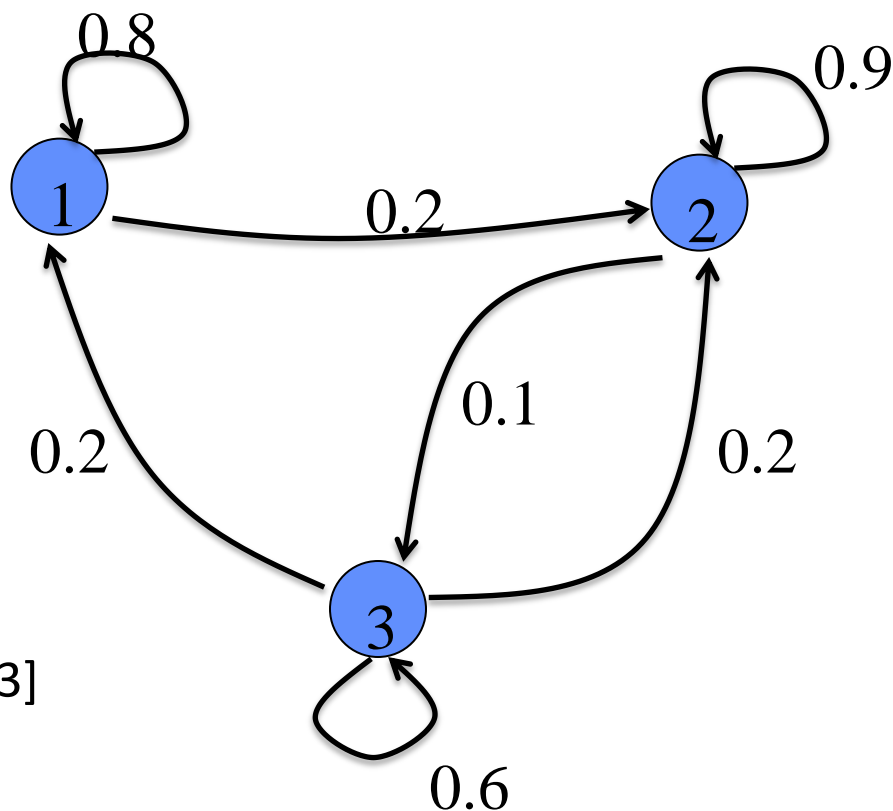
Simple Example of a Markov Chain

K = 3 states

Transition matrix A:

| | | |
|-----|-----|-----|
| 0.8 | 0.2 | 0.0 |
| 0.0 | 0.9 | 0.1 |
| 0.2 | 0.2 | 0.6 |

Initial state vector P_0 : $[1/3 \ 1/3 \ 1/3]$



Steady-State (Equilibrium) Distribution for a Markov Chain

Irreducibility:

- A Markov chain is irreducible if there is a directed path from any node to any other node

Steady-state distribution π for an irreducible Markov chain*:

π_i = probability that in the long run, chain is in state i

From Markov chain theory, the steady state π 's are the solutions to $\pi = A^t \pi$

Note that this is exactly the same as our earlier recursive equations for node importance in a graph!

*Note: technically, for a meaningful solution to exist for π , A must be both irreducible and aperiodic

Markov Chain Interpretation of PageRank

- W is a stochastic matrix (rows sum to 1) by definition
 - can interpret W as defining the transition probabilities in a Markov chain
 - w_{ij} = probability of transitioning from node i to node j

- Markov chain interpretation:

$$\underline{r} = W^T \underline{r}$$

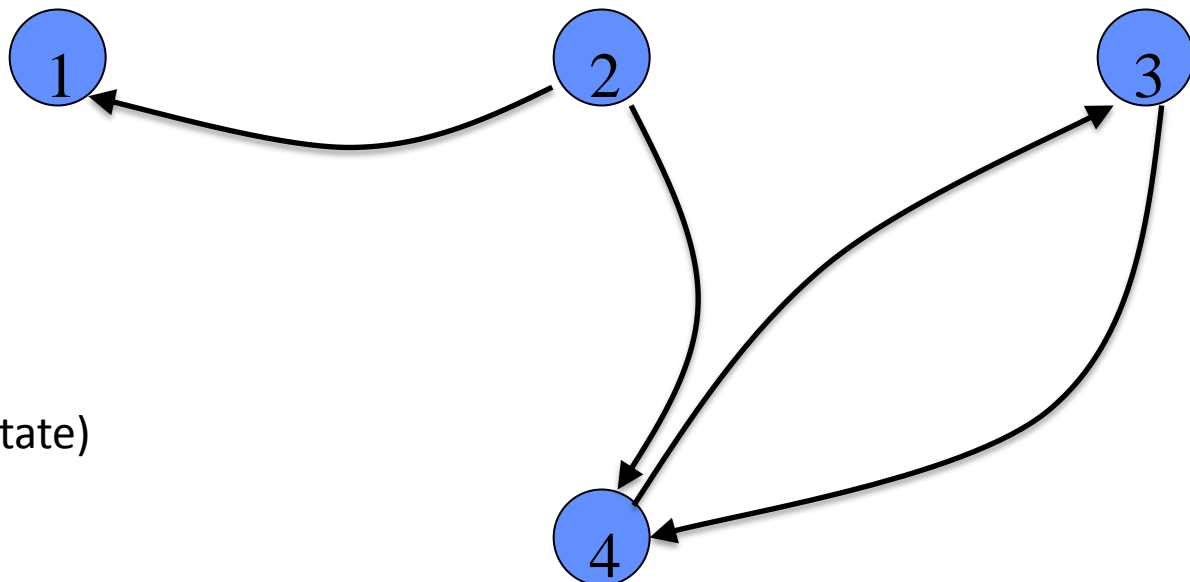
-> these are the solutions of the steady-state probabilities for a Markov chain

page importance \Leftrightarrow steady-state Markov probabilities \Leftrightarrow eigenvector

The Random Surfer Interpretation

- Recall that for the Web model, we set $w_{ij} = 1/\text{outdegree}(i)$
- Thus, in using W for computing importance of Web pages, this is equivalent to a model where:
 - We have a random surfer who surfs the Web for an infinitely long time
 - At each page the surfer randomly selects an outlink to the next page
 - “importance” of a page = fraction of visits the surfer makes to that page
 - this is intuitive: pages that have better connectivity will be visited more often

Potential Problem with “Sinks” in the Web Graph



Page 1 is a “sink” (no outlink)
(also known as an absorbing state)

Pages 3 and 4 are also “sinks” (no outlink from the system)

Markov chain theory tells us that no steady-state solution exists
- depending on where you start you will end up at 1 or {3, 4}

Markov chain is “reducible”

Making the Web Graph Irreducible

- One simple solution to our problem is to modify the Markov chain:
 - With probability α the random surfer jumps to any random page in the system (with probability of $1/n$, conditioned on such a jump)
 - With probability $1-\alpha$ the random surfer selects an outlink (randomly from the set of available outlinks)
- The resulting transition graph is fully connected \Rightarrow Markov system is irreducible \Rightarrow steady-state solutions exist
- Typically α is chosen to be between 0.1 and 0.2 in practice
- But now the graph is dense - so we lose the nice sparsity for computation!
However, power iterations can be written as:

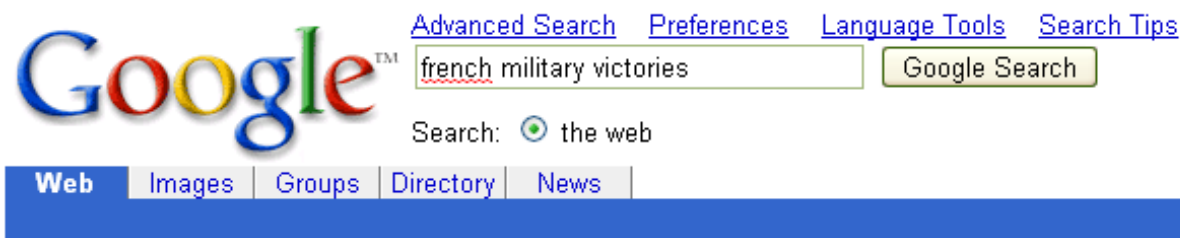
$$\underline{r}^{(k+1)} = (1-\alpha) W^T \underline{r}^{(k)} + (\alpha/n) \underline{1}^T$$
 - Complexity is still $O(n)$ per iteration for sparse W

The PageRank Algorithm

- S. Brin and L. Page, The anatomy of a large-scale hypertextual search engine, in Proceedings of the 7th WWW Conference, 1998.
- PageRank = the method on the previous slide, applied to the entire Web graph
 - Crawl the Web
 - Store both connectivity and content
 - Calculate (off-line) the “pagerank” r for each Web page using the power iteration method
- How can this be used to answer Web queries:
 - Terms in the search query are used to limit the set of pages of possible interest
 - Pages are then ordered for the user via precomputed pageranks
 - The Google search engine combines r with text-based measures
 - This was the first demonstration that link information could be used for content-based search on the Web

Link Manipulation

Query = french military victories



Did you mean: [french military defeats](#)

No standard web pages containing all your search terms were found.

Your search - **french military victories** - did not match any documents.

Suggestions:

- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Try fewer keywords.

Also, you can try [Google Answers](#) for expert help with your search.

