Matrix Completion over GF(2) with Applications to Index Coding

Akhilesh Soni

Jeff Linderoth

Jim Luedtke

needs a haircut

Daniel Pimentel-Alarcón

Department of Industrial and Systems Engineering
Department of Biostatistics and Medical Informatics
University of Wisconsin-Madison

ICERM

March 1, 2023

Research Supported by American Family Insurance

ChatGPT To The Rescue

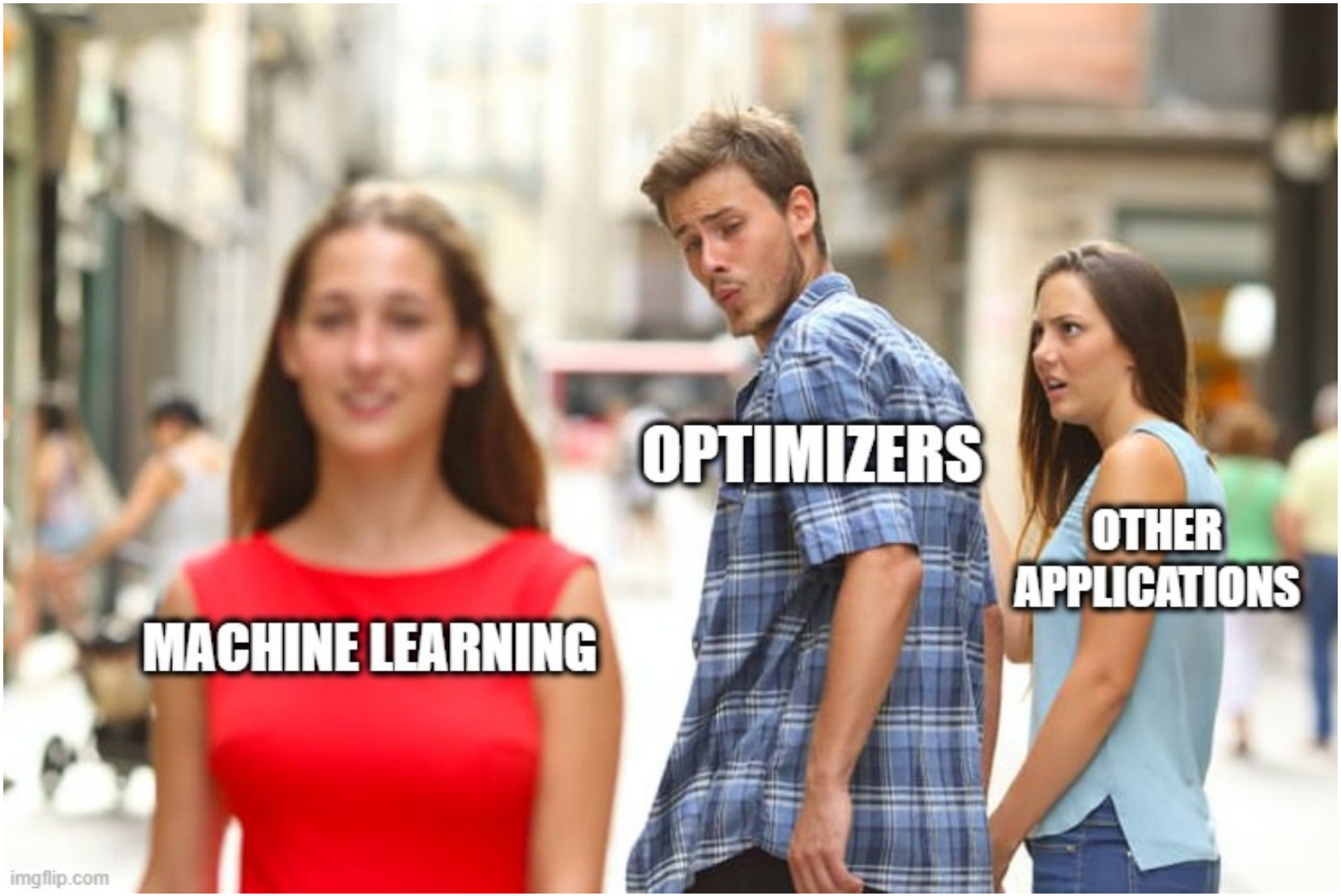
Write a limerick apologizing for this math talk being work in progress

There once was a math talk I gave,
But progress had not been fully made,
It was work in progress,
Please don't digress,
I'm sorry for the points left unweighed!

