Data Mining and Matrices
01 – Introduction

Prof. Dr. Rainer Gemulla

Universität Mannheim

HWS 2015
Outline

1. What is data mining?

2. What is a matrix?

3. Why data mining and matrices?

4. Summary
Outline

1. What is data mining?

2. What is a matrix?

3. Why data mining and matrices?

4. Summary
What is data mining?

“Data mining is the process of discovering knowledge or patterns from massive amounts of data.” (Encyclopedia of Database Systems)

Estimated $100 billion industry around managing and analyzing data.
What is data mining?

“What data mining is the process of discovering knowledge or patterns from massive amounts of data.” (Encyclopedia of Database Systems)

- Science
  - The Sloan Digital Sky Survey gathered 140TB of information
  - NASA Center for Climate Simulation stores 32PB of data
  - 3B base pairs exist in the human genome
  - LHC registers 600M particle collisions per second, 25PB/year

- Social/business data
  - 1M customer transactions are performed at Walmart per hour
  - 25M Netflix customers view and rate hundreds of thousands of movies
  - 40B photos have been uploaded to Facebook
  - 200M active Twitter users write 400M tweets per day
  - 4.6B mobile-phone subscriptions worldwide

- Government, health care, news, stocks, books, web search, ...
What is data mining?

“What data mining is the process of discovering knowledge or patterns from massive amounts of data.” (Encyclopedia of Database Systems)

Prediction

Clustering

Outlier detection

“Regnet es am Siebenschläfertag, der Regen sieben Wochen nicht weichen mag.” (German folklore)

Pattern mining
What is data mining?

“Data mining is the **process** of discovering knowledge or patterns from massive amounts of data.” (Encyclopedia of Database Systems)
What is a matrix?
Womb

- *mater* (Latin) = mother
- *matrix* (Latin) = pregnant animal
- *matrix* (Late Latin) = womb also source, origin
- Since 1550s: place or medium where something is developed
- Since 1640s: embedding or enclosing mass
Rectangular arrays of numbers

- “Rectangular arrays” known in ancient China (rod calculus, estimated as early as 300BC)

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

- Term “matrix” coined by J.J. Sylvester in 1850
System of linear equations

Three sheafs of a good crop, two sheafs of a mediocre crop, and one sheaf of a bad crop are sold for 39 dou. Two sheafs of good, three mediocre, and one bad are sold for 34 dou; and one good, two mediocre, and three bad are sold for 26 dou. What is the price received for each sheaf of a good crop, each sheaf of a mediocre crop, and each sheaf of a bad crop? Chiu-chang Suan-shu (Nine Chapters on Arithmetic), \( \approx 200BC \)

- Systems of linear equations can be written as matrices

\[
\begin{align*}
3x + 2y + z &= 39 \\
2x + 3y + z &= 34 \\
x + 2y + 3z &= 26
\end{align*}
\]

\[
\begin{pmatrix}
3 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 2 & 3
\end{pmatrix}
\begin{pmatrix}
x \\ y \\ z
\end{pmatrix}
= 
\begin{pmatrix}
39 \\ 34 \\ 26
\end{pmatrix}
\]

- and then be solved using linear algebra methods

\[
\begin{pmatrix}
3 & 2 & 1 \\ 5 & 1 & 24 \\ 12 & 33
\end{pmatrix}
\begin{pmatrix}
x \\ y \\ z
\end{pmatrix}
= 
\begin{pmatrix}
9.25 \\ 4.25 \\ 2.75
\end{pmatrix}
\]
Set of data points

\[
\begin{pmatrix}
-3.84 & -2.21 \\
-3.33 & -2.19 \\
-2.55 & -1.47 \\
-2.46 & -1.25 \\
-1.49 & -0.76 \\
-1.67 & -0.39 \\
-1.3 & -0.59 \\
\vdots & \vdots \\
1.59 & 0.78 \\
1.53 & 1.02 \\
1.45 & 1.26 \\
1.86 & 1.18 \\
2.04 & 0.96 \\
2.42 & 1.24 \\
2.32 & 2.03 \\
2.9 & 1.35 \\
\end{pmatrix}
\]
Linear maps

- Linear maps from $\mathbb{R}^3$ to $\mathbb{R}$
  
  \[
  f_1(x, y, z) = 3x + 2y + z \\
  f_2(x, y, z) = 2x + 3y + z \\
  f_3(x, y, z) = x + 2y + 3z \\
  f_4(x, y, z) = x
  \]

