DATA MINING - 1DL105, 1DL111

Fall 2007

An introductory class in data mining

http://user.it.uu.se/~udbl/dut-ht2007/
alt. http://www.it.uu.se/edu/course/homepage/infoutv/ht07

Kjell Orsborn
Uppsala Database Laboratory
Department of Information Technology, Uppsala University,
Uppsala, Sweden

Introduction to Data Mining: Web Mining

(slides + supplemental articles)
ref book (used for slides): Data mining / Dunham

Kjell Orsborn

Department of Information Technology
Uppsala University, Uppsala, Sweden

Web Mining Outline

Goal: Examine the use of data mining on the World Wide Web

- Introduction
- Web Content Mining
- Web Structure Mining
- Web Usage Mining

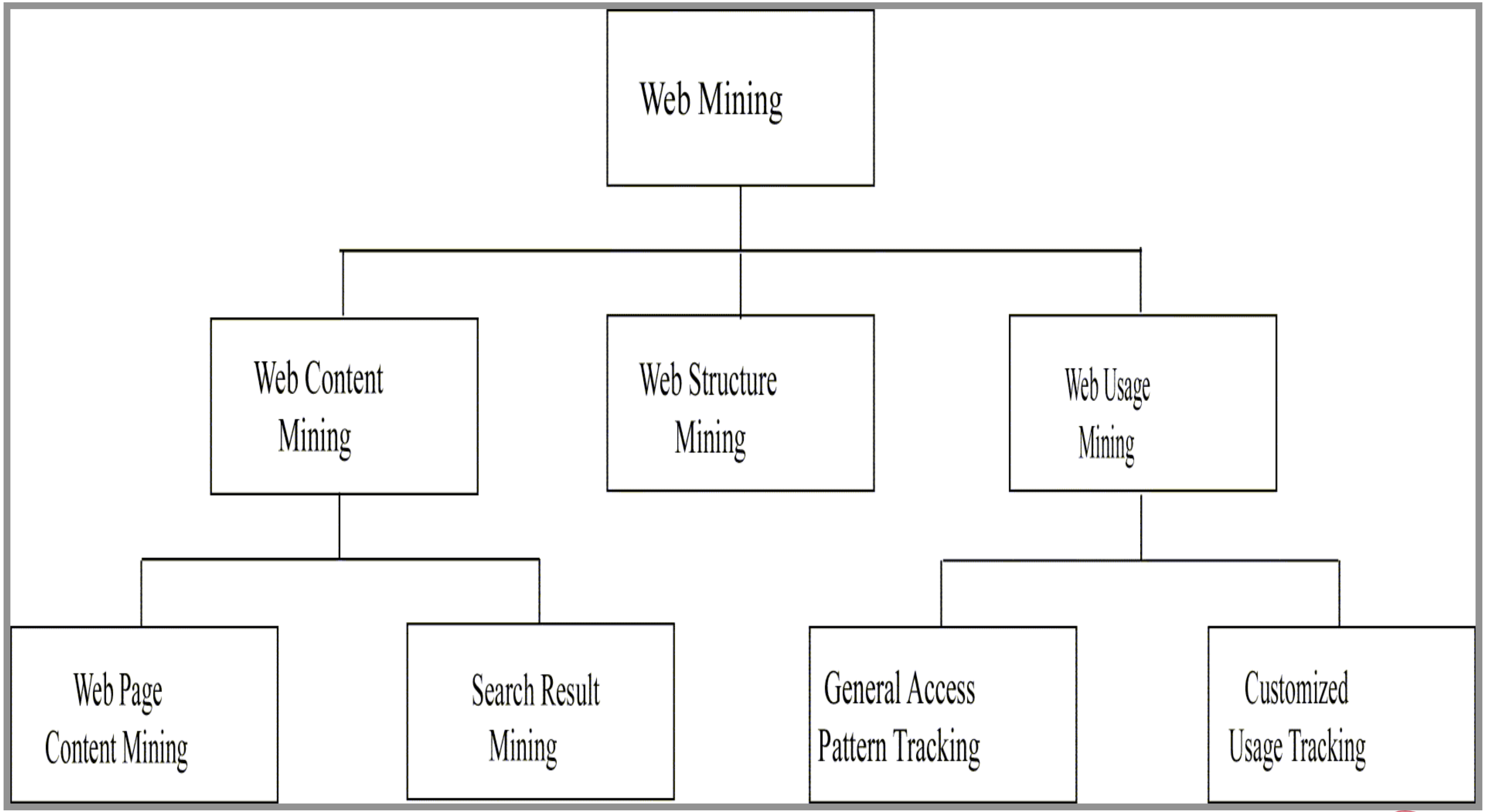
Web Mining Issues

- Size
  - >350 million pages (1999)
  - Grows at about 1 million pages a day
  - Google indexes 3 billion documents
- More recent figures:
  - According to a 2001 study, there were more than 550 billion documents (approximately 7,500 terabytes of data) on the Web, mostly in the "invisible web", or deep web.
  - A study, dated January 2005, queried the Google, MSN, Yahoo!, and Ask Jeeves search engines with search terms from 75 different languages and determined that there were over 11.5 billion web pages in the publicly indexable Web, also termed the the *surface web*.
- Diverse types of data

Web data

- Web pages
- Intra-page structures
- Inter-page structures
- Usage data
- Supplemental data
  - Profiles
  - Registration information
  - Cookies

Web Mining Taxonomy



Modified from [zai01]

Web content mining

- Extends work of basic search engines
- Search engines
  - IR application
  - Crawlers
  - Indexing
  - Profiles
  - Link analysis
- Text mining functions (from basic to advanced)
  - Keyword
  - Term associations
  - Similarity search (between query and document)
  - Classification and clustering
  - Natural language processing

