Delegation-Relegation for Boolean Matrix Factorization

Florent Avellaneda and Roger Villemaire

Université du Québec à Montréal (UQAM)

AAAI 2024

UQÀM

Matrix Factorization Problem

$$M = \begin{vmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{vmatrix}$$

Goal:

Find $A_{m \times k}$ and $B_{k \times n}$ such that $A \times B \approx M$

$$(A \times B)_{i,j} = \sum_{\ell=1}^{k} A_{i,\ell} \times B_{\ell,j}$$

Matrix Factorization Problem

$$M = \begin{vmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{vmatrix}$$

Goal:
Find $A_{m \times k}$ and $B_{k \times n}$ such that $A \times B \approx M$

$$(A \times B)_{i,j} = \sum_{\ell=1}^{k} A_{i,\ell} \times B_{\ell,j}$$

Example of a rank 2 factorization ($k = 2$):

$$\begin{array}{cc} & \begin{vmatrix} b_{0,0} & b_{0,1} & b_{0,2} \\ b_{1,0} & b_{1,1} & b_{1,2} \end{vmatrix} \\ \begin{vmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \\ a_{2,0} & a_{2,1} \end{vmatrix} & \begin{vmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{vmatrix} \end{array}$$

Constraints:

$$\forall i, j : \sum_{\ell=0}^{k} a_{i,\ell} \times b_{\ell,j} \approx M_{i,j}$$

Solution with SVD

$$\begin{vmatrix} 0.5 & 0.7 & 0.5 \\ -0.7 & 0 & 0.7 \end{vmatrix}$$

$$\begin{vmatrix} 1.1 & 0.7 \\ 1.7 & 0 \\ 1.2 & -0.7 \end{vmatrix} \begin{vmatrix} 0.11 & 0.84 & 1.09 \\ 0.89 & 1.19 & 0.85 \\ 1.09 & 0.84 & 0.11 \end{vmatrix}$$

Solution with SVD

$$\begin{vmatrix} 0.5 & 0.7 & 0.5 \\ -0.7 & 0 & 0.7 \end{vmatrix}$$

$$\begin{vmatrix} 1.1 & 0.7 \\ 1.7 & 0 \\ 1.2 & -0.7 \end{vmatrix} \quad \begin{vmatrix} 0.11 & 0.84 & 1.09 \\ 0.89 & 1.19 & 0.85 \\ 1.09 & 0.84 & 0.11 \end{vmatrix}$$

Problems:

- No exact solution of rank 2

Solution with SVD

$$\begin{vmatrix} 0.5 & 0.7 & 0.5 \\ -0.7 & 0 & 0.7 \end{vmatrix}$$

$$\begin{vmatrix} 1.1 & 0.7 \\ 1.7 & 0 \\ 1.2 & -0.7 \end{vmatrix} \quad \begin{array}{l} \begin{vmatrix} 0.11 & 0.84 & 1.09 \\ 0.89 & 1.19 & 0.85 \\ 1.09 & 0.84 & 0.11 \end{vmatrix} \begin{array}{l} \text{Alice} \\ \text{Bob} \\ \text{Charle} \end{array} \\ \begin{array}{ccc} \text{Paw Patrol} & \text{Shrek} & \text{Matrix} \end{array} \end{array}$$

Problems:

- No exact solution of rank 2
- Poor interpretability of the factorization

Boolean Matrix Factorization Problem

$$M = \begin{vmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{vmatrix}$$

Goal:

Find $A_{m \times k}$ and $B_{k \times n}$ such that $A \circ B = M$

$$(A \circ B)_{i,j} = \bigvee_{\ell=1}^{k} A_{i,\ell} \wedge B_{\ell,j}$$

Boolean Matrix Factorization Problem

$$M = \begin{vmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{vmatrix}$$

Goal:
Find $A_{m \times k}$ and $B_{k \times n}$ such that $A \circ B = M$

$$(A \circ B)_{i,j} = \bigvee_{\ell=1}^{k} A_{i,\ell} \wedge B_{\ell,j}$$

Example of a rank 2 factorization ($k = 2$):

$$\begin{array}{cc} & \begin{vmatrix} b_{0,0} & b_{0,1} & b_{0,2} \\ b_{1,0} & b_{1,1} & b_{1,2} \end{vmatrix} \\ \begin{vmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \\ a_{2,0} & a_{2,1} \end{vmatrix} & \begin{vmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{vmatrix} \end{array}$$

Constraints :

$$\forall i, j : \bigvee_{\ell=0}^{k} a_{i,\ell} \wedge b_{\ell,j} = M_{i,j}$$

Solution with BMF

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

$$\begin{vmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{vmatrix} \quad \begin{array}{c} \begin{array}{cc} \begin{vmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix} & \begin{array}{l} \text{Alice} \\ \text{Bob} \\ \text{Charle} \end{array} \end{array} \\ \begin{array}{ccc} \text{Paw Patrol} & \text{Shrek} & \text{Matrix} \end{array} \end{array}$$

