Deriving Dynamics of Web Pages: A Survey

Marilena Oita[1,2] and Pierre Senellart[1]

[1] Telecom ParisTech, INFRES DbWeb team

[2] Webdam project, INRIA Saclay Ile-de-France

TWAW, Hyderabad, India
28th of March, 2011

Outline

Web's dynamics
evolution which implies ephemerality

the Web is dynamic by nature
$\longrightarrow$ difficulty to:

- keep track of the updated information
- capture relevant changes

the strategy of crawl must be adapted to the change frequency of the Web page (=its dynamics)

OBS: dynamics varies in time,
so is usually very difficult to determine its patterns
without a deeper knowledge of the Web page/site in cause

Use case: Incremental Crawl
problematique

- as opposed to snapshot crawl, it actively crawls only *changed parts* of a Web page
- **Q:** how often the Web page shall be crawled in order not to miss changes?

Crawler's difficulty: the URL doesn't change, but the page itself does!
$\rightarrow$ new versions of the same object referenced through the URL

Usually, the temporal properties of Web pages are empirically inferred:
the frequency of change $=$ the mean of the intervals between the update timestamps

Change types
related to the Web page's characteristics

Changes in Web documents can occur at different levels:

- content: changes in the textual data
- structure: related to the hierarchical model of a Web page
- presentation: the way of presenting the information (visually)
- behaviour: in HTML active elements

Detecting change

and derive temporal properties from Web pages

is needed in many application domains:

- Web crawling
  1. versioning
  2. adjusting the refresh rate
  3. maintaining the temporal coherence of linked pages
- information monitoring systems
- Web caching improving
- servicing of continuous queries
- data mining

Detecting change

and derive temporal properties from Web pages

is needed in many application domains:

- Web crawling
  1. versioning
  2. adjusting the refresh rate
  3. maintaining the temporal coherence of linked pages
- information monitoring systems
- Web caching improving
- servicing of continuous queries
- data mining etc...

Techniques
for deriving temporal properties from Web pages

Different approaches:

1. static: timestamps can be identified in content or dynamics is directly derived from linked (meta)data files
2. dynamic: change reveals itself in the active process of comparing versions
3. estimative: an estimation is produced, based on an initial change history and a statistical model

Timestamping

that operates on the Web page itself

1. check HTTP timestamp
   - ETag: "497bef-1fcb-47f20645"
   - Last-Modified: Tue, 01 Apr 2008 09:54:13 GMT
   - Cache-Control: max-age=60, private
   - Expires: Tue, 01 Apr 2008 13:25:55 GMT
1. check for timestamp in content:
   - keywords that denote time
   - date recognition

Timestamping
that operates on the Web page's references

- using the linked data: the neighborhood
- use documents that contain temporal metadata and *refer* to a Web page / site

Timestamping
that operates on the Web page's references

- using the linked data: the neighborhood
- use documents that contain temporal metadata and *refer* to a Web page / site
  (reliable, but not always available)

1. RSS feeds: *pubDate*, *lastBuildDate*, *ttl*
2. Sitemaps: *lastmod*, *changefreq* (for a given URL)

Dynamic: actively compare versions

to detect change

In the comparison process, we must:

1. have versions to compare
2. define a model of the Web page
3. specify similarity metrics between model elements

Dynamic approaches:

- suppose versions provided most of the time
- it would set the timestamp of a new version as the date when change was detected

Techniques of assessing dynamics
Models

From the Web page modeling point of view, we have:

1. flat-file string : structure and code properties are ignored in the process
2. DOM tree

   for a tree entity (node/subtree/branch) the data structure usually contains:
   - an entity's id
   - child-parent relationship
   - tag name
   - content
   - the level (=*depth*) of an entity in the tree

Techniques of assessing dynamics

Models

3. weighted bipartite graph
   - from a unordered tree model, after pruning, a set of nodes remains
   - nodes are linked through weighted edges
   - the weight represents the edit scripting cost of transforming an entity into another
   - usually Hungarian algorithm is applied
4. Page Digest encoding
   - clear separation between content and structure
   - high efficiency because of faster parsing

Pre-processing steps in hierarchical models
trees

Having 2 versions of a Web page, do:

1. transform each from HTML to XML: parsing HTML tag soup into a clean tree structure using XSLT/XPath
2. filtering of irrelevant tag elements (for instance, scripts)
3. apply similarity metrics between the model entities of the two versions
4. pruning: eliminate identical (or too dissimilar) elements
5. apply the actual technique(algorithm) for change detection

Similarity metrics
used in change detection

String matching techniques:

1. Jaccard-based
2. hash-based (signatures, shingling)
3. longest common subsequence: diff algorithms (*HTMLDiff*)
4. root mean square (RMS) of the string's ASCII codes

Similarity metrics
used in change detection

Matching in hierarchical models:

1. edit scripting: *MH-Diff*, *SCD*
2. trivalent quantitative formula for change: *CMW*