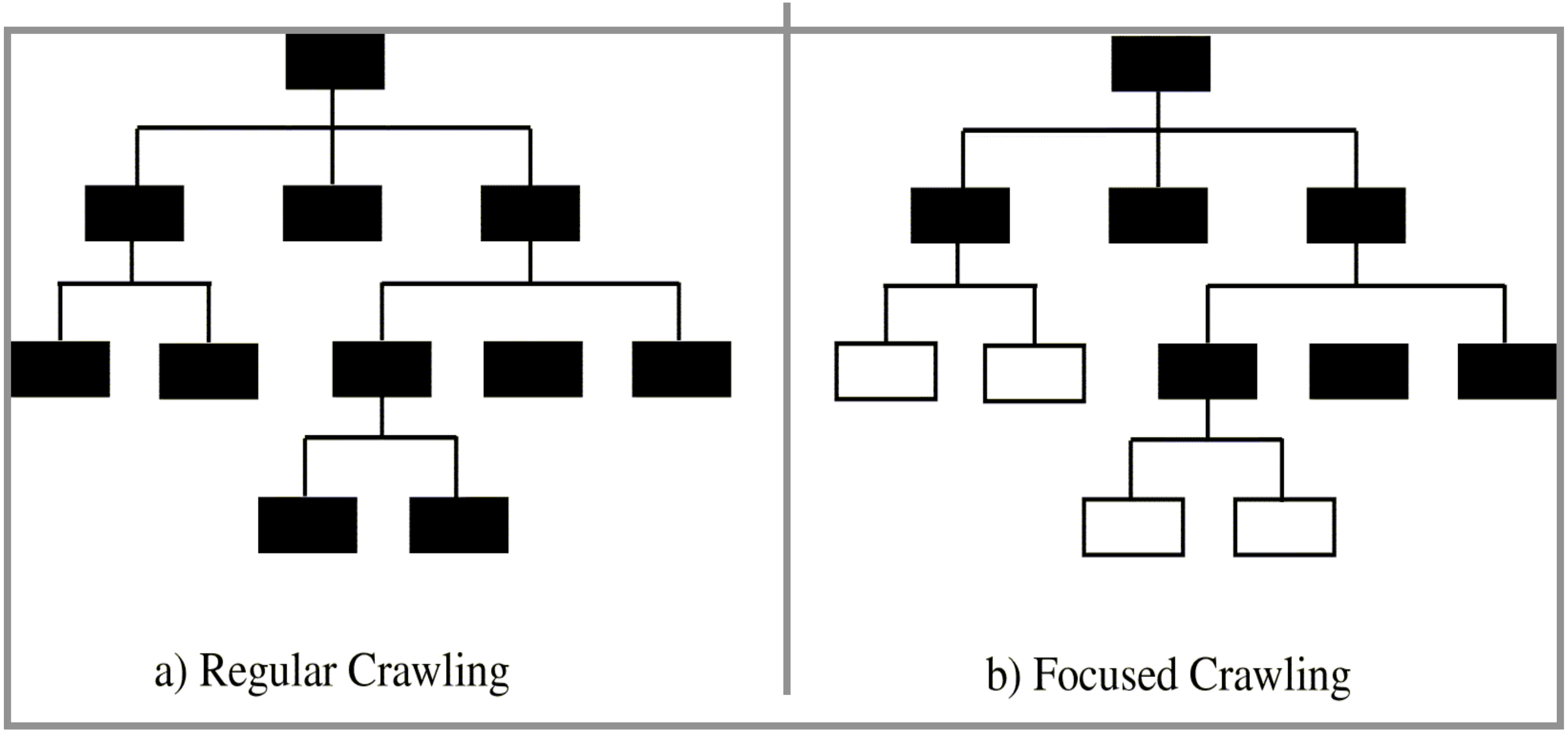
Crawlers

- *Robot (spider)* traverses the hypertext structure in the Web.
  - Collect information from visited pages
  - Used to construct indexes for search engines
- *Traditional crawler* – visits entire Web (?) and replaces index
- *Periodic crawler* – visits portions of the Web and updates subset of index
- *Incremental crawler* – selectively searches the Web and incrementally modifies index
- *Focused crawler* – visits pages related to a particular subject

Focused crawler

- Only visit links from a page if that page is determined to be relevant.
- Classifier is static after learning phase.
- Components:
  - Classifier which assigns relevance score to each page based on crawl topic.
  - Distiller to identify *hub pages.*
  - Crawler visits pages to based on crawler and distiller scores.
- Classifier to related documents to topics
- Classifier also determines how useful outgoing links are
- ***Hub Pages*** contain links to many relevant pages. Must be visited even if not high relevance score.