Outline

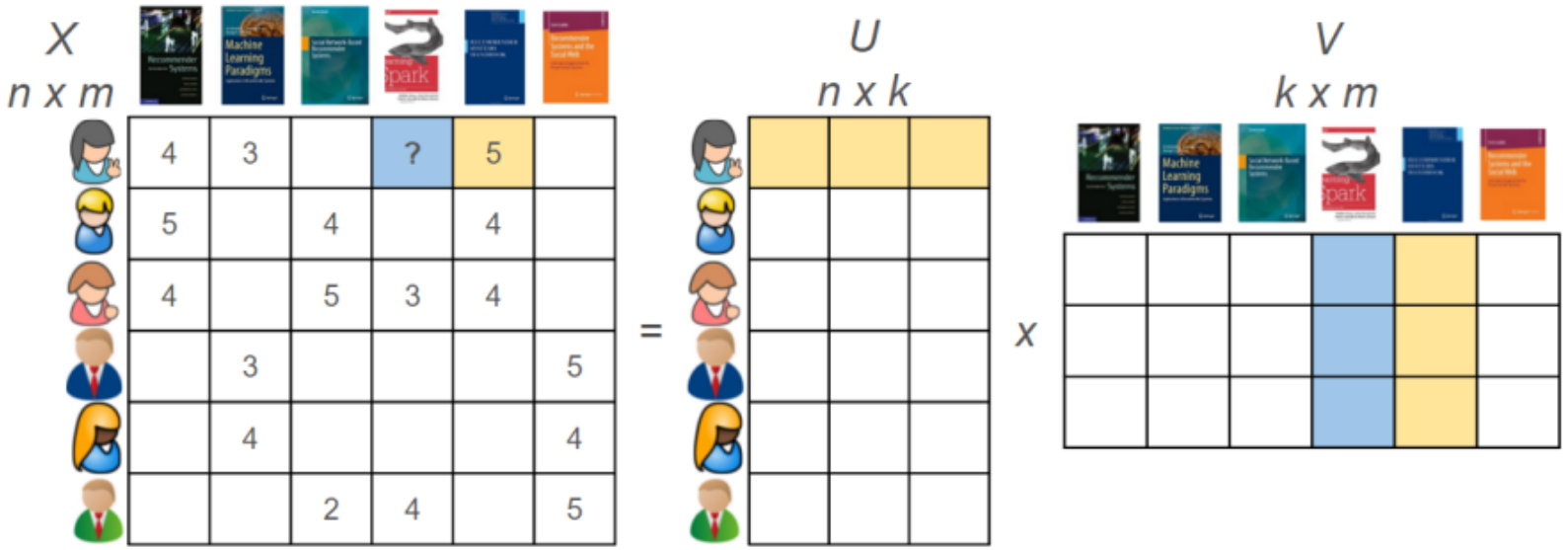
- Matrix completion
- Binary matrix factorization and completion
- Index coding
- Three IP Formulations
  1. McCormick + Integer Variable
  2. McCormick + Parity Disjunction
  3. McCormick-Free
- Less than impressive computational results

Jeff Wants In On The Action



Low-Rank Matrix Completion: Netflix Problem

- There exists a matrix $X \in \mathbb{R}^{d \times n}$ whose entries are only known for a fraction of the elements $\Omega \subset [d] \times [n]$
- To complete the matrix, we must assume some structure.
- Here we assume X is low-rank: $X = UV$ for some $U \in \mathbb{R}^{d \times r}$, $V \in \mathbb{R}^{r \times n}$



0-1 Matrix Completion?

- In some earlier work sponsored by American Family, we did a combination of matrix completion and clustering—Subspace clustering with missing data
- They asked us to try it out on their data matrix—which was a 0-1 matrix (?!)
- Doing "normal" low-rank matrix completion, say using nuclear norm, or any other very powerful methods, is *not* going to give 0-1 values for the missing entries
- Even if you fill in unknowns with real values, the points typically don't lie on a (low dimensional) hyperplane in $\mathbb{R}$

What to do?

- Don't do it over $\mathbb{R}$.
- What about Boolean Algebra ($1 + 1 = 1$)— natural for revealing "low-dimensional" characteristics

Boolean Algebra: 1+1 = 1. (Logical Or)



$$X = \begin{array}{c} \\ \text{Long Hair} \\ \text{Loves MIP} \\ \text{Cheesehead} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Simge} & \text{Jim} & \text{Jeff} \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{array}$$

Two Groups of People, Two Traits

- Simge and Jim have long hair and love MIP
- Jim and Jeff love MIP and are cheeseheads

Two Factors

$$X = \begin{array}{c} \\ \text{Long Hair} \\ \text{Loves MIP} \\ \text{Cheesehead} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Simge} & \text{Jim} & \text{Jeff} \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{array} = \begin{array}{c} \begin{array}{cc} \text{T1} & \text{T2} \end{array} \\ \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \end{array} \circ \begin{array}{c} \begin{array}{ccc} \text{Simge} & \text{Jim} & \text{Jeff} \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{array}$$