[Google Home](#) - [Advertise with Us](#) - [Search Solutions](#) - [Services & Tools](#) - [Jobs, Press, & Help](#)

Conclusions

- PageRank algorithm was the first algorithm for link-based search
 - Many extensions and improvements since then
 - Same idea used in social networks for determining importance
- Real-world search involves many other aspects besides PageRank
 - E.g., use of logistic regression for ranking
 - Learns how to predict relevance of page (represented by bag of words) relative to a query, using historical click data
 - See paper by Joachims on class Web page

Web Analytics

Web Logs

| Field | Description |
|-------------------|--|
| Date | The date that the activity occurred |
| Time | The time that the activity occurred |
| Client IP address | The IP address of the client that accessed your server |
| User Name | The name of the authenticated user who access your server |
| Server Port | The port number the client is connected to |
| Method | The action the client was trying to perform |
| URI Stem | The resource accessed |
| URI Query | The query, if any, the client was trying to perform |
| Protocol Status | The status of the action, in HTTP or FTP terms |
| Bytes Sent | The number of bytes sent by the server |
| Bytes Received | The number of bytes received by the server |
| Time Taken | The duration of time, in milliseconds, that the action consumed |
| Protocol Version | The protocol (HTTP, FTP) version used by the client |
| Host | Display the content of the host header |
| User Agent | The browser used on the client |
| Cookie | The content of the cookie sent or received, if any |
| Referrer | The previous site visited by the user (including search query if any). |

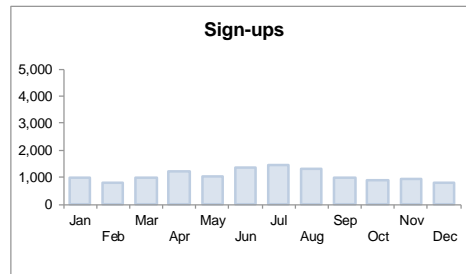
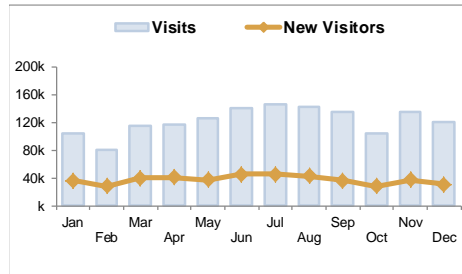
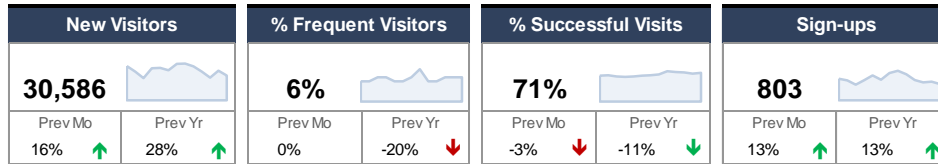
Web Logs

| Field | Description |
|-------------------|--|
| Date | The date that the activity occurred |
| Time | The time that the activity occurred |
| Client IP address | The IP address of the client that accessed your server |
| User Name | The name of the authenticated user who access your server |
| Server Port | The port number the client is connected to |
| Method | The action the client was trying to perform |
| URI Stem | The resource accessed |
| URI Query | The query, if any, the client was trying to perform |
| Protocol Status | The status of the action, in HTTP or FTP terms |
| Bytes Sent | The number of bytes sent by the server |
| Bytes Received | The number of bytes received by the server |
| Time Taken | The duration of time, in milliseconds, that the action consumed |
| Protocol Version | The protocol (HTTP, FTP) version used by the client |
| Host | Display the content of the host header |
| User Agent | The browser used on the client |
| Cookie | The content of the cookie sent or received, if any |
| Referrer | The previous site visited by the user (including search query if any). |

Web Log Analytics

- Use Web logs to generate statistics on....
 - Which pages are visited
 - Where users came from (referrer page)
 - Geographic distribution of visitors (from IP addresses)
 - How many pages were clicked
 - Actions taken by visitors (queries, purchases, etc)
 - How long people stayed on pages/site (difficult to do accurately)
- Software tools for reports and interactive dashboards
 - E.g., Google Analytics (free)
- Useful for understanding user behavior on a particular site
- Reference:
 - Web Analytics 2.0, Avinash Kaushik, Wiley, 2010

December 2010



| Acquisition Source | Visits | New Visitors | % New Visitors | % Frequent Visitors | % Successful Visits | Sign-ups | Sign-up Rate 1st Visit |
|--------------------|---------|--------------|----------------|---------------------|---------------------|----------|------------------------|
| All | 119,285 | 30,586 | 26% | 6% | 71% | 803 | 30% |
| Direct Traffic | 42,765 | 10,004 | 23% | 18% | 74% | 200 | 31% |
| Organic Search | 36,162 | 9,785 | 27% | 19% | 65% | 324 | 28% |
| Pay Per Click | 16,887 | 4,870 | 29% | 8% | 72% | 121 | 31% |
| Display Ads | 2,652 | 1,080 | 41% | 4% | 72% | 45 | 31% |
| YouTube | 7,652 | 1,599 | 21% | 4% | 67% | 37 | 29% |
| Social Sites | 1,625 | 468 | 29% | 3% | 71% | 31 | 30% |

| Pay-Per-Click | |
|-------------------|------------------|
| \$3,000 | Budget |
| \$24.79 | Cost per Sign-up |
| Email Newsletters | |
| \$8,957 | Emails Delivered |
| 11.0% | Bounce Rate |
| 0.02% | Unsubscribe Rate |
| Display Ads | |
| \$1,000 | Budget |
| \$22.22 | Cost per Sign-up |

| 1: Form Start | |
|-----------------------|---------------------|
| Completions | Drop-off, Prev Step |
| 1,287 | -- |
| 2: Form Completion | |
| Completions | Drop-off, Prev Step |
| 927 | 28% |
| 3: Validated Sign-ups | |
| Completions | Drop-off, Prev Step |
| 803 | 13% |

Key Findings & Recommendations

This dashboard is for Atlantis Aquariums, a fictional manufacturer of low maintenance aquariums sold through a national chain of pet stores. Atlantis doesn't sell direct, but has a consumer-focused website for the purpose of driving offline sales of filters (their most profitable product).

The educational-focused website features many high quality educational videos. There are calls to action throughout the site for visitors' to sign up for the opt-in email list which is used to remind aquarium owners when to replace their filters.

Atlantis has a small online marketing budget and relies heavily on organic search to drive traffic to the website, but they they also have a few small PPC campaigns and some banner ads. Atlantis has recently added their videos to YouTube and is beginning to track referrals from social sites such as Facebook, MySpace and Twitter.

| Rank | Top 10 Converting Videos | CR % |
|------|----------------------------------|------|
| 1 | Filter Facts for A-fish-ionados | 7.3% |
| 2 | Keeping Your Fish Healthy | 5.1% |
| 3 | Finding the Right Filter | 4.9% |
| 4 | Introducing Aquatic Plants | 4.5% |
| 5 | Feeding Your Fish | 4.3% |
| 6 | Snails: Quicker Picker Uppers | 4.1% |
| 7 | Basics of Saltwater Fishtanks | 4.0% |
| 8 | Dos and Don'ts of Aquarium Décor | 3.9% |
| 9 | Four Horsemen of the Apaqualypse | 3.9% |
| 10 | Avoiding Aquarium Conflict | 3.5% |

