The 8M Algorithm from Today’s Perspective

Radim Belohlavek, Martin Trnecka

DEPARTMENT OF COMPUTER SCIENCE
PALACKÝ UNIVERSITY OLOMOUC

CLA 2018
14th International Conference on
Concept Lattices and Their Applications
Olomouc, Czech Republic, June 12–14, 2018
Our Contributions

- Boolean matrix factorization (BMF)
- Current research = design of new factorization algorithms
- Present and analyze 8M method
  - unknown in present research on BMF
  - (first) complete description of the 8M algorithm
  - improvement of the 8M algorithm (8M+)
  - lessons performance of existing algorithms
**Boolean Matrix Factorization**

- **A general aim:** for a given matrix $I \in \{0, 1\}^{n \times m}$ find matrices $A \in \{0, 1\}^{n \times k}$ and $B \in \{0, 1\}^{k \times m}$ for which $I$ (approximately) equals $A \circ B$, $k$ reasonably small.

- $\circ$ is the Boolean matrix product

$$ (A \circ B)_{i,j} = \max_{l=1}^{k} \min(A_{il}, B_{lj}). $$

$$
\begin{pmatrix}
10111 \\
01101 \\
01001 \\
10110
\end{pmatrix}
\circ
\begin{pmatrix}
110 \\
011 \\
001 \\
100
\end{pmatrix}
= 
\begin{pmatrix}
10010 \\
00101 \\
01001
\end{pmatrix}
$$

- Various terminology and notation (including FCA)

- Factors = interesting patterns that help explain data
Error Measure

- $I$ (approximately) equals $A \circ B$
- Assessed by means of the metric $E(\cdot, \cdot)$

$$E(C, D) = \sum_{i,j=1}^{m,n} |C_{ij} - D_{ij}|.$$ 

- Two components of $E$

$$E(I, A \circ B) = E_u(I, A \circ B) + E_o(I, A \circ B),$$

where

$$E_u(I, A \circ B) = |\{\langle i, j \rangle ; I_{ij} = 1, (A \circ B)_{ij} = 0\}|,$$

$$E_o(I, A \circ B) = |\{\langle i, j \rangle ; I_{ij} = 0, (A \circ B)_{ij} = 1\}|.$$ 

- Non-symmetry of undercovering and overcovering error
8M

- Statistical software package known as BMDP
- Developed in 1960s at the University of California in Los Angeles (W. J. Dixon)
- Developed by: M. R. Mickey, L. Engelman and P. Mudle
- 8M method has been added to BMDP in the late 1970s
- Probably the oldest BMF method
- No longer available


- Incomplete description → several blindspots
- Partially black box analysis of 8M
Basic Idea of 8M

**Input:**
- \( I \in \{0, 1\}^{n \times m} \) ... Boolean matrix
- \( k \) ... number of desired factors
- \( \text{init} \) ... number of initial factors
- \( \text{cost} \) ... determines significance of overcovering

**Output:**
- \( A \in \{0, 1\}^{n \times k} \) and \( B \in \{0, 1\}^{k \times m} \)
Basic Idea of 8M: main procedure

**Algorithm 1: 8M**

\[ B \leftarrow \text{COMPUTEINITIALFACTORS}(\text{init}) \]
\[ A \leftarrow 0_{n \times \text{init}} \]
\[ f \leftarrow \text{init} \]
\[ \text{REFINEMATRICESAB}(A, B, I, \text{cost}) \]
\[ k\text{Reached} \leftarrow 0 \]

while \( k\text{Reached} < 2 \text{ or } I \leq A \circ B \) do

    foreach \( i, j \) do if \( I_{ij} > (A \circ B)_{ij} \) then \( \Delta^+_{ij} \leftarrow 1 \) else \( \Delta^+_{ij} \leftarrow 0 \)

    add column \( j \) of \( \Delta^+ \) with the largest count of 1s as new column to \( A \)
    add row of 0s as new row to \( B \) and set entry \( j \) of this row to 1
    \[ f \leftarrow f + 1 \]
    \[ \text{REFINEMATRICESAB}(A, B, I, \text{cost}) \]
    if another two new factors were added then
        remove column \( A_{(f-2)} \) from \( A \) and row \( B_{(f-2)} \) from \( B \)
        \[ f \leftarrow f - 1 \]
        \[ \text{REFINEMATRICESAB}(A, B, I, \text{cost}) \]
    if \( f = k \) then \( k\text{Reached} \leftarrow k\text{Reached} + 1 \)

return \( A, B \)
Basic Idea of 8M: refine matrices

**Algorithm 2:** \texttt{RefineMatricesAB}

\texttt{repeat}
\hspace{1em} \texttt{RefineMatrixA}(A, B, I, cost)
\hspace{1em} \texttt{RefineMatrixB}(A, B, I, cost)
\texttt{until loop executed 3 times or A and B did not change}