- Writing $X = \vee_{k=1}^{r} u^k (v^k)^\top$ reveals the fundamental "traits", and classifies individuals depending on which traits they have
- So we started working on integer programming approaches to matrix factorization and completion in Boolean algebra

I Hate This Guy

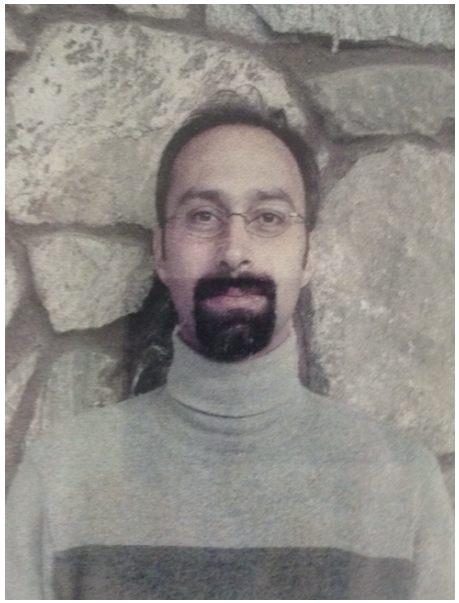
Binary Matrix Factorisation and Completion via Integer Programming

Oktay Günlük
Cornell University, ong5@cornell.edu

Raphael A. Hauser, Réka Á. Kovács
University of Oxford, The Alan Turing Institute, hauser@maths.ox.ac.uk, reka.kovacs@maths.ox.ac.uk

Binary matrix factorisation is an essential tool for identifying discrete patterns in binary data. In this paper we consider the rank-k binary matrix factorisation problem (k-BMF) under Boolean arithmetic: we are given an $n \times m$ binary matrix X with possibly missing entries and need to find two binary matrices A and B of dimension $n \times k$ and $k \times m$ respectively, which minimise the distance between X and the Boolean product of A and B in the squared Frobenius distance. We present a compact and two exponential size integer programs (IPs) for k-BMF and show that the compact IP has a weak LP relaxation, while the exponential size IPs have a stronger equivalent LP relaxation. We introduce a new objective function, which differs from the traditional squared Frobenius objective in attributing a weight to zero entries of the input matrix that is proportional to the number of times the zero is erroneously covered in a rank-k factorisation. For one of the exponential size IPs we describe a computational approach based on column generation. Experimental results on synthetic and real word datasets suggest that our integer programming approach is competitive against available methods for k-BMF and provides accurate low-error factorisations.

Key words: binary matrix factorisation, binary matrix completion, column generation, integer programming
MSC2000 subject classification: 90C10
OR/MS subject classification: Integer Programming
History:



Oktay Ruined It—Nothing Left To Do

- IP Formulations
- Strong Formulations
- Column Generation Approaches.

