15-826: Multimedia Databases and Data Mining

Lecture #18 – SVD part II: case studies
C. Faloutsos

Must-read Material

• MM Textbook Appendix D

Outline

Goal: ‘Find similar / interesting things’
• Intro to DB
  • Indexing - similarity search
  • Data Mining

Copyright: C. Faloutsos (2013)
Indexing - Detailed outline

- primary key indexing
- secondary key / multi-key indexing
- spatial access methods
- fractals
- text
- Singular Value Decomposition (SVD)
  - multimedia
  - ...

SVD - Detailed outline

- Motivation
- Definition - properties
- Interpretation
- Complexity
- Case studies
  - SVD properties
  - Conclusions

SVD - Case studies

- multi-lingual IR; LSI queries
- compression
- PCA - ‘ratio rules’
- Karhunen-Loewe transform
- query feedbacks
- google/Kleinberg algorithms
Case study - LSI

Q1: How to do queries with LSI?
Q2: multi-lingual IR (english query, on spanish text?)

Problem: Eg., find documents with 'data'

\[ \begin{bmatrix}
1 & 1 & 0 & 0 \\
2 & 2 & 0 & 0 \\
1 & 1 & 0 & 0 \\
5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \times \begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27 \\
\end{bmatrix} \times \begin{bmatrix}
9.64 & 0 & 5.29 \\
0 & 0.58 & 0.58 \\
0 & 0 & 0.71 & 0.71 \\
\end{bmatrix} \]

A: map query vectors into ‘concept space’ – how?
Case study - LSI

Q1: How to do queries with LSI?
A: map query vectors into ‘concept space’ – how?

A: inner product (cosine similarity) with each ‘concept’ vector \( v_i \)

q = \begin{bmatrix} data & brain & lung \\ \top & 0 & 0 & 0 \end{bmatrix}

v1

v2

q o v1

v1

v2

term1

Case study - LSI

compactly, we have:

\[ q \mathbf{V} = q_{\text{concept}} \]

Eg:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.58 & 0 \\
0.58 & 0 \\
0 & 0.71 \\
0 & 0.71
\end{bmatrix}
\]

CS-concept

\[
\begin{bmatrix}
0.58 & 0
\end{bmatrix}
\]

term-to-concept similarities

Case study - LSI

Drill: how would the document (‘information’, ‘retrieval’) be handled by LSI? A: SAME:

\[ d_{\text{concept}} = d \mathbf{V} \]

Eg:

\[
\begin{bmatrix}
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.58 & 0 \\
0.58 & 0 \\
0.58 & 0 \\
0 & 0.71 \\
0 & 0.71
\end{bmatrix}
\]

CS-concept

\[
\begin{bmatrix}
1.16 & 0
\end{bmatrix}
\]

term-to-concept similarities
Case study - LSI

Observation: document (‘information’, ‘retrieval’) will be retrieved by query (‘data’), although it does not contain ‘data’!!

\[
\begin{align*}
\text{CS-concept} \quad &\quad \text{retrieval} \\
\text{data} \quad &\quad \text{brain} \\
\text{lung} \\
\end{align*}
\]

\[
\begin{bmatrix}
0 & 1 & 1 & 0 & 0
\end{bmatrix} \quad \begin{bmatrix}
1.16 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0
\end{bmatrix} \quad \begin{bmatrix}
0.58 & 0
\end{bmatrix}
\]

Q1: How to do queries with LSI?
Q2: multi-lingual IR (english query, on spanish text?)

Case study - LSI

• Problem:
  – given many documents, translated to both languages (eg., English and Spanish)
  – answer queries across languages
Case study - LSI

- Solution: ~ LSI

<table>
<thead>
<tr>
<th>CS</th>
<th>MD</th>
</tr>
</thead>
<tbody>
<tr>
<td>inf</td>
<td>data</td>
</tr>
<tr>
<td>brain</td>
<td>lung</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>1 1 1 0 0</td>
</tr>
<tr>
<td>2 2 0 0</td>
<td>1 2 2 0 0</td>
</tr>
<tr>
<td>1 1 1 0 0</td>
<td>1 1 1 0 0</td>
</tr>
<tr>
<td>5 5 0 0 0</td>
<td>5 5 4 0 0</td>
</tr>
<tr>
<td>0 0 0 2 2</td>
<td>0 0 0 2 2</td>
</tr>
<tr>
<td>0 0 0 3 3</td>
<td>0 0 0 2 3</td>
</tr>
<tr>
<td>0 0 0 1 1</td>
<td>0 0 0 1 1</td>
</tr>
</tbody>
</table>

SVD - Case studies

- multi-lingual IR; LSI queries
- compression
- PCA - ‘ratio rules’ & visualization
- Karhunen-Lowe transform
- query feedbacks
- google/Kleinberg algorithms

Case study: compression

[Korn+97]

Problem:

- given a matrix
- compress it, but maintain ‘random access’

(surprisingly, its solution leads to data mining and visualization...)
**Problem - specs**

- ~10^6 rows; ~10^3 columns; no updates;
- random access to any cell(s) ; small error: OK

