Microsoft Research

User Browsing Behavior-driven Web Crawling

Minghai Liu, Rui Cai, Ming Zhang, and Lei Zhang

Microsoft Research, Asia

School of EECS, Peking University

Microsoft Research

PEKING UNIVERSITY

Ordering Policies for Web Crawling

- Ordering policy
  - To prioritize the URLs in a crawling queue
  - The key is importance measure of a URL

- Existing policies adopt various hypotheses of URL importance

Link Structure	Semantic-driven	Site-level
• Breadth-first • In-degree • PageRank and its derivatives	• Topical crawler • Focused crawler • Search impact	• Structure-driven • Forum crawling

Limitations

- **General** policies (link structure-based) cannot optimize the performance of a particular website
  - The Web becomes more dynamic, deep, and complex
  - URLs with low PageRank scores still attract considerable traffic
- **Specific** policies (semantic-driven and site-level) cannot be scaled up to the whole Web
  - Heavy maintain cost, and unaffordable human efforts
- Just characterize user interest indirectly and incompletely
- How to predict the importance of newly created (unseen) URLs?

Microsoft Research

User Browsing Behavior from Log Data

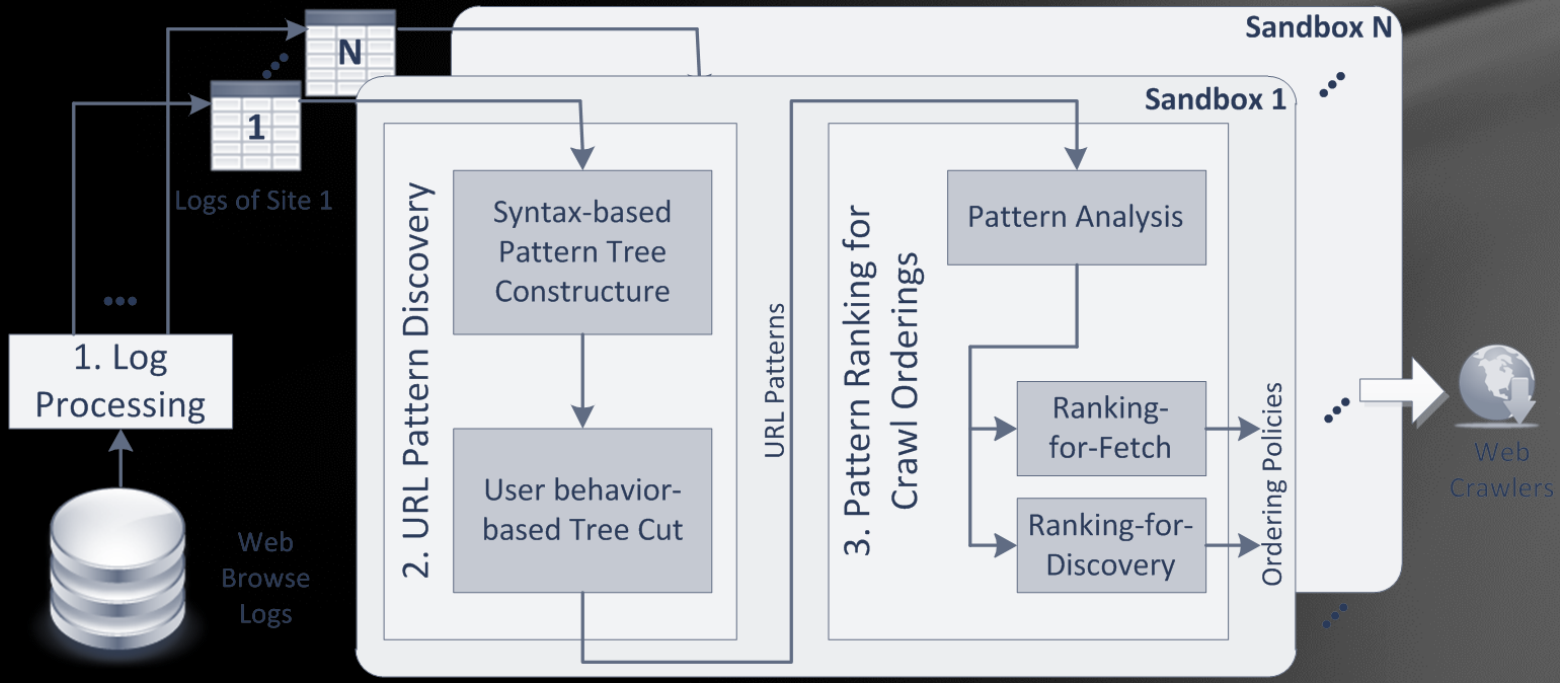
- As another valuable information to guide crawling
  - Directly reflect user interest
  - Rich knowledge, covers most important sites on the Web
- How to leverage log data in crawling?
  - Simply prioritizing a URL according to its frequency being recorded in the log? Impractical!
    - Log data is quite sparse, covers less than 10% URLs in a website
    - User behaviors on a single URL are noisy and unstable
    - URLs retired very rapidly
  - Aggregate log data through data mining