$$\mathbb{F}_2?$$

$$1+1=0$$

Binary Matrix Factorization/Completion

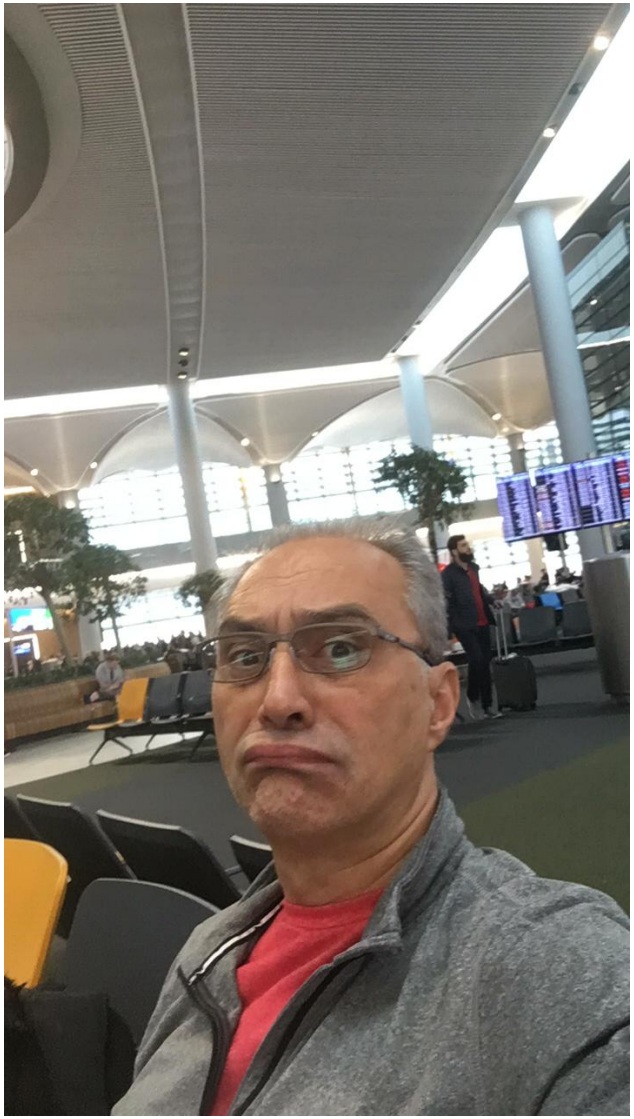
Matrix Factorization

- Boolean: Find smallest r such that $X = \vee_{k=1}^{r} u^k (v^k)^\top$, where $u^k \in \{0,1\}^d, v^k \in \{0,1\}^n$. This is hard
- $\mathbb{F}_2$: Find smallest r such that $X = \oplus_{k=1}^{r} u^k (v^k)^\top$, where $u^k \in \{0,1\}^d, v^k \in \{0,1\}^n$. This is easy

Matrix Completion. Given $\Omega \subset [d] \times [n]$, $X_{ij} \in \{0,1\}\ \forall ij \in \Omega$, $r \in \mathbb{Z}_+$

- Find $u^k \in \{0,1\}^d, v^k \in \{0,1\}^n$ to $\min \|X_{ij} - \vee_{k=1}^{r} u^k (v^k)^\top)\|_\Omega$. This is hard.
- Find $u^k \in \{0,1\}^d, v^k \in \{0,1\}^n$ to $\min \|X_{ij} - \oplus_{k=1}^{r} u^k (v^k)^\top)\|_\Omega$. This is hard.

What Oktay Said



"Matrix Completion in $\mathbb{F}_2$?!?!
Why on earth would anyone want
to solve that problem?"

Index Coding (with Side Information)

- We have a collection of n messages/packets, each in $\{0,1\}^t$, and a collection of n receivers.
  - Each receiver wants to know one of the messages
  - Each receiver "knows" (has cached) some subset of the packets—Just not the one it wants to know
- Central broadcaster knows which packets are cached at each receiver

Index Coding

Broadcast a minimum number of messages so that each receiver can recover its message using its local information

Intuition

Send a basis of "known" information $\Rightarrow$ each receiver can compute their own message. Min rank is minimum number of messages

Index Coding: Example

Receiver	Has Messages
1	2,5
2	1,5
3	2,4
4	2,3
5	1,3,4

$$X = \begin{array}{c} \\ M1 \\ M2 \\ M3 \\ M4 \\ M5 \end{array} \begin{array}{c} \begin{array}{ccccc} R1 & R2 & R3 & R4 & R5 \end{array} \\ \begin{bmatrix} 1 & - & 0 & 0 & - \\ - & 1 & - & - & 0 \\ 0 & 0 & 1 & - & - \\ 0 & 0 & - & 1 & - \\ - & - & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$X = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

- Broadcast two messages: (M1 + M2 + M5, M2 + M3 + M4)
- All receivers can reconstruct their desired message

Matrix Completion in $\mathbb{F}_2$?—State of the Art?

- No exact method in literature for matrix completion in $\mathbb{F}_2$ (!?)
- Heuristic pruning-based enumeration method in Esfahanizadeh, Lahuoti, and Hassibi, able to find (known) min rank solution for 7 by 7 instance every time in around 1 second.
- For 14 by 14 instance, in 30 min, they (sometimes) find rank 5 solution, sometimes find rank 6 solution.

MIP People Do It Exactly

Or at least up to floating point accuracy?

- We aim to build first(?) exact solver for this class of problems

MIP Formulations for Matrix Completion in $\mathbb{F}_2$

- Some sets we will use

$$\mathcal{I} := \{(u, v, z) \in \{0,1\}^{2r+1} \mid z = \oplus_{k=1}^{r} u_k v_k\}$$
$$\mathcal{P} := \{(y, z) \in \{0,1\}^{r+1} \mid z = \oplus_{k=1}^{r} y_k\}$$
$$\mathcal{M} := \{(u, v, y) \in \{0,1\}^{3r} \mid y_k = u_k v_k \ \forall k \in [r]\}$$

- Note that $\text{proj}_{u,v,z}(\mathcal{P} \cap \mathcal{M}) = \mathcal{I}$
- Matrix Completion in $\mathbb{F}_2$:

$$\min \sum_{(ij)\in\Omega} |X_{ij} - z_{ij}|$$
$$(u^i, v^j, z_{ij}) \in \mathcal{I}_{ij} \ \forall ij \in \Omega$$

- Note that $u^i, v^j \in \{0,1\}^r$

Writing $\mathcal{M}$ as MIP

- Everyone (at least at this meeting) knows how to write $\mathcal{M}$ as the set of {0,1}-points inside a polyhedron. ($\mathcal{M}$ is for McCormick.)

