BOOLEAN MATRIX FACTORISATIONS IN DATA MINING (AND ELSEWHERE)

Pauli Miettinen
1 March 2014
In the sleepy days when the provinces of France were still quietly provincial, matrices with Boolean entries were a favored occupation of aging professors at the universities of Bordeaux and Clermont-Ferrand. But one day…

Gian-Carlo Rota

Foreword to Boolean matrix theory and applications by K. H. Kim, 1982
MOTIVATING EXAMPLE

Images by John Tenniel, openclipart.org, and Wikipedia
<table>
<thead>
<tr>
<th>Feature</th>
<th>Alice</th>
<th>Bob</th>
<th>Prince</th>
</tr>
</thead>
<tbody>
<tr>
<td>long-haired</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>well-known</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>male</td>
<td>✘</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
BINARY MATRIX

long-haired
well-known
male

( 1 | 1 | 1 | 0 )

( 1 | 1 | 0 )
FREQUENT ITEMSET MINING

- **Data:** Transactions over items
- **Goal:** Extract all sets of items that appear in many-enough transactions
- **Problem:** Too many frequent itemsets
  - Every subset of a frequent itemset is frequent
- **Solution:** Maximal, closed, and non-derivable itemsets
STILL TOO MANY ITEMSETS
TILING DATABASES

- **Goal:** Find itemsets that cover the transaction data
  - Itemset $I$ covers item $i$ in transaction $T$ if $i \in I \subseteq T$

- **Minimum tiling:** Find the smallest number of tiles that cover all items in all transactions

- **Maximum $k$-tiling:** Find $k$ tiles that cover the maximum number of item–transaction pairs

- If you have a set of tiles, these reduce to the Set Cover problem
TILING AS A MATRIX FACTORIZATION

\[
\begin{pmatrix}
1 & 1 & 0 \\
0 & 1 & 1 \\
\end{pmatrix}
\begin{pmatrix}
1 & 0 \\
0 & 1 \\
\end{pmatrix}
= 
\begin{pmatrix}
1 & 1 & 0 \\
0 & 1 & 1 \\
\end{pmatrix} 
\times 
\begin{pmatrix}
1 & 1 & 0 \\
0 & 1 & 1 \\
\end{pmatrix}
\]
BOOLEAN FACTORISATIONS
• The **Boolean matrix product** of two binary matrices $A$ and $B$ is their matrix product under Boolean semi-ring

$$(A \circ B)_{ij} = \bigvee_{i=1}^{k} a_{ik} b_{kj}$$

• The **Boolean matrix factorisation** of a binary matrix $A$ expresses it as a Boolean product of two binary factor matrices $B$ and $C$, that is, $A = BoC$
MATRIX RANKS

• The (Schein) **rank** of a matrix $\mathbf{A}$ is the least number of rank-1 matrices whose sum is $\mathbf{A}$

  $$\mathbf{A} = \mathbf{R}_1 + \mathbf{R}_2 + \ldots + \mathbf{R}_k$$

• Matrix is rank-1 if it is an outer product of two vectors

• The **Boolean rank** of binary matrix $\mathbf{A}$ is the least number of binary rank-1 matrices whose element-wise or is $\mathbf{A}$

  • The least $k$ such that $\mathbf{A} = \mathbf{B} \circ \mathbf{C}$ with $\mathbf{B}$ having $k$ columns
The Boolean rank of a matrix $A$ is the least number of complete bipartite subgraphs needed to cover every edge of the induced bipartite graph $G(A)$.
BOOLEAN RANK AND BICLQUIES

A B C
1 1 1 0
1 1 1 1
0 1 1 1
1 1 0 1
0 1 1 1

A B C
1 1 0
1 1 1
1 0 1
1 1 1

A B C
1
2
3
The Boolean rank of a matrix $A$ is the least number of subsets of $U(A)$ needed to cover every set of the induced collection $C(A)$.

For every $C$ in $C(A)$, if $S$ is the collection of subsets, have subcollection $S_C$ such that

$$\bigcup_{S \in S_C} S = C$$
THE MANY NAMES OF BOOLEAN RANK

- Minimum tiling (data mining)
- Rectangle covering number (communication complexity)
- Minimum bi-clique edge covering number (Garey & Johnson GT18)
- Minimum set basis (Garey & Johnson SP7)
- Optimum key generation (cryptography)
- Minimum set of roles (access control)
COMPARISON OF RANKS

• Boolean rank is NP-hard to compute
  • And as hard to approximate as the minimum clique
• Boolean rank can be less than normal rank
  • $\text{rank}_B(A) = O(\log_2(\text{rank}(A)))$ for certain $A$

$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$

• Boolean rank is never more than the non-negative rank
APPORXIMATE FACTORISATIONS