Statistical methods

change as a random event that can be forsee based on the history

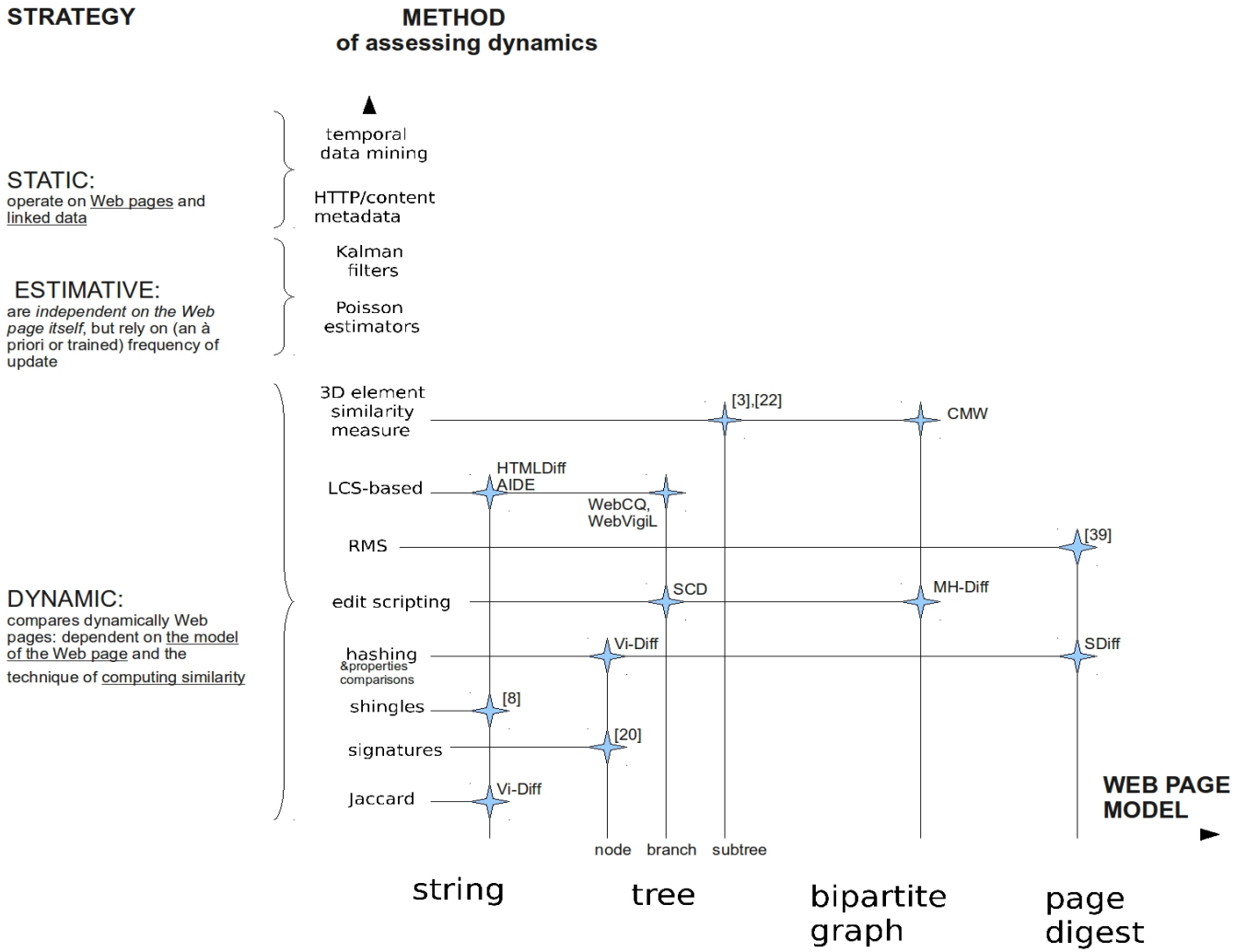
Having an observed change history (a set of versions), estimative models predict the next date of change.

Models:

1. Poisson process: model random events, that occur independently (homogenuous Poisson)
2. Kalman filters: recursive estimator that gives the internal state of a linear dynamic system, from a series of measurements (timestamps, in this case!)

Summary of the presented approaches

resuming



Discussion
and personal vision

1. further studies on static approaches to timestamping
2. relevant change detection: a challenge because
   we need to define what is important and a measure of it:
   the role of semantics and the standardization of time aspects in protocols

Thank You!

Questions?

References

1. **MH-Diff**: Chawathe and Garcia-Molina. Meaningful change detection in structured data
2. **Poisson**: Cho and Garcia-Molina. Estimating frequency of change
3. **CMW**: Flesca and Masciari. Efficient and effective Web page change detection
4. **SCD**: Lim and Ng. An automated change-detection algorithm for HTML documents based on semantic hierarchies
5. **on relevance**: Oita and Senellart. Archiving data objects using web feeds
6. **Page Digest**: Rocco, Buttler and L. Liu. Page digest for large-scale Web services