The Boolean Column and Column–Row Matrix Decompositions

Pauli Miettinen

16 September 2008
This Talk

1. Background.
2. Propose new decompositions combining two previously-proposed ideas.
3. Study the computational complexity of the problems.
   - Relate the results to other, known ones.
4. Propose simple algorithms for the problems.
5. Some experimental evaluation.
Outline

1. Background
2. Problem Definitions
3. Computational Complexity
4. Algorithms
5. Experiments
6. Conclusions
Background: Column and Column–Row Decompositions

Given a matrix $A$, represent it using

- linear combinations of a subset of its columns, i.e. $A \approx CX$ (CX decomposition)
  - Finding columns of $C$ is known as the Column Subset Selection problem.
  - Resembles feature selection.
- combinations of a subset of its columns and a subset of its rows, i.e. $A \approx CUR$ (CUR decomposition)

Lot-studied in math, recently gained interest in CS

- 1 + 1 papers in KDD’08, 2 papers in SODA’09 . . .
Background: Boolean Matrix Decompositions

Given a binary matrix $A$, represent it as $A \approx X \circ Y$, where $X$ and $Y$ are binary.

- Matrix multiplication is done over the Boolean semiring.
  - i.e. addition defined as $1 + 1 = 1$
- Can yield increased interpretability and decreased reconstruction error.
- Combinatorial problem, results from numerical linear algebra do not apply.
- Studied in combinatorics (Boolean or Schein rank), and in data mining
  - discrete basis problem (PKDD’06), role mining problem (ICDE’08), KDD’08 workshop on data mining using matrices and tensors, ...
Outline

1. Background
2. Problem Definitions
3. Computational Complexity
4. Algorithms
5. Experiments
6. Conclusions
Boolean CX and CUR Decompositions

Problem (Boolean CX Decomposition, BCX)

Given a matrix $A \in \{0, 1\}^{m \times n}$ and an integer $k$, find an $m \times k$ binary matrix $C$ of $k$ columns of $A$ and a matrix $X \in \{0, 1\}^{k \times n}$ minimizing $d_1(A, C \circ X) = \sum_{i,j} |(A)_{ij} - (C \circ X)_{ij}|$.

Problem (Boolean CUR Decomposition, BCUR)

Given a matrix $A \in \{0, 1\}^{m \times n}$ and integers $k$ and $r$, find an $m \times k$ binary matrix $C$ of $k$ columns of $A$, an $r \times n$ binary matrix $R$ of $r$ rows of $A$, and a matrix $U \in \{0, 1\}^{k \times r}$ minimizing $d_1(A, C \circ U \circ R) = \sum_{i,j} |(A)_{ij} - (C \circ U \circ R)_{ij}|$. 
Columns of $A$ represent corners in Boolean hypercube

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$
Columns of $A$ represent corners in Boolean hypercube

$$
\begin{pmatrix}
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1
\end{pmatrix}
$$
Columns of $A$ represent corners in Boolean hypercube

$$\begin{pmatrix}
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1
\end{pmatrix}$$
Columns of $A$ represent corners in Boolean hypercube

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

- $C$ selects some of the corners

$$
\begin{pmatrix}
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1
\end{pmatrix}
$$
**BCX Visualized**

- $C$ selects some of the corners

\[
\begin{pmatrix}
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1
\end{pmatrix}
\]
The remaining corner is presented as a sum of the selected corners.

\[
\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \approx \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \end{pmatrix}
\]
The remaining corner is presented as a **Boolean** sum of the selected corners.

\[
\begin{pmatrix}
1 \\
1 \\
1
\end{pmatrix}
= \begin{pmatrix}
1 & 0 \\
1 & 1 \\
0 & 1
\end{pmatrix} \circ \begin{pmatrix}
1
\end{pmatrix}
\]
The remaining corner is presented as a **Boolean** sum of the selected corners.

