A Clustered Discrete Markov Chain Model for Web Site Traversal

Greg Ridgeway
Department of Statistics
University of Washington

Steve Altschuler
Microsoft Research

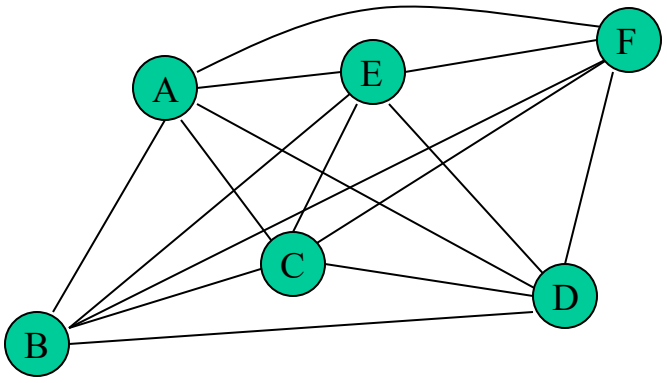
http://www.stat.washington.edu/greg

Outline

- The motivation: server side learning of web navigation
- The clustered discrete Markov model
- Parameter estimation via MCMC
- Results and computational issues
- Are we really predictable?

Learning Web Navigation

- Learn common paths
- Classify user groups
- Discover inefficiencies in complex web site designs
- Collaborative filtering



The Data

- “Cookies” uniquely identify the site users
- Web servers record all requests in log files
  - cookie id
  - date/time stamp
  - resource requested
  - URL of requesting page
- Each observation is a finite sequence of discrete states

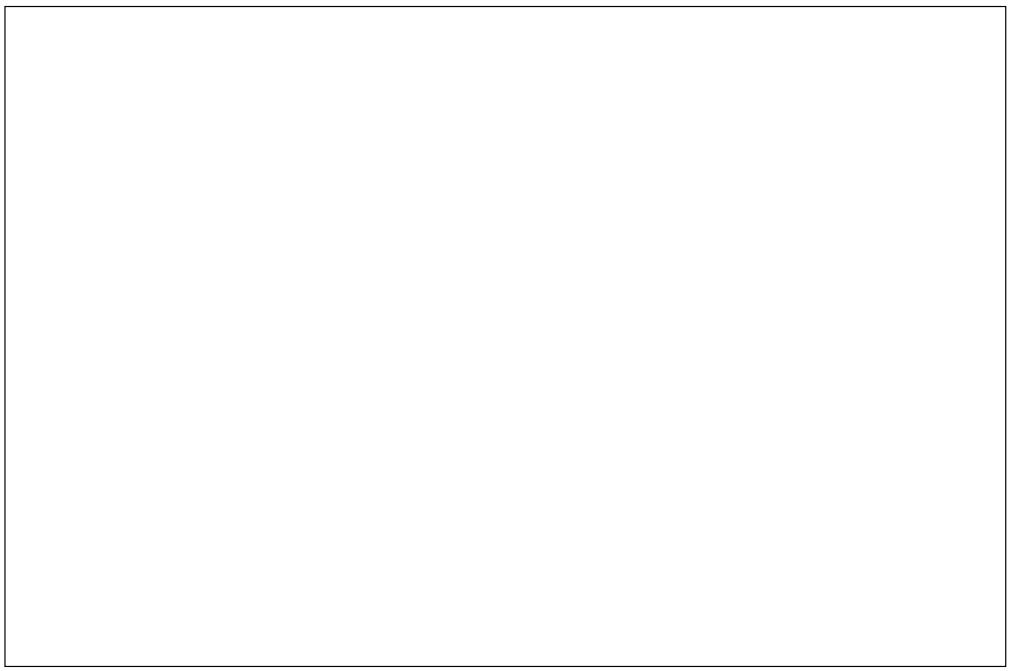
Complexities

- Browsers cache some resources, hiding some requests from the server - the “Back” button
- Users are capable of violating the intended graph structure by moving to an arbitrary node
- Dynamic Web sites