- Linear map $f_1$ written as a matrix
  
  \[
  \begin{pmatrix}
  3 & 2 & 1 \\
  \end{pmatrix}
  \begin{pmatrix}
  x \\
  y \\
  z \\
  \end{pmatrix}
  = f_1(x, y, z)
  \]

- Linear map from $\mathbb{R}^3$ to $\mathbb{R}^4$
  
  \[
  \begin{pmatrix}
  3 & 2 & 1 \\
  2 & 3 & 1 \\
  1 & 2 & 3 \\
  1 & 0 & 0 \\
  \end{pmatrix}
  \begin{pmatrix}
  x \\
  y \\
  z \\
  \end{pmatrix}
  = \begin{pmatrix}
  f_1(x, y, z) \\
  f_2(x, y, z) \\
  f_3(x, y, z) \\
  f_4(x, y, z) \\
  \end{pmatrix}
  \]
Graphs

Adjacency matrix
Objects and attributes

Anna, Bob, and Charlie went shopping
- Anna bought butter and bread
- Bob bought butter, bread, and beer
- Charlie bought bread and beer

<table>
<thead>
<tr>
<th></th>
<th>Bread</th>
<th>Butter</th>
<th>Beer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anna</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Customer transactions

<table>
<thead>
<tr>
<th></th>
<th>Data</th>
<th>Matrix</th>
<th>Mining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book 1</td>
<td>5</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Book 2</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Book 3</td>
<td>4</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

Document-term matrix

<table>
<thead>
<tr>
<th></th>
<th>Avatar</th>
<th>The Matrix</th>
<th>Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Bob</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Charlie</td>
<td>5</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Incomplete rating matrix

<table>
<thead>
<tr>
<th></th>
<th>Jan</th>
<th>Jun</th>
<th>Sep</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saarbrücken</td>
<td>−1</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>Helsinki</td>
<td>−6.5</td>
<td>10.9</td>
<td>8.7</td>
</tr>
<tr>
<td>Cape Town</td>
<td>15.7</td>
<td>7.8</td>
<td>8.7</td>
</tr>
</tbody>
</table>

Cities and monthly temperatures

Many different kinds of data fit this object-attribute viewpoint.
What is a matrix?

- A means to describe *computation*
  - Rotation
  - Rescaling
  - Permutation
  - Projection
  - ...

- A means to describe *data*

<table>
<thead>
<tr>
<th>Rows</th>
<th>Columns</th>
<th>Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
<td>Attributes</td>
<td>Values</td>
</tr>
<tr>
<td>Equations</td>
<td>Variables</td>
<td>Coefficients</td>
</tr>
<tr>
<td>Data points</td>
<td>Axes</td>
<td>Coordinates</td>
</tr>
<tr>
<td>Vertices</td>
<td>Vertices</td>
<td>Edges</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Matrix representation]

In data mining, we make use of both viewpoints simultaneously.
Outline

1. What is data mining?

2. What is a matrix?

3. Why data mining and matrices?

4. Summary
Key tool: Matrix decompositions

A matrix decomposition of a data matrix $D$ is given by three matrices $L$, $M$, $R$ such that

$$D = LMR,$$

where

- $D$ is an $m \times n$ data matrix,
- $L$ is an $m \times r_1$ matrix,
- $M$ is an $r_1 \times r_2$ matrix,
- $R$ is an $r_2 \times n$ matrix, and
- $r_1$ and $r_2$ are integers $\geq 1$.
- Often $r_1 = r_2 = r \geq 1$

$$d_{ij} = \sum_{k,k'} l_{ik} m_{kk'} r_{k'j}$$
Why matrix decompositions?

- Decompositions as just defined are not really helpful
  - Suppose we set $r = n$, $L = D$, $M = R = I_n$ (the $n \times n$ identity matrix)
  - Then $D = LMR = DI_nI_n = D$
  - But this does not provide insight

- To make decompositions useful, we want the decomposition to satisfy certain (carefully chosen) properties or constraints

- For example: we may want $r$ to be small
  - Each object is represented by $n$ numbers in $D$
  - Each object is represented by $r$ numbers in $L$
  - If $r < n$, we performed some form of compression

- Another example: we may want factors to have certain properties
  - Compare: integer factorization
  - $391 = 17 \cdot 13$
Approximate matrix decompositions

A **approximate matrix decomposition** of data matrix $D$ is given by three matrices $L$, $M$, $R$ such that

$$D \approx LMR = \hat{D},$$

where each matrix has conforming dimensions (as before).