$$\mathcal{M} = \{(u, v, y) \in \{0,1\}^{3r} \mid y_k \leq u_k, y_k \leq v_k, y_k \geq u_k + v_k - 1 \ \forall k \in [r]\}$$

- Oktay told me that

$$\mathrm{LP}(\mathcal{M}) := \{(u, v, y) \in [0,1]^{3r} \mid y_k \leq u_k, y_k \leq v_k \\ y_k \geq u_k + v_k - 1 \ \forall k \in [r]\} = \mathrm{conv}(\mathcal{M})$$

- It is also true (by separability) that

$$\mathrm{conv}(\mathcal{P} \cap \mathcal{M}) = \mathrm{conv}(\mathcal{P}) \cap \mathrm{conv}(\mathcal{M}).$$

Writing $\mathcal{P}$ as MIP

- Consider the general integer set:

$$\mathcal{Z} := \{(y, z, t) \in \{0,1\}^{r+1} \times \mathbb{Z} \mid \sum_{k=1}^{r} y_k - 2t = z\}$$

- It is easy to see that $\mathcal{Z} = \mathcal{P}$
- So we have our "first" MILP formulation for matrix completion in $\mathbb{F}_2$:

$$\min \quad \sum_{(ij)\in\Omega} |X_{ij} - z_{ij}|$$

$$(u^i, v^j, y^{ij}) \in \mathcal{M}_{ij} \quad \forall ij \in \Omega$$
$$(y^{ij}, z_{ij}, t_{ij}) \in \mathcal{Z}_{ij} \quad \forall ij \in \Omega$$

Computational Experiments

- $X \in \{0,1\}^{10\times 10}$ will have $\mathbb{F}_2$-rank 4.
- Use MIP formulation to find "closest" rank r matrix for $r \leq 4$
- Let Ω be all matrix elements, and then start to (randomly) remove a fraction of the entries

Computational Results

% Missing	Rank	Time	Nodes	Opt
0	1	0.05	1	36
0	2	41.81	70237	24
0	3	7184.56	10437394	12
0	4	0.49	1	0
10	1	0.03	1	31
10	2	14.04	27757	17
10	3	320.59	996422	7
10	4	0.03	1	0
20	1	0.01	1	26
20	2	2.91	5872	14
20	3	4106.07	13393830	8
20	4	2.55	2430	0

Results are a Pig!

- 460 binary vars, 100 integer vars $>$ 10M nodes?

How to Improve?

- The LP relaxation of the parity condition:

$$\mathrm{LP}(\mathcal{Z}) := \{(y, z, t) \in [0,1]^{r+1} \times \mathbb{R}_+ \mid 2t = \sum_{i=1}^{r} y_i - z\}$$

  is very far from the convex hull of the true parity conditions:

$$\mathrm{proj}_{yz}\, \mathrm{LP}(\mathcal{Z}) \subset \mathrm{conv}(\mathcal{P})$$

- But lots is known about how to model parity conditions

Parity Polyhedra

$$P_E = \text{conv}\{x \in \{0,1\}^n \mid \sum_{i=1}^{n} x_i \text{ is even } \}$$

$$P_O = \text{conv}\{x \in \{0,1\}^n \mid \sum_{i=1}^{n} x_i \text{ is odd } \}$$

$$P_E = \{x \in [0,1]^n \mid \sum_{i \in S} x_i - \sum_{i \notin S} x_i \leq |S| - 1, \forall \text{ odd } S \subset [n]\}$$

$$P_O = \{x \in [0,1]^n \mid \sum_{i \in S} x_i - \sum_{i \notin S} x_i \leq |S| - 1, \forall \text{ even } S \subset [n]\}$$

- There are also small (even linear-size) extended formulations for P_E and P_O
- From these, and using disjunctive programming, we can give an extended formulation for $\text{conv}(\mathcal{P})$

One Extended Formulation for $\mathrm{conv}(\mathcal{P})$

- Let $D \in [0,1]^{3r+1}$ be the set of points satisfying bound constraints and the inequalities

$$\sum_{k\in S} y^o_k - \sum_{k\notin S} y^o_k \le (|S|-1)z \qquad \forall \text{ even } S \subseteq [r]$$

$$\sum_{k\in S} y^e_k - \sum_{k\notin S} y^e_k \le (|S|-1)(1-z) \qquad \forall \text{ odd } S \subseteq [r]$$

