15-826: Multimedia Databases and Data Mining

Lecture #15: Text - part II

C. Faloutsos

Must-read Material

• Textbook, Chapter 6

Optional (but terrific to read)

Outline

Goal: ‘Find similar / interesting things’

- Intro to DB
- Indexing - similarity search
- Data Mining

Indexing - Detailed outline

- primary key indexing
- secondary key / multi-key indexing
- spatial access methods
- fractals
- text
- multimedia
- ...

Text - Detailed outline

- text
  - problem
  - full text scanning
  - inversion
  - signature files
  - clustering
  - information filtering and LSI
Q: space overhead?

A: mainly, the postings lists
Text - Inversion

• how to organize dictionary?
  – B-tree, hashing, TRIEs, PATRICIA trees, ...
• stemming – Y/N?
• insertions?

Text - Inversion

• variations:
• Parallelism [Tomasic+,93]
• Insertions [Tomasic+94], [Brown+]
  – ‘zipf’ distributions
• Approximate searching (‘glimpse’ [Wu+])
Text - Inversion

• postings list – more Zipf distr.: eg., rank-frequency plot of ‘Bible’

\[
\log(\text{freq}) = \frac{1}{\text{rank}} / \ln(1.78V)
\]

Text - Inversion

• postings lists
  – Cutting+Pedersen
    • (keep first 4 in B-tree leaves)
  – how to allocate space: [Faloutsos+92]
    • geometric progression
  – compression (Elias codes) [Zobel+] – down to 2% overhead!
  – Compression and doc reordering [Blandford+2002]

Conclusions

• Conclusions: needs space overhead (2%-300%), but it is the fastest
Text - Detailed outline

- text
  - problem
  - full text scanning
  - inversion
  - signature files
    - clustering
    - information filtering and LSI

Signature files

- idea: ‘quick & dirty’ filter

  ![Signature files diagram]

- then, do seq. scan on sign. file and discard ‘false alarms’

- Adv.: easy insertions; faster than seq. scan
- Disadv.: O(N) search (with small constant)

Q: how to extract signatures?
Signature files

- A: superimposed coding!! [Mooers49], ...

Word | Signature
-----|----------
data  | 001 000 110 010
base  | 000 010 101 001
doc.signature | 001 010 111 011

m (=4 bits/word)
F (=12 bits sign. size)

---

Signature files

- A: superimposed coding!! [Mooers49], ...

Word | Signature
-----|----------
data  | 001 000 110 010
base  | 000 010 101 001
doc.signature | 001 010 111 011
data | ↑ ↑ ↑

actual match

---

Signature files

- A: superimposed coding!! [Mooers49], ...

Word | Signature
-----|----------
data  | 001 000 110 010
base  | 000 010 101 001
doc.signature | 001 010 111 011
retrieval | ↑ ↑ ↑

actual dismissal
Signature files

- A: superimposed coding!! [Mooers49], ...

<table>
<thead>
<tr>
<th>Word</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>001 000 110 010</td>
</tr>
<tr>
<td>base</td>
<td>000 010 101 001</td>
</tr>
<tr>
<td>doc.signature</td>
<td>001 010 111 011</td>
</tr>
</tbody>
</table>

false alarm ('false drop')

• Q1: How to choose $F$ and $m$?
• Q2: Why is it called ‘false drop’?
• Q3: other apps of signature files?
Signature files

• Q1: How to choose $F$ and $m$?

<table>
<thead>
<tr>
<th>Word</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>001 000 110 010</td>
</tr>
<tr>
<td>base</td>
<td>000 010 101 001</td>
</tr>
<tr>
<td>doc.signature</td>
<td>001 010 111 011</td>
</tr>
</tbody>
</table>

$F$ (=12 bits sign. size)
$m$ (=4 bits/word)

• Q2: Why is it called ‘false drop’?

• Q3: other apps of signature files?
Signature files

• Q2: Why is it called ‘false drop’?
• Old, but fascinating story [1949]
  – how to find qualifying books (by title word, and/or author, and/or keyword)
  – in O(1) time?
  – without computers

• Solution: edge-notched cards

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  • each title word is mapped to m numbers (how?)
  • and the corresponding holes are cut out:

• Solution: edge-notched cards

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
  |   |   | data

  ‘data’ -> #1, #39
Signature files

• Search, e.g., for ‘data’: activate needle #1, #39, and shake the stack of cards!

1 2 3 ...

40

data

‘data’ -> #1, #39

Signature files

• Also known as ‘zatocoding’, from ‘Zator’ company.

Signature files

• Q1: How to choose $F$ and $m$ ?
• Q2: Why is it called ‘false drop’?
• Q3: other apps of signature files?
Signature files

• Q3: other apps of signature files?
• A: anything that has to do with ‘membership testing’: does ‘data’ belong to the set of words of the document?

<table>
<thead>
<tr>
<th>Word</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>001 000 110 010</td>
</tr>
<tr>
<td>base</td>
<td>000 010 101 001</td>
</tr>
<tr>
<td>doc.signature</td>
<td>001 010 111 011</td>
</tr>
</tbody>
</table>

Signature files

• UNIX’s early ‘spell’ system [McIlroy]
• Bloom-joins in System R* [Mackert+] and ‘active disks’ [Riedel99]
• differential files [Severance+Lohman]

App#1: Unix’s spell

Dictionary

~30,000 words

aaron
apple
...

zoo

? electroencephalogram

What to do if the dictionary does not fit in memory (~1980)?
App#1: Unix’s spell

A: allow for a few typos! And use
• huge bit string ($2^{27}$)
• Hash each dictionary word to a bit
• Compress the string

Sub-questions:
• Q1: How often do we allow typos?
• Q2: Will the (compressed) bit string fit in memory?
• Q3: How to compress the bit string?

App#2: Bloom-joins

<table>
<thead>
<tr>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S</th>
<th>A</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

R join S (@PIT)
App#2: Bloom-joins

Idea: reduce transmission cost: ‘R semijoin S’

That is,
- ‘S’ ships its unique values of ‘A’
- ‘R’ deletes non-matching tuples
- (and they both send their tuples to PIT)

Q: what if we want to send at most, say 100 bytes NY -> Chicago?

A: Bloom-join! Send a bloom filter of the S.A values
**App#3: Differential files**

**Problem definition:**
- A large file (e.g., with EMPLOYEE records), nicely packed and organized (e.g., B-tree)
- A few insertions/deletions, that we would like to keep separate, and merge, at night
- How to search, e.g., for employee #123?

---

**Differential file**

<table>
<thead>
<tr>
<th>ssn</th>
<th>name</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>503</td>
<td></td>
<td></td>
</tr>
<tr>
<td>509</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Q:** How to search, e.g., for employee #123?

**A:** bloom-filter, for keys of diff. file

---

**Q:** What are the advantages of differential files?

**A:** <see paper, for 10(!) of them>
Signature files - conclusions

- easy insertions; slower than inversion
- brilliant idea of 'quick and dirty' filter:
  quickly discard the vast majority of non-qualifying elements, and focus on the rest.

References


References

References - cont’d


References - cont’d

• Mackert, L. M. and G. M. Lohman (August 1986). R* Optimizer Validation and Performance Evaluation for Distributed Queries. Proc. of 12th Int. Conf. on Very Large Data Bases (VLDB), Kyoto, Japan.


References - cont’d


References - cont’d