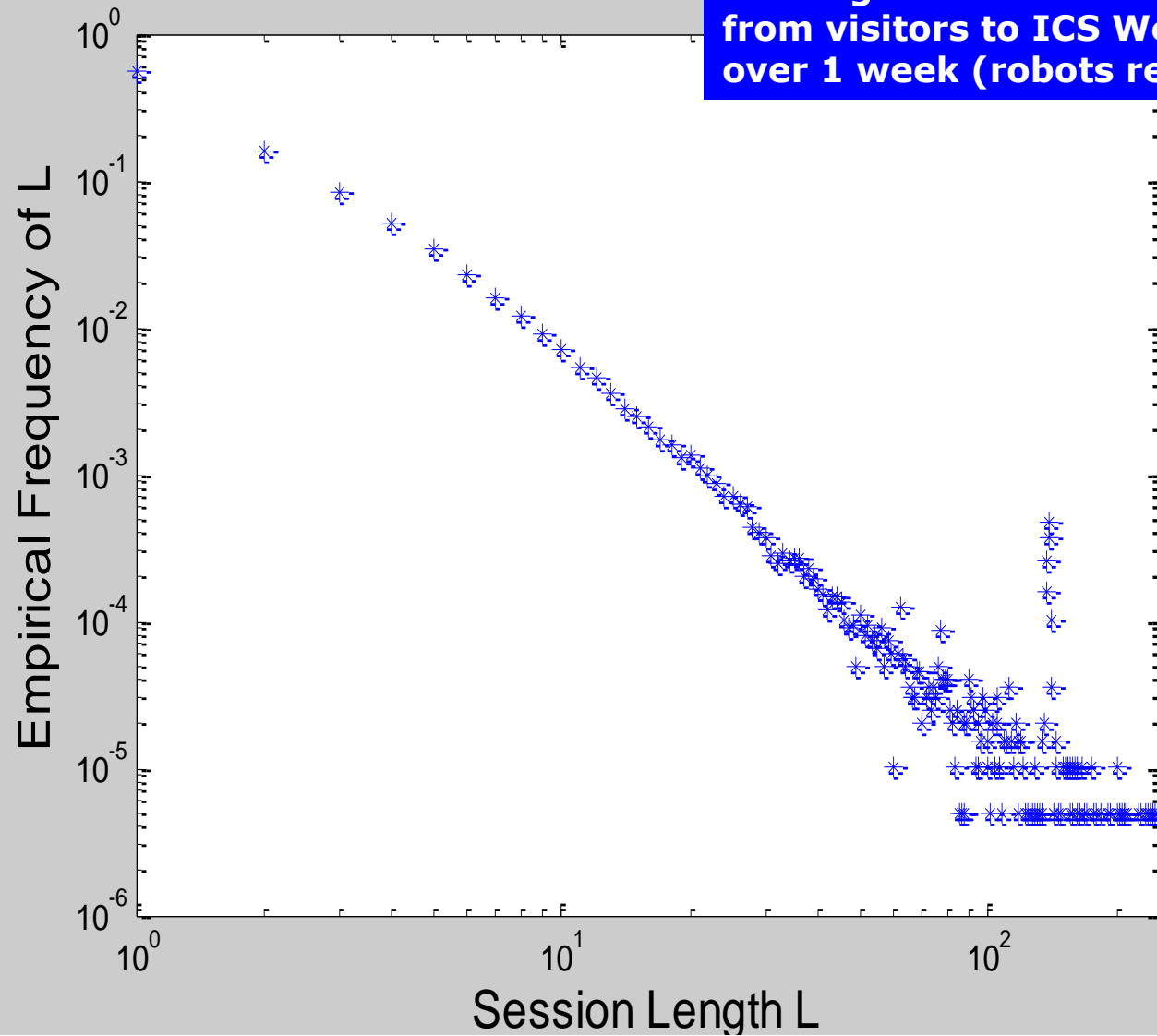
© 2009 Stratigent, LLC. All rights reserved. To learn the techniques that were used to create this dashboard please visit our website: www.stratigent.com



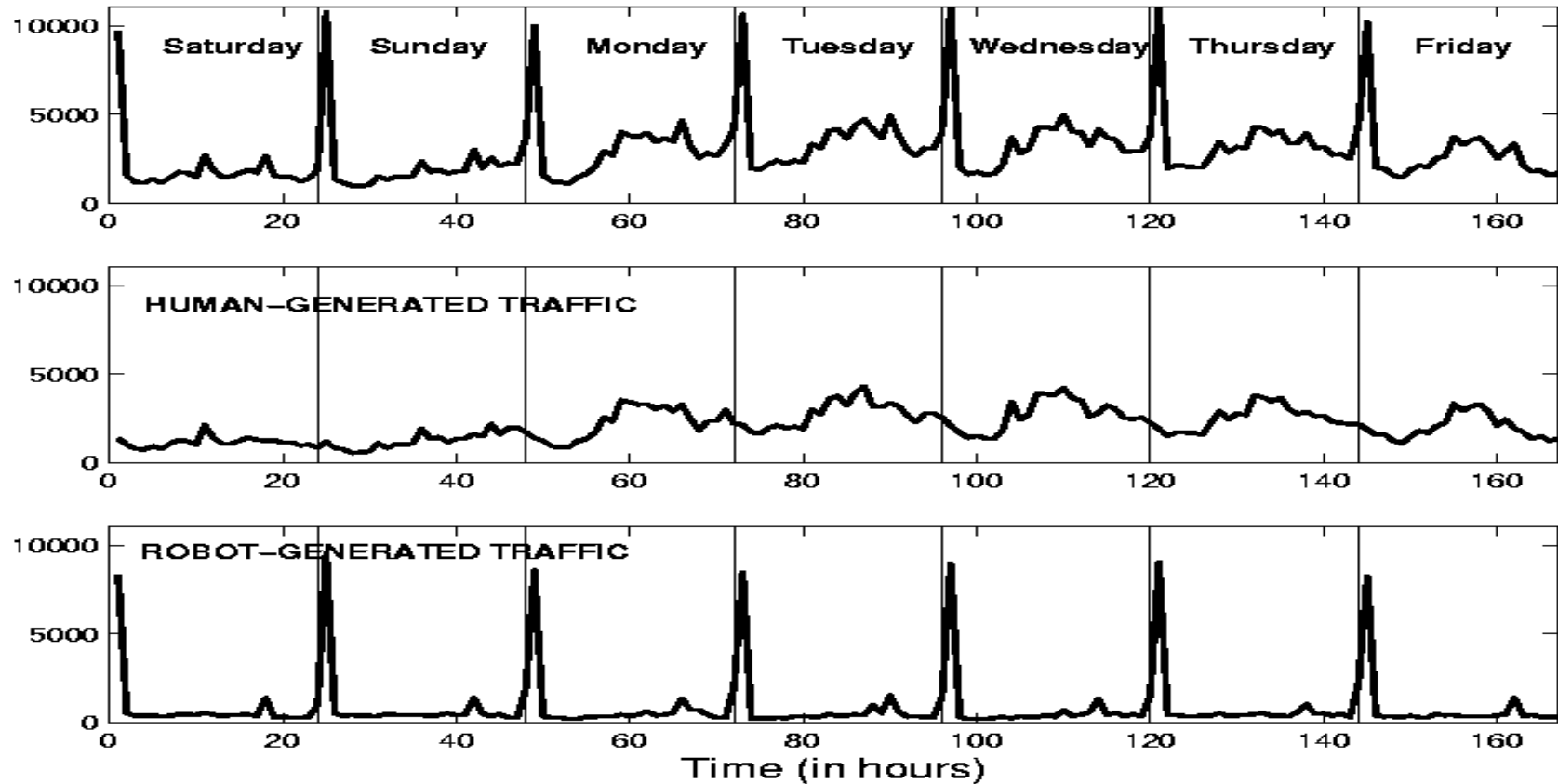
Descriptive Summary Statistics

- Histograms, scatter plots, time-series plots
 - Very important!
 - Helps to understand the big picture
 - Provides “marginal” context for any model-building
 - models aggregate behavior, not individuals
 - Challenging for Web log data
- Examples
 - Session lengths (e.g., power laws)
 - Click rates as a function of time, content

L = number of page requests in a single session from visitors to ICS Web site over 1 week (robots removed)



A time-series plot of ICS Website data



Number of page requests per hour as a function of time from page requests in the www.ics.uci.edu Web server logs

Identifying individual users from Web server logs

- Useful to associate specific page requests to specific individual users
- IP address most frequently used
- Disadvantages
 - One IP address can belong to several users
 - Dynamic allocation of IP address
- Better to use cookies (or login ID if available)
 - Information in the cookie can be accessed by the Web server to identify an individual user over time
 - Actions by the same user during different sessions can be linked together
- Another option is to enforce user registration
 - High reliability
 - But can discourage potential visitors
 - Large portals (such as Yahoo!) have high fraction of logged-in users

