LiWA
Living Web Archives

Databases and
Information Systems
Prof. Dr. G. Weikum

Data Quality in Web Archiving

Marc Spaniol

Saarbrücken, June 4, 2009

LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

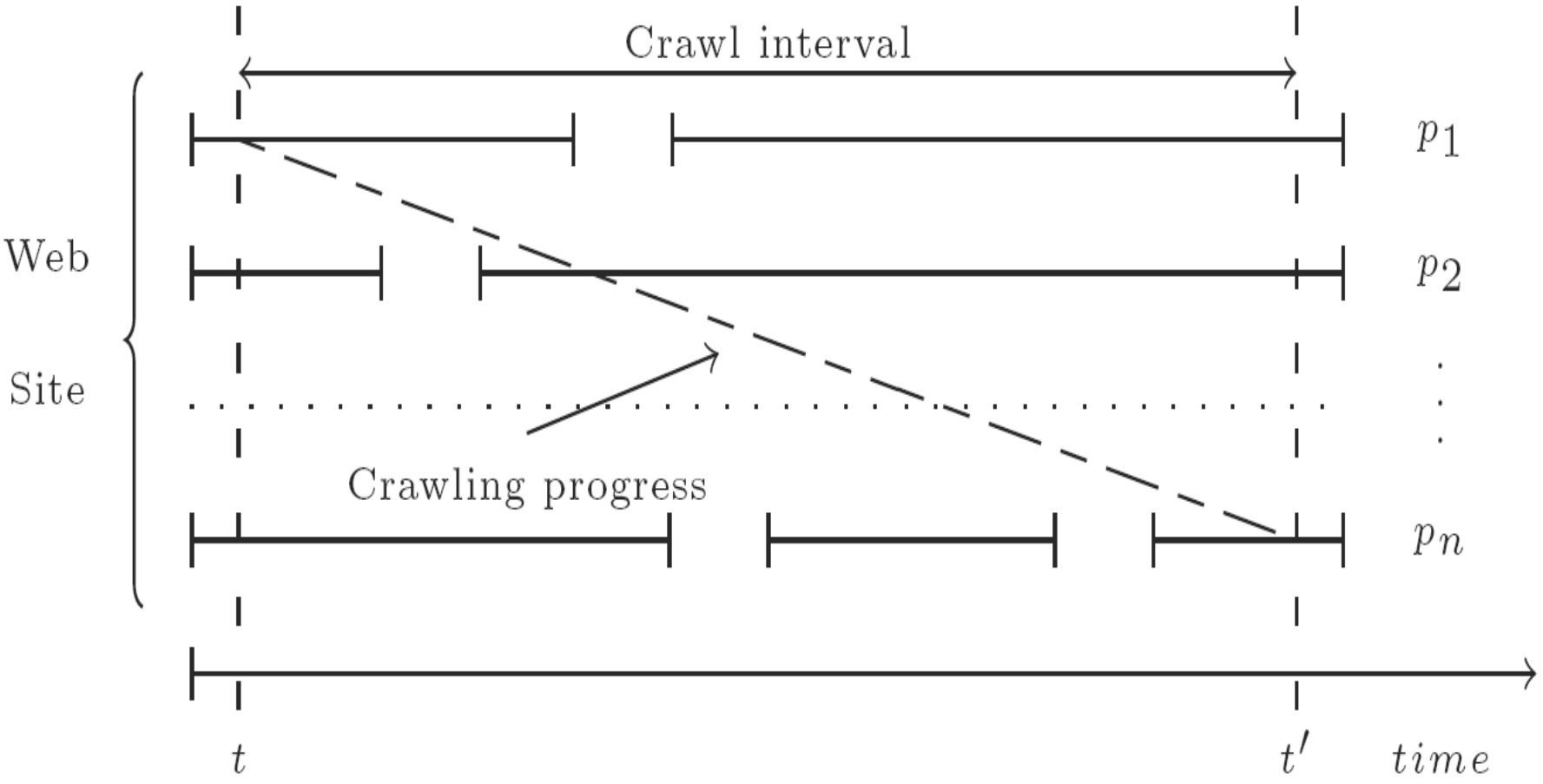
mpii

Databases and Information Systems
Prof. Dr. G. Weikum

Agenda

- Motivation and problem statement
- Document similarity
  - Shingling
  - Minhashing
  - Locality-sensitive hashing
- Coherence framework
  - Coherence
  - Observable coherence
  - Measurable coherence
  - Inducible coherence
- Coherence improved capturing
  - Probability of (in-)coherence
  - Crawl scheduling
  - Experimental results
- Incoherence detection
- Summary

Motivation



LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

The Challenge of Archive Coherence

- Crawler operations
  - Visit (pages)
    - Extract (links from pages)
    - Compare (versions of pages)
  - Follow (links)

- Website operations
  - Modifications “inside” pages
    - Content (text)
    - Structure (links)
  - Modifications “inside” site
    - Page creation
    - Page deletion

Taking place in parallel

⇓

Potentially incoherent

Databases and Information Systems
Prof. Dr. G. Weikum

LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

mpii

Databases and Information Systems
Prof. Dr. G. Weikum

Potential Pitfalls in Web Archiving

- Crawling takes a long (!) time
  - Politeness
  - Multiple seeds per crawl
  - Spam
- Crawlers aren't "really" smart
  - Highly volatile against dynamics in CMS
  - Easy to be trapped, if not exactly configured
  - Doesn't recognize patterns of "identical" contents

  ⇒ Pre-analysis of site(s) needed
- Some examples of crawler behavior
  - Enjoy link generation from JavaScript, PHP, etc.
  - Tend to go for shopping
  - Like time travelling in calendars

⇒ Crawling is simply "unpredictable"
⇒ Crawlers need "constant" monitoring

⇒ Archive in Danger!

Smart(er) Crawling Strategies

Evaluation of Crawl Coherence

LiWA
Living Web Archives

Marc Spaniol

Problem Statement

- What means coherence?
  - "The action or fact of cleaving or sticking together"
  - "Harmonious connexion of the several parts, so that the whole 'hangs together'"

Oxford English Dictionary [http://dictionary.oed.com]

- Temporal coherence in Web archiving:
  - Capturing Web sites as "authentic" as possible
  - Ensure an "as of time point x (or interval [x, y])" capture of a Web site

⇒ Periodic domain scope crawls of Web sites to obtain a best possible representation with respect to a time point / interval

LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

Databases and Information Systems
Prof. Dr. G. Weikum

Measuring Crawl Coherence

- Online vs. offline
- Online coherence
  - "Memoryless"
  - Measures change at runtime
    - Evaluate coherence between crawl/revisit tuples
    - Evaluate coherence relative to begin of crawl
  - Identifies potential incoherence already at time of crawl
  - Covers a (minimum) interval / time-point of coherence
- Offline coherence
  - Utilizes existing domain knowledge
  - Measures change from the archive
  - Evaluates coherence between (any) captured tuples
  - May incorporate more sophisticated machine learning strategies
  - Helps to predict media specific change rates

LiWA
Living Web Archives

Marc Spaniol

Incoherence Categories

- Removed contents
- Structural (changed link structure)
  - Links to new contents added
  - Links to removed contents deleted
- Content wise
  - Major changes: Text added or deleted (in “large” sections)
  - Minor changes: Text changed (in “small” sections)

$\Rightarrow$ Comparison of document similarities (syntactically not semantically)

LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

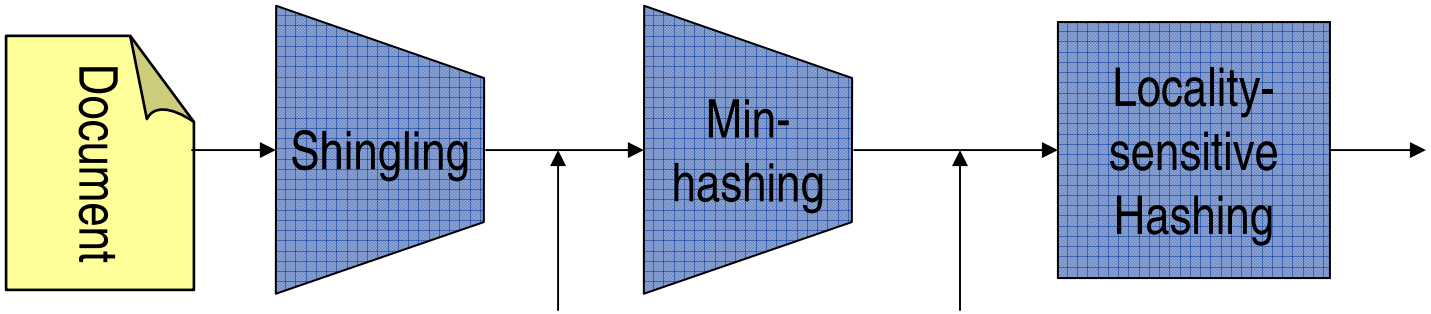
Document Similarity

- In general:
  - Given a body of documents, e.g., the Web
  - Find pairs of docs that have a lot of text in common
  - ⇒ Identify mirror sites, approximate mirrors, plagiarism, quotation of one document in another, “good” document with random spam, etc.

