Probabilistic Scheduling for Top-k Index Processing

Anna Moleda

Max Planck Institute for Computer Science
International Max Planck Research School

supervised by: Martin Theobald and Gerhard Weikum

Oberseminar, 23th of March 2005

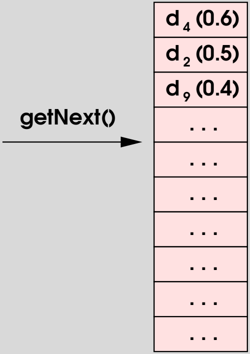
Example Query

- natural language query:

  What is the history of the Phoenix symbol?
- query in terms: history phoenix symbol
- get index lists for query terms

How to process index lists?

sorted access



getNext()

d_4 (0.6)

d_2 (0.5)

d_9 (0.4)

. . .

. . .

. . .

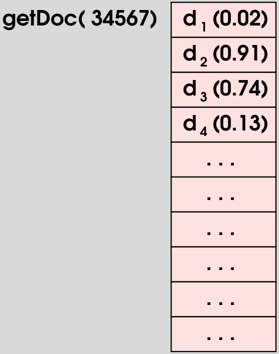
. . .

. . .

. . .

. . .

random access



getDoc(34567)

d_1 (0.02)

d_2 (0.91)

d_3 (0.74)

d_4 (0.13)

. . .

. . .

. . .

. . .

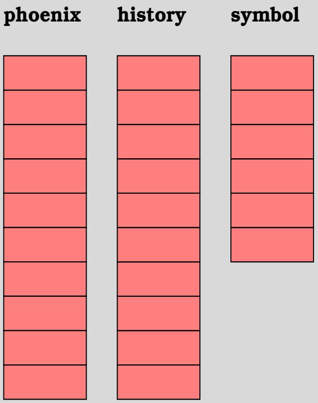
. . .

. . .