Our Idea — URL Pattern

- Summarize log data with URL patterns, and design crawl ordering policies at pattern-level
  - URLs in a website follow syntax schemas defined by its designers
  - URLs belonging to the same pattern act similar functional roles
- Benefits of URL patterns
  - Robust to noise, steady in a relatively long period, generalized to predict unseen URLs
  - Go deep to optimize site-specific performance, and go wide to provide a web-scale solution
- Technical obstacles
  - How to determine the granularities of URL patterns?
    - Coarse – cannot distinguish URLs with different user behaviors
    - Subtle – overfitting and poor generalization ability
  - How to leverage URL patterns to design ordering policies?

Framework Overview

- Log data format — triple
  - $<URL_t, URL_r, GUID>$
- System framework — run in parallel



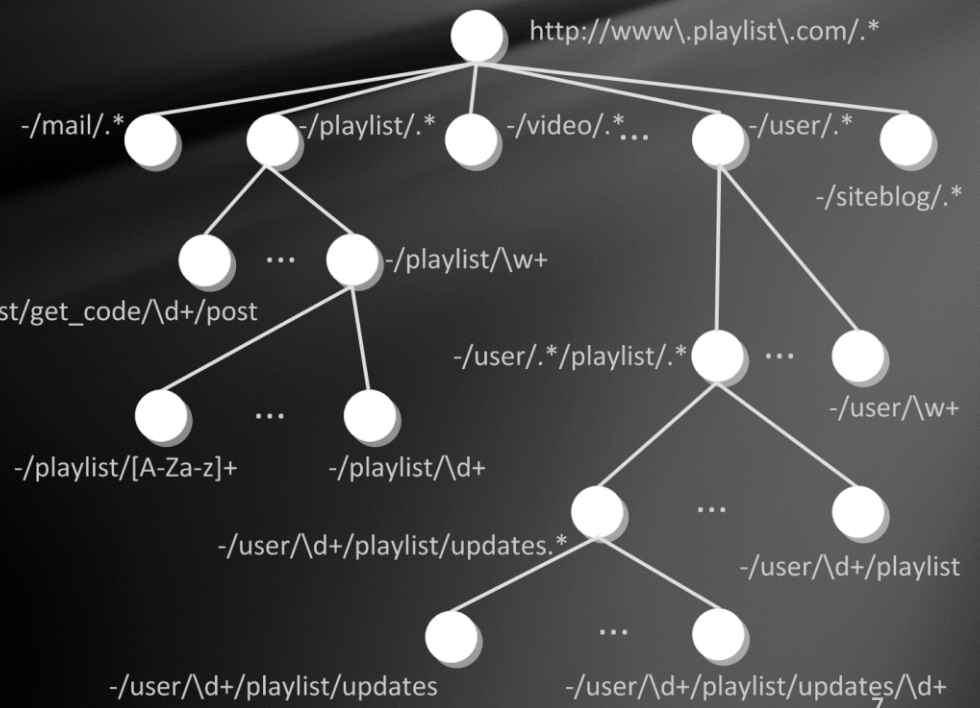
Algorithm: Pattern Tree Construction

- URL decomposition
  - <key, value> pair
  - RFC 3986

http://www.playlist.com/mail/compose?recipient=mike

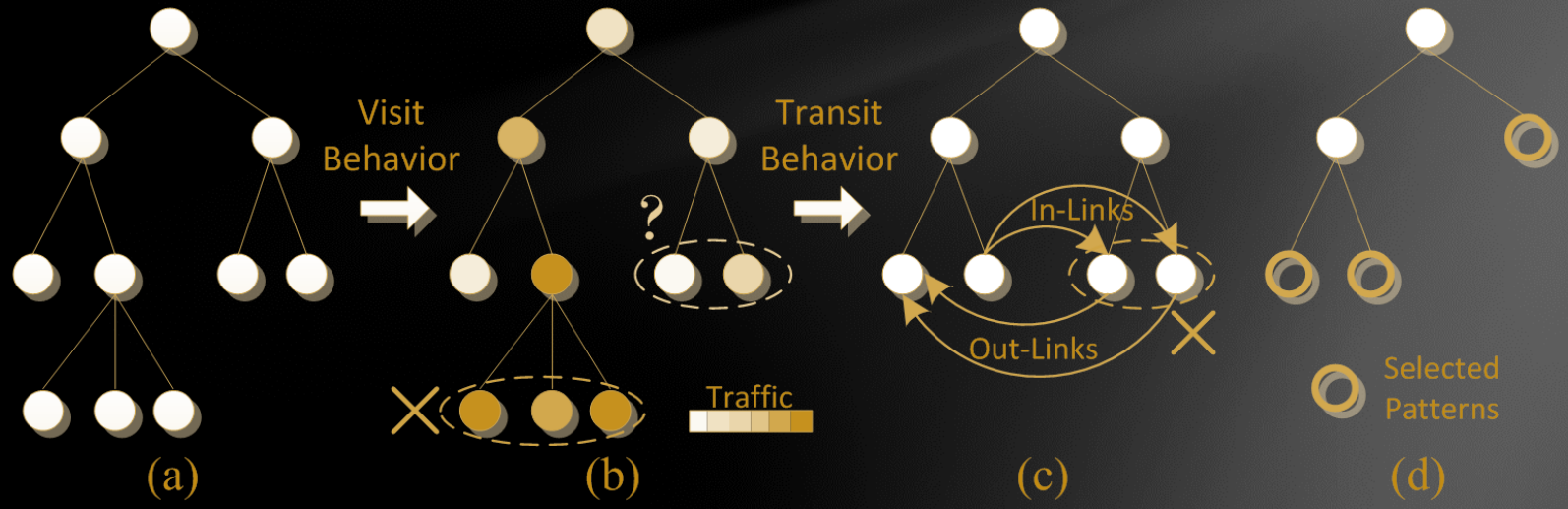
Component (Key)	Value
Protocol	http
Authority	www.playlist.com
Path_Level_0	mail
Path_Level_1	compose
Query_Recipient	mike

- Pattern tree construction
  - A top-down process
  - Considering the distribution of values under a particular key
  - Split URLs according to the key which has the most concentrated distribution in each iteration
  - Lei et al. WWW 2010