- In the case of data quality in Web archiving:
  - Characterize change (diff) between two versions of page
  - Identify relevant aspects of changes to web pages and sites
    - Content: full, – banners, – links, – photos, – style, etc.
    - Links: all, non-navigational, intra-site, etc.
  - Quantify the amount of changes
  - ⇒ Filtering of irrelevant changes

Marc Spaniol

Basic Strategy



Candidate pairs: Those pairs of signatures that need to be tested for similarity

The set of strings of length *k* that appear in the document

Signatures: Short integer vectors that represent the sets, and reflect their similarity

LiWA
Living Web Archives

Marc Spaniol

Shingles – k-grams

- Representation of a document by its set of shingles (or k-grams)
  - Documents that have lots of shingles in common have similar text
  - The text may even appear in different order

$\Rightarrow$ Similar documents are very likely to have many shingles in common

- Selection of k having a “useful” size is crucial:
  - If k is too small, documents might have too many shingles in common
  - If k is too large, compression is not good

- Heuristic experience:
  - k = 5 is OK for short documents
  - k = 10 is better for long documents

LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

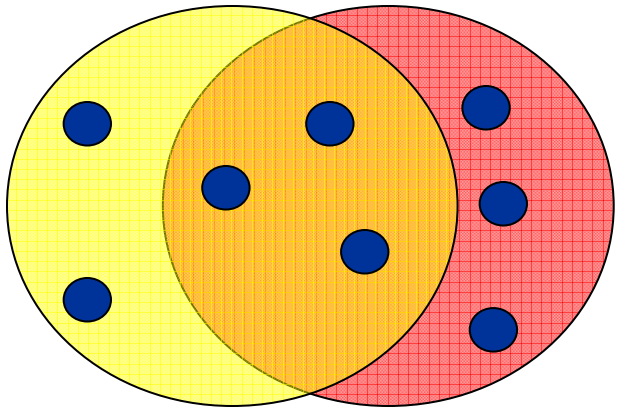
Databases and Information Systems
Prof. Dr. G. Weikum

Basic Data Model: Sets

- Many similarity problems can be expressed as finding subsets of some universal set that have significant intersection
- A k-shingle (or k-gram) of a document is a sequence of k characters that appears in the document
- Documents are represented by their k-shingles
  - All possible k-shingles represent universe *U*
  - Degree of “overlap” between shingle sets represents similarity of documents
- Example:
  - Document = abcab
  - $k = 2$
  - Set of 2-shingles = {ab, bc, ca}.
- Option: Regard shingles as a bag → count “ab” twice

Jaccard Similarity of Sets

- The Jaccard similarity of two sets C_1 and C_2:
  Size of their intersection divided by the size of their union:
  $Sim(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$



3 elements in intersection
8 elements in union

$\Rightarrow$ Jaccard similarity = 3/8

LiWA
Living Web Archives

Marc Spaniol

From Sets to Boolean Matrices

- Matrix representation of data in the form of subsets of a universal set
  - Rows = elements of the universal set (shingles)
  - Columns = sets (documents)
  - 1 in row *r*, column *c* iff document *c* contains shingle *r*
  - Column similarity is the Jaccard similarity of the sets of their rows with 1

$\Rightarrow$ Typically the matrix is sparse

- Implementation
  - Might not really represent the data by a boolean matrix
  - List of places where there is a non-zero value.

$\Rightarrow$ Matrix illustration is conceptually useful

Databases and Information Systems
Prof. Dr. G. Weikum

LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

Databases and Information Systems
Prof. Dr. G. Weikum

Row Types

- Given columns C_1 and C_2, rows may be classified as:

C_1	C_2		
1	1	←	a
1	0	←	b
0	1	←	c
0	0	←	d

- a = # rows of type a , etc.
- Note: $Sim(C_1, C_2) = a / (a + b + c)$

Example

C_1	C_2		
0	1	←	
1	0	←	
1	1	←	✓
0	0		
1	1	←	✓
0	1	←	

$Sim(C_1, C_2) = 2/5 = 0.4$

LiWA
Living Web Archives

Marc Spaniol

Problems

- Computational complexity
  - Creation of shingles
  - Comparison of columns (often pair wise)
- "Compression" of columns wanted, so that
  - Similar documents obtain related signatures
  - Dissimilar documents receive discriminative signatures
- ~~Idea:~~
  - ~~Pick m ($m << k$) rows at random~~
  - ~~Let the signature of column C be the m bits of C in those rows~~
  - ~~$\Rightarrow$ Matrix is sparse~~
  - ~~$\Rightarrow$ Many columns will have 00. . .0 as a signature~~
  - ~~$\Rightarrow$ "Everything" is dissimilar because their 1's are in different rows~~

Minhashing

- Key idea: "Hash" each column C_i to a small *signature* S_i
- Basic goals:
  - S_i is sufficiently small (e.g. to be processed in the main memory)
  - $Sim(C_1, C_2)$ corresponds to $Sim(S_1, S_2)$
- Basic idea:
  - Imagine the rows permuted randomly and equally distributed
  - Define hash function $h(C)$ to compute smallest number of the (in the permuted order) row in which column C has 1
  - Use several ($m << k$) independent hash functions to create a signature
  - Optional: Check that columns with similar signatures are really similar

Minhashing Example

Input matrix → Signature matrix *M*

h_1	h_2	h_3	C_1	C_2	C_3	C_4
1	4	3	1	0	1	0
3	2	4	1	0	0	1
7	1	7	0	1	0	1
6	3	6	0	1	0	1
2	6	1	0	1	0	1
5	7	2	1	0	1	0
4	5	5	1	0	1	0

S_1	S_2	S_3	S_4
2	1	2	1
2	1	4	1
1	2	1	2

Similarities:

	$C_1 C_3$	$C_2 C_4$	$C_1 C_2$	$C_3 C_4$
Col/Col	0.75	0.75	0	0
Sig/Sig	0.67	1.00	0	0

LiWA
Living Web Archives

Challenges

- In general:
  - Suppose huge documents (e.g., 1 billion rows)
  - Hard to pick a random permutation from 1…billion
  - Representing a random permutation requires 1 billion entries
  - Accessing rows in permuted order leads to thrashing

  $\Rightarrow$ A good approximation to permuting rows: Pick $m << k$ hash functions

- In the case of data quality in Web archiving:
  - Document size less problematic
  - Usually only few pair wise comparisons needed
  - Random contents or automatically generated contents in CMS ruin all efforts

  $\Rightarrow$ Data scrubbing is crucial

$\Rightarrow$ Defining “good” hash functions are an own research topic!

LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

Implementation

- For each column c and each hash function h_i, keep a "slot" $M(i, c)$
- $M(i, c)$ becomes the smallest value of $h_i(r)$ having 1 in column c at row r
- $h_i(r)$ gives order of rows for i th permuation

for each row r
  for each column c
    if c has *1* in row r
      for each hash function h_i **do**
        if $h_i(r)$ is a smaller value than $M(i, c)$ **then**
          $M(i, c) := h_i(r)$;

- Optimization:
  - Our case: columns = documents, rows = shingles
  - Sort matrix once so it is by row
  - Compute $h_i(r)$ only once for each row

Example

Row	C_1	C_2
1	1	0
2	1	1
3	0	1
4	1	1
5	0	1

$h(x) = x \bmod 5$

$g(x) = 2x+1 \bmod 5$

	S_1	S_2
$h(1) = 1$	1	-
$g(1) = 3$	3	-
$h(2) = 2$	1	2
$g(2) = 0$	0	0
$h(3) = 3$	1	2
$g(3) = 2$	0	0
$h(4) = 4$	1	2
$g(4) = 4$	0	0
$h(5) = 0$	1	0
$g(5) = 1$	0	0

LiWA
Living Web Archives

Marc Spaniol

Challenges

- In general:
  - Data represents a large number of objects in main memory
    - Document shingles
    - Signatures after minhashing
  - Only few pairs (*candidate pairs*) are sufficiently similar to be compared
  - Comparison of signatures of all pairs of columns is very (!) expensive

  $\Rightarrow 10^6$ columns: $5*10^{11}$ column-comparisons (quadratic in the number of columns)

- In the case of data quality in Web archiving:
  - Coherence evaluation wrt to non-relevant changes
    - *Candidate pairs c*, *d* are given beforehand
    - Similarity comparison of signatures according to predefined threshold
  - Long term coherence studies are more computational intense

  $\Rightarrow$ *Candidate pair* detection more important for Web spam or duplicate detection

$\Rightarrow$ Locality-sensitive hashing

Databases and Information Systems
Prof. Dr. G. Weikum

LiWA
Living Web Archives

Marc Spaniol

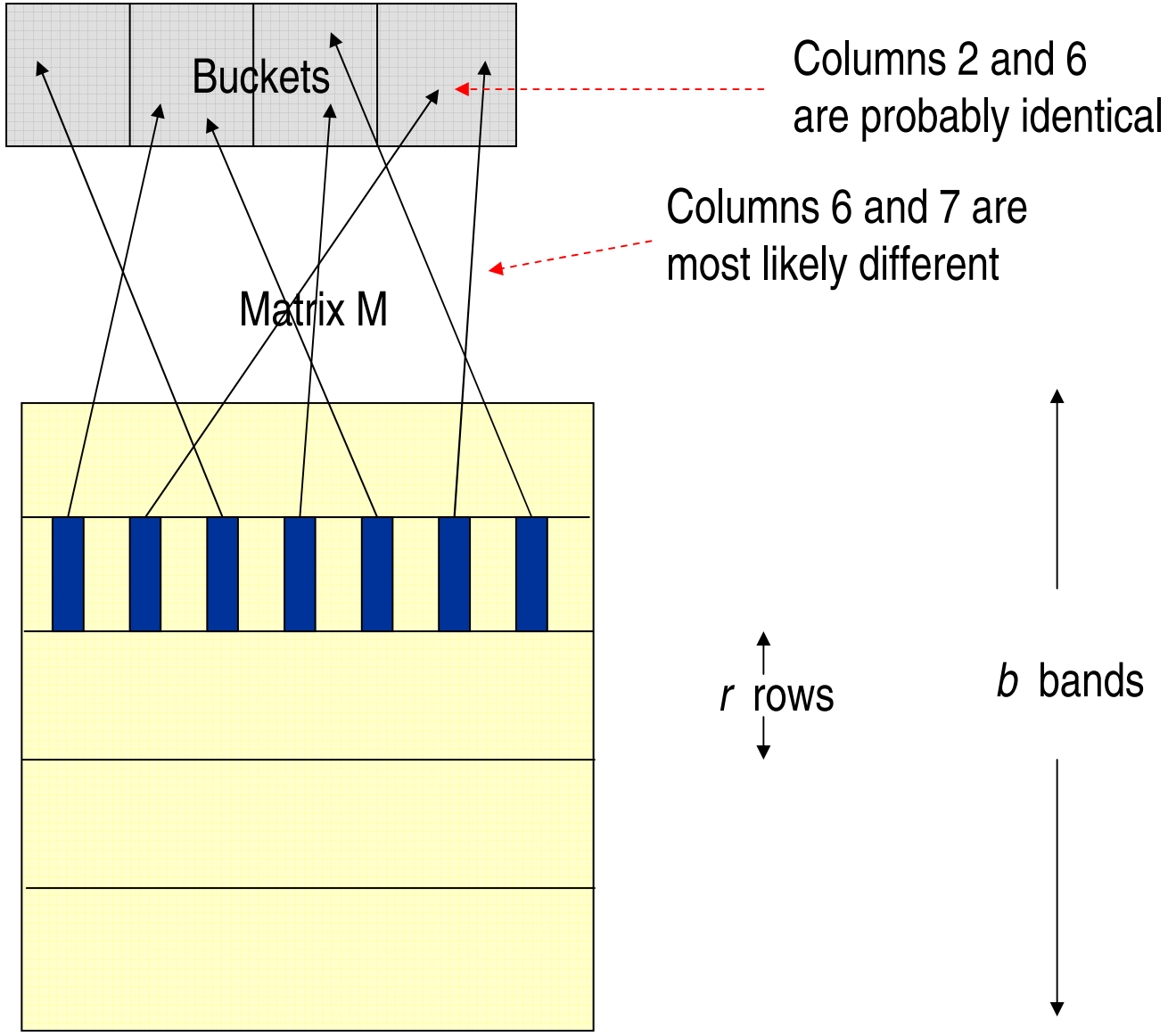
Locality-Sensitive Hashing

- Basic idea
  - Function f(x,y) tells whether or not *x* and *y* is a *candidate pair*
  - *Candidate pairs* of elements whose similarity must be evaluated

