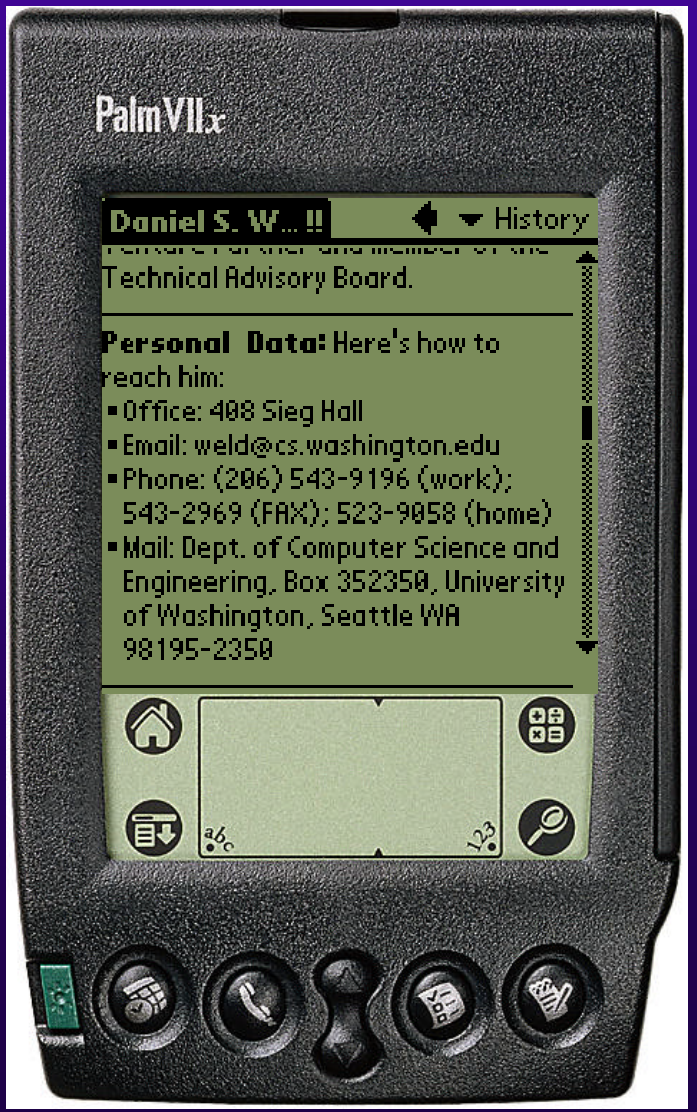
Adaptive Web Navigation for Wireless Devices

Corin R. Anderson
Pedro Domingos
Daniel S. Weld

University of Washington Seattle, WA, USA

Wireless web browsing

- Wireless web navigation is frustrating
  - Few sites are designed “wireless friendly”
  - Must scroll extensively to find link on page
  - May need to follow many links to find page