Therefore...
Defining and extracting sequences may not be trivial

Constrained EM

Hartigan's K-means algorithm



Approaches

- Shahabi, *et al* propose using the "path-mining" algorithm for *client* side learning
  - define a distance between paths based on node and link similarity
  - use K-means algorithm to "discover" clusters of paths
- Markov probability transition models

Discrete Markov Process

- Observe sequence of transitions
- Initial state distribution

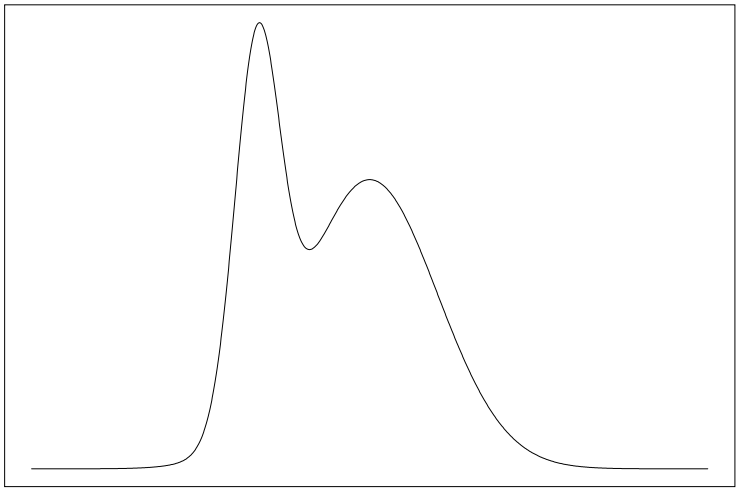
$$\begin{bmatrix} p_1 & \cdots & p_s \end{bmatrix}$$

- Probability transition matrix

$$\begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1,s} \\ P_{21} & P_{22} & & \vdots \\ \vdots & & \ddots & P_{s-1,s} \\ P_{s,1} & \cdots & P_{s,s-1} & P_{ss} \end{bmatrix}$$

Assumes all users are the same

Mixture modeling



- Can identify “true” clusters
- Increases flexibility in parametric models
- Accounts for heterogeneity in the population

Clustered Markov Process Model

- A model of a mixture of Markov transition matrices
- Assumes that one of *m* Markov transition matrices generated a process
- Unknown:
  - the initial state distributions
  - the transition probability matrices
  - the mixture proportions
  - which process came from which cluster. *If this is known then estimation is trivial*

Likelihood function

$$f(\underline{n} \mid \underline{p}, \underline{P}, \underline{\delta}) =$$

$$\prod_{k=1}^{N} \prod_{l=1}^{m} \left({}_{(l)}p_{i_0^{(k)}} \prod_{i=1}^{s} \prod_{j=1}^{s} {}_{(l)}P_{ij}^{n_{ij}^{(k)}} \right)^{\delta_l^{(k)}}$$

- 1st product is across processes
- 2nd product is across clusters
- In parentheses… likelihood associated with process k generated from cluster ℓ
- Exponent indicates whether cluster ℓ generated process k

Bayesian Estimation

Hierarchical prior for $\delta^{(k)}$

$$\delta^{(k)} \sim \text{Mult}(m, \underline{\alpha})$$

$$\underline{\alpha} \sim \text{Dirichlet}(\underline{1}_m)$$

Uniform prior on the simplex for

- Each cluster's initial state distribution
- Each row in each probability transition matrix

Sampling via MCMC

Gibbs sampling - Full conditionals

$$f({}_{(l)}p|{}_{(l)}p^{-},\underline{n}) \equiv$$

$$\text{Dirichlet}\left(1+\sum_{k=1}^{N}\delta_l^{(k)}\,\text{I}(i_0^{(k)}=1),\ \ldots,\ 1+\sum_{k=1}^{N}\delta_l^{(k)}\,\text{I}(i_0^{(k)}=s)\right)$$

$$f(\underline{{}_{(l)}P_{i\bullet}} \mid \underline{{}_{(l)}P_{i\bullet}^{-}},\underline{n}) \equiv$$