- Process (simplified)
  - Divide matrix *M* into *b* bands of *r* rows
  - For each band, hash its portion of each column to a hash table with *k* buckets
    - Make *k* as large as possible
  - Candidate column pairs are those that hash to the same bucket for ≥ 1 band
  - Tune *b* and *r* to catch most similar pairs, but few non-similar pairs

  ⇒ Similarity comparison of signatures of candidate pairs only

Example


Buckets
Columns 2 and 6 are probably identical
Columns 6 and 7 are most likely different
Matrix M
r rows
b bands

LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

Coherence Framework

- Basic Assumptions
  - Web site to be crawled consists of *n* Web pages
  - Changes of Web pages occur per time unit and independent of each other
  - Change rates are assumed / given
  - Delay between downloads of pages is the same
  - Download time is neglected

- Basic Notation
  - Crawl: c
  - Web pages: $p_1,\ldots, p_n$
  - Change probability of page p_i: λ_i
  - Time of downloading page p_i: $t(p_i)$
  - Last modified value of page p_i: μ_i
  - Content hash or etag of page p_i: $\theta(p_i)$
  - Crawl interval: $[t_s,t_e]$

LiWA
Living Web Archives

Marc Spaniol

Coherence

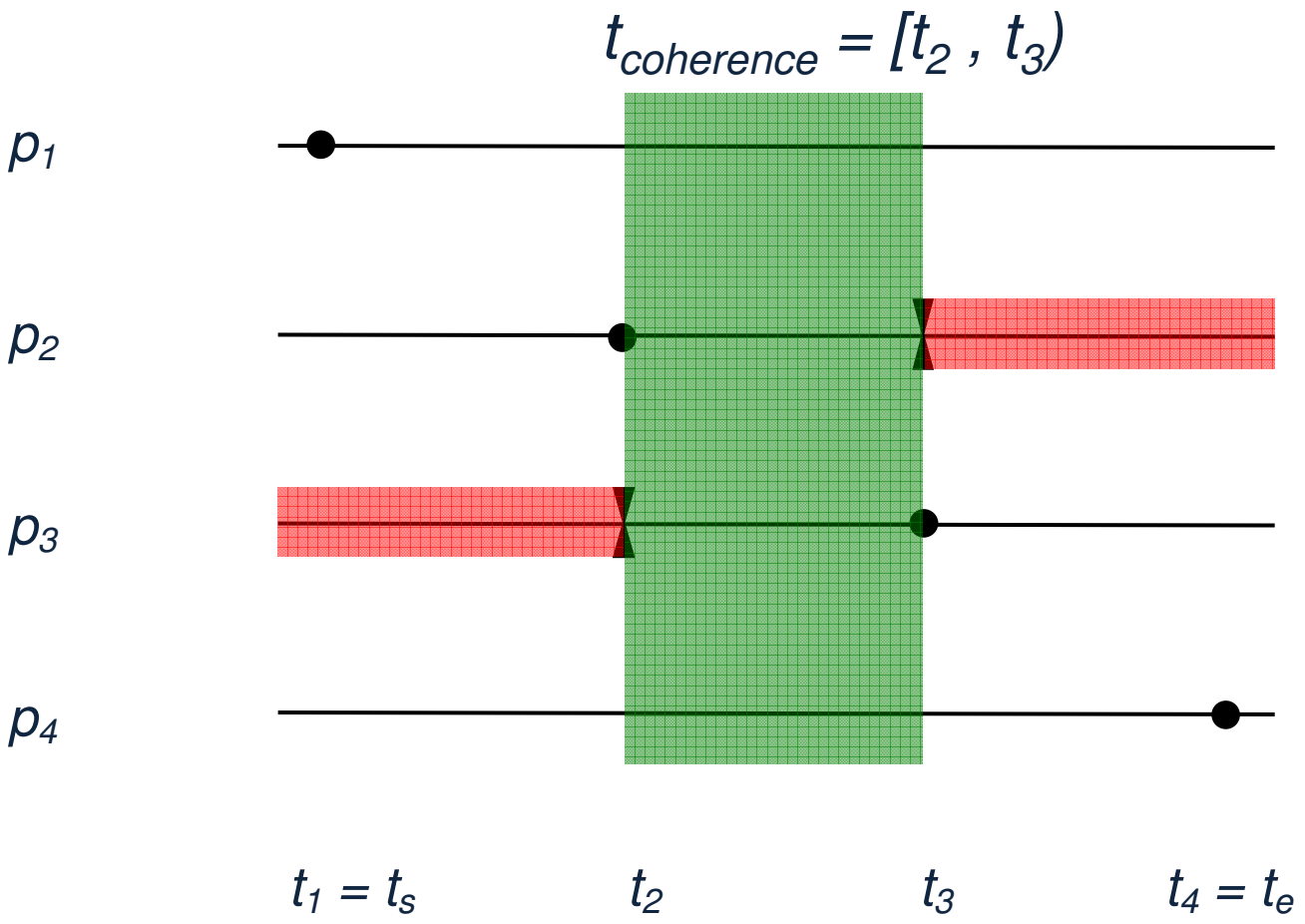
Definition:

1. A single Web page is always coherent.
2. The invariance interval $[\mu_i,\mu_i^*]$ of page p_i lies between the last modified time stamp μ_i at time $t(p_i)$ of downloading p_i ($\mu_i \leq t(p_i)$) and the next change μ_i^* following $t(p_i)$.
3. Two or more pages are coherent if

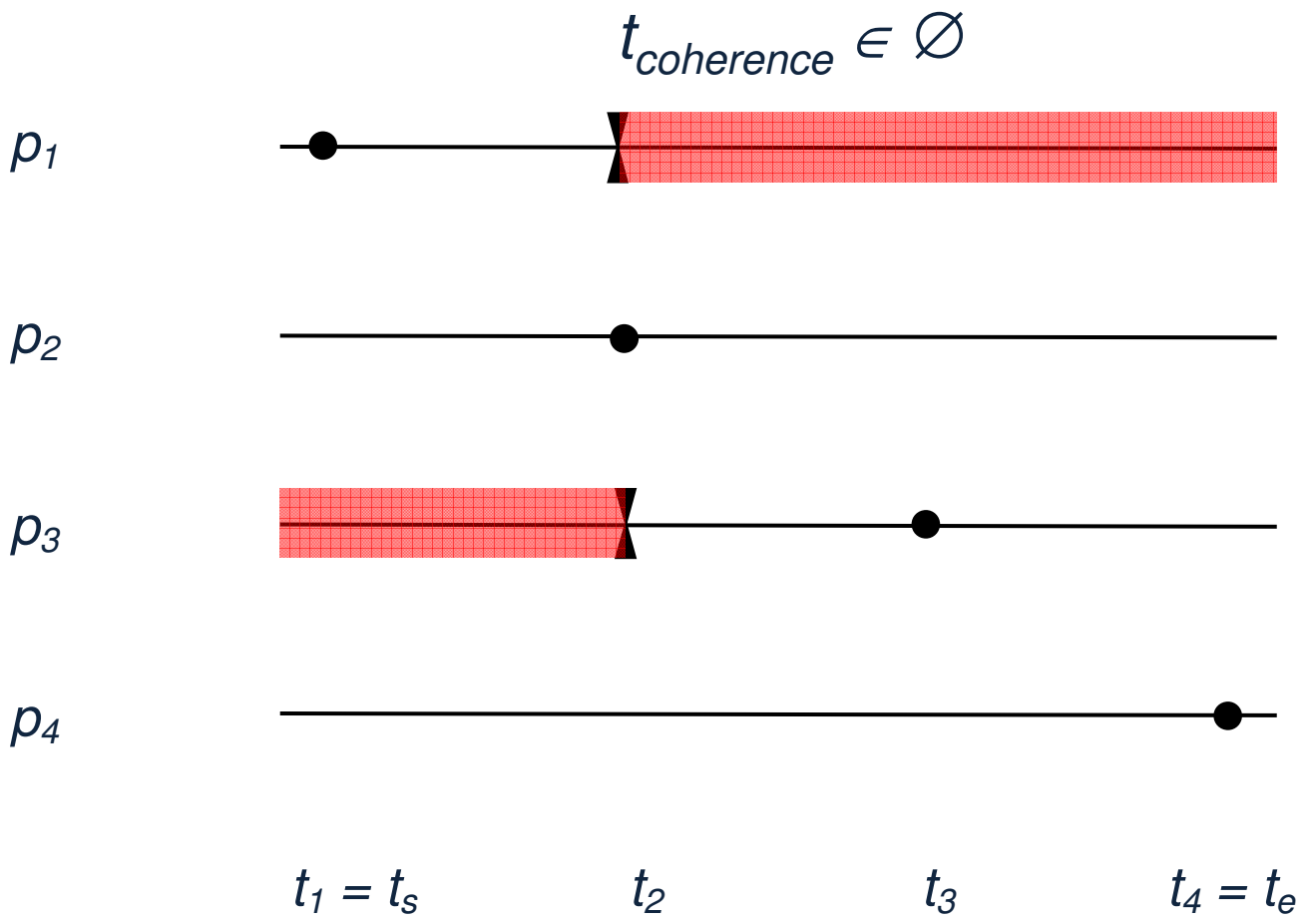
   there is a time point (or interval) $t_{coherence}$ so that a non-empty intersection among the invariance interval of all pages exists:

$$\forall p_i, \exists t_{coherence} : t_{coherence} \in \bigcap_{i=1}^{n} [\mu_i, \mu_i^*] \neq \emptyset$$

