On constrained coding and contingency tables

Yuwen Huang

Department of Information Engineering
and Department of Computer Science and Engineering
The Chinese University of Hong Kong
yuwen.huang@ieee.org

A talk at School of Cyber Science and Engineering
Southeast University

Today's talk is based on a **joint work with Professor Pascal O. Vontobel** at CUHK:

Y. Huang, P. O. Vontobel, "The Bethe partition function and the SPA for factor graphs based on homogeneous real stable polynomials,"
2024 IEEE International Symposium on Information Theory (ISIT),
3029-3034.

Outline

Motivation

A setup based on binary matrices with prescribed row sums and column sums

Graphical-model-based approximation method

Main results

A more general setup

Main results for a more general setup

Numerical results

Future works and connection to other works

Outline

- **Motivation**

A setup based on binary matrices with prescribed row sums and column sums

Graphical-model-based approximation method

Main results

A more general setup

Main results for a more general setup

Numerical results

Future works and connections to other works

Contingency Table

A contingency table is a type of **table** in a **matrix format** that displays the **multivariate frequency distribution** of the variables.

Example of a 2×2 contingency table

Suppose we have two **categorical variables**:

1. **gender** (male or female),
2. **handedness** (right or left handed).

Conduct a **simple random sampling** and obtain a **size 100 data** that is **summarized** by the **following table**:

	Right-hand	Left-hand	Total
Male	50	6	56
Female	40	4	44
Total	90	10	100

Contingency Table

Contingency tables are **fundamental** objects across the sciences including

1. **statistics** [FLL17][1],
2. **combinatorics and graph theory** [Bar09][2],
3. **discrete geometry and combinatorial optimization** [DK14][3],
4. **algebraic and enumerative combinatorics** [PP20][4].

[1] M. W. Fagerland, S. Lydersen and P. Laake, Statistical analysis of contingency tables, CRC Press, 2017.

[2] A. Barvinok, "Asymptotic estimates for the number of contingency tables, integer flows, and volumes of transportation polytopes," Int. Math. Res. Notices, 2009.

[3] J. A. De Loera and E. D. Kim, "Combinatorics and geometry of transportation polytopes: an update," Discrete geometry and algebraic combinatorics, 2014

[4] I. Pak and G. "Panova, Bounds on Kronecker coefficients via contingency tables", Linear Algebra Appl., 2020.

Two-dimensional constrained coding

Why do we need **constrained coding**?

Answer from [MMH01][5]:

- In most **data recording** systems and many **data communication systems**, some sequences are **more prone to error** than others.
- In order to **reduce the likelihood of error**, we need to impose **constraints on the sequences** that are allowed to be **recorded or transmitted**.
- Given such constraints, it is then necessary to **encode arbitrary user sequences** into sequences that **satisfy the constraint**.

[5] B. H. Marcus, R. M. Roth, and P H. Siegel, "An introduction to coding for constrained systems," Lecture notes, 2001.

Two-dimensional constrained coding

Why do we consider **two-dimensional constrained coding**?

Answer from [N+23][6]:

1. In **optical recording**, *e.g.*, recording data in a **CD**, the recording device is a **surface**, and the recording data is in **two dimensional**.
2. In **resistive memory technologies**, the memory cell is a passive **two-terminal device** that can be **both read and written** over a simple crossbar structure.
3. These two models offer a **huge density advantage**, however, face new reliability issues and introduce **two-dimensional constraints**.

[6] T. T. Nguyen, K. Cai, H. M. Kiah, K. A. S. Immink, and Y. M. Chee, "Two-dimensional RC/SW constrained codes bounded weight and almost balanced weight," IEEE Trans. Inf. Theory, 2023.

Two-dimensional constrained coding

In this talk, we discuss **two-dimensional constrained binary coding** where the constrains are specified by **the column sum and the row sums**.

An introductory example

Consider the set of all **binary** 3×3 **matrices**.

We want to know the number of **binary** 3×3 matrices with **row sums** and **column sums** equaling **two**.

The following are **example binary** 3×3 matrices:

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

Two-dimensional constrained coding

An introductory example

Consider the set of all **binary** 3×3 **matrices**.

We want to know the number of **binary** 3×3 matrices with **row sums** and **column sums** equaling **two**.