Algorithm for Sorted Accesses

Top-3

Candidates

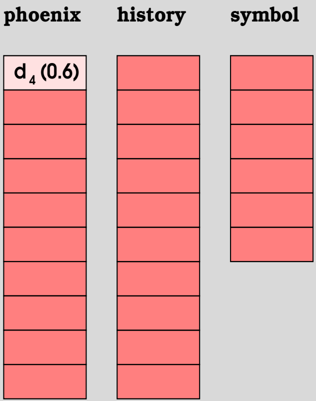


Algorithm for Sorted Accesses

Top-3

$0.6 \leq d_4 \leq 2.6$

Candidates

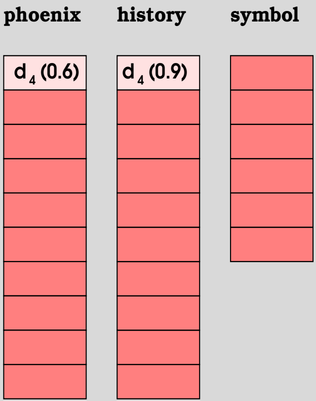


Algorithm for Sorted Accesses

Top-3

$1.5 \leq d_4 \leq 2.5$

Candidates

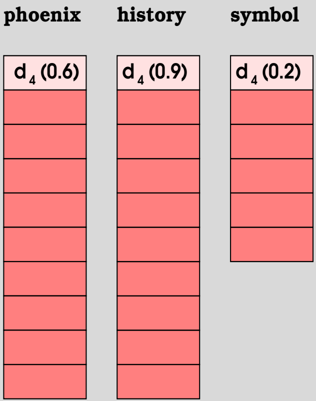


Algorithm for Sorted Accesses

Top-3

$d_4 = 1.7$

Candidates



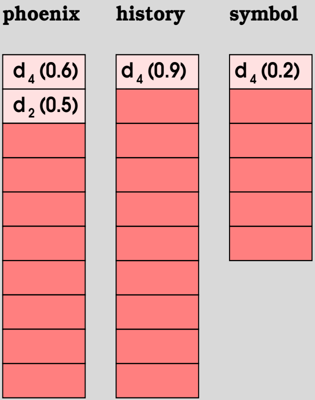
Algorithm for Sorted Accesses

Top-3

$d_4 = 1.7$

$0.5 \leq d_2 \leq 1.6$

Candidates



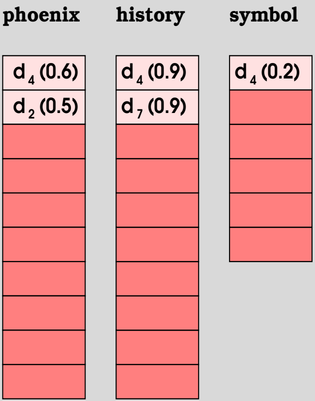
phoenix	history	symbol
d_4 (0.6)	d_4 (0.9)	d_4 (0.2)
d_2 (0.5)		

Algorithm for Sorted Accesses

Top-3

$d_4 = 1.7$
$0.5 \leq d_2 \leq 1.6$
$0.9 \leq d_7 \leq 1.6$

Candidates



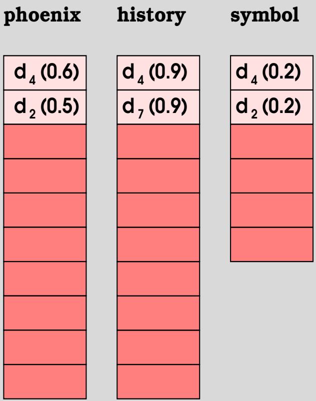
phoenix	history	symbol
d$_4$ (0.6)	d$_4$ (0.9)	d$_4$ (0.2)
d$_2$ (0.5)	d$_7$ (0.9)	🟥
🟥	🟥	🟥
🟥	🟥	🟥
🟥	🟥	🟥
🟥	🟥	🟥
🟥	🟥	
🟥	🟥	
🟥	🟥	
🟥	🟥	

Algorithm for Sorted Accesses

Top-3

$d_4 = 1.7$
$0.7 \leq d_2 \leq 1.6$
$0.9 \leq d_7 \leq 1.6$

Candidates



phoenix	history	symbol
d_4 (0.6)	d_4 (0.9)	d_4 (0.2)
d_2 (0.5)	d_7 (0.9)	d_2 (0.2)

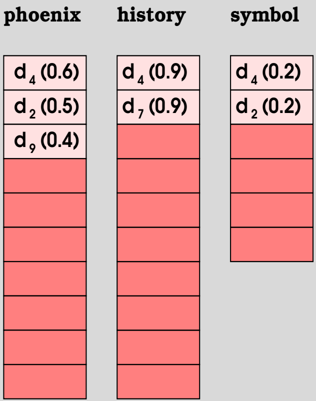
Algorithm for Sorted Accesses

Top-3

$d_4 = 1.7$
$0.7 \leq d_2 \leq 1.6$
$0.9 \leq d_7 \leq 1.5$

Candidates

$0.4 \leq d_9 \leq 1.5$



phoenix	history	symbol
d_4 (0.6)	d_4 (0.9)	d_4 (0.2)
d_2 (0.5)	d_7 (0.9)	d_2 (0.2)
d_9 (0.4)		

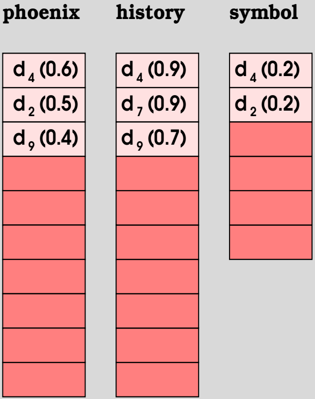
Algorithm for Sorted Accesses

Top-3

$d_4 = 1.7$
$0.7 \leq d_2 \leq 1.4$
$0.9 \leq d_7 \leq 1.5$

Candidates

$1.1 \leq d_9 \leq 1.3$



phoenix	history	symbol
d_4 (0.6)	d_4 (0.9)	d_4 (0.2)
d_2 (0.5)	d_7 (0.9)	d_2 (0.2)
d_9 (0.4)	d_9 (0.7)	