Focused crawler

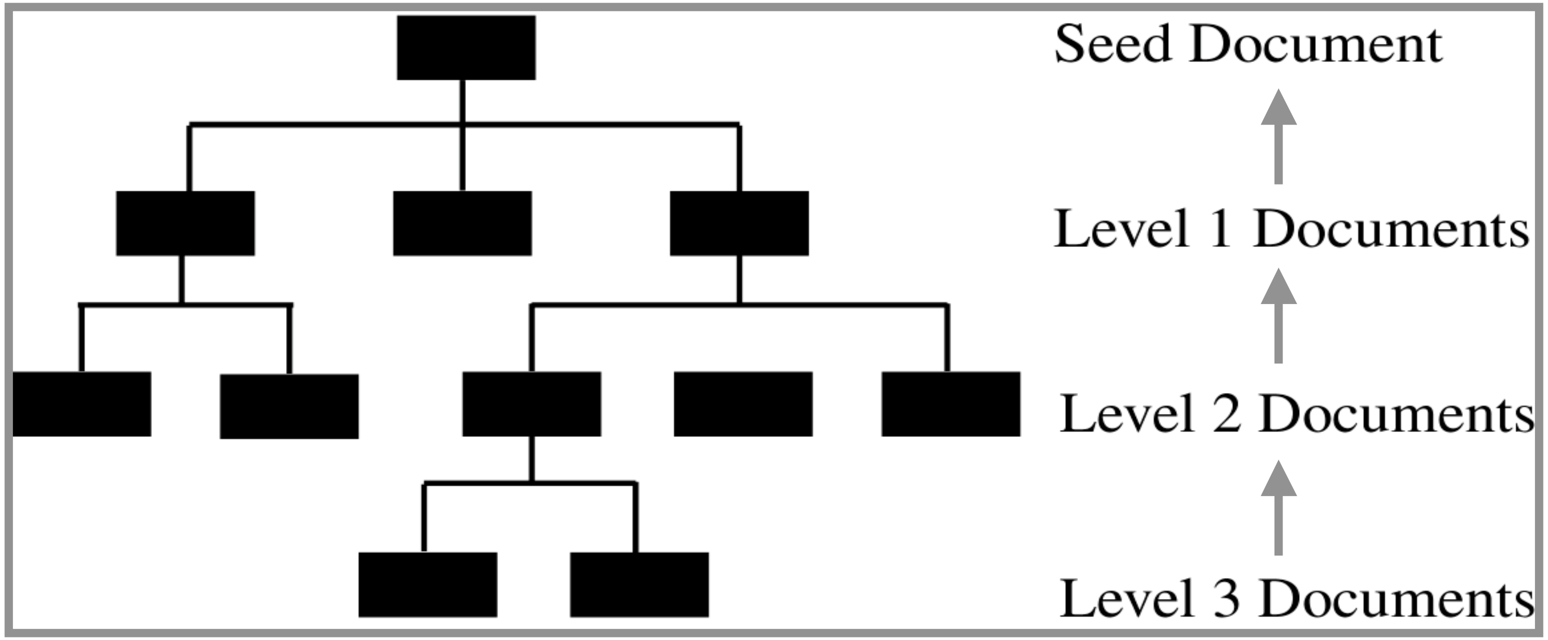


UPPSALA UNIVERSITET

Context focused crawler

- Context Graph:
  - Context graph created for each seed document.
  - Root is the seed document.
  - Nodes at each level show documents with links to documents at next higher level.
  - Updated during crawl itself.
- Approach:
  - Construct context graph and classifiers using seed documents as training data.
  - Perform crawling using classifiers and context graph created.

Context graph



Virtual web view

- *Multiple Layered DataBase (MLDB)* built on top of the Web.
- Each layer of the database is more generalized (and smaller) and centralized than the one beneath it.
- Upper layers of MLDB are structured and can be accessed with SQL type queries.
- Translation tools convert Web documents to XML.
- Extraction tools extract desired information to place in first layer of MLDB.
- Higher levels contain more summarized data obtained through generalizations of the lower levels.

Personalization

- Web access or contents tuned to better fit the desires of each user.
- Manual techniques identify user’s preferences based on profiles or demographics.
- *Collaborative filtering* identifies preferences based on ratings from similar users.
- *Content based filtering* retrieves pages based on similarity between pages and user profiles.

Web structure mining

- Mine structure (links, graph) of the Web
- Techniques
  - PageRank
  - CLEVER
  - HITS
- Create a model of the Web organization.
- May be combined with content mining to more effectively retrieve important pages.

PageRank

- Used by Google
- Prioritize pages returned from search by looking at Web structure.
- Importance of page is calculated based on number of pages which point to it – ***Backlinks***.
- Weighting is used to provide more importance to backlinks coming form important pages.