The following are **example binary** 3×3 matrices:

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}}_{\times}, \quad \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}}_{\checkmark}, \quad \underbrace{\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}}_{\checkmark}.$$

The number of such matrices is **3!**.

Two-dimensional constrained coding

An introductory example

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}, \qquad \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

- These binary matrices can be viewed as **binary contingency tables** of size 3×3 with **row sums** and **column sums** equaling **two**.
- The number of such **binary contingency tables** is $3!$.

We focus on **counting the number** of **binary contingency tables** / **two-dimensional binary codes** with prescribed row sums and column sums.

This counting problem is **non-trivial** in general.

Two-dimensional constrained binary codes and contingency tables

A great deal of effort is made to **approximate** the number of **contingency tables with prescribed row sums and column sums**.

A **variety of tools** have been developed in **different areas**, such as

1. the **traditional and probabilistic divide-and-conquer** [DZ15+][7],
2. the **Markov chain Monte Carlo (MCMC)** algorithms [C+06][8],
3. **approximation algorithms** [B+10][9].

---

[7] S. DeSalvo and J. Y. Zhao, "Random sampling of contingency tables via probabilistic divide-and-conquer," Comput. Statist., 2020.

[8] M. Cryan, M. Dyer, L. A. Goldberg, M. Jerrum and R. Martin, "Rapidly mixing Markov chains for sampling contingency tables with a constant number of rows," SIAM J. Comput., 2006.

[9] A. Barvinok, Z. Luria, A. Samorodnitsky and A. Yong, "An approximation algorithm for counting contingency tables," Random Structures Algorithms, 2010

Outline

Setup

Definition

1. $[n] \triangleq \{1, 2, \ldots, n\}$ for $n \in \mathbb{Z}_{\geq 1}$ and $[m] \triangleq \{1, 2, \ldots, m\}$ for $m \in \mathbb{Z}_{\geq 1}$.
2. $\boldsymbol{x} = \big(x(i,j)\big)_{i \in [n], j \in [m]}$: a **$\{0,1\}$-valued matrix** of size $n \times m$.
3. For the ***i*-th row** $\boldsymbol{x}(i,:)$, we introduce an integer r_i and impose a **constraint** on the **row sum**:
$$\mathcal{X}_{r_i} = \left\{ \boldsymbol{x}(i,:) \,\middle|\, \sum_{j \in [m]} x(i,j) = r_i \right\}.$$
4. For the ***j*-th column** $\boldsymbol{x}(:,j)$, we introduce an integer c_j and impose a **constraint** on the **column sum**:
$$\mathcal{X}_{c_j} = \left\{ \boldsymbol{x}(:,j) \,\middle|\, \sum_{i \in [n]} x(i,j) = c_j \right\}.$$

Setup

Definition

5. The set of **valid configurations** is defined to be

$$\mathcal{C} \triangleq \left\{ \boldsymbol{x} \in \{0,1\}^{n\times n} \;\middle|\; \begin{array}{c} \boldsymbol{x}(i,:) \in \mathcal{X}_{r_i},\ \forall i \in [n], \\ \boldsymbol{x}(:,j) \in \mathcal{X}_{c_j},\ \forall j \in [m] \end{array} \right\},$$

the set of **binary matrices** such that the **i-th row sum** is r_i and the **j-th column sum** is c_j.

6. We want to compute the number of the **valid configurations** $|\mathcal{C}|$.

Outline

Motivation

A setup based on binary matrices with prescribed row sums and column sums

- **Graphical-model-based approximation method**

Main results

A more general setup

Main results for a more general setup

Numerical results

Future works and connection to other works

Graphical-model-based approximation method

Main idea

1. Define a **standard factor graph (S-FG)** N whose partition function equals

$$Z(\mathsf{N}) = |\mathcal{C}|.$$

2. Run the **sum product algorithm (SPA)**, a.k.a. **belief propagation** (BP), on the S-FG N to compute the **Bethe approximation of** $|\mathcal{C}|$, denoted by $Z_{\mathrm{B}}(\mathsf{N})$.

Graphical-model-based approximation method

Example

Consider $n = m = 3$ **and** $r_i = c_j = 2$, *i.e.*, $\boldsymbol{x} \in \{0,1\}^{3\times 3}$.