- We often look at approximate decompositions
  - Data is noisy anyway
  - Approximation may remove noise
  - Allows to focus on global (small $r$) or local (large $r$) patterns
  - Often more insightful
  - More efficient to compute

- $\approx$ defined by some **loss function** $L(D, \hat{D})$
  - E.g., squared error
  - Low means good approximation, high means bad
  - Finding the best approximation (smallest loss) can be hard

- We often say “matrix decomposition” when we actually mean “approximate matrix decomposition”
Some matrix decompositions

- There are many different decompositions, each enforcing different constraints and serving a different purpose
  - Singular value decomposition (SVD)
  - $k$-means (crisp or fuzzy)
  - Non-negative matrix factorization (NMF)
  - Semi-discrete decomposition (SDD)
  - Boolean matrix decomposition (BMF)
  - Independent component analysis (ICA)
  - Matrix completion
  - Probabilistic matrix factorization
  - Tensor decompositions
  - ... 

- Picking the right one for the problem at hand is hard, experience helps

- Decompositions are not always easy (and often hard) to compute
Example: Singular value decomposition

\[
D_{50 \times 2}
\]

\[
L_{50 \times 2}
\]

\[
M_{2 \times 2}
\]

\[
R_{2 \times 2}
\]

\[
\begin{pmatrix}
11.73 & 0 \\
0 & 1.71
\end{pmatrix}
\]
Example: k-Means

VQ

L

×

R∗j

=

Original

D∗j

\[ VQ \text{ and } PCA \text{ learn to represent faces with a set of basis images resembling parts of faces. Unlike VQ and PCA, NMF learns to represent faces with a set of basis images resembling parts of faces.} \]
Example: Non-negative matrix factorization

\[ D_{*j} = \hat{D}_{*j} = L \times R_{*j} \]
**Example: Latent Dirichlet allocation**

<table>
<thead>
<tr>
<th>“Arts”</th>
<th>“Budgets”</th>
<th>“Children”</th>
<th>“Education”</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEW</td>
<td>MILLION</td>
<td>CHILDREN</td>
<td>SCHOOL</td>
</tr>
<tr>
<td>FILM</td>
<td>TAX</td>
<td>WOMEN</td>
<td>STUDENTS</td>
</tr>
<tr>
<td>SHOW</td>
<td>PROGRAM</td>
<td>PEOPLE</td>
<td>SCHOOLS</td>
</tr>
<tr>
<td>MUSIC</td>
<td>BUDGET</td>
<td>CHILD</td>
<td>EDUCATION</td>
</tr>
<tr>
<td>MOVIE</td>
<td>BILLION</td>
<td>YEARS</td>
<td>TEACHERS</td>
</tr>
<tr>
<td>PLAY</td>
<td>FEDERAL</td>
<td>FAMILIES</td>
<td>HIGH</td>
</tr>
<tr>
<td>MUSICAL</td>
<td>YEAR</td>
<td>WORK</td>
<td>PUBLIC</td>
</tr>
<tr>
<td>BEST</td>
<td>SPENDING</td>
<td>PARENTS</td>
<td>TEACHER</td>
</tr>
<tr>
<td>ACTOR</td>
<td>NEW</td>
<td>SAYS</td>
<td>BENNETT</td>
</tr>
<tr>
<td>FIRST</td>
<td>STATE</td>
<td>FAMILY</td>
<td>MANIGAT</td>
</tr>
<tr>
<td>YORK</td>
<td>PLAN</td>
<td>WELFARE</td>
<td>NAMPHY</td>
</tr>
<tr>
<td>OPERA</td>
<td>MONEY</td>
<td>MEN</td>
<td>STATE</td>
</tr>
<tr>
<td>THEATER</td>
<td>PROGRAMS</td>
<td>PERCENT</td>
<td>PRESIDENT</td>
</tr>
<tr>
<td>ACTRESS</td>
<td>GOVERNMENT</td>
<td>CARE</td>
<td>ELEMENTARY</td>
</tr>
<tr>
<td>LOVE</td>
<td>CONGRESS</td>
<td>LIFE</td>
<td>HAITI</td>
</tr>
</tbody>
</table>

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. “Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services,” Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center’s share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.
What can we do with matrix decompositions?

- Separate data from multiple processes
- Remove noise from the data
- Remove redundancy from the data
- Reveal latent structure and similarities in the data
- Cluster the data
- Fill in missing entries
- Find local patterns
- Reduce space consumption
- Reduce computational cost
- Aid visualization
- ...