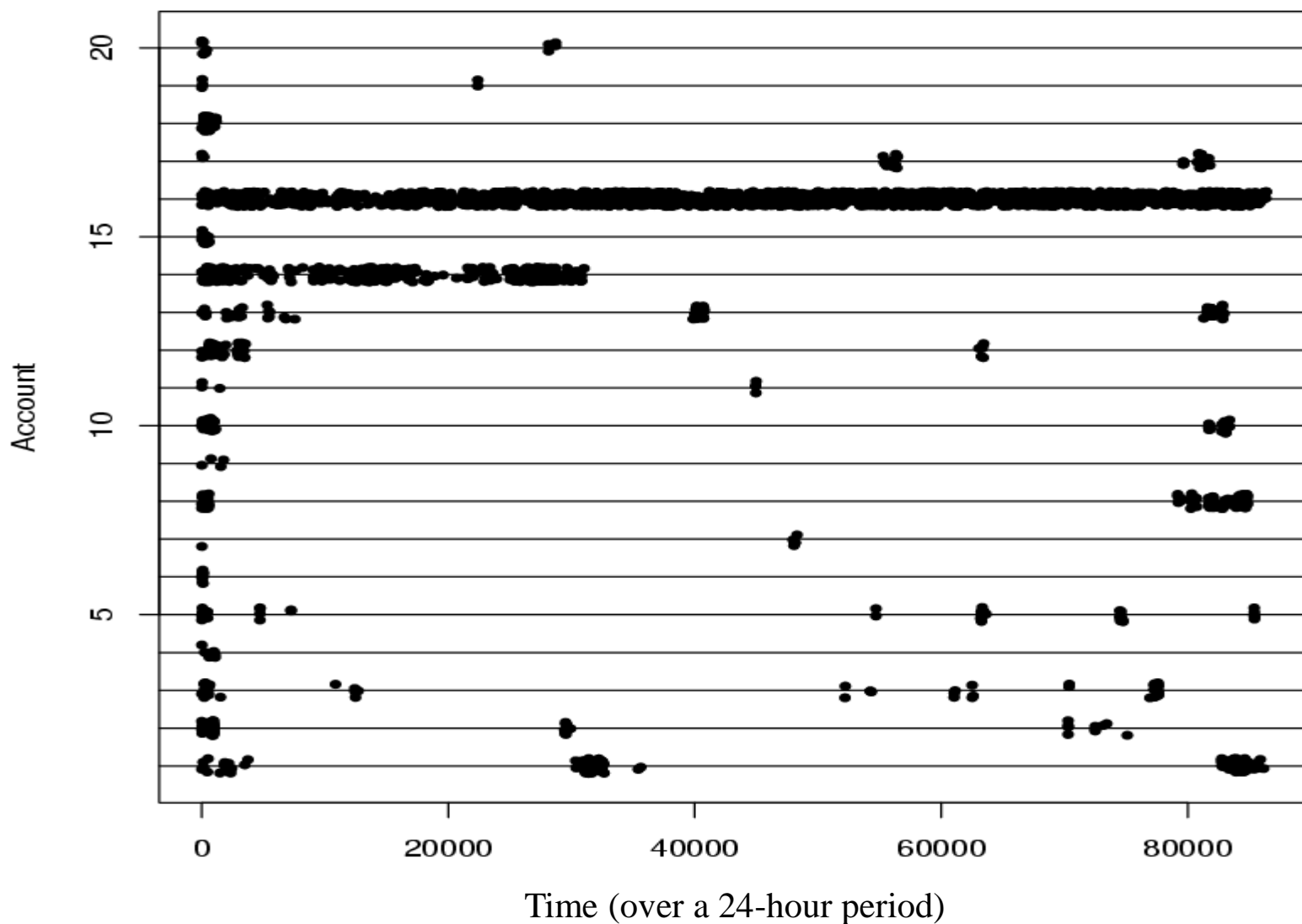
Sessionizing

- Time oriented (robust)
 - e.g., by gaps between requests
 - not more than 20 minutes between successive requests
 - this is a heuristic – but is a standard “rule” used in practice
- Navigation oriented (good for short sessions and when timestamps unreliable)
 - Referrer is previous page in session, or
 - Referrer is undefined but request within 10 secs, or
 - Link from previous to current page in web site

Client-side data

- Advantages of collecting data at the client side:
 - Direct recording of page requests (eliminates 'masking' due to caching)
 - Recording of all browser-related actions by a user (including visits to multiple websites)
 - More-reliable identification of individual users (e.g. by login ID for multiple users on a single computer)
- Preferred mode of data collection for studies of navigation behavior on the Web
- Companies like ComScore and Nielsen use client-side software to track home computer users

Example of client-side data from Alexa, each “dot” is a URL request in a browser



Summary of Issues in Web Log Analytics

- Traffic may be dominated by “robots”
 - E.g., search engine Web crawlers
- Users can be identified via
 - IP address (noisy)
 - Cookie (noisy)
 - Login (ideal) – can be linked to other data
- Ramifications
 - Data is very noisy
 - Data on each user is often quite error-prone
- Nonetheless,
 - Say 30% of user models are good
 - So 30% of users see relevant ads (and the other 70% the usual default ads)
 - Can be millions of dollars better than default ads for all 100%

**CASE STUDY:
CLUSTERS OF MARKOV CHAINS FOR MODELING USER
NAVIGATION ON A WEBSITE**

Markov Models for Modeling User Navigation

- General approach is to use a finite-state Markov chain
 - Each state can be a specific Web page or a category of Web pages
 - If only interested in the order of visits (and not in time), each new request can be modeled as a transition of states
- Issues
 - Self-transition
 - Time-independence

Modeling Web Page Requests with Markov chain mixtures

- MSNBC Web logs (circa 2000)
 - Order of 2 million individual users per day
 - different session lengths per individual
 - difficult visualization and clustering problem
- WebCanvas
 - uses mixtures of Markov chains to cluster individuals based on their observed sequences
 - software tool: EM mixture modeling + visualization

Next few slides are based on material in:

I. Cadez et al, Model-based clustering and visualization of navigation patterns on a Web site, *Journal of Data Mining and Knowledge Discovery*, 2003.

MSNBC Cover - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Guide Print Security Stop Netscape

Bookmarks Location: <http://www.msnbc.com/news/default.asp?cp1=1> What's Related

Instant Message WebMail People Yellow Pages Download New & Cool Channels

Updated: 12:02 ET Jun. 23, 2001

Today show
 Nightly News
 Dateline NBC
 MSNBC Cable
 News
 Business
 Sports
 Local
 Health
 Technology
 Living • Travel
 TV News
 Opinions
 Weather
 Shop@MSNBC
 MSN.com
 Headlines

MSNBC
 #1 NEWS SITE

POLITICS
Bush backs genetic equality
 • Law would bar discrimination

Facing justice
 • Serb government clears path to extradite Milosevic

Journey to heal
 • Pope lands in Ukraine on mission to ease tensions

Rocker moved
 • Braves trade away reliever who spurred controversy

Office DEPOT
 Advertisement

Pentagon seeks extra \$18 billion
 • Rumsfeld hopes to hike his budget

China uneasy about U.S. relations
 • WashPost: Competition with Beijing

Artificial pancreas show promise
 • Option could help diabetes patients

Tech is 'weakest link' in economy
 • Sector will face a sluggish rebound

Contraception finally comes of age
 • Newsweek: The Pill's rocky history

ENVISION THE SAVINGS
 Click Here
 uBid.com

Get MSNBC HEADLINES
 — on YOUR web site

SEARCH MSNBC **GO**

MSNBC QUICK LINKS

| | |
|----------------------|-----------------------|
| • Free News Alert | • International news |
| • Smart Tags | • Comics |
| • Voice your views | • Gossip |
| • Letters to editor | • Travel news |
| • Stock quote | • Week in Pictures |
| • Health Horizons | • Use our top stories |
| • Readers' favorites | • Crossword |
| • Local biz news | • Horoscope |

CASINO ON NET.COM **Go!**

Newsweek
Steven Levy
 • 'Japan's Bill Gates' wants broadband for all

Venus is the one
 A. Grant / AP
 • Evert: Healthy Williams should win Wimbledon
 • Collins: Pick Sampras

INSIDE MSNBC.COM

- Conflicting ratings for Ford F-150
- 'Boomburbs' mark an era of sprawl
- Processed meat tied to colon cancer
- Opinions: Courts can't fight terror
- Newsweek: Global gay persecution
- New products coming at PC Expo