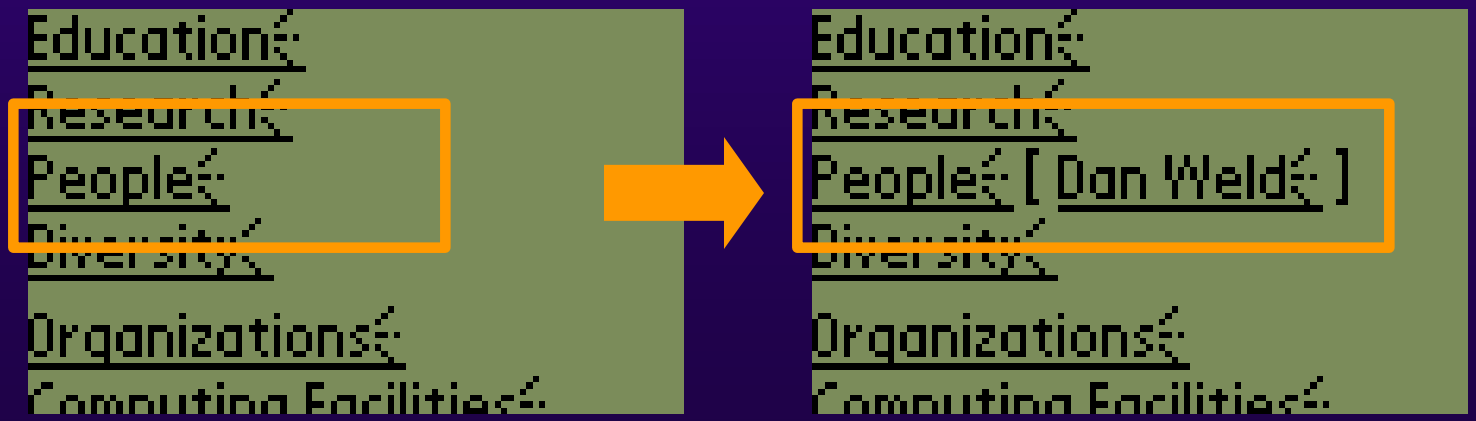


Improving the wireless web

- Web site personalizers [Anderson et al. 2001]
  - Intermediary between server and visitor
  - Adapts and customizes site for each visitor
  - Personalizations based on user model learned from web access patterns
- Personalization important in general, but particularly poignant for wireless visitors
- A key personalization: adapting navigation

Adapting navigation: shortcuts

- Focus on information gathering tasks
  - Users look for info on a particular page
  - Info-tasks dominate wireless behavior
- Idea: provide a shortcut link directly to destination



This afternoon, I will…

- Formalize the shortcut selection problem
- Present our MinPath algorithm for finding shortcuts
- Describe a variety of web behavior models employed by MinPath
- Discuss experimental evidence supporting the MinPath approach

Trails

- A trail is a sequence of page requests…

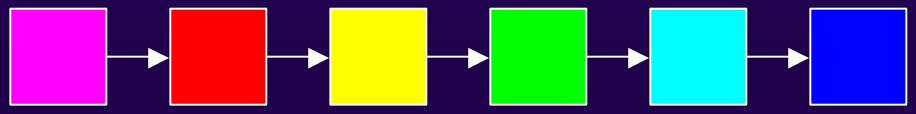
$$< p_0, p_1, p_2, \ \ldots, p_n >$$

- …coherent in time…

$$\text{time}(p_{i-1}) < \text{time}(p_i) < \text{time}(p_{i-1}) + window$$

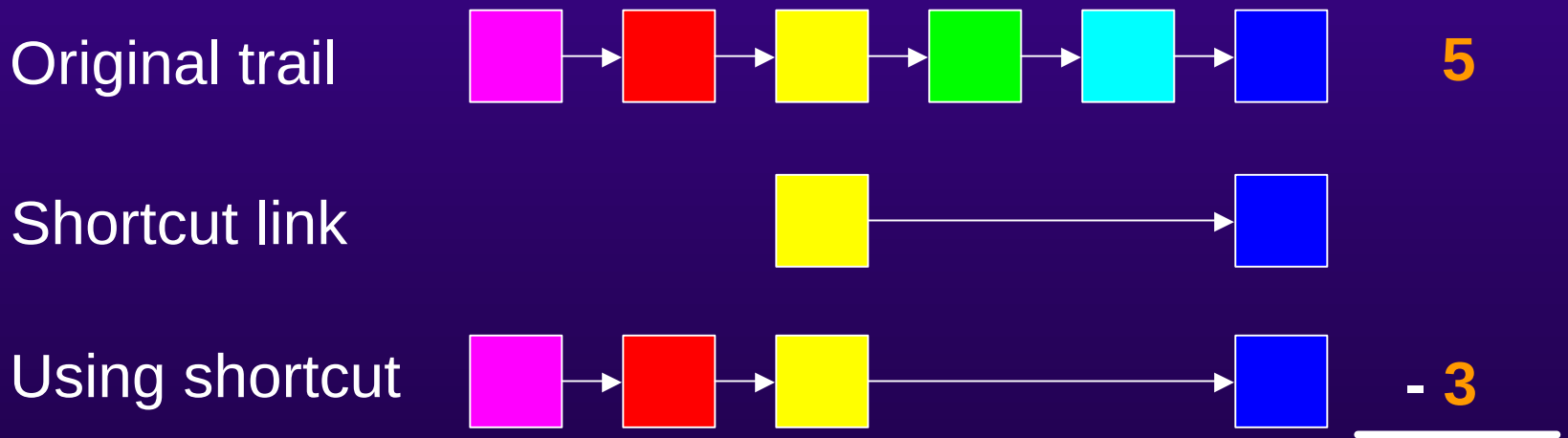
- …and coherent in space

$$\$ \text{ link } p_0 \rightarrow p_1, p_1 \rightarrow p_2, p_2 \rightarrow p_3, \ldots$$