**Algorithm 3:** \texttt{RefineMatrixA}

\texttt{foreach row } \(i \in \{1, \ldots, n\} \text{ do}
\hspace{1em} y \leftarrow I_i; Z \leftarrow B; A_i \leftarrow 0
\texttt{repeat}
\hspace{2em} \texttt{foreach factor } l \in 1, \ldots, f \text{ do}
\hspace{3em} m_l \leftarrow \sum_{j=1}^{m} y_j \cdot Z_{lj} - \text{cost} \cdot \sum_{j=1}^{m} (1 - y_j) \cdot Z_{lj}
\hspace{2em} \texttt{select } p \text{ for which } m_p = \max_l m_l
\hspace{2em} \texttt{if } m_p > 0 \text{ then}
\hspace{3em} A_{ip} \leftarrow 1
\hspace{3em} \texttt{foreach } j \in \{1, \ldots, m\} \text{ do}
\hspace{4em} \texttt{if } Z_{pj} = 1 \text{ then}
\hspace{5em} Z_{j} \leftarrow 0; y_j \leftarrow 0
\hspace{2em} \texttt{until } m_p > 0
Algorithm 4: ComputeInitialFactors

\( C \leftarrow m \times m \) Boolean matrix with all entries equal to 0

\[
\text{foreach } C_{ij} \text{ do}
\]

\[
\quad \text{if } I_{-i} \leq I_{-j} \text{ and } |I_{-i}| > 0 \text{ then}
\]

\[
\quad \quad C_{ij} \leftarrow 1
\]

\[
\text{remove all duplicate and empty rows from } C
\]

\[
 f \leftarrow 0
\]

\[
\text{foreach row } i \in 1, \ldots, m \text{ of matrix } C \text{ do}
\]

\[
\quad \text{if row } C_{i-} \text{ has entry } j \text{ for which } C_{ij} = 1 \text{ and } C_{kj} = 0 \text{ for all } k < i \text{ then}
\]

\[
\quad \quad f \leftarrow f + 1
\]

\[
\quad \quad \text{add row } C_{i-} \text{ as a new row to } B
\]

\[
\text{if } f = \text{init} \text{ then}
\]

\[
\quad \text{return } B
\]
Basic Idea of 8M

1. Computing *init* initial factors
   - similarity with *Asso* algorithm

2. Iteratively computes new factors until \( k \) factors are obtained

3. Generating new factor via Boolean regression

4. Previously generated factors are revisited and dropped
   - adds two factors, then removes factor generated two steps back
   - \( k = 6 \), sequence: 2, 3, 4, 3, 4, 5, 4, 5, 6, 5, 6
Comparison with Other Methods

- **TILING**

- **ASSO**

- **GRECOND**

- **HYPER**

- **PaNDA**
Comparison with Other Methods: results

(a) Mushroom

(b) Set X1

Figure: Coverage quality of the first $l$ factors on real and synthetic data.
8M from Today’s Perspective

- Improvements of 8M
- Lessons from 8M
Improvements of 8M

- 8M+
- New initialization step
- Very fast strategy of GRECOND algorithm
- No overcovering error
Comparison of 8M and 8M+

Figure: Coverage quality of the first $l$ factors on real data: 8M vs. 8M+.
Lesson from 8M

- Revisiting the previously generated factors
- Significant aspect
- Non-symmetry of undercovering and overcovering error
- Existing algorithms do not use any kind of revisiting
- Improvement of existing algorithms
- Removes factors driven by parameter $p$
Lesson from 8M: improvement of GreConD

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$k$</th>
<th>$c$</th>
<th>orig.</th>
<th>0</th>
<th>0.01</th>
<th>0.02</th>
<th>0.03</th>
<th>0.04</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emea</td>
<td>42</td>
<td>1.000</td>
<td>34</td>
<td>29</td>
<td>26</td>
<td>25</td>
<td>24</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Chess</td>
<td>124</td>
<td>1.000</td>
<td>119</td>
<td>72</td>
<td>62</td>
<td>55</td>
<td>51</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>Firewall 1</td>
<td>66</td>
<td>1.000</td>
<td>65</td>
<td>17</td>
<td>10</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Firewall 2</td>
<td>10</td>
<td>1.000</td>
<td>10</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Mushroom</td>
<td>120</td>
<td>1.000</td>
<td>113</td>
<td>81</td>
<td>73</td>
<td>69</td>
<td>65</td>
<td>61</td>
<td></td>
</tr>
</tbody>
</table>

$p$

R. Belohlavek, M. Trnecka (Palacký University Olomouc)
Conclusions

- Detailed description of 8M
- Improvement of 8M
- New ideas for current BMF algorithms
- Explore revisiting of factors
Thank you