<table>
<thead>
<tr>
<th>company</th>
<th>W0</th>
<th>W1</th>
<th>T0</th>
<th>T1</th>
<th>T0</th>
<th>T1</th>
<th>S0</th>
<th>S1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC Inc.</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEF Inc.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GHI Inc.</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XYZ Co.</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sailboat</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sailboat</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Idea**

- SVD - reminder
  - space savings: 2:1
  - minimum RMS error

- first singular vector
Case study: compression

- Outliers?
  - A: treat separately (SVD with ‘Deltas’)

First singular vector

Day 1

Compression - Performance

- 3 pass algo (→ scalability) (HOW?)
- Random cell(s) reconstruction
- 10:1 compression with < 2% error

Performance - scaleup
Compression - Visualization

- no Gaussian clusters; Zipf-like distribution

SVD - Case studies

- multi-lingual IR; LSI queries
- compression
- PCA - ‘ratio rules’
- Karhunen-Lowe transform
- query feedbacks
- google/Kleinberg algorithms

PCA - ‘Ratio Rules’

[Korn+00] Typically: ‘Association Rules’ (eg.,
{bread, milk} -> {butter}
But, can we discover more details? like:
$\text{bread} : \text{milk} : \text{butter} \sim \$2 : \$4 : \$3
PCA - ‘Ratio Rules’

Idea: try to find ‘concepts’:
• singular vectors dictate rules about ratios:
  bread:milk:butter = 2:4:3

\[ \text{\$ on butter} \]
\[ \text{\$ on milk} \]
\[ \text{\$ on bread} \]

PCA - ‘Ratio Rules’

Identical to PCA = Principal Components Analysis

\( \checkmark \) – Q1: which set of rules is ‘better’?
\( \checkmark \) – Q2: how to reconstruct missing/corrupted values?
\( \checkmark \) – Q3: is there need for binary/bucketized values? NO
\( \checkmark \) – Q4: how to interpret the rules (= ‘principal components’)?

PCA - Ratio Rules

NBA dataset
~500 players;
~30 attributes
PCA - Ratio Rules

- PCA: get singular vectors v1, v2, ...
- ignore entries with small abs. value
- try to interpret the rest

NBA dataset - V matrix (term to ‘concept’ similarities)

<table>
<thead>
<tr>
<th>field</th>
<th>RR1</th>
<th>RR2</th>
<th>RR3</th>
</tr>
</thead>
<tbody>
<tr>
<td>minutes played</td>
<td>.808</td>
<td></td>
<td></td>
</tr>
<tr>
<td>field goals</td>
<td>.103</td>
<td>.103</td>
<td></td>
</tr>
<tr>
<td>goal attempts</td>
<td>-.089</td>
<td>.502</td>
<td></td>
</tr>
<tr>
<td>points</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>total rebounds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>assists</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>steals</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ratio Rules - example

- RR1: minutes:points = 2:1
- corresponding concept?
Ratio Rules - example

• RR1: minutes:points = 2:1
• corresponding concept?

• A: ‘goodness’ of player
Ratio Rules - example

• RR2: points: rebounds negatively correlated (!)

<table>
<thead>
<tr>
<th>field</th>
<th>RR2</th>
<th>RR2</th>
<th>RR2</th>
</tr>
</thead>
<tbody>
<tr>
<td>minutes played</td>
<td>.893</td>
<td>-.4</td>
<td></td>
</tr>
<tr>
<td>field goals</td>
<td>.106</td>
<td>.499</td>
<td></td>
</tr>
<tr>
<td>goal attempts</td>
<td>-.289</td>
<td>.502</td>
<td></td>
</tr>
<tr>
<td>points</td>
<td>-.481</td>
<td>-.481</td>
<td></td>
</tr>
<tr>
<td>total rebounds</td>
<td>-.67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>assists</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>steals</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• RR2: points: rebounds negatively correlated (!) - concept?

• A: position: offensive/defensive
SVD - Case studies

- multi-lingual IR; LSI queries
- compression
- PCA - ‘ratio rules’ & visualization
- Karhunen-Loeve transform
- query feedbacks
- google/Kleinberg algorithms

K-L transform

[Duda & Hart]; [Fukunaga]

A subtle point:
SVD will give vectors that go through the origin

K-L transform

A subtle point:
SVD will give vectors that go through the origin
Q: how to find $v_1'$?
K-L transform

A subtle point:
SVD will give vectors that go through the origin
Q: how to find $v_1'$?

A: ‘centered’ PCA, i.e., move the origin to center of gravity

• How to ‘center’ a set of vectors (= data matrix)?
• What is the covariance matrix?
• A: see textbook
• (‘whitening transformation’)

K-L transform

A subtle point:
SVD will give vectors that go through the origin
Q: how to find $v_1'$?

A: ‘centered’ PCA, i.e., move the origin to center of gravity and THEN do SVD

K-L transform
Conclusions

- SVD: popular for dimensionality reduction / compression
- SVD is the ‘engine under the hood’ for PCA (principal component analysis)
- ... as well as the Karhunen-Loeve transform
- (and there is more to come ...)

References


References

References