$$y_k = y^o_k + y^e_k \qquad \forall k \in [r]$$

$$y^o_k \le z \qquad \forall k \in [r]$$

$$y^e_k \le 1 - z \qquad \forall k \in [r]$$

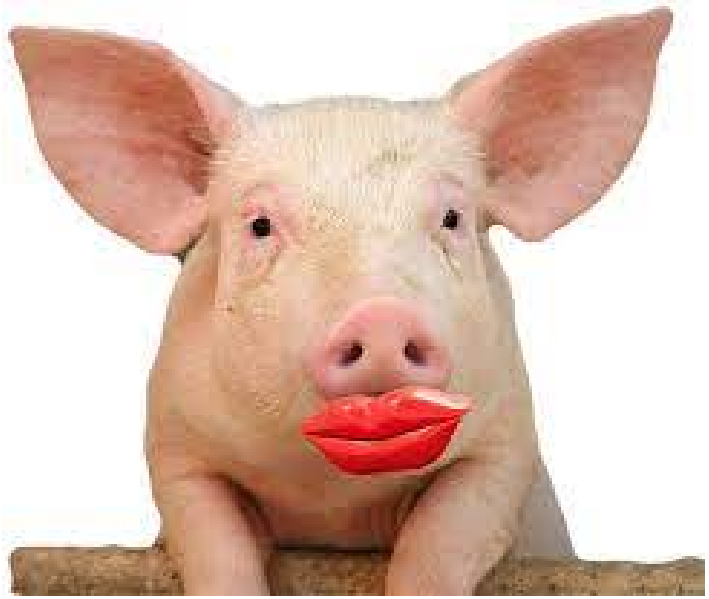
Thms:

$$\mathrm{conv}(\mathcal{P}) = \mathrm{proj}_{y,z}\, D \qquad \mathrm{conv}(\mathcal{P} \cap \mathcal{M}) = D \cap \mathrm{LP}(\mathcal{M})$$

MIP Formulation 2

$$\min \quad \sum_{(ij)\in\Omega} |X_{ij} - z_{ij}|$$

$$(u^i, v^j, y^{ij}) \in \mathcal{M}_{ij} \quad \forall (ij) \in \Omega$$
$$(y^{ij}, y^{o,ij}, y^{e,ij}, z_{ij}) \in D_{ij} \quad \forall (ij) \in \Omega$$
$$z_{ij} \in \{0, 1\} \quad \forall ij \in \Omega$$



Computational Results: MIP1 v MIP2

MIP	% Missing	Rank	Time	Nodes	Opt
1	0	2	41.81	70237	24
2	0	2	9.42	13746	24
1	0	3	7184.56	10437394	12
2	0	3	2137.15	1272534	12
1	10	2	14.04	27757	17
2	10	2	6.63	20296	17
1	10	3	320.59	996422	7
2	10	3	357.02	353021	7
1	20	2	2.91	5872	14
2	20	2	3.64	8927	14
1	20	3	4106.07	13393830	8
2	20	3	2199.89	2366186	8

Team Reactions

"Why do you all keep talking about putting lipstick on a pig?"

"Aunque la mona se vista de seda, mona se queda"

(You can dress a monkey in silk, but it's still a monkey)

Keep Trying!

- Can we directly model the set

$$\mathcal{I} = \{(u, v, z) \in \{0,1\}^{2r+1} \mid z = \oplus_{k=1}^{r} u_k v_k\}$$

  without using auxiliary variables?
- Yes! Let $\mathcal{T}$ be the set of all tri-partitions of $[r]$

$$\mathcal{T} := \{S \subseteq [r], Q \subseteq [r], T \subseteq [r] \mid S \cup Q \cup T = [r] \\ S \cap Q = \emptyset, S \cap T = \emptyset, Q \cap T = \emptyset\}$$

- Consider families of inequalities

$$z + u(S) + v(S) - u(Q) - v(T) \leq 2|S| \ \forall (S, Q, T) \in \mathcal{T} \text{ with } |S| \text{ even} \quad (1)$$

$$z - u(S) - v(S) + u(Q) + v(T) \geq 1 - 2|S| \ \forall \ (S, Q, T) \in \mathcal{T} \text{ with } |S| \text{ odd} \quad (2)$$

Theorems

Theorem

- These (exponentially many in r) inequalities give a direct formulation of $\mathcal{I}$:

$$\mathcal{F} = \{(u, v, z) \in \{0,1\}^{2r+1} \mid (1), (2)\}$$

- All inequalities are necessary

"Theorem"