$$\text{Dirichlet}\left(1+\sum_{k=1}^{N}\delta_l^{(k)}n_{i1}^{(k)},\ldots,1+\sum_{k=1}^{N}\delta_l^{(k)}n_{is}^{(k)}\right)$$

$$f(\underline{\alpha} \mid \underline{\alpha^{-}},\underline{n}) \equiv$$

$$\text{Dirichlet}\left(1+\sum_{k=1}^{N}\delta_1^{(k)},\ldots,1+\sum_{k=1}^{N}\delta_m^{(k)}\right)$$

$$f(\delta^{(k)} \mid \delta^{(k)-},\underline{n}) \equiv$$

$$\text{Mult}\left(1,\tfrac{1}{Z}\left[\alpha_1 \bullet_{(1)}p_{i_0^{(k)}}\prod_{i=1}^{s}\prod_{j=1}^{s}{}_{(1)}P_{ij}^{n_{ij}^{(k)}},\ldots,\alpha_m \bullet_{(m)}p_{i_0^{(k)}}\prod_{i=1}^{s}\prod_{j=1}^{s}{}_{(m)}P_{ij}^{n_{ij}^{(k)}}\right]\right)$$

Example results

4829 processes from P_1

171 processes from P_2

$$P_1 = \begin{bmatrix} 0.26 & 0.43 & 0.13 & 0.18 \\ 0.06 & 0.37 & 0.19 & 0.38 \\ 0.86 & 0.05 & 0.04 & 0.05 \\ 0.32 & 0.38 & 0.20 & 0.10 \end{bmatrix} \quad P_2 = \begin{bmatrix} 0.07 & 0.17 & 0.19 & 0.57 \\ 0.12 & 0.15 & 0.08 & 0.65 \\ 0.35 & 0.03 & 0.32 & 0.30 \\ 0.27 & 0.17 & 0.14 & 0.42 \end{bmatrix}$$

10,000 Gibbs sampling iterations

$$P_1 = \begin{bmatrix} 0.26\,(0.003) & 0.43\,(0.004) & 0.13\,(0.003) & 0.18\,(0.003) \\ 0.06\,(0.002) & 0.37\,(0.003) & 0.19\,(0.003) & 0.38\,(0.004) \\ 0.86\,(0.004) & 0.05\,(0.002) & 0.05\,(0.002) & 0.05\,(0.002) \\ 0.32\,(0.004) & 0.38\,(0.004) & 0.20\,(0.004) & 0.10\,(0.003) \end{bmatrix}$$

$$P_2 = \begin{bmatrix} 0.08\,(0.02) & 0.17\,(0.03) & 0.20\,(0.02) & 0.56\,(0.04) \\ 0.10\,(0.02) & 0.14\,(0.03) & 0.10\,(0.02) & 0.66\,(0.03) \\ 0.38\,(0.04) & 0.04\,(0.01) & 0.34\,(0.03) & 0.24\,(0.03) \\ 0.27\,(0.02) & 0.18\,(0.02) & 0.14\,(0.01) & 0.41\,(0.02) \end{bmatrix}$$

- Misclassification

	$\mathbf{P_1}$	$\mathbf{P_2}$	
$\mathbf{P_1}$	4814 (99.7%)	15 (0.3%)	4829
$\mathbf{P_2}$	60 (35.1%)	111 (64.9%)	171

- Mixture proportion is 97%, 3%

Scaling

- Algorithm is rectangular in
  - number of processes
  - number of clusters
  - number of iterations
  - average process length
  - the square of the number of nodes
- Further study with 10 clusters, 100 node graph
  - 8-20 hours
  - possible label switching if clusters are “close”

Constrained EM offers computational advantages

- Makes hard cluster assignments
- Iterates until no processes are reassigned
- Good MCMC starting values
- K-means is constrained EM for normal data

☺ No label switching

☺ Convergence in minutes

☹ May converge to local maxima

Modeling Web Navigation

- Offers a reasonable and coherent model for graph traversal
- Offers parameters that are easily interpretable for web designers
- Offers a predictive model for the most likely to be next requested resource