- Noise usually makes real-world matrices (almost) full rank
- We want to find a good low-rank approximation
  - The goodness is measured using the Hamming distance
- Given $A$ and $k$, find $B$ and $C$ such that $B$ has $k$ columns and $|A - BoC|$ is minimised
- No easier than finding the Boolean rank
THE BASIS USAGE PROBLEM

• Finding the factorisation is hard even if we know one factor matrix

• Problem. Given B and A, find X such that |AoX – B| is minimised

• We can replace B and X with column vectors

  • |Aox – b| versus ||Ax – b||

• Normal algebra: Moore–Penrose pseudo-inverse

• Boolean algebra: no polylogarithmic approximation
The minimum-error projection under Boolean algebra is equivalent to the following problem:

**Positive-Negative Partial Set Cover (±PSC).**

Given a triple \((P, N, Q)\), where \(P\) and \(N\) are disjoint sets and \(Q \subseteq 2^{P \cup N}\), find a subcollection \(D \subseteq Q\) that minimises \(|P \setminus (uD)| + |N \cap (uD)|\).
EXAMPLE

defines the sets

defines the sign

\(a\)

\(B\)
ALGORITHMS

Images by Wikipedia users Arab Ace and Sheilalau
THE BASIS USAGE

• Peleg’s algorithm approximates within $2\sqrt{[(k+a)\log a]}$

  • $a$ is the maximum number of 1s in $A$’s columns

• Optimal solution

  • Either an $O(2^k knm)$ exhaustive search, or an integer program

• Greedy algorithm: select each column of $B$ if it improves the residual error
THE ASSO ALGORITHM

• Heuristic – too many hardness results to hope for good provable results in any case

• **Intuition**: If two columns share a factor, they have 1s in same rows

• Noise makes detecting this harder

• Pairwise row association rules reveal (some of) the factors

• \( \Pr[a_{ik} = 1 \mid a_{jk} = 1] \)
THE PANDA ALGORITHM

- **Intuition**: every good factor has a noise-free core

- Two-phase algorithm:
  1. Find error-free core pattern (maximum area itemset/tile)
  2. Extend the core with noisy rows/columns

- The core patterns are found using a greedy method

- The 1s already belonging to some factor/tile are removed from the residual data where the cores are mined
SELECTING THE RANK

\[
\begin{bmatrix}
1 & 1 & 0 \\
1 & 1 & 0 \\
1 & 1 & 0 \\
1 & 1 & 0 \\
0 & 1 & 0 \\
1 & 1 & 0 \\
0 & 1 & 0 \\
1 & 1 & 0 \\
0 & 1 & 0 \\
1 & 1 & 0 \\
\end{bmatrix}
\]
• **Goal:** Separate noise from structure

• We assume data has correct type of structure

  • There are $k$ factors explaining the structure

  • Rest of the data does not follow the structure (noise)

• But how to decide where structure ends and noise starts?
MINIMUM DESCRIPTION LENGTH PRINCIPLE

• The best model (order) is the one that allows you to explain your data with least number of bits

• Two-part (crude) MDL: the cost of model \( L(\mathcal{H}) \) plus the cost of data given the model \( L(D | \mathcal{H}) \)

• Problem: how to do the encoding

• All involved matrices are binary: well-known encoding schemes
FITTING BMF TO MDL

- MDL requires exact representation
FITTING BMF TO MDL

- Two-part MDL: minimise $L(\mathcal{H}) + L(D \mid \mathcal{H})$

- Model $L(\mathcal{H})$

- Data given model $L(D \mid \mathcal{H})$

- $B \circ C$

- $E$
EXAMPLE: ASSO & MDL
• Many real-world binary matrices are sparse

• Representing sparse matrices with sparse factors is desirable
  • Saves space, improves usability, …

• Sparse matrices should be computationally easier
SPARSE FACTORISATIONS #1

• Any binary matrix $A$ that admits rank-$k$ BMF has factorisation to matrices $B$ and $C$ such that $|B| + |C| \leq 2|A|$

• $|A|$ is the number of non-zeros in $A$

• Can be extended to approximate factorisations

• Tight result (consider a case when $A$ has exactly one 1)
SPARSE FACTORISATIONS #2

- Let $s(A) = \frac{\text{zeros}(A)}{nm} = 1 - \frac{|A|}{nm}$ be the fraction of zeros in $A$

- If $B$ and $C$ are an underapproximation of $A$ (that is, $B \preceq C \preceq A$), then $s(B) + s(C) \geq s(A)$
CONCLUSIONS

Thank You!

• Boolean matrix factorisations are a topic older than I am

• Applications in many fields of CS and Math

• Approximate factorisations are an interesting tool for data mining

• Work is not done yet…