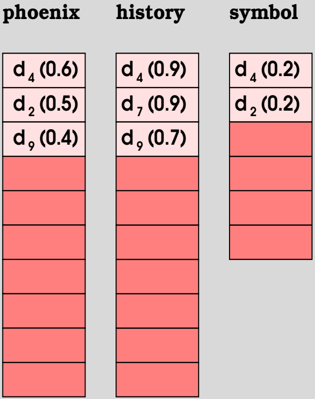
Algorithm for Sorted Accesses

Top-3

$d_4 = 1.7$
$1.1 \leq d_9 \leq 1.3$
$0.9 \leq d_7 \leq 1.5$

Candidates

$0.7 \leq d_2 \leq 1.4$



phoenix	history	symbol
d_4 (0.6)	d_4 (0.9)	d_4 (0.2)
d_2 (0.5)	d_7 (0.9)	d_2 (0.2)
d_9 (0.4)	d_9 (0.7)	

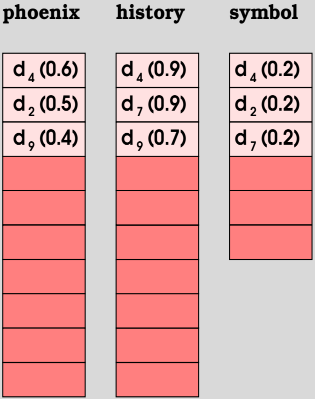
Algorithm for Sorted Accesses

Top-3

$d_4 = 1.7$
$1.1 \leq d_9 \leq 1.3$
$1.1 \leq d_7 \leq 1.3$

Candidates

$0.7 \leq d_2 \leq 1.4$



phoenix	history	symbol
d_4 (0.6)	d_4 (0.9)	d_4 (0.2)
d_2 (0.5)	d_7 (0.9)	d_2 (0.2)
d_9 (0.4)	d_9 (0.7)	d_7 (0.2)

Algorithm for Sorted Accesses

Top-3

$d_4 = 1.7$
$1.1 \leq d_9 \leq 1.3$
$1.1 \leq d_7 \leq 1.3$

Candidates

$0.7 \leq d_2 \leq 1.4$

phoenix	history	symbol
d_4 (0.6)	d_4 (0.9)	d_4 (0.2)
d_2 (0.5)	d_7 (0.9)	d_2 (0.2)
d_9 (0.4)	d_9 (0.7)	d_7 (0.2)
...	...	...
...	...	...
...	...	...
...	...	
...	...	
...	...	
...	...	

The Variety of Lists

- list length
- score domain $[1.0, 0.0]$

The Variety of Lists

- list length
  - long lists - $> 200\,000$ items
  - short lists - $< 5\,000$ items
- score domain $[1.0, 0.0]$

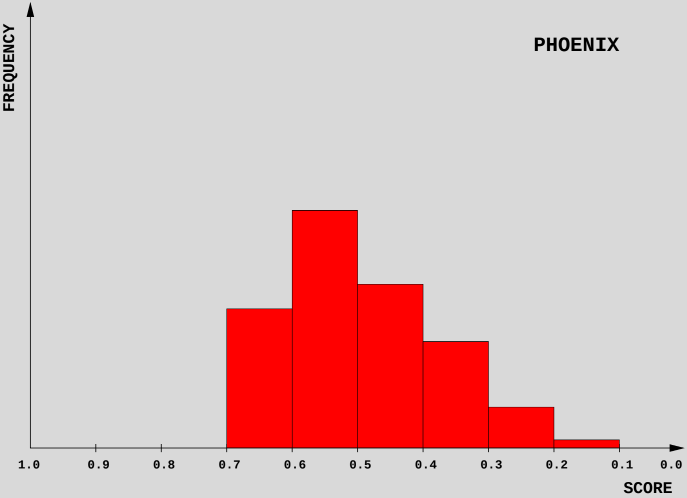
The Variety of Lists

- list length
  - long lists - $>200\,000$ items
  - short lists - $<5\,000$ items
- score domain $[1.0, 0.0]$
  - lists with high scores will give us good candidates
  - lists with low scores will reduce candidates' best scores