- $PR(p) = c\ (PR(1)/N_1 + \ldots + PR(n)/N_n)$
  - PR(i): PageRank for a page i which points to target page p.
  - N_i: number of links coming out of page i
  - c: is a value between 0 and 1 used for normalization

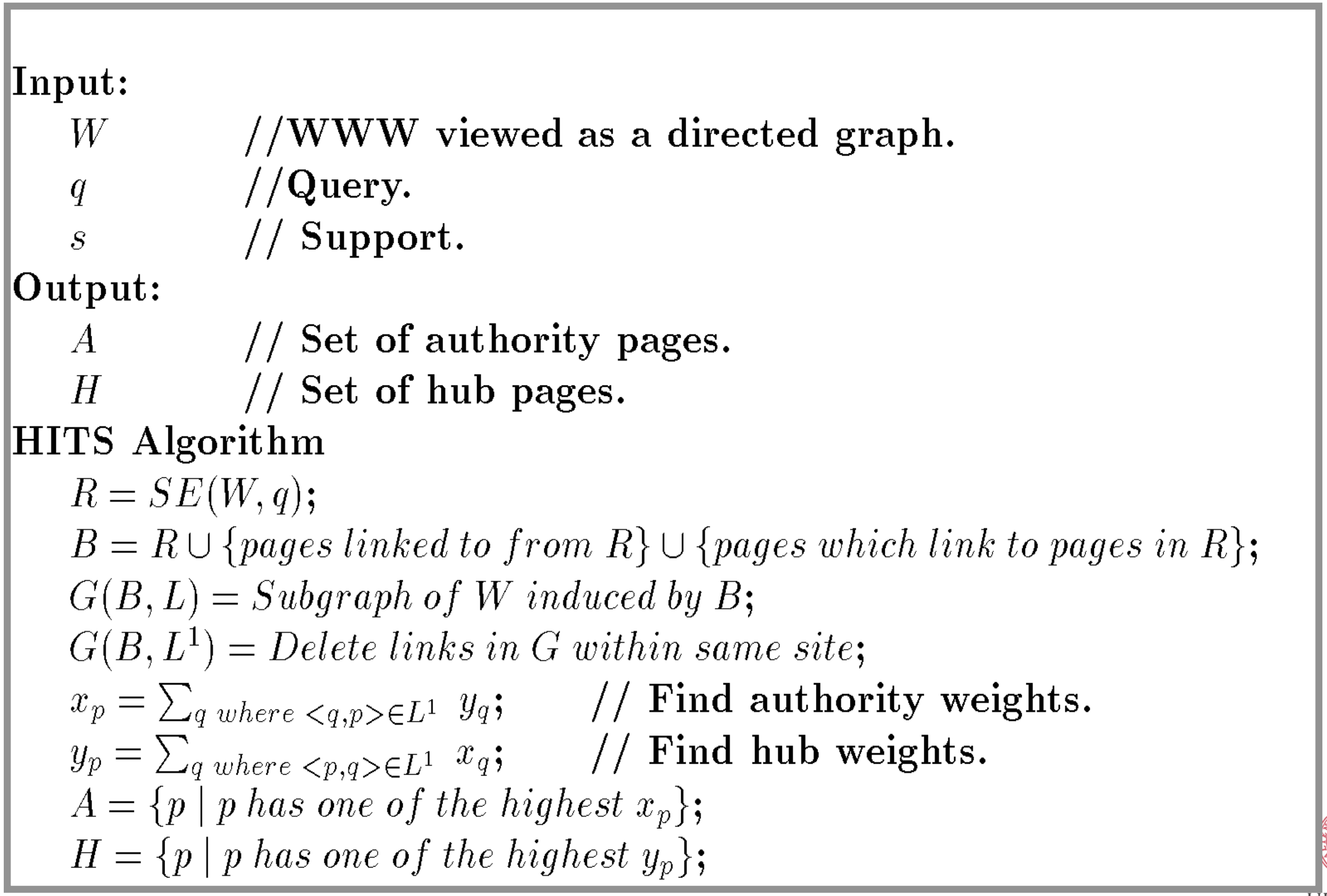
CLEVER

- Identify authoritative and hub pages.
- ***Authoritative Pages*** :
  - Highly important pages.
  - Best source for requested information.
- ***Hub Pages*** :
  - Contain links to highly important pages.