- The LP relaxation of the set is the convex hull

$$\text{conv}(\mathcal{I}) = \{(u, v, z) \in [0,1]^{2r+1} \mid (1), (2)\}$$

- "Theorem" because Jim hasn't proved it yet

MIP Formulation 3

$$\min \quad \sum_{(ij)\in\Omega} |x_{ij} - z_{ij}|$$

$$(u^i, v^j, z_{ij}) \in \mathcal{I}_{ij} \quad \forall (ij) \in \Omega$$

Computational Results

MIP	% Missing	Rank	Time	Nodes	Opt
1	0	2	41.81	70237	24
2	0	2	9.42	13746	24
3	0	2	5.00	12588	24
1	0	3	7184.56	10437394	12
2	0	3	2137.15	1272534	12
3	0	3	1765.4	1962326	12
1	10	2	14.04	27757	17
2	10	2	6.63	20296	17
3	10	2	3.65	22560	17
1	10	3	320.59	996422	7
2	10	3	357.02	353021	7
3	10	3	188.81	332773	7
1	20	2	2.91	5872	14
2	20	2	3.64	8927	14
3	20	2	4.28	3357	14
1	20	3	4106.07	13393830	8
2	20	3	2199.89	2366186	8
3	20	3	381.94	645413	8

Discussion

- Frankly, the computational results are not where we want them to be.
- We can now only "reliably" solve linear index coding problems of sizes up to around 12 by 12.
- And worse, the "monkey in silk" formulation or the "pig in lipstick formulation" aren't typically any better than the "pig" formulation

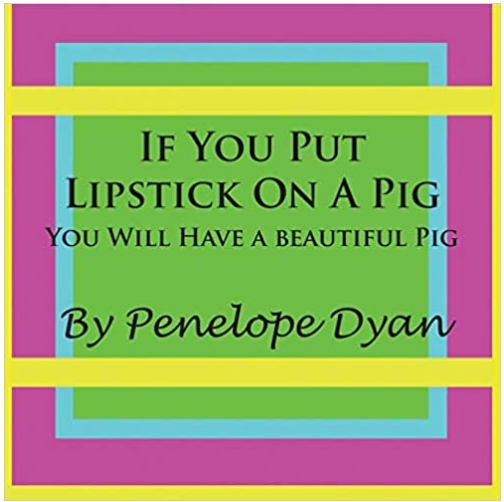
A Word on Separation

- We don't do it—Our computational results (to this point) just explicitly enumerate all inequalities
- However, separation of the inequalities is "trivial" (linear time)

Can we do more?

- MIP3 (Silk Monkey) formulation is

$$(u^i, v^j, z_{ij}) \in \text{conv}(\mathcal{I}_{ij}) \quad \forall (ij) \in \Omega$$
$$(u^i, v^j, z_{ij}) \in \{0,1\}^{dr+rn+|\Omega|}$$



- We know the intersection of the convex hulls
- If it were only true that

$$\text{conv}\left(\cap_{ij\in\Omega} \mathcal{I}_{ij}\right) = \cap_{ij\in\Omega} \text{conv}(\mathcal{I}_{ij})$$

we wouldn't need integer variables.

ChatGPT To the Rescue!

Prove that the convex hull of the intersection of sets is equal to the intersection of the sets' convex hulls

To prove that the convex hull of the intersection of sets is equal to the intersection of the sets' convex hulls, we need to show that both contain the same set of points.

Let S1, S2, ..., Sn be a collection of sets, and let C1, C2, ..., Cn be their respective convex hulls. We want to show that:

convex hull (S1 ∩ S2 ∩ ... ∩ Sn) = C1 ∩ C2 ∩ ... ∩ Cn

To prove this, we need to show that any point that belongs to one set also belongs to the other set.

First, let us consider the left-hand side of the equation. The convex hull of the intersection of sets, convex hull (S1 ∩ S2 ∩ ... ∩ Sn), is the smallest convex set that contains all the points in S1 ∩ S2 ∩ ... ∩ Sn.

ChatGPT Proves $\text{conv}(\cap_i(X_i)) = \cap \text{conv}(X_i)$

Since the intersection of sets is a subset of each set, any point in the intersection must be in each set. Therefore, the convex hull of the intersection contains all the points in each set's convex hull.

Now, let us consider the right-hand side of the equation. The intersection of the convex hulls, C1 ∩ C2 ∩ ... ∩ Cn, is the set of all points that belong to all the convex hulls.

Since each convex hull contains all the points in its corresponding set, any point that belongs to all the convex hulls must belong to the intersection of the sets. Therefore, the intersection of the convex hulls contains all the points in the convex hull of the intersection.

Combining these two facts, we can conclude that:

convex hull (S1 ∩ S2 ∩ ... ∩ Sn) ⊆ C1 ∩ C2 ∩ ... ∩ Cn

and

C1 ∩ C2 ∩ ... ∩ Cn ⊆ convex hull (S1 ∩ S2 ∩ ... ∩ Sn)

Hence, we have shown that the convex hull of the intersection of sets is equal to the intersection of the sets' convex hulls.