Enter your ZIP code to get local news, sports, and weather

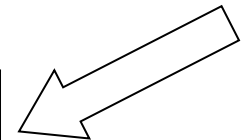
ZIP Enter favorite **GO**

Document: Done

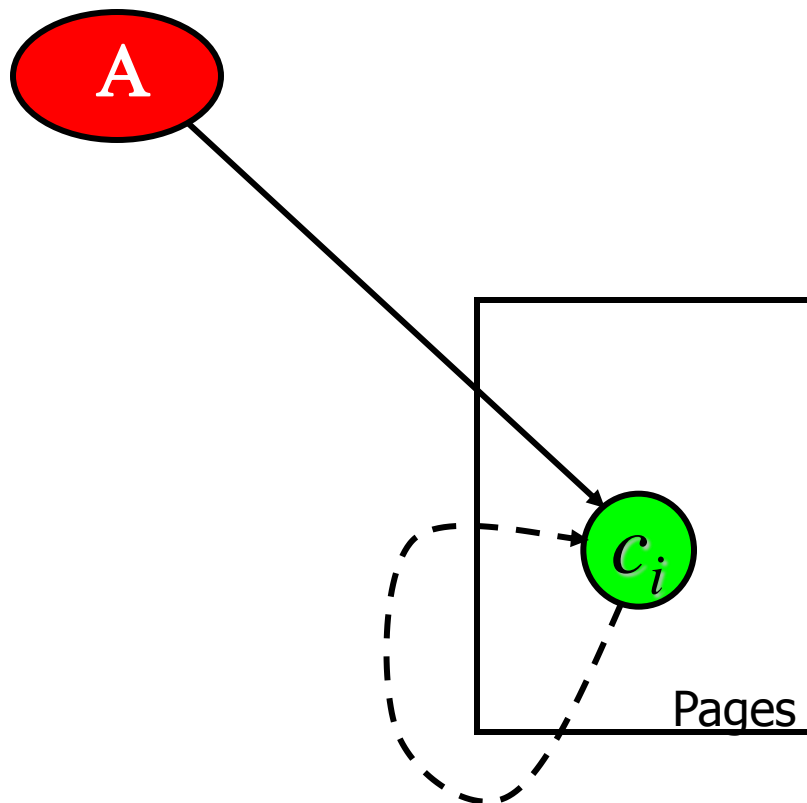
From Web logs to sequences

128.195.36.195, -, 3/22/00, 10:35:11, W3SVC, SRVR1, 128.200.39.181, 781, 363, 875, 200, 0, GET, /top.html, -,
 128.195.36.195, -, 3/22/00, 10:35:16, W3SVC, SRVR1, 128.200.39.181, 5288, 524, 414, 200, 0, POST, /spt/main.html, -,
 128.195.36.195, -, 3/22/00, 10:35:17, W3SVC, SRVR1, 128.200.39.181, 30, 280, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.195.36.101, -, 3/22/00, 16:18:50, W3SVC, SRVR1, 128.200.39.181, 60, 425, 72, 304, 0, GET, /top.html, -,
 128.195.36.101, -, 3/22/00, 16:18:58, W3SVC, SRVR1, 128.200.39.181, 8322, 527, 414, 200, 0, POST, /spt/main.html, -,
 128.195.36.101, -, 3/22/00, 16:18:59, W3SVC, SRVR1, 128.200.39.181, 0, 280, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.200.39.17, -, 3/22/00, 20:54:37, W3SVC, SRVR1, 128.200.39.181, 140, 199, 875, 200, 0, GET, /top.html, -,
 128.200.39.17, -, 3/22/00, 20:54:55, W3SVC, SRVR1, 128.200.39.181, 17766, 365, 414, 200, 0, POST, /spt/main.html, -,
 128.200.39.17, -, 3/22/00, 20:54:55, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.200.39.17, -, 3/22/00, 20:55:07, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.200.39.17, -, 3/22/00, 20:55:36, W3SVC, SRVR1, 128.200.39.181, 1061, 382, 414, 200, 0, POST, /spt/main.html, -,
 128.200.39.17, -, 3/22/00, 20:55:36, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.200.39.17, -, 3/22/00, 20:55:39, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.200.39.17, -, 3/22/00, 20:56:03, W3SVC, SRVR1, 128.200.39.181, 1081, 382, 414, 200, 0, POST, /spt/main.html, -,
 128.200.39.17, -, 3/22/00, 20:56:04, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.200.39.17, -, 3/22/00, 20:56:33, W3SVC, SRVR1, 128.200.39.181, 0, 262, 72, 304, 0, GET, /top.html, -,
 128.200.39.17, -, 3/22/00, 20:56:52, W3SVC, SRVR1, 128.200.39.181, 19598, 382, 414, 200, 0, POST, /spt/main.html, -,

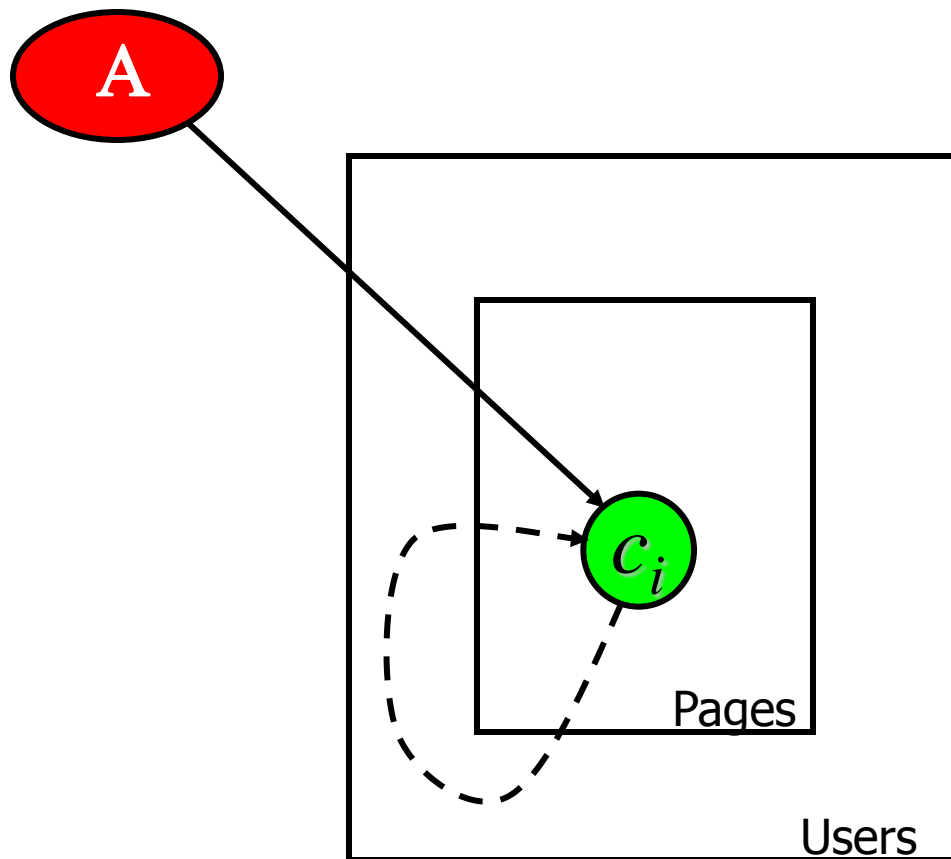
| | | | | | | | | | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| User 1 | 2 | 3 | 2 | 2 | 3 | 3 | 3 | 1 | 1 | 1 | 3 | 1 | 3 | 3 | 3 | 3 |
| User 2 | 3 | 3 | 3 | 1 | 1 | 1 | | | | | | | | | | |
| User 3 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | | | | | | | | |
| User 4 | 1 | 5 | 1 | 1 | 1 | 5 | 1 | 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| User 5 | 5 | 1 | 1 | 5 | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | |



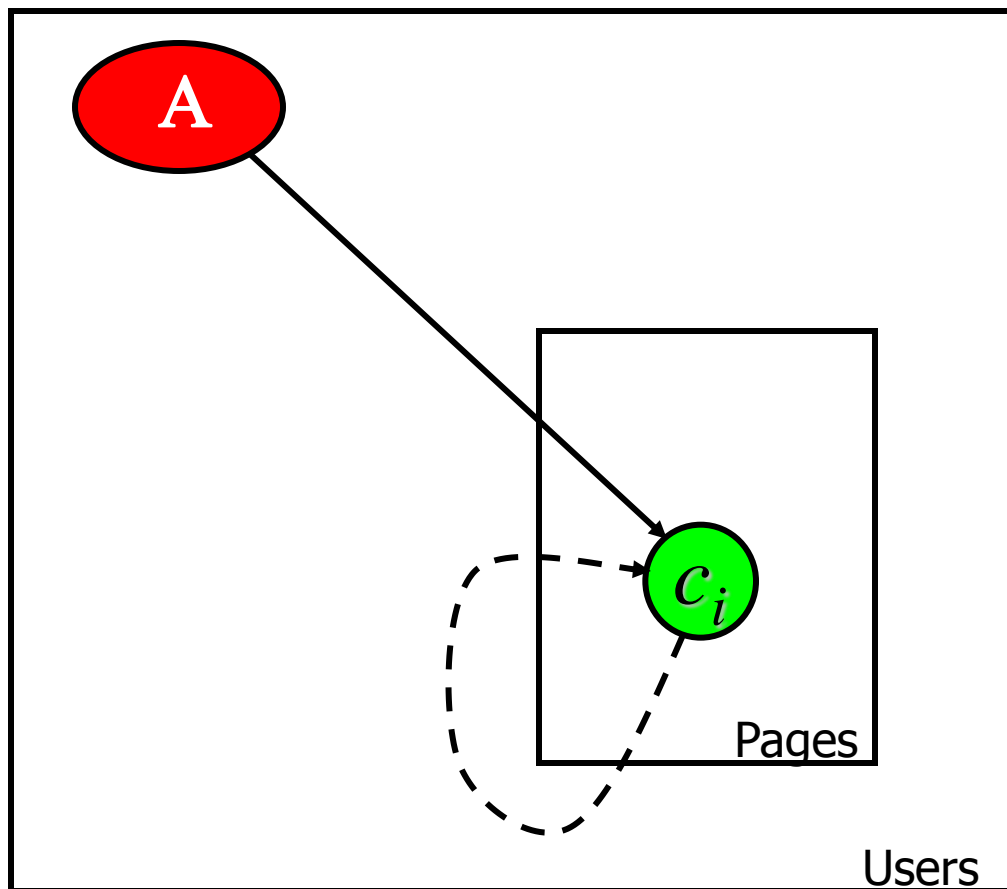
Graphical Model for Markov Chains



Multiple Users...One Common Markov Chain

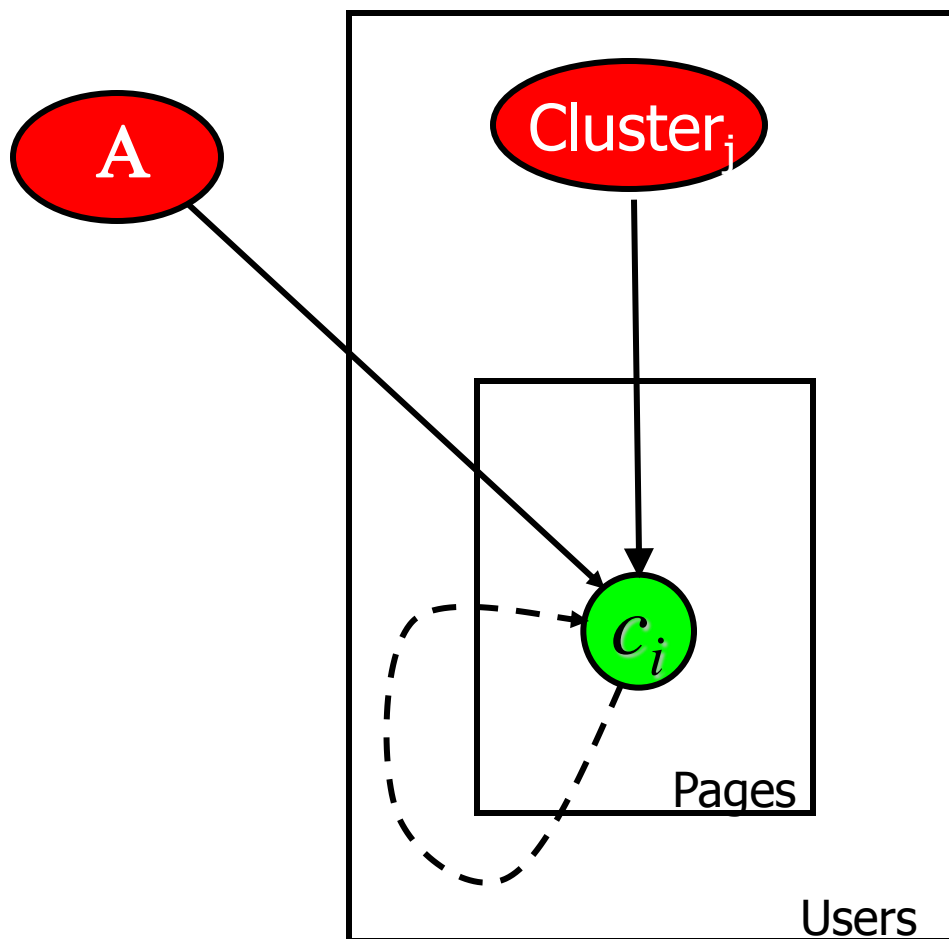


Multiple Users...One Chain per User



One Chain per Cluster of Users

Cadez, Meek, Heckerman, Smyth, 2003



Likelihood of a Sequence in a Markov Chain Model

$$P(\text{sequence}) = P_0(\text{first state}) \times \prod_{ij} (a_{ij})^{n_{ij}}$$

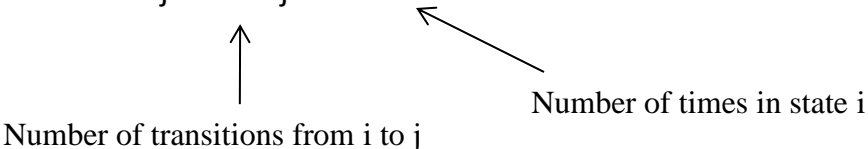
Number of transitions from i to j

Transition probability from i to j

Maximum likelihood estimation:

- given the n_{ij} 's, find the a_{ij} 's that maximize $P(\text{sequence})$

Probability Estimation in Markov Chains

$$p(\text{state } j \mid \text{state } i) = a_{ij} = n_{ij} / n_i$$


Number of transitions from i to j

Number of times in state i

With smoothing (or priors)

$$a_{ij} = (n_{ij} + \alpha_{ij}) / (n_i + \sum_m \alpha_{im})$$

where $\alpha_{ij} / \sum_m \alpha_{im}$ is the prior probability of transitioning from i to j

and where $\sum_m \alpha_{im}$ is the strength (equivalent sample size) of the prior

Could set $\alpha_{ij} = \alpha_j$ (same priors for transitioning out of each state) or
 $\alpha_{ij} = \alpha$ (same priors for transitioning in and out of each state)

Probability Estimation with Multiple Sequences

T sequences

Assume sequences are independent

n_{ijt} = number of times going from i to j in sequence t, $t = 1, \dots, T$

$$p(\text{state } j \mid \text{state } i) = a_{ij} = \sum_t n_{ijt} / \sum_t n_{it}$$

Number of transitions from i to j in sequence t

Number of times in state i in sequence t

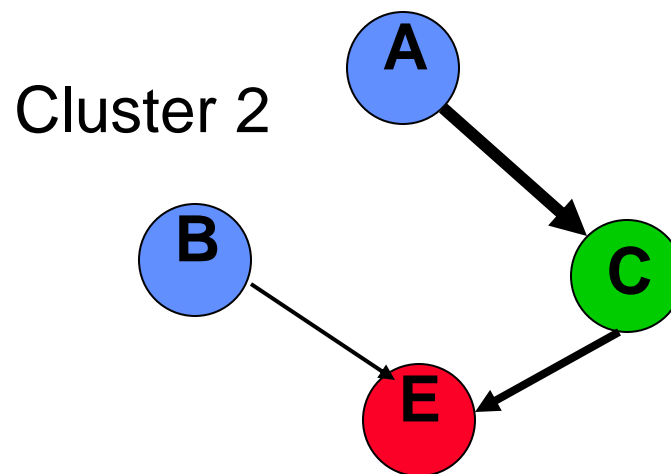
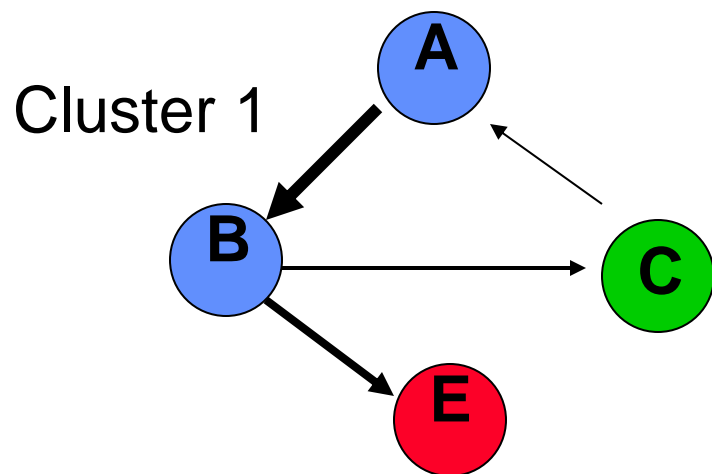
With smoothing:

$$a_{ij} = (\sum_t n_{ijt} + \alpha_{ij}) / (\sum_t n_{it} + \sum_m \alpha_{im})$$

Clusters of Markov Chains

- Assume our data is generated by K different Markov chains
 - Each chain has its own parameters A, P_0
 - This is a mixture of Markov chains
- T sequences – but we don't know which sequence came from which cluster
- Chicken-and-egg problem
 - If we knew which cluster each sequence belonged to, we could group sequences by cluster, and estimation of cluster parameters is easy
 - If we knew the parameters for each Markov chain, we could figure out (by Bayes rule) the most likely cluster for each sequence

Clusters of Probabilistic State Machines



Motivation:
approximate the heterogeneity of Web surfing behavior

EM Algorithm for Markov Clusters

EM = expectation-maximization

E-step

- For each sequence, and given current parameter estimates for each cluster, estimate $p(\text{cluster } k \mid \text{sequence})$, $k = 1, \dots, K$

M-step

- Given $p(\text{cluster } k \mid \text{sequence})$, estimate A_k and P_{k0} for each cluster

Algorithm:

- Start with an initial random guess at parameters or $p(\text{cluster } k \mid \text{sequence})$
- Iteration = pair of EM steps
- Halt iterations when parameters or $p(\text{cluster } k \mid \text{sequence})$ are not changing

Guaranteed to converge to a local maximum of the likelihood (under general conditions)

E Step of EM Algorithm

T sequences

$P(\text{cluster } k \mid \text{sequence } t)$

proportional to $P(\text{sequence } t \mid \text{cluster } k) P(\text{cluster } k)$

$P(\text{sequence } t \mid \text{cluster } k)$

$$= p_{kt} = P_{k0}(\text{first state}) \times \prod_{ij} (a_{kij})^{n_{ijt}}$$

Number of times going
from state i to state j in sequence t

↑
Current parameter estimates for cluster k

Compute these “membership” probabilities for each sequence and each cluster k.
Yields a $T \times K$ matrix of membership probabilities

M Step of EM Algorithm

For each transition probability parameter in each cluster $k=1,..K$

$$a_{kij} = \left(\sum_t n_{ijt} p_{kt} \right) / \left(\sum_t n_{it} p_{kt} \right)$$

Diagram illustrating the formula for the transition probability parameter a_{kij} in the M-step of the EM algorithm. The formula is:

$$a_{kij} = \left(\sum_t n_{ijt} p_{kt} \right) / \left(\sum_t n_{it} p_{kt} \right)$$

Annotations:

- Transition probability $i \rightarrow j$ for cluster k (points to a_{kij})
- Transitions from i to j in sequence t are “fractionally weighted” by p_{kt} , probability that sequence t came from cluster k (points to $n_{ijt} p_{kt}$)
- Number of times in state i in sequence t are “fractionally weighted” by p_{kt} , probability that sequence t came from cluster k (points to $n_{it} p_{kt}$)