The i-**th row** $\boldsymbol{x}(i,:) \in \mathcal{X}_{r_i}$ and the j-**th column** $\boldsymbol{x}(:,j) \in \mathcal{X}_{c_j}$, where

$\mathcal{X}_{r_i} = \{(1,1,0),(0,1,1),(1,0,1)\}, \quad \mathcal{X}_{c_j} = \{(1,1,0)^\mathsf{T},(0,1,1)^\mathsf{T},(1,0,1)^\mathsf{T}\}.$

1. The **local functions**:

$$f_{\mathrm{l},i}\big(\boldsymbol{x}(i,:)\big) \triangleq \begin{cases} 1 & \text{if } \boldsymbol{x}(i,:) \in \mathcal{X}_{r_i} \\ 0 & \text{otherwise} \end{cases}, \quad f_{\mathrm{r},j}\big(\boldsymbol{x}(:,j)\big) \triangleq \begin{cases} 1 & \text{if } \boldsymbol{x}(:,j) \in \mathcal{X}_{c_j} \\ 0 & \text{otherwise} \end{cases}.$$

2. The **support** of the local functions:

$$\mathcal{X}_{f_{\mathrm{l},i}} \triangleq \big\{\boldsymbol{x}(i,:) \in \{0,1\}^3 \;\big|\; f_{\mathrm{l},i}\big(\boldsymbol{x}(i,:)\big) > 0\big\} = \mathcal{X}_{r_i},$$
$$\mathcal{X}_{f_{\mathrm{r},j}} \triangleq \big\{\boldsymbol{x}(:,j) \in \{0,1\}^3 \;\big|\; f_{\mathrm{r},j}\big(\boldsymbol{x}(:,j)\big) > 0\big\} = \mathcal{X}_{c_j}.$$

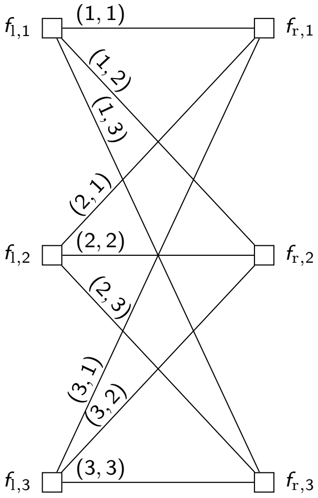
Graphical-model-based approximation method

3. The $\{0,1\}$-valued **global function**:

$$g(\boldsymbol{x}) \triangleq f_{\mathrm{l},1}\big(x(1,1), x(1,2), x(1,3)\big) \cdot f_{\mathrm{l},2}\big(x(2,1), x(2,2), x(2,3)\big) \cdots f_{\mathrm{r},2}\big(x(1,2), x(2,2), x(3,2)\big) \cdot f_{\mathrm{r},3}\big(x(1,3), x(2,3), x(3,3)\big).$$

The **previously defined** set of **valid configurations** is equal to the **support** of the global function:

$$\mathcal{C} = \left\{ \boldsymbol{x} \in \{0,1\}^{3\times 3} \;\middle|\; g(\boldsymbol{x}) > 0 \right\}.$$



4. The **partition function**:

$$Z(\mathsf{N}) \triangleq \sum_{\boldsymbol{x} \in \{0,1\}^{3\times 3}} g(\boldsymbol{x}) = |\mathcal{C}|.$$

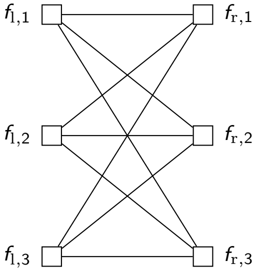
Graphical-model-based approximation method

5. The **Bethe approximation** of the partition function, *i.e.*, the **Bethe partition function**, is defined to be

$$Z_{\mathrm{B}}(\mathsf{N}) \triangleq \exp\Big(- \min_{\boldsymbol{\beta} \in \mathcal{L}(\mathsf{N})} F_{\mathrm{B}}(\boldsymbol{\beta}) \Big),$$

   where F_{B} is the **Bethe free energy (BFE)** function,

   where $\mathcal{L}(\mathsf{N})$ is the **local marginal polytope** (LMP) (see, *e.g.*,[WJ08][10]).



6. Then we run the **sum-product algorithm (SPA)**, a.k.a. **belief propagation (BP)**, on the S-FG N to get $Z_{\mathrm{B}}(\mathsf{N})$.

[10] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," Foundation and Trends in Machine Learning, 2008.

