15-826: Multimedia Databases and Data Mining

Lecture #19: SVD - part II (case studies)

C. Faloutsos

Must-read Material

- Textbook Appendix D

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
- 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-Lowe 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’

\[
\begin{bmatrix}
\text{data} \\
\text{retrieval} \\
\text{brain} \\
\text{lung}
\end{bmatrix}
= \begin{bmatrix}
1 & 1 & 0 & 0 \\
2 & 2 & 0 & 0 \\
1 & 1 & 0 & 0 \\
5 & 5 & 0 & 0
\end{bmatrix}
\times
\begin{bmatrix}
0.18 & 0.35 & 0.18 & 0.09 \\
0.03 & 0.53 & 0.80 & 0.27 \\
9.64 & 0 & 5.29 & x \\
0.58 & 0.58 & 0.58 & 0.71 & 0.71
\end{bmatrix}
\times
\begin{bmatrix}
0.18 \\
0.35 \\
0.18 \\
0.09
\end{bmatrix}
\]

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?

\[
q = \begin{bmatrix}
\text{data} & \text{inf} & \text{brain} & \text{lung} \\
1 & 0 & 0 & 0
\end{bmatrix}
\]

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

\[
v_1 \cdot v_2 = \cos(q, v_1)
\]
Case study - LSI

compactly, we have:

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

Eg:

\[
\begin{bmatrix}
\text{data} & \text{retrieval}
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0.58 & 0 & 0
\end{bmatrix}
= \begin{bmatrix}
0.58 \\
0.58
\end{bmatrix}
\]

TERM TO CONCEPT

similarities

CS-concept

Case study - LSI

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

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

Eg:

\[
\begin{bmatrix}
\text{data} & \text{retrieval} & \text{brain} & \text{lung}
\end{bmatrix}
\begin{bmatrix}
0 & 0.58 & 0.58 & 0 \\
1 & 0.58 & 0.58 & 0
\end{bmatrix}
= \begin{bmatrix}
1.16 \\
0
\end{bmatrix}
\]

TERM TO CONCEPT

similarities

CS-concept
Case study - LSI

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

CS-concept

\[ d = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \end{bmatrix} \quad q = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \]

\[ \begin{bmatrix} 1.16 \ 0 \end{bmatrix} \quad \begin{bmatrix} 0.58 \ 0 \end{bmatrix} \]

Case study - LSI

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

| retrieval | data
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>inf</td>
<td>brain</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>1 1 0 0</td>
</tr>
<tr>
<td>2 2 0 0</td>
<td>1 2 0 0</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>1 1 1 0</td>
</tr>
<tr>
<td>5 5 0 0</td>
<td>5 5 4 0</td>
</tr>
<tr>
<td>0 0 2 2</td>
<td>0 0 2 2</td>
</tr>
<tr>
<td>0 0 0 3</td>
<td>0 0 2 3</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>0 0 1 1</td>
</tr>
</tbody>
</table>

SVD - Case studies

- multi-lingual IR; LSI queries
- compression
- PCA - ‘ratio rules’
- 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>Source</th>
<th>7/3/90</th>
<th>7/13/90</th>
<th>7/23/90</th>
<th>7/33/90</th>
<th>7/43/90</th>
<th>7/53/90</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC Inc.</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEF Inc.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRU Inc.</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KLM Co.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sapphire</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Czar Jewel</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

---

Idea

SVD - reminder

• space savings: 2:1
• minimum RMS error
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

\[ \text{Sample data distribution} \]

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:
- which set of rules is ‘better’?
- how to reconstruct missing/corrupted values?
- need binary/bucketized values
PCA - ‘Ratio Rules’

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

\$ on butter

\$ on milk

\$ on bread

Q1: which set of rules is ‘better’?
Q2: how to reconstruct missing/corrupted values?
Q3: is there need for binary/bucketized values?
Q4: how to interpret the rules (= ‘principal components’)?

Q2: how to reconstruct missing/corrupted values?

Eg:
• rule: bread:milk = 3:4
• a customer spent $6 on bread - how about milk?
PCA - ‘Ratio Rules’

pictorially:

harder cases: overspecified/underspecified

over-specified:
• milk:bread:butter = 1:2:3
• a customer got
  • $2 bread and $4 milk
  • how much milk?

Answer: minimize distance between ‘feasible’ and ‘expected’ values (using SVD...)

harder cases: underspecified
PCA - ‘Ratio Rules’

bottom line: we can reconstruct any count of missing values
This is very useful:
• can spot outliers (how?)
• can measure the ‘goodness’ of a set of rules (how?)

Identical to PCA = Principal Components Analysis

→ Q1: which set of rules is ‘better’?
✓ Q2: how to reconstruct missing/corrupted values?
→ Q3: is there need for binary/bucketized values?
→ Q4: how to interpret the rules (= ‘principal components’)?

• Q1: which set of rules is ‘better’?
• A: the ones that needs the fewest outliers:
  – pretend we don’t know a value (eg., $ of ‘Smith’ on ‘bread’)
  – reconstruct it
  – and sum up the squared errors, for all our entries
• (other answers are also reasonable)
PCA - ‘Ratio Rules’

Identical to PCA = Principal Components Analysis

- Q1: which set of rules is ‘better’?
- Q2: how to reconstruct missing/corrupted values?
- Q3: is there need for binary/bucketized values?
- Q4: how to interpret the rules (= ‘principal components’)?

PCA - ‘Ratio Rules’

Identical to PCA = Principal Components Analysis

- Q1: which set of rules is ‘better’?
- Q2: how to reconstruct missing/corrupted values?
- Q3: is there need for binary/bucketized values? NO
- 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

PCA - Ratio Rules

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>.508</td>
<td>- .4</td>
<td></td>
</tr>
<tr>
<td>field goals</td>
<td>.106</td>
<td>.198</td>
<td>.502</td>
</tr>
<tr>
<td>goal attempts</td>
<td>- .399</td>
<td>- .485</td>
<td>- .485</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>RR1</th>
<th>RR2</th>
<th>RR3</th>
</tr>
</thead>
<tbody>
<tr>
<td>minutes played</td>
<td>.899</td>
<td></td>
<td></td>
</tr>
<tr>
<td>field goals</td>
<td>-.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>goal attempts</td>
<td>.199</td>
<td>.502</td>
<td>.97</td>
</tr>
<tr>
<td>points</td>
<td>-.489</td>
<td></td>
<td></td>
</tr>
<tr>
<td>total rebounds</td>
<td>-.485</td>
<td></td>
<td></td>
</tr>
<tr>
<td>turnover</td>
<td>-.61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>steals</td>
<td>-.67</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’
- 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

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, ie., move the origin to center of gravity

---

**K-L transform**

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

Q: how to find $v_1'$?

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

---

**K-L transform**

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

---
Conclusions

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

References


References

References