Next Steps: Two Rows of $\mathcal{U}$

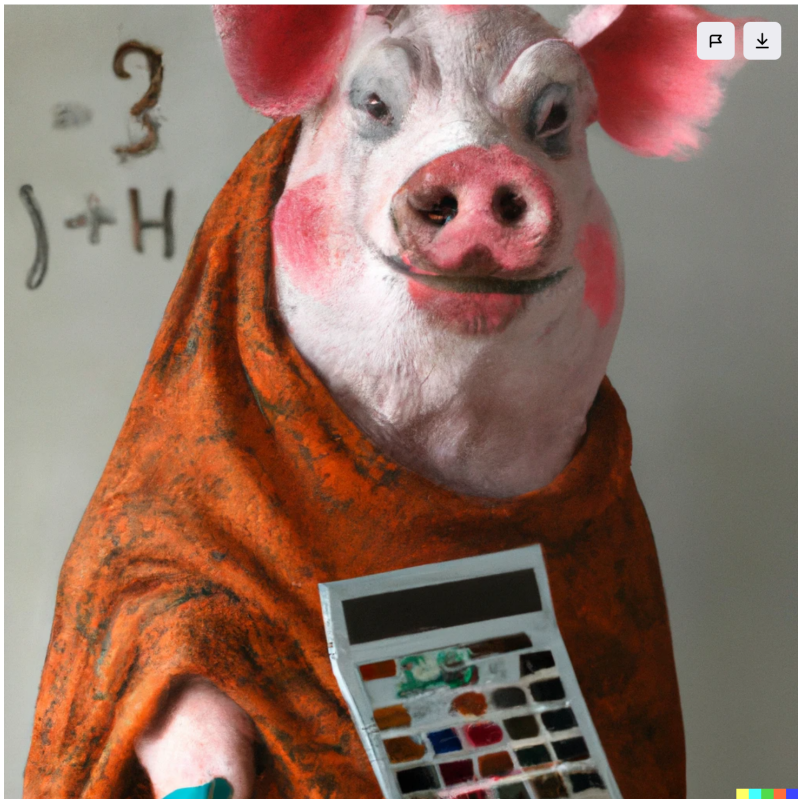
$$\mathcal{I}_2 = \{(u^1, u^2, v, z_1, z_2) \in \{0,1\}^{3r+2} \mid z_1 = \oplus_{k=1}^{r} u_k^1 v_k, z_2 = \oplus_{k=1}^{r} u_k^2 v_k\}$$

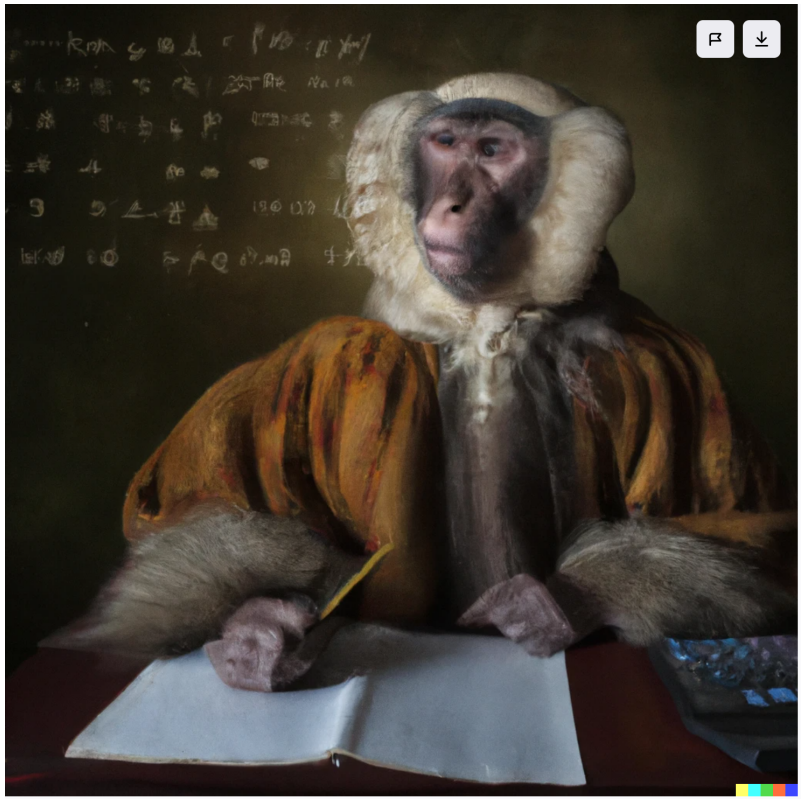
Always Leave the Audience Wanting More

- Sorry, those will have to wait until next time
- I actually think there is a good chance we will get inequalities for all of the rows and just one column.

Any Questions?

- Three MIP Formulation for Matrix Completion in $\mathbb{F}_2$





Another ChatGPT Apology Limerick

Write a limerick apologizing for a bad math talk

There once was a talk full of math,
But the speaker was quite in a bath,
The graphs were all skewed,
The theorems were crude,
I'm sorry for causing your wrath!