Solution with BMF

$$
\begin{array}{cc}
 & \begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{vmatrix} \\
\begin{vmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{vmatrix} &
\begin{array}{cl}
\begin{vmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix} & \begin{array}{l} \text{Alice} \\ \text{Bob} \\ \text{Charle} \end{array} \\
\begin{array}{ccc} \text{Paw Patrol} & \text{Shrek} & \text{Matrix} \end{array} &
\end{array}
\end{array}
$$

Advantages:

- Exact solution of rank 2

Solution with BMF

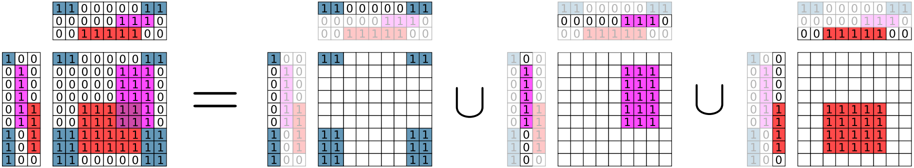
$$\begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{vmatrix}$$

$$\begin{vmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{vmatrix} \quad \begin{array}{cccl} \begin{vmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix} & \begin{array}{l} \text{Alice} \\ \text{Bob} \\ \text{Charle} \end{array} \\ \text{Paw Patrol} \quad \text{Shrek} \quad \text{Matrix} & \end{array}$$

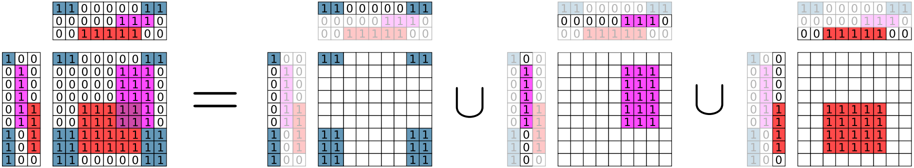
Advantages:

- Exact solution of rank 2
- Good interpretability of the factorization

A Boolean Matrix Factorization of order k involves covering all the 1s with k blocks:

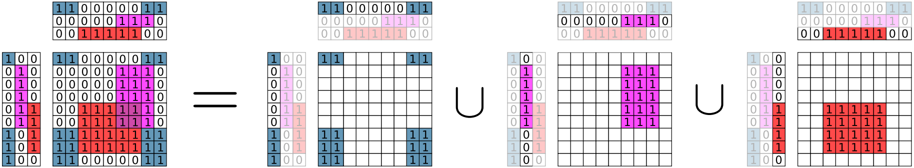


A Boolean Matrix Factorization of order k involves covering all the 1s with k blocks:



Idea: The fewer the number of 1s, the easier it should be to find a factorization.

A Boolean Matrix Factorization of order k involves covering all the 1s with k blocks:



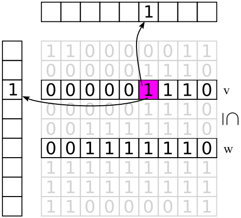
Idea: The fewer the number of 1s, the easier it should be to find a factorization.

Remark: This idea has been introduced in a particular family of BMF algorithms based on *formal concept analysis* such as Iteress [Belohlavek, Outrata and Trnecka].

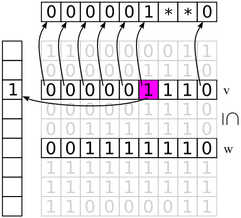
Delegation and Relegation Operators

1	1	0	0	0	0	0	1	1	
0	0	0	0	0	1	1	1	0	
0	0	0	0	0	1	1	1	0	v
0	0	0	0	0	1	1	1	0	$\subseteq$
0	0	1	1	1	1	1	1	0	
0	0	1	1	1	1	1	1	0	w
1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	
1	1	0	0	0	0	0	1	1	