Matrix decompositions can make data mining algorithms more effective. They may also provide insight into the data by themselves.
Factor interpretation of matrix decompositions

Assume that $M$ is diagonal. Consider object $i$.

- Row of $R = \text{part (or piece), called } \text{latent factor} \text{ ("latent object")}$
- Entry of $M = \text{weight of corresponding part}$
- Row of $MR = \text{weighted part}$
- Row of $L = \text{representation of the object in terms of the weighted parts (} r \text{ pieces of information)}$
- $r \text{ forces "compactness" (often } r < n)$

Each object can be viewed as a combination of $r \text{ (weighted) "latent objects" (or "prototypical objects")}. Similarly, each attribute can be viewed as a combination of $r \text{ (weighted) "latent attributes."}$

(e.g., latent attribute $= \text{"body size";} \text{ latent object relates body size to real attributes such as "height", \text{ "weight", \text{ "shoe size"}})$
Example: Weather data ($r = 1$)

<table>
<thead>
<tr>
<th></th>
<th>Jan</th>
<th>Apr</th>
<th>Jul</th>
<th>Oct</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>9.05</td>
<td>16.55</td>
<td>26.73</td>
<td>18.75</td>
<td>17.81</td>
</tr>
<tr>
<td>0.62</td>
<td>Stockholm</td>
<td>-0.70</td>
<td>8.60</td>
<td>21.90</td>
<td>9.90</td>
</tr>
<tr>
<td>0.69</td>
<td>Minsk</td>
<td>-2.10</td>
<td>12.20</td>
<td>23.60</td>
<td>10.20</td>
</tr>
<tr>
<td>0.83</td>
<td>London</td>
<td>7.90</td>
<td>13.30</td>
<td>22.80</td>
<td>15.20</td>
</tr>
<tr>
<td>0.90</td>
<td>Budapest</td>
<td>1.20</td>
<td>16.30</td>
<td>26.50</td>
<td>16.10</td>
</tr>
<tr>
<td>0.88</td>
<td>Paris</td>
<td>6.90</td>
<td>14.70</td>
<td>24.40</td>
<td>15.80</td>
</tr>
<tr>
<td>0.98</td>
<td>Bucharests</td>
<td>1.50</td>
<td>18.00</td>
<td>28.80</td>
<td>18.00</td>
</tr>
<tr>
<td>1.09</td>
<td>Barcelona</td>
<td>12.40</td>
<td>17.60</td>
<td>27.50</td>
<td>21.50</td>
</tr>
<tr>
<td>1.14</td>
<td>Rome</td>
<td>11.90</td>
<td>17.70</td>
<td>30.30</td>
<td>21.40</td>
</tr>
<tr>
<td>1.16</td>
<td>Lisbon</td>
<td>14.80</td>
<td>19.80</td>
<td>27.90</td>
<td>22.50</td>
</tr>
<tr>
<td>1.24</td>
<td>Athens</td>
<td>12.90</td>
<td>20.30</td>
<td>32.60</td>
<td>23.10</td>
</tr>
<tr>
<td>1.21</td>
<td>Valencia</td>
<td>16.10</td>
<td>20.20</td>
<td>29.10</td>
<td>23.60</td>
</tr>
<tr>
<td>1.27</td>
<td>Malta</td>
<td>16.10</td>
<td>20.00</td>
<td>31.50</td>
<td>25.20</td>
</tr>
</tbody>
</table>

\[ D \]
Example: Weather data \((r = 1)\), reconstruction

\[
\begin{array}{|c|c|c|c|c|}
\hline
\text{Location} & \text{Jan} & \text{Apr} & \text{Jul} & \text{Oct} & \text{Year} \\
\hline
\text{Stockholm} & 5.65 & 10.33 & 16.68 & 11.70 & 11.11 \\
\text{Minsk} & 6.21 & 11.36 & 18.35 & 12.87 & 12.23 \\
\text{London} & 7.55 & 13.80 & 22.28 & 15.63 & 14.85 \\
\text{Budapest} & 8.11 & 14.83 & 23.94 & 16.80 & 15.96 \\
\text{Paris} & 7.96 & 14.56 & 23.52 & 16.50 & 15.67 \\
\text{Bucharest} & 8.91 & 16.30 & 26.32 & 18.47 & 17.54 \\
\text{Barcelona} & 9.88 & 18.06 & 29.17 & 20.46 & 19.44 \\
\text{Rome} & 10.28 & 18.80 & 30.35 & 21.30 & 20.23 \\
\text{Lisbon} & 10.47 & 19.15 & 30.92 & 21.70 & 20.61 \\
\text{Athens} & 11.21 & 20.50 & 33.11 & 23.23 & 22.07 \\
\text{Valencia} & 10.92 & 19.96 & 32.24 & 22.62 & 21.48 \\
\text{Malta} & 11.47 & 20.98 & 33.88 & 23.77 & 22.58 \\
\hline
\end{array}
\]