Coherence by Example



Coherence by Example



LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

Coherence Summary

- Definition
  - Simple
  - Straightforward
- Implementation
  - Impractical
  - Crawlers are

    aware of the past (“backward-conscious”)

    but

    unaware of the future (“forward-blind”)

⇒ Only limited applicability in a real life scenario

Observable Coherence

Definition:

Two or more pages are observable coherent if

there is a single timepoint (or interval) $t_{coherence}$ so that there is a non-empty intersection of the intervals spanning the respective download time $t(p_i)$ and the corresponding last modified stamp μ_i retrieved at time of download ($\mu_i \leq t(p_i)$):

$$\forall p_i, \exists t_{coherence} : \bigcap_{i=1}^{n} [\mu_i, t(p_i)] \neq \varnothing \wedge t_{coherence} \in [\mu_i, t(p_i)]$$

LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

Measurable Coherence

- Specialization of observable coherence
  - Makes observable coherence measurable in a real life scenario
  - Overcomes “right-hand side blindness” of crawlers
- Ability to issue a guaranteed coherence statement
  - Valid for all contents of a Web site
  - “Regardless” of crawl duration
- Suitable coherence time point (or interval) $t_{coherence}$ needed

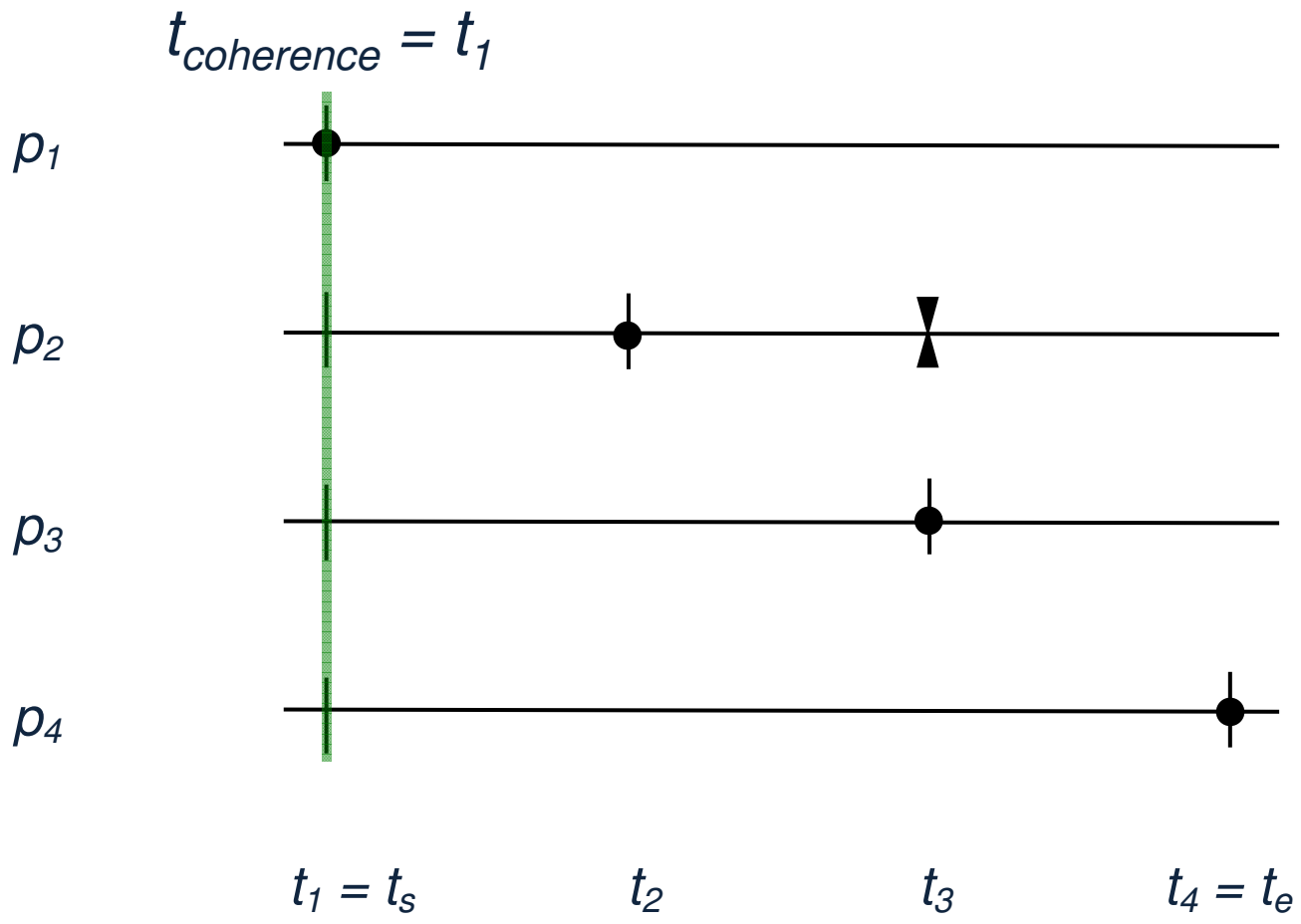
$\Rightarrow$ Full control is only given for $t_{coherence} = t_s$

mpii

LiWA
Living Web Archives

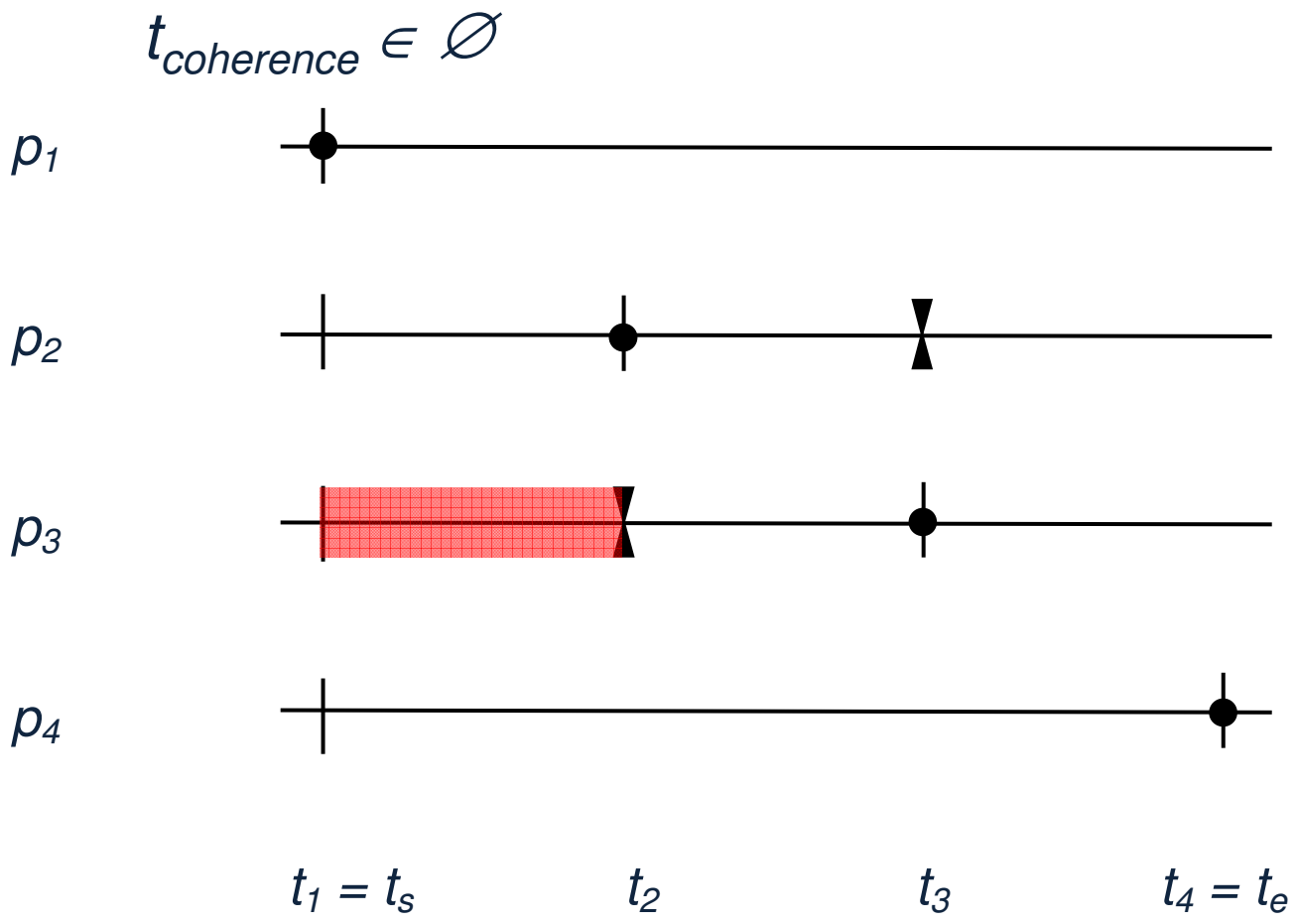
Marc Spaniol

Measurable Coherence by Example



Databases and Information Systems
Prof. Dr. G. Weikum

Measurable Coherence by Example



LiWA
Living Web Archives

Marc Spaniol

Quantifying Measurable Coherence

Error function

$$f(p_i) = \begin{cases} 0 & \text{, if } \mu_i \leq t_s \\ 1 & \text{, else} \end{cases}$$

Coherence function

$$C(c) = 1 - \frac{\sum_{i=1}^{n} f(p_i)}{n} \quad \text{, n} \geq 1$$

LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

Measurable Coherence Summary

- Definition
  - Makes coherence become measurable
  - Relative to start of crawl
- Implementation
  - “Easy”
  - Ad-hoc verifiable
  - Efficient
  - Produces no extra traffic
  - Relies on accuracy of last modified stamps

$\Rightarrow$ Only limited applicability in a real life scenario

Inducible Coherence

Definition:

Two or more pages are inducible coherent if

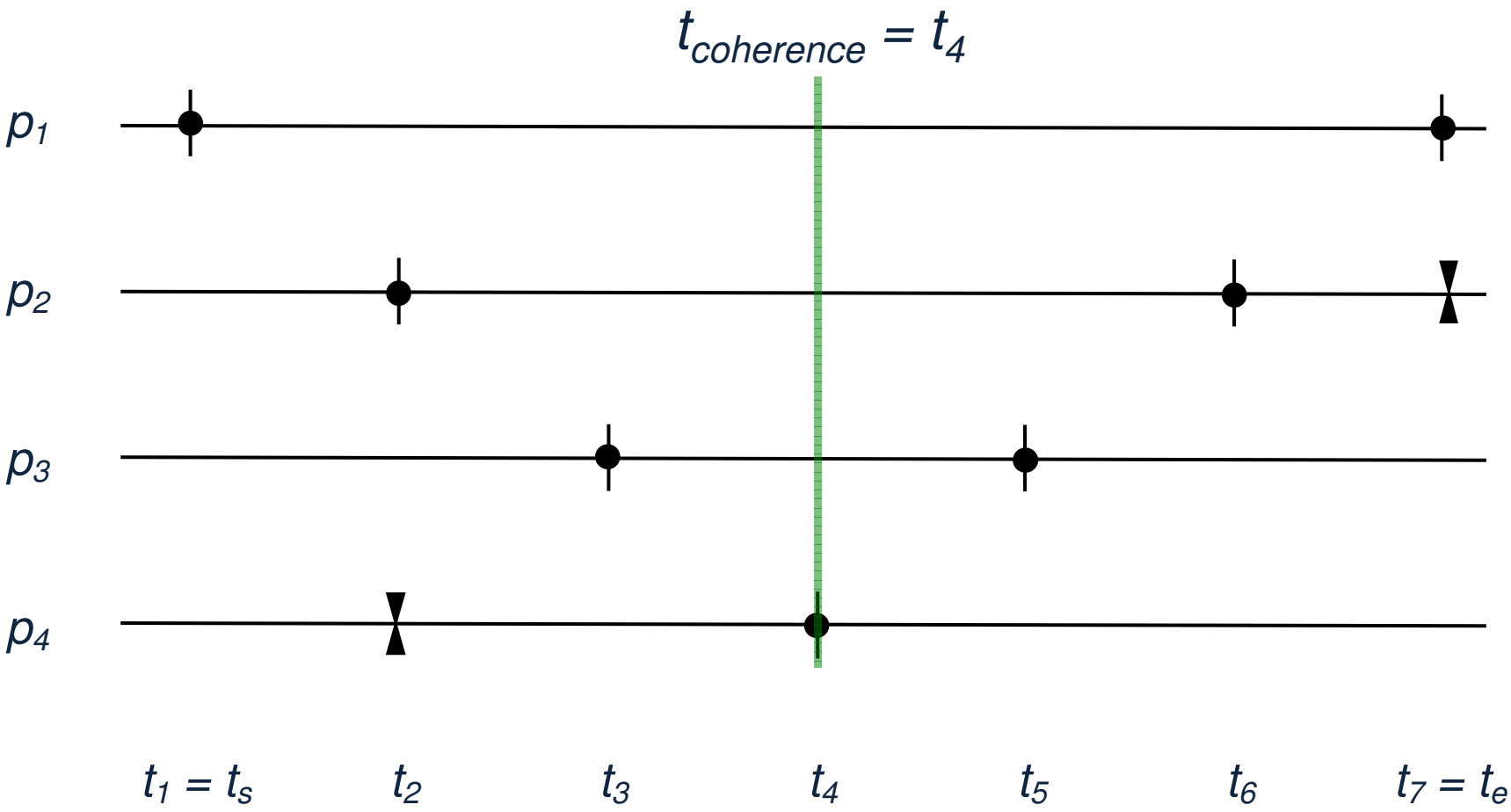
there is a time point $t_{coherence}$ between the visit of pages $t(p_i)$ and the subsequent revisit $t(\tilde{p}_i)$ where the etag or content hash θ of corresponding pages ($\theta(m)$ having $m \in \{p_i, \tilde{p}_i\}$) has not changed:

$$\forall p_i, \exists t_{coherence} : \theta(p_i) = \theta(\tilde{p}_i) \wedge t_{coherence} \in \bigcap_{i=1}^{n} [t(p_i), t(\tilde{p}_i)]$$

LiWA
Living Web Archives

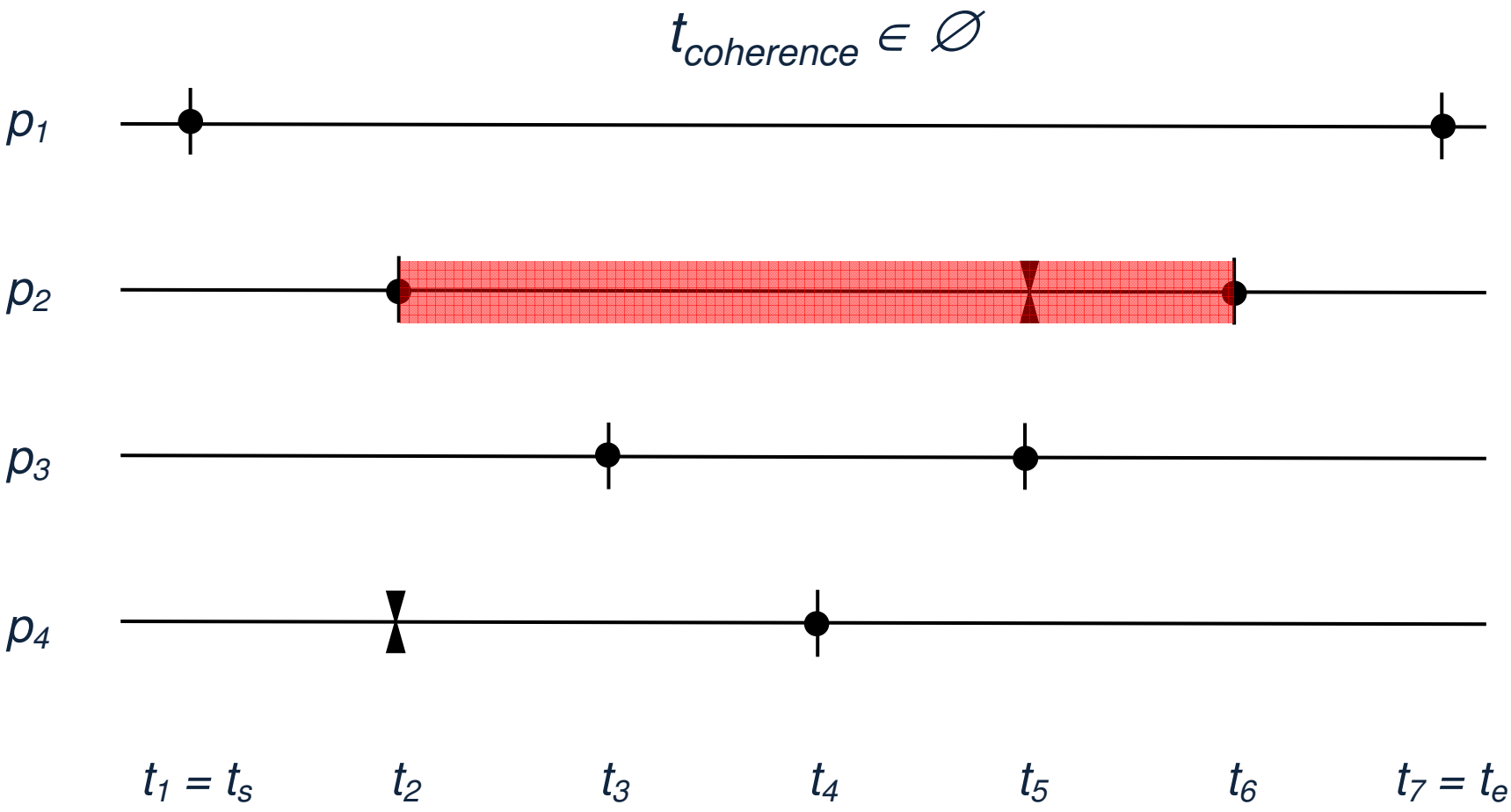
Marc Spaniol

Inducible Coherence by Example



Databases and Information Systems
Prof. Dr. G. Weikum

Inducible Coherence by Example



LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

Quantifying Inducible Coherence

Error function

$$f(p_i, \tilde{p}_i) = \begin{cases} 0 & \text{, if } \theta(p_i) = \theta(\tilde{p}_i) \\ 1 & \text{, else} \end{cases}$$

Coherence function

$$C(c) = 1 - \frac{\sum_{i=1}^{n} f(p_i, \tilde{p}_i)}{n} \quad , n \geq 1$$

Databases and Information Systems
Prof. Dr. G. Weikum

LiWA
Living Web Archives

Marc Spaniol

Inducible Coherence Summary

- Definition
  - Makes coherence of improper dated contents become measurable
  - Relative to end of crawl / start of revisit
- Implementation
  - More complex
  - Still efficient
    - Associated database
    - Politeness delays are the bottleneck
  - Produces extra traffic
    - Mostly conditional gets
    - Few “full” downloads
  - Full check on proper dating of contents