HITS

- Hyperlink-Induced Topic Search
- Based on a set of keywords, find set of relevant pages – R.
- Identify hub and authority pages for these.
  - Expand R to a base set, B, of pages linked to or from R.
  - Calculate weights for authorities and hubs.
- Pages with highest ranks in R are returned.

HITS algorithm



```
Input:
    $W$        //WWW viewed as a directed graph.
    $q$        //Query.
    $s$        // Support.
Output:
    $A$        // Set of authority pages.
    $H$        // Set of hub pages.
HITS Algorithm
    $R = SE(W, q)$;
    $B = R \cup \{pages\ linked\ to\ from\ R\} \cup \{pages\ which\ link\ to\ pages\ in\ R\}$;
    $G(B, L) = Subgraph\ of\ W\ induced\ by\ B$;
    $G(B, L^1) = Delete\ links\ in\ G\ within\ same\ site$;
    $x_p = \sum_{q\ where\ <q,p> \in L^1} y_q$;        // Find authority weights.
    $y_p = \sum_{q\ where\ <p,q> \in L^1} x_q$;        // Find hub weights.
    $A = \{p \mid p\ has\ one\ of\ the\ highest\ x_p\}$;
    $H = \{p \mid p\ has\ one\ of\ the\ highest\ y_p\}$;
```

Web usage mining

- Performs mining on web usage data or web logs (clickstreams)
  - Examined both from a server …
    - Uncover info about site where service reside
    - Can e.g. improve design
  - ... and a client perspective
    - Uncovers info about user or group
    - Can e.g. improve prefeching and caching
- Applications of web usage mining
  - Personalization
  - Improve structure of a site's Web pages
  - Aid in caching and prediction of future page references
  - Improve design of individual pages
  - Improve effectiveness of e-commerce (sales and advertising)

Web usage mining activities

- Preprocessing Web log
  - Cleanse
  - Remove extraneous information
  - Sessionize

    Session: Sequence of pages referenced by one user at a sitting.
- Pattern Discovery
  - Count patterns that occur in sessions
  - *Pattern* is sequence of pages references in session.
  - Similar to association rules
    - Transaction: session
    - Itemset: pattern (or subset)
    - Order is important
- Pattern Analysis

Association analysis in web mining

- Web Mining:
  - Content
  - Structure
  - Usage
- Frequent patterns of sequential page references in Web searching.
- Uses:
  - Caching
  - Clustering users
  - Develop user profiles
  - Identify important pages

Web usage mining issues

- Identification of exact user not possible.
- Exact sequence of pages referenced by a user not possible due to caching.
- Session not well defined
- Security, privacy, and legal issues

Web log cleansing

- Replace source IP address with unique but non-identifying ID.
- Replace exact URL of pages referenced with unique but non-identifying ID.
- Delete error records and records containing not page data (such as figures and code)

Sessionizing

- Divide Web log into sessions.
- Two common techniques:
  - Number of consecutive page references from a source IP address occurring within a predefined time interval, such as 30 min (empirical studies show 25,5 min).
  - All consecutive page references from a source IP address where the interclick time is less than a predefined threshold.