- **Green** line represents rank-2 SVD approximation of the **blue** corner.

\[
\begin{pmatrix}
1 \\
1 \\
1
\end{pmatrix}
\approx
\begin{pmatrix}
0.8536 \\
1.2071 \\
0.8536
\end{pmatrix}
\]
The whole rank-2 SVD approximation is the following.

$$U \Sigma V^T = \begin{pmatrix} 1.1036 & 0.8536 & 0.1036 \\ 0.8536 & 1.2071 & 0.8536 \\ 0.1036 & 0.8536 & 1.1036 \end{pmatrix}$$
Two Subproblems To Solve 1: Basis Usage

Problem (Basis Usage, BU)

Given matrices \( A \in \{0, 1\}^{n \times m} \) and \( C \in \{0, 1\}^{n \times k} \), find a matrix \( X \in \{0, 1\}^{k \times m} \) minimizing \( d_1(A, C \circ X) = \sum_{i,j} |(A)_{ij} - (C \circ X)_{ij}| \).

Few notes:

1. General definition: \( C \) does not have to have \( A \)'s columns.
2. Each column of \( X \) is independent!

Thus, an equivalent problem is:

Problem

Given a vector \( a \in \{0, 1\}^n \) and a matrix \( C \in \{0, 1\}^{n \times k} \), find a vector \( x \in \{0, 1\}^k \) minimizing \( \sum_i |a_i - (C \circ x)_i| \).
Two Subproblems To Solve 2: Mixing Matrix

Problem (Mixing Matrix, MM)

Given matrices $A \in \{0, 1\}^{n \times m}$, $C$ of $k$ columns of $A$, and $R$ of $r$ rows of $A$, find a matrix $U \in \{0, 1\}^{k \times r}$ minimizing

$$d_1(A, C \circ U \circ R) = \sum_{i,j} |(A)_{ij} - (C \circ U \circ R)_{ij}|.$$

- Now $C$ and $R$ are restricted to column and row subsets.
- No element of $U$ is independent.

$$(C \circ U \circ R)_{ij} = \bigvee_{h=1}^{k} \bigvee_{l=1}^{r} c_{ih} \land u_{hl} \land r_{lj}.$$

$\Rightarrow$ Element $u_{hl}$ can change $(C \circ U \circ R)_{ij}$ only when $c_{ih} = r_{lj} = 1.$
Outline

1. Background
2. Problem Definitions
3. Computational Complexity
4. Algorithms
5. Experiments
6. Conclusions
The Positive–Negative Partial Set Cover problem ($\pm$PSC):

- Cover as many of the positive elements as possible while minimizing the number of covered negative elements.

BU and $\pm$PSC problems are essentially the same.

1. BU with $A$ having only 1 column is no easier than other instances.

2. $C =$ incidence matrix of the set system; $a =$ positive ($a_i = 1$) and negative ($a_i = 0$) elements; $x$ selects the sets to the cover.
Complexity of the BU Problem (2/3): The Negative Side

Theorem

1. Unless \( P = NP \), then for any \( \varepsilon > 0 \) there exists no poly-time approximation algorithm for BU with ratio

\[
\Omega \left( 2^{\log^{1-\varepsilon} (k^4)} \right).
\]

2. Unless \( NP \subseteq DTIME(n^{polylog(n)}) \), then for any \( \varepsilon > 0 \) there exists no poly-time approximation algorithm for BU with ratio

\[
\Omega \left( 2^{\log^{1-\varepsilon} f} \right),
\]

where \( f \) is the maximum number of 1s in \( A \)’s columns.

N.B. \( 2^{\log^{1-\varepsilon} n} \) is superpoly-logarithmic and subpolynomial.

Pauli Miettinen

The Boolean Column and Column–Row Matrix Decompositions
Complexity of the BU Problem (3/3): The Positive Side

Theorem

There is a poly-time approximation algorithm with ratio $2\sqrt{(k + f) \log f}$.