\(\hat{D}\) (RMSE: 2.66)
Example: Weather data \( (r = 2) \), reconstruction

\[
\begin{bmatrix}
1.00 & 9.05 & 16.55 & 26.73 & 18.75 & 17.81 \\
1.00 & -4.14 & 0.27 & 2.32 & -0.89 & -0.69 \\
\end{bmatrix}
\]

<table>
<thead>
<tr>
<th></th>
<th>Jan</th>
<th>Apr</th>
<th>Jul</th>
<th>Oct</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stockholm</td>
<td>-1.34</td>
<td>10.79</td>
<td>20.59</td>
<td>10.20</td>
<td>9.95</td>
</tr>
<tr>
<td>Minsk</td>
<td>-2.52</td>
<td>11.94</td>
<td>23.23</td>
<td>10.99</td>
<td>10.77</td>
</tr>
<tr>
<td>London</td>
<td>7.54</td>
<td>13.80</td>
<td>22.28</td>
<td>15.63</td>
<td>14.85</td>
</tr>
<tr>
<td>Budapest</td>
<td>1.82</td>
<td>15.24</td>
<td>27.46</td>
<td>15.45</td>
<td>14.91</td>
</tr>
<tr>
<td>Paris</td>
<td>6.71</td>
<td>14.65</td>
<td>24.22</td>
<td>16.23</td>
<td>15.46</td>
</tr>
<tr>
<td>Bucharests</td>
<td>2.31</td>
<td>16.74</td>
<td>30.02</td>
<td>17.05</td>
<td>16.44</td>
</tr>
<tr>
<td>Barcelona</td>
<td>12.61</td>
<td>17.88</td>
<td>27.64</td>
<td>21.05</td>
<td>19.90</td>
</tr>
<tr>
<td>Rome</td>
<td>11.55</td>
<td>18.71</td>
<td>29.64</td>
<td>21.57</td>
<td>20.44</td>
</tr>
<tr>
<td>Lisbon</td>
<td>15.00</td>
<td>18.85</td>
<td>28.39</td>
<td>22.67</td>
<td>21.36</td>
</tr>
<tr>
<td>Athens</td>
<td>12.65</td>
<td>20.41</td>
<td>32.31</td>
<td>23.54</td>
<td>22.31</td>
</tr>
<tr>
<td>Malta</td>
<td>16.10</td>
<td>20.67</td>
<td>31.29</td>
<td>24.76</td>
<td>23.35</td>
</tr>
</tbody>
</table>

\( \hat{D} \)  

(RMSE: 0.60)
Example: Weather data \((r = 2)\), plot

![Weather data plot](image-url)
Example: Netflix prize data

(≈ 500k users, ≈ 17k movies, ≈ 100M ratings)
Other interpretations

- Geometric interpretation
  - Transformation of $n$-dimensional space in $r$-dimensional space
  - Row of $R = $ new axes
  - Row of $L = $ new coordinates

- Component interpretation
  - $D$ is viewed as consisting of $r$ layers (of same shape as $D$)
  - $k$-th layer described by $L^*k m_{kk} R_{k^*}$
  - $D = \sum_k L^*k m_{kk} R_{k^*}$

- Graph interpretation
  - $D$ is thought of as a bipartite graph with object and attribute vertexes
  - Edge weights measure association b/w objects and attributes
  - Decomposition thought of as a tripartite graph with row, waypoint, and column vertexes

All interpretations are useful (more later).
Outline

1. What is data mining?

2. What is a matrix?

3. Why data mining and matrices?

4. Summary
Lessons learned

- Data mining = from data to knowledge
  → Prediction, clustering, outlier detection, patterns

- Matrices are common representation of datasets
  → Linear maps, data points, sets, graphs, relational data, . . .

- Often: rows = objects, columns = attributes

- Matrix decompositions reveal structure in the data
  → $D \approx LMR$

- Many different decompositions with different applications
  → SVD, $k$-means, NMF, SDD, BMF, ICA, completion, ...

- Factor interpretation: objects described by “prototypical objects”
Suggested reading

- Skillicorn, Ch. 1: *Data Mining*
- Skillicorn, Ch. 2: *Matrix Decompositions*