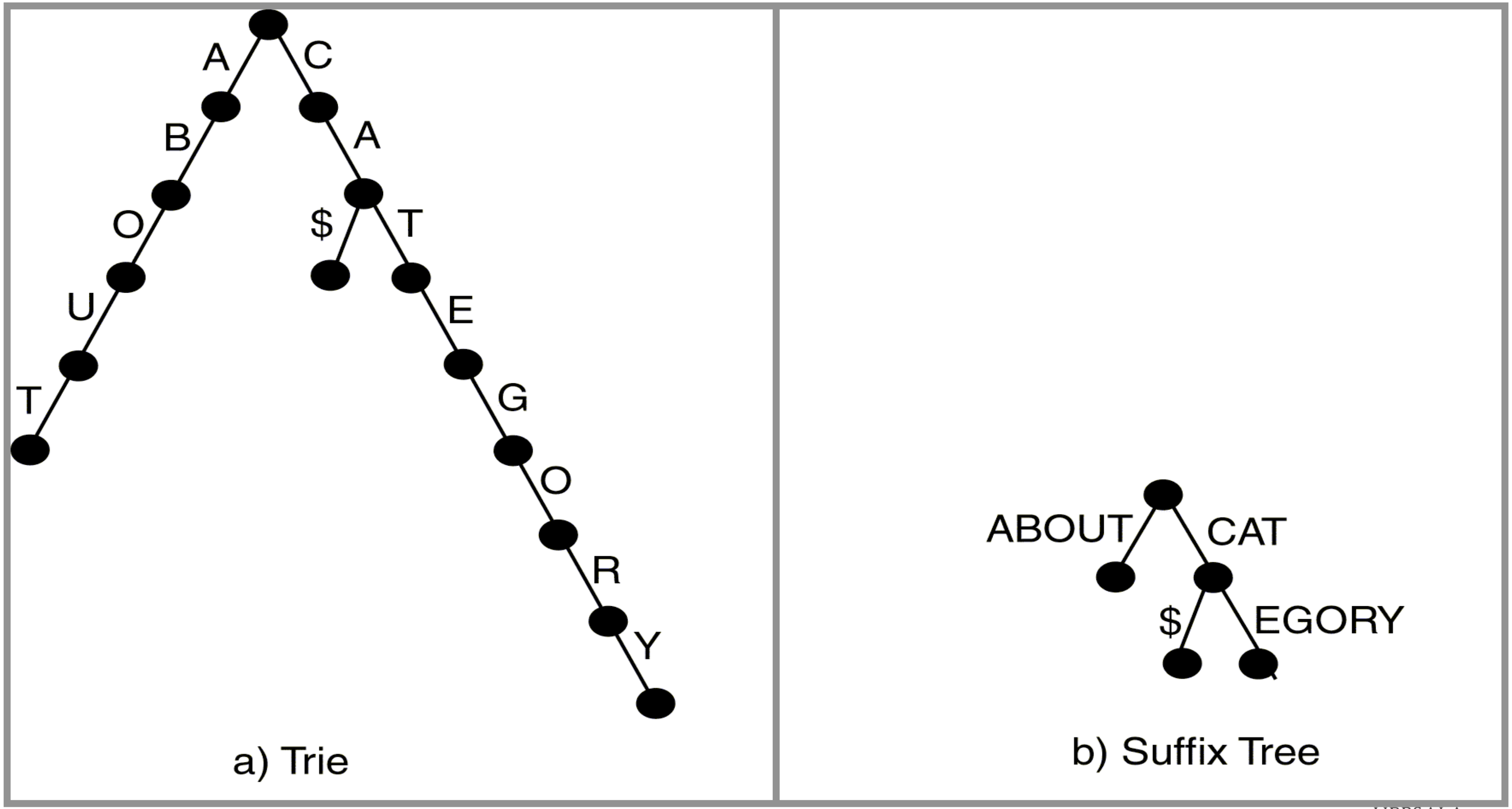
Data structures

- Keep track of patterns identified during Web usage mining process
- Common techniques:
  - Trie
  - Suffix Tree
  - Generalized Suffix Tree
  - WAP Tree

Trie vs. Suffix tree

- ***Trie:***
  - Rooted tree
  - Edges labeled which character (page) from pattern
  - Path from root to leaf represents pattern.
- ***Suffix tree:***
  - Single child collapsed with parent. Edge contains labels of both prior edges.

Trie and Suffix tree



a) Trie

b) Suffix Tree

Generalized Suffix tree

- Suffix tree for multiple sessions.
- Contains patterns from all sessions.
- Maintains count of frequency of occurrence of a pattern in the node.
- ***WAP Tree:***

  Compressed version of generalized suffix tree

Types of patterns

- Algorithms have been developed to discover different types of patterns.
- Properties:
  - *Ordered* – Pages (characters) must occur in the exact order in the original session.
  - *Duplicates* – Duplicate pages are allowed in the pattern.
  - *Consecutive* – All pages in pattern must occur consecutive in given session.
  - *Maximal* – Not subsequence of another pattern.