⇒ LiWA temporal coherence processor V1.0 in Heritrix

Databases and Information Systems
Prof. Dr. G. Weikum

LiWA
Living Web Archives

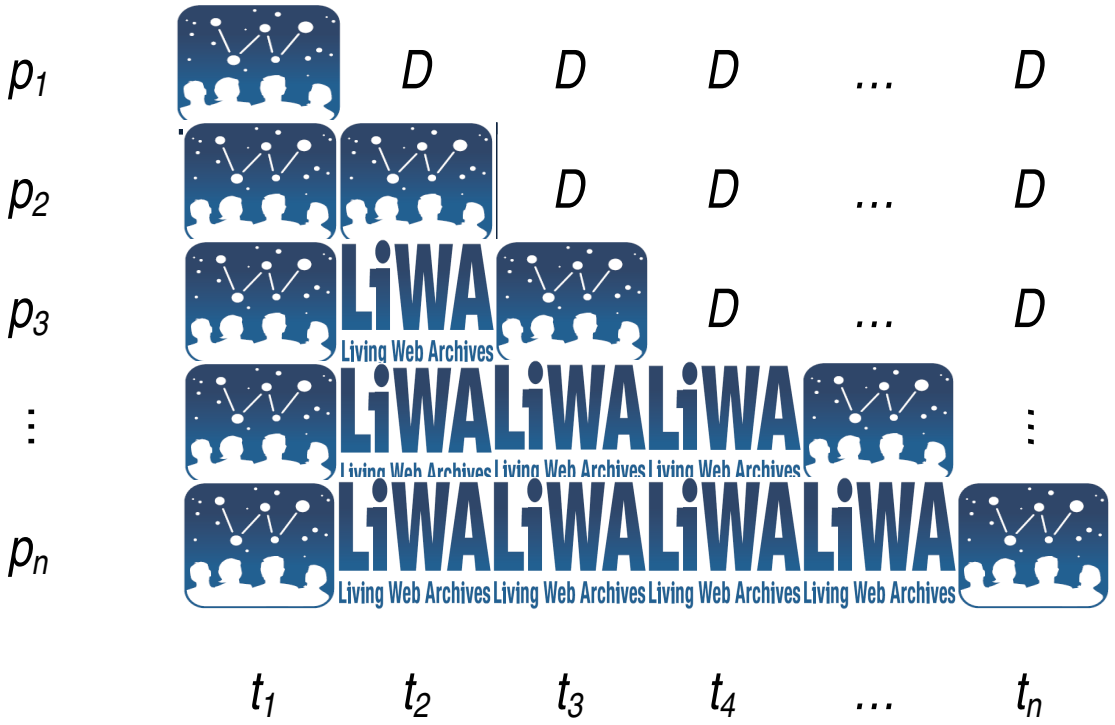
Data Quality in Web Archiving

Marc Spaniol

Databases and Information Systems
Prof. Dr. G. Weikum

Crawl Improvement: Measurable Coherence

- Conflict probability: $\kappa(p_i) = 1 - (1 - \lambda_i)^{t(p_i) - t_s}$
- Crawling so that conflicts are "tolerable": $\kappa(p_i) < \eta$



LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

mpii

Databases and Information Systems
Prof. Dr. G. Weikum

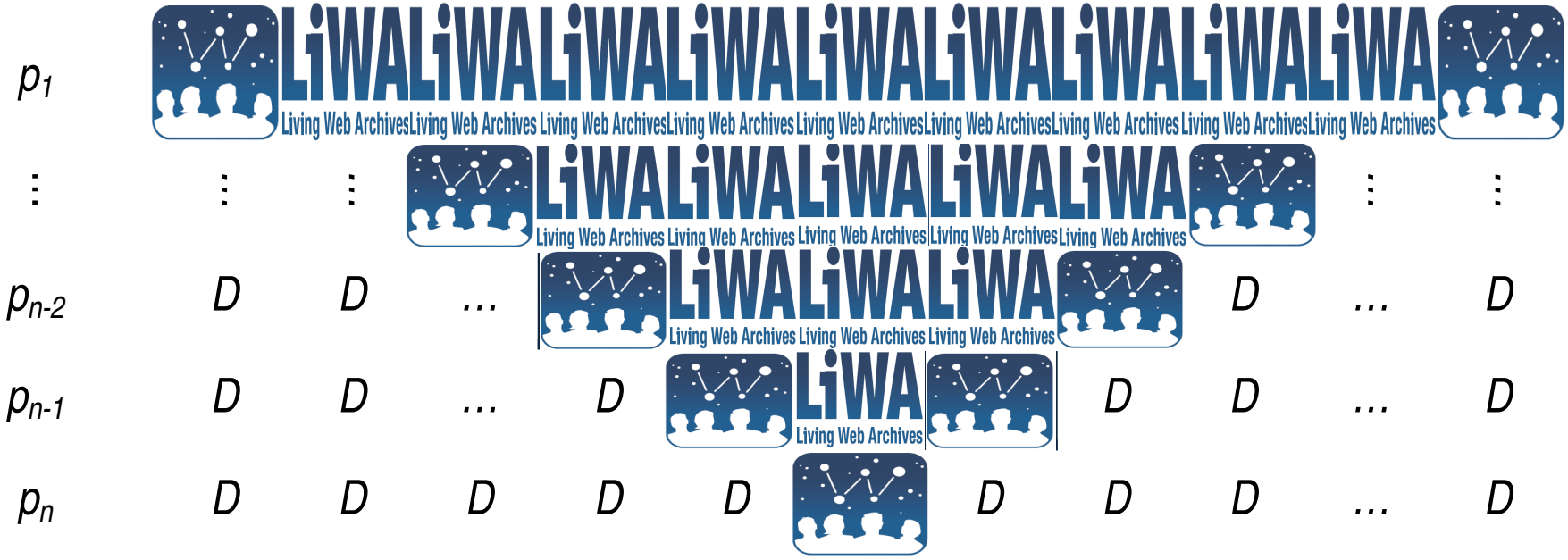
Improved Measurable Coherence Crawl Scheduling

input: $p_1,\ldots,p_n$ - list of pages in descending order of λ_i,
η - readiness to assume risk threshold

begin
Start with: *slot = 1*
while *slot* $\leq$ *n*
do
if $\kappa(p_{slot}) < \eta$ **then** /* **no conflict expected** */
Download page p_{slot}
end
Continue with next iteration: *slot ++*
end
Download skipped pages in reversed order of their index
end

Crawl Improvement: Inducible Coherence

- Conflict probability: $\kappa(p_i) = 1 - (1 - \lambda_i)^{t(\tilde{p}_i) - t(p_i)}$
- Crawling so that conflicts are "tolerable": $\kappa(p_i) < \eta$



LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

Databases and Information Systems
Prof. Dr. G. Weikum

Improved Inducible Coherence Crawl Scheduling

input: $p_1,\ldots,p_n$ - list of pages in descending order of λ_i,
η - readiness to assume risk threshold

begin
Start with: *slot = 1*, *lastpromising = n*
while *slot* $\leq$ *lastpromising*
do
if $\kappa(p_{slot}) \geq \eta$ **then** /* **conflict expected!** */
Move p_{slot} to position *lastpromising*
Decrease promising boundary: *lastpromising* −−
end
else
Increase promising boundary: *promising* ++
end
end
slot = n **while** *slot* ≥ 1
do /* **visit from hopeless to promising** */
Download page p_{slot}
Decrease slot counter: *slot* −−
end
slot = 2 **while** *slot* $\leq n$
do /* **revisit from promising to hopeless** */
Revisit page p_{slot}
Increase slot counter: *slot* ++
end
end

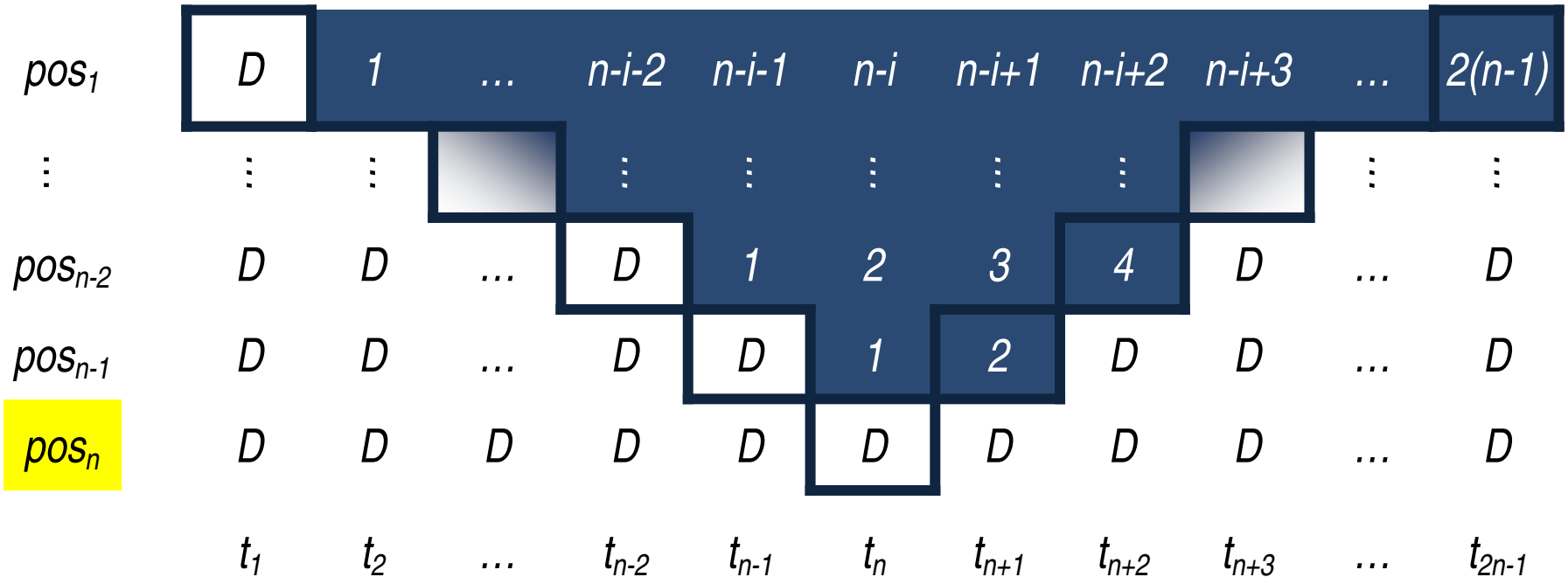
Inducible Coherence Scheduling: pos_n

Position: pos_n
Remaining λ_i : $\lambda_{n-1} >> \lambda_{n-n-2} >> \lambda_{n-2} >> \lambda_1 > \lambda_1$

Test: $1 - (1 - \lambda_n)^0 > \theta$?
$\Rightarrow$ *No!*

Promising: λ_n

Hopeless:



Inducible Coherence Scheduling: pos_{n-1}

Position: pos_{n-1}
Remaining λ_i : $\lambda_{n-3} > \lambda_{n-2} > \lambda_{n-1} > \ldots > \lambda_1$

Test: $1 - (1 - \lambda_{n-3})^2 > \theta$?
e.g. Yes!