Algorithm: Pattern Selection

- Cut the pattern tree and stop at the levels, at which different tree nodes (patterns) have different user browsing behaviors
  - Two behaviors: ***visit*** (content pages) and ***transit*** (hub pages)
- Current solution: two steps
  - visit-based and transit-based tree-cuts



Algorithm: Pattern Ranking

- Two Crawling Scenarios
  - Comprehensively fetching a website (batch mode)
  - Timely discovering new content (incremental mode)
    - Monitor "hub" pages
    - Crawl news / forum / social network sites
- How to rank patterns?
  - Behavior graph
    - The browsing structure among URL patterns
    - The transition probabilities are based on user voting
  - Rank with HITS but NOT PageRank
    - The authority and hub scores perfectly match the two aforementioned scenarios

Experimental Results

- Nice advantages of URL patterns
  - Generalization ability: cover 99% URLs in a website
  - Distinguishability: URLs from the same pattern are consistent on page layouts
  - Summarization ability: pattern-level traffic distribution can well approximate the raw URL-level log data
  - Temporal reliability: still cover 90% URLs after 6 months
- Better crawling performance
  - Detailed comparisons please refer to the paper
  - The algorithms have been successfully shipped to Bing

Thanks!

Q & A

More information please visit

http://research.microsoft.com/en-us/projects/website_structure_mining/