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'

\[
\begin{array}{c}
\text{CS} \\
\text{MD}
\end{array}
\begin{bmatrix}
1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 \\
1 & 1 & 1 & 0 \\
5 & 5 & 5 & 0 \\
0 & 0 & 0 & 2 \\
0 & 0 & 0 & 3 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
0.18 & 0 & 0.36 & 0 \\
0.18 & 0 & 0.90 & 0 \\
0 & 0.53 & 0 & 0.80 \\
0 & 0.27 & 0 & 0.71 \\
\end{bmatrix}
\begin{bmatrix}
9.64 & 0 & 0 \\
0 & 0 & 5.29 \\
\end{bmatrix}
\begin{bmatrix}
9.58 & 0.58 & 0.58 & 0 \\
0 & 0 & 0 & 0.71 \\
0 & 0 & 0 & 0.71 \\
\end{bmatrix}
\]

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

A: inner product (cosine similarity) with each 'concept' vector \( v_i \)
**Case study - LSI**

Compactly, we have:

\[ q_{\text{concept}} = q^T V \]

Eg:

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

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

\[
\text{CS-concept} = \begin{bmatrix}
1 & 0.58 & 0
\end{bmatrix}
\]

**Case study - LSI**

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

**Case study - LSI**

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

\[ d_{\text{concept}} = d^T V \]

Eg:

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

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

\[
\text{CS-concept} = \begin{bmatrix}
1 & 1.16 & 0
\end{bmatrix}
\]
Case study - LSI

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

\[
\begin{array}{c}
\text{brain} \\
\text{lung} \\
\text{data} \\
\text{retrieval}
\end{array}
\]

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

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

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

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

Q1: How to do queries with LSI?

Q2: multi-lingual IR (english query, on spanish text?)

• 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></th>
<th>data</th>
<th>inf</th>
<th>brain</th>
<th>lung</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MD</td>
<td></td>
<td></td>
<td></td>
<td></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

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

Idea
SVD - reminder

- space savings: 2:1
- minimum RMS error

Case study: compression

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

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-Loewe 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

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

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’

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...)
PCA - ‘Ratio Rules’

harder cases: underspecified

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’)?
PCA - ‘Ratio Rules’

• 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)

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’)?

⇒ – Q3: is there need for binary/bucketized values? NO
⇒ – Q4: how to interpret the rules (= ‘principal components’)?

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 - \( V \) matrix (term to 'concept' similarities)

\[
\begin{array}{c|ccc}
\text{term} & \text{field goals} & \text{rebounds} & \text{points} \\
\hline
\text{field goals} & 0.84 & 0.29 & 0.29 \\
\text{rebounds} & -0.49 & 0.69 & 0.69 \\
\text{points} & -0.36 & 0.67 & 0.67 \\
\end{array}
\]

\( v_1 \)

• PCA: get singular vectors \( v_1, v_2, \ldots \)
• ignore entries with small abs. value
• try to interpret the rest
Ratio Rules - example

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

[Diagram showing scatter plot with points labeled v1]
Ratio Rules - example

- RR1: minutes:points = 2:1
- corresponding concept?
- A: ‘goodness’ of player

Ratio Rules - example

- RR2: points: rebounds negatively correlated(!)

Ratio Rules - example

- RR2: points: rebounds negatively correlated(!) - concept?
Ratio Rules - example

• RR2: points: rebounds negatively correlated(!) - concept?
• A: position: offensive/defensive

SVD - Case studies

• multi-lingual IR; LSI queries
• compression
• PCA - ‘ratio rules’
  • Karhunen-Lowe 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 $v_1^*$?

A: ‘centered’ PCA, i.e., 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-Loeve transform
• (and there is more to come ...)

References

References