“  ”

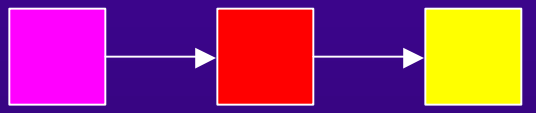
Shortcut link

- Connects previously unconnected pages
- Savings is # links skipped in given trail

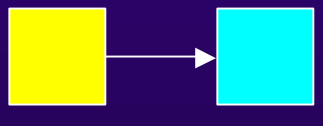


Shortcut link selection problem

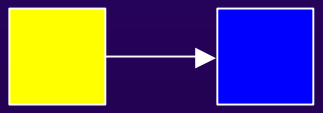
- Given:
  - trail prefix $<p_0, \ldots, p_i>$
  - visitor's past trails
  - trails of other visitors
  - maximum number of shortcuts m
- Output:
  - list of shortcuts that minimizes the expected number of links to the destination



$p_i \rightarrow q_1$



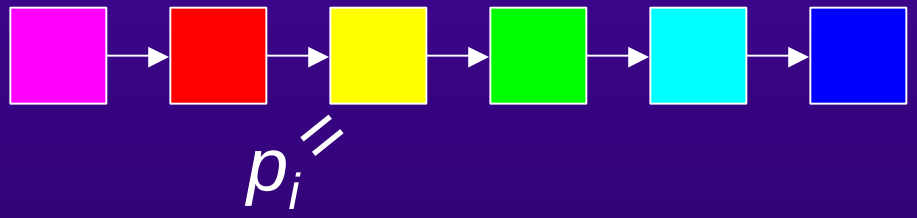
$p_i \rightarrow q_2$



…

Finding shortcuts

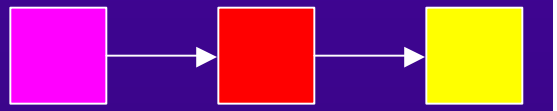
- If we know the whole trail…

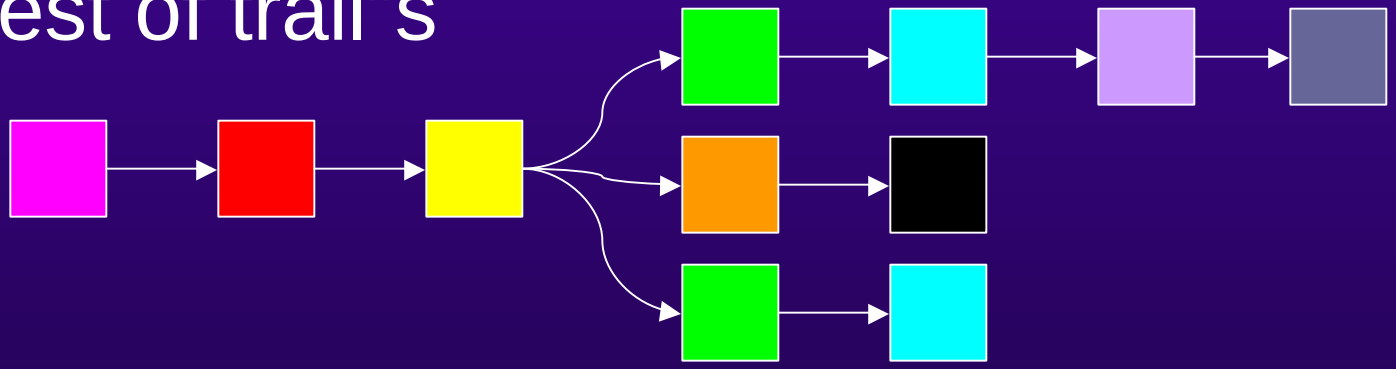


- …finding the right shortcut is easy

- Unfortunately, omniscience is hard to come by

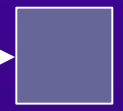
MinPath approach

- All we really know is the prefix 
- MinPath: conceptually try all possible "rest of trail"s