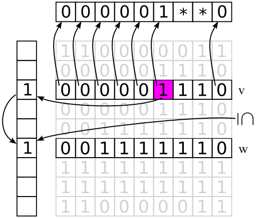
Delegation and Relegation Operators



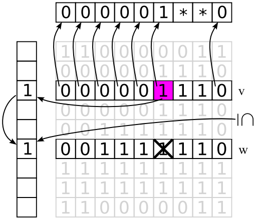
Delegation and Relegation Operators



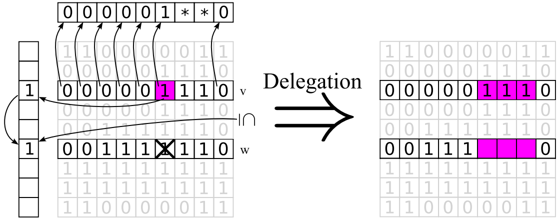
Delegation and Relegation Operators



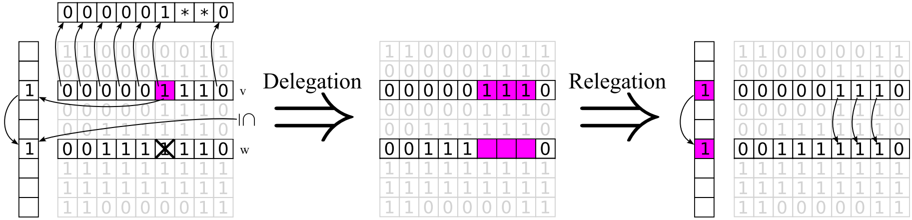
Delegation and Relegation Operators



Delegation and Relegation Operators



Delegation and Relegation Operators



Definitions

Definition. A matrix $X'_{m\times n}$ is *existentially included* in a matrix $X_{m\times n}$ (denoted $X' \leq^{\exists} X$) if there is no i, j such that $X'_{i,j} = 1$ and $X_{i,j} = 0$.

Definition. A matrix $X'_{m\times n}$ is *universally included* in a matrix $X_{m\times n}$ (denoted $X' \leq^{\forall} X$) if for all i, j, if $X_{i,j} = 0$, then $X'_{i,j} = 0$.

Definition. A matrix $X'_{m\times n}$ is *consistent* with a matrix $X_{m\times n}$ (denoted $X' \simeq X$) if $X \leq^{\exists} X'$ and $X' \leq^{\exists} X$.

Delegation

We denote by $X^{v\downarrow w}$ the *delegation* of the line w to the line v in the matrix X, and by $X^{v\to w}$ the *delegation* of the column w to the column v in the matrix X.

$$X^{v\downarrow w}_{i,j} = \begin{cases} 0 & \text{if } i = v \text{ and } X_{w,j} = 0, \\ \emptyset & \text{if } i = w \text{ and } X_{v,j} = 1, \\ X_{i,j} & \text{otherwise.} \end{cases}$$

$$X^{v\to w}_{i,j} = \begin{cases} 0 & \text{if } j = v \text{ and } X_{i,w} = 0, \\ \emptyset & \text{if } j = w \text{ and } X_{i,v} = 1, \\ X_{i,j} & \text{otherwise.} \end{cases}$$

Relegation

We denote by $A^{v\uparrow w}$ the *relegation* of the line w from the line v in the matrix A and by $B^{v\leftarrow w}$ the *relegation* of the column w from the column v in the matrix B.

$$A^{v\uparrow w}_{i,j} = \begin{cases} 1 & \text{if } i = w \text{ and } A_{v,j} = 1, \\ A_{i,j} & \text{otherwise.} \end{cases}$$

$$B^{v\leftarrow w}_{i,j} = \begin{cases} 1 & \text{if } j = w \text{ and } B_{i,v} = 1, \\ B_{i,j} & \text{otherwise.} \end{cases}$$

Theorems

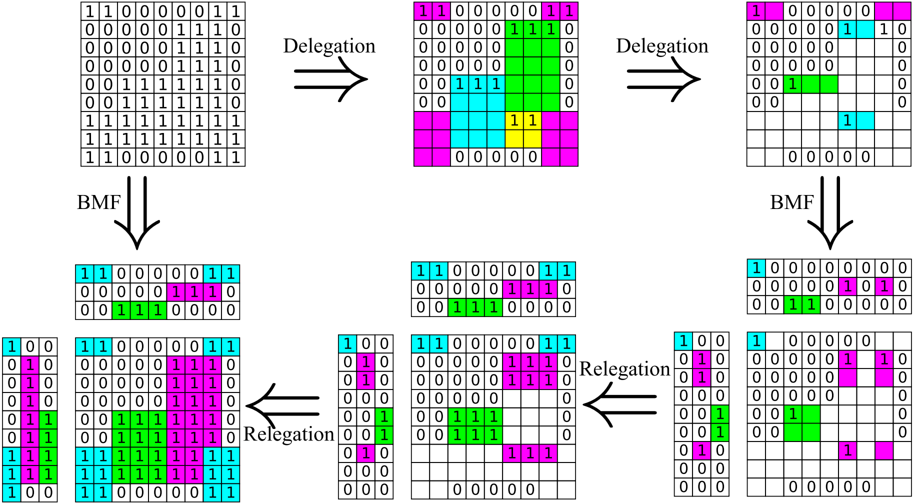
Theorem 1. Let v, w be such that $X_{v,:} \leq^{\exists} X_{w,:}$.

- If $(A \circ B) \simeq X^{v\downarrow w}$ then $(A^{v\uparrow w} \circ B) \simeq X$.
- If $(A \circ B) \simeq X^{v\rightarrow w}$ then $(A \circ B^{v\leftarrow w}) \simeq X$.

