15-826: Multimedia Databases and Data Mining

Text - part I
C. Faloutsos

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

Problem - Motivation
• Eg., find documents containing “data”, “retrieval”
• Applications:

Problem - Motivation
• Eg., find documents containing “data”, “retrieval”
• Applications:
  – Web
  – law + patent offices
  – digital libraries
  – information filtering
Problem - Motivation

• Types of queries:
  – boolean ('data' AND 'retrieval' AND NOT ...)
  – additional features ('data' ADJACENT 'retrieval')
  – keyword queries ('data', 'retrieval')

• How to search a large collection of documents?

Full-text scanning

• Build a FSA; scan

  ABRACADABRA      text
  CAB              pattern

• for single term:
  – (naive: O(N*M))
  – Knuth Morris and Pratt (~'77)
    • build a small FSA; visit every text letter once only,
      by carefully shifting more than one step

  ABRACADABRA      text
  CAB              pattern
**Full-text scanning**

- for single term:
  - (naive: $O(N \times M)$)
  - Knuth Morris and Pratt (’77)
  - Boyer and Moore (’77)
    - preprocess pattern; start from right to left & skip!

ABRACADABRA    text
CAB            pattern

---

**Full-text scanning**

ABRACADABRA    text
CAB          pattern

---

**Full-text scanning**

ABRACADABRA    text
OMINOUS     pattern

Boyer+Moore: fastest, in practice
Sunday (‘90): some improvements

---

**Full-text scanning**

- For multiple terms (w/o “don’t care” characters): Aho+Corasic (’75)
  - again, build a simplified FSA in $O(M)$ time
- Probabilistic algorithms: ‘fingerprints’
  (Karp + Rabin ‘87)

---

**Full-text scanning**

- Approximate matching - string editing
distance:
  $d(‘survey’, ‘surgery’) = 2$
  = min # of insertions, deletions, substitutions to transform the first string into the second
SURVEY
SURGERY
**Full-text scanning**

- **string editing** distance - how to compute?
- A: dynamic programming
  
  \[
  \text{cost}(i, j) = \text{cost}(i-1, j-1)
  \]

  else

  \[
  \text{cost}(i, j) = \min (\begin{array}{l}
  1 + \text{cost}(i, j-1) // \text{deletion} \\
  1 + \text{cost}(i-1, j-1) // \text{substitution} \\
  1 + \text{cost}(i-1, j) // \text{insertion}
  \end{array})
  \]

---

**String editing distance**

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>U</th>
<th>R</th>
<th>V</th>
<th>E</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>φ</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>φ</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**String editing distance**

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>U</th>
<th>R</th>
<th>V</th>
<th>E</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**String editing distance**

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>U</th>
<th>R</th>
<th>V</th>
<th>E</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**String editing distance**

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>U</th>
<th>R</th>
<th>V</th>
<th>E</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**String editing distance**

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>U</th>
<th>R</th>
<th>V</th>
<th>E</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>S</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>U</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>R</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>R</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Y</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

---

**Full-text scanning**

Complexity: $O(M^N)$ (when using a matrix to ‘memoize’ partial results)
Full-text scanning

Conclusions:
• Full text scanning needs no space overhead, but is slow for large datasets