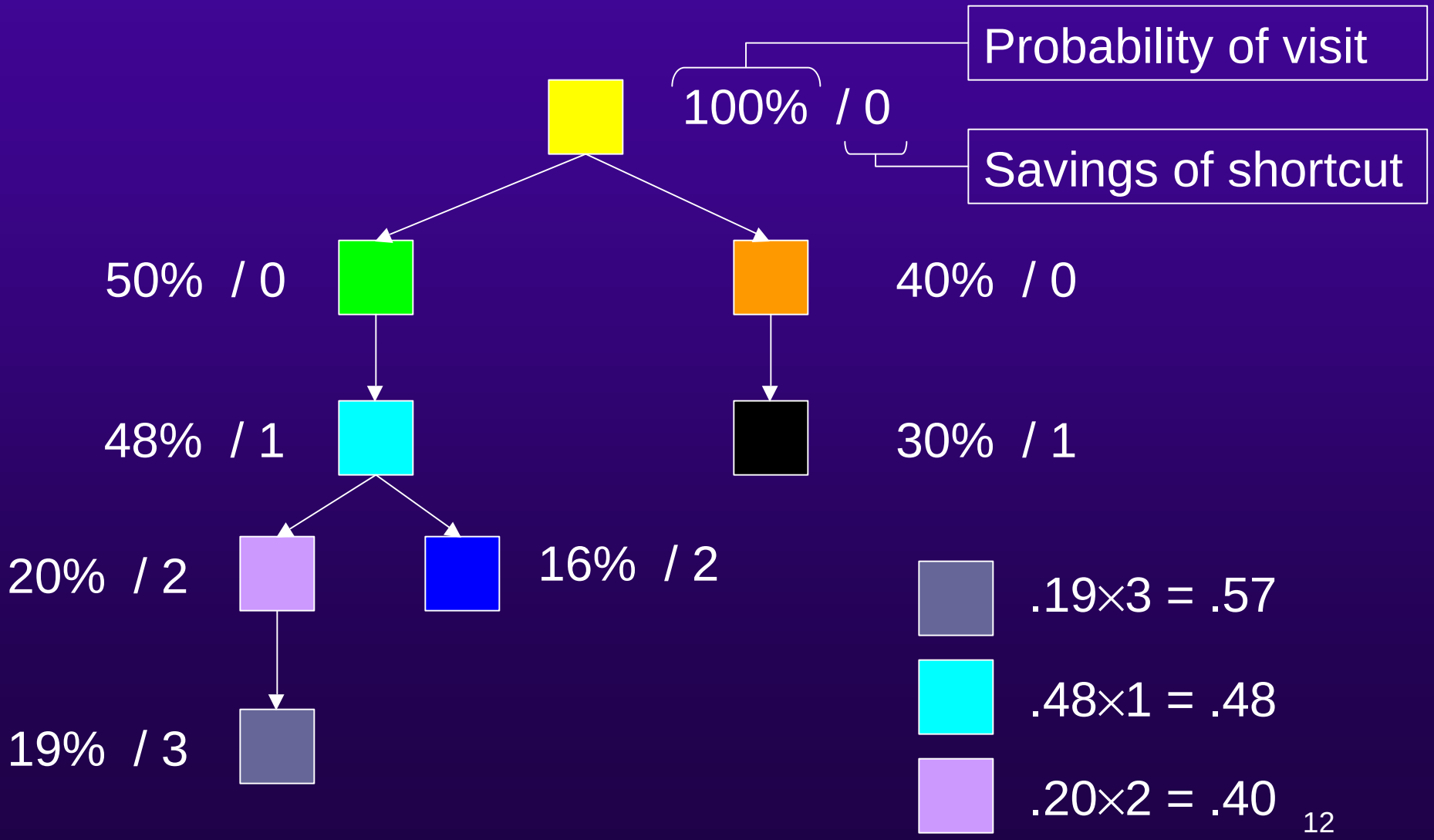


- Each suggests a shortcut and savings
- Each has some probability
- Product is expected savings; take top m_{10}