Promising: λ_{n-3}, λ_n

Hopeless: λ_{n-1} λ_{n-2}

pos_1	D	1	…	n-i-2	n-i-1	n-i	n-i+1	n-i+2	n-i+3	…	2(n-1)
⋮	⋮	⋮		⋮	⋮	⋮	⋮	⋮		⋮	⋮
pos_{n-2}	D	D	…	D	1	2	3	4	D	…	D
pos_{n-1}	D	D	…	D	D	1	2	D	D	…	D
pos_n	D	D	D	D	D	D	D	D	D	…	D
	t_1	t_2	…	t_{n-2}	t_{n-1}	t_n	t_{n+1}	t_{n+2}	t_{n+3}	…	t_{2n-1}

Inducible Coherence Scheduling: pos_{n-2}

Position: pos_{n-2}
Remaining λ_i : $\lambda_{n-3} > \ldots > \lambda_1$

Test: $1 - (1 - \lambda_{n-3})^4 > \theta$?
e.g. Yes!

Promising: λ_{n-3}, λ_{n-2}, λ_n

Hopeless: λ_{n-1}, λ_{n-2}, λ_{n-4}

pos_1	D	1	…	n-i-2	n-i-1	n-i	n-i+1	n-i+2	n-i+3	…	2(n-1)
⋮	⋮	⋮		⋮	⋮	⋮	⋮	⋮		⋮	⋮
pos_{n-2}	D	D	…	D	1	2	3	4	D	…	D
pos_{n-1}	D	D	…	D	D	1	2	D	D	…	D
pos_n	D	D	D	D	D	D	D	D	D	…	D
	t_1	t_2	…	t_{n-2}	t_{n-1}	t_n	t_{n+1}	t_{n+2}	t_{n+3}	…	t_{2n-1}

Inducible Coherence Scheduling: pos_{n-k}

Position: pos_k
Remaining λ_i : $\lambda_i > \lambda_1$

Test: $1 - (1 - \lambda_2)^{2(k-1)} > \theta$?
e.g. Mo's!

Promising: $\lambda_4, \ldots, \lambda_{n-7}, \lambda_{n-6}, \lambda_{n-5}, \lambda_{n-3}, \lambda_n \; \lambda_n$

Hopeless: $\lambda_{n-1}, \lambda_{n-2}, \lambda_{n-4}, \lambda_{n-8}, \lambda_{n-12}, \ldots, \lambda_3 \; \lambda_2$

pos_1	D	1	…	n-i-2	n-i-1	n-i	n-i+1	n-i+2	n-i+3	…	2(n-1)
⋮	⋮	⋮		⋮	⋮	⋮	⋮	⋮		⋮	⋮
pos_{n-2}	D	D	…	D	1	2	3	4	D	…	D
pos_{n-1}	D	D	…	D	D	1	2	D	D	…	D
pos_n	D	D	D	D	D	D	D	D	D	…	D
	t_1	t_2	…	t_{n-2}	t_{n-1}	t_n	t_{n+1}	t_{n+2}	t_{n+3}	…	t_{2n-1}

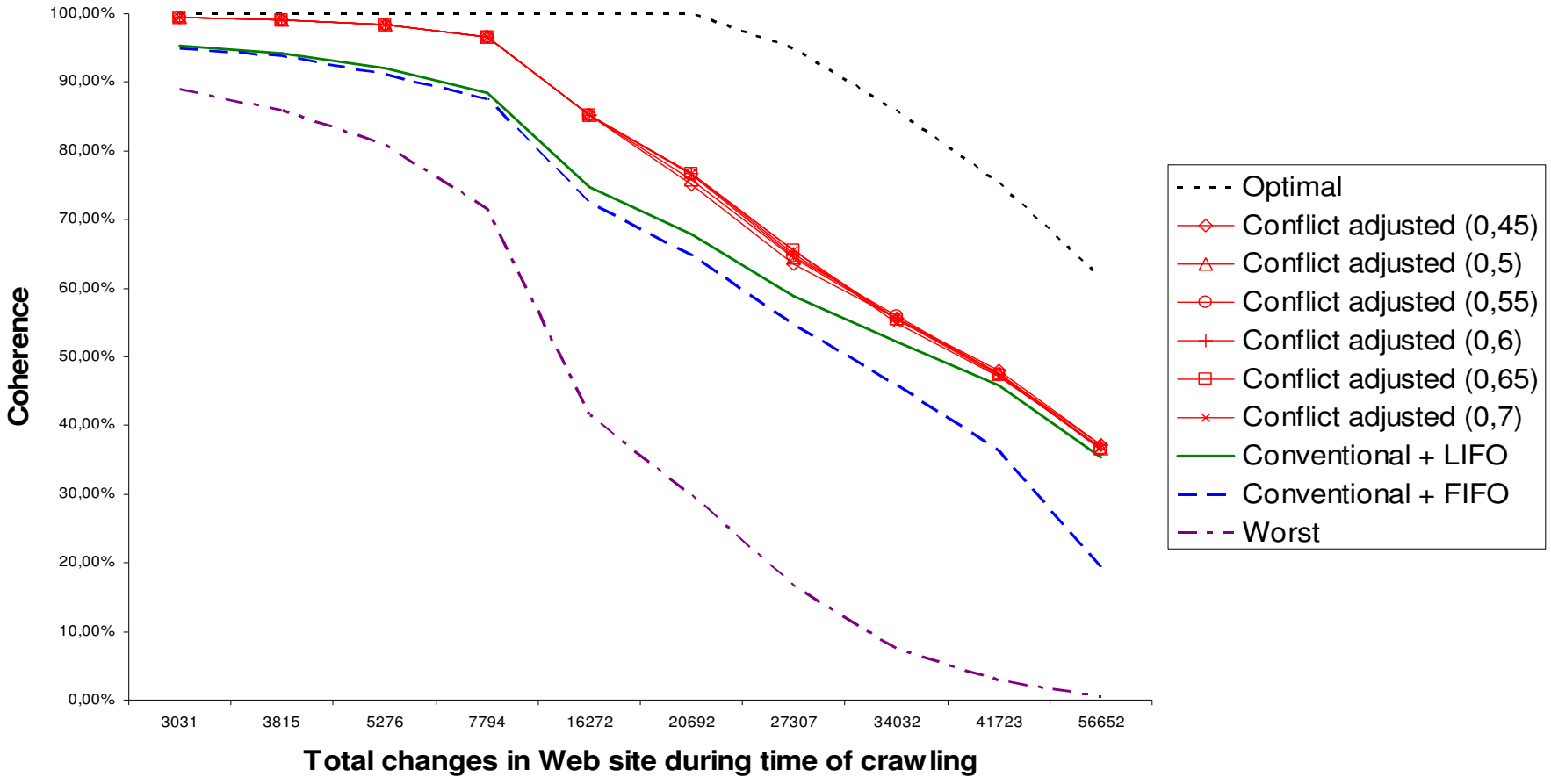
Final Crawl Sequence

Visit: $\lambda_{n-1}, \lambda_{n-2}, \lambda_{n-4}, \lambda_{n-8}, \lambda_{n-12}, \ldots, \lambda_3, \lambda_2, \lambda_1, \lambda_4, \ldots, \lambda_{n-7}, \lambda_{n-6}, \lambda_{n-5}, \lambda_{n-3}, \lambda_n$

Revisit: $\lambda_{n-3}, \lambda_{n-5}, \lambda_{n-6}, \lambda_{n-7}, \ldots, \lambda_4, \lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{n-12}, \lambda_{n-6}, \lambda_{n-5}, \lambda_{n-3}, \lambda_{n-1}$

pos_1	D	1	…	n-i-2	n-i-1	n-i	n-i+1	n-i+2	n-i+3	…	2(n-1)
⋮	⋮	⋮		⋮	⋮	⋮	⋮	⋮		⋮	⋮
pos_{n-2}	D	D	…	D	1	2	3	4	D	…	D
pos_{n-1}	D	D	…	D	D	1	2	D	D	…	D
pos_n	D	D	D	D	D	D	D	D	D	…	D
	t_1	t_2	…	t_{n-2}	t_{n-1}	t_n	t_{n+1}	t_{n+2}	t_{n+3}	…	t_{2n-1}