Theorem 2. Let v, w be such that $X_{v,:} \leq^{\forall} X_{w,:}$.

- $(A^{v\uparrow w} \circ B)$ is an optimal BMF for X if and only if $(A \circ B)$ is an optimal BMF for $X^{v\downarrow w}$.
- $(A \circ B^{v\leftarrow w})$ is an optimal BMF for X if and only if $(A \circ B)$ is an optimal BMF for $X^{v\rightarrow w}$.

Algorithm



Benchmark

We conducted an evaluation of our methods, Simpli$^{\exists}$ and Simpli$^{\forall}$, on well-established datasets from the literature [UCI].

We focusing on two key aspects: the degree of simplification they achieve and their effect on the time savings when performing factorizations on the simplified matrices using existing constraint-based BMF solvers:

- CG [Kovacs, Gunluk and Hauser]
- OptiBlock [Avellaneda and Villemaire]

Dataset	Characteristics		# ones after simplify		
	Size	# Ones	Iteress	Simpli$^{\forall}$	Simpli$^{\exists}$
Advert.	3279×1557	45139	5941	<u>3942</u>	**705**
Chess	3196×39	25582	<u>368</u>	780	**38**
DNA	4590×392	26527	1556	<u>539</u>	**367**
Firewa.	365×709	31951	2744	<u>88</u>	**65**
Flare	1066×43	9283	2928	<u>1950</u>	**428**
Heart	270×382	3036	<u>325</u>	1459	**270**
Iris	150×126	750	<u>502</u>	515	**486**
Lymph	148×54	1823	<u>1288</u>	1543	**1283**
Paleo	501×139	3537	<u>284</u>	1853	**139**
Student	395×176	9254	<u>8488</u>	8517	**8470**
Thorac.	470×340	3376	<u>2373</u>	2439	**2310**
Tictac.	958×28	8954	**8954**	**8954**	**8954**
Wine	178×1279	2492	816	<u>190</u>	**178**
Zoo	101×28	640	<u>85</u>	108	**25**

Dataset	BMF with CG : time (rank)			
	ORIGINAL	ITERESS	SIMPLI$^{\forall}$	SIMPLI$_{0}^{\exists}$
ADVERT.	3h (1556)	3h (1596)	3h (1556)	**3h (704)**
CHESS	3h (38)	1s (38)	**20m (38*)**	1s (38)
DNA	1m (392)	3h (368)	3h (384)	**5m (367)**
FIREWA.	3h (64)	9m (65)	**1h (64*)**	1s (65)
FLARE	2m (43)	3h (42)	**1h (42*)**	3h (42)
HEART	3h (270)	**9m (270)**	3h (270)	**9m (270)**
IRIS	**10m (121*)**	8m (121)	**10m (121*)**	9m (121)
LYMPH	3h (52)	20m (53)	**3h (52*)**	40m (53)
PALEO	3h (139)	**1s (139)**	3h (139)	**1s (139)**
STUDENT	**3h (176)**	**3h (176)**	**3h (176)**	**3h (176)**
THORAC.	**3h (304)**	**3h (304)**	**3h (304)**	**3h (304)**
TICTAC.	**3h (28)**	**3h (28)**	**3h (28)**	**3h (28)**
WINE	3h (178)	20s (178)	**4m (178*)**	20s (178)
ZOO	3h (25)	1s (25)	**3h (25*)**	1s (25)

Data	BMF with OptiBlock : time (rank)			
	ORIGINAL	SIMPLI$^{\forall}$	ITERESS	SIMPLI$_{0}^{\exists}$
ADVERT.	3h (794)	3h (749)	3h (711)	**3h (703)**
CHESS	1m (38)	20s (38)	**10s (38)**	**10s (38)**
DNA	3h (497)	3h (373)	1h (368)	**1h (367)**
FIREWA.	2m (64)	**1m (64)**	30s (65)	30s (65)
FLARE	14s (42)	14s (42)	**4s (42)**	**4s (42)**
HEART	90m (270)	90m (270)	**2m (270)**	**2m (270)**
IRIS	**10s (121)**	10s (122)	10s (122)	20s (122)
LYMPH	5s (54)	6s (54)	3s (55)	**3s (54)**
PALEO.	4m (139)	2m (139)	**30s (139)**	**30s (139)**
STUDENT	6m (176)	6m (176)	5m (177)	**4m (176)**
THORAC.	**10m (304)**	20m (306)	20m (306)	20m (305)
TICTAC.	**3s (28)**	**3s (28)**	**3s (28)**	**3s (28)**
WIN.	6m (178)	**2m (178)**	**2m (178)**	**2m (178)**
ZOO	**1s (25)**	**1s (25)**	**1s (25)**	**1s (25)**