Outline

Main results

1. The **projection** of the **local marginal polytope** on the **edges** in N **equals** conv($\mathcal{C}$).
   (For general S-FGs, this projection is a **relaxation** of conv($\mathcal{C}$), *i.e.*, conv($\mathcal{C}$) is a **strict subset** of this **projection**.)
2. For the **typical case** where N has an **SPA fixed point** consisting of **positive-valued messages only**,
   the SPA finds the value of $Z_{\mathrm{B}}(\mathrm{N})$ **exponentially fast**.
3. The **BFE function** has some **convexity properties**.

Comments

- A **generalization** of parts of the results in [Von13][11].
- Even though the S-FG has a **non-trivial cyclic structure**,
  the SPA has **a good performance**.

[11] P. O. Vontobel, The Bethe permanent of a nonnegative matrix," IEEE Trans. Inf. Theory, 2013.

Main results

Comments

For the setup where $n = m$, $r_i = 1$, and $c_j = 1$, it holds that

- $\mathcal{C} = \{\boldsymbol{x} \mid \boldsymbol{x}$ is a **permutation matrix** of size n-by-$n\}$
- The **projection** of the **LMP** on the **edges** equals the set of **doubly stochastic matrices** of size n-by-n.

Birkhoff–von Neumann theorem

The set of **doubly stochastic matrices** of size n-by-n is the **convex hull** of the set of the **permutation matrices** of size n-by-n.

The main result that $\mathrm{conv}(\mathcal{C})$ equals the **projection** of the **LMP** on the **edges** for our considered S-FG, can be viewed as a **generalization**.

Outline

Motivation

A setup based on binary matrices with prescribed row sums and column sums

Graphical-model-based approximation method

Main results

- **A more general setup**

Main results for a more general setup

Numerical results

Future works and connection to other works

A more general setup

An example S-FG

Consider $n = m = 3$ and $r_i = c_j = 2$. Then

$$f_{1,i}\big(\boldsymbol{x}(i,:)\big) = \begin{cases} 1 & \text{if } \boldsymbol{x}(i,:) \in \{(1,1,0),(0,1,1),(1,0,1)\} \\ 0 & \text{otherwise} \end{cases},$$

which corresponds to a **multi-affine homogeneous real stable (MAHRS) polynomial** w.r.t. the **indeterminates** in $\boldsymbol{L} \triangleq \big(L_1, L_2, L_3\big) \in \mathbb{C}^3$:

$$p_i(\boldsymbol{L}) = \sum_{\boldsymbol{x}(i,:)\in\{0,1\}^3} f_{1,i}\big(\boldsymbol{x}(i,:)\big) \cdot \prod_{j\in[3]} \big(L_j\big)^{\boldsymbol{x}(i,j)}$$

$$= L_1 \cdot L_2 + L_2 \cdot L_3 + L_1 \cdot L_3.$$

Remark

- For details of **real stable polynomials**, see, *e.g.*,

  S. O. Gharan, "Course notes of polynomial paradigm in algorithm design," 2020, Lecture 3.

Consider a **more general** setup where **each local function** is defined based on a **(possibly different) MAHRS polynomial**.

Do the previous results **hold** in this **more general setup**?

Yes!

An MAHRS Polynomials-based S-FG

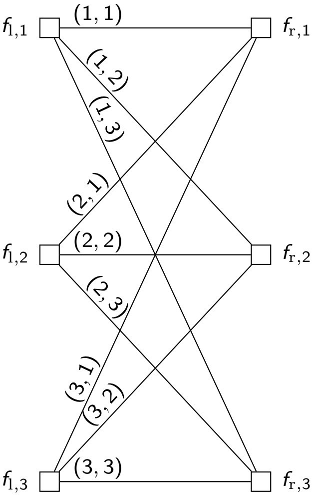
The standard factor graph (S-FG) N consists of

1. **edges**: $(1,1),(1,2),\ldots,(3,3)$;
2. **Binary** matrix

$$\boldsymbol{x} \triangleq \begin{pmatrix} x(1,1) & x(1,2) & x(1,3) \\ x(2,1) & x(2,2) & x(2,3) \\ x(3,1) & x(3,2) & x(3,3) \end{pmatrix}.$$

3. **Nonnegative-valued** local functions

   $f_{\mathrm{l},1},\ldots,f_{\mathrm{r},3}$