Improved Inducible Coherence Crawling: Simulation of a Website 10.000 contents



LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

Databases and Information Systems
Prof. Dr. G. Weikum

LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

Databases and Information Systems
Prof. Dr. G. Weikum

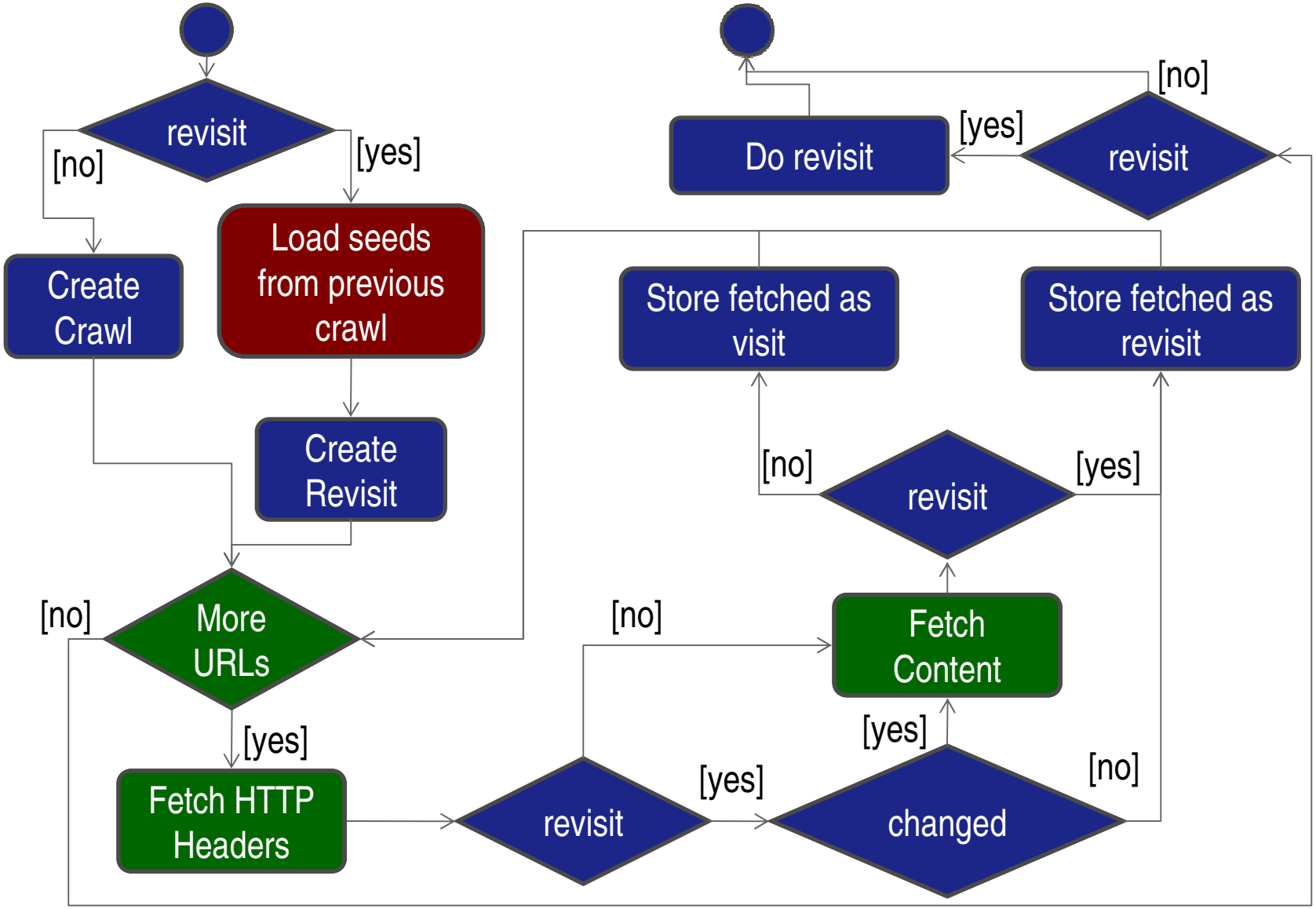
Incoherence Detection

- "Easy" for offline coherence analysis
  - Data is stored in the archive
  - Efficiency "unproblematic"
- "Difficult" for online coherence analysis
  - Data needs to be "tapped" from the crawler
  - Efficiency is a key issue
- Proper dating of content versions required
- Multistage change measurement procedure
  1) *Conditional GET (etag comparison)*
  2) *Check content timestamp (last modified comparison)*
  3) *Compare a hash of the page with a stored hash*
  4) *Non-significant differences (ads, fortunes, request timestamp)*
     - *only hash text content, or "useful" text content*
     - *compare distribution of n-grams (shingling)*
     - *compute edit distance with previous version*

Data Quality in Web Archiving

Marc Spaniol

Online Coherence Analysis



Databases and Information Systems
Prof. Dr. G. Weikum

LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

Databases and Information Systems
Prof. Dr. G. Weikum

Experimental Results

- Crawling with Heritrix crawler (LiWA coherence analysis processor integrated)
  - (Meta-)Data extraction embedded
  - Revisiting strategy integrated
- Experimental crawls on the mpi-inf.mpg.de domain
  - ~ 65.000 contents
  - > 25 GB
  - ~ 4,5 hours
  - ~ 6 documents per second
- Coherence defect analysis
  - Incoherence < 3 ‰
    - ~ 1 ‰ dynamics in dns look-ups
    - ~ 1 ‰ dynamically created wiki pages
    - ~ 1 ‰ dynamically created pages by bioinf.mpi-inf.mpg.de CMS
    - < 5 pages with relevant changes
  - Largest coherent sub graph: ~20.000 contents

LiWA
Living Web Archives

Marc Spaniol

Databases and
Information Systems
Prof. Dr. G. Weikum

Post Processing via GraphML

- File format for graphs
  - Core for the description of structural properties
  - Extension mechanisms for application-specific data
- Main features include
  - Directed, undirected, and mixed graphs
  - Hypergraphs
  - Hierarchical graphs
  - Graphical representations
  - References to external data
  - Application-specific attribute data
  - Light-weight parsers
- Based on XML
- Applied in many graph related software applications

LiWA
Living Web Archives

Data Quality in Web Archiving

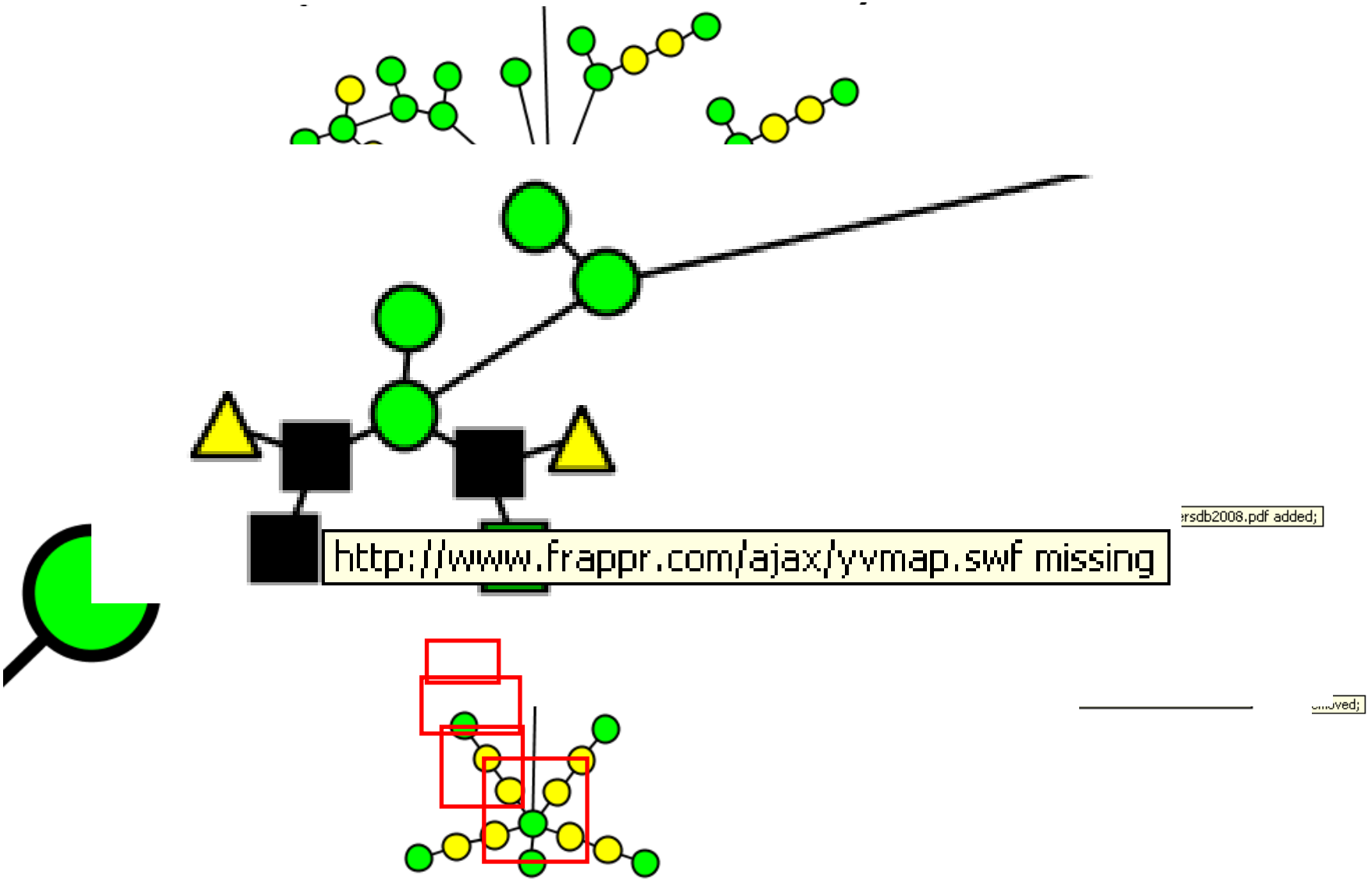
Marc Spaniol

Databases and Information Systems
Prof. Dr. G. Weikum

GraphML Excerpt

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns/graphml"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:y="http://www.yworks.com/xml/graphml"
    xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns/graphml
    http://www.yworks.com/xml/schema/graphml/1.0/ygraphml.xsd">
    …
    <graph edgedefault="directed" id="G229">
        <node id="http://www.mpi-inf.mpg.de/index.html">
            <data key="d0">
                <y:ShapeNode>
                    <y:Geometry width="10.003" height="10.003"/>
                    <y:Fill color="#00FF00" transparent="false"/>
                    <y:Shape type="ellipse"/>
                </y:ShapeNode>
            </data>
            <data key="d1">http://www.mpi-inf.mpg.de/index.html OK</data>
        </node>
        …
        <edge source="http://www.mpi-inf.mpg.de/index.html" target="dns:www.mpi-inf.mpg.de"/>
        …
    </graph>
</graphml>
```

Online Coherence Visualization: mpi-inf.mpg.de



LiWA
Living Web Archives

Data Quality in Web Archiving

Marc Spaniol

Databases and Information Systems
Prof. Dr. G. Weikum

LiWA
Living Web Archives

Marc Spaniol

Summary

- Measuring data quality in Web archives requires
  - Identification of relevant coherence defects
  - Data scrubbing in order to compare relevant document (sub-)sections
  - Efficient computations
    - Shingling
    - Minhashing
    - (Local sensitive hashing)
- Identification of coherence
  - HTTP time stamps → Measurable coherence
  - "Virtual" time stamps → Inducible coherence
- Data quality in Web archiving
  - Quantifiable even with improper dated contents
  - Improvable by around 10% given non-pathological Web sites (scheduling only!)
  - Can be further improved by more sophisticated search strategies
    - Partial captures
    - Identification of media- and/or URL-specific change behavior

LiWA
Living Web Archives

References

[Chak03] S. Chakrabarti: “Mining the Web”. Morgan Kaufmann, 2003.

[Masa06] J. Masanès: “Web Archiving”. Springer, New York, Inc., Secaucus, NJ, 2006.

[Ullm00] J. Ullman: “Correlated Items”. CS345 --- Lecture Notes, 2000.
http://www-db.stanford.edu/~ullman/mining/minhash.pdf
[last access: June 3, 2009]

[SDM*09] M. Spaniol, D. Denev, A. Mazeika, P. Senellart and G. Weikum: “Data Quality in Web Archiving”. Proceedings of the 3rd Workshop on Information Credibility on the Web (WICOW 2009), pp. 19-26, 2009.
http://www.dl.kuis.kyoto-u.ac.jp/wicow3/papers/p19-spaniolA.pdf
[last access: June 3, 2009]