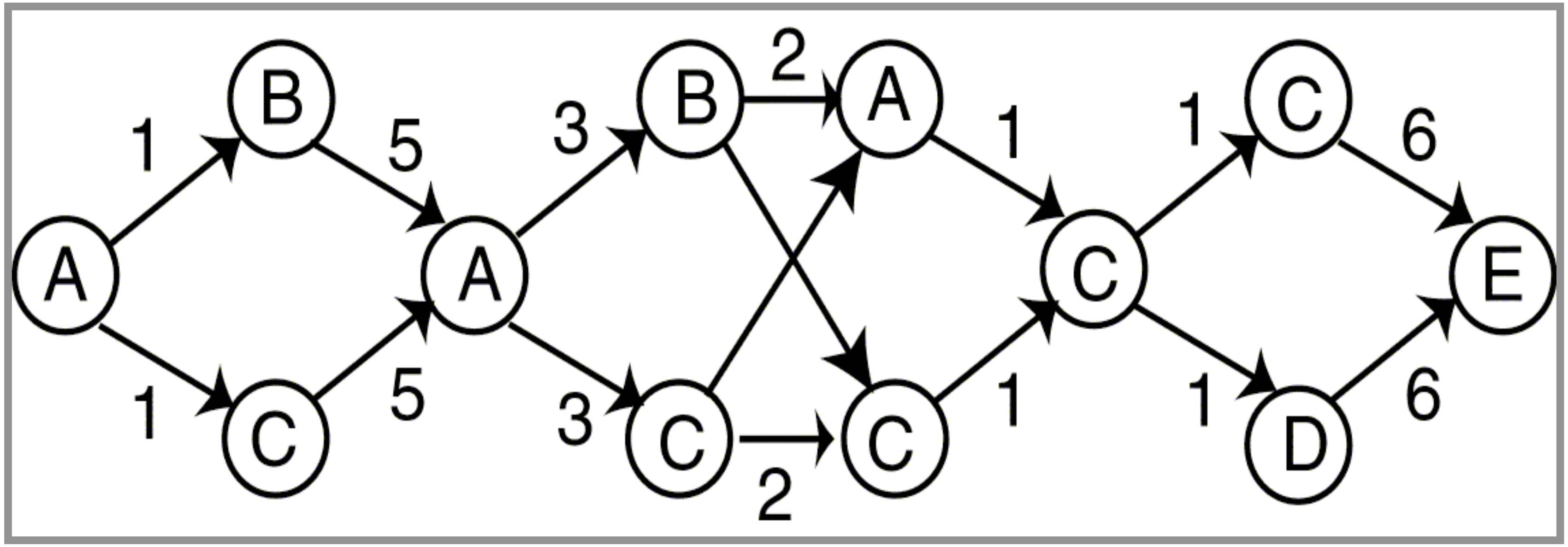
Pattern types

- Association rules
  - None of the properties hold (no order, no duplicates, no consecutive or maximal patterns)
- Episodes
  - Only ordering holds
- Sequential patterns
  - Ordered and maximal
- Forward sequences
  - Backlinks and reloads eliminated
  - Ordered, consecutive, and maximal
- Maximal frequent sequences
  - Support calculated in reference to length of sequence, i.e. no of clicks
  - All properties hold

Episodes

- Partially ordered set of pages
- *Serial episode* – totally ordered with time constraint
- *Parallel episode* – partial ordered with time constraint
- *General episode* – partial ordered with no time constraint

DAG for Episode


A
1
B
1
C
5
5
A
3
B
3
C
2
A
2
C
1
1
C
1
C
1
D
6
6
E