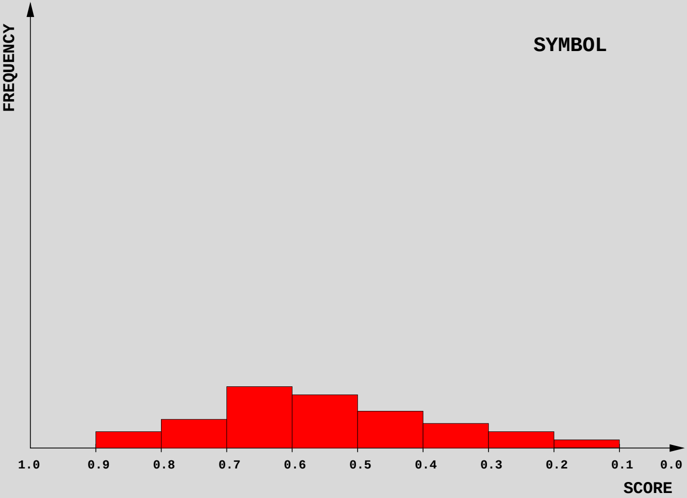
Schedules

- round-robin
- predicting score decrease
- candidates pruning

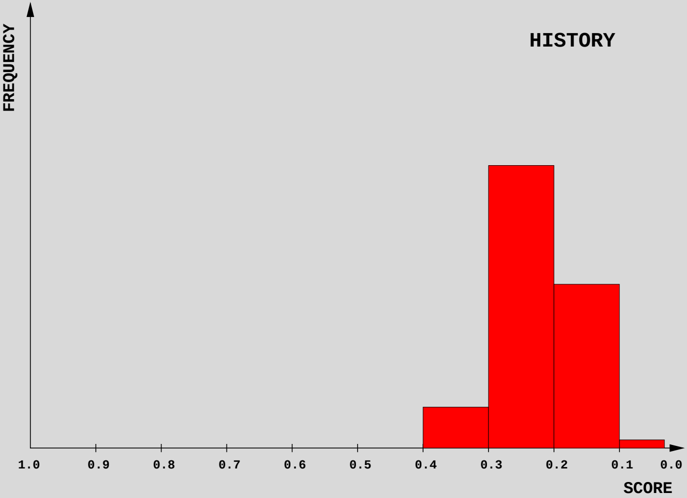
Predicting Score Decrease



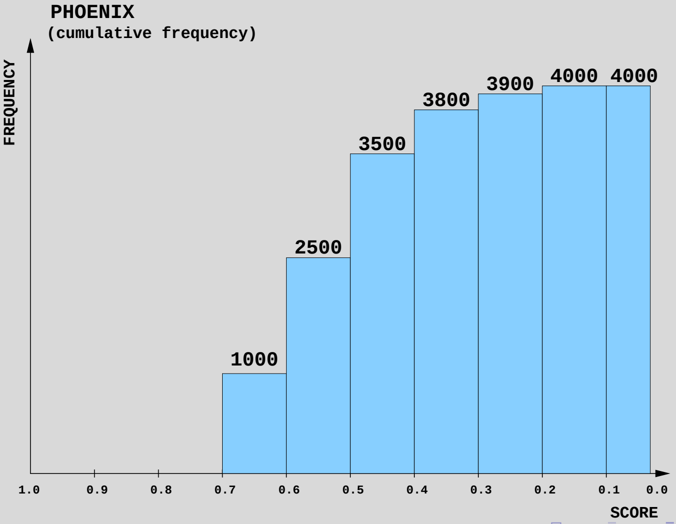
Predicting Score Decrease



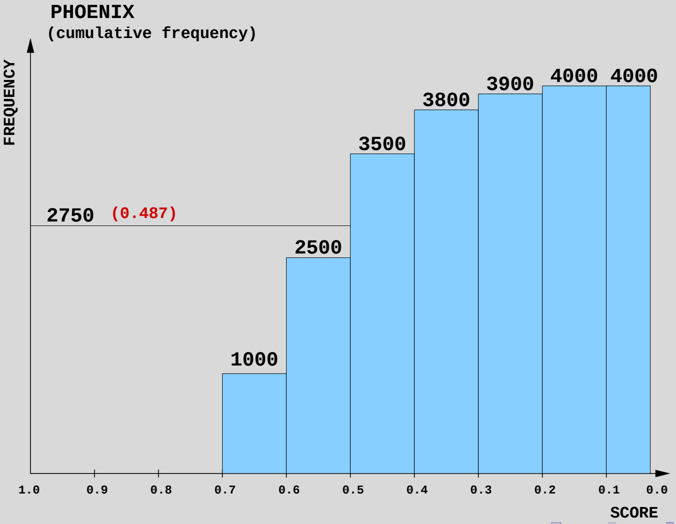
Predicting Score Decrease



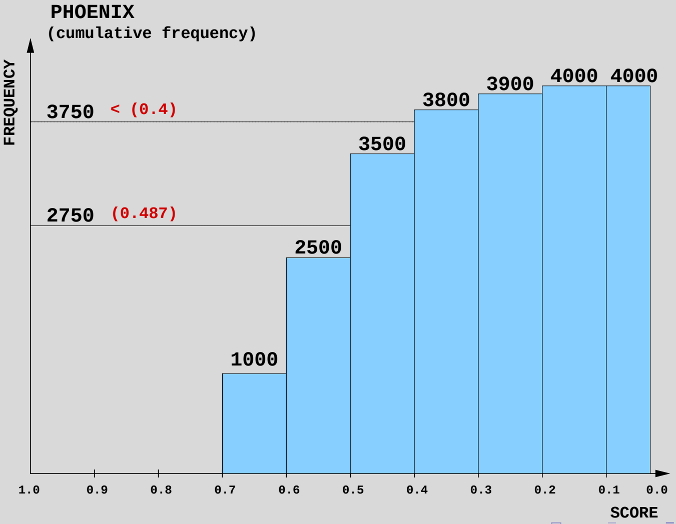
Predicting Score Decrease



Predicting Score Decrease



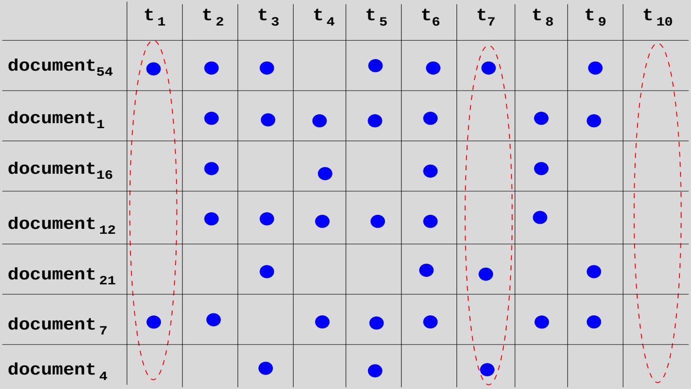
Predicting Score Decrease



Candidates Pruning

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
document$_{54}$	●	●	●		●	●	●		●	
document$_{1}$		●	●	●	●	●		●	●	
document$_{16}$		●		●		●		●		
document$_{12}$		●	●	●	●	●		●		
document$_{21}$			●			●	●		●	
document$_{7}$	●	●		●	●	●		●	●	
document$_{4}$			●		●		●			

Candidates Pruning



	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
$document_{54}$	●	●	●		●	●	●		●	
$document_1$		●	●	●	●	●		●	●	
$document_{16}$		●		●		●		●		
$document_{12}$		●	●	●	●	●		●		
$document_{21}$			●			●	●		●	
$document_7$	●	●		●	●	●		●	●	
$document_4$			●		●		●			

Random Accesses

- conjunctive queries
- negation
- small number of candidates left
- very bad score distribution and long lists

Miscellaneous

- dynamic threshold
- scanning in phases
- dynamic number of threads

Results for Sorted Access Only

.GOV collection, 50 queries

Results for Sorted Access Only

.GOV collection, 50 queries

- prediction of score decrease - 0% :-(

Results for Sorted Access Only

.GOV collection, 50 queries

- prediction of score decrease - 0% :-(
- candidates pruning - $7 - 14\%$