Calculating trail probability

- Goal: compute P(destination) ... 
- But training data is sparse
  - Relatively little data compared to number of possible destinations
- Instead, compute P(next request)
- Compose predictions to build “rest of trail”

Traversing for expected savings



Predictive model

- At heart is predictive model of navigation:

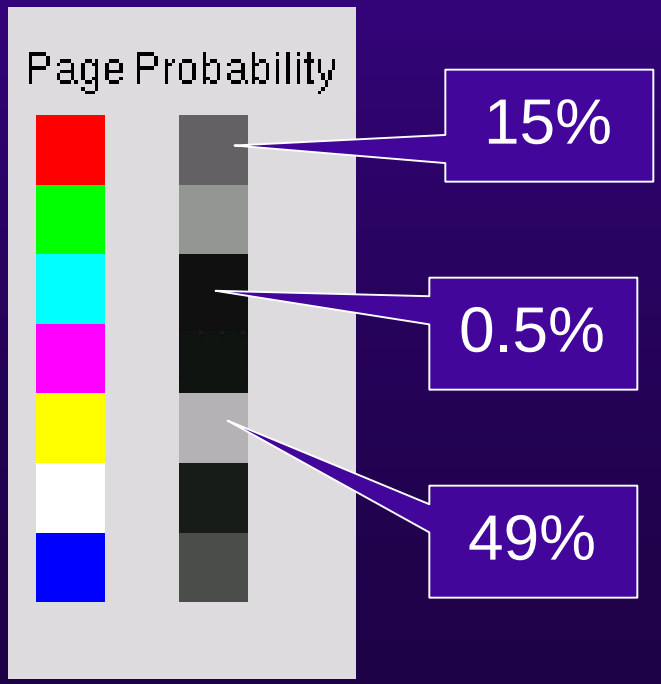
$$P(\text{next request} = \boxed{?} \mid \textit{stuff})$$

- "Stuff" can include:
  - Nothing at all!
  - Relation to other visitors (cluster membership)
  - Sequential information (pages in current trail)
  - Cluster and sequential information

Unconditional model

- Ignore all that stuff!

$$P(\text{next request} = q) = \frac{\#\text{ times } q \text{ requested in the past}}{\text{Total \# pages requests in the past}}$$