Compute P_{k0} (first state) in a similar fashion

Also estimate $P(\text{cluster } k) = (1/T) \sum_t p_{kt}$

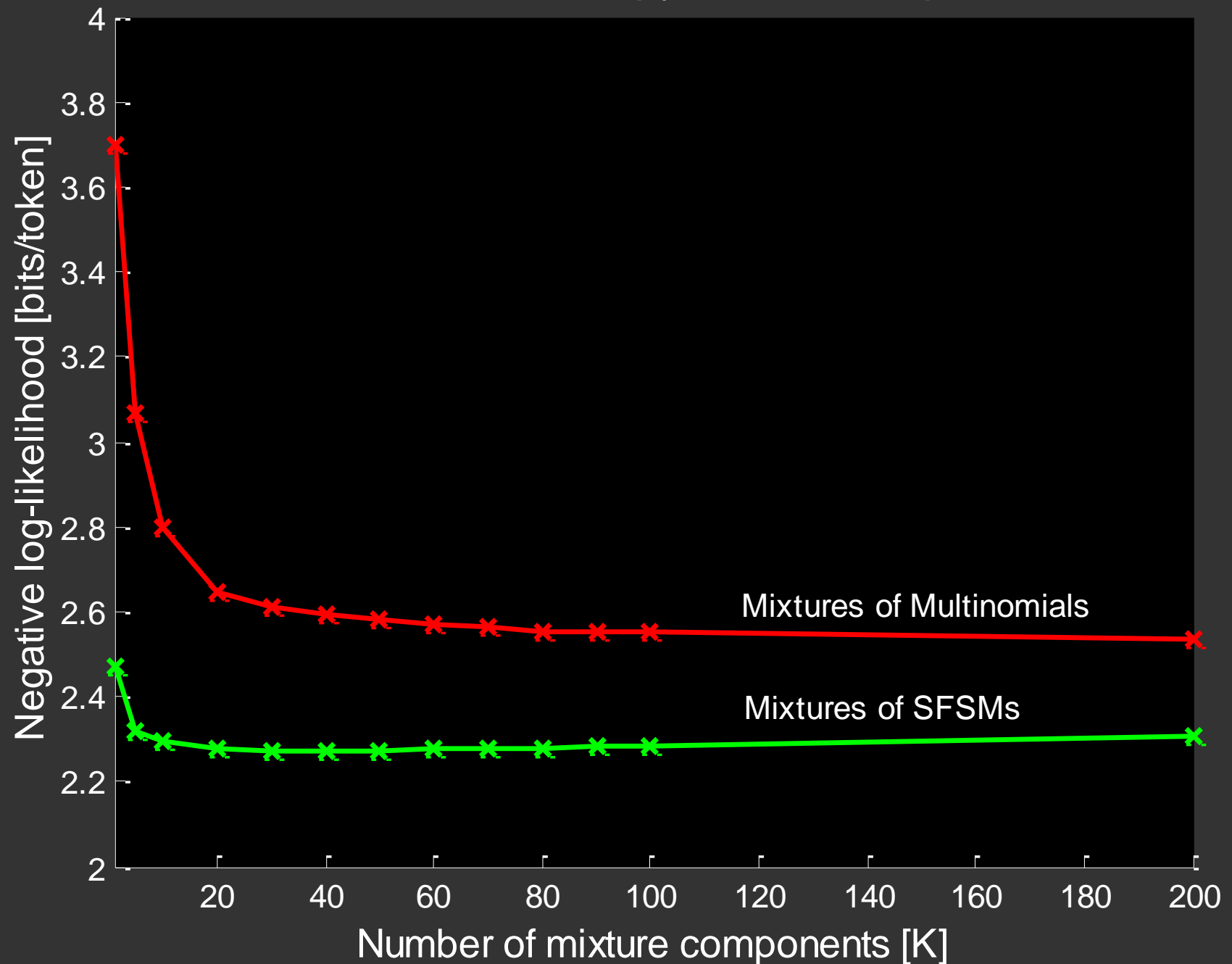
Time Complexity

- E-Step
 - T sequences, average length L
 - K clusters
 - Compute $T \times K$ matrix
 - Each entry takes $O(L)$ time to compute
 - $O(TKL)$ overall
- M-step
 - For each of M^2 transition probabilities
 - Sum over each sequence T
 - $O(T M^2)$
 - other parameters take less time

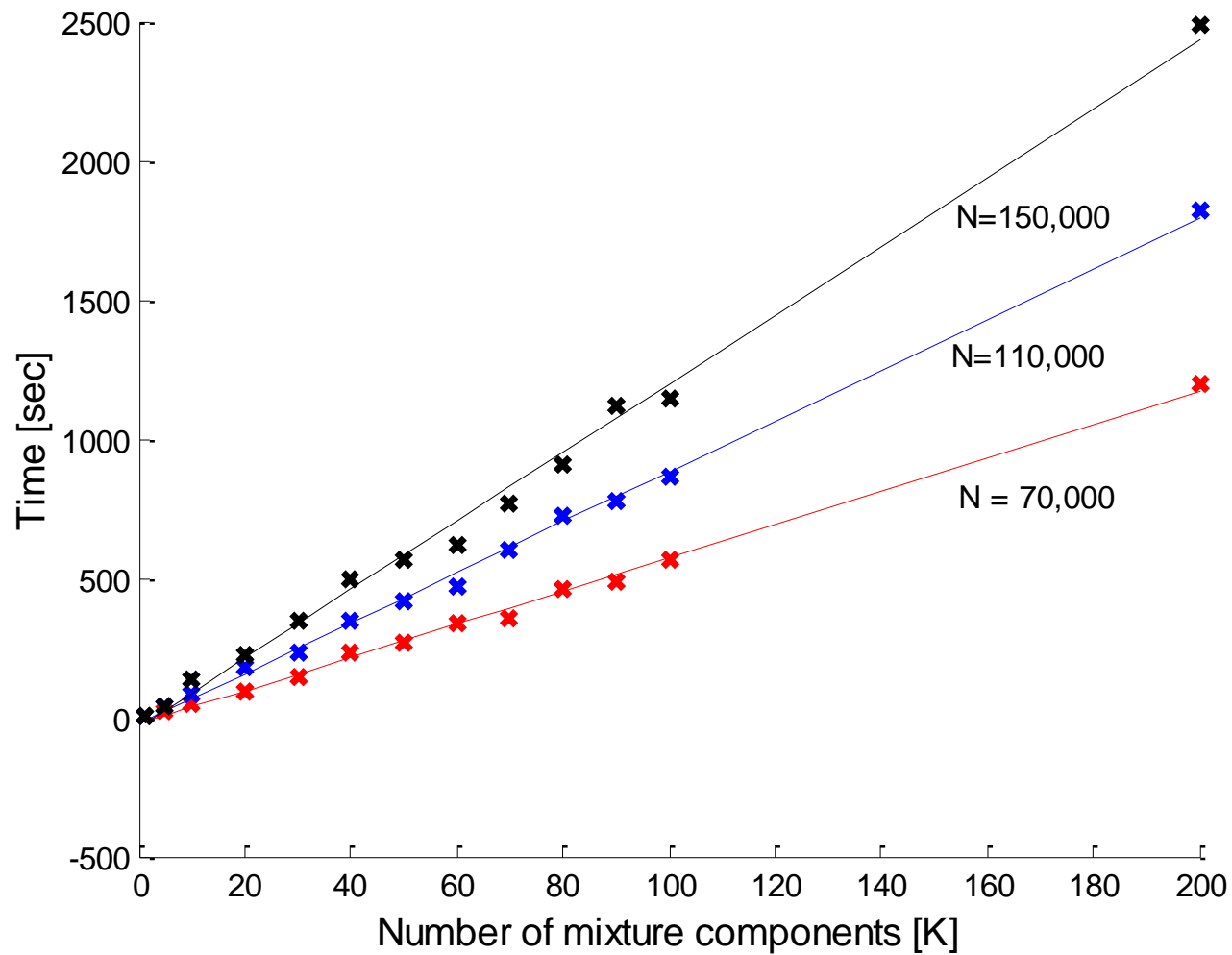
Experimental Methodology

- Model Training:
 - fit 2 types of models
 - mixtures of histograms (multinomials)
 - mixtures of finite state machines
 - Train on a full day's worth of MSNBC Web data
- Model Evaluation:
 - “one-step-ahead” prediction on unseen test data
 - Test sequences from a different day of Web logs
 - compute $\log P(\text{user's next click} \mid \text{previous clicks, model})$
 - Using equation on the previous slide
 - logP score:
 - Rewarded if next click was given high P by the model
 - Punished if next click was given low P by the model
 - negative average of logP scores \sim “predictive entropy”
 - Has a natural interpretation
 - Lower bounded by 0 bits (perfect prediction)
 - Upper bounded by $\log M$ bits, where M is the number of categories

