The Boolean Column and Column–Row Matrix Decompositions

Pauli Miettinen

16 September 2008

UNIVERSITY OF HELSINKI

This Talk

1. Background.
2. Propose new decompositions combining two previusly-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

Background: Column and Column–Row Decompositions

Given a matrix $\mathbf{A}$, represent it using

- linear combinations of a subset of its columns, i.e. $\mathbf{A} \approx \mathbf{CX}$ (CX decomposition)
  - Finding columns of $\mathbf{C}$ 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. $\mathbf{A} \approx \mathbf{CUR}$ (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 $\mathbf{A} \approx \mathbf{X} \circ \mathbf{Y}$, where $\mathbf{X}$ and $\mathbf{Y}$ 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, ...

UNIVERSITY OF HELSINKI

Outline

UNIVERSITY OF HELSINKI

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* $d_1(\mathbf{A}, \mathbf{C} \circ \mathbf{X}) = \sum_{i,j} \left|(\mathbf{A})_{ij} - (\mathbf{C} \circ \mathbf{X})_{ij}\right|$.

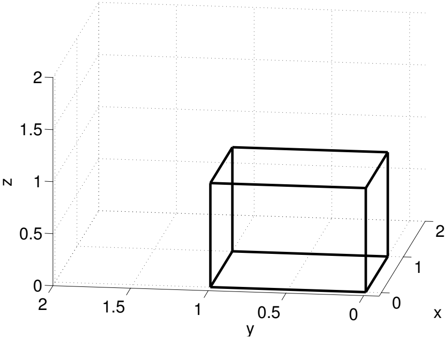
Problem (Boolean CUR Decomposition, BCUR)

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

BCX Visualized

- Columns of $\boldsymbol{A}$ represent corners in Boolean hypercube

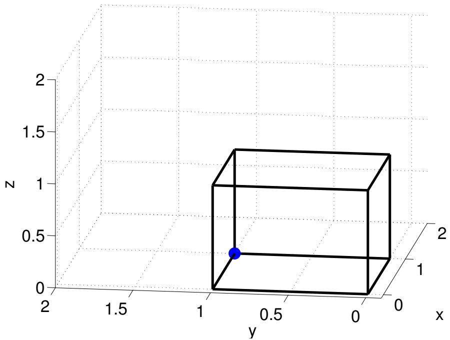
$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$



BCX Visualized

- Columns of A represent corners in Boolean hypercube

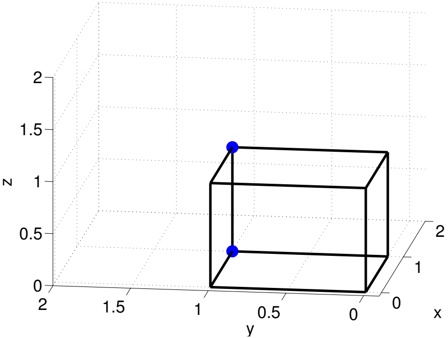
$$\begin{pmatrix} \color{blue}{1} & 1 & 0 \\ \color{blue}{1} & 1 & 1 \\ \color{blue}{0} & 1 & 1 \end{pmatrix}$$



BCX Visualized

- Columns of A represent corners in Boolean hypercube

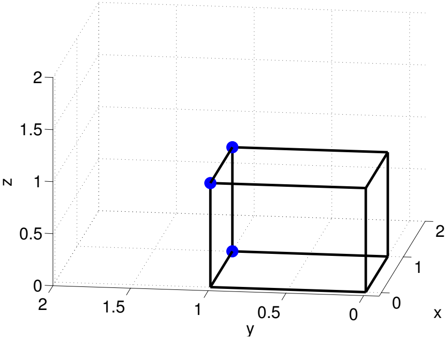
$$\begin{pmatrix} \color{blue}{1} & \color{blue}{1} & 0 \\ \color{blue}{1} & \color{blue}{1} & 1 \\ \color{blue}{0} & \color{blue}{1} & 1 \end{pmatrix}$$



BCX Visualized

- Columns of A represent corners in Boolean hypercube

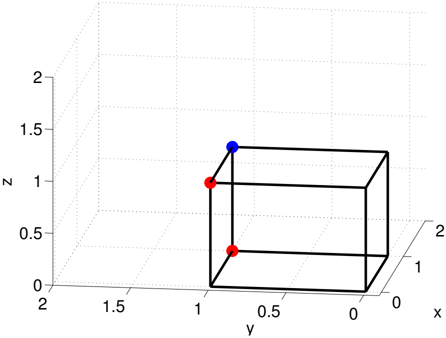
$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$