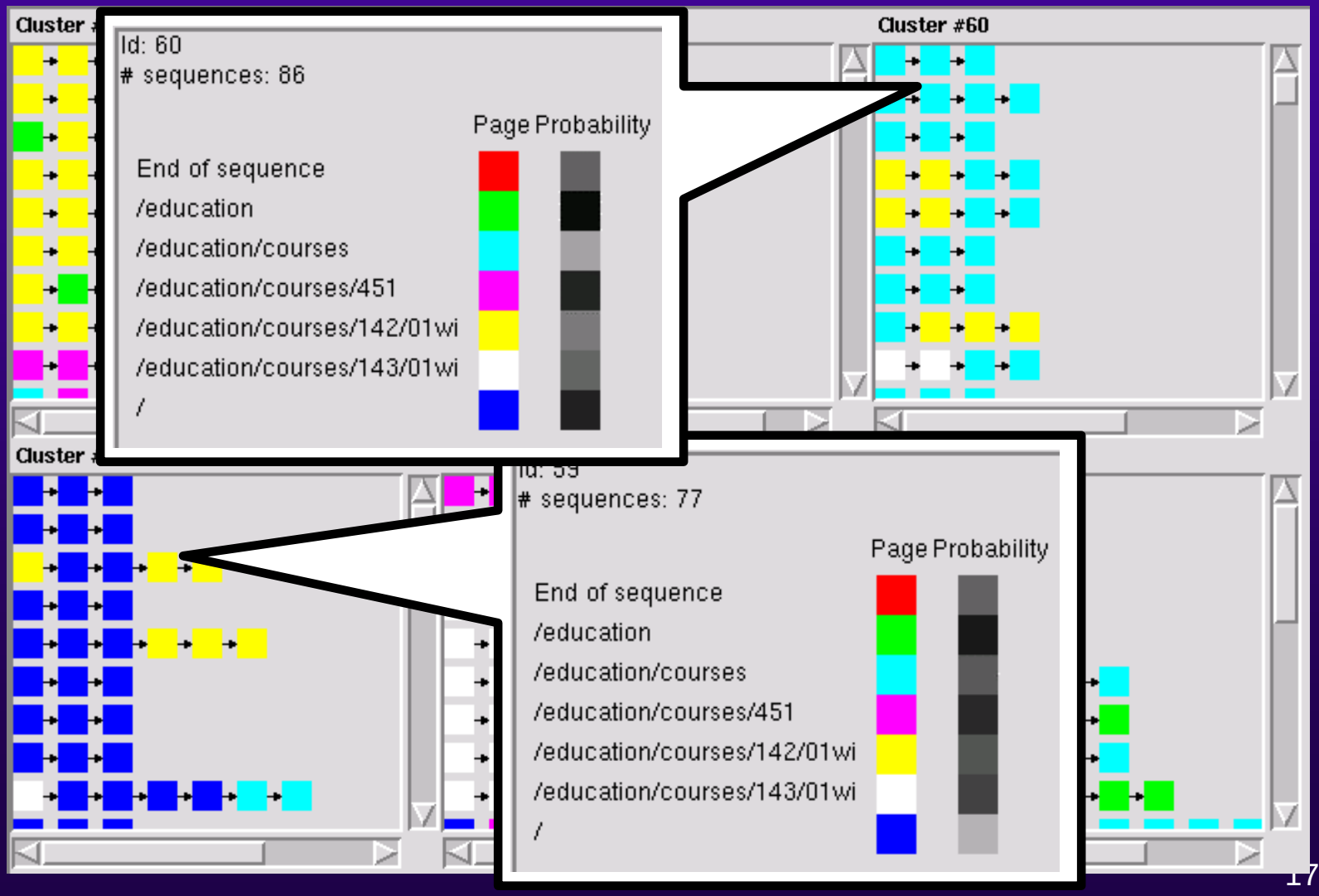
Assuaging data sparseness

- Seldom-visited URLs difficult to estimate
- Instead, aggregate URL usage by prefix
- At performance time, MinPath:
  1. Computes prefixes for all links on page
  2. Computes probability for each unique prefix
  3. Normalizes these probabilities
  4. Assigns probabilities to links with same prefix using uniform priors

Clustering visitors' data

- Idea: cluster behavior of all visitors, and condition probability on cluster membership
- Replace single model with mixture model
  - Offline, use EM to cluster trails, build models
  - At runtime, assign current visitor to clusters
  - Use corresponding mixture of models
- Simplest cluster model: unconditional
  - Naïve Bayes mixture model [AutoClass]

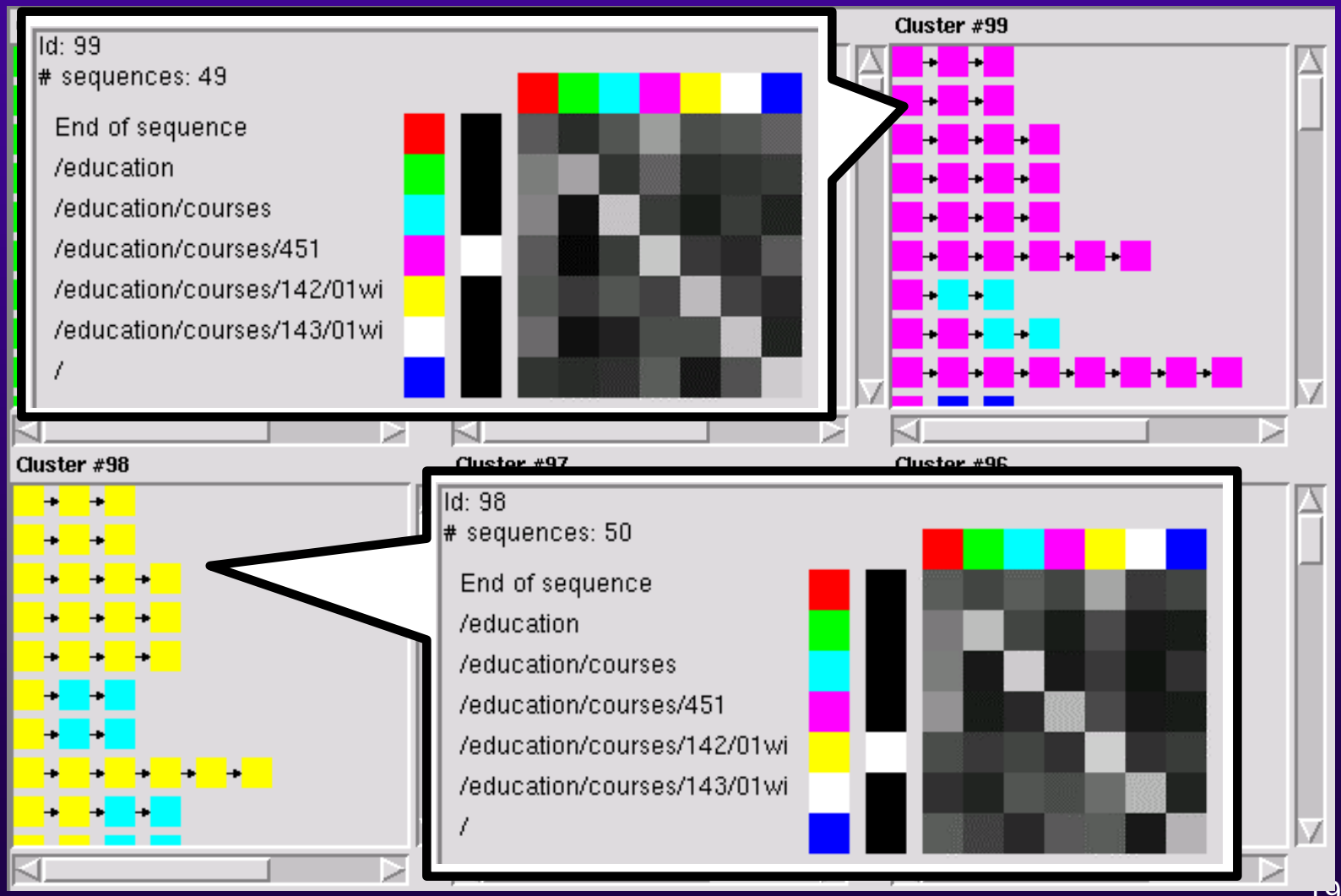
Naïve Bayes mixture



Page sequence: Markov models

- Condition on sequence of pages in trail
- First order: one page of history
- Markov model states are pages, transitions are links
- Markov + clustering = Mixtures of Markov models

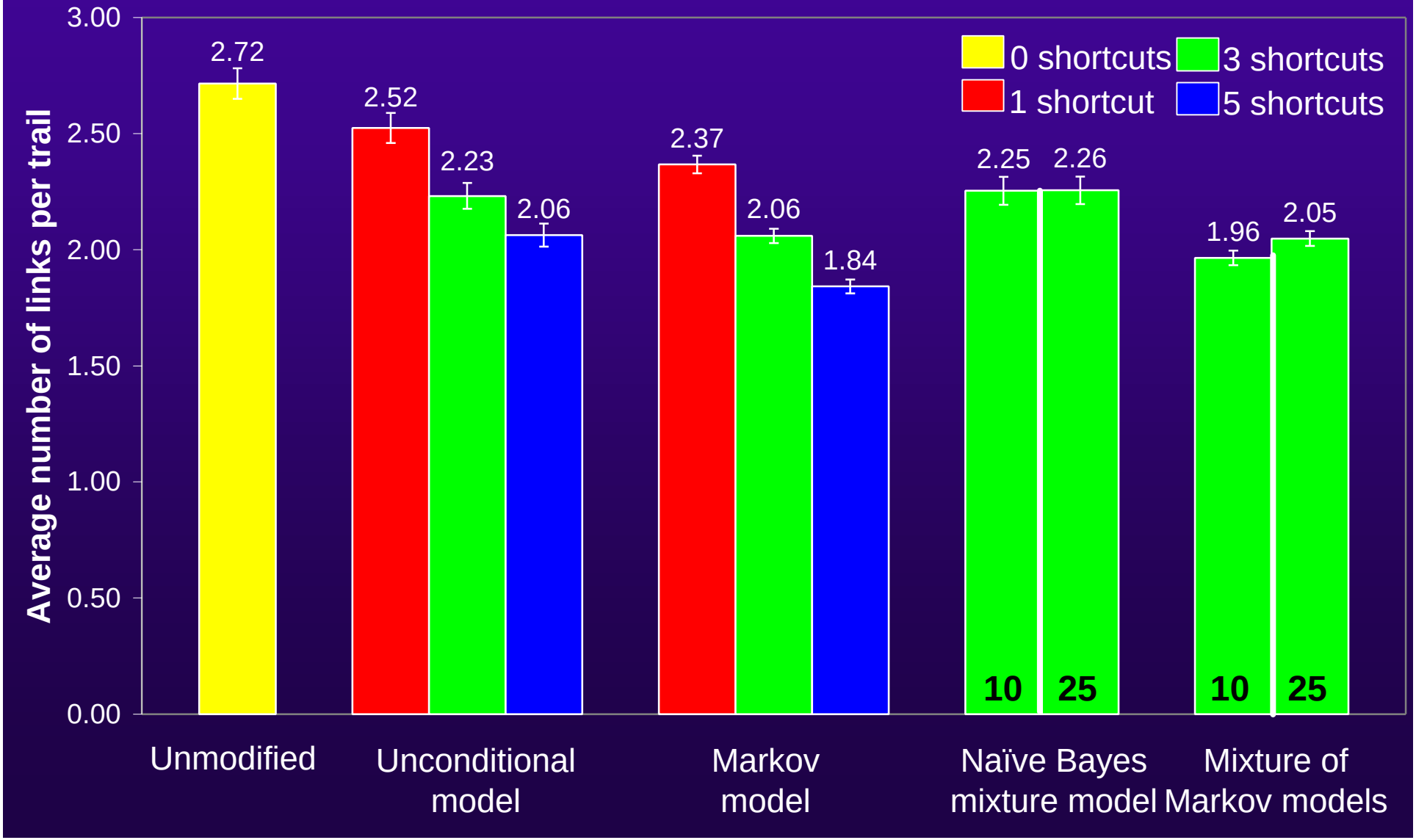
Mixtures of Markov models



Experiments

- Use real-world data (www.cs, Sept. 2000)
- Train on 20 days logs (35,000 trails)
- Test on 1.5 days (2,500 trails)
- Consider only trails with link length > 2
  - Short trails can't be shortened any further!
- Performance is # links saved while reaching destination

MinPath's performance



Mixture model assignment

- How do we assign current visitor to clusters?
  - Hard (assign to one cluster) or soft (many)
  - Use or ignore visitor's past trails
  - Use or ignore visitor's current-trail prefix
- Results:
  - Soft assignment, using current-trail prefix, but ignoring past trails is best

Related work

- Adapting site by mining usage logs
  - PageGather and IndexFinder [Perkowitz & Etzioni]
- Personalization agents & recommenders
  - The Daily Learner [Billsus, et al.]; WebWatcher [Joachims, et al.]
  - Letizia [Lieberman]; SurfLen [Fu, et al.]
- Sequence clustering
  - WebCANVAS [Cadez, et al.]

Ongoing work

- Intelligently choosing anchors for shortcuts
  - Concise but descriptive
- Considering other adaptations
  - Real-time approach for content elision
- Employ a declarative model of site
  - Adapt site at “higher level”
- Applying ideas to adaptive user interface
  - Web site, user interface have analogous parts
  - How well do adaptivity ideas carry over?

Summary

- Wireless web today is frustrating
- MinPath improves navigation by finding shortcut links
  - Selects shortcuts by expected savings
  - Predicts destinations by predicting each navigation step separately
  - Builds mixture models using all visitors' data
- Impact: MinPath finds shortcuts in real time, realizes 44% of possible savings