Predictive Entropy Out-of-Sample

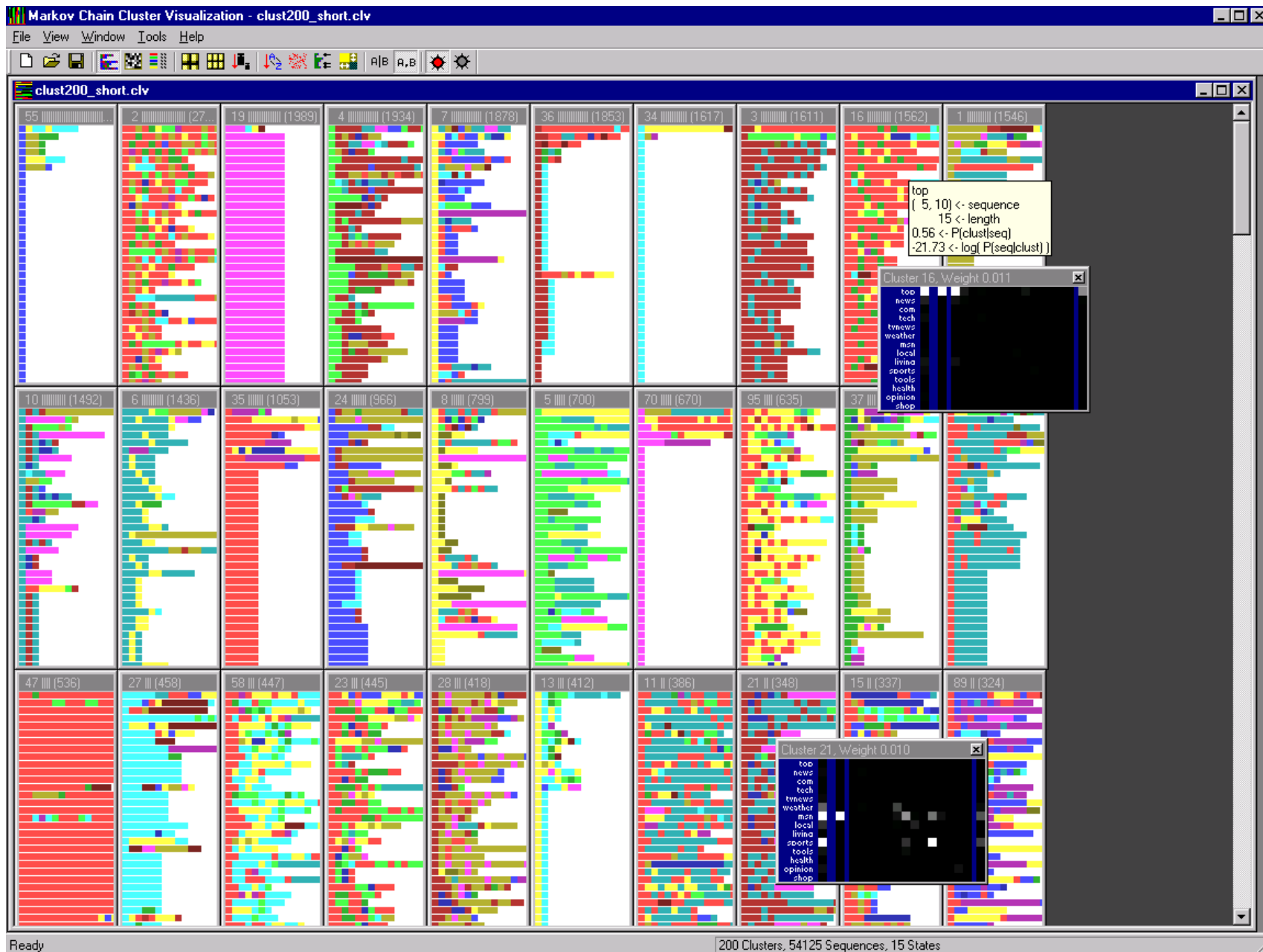


Timing Results



WebCanvas

- Software tool for Web log visualization
 - uses Markov mixtures to cluster data for display
 - extensively used within Microsoft
 - also applied to non-Web data (e.g., how users navigate in Word, etc)
 - Algorithm and visualization are in SQLServer (the “sequence mining” tool)
- Model-based visualization
 - random sample of actual sequences
 - interactive tiled windows displayed for visualization
 - more effective than
 - planar graphs
 - traffic-flow movie in Microsoft Site Server v3.0



Markov Chain Cluster Visualization - clust200_short.clv

File View Window Tools Help



clust200_short.clv



Ready

200 Clusters, 54125 Sequences, 15 States

Home Library Learn Downloads Support Community

Sign in | United States - English | Settings | Help

Search MSDN with Bing

- MSDN Library
- Servers and Enterprise Development
- SQL Server
- SQL Server 2008 R2
- Product Documentation
- SQL Server 2008 R2 Books Online
- Analysis Services - Data Mining
- Planning and Architecture
- Logical Architecture (Analysis Services - Data Mining)
- Data Mining Algorithms (Analysis Services - Data Mining)
- Microsoft Decision Trees Algorithm
- Microsoft Clustering Algorithm
- Microsoft Naive Bayes Algorithm
- Microsoft Association Algorithm
- Microsoft Sequence Clustering Algorithm**
- Microsoft Time Series Algorithm
- Microsoft Neural Network Algorithm
- Microsoft Logistic Regression Algorithm
- Microsoft Linear Regression Algorithm
- Plugin Algorithms
- Feature Selection in Data Mining
- Missing Values (Analysis Services - Data Mining)

Community Content

- Add code samples and tips to enhance this topic.

More...

Microsoft Sequence Clustering Algorithm

SQL Server 2008 R2 | Other Versions | 1 out of 1 rated this helpful | Rate this topic

The Microsoft Sequence Clustering algorithm is a sequence analysis algorithm provided by Microsoft SQL Server Analysis Services. You can use this algorithm to explore data that contains events that can be linked by following paths, or *sequences*. The algorithm finds the most common sequences by grouping, or clustering, sequences that are identical. The following are some examples of sequences:

- Data that describes the click paths that are created when users navigate or browse a Web site.
- Data that describes the order in which a customer adds items to a shopping cart at an online retailer.

This algorithm is similar in many ways to the Microsoft Clustering algorithm. However, instead of finding clusters of cases that contain similar attributes, the Microsoft Sequence Clustering algorithm finds clusters of cases that contain similar paths in a sequence.

Example

The Adventure Works Cycles Web site collects information about what pages site users visit, and about the order in which the pages are visited. Because the company provides online ordering, customers must log in to the site. This provides the company with click information for each customer profile. By using the Microsoft Sequence Clustering algorithm on this data, the company can find groups, or clusters, of customers who have similar patterns or sequences of clicks. The company can then use these clusters to analyze how users move through the Web site, to identify which pages are most closely related to the sale of a particular product, and to predict which pages are most likely to be visited next.

How the Algorithm Works

The Microsoft Sequence Clustering algorithm is a hybrid algorithm that combines clustering techniques with Markov chain analysis to identify clusters and their sequences. One of the hallmarks of the Microsoft Sequence Clustering algorithm is that it uses sequence data. This data typically represents a series of events or transitions between states in a dataset, such as a series of product purchases or Web clicks for a particular user. The algorithm examines all transition probabilities and measures the differences, or distances, between all the possible sequences in the dataset to determine which sequences are the best to use as inputs for clustering. After the algorithm has created the list of candidate sequences, it uses the sequence information as an input for the EM method of clustering.

For a detailed description of the implementation, see [Microsoft Sequence Clustering Algorithm Technical Reference](#).

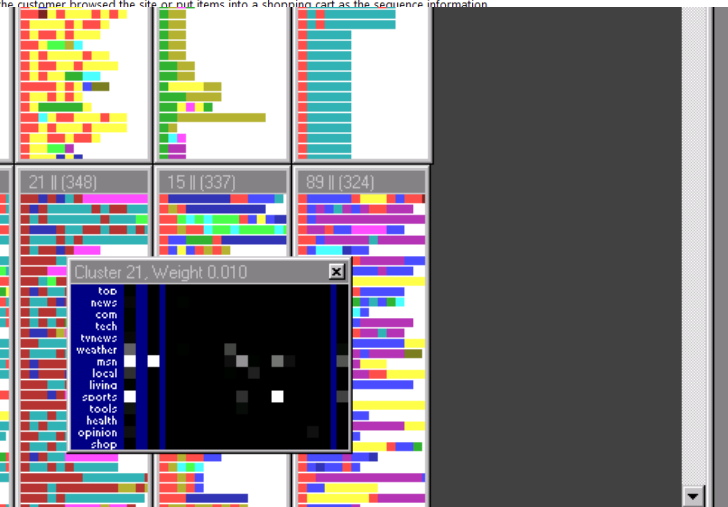
Data Required for Sequence Clustering Models

When you prepare data for use in training a sequence clustering model, you should understand the requirements for the particular algorithm, including how much data is needed, and how the data is used.

The requirements for a sequence clustering model are as follows:

- A single key column** A sequence clustering model requires a key that identifies records.
- A sequence column** For sequence data, the model must have a nested table that contains a sequence ID column. The sequence ID can be any sortable data types. For example, you can use a Web page identifier, an integer, or a text string, as long as the column identifies the events in a sequence. Only one sequence identifier is allowed for each sequence, and only one type of sequence is allowed in each model.
- Optional non sequence attributes** The algorithm supports the addition of other attributes that are not related to sequencing. These attributes can include nested columns.

For example, in the example cited earlier of the Adventure Works Cycles Web site, a sequence clustering model might include order information as the case table, demographics about the specific customer for each order as non-sequence attributes, and a nested table containing the sequence in which the customer browsed the site or put items into a shopping cart as the sequence information.



Insights from WebCanvas for MSNBC data

- From msnbc.com site administrators....
 - significant heterogeneity of behavior
 - relatively focused activity of many users
 - typically only 1 or 2 categories of pages
 - many individuals not entering via main page
 - detected problems with the weather page
 - missing transitions (e.g., tech \Leftrightarrow business)

Possible Extensions of this Approach

- Adding time-dependence
 - adding time-between clicks, time of day effects
- Uncategorized Web pages
 - coupling page content with sequence models
- Modeling “switching” behaviors
 - allowing users to switch between behaviors
 - Could use a topic-style model: users = mixtures of behaviors
 - e.g., Girolami M & Kaban A., Sequential Activity Profiling: Latent Dirichlet Allocation of Markov Chains, *Journal of Data Mining and Knowledge Discovery*, Vol 10, 175-196.