BCX Visualized

- $\mathbf{C}$ selects some of the corners

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

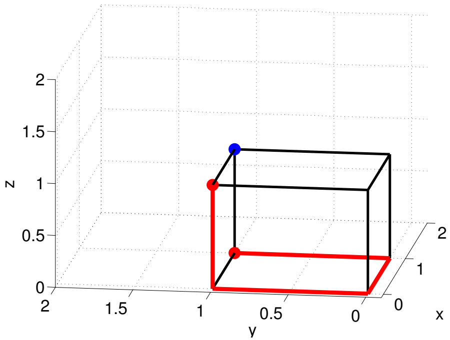


UNIVERSITY OF HELSINKI

BCX Visualized

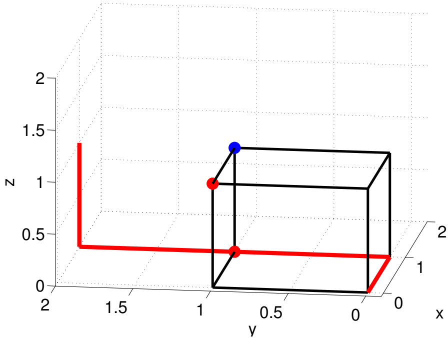
- $\mathbf{C}$ selects some of the corners

$$\begin{pmatrix} \color{red}{1} & \color{blue}{1} & \color{red}{0} \\ \color{red}{1} & \color{blue}{1} & \color{red}{1} \\ \color{red}{0} & \color{blue}{1} & \color{red}{1} \end{pmatrix}$$



BCX Visualized

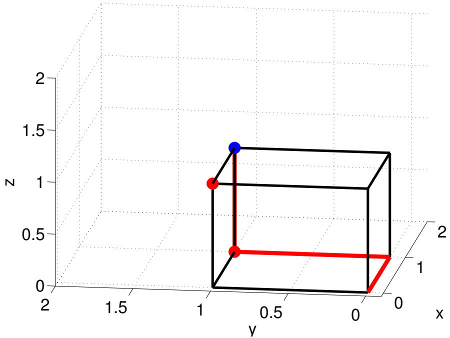
- The remaining corner is presented as a sum of the selected corners.



$$\begin{pmatrix} \color{blue}{1} \\ \color{blue}{1} \\ \color{blue}{1} \end{pmatrix} \approx \begin{pmatrix} \color{red}{1} & \color{red}{0} \\ \color{red}{1} & \color{red}{1} \\ \color{red}{0} & \color{red}{1} \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

BCX Visualized

- The remaining corner is presented as a **Boolean** sum of the selected corners.

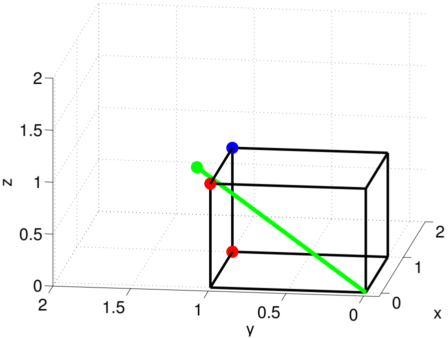


$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

BCX Visualized

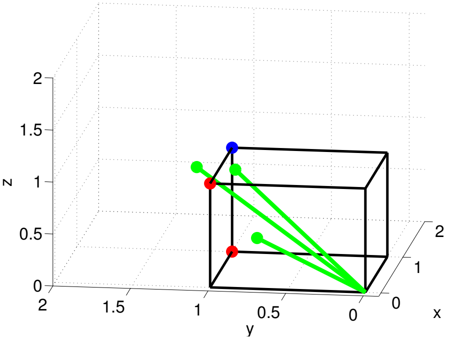
- The remaining corner is presented as a **Boolean** sum of the selected corners.
- Green line represents rank-2 SVD approximation of the blue corner.

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \approx \begin{pmatrix} 0.8536 \\ 1.2071 \\ 0.8536 \end{pmatrix}$$



BCX Visualized

- The whole rank-2 SVD approximation is the following.



$$\mathbf{U\Sigma V}^T = \begin{pmatrix} 1.1036 & 0.8536 & 0.1036 \\ 0.8536 & 1.2071 & 0.8536 \\ 0.1036 & 0.8536 & 1.1036 \end{pmatrix}$$

Two Subproblems To Solve 1: Basis Usage

Problem (Basis Usage, BU)

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

Few notes:

1. General definition: $\mathbf{C}$ does **not** have to have $\mathbf{A}$'s columns.
2. Each column of $\mathbf{X}$ is independent!