The algorithm needs to solve the classical Set Cover multiple times with inflated input instances.
Complexity of the MM Problem

**Theorem**

*The MM problem can be reduced to the $\pm$PSC problem in an approximation-preserving way.*

**Theorem**

*The $\pm$PSC problem can be reduced to the MM problem preserving the approximability up to constant factors.*

- The results for the BU problem hold for the MM problem.
- **Caveat!** The parameters have changed
  - no meaningful counterpart to $f$
  - $k$ becomes to $\max\{k, r\}/2$. 
Complexity of the BCX Problem

- The hardness of BU does not automatically mean that BCX is hard.
- Nevertheless, via a reduction from BU we get that (the decision version of) BCX is NP-complete.
  - This reduction is not approximation-preserving.
- The complexity of the BCUR problem is an open question.
Linear and Boolean Worlds: A Comparison

Linear world

- Finding $x$ to minimize $\|Cx - a\|$ (i.e. least-squares fitting) is poly-time.
- Finding $U$ to minimize $\|CUR - A\|$ is poly-time.
- Complexity of the Column Subset Selection problem is unknown.

Boolean world

- Finding $x$ to minimize $\|C \odot x - a\|$ (i.e. the BU problem) is hard even to approximate.
- Finding $U$ to minimize $\|C \odot U \odot R - A\|$ (i.e. the MM problem) is hard even to approximate.
- The BCX problem is NP-hard.
Outline

1. Background
2. Problem Definitions
3. Computational Complexity
4. Algorithms
5. Experiments
6. Conclusions
Finding $C$ and $R$

Local-search heuristic $\text{Loc}$:

1. start with random columns in $C$
2. while reconstruction error decreases do
   1. swap a column of $C$ with a column of $A$ not in $C$ if this reduces the reconstruction error most
3. return $C$

Find $R$ by running $\text{Loc}$ to $A^\top$.

We need to know some $X$ to know how good a swap is.

⇒ Use greedy cover function: column $c^i$ is used to cover column $a^j$ (i.e. $x_{ij} = 1$) if $c^i$ covers more yet uncovered $1$s of $a^j$ than it covers uncovered $0$s.
Finding $X$ and $U$

- **Loc & ±PSC**: Use the ±PSC algorithm to find $X$. Practically infeasible to $U$.
- **Loc & IterX**: Iteratively update rows of $X$ using the *cover*-function.
- **Loc & IterU**: Start with empty $U$ and flip $u_{hl}$ if it decreases the error; iterate until convergence.
- **Loc & Maj**: For each $a_{ij}$ mark which $u_{hl}$ should be set to 1 or 0, and select $u_{hl}$ to be the (weighted) majority of the opinions.
  - Recall: $u_{hl}$ can change the value of $(C \circ U \circ R)_{ij}$ only if $c_{ih} = r_{lj} = 1$. 
Outline

1. Background
2. Problem Definitions
3. Computational Complexity
4. Algorithms
5. Experiments
6. Conclusions
### Other Algorithms

For general CX and CUR decompositions:
- **DMM** by Drineas, Mahoney, and Muthukrishnan (ESA, APPROX, and arXiv 2006–07)
  - based on sampling, approximates SVD within $1 + \varepsilon$ w.h.p., but needs lots of columns in $C$.

For general decompositions:
- **SVD**
  - lower bound for linear methods; in practice also a lower bound to all methods

For general Boolean matrix decompositions:
- **DBP** by Miettinen et al.
  - theoretical lower bound for Boolean methods
Synthetic Data

[B]CX decomposition, noise varies
- Results of continuous methods are rounded for improved accuracy.

[B]CUR decomposition, k varies

Pauli Miettinen
The Boolean Column and Column–Row Matrix Decompositions
Conclusions

- Boolean CX and CUR decompositions are potential tools for data mining.
- The problems are hard even to approximate, somewhat contrast to linear decompositions.
- Open questions: approximability of BCX, complexity of BCUR.
- Simple algorithms work up to some level, better ones are sought.
Conclusions

- Boolean CX and CUR decompositions are potential tools for data mining.
- The problems are hard even to approximate, somewhat contrast to linear decompositions.
- Open questions: approximability of BCX, complexity of BCUR.
- Simple algorithms work up to some level, better ones are sought.

Thank You!

Pauli Miettinen
The Boolean Column and Column–Row Matrix Decompositions