Generating Intelligent Links to Web Pages by Mining Access Patterns of Individuals and the Community

Benjamin Lambert and Omid Fatemieh
University of Illinois at Urbana-Champaign

The Problem

• How can we best assist a person browsing the Web by providing links to the pages that they are looking for?
• There are many reasons to do this, for example:
  1. Pages hidden in a large web site.
  2. Help people find relevant pages that there is no link to them, or are a few clicks away.
     • Seminar announcements.
• New ideas for solving this problem:
  • Using recent activity to make recommendations.
  • Using the contents of Web pages to make recommendations.
  • Combining data mining and user modeling approaches.
  • Using a machine learning approach.

The Big Picture

Web server logs:
• NASA Kennedy Space Center Web server log collected over July and August 1995.
• CS Department Web server logs from Nov 1, 2003 to Nov 11, 2003.
• The logs are long lists of Web page requests, each request is represented by:
  • IP, time and date, the requested page, etc.
  • Actual IP addresses were removed for privacy reasons.
  • The following are discarded:
    • Requests for files of type .jpg, .css, etc.
    • Requests from crawlers (robots.txt).
    • Unsuccessful GETs (code 200 only).
    • Refreshes (consecutive requests for the same page).
• The logs are long lists of Web page requests, each request is represented by:
  • IP, time and date, the requested page, etc.
  • Actual IP addresses were removed for privacy reasons.
  • The following are discarded:
    • Requests for files of type .jpg, .css, etc.
    • Requests from crawlers (robots.txt).
    • Unsuccessful GETs (code 200 only).
    • Refreshes (consecutive requests for the same page).

Data

• Web server logs:
  • CS Department Web server logs from Dec 6, 2004 to Feb 28, 2005.
  • NASA Kennedy Space Center Web server log collected over July and August 1995.
  • CS Department Web server logs from Nov 1, 2003 to Nov 11, 2003.
• The logs are long lists of Web page requests, each request is represented by:
  • IP, time and date, the requested page, etc.
  • These provide the user with links to the most frequently clicked links on the current page.
• The logs are long lists of Web page requests, each request is represented by:
  • IP, time and date, the requested page, etc.
  • These provide the user with links to the most frequently clicked links on the current page.
• The logs are long lists of Web page requests, each request is represented by:
  • IP, time and date, the requested page, etc.
  • These provide the user with links to the most frequently clicked links on the current page.

Markov Model

• Recommendations by a Markov Model:
  • We implemented a recommending model using first order Markov Models.
  • These provide the user with links to the most frequently clicked links on the current page.
• Recommendations by a Markov Model:
  • We implemented a recommending model using first order Markov Models.
  • These provide the user with links to the most frequently clicked links on the current page.
• Recommendations by a Markov Model:
  • We implemented a recommending model using first order Markov Models.
  • These provide the user with links to the most frequently clicked links on the current page.

Methodology

• Machine Learning
  • We use the SNoW learning architecture to learn multi-class classifiers to recommend Web pages.
  • Training and testing examples are composed from the pages a user has already visited this session.
  • We use SNoW to extract features for each example. Feature types include: unigram, bigram, and “sparse collocation.”
  • The target class is the next page.
  • Classifiers are trained with: winnow, perceptron, and naive Bayes.
• Dynamic Markov Model
  • We implemented a “dynamic first order M.M.”
  • By sequential pattern mining we can:
    • Recommend shortcuts: if A, B, C, D, E occurs frequently, we may consider adding a shortcut from A to E, e.g.:
      • /Æ/Æ/Æ/info/facultypositions.php : (0.00732)
    • The relative support for PrefixSpan has to be:
      1. Small to get many recommendations.
      2. Large to get good quality recommendation.
• Sequential Pattern Mining
  • By sequential pattern mining we can:
    • Recommend shortcuts: if A, B, C, D, E occurs frequently, we may consider adding a shortcut from A to E, e.g.:
      • /Æ/Æ/Æ/info/facultypositions.php : (0.00732)
    • We used the PrefixSpan (Han et al.) for mining frequent sequential patterns.
    • The PrefixSpan mining has to be:
      1. Small to get many recommendations.
      2. Large to get good quality recommendation.
• Implementation and Evaluation
  • Implementation:
    • We crawled the CS Department Web site and used JavaScript to modify the links so that clicking on a link causes our Perl script to execute.
    • Our Perl script uses the pre-built model, as well as user’s recent history (using cookies) to make recommendations in the left frame.
    • The Perl script fetches the actual requested page and puts it in the right frame.
  • Evaluation:
    • The machine learning algorithms and Markov models get a hit if one of their top k (1-3) recommendations is next.
    • The mined patterns get a hit if the user:
      • requests the recommended page this session
      • remains on the page for at least t seconds
• Conclusion
  • Each of the proposed approaches can predict different “next clicks.”
    1. Markov Model makes recommendations based on what most of the other users have clicked from the current page.
    2. The dynamic Markov model is good for cases in which for some reason many people rush to the web site, looking for the same page.
    3. The machine learning algorithm is smarter and takes the user’s recent activities (and also content of pages) into account as well.
    4. The sequential pattern mining algorithm can help in predicting pages that are a few clicks away or have no direct link to them.
    5. The accuracy of the Sequential Pattern Mining algorithm is higher, but each method has its merits:
      • A combination of the above approaches may best meet users’ needs.

This research is supported by NSF grant EIA-0224453 and the cluster computing equipment donated by AMD.