Thus, an equivalent problem is:

Problem

Given a vector $\mathbf{a} \in \{0,1\}^{n}$ *and a matrix* $\mathbf{C} \in \{0,1\}^{n \times k}$, *find a vector* $\mathbf{x} \in \{0,1\}^{k}$ *minimizing* $\sum_i \left|a_i - (\mathbf{C} \circ \mathbf{x})_i\right|$.

Two Subproblems To Solve 2: Mixing Matrix

Problem (Mixing Matrix, MM)

Given matrices $\mathbf{A} \in \{0,1\}^{n \times m}$, $\mathbf{C}$ *of* k *columns of* $\mathbf{A}$, *and* $\mathbf{R}$ *of* r *rows of* $\mathbf{A}$, *find a matrix* $\mathbf{U} \in \{0,1\}^{k \times r}$ *minimizing* $d_1(\mathbf{A}, \mathbf{C} \circ \mathbf{U} \circ \mathbf{R}) = \sum_{i,j} \left| (\mathbf{A})_{ij} - (\mathbf{C} \circ \mathbf{U} \circ \mathbf{R})_{ij} \right|$.

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

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

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

Outline

UNIVERSITY OF HELSINKI

Complexity of the BU Problem (1/3): Background

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$(n^{\text{polylog}(n)})$, *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* 1*s in* $\mathbf{A}$*'s columns.*

- **N.B.** $2^{\log^{1-\varepsilon} n}$ is superpolylogarithmic and subpolynomial.

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

Boolean world

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

Outline

Finding $\mathbf{C}$ and $\mathbf{R}$

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

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

- Find $\mathbf{R}$ by running $\texttt{Loc}$ to $\mathbf{A}^T$.
- We need to know **some** $\mathbf{X}$ to know how good a swap is.
  - $\Rightarrow$ Use greedy *cover* function: column $\mathbf{c}^i$ is used to cover column $\mathbf{a}^j$ (i.e. $x_{ij} = 1$) if $\mathbf{c}^i$ covers more **yet uncovered 1s** of $\mathbf{a}^j$ than it covers **uncovered 0s**.

Finding $\mathbf{X}$ and $\mathbf{U}$

- `Loc & ±PSC`: Use the ±PSC algorithm to find $\mathbf{X}$.
  - Practically infeasible to $\mathbf{U}$.
- `Loc & IterX`: Iteratively update rows of $\mathbf{X}$ using the *cover*-function.
- `Loc & IterU`: Start with empty $\mathbf{U}$ and flip u_{hl} if it decreases the error; iterate untill 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 $(\mathbf{C} \circ \mathbf{U} \circ \mathbf{R})_{ij}$ only if $c_{ih} = r_{lj} = 1$.

UNIVERSITY OF HELSINKI

Outline

Other Algorithms

For general CX and CUR decompositions:

- $\mathrm{8\,4\,4}$ by Berry, Pulatova, and Stewart (ACM Trans. Math. Softw. 2005)
- 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.

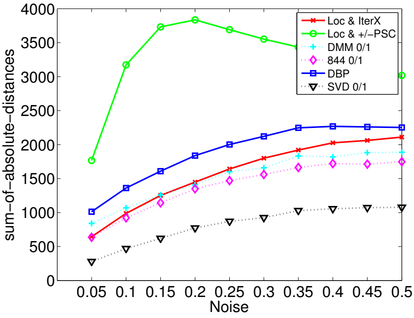
For general decompositions:

- SVD
  - lower bound for linear methods; in practice also a lower bound to all methods

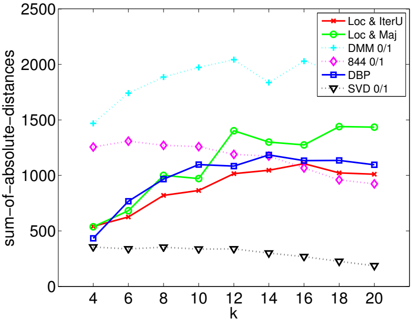
For general Boolean matrix decompositions:

- DBP by Miettinen et al.
  - theoretical lower bound for Boolean methods

Synthetic Data



[B]CX decomposition, noise varies



[B]CUR decomposition, k varies

- Results of continuous methods are rounded for improved accuracy.

UNIVERSITY OF HELSINKI

Outline

UNIVERSITY OF HELSINKI

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.

UNIVERSITY OF HELSINKI

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!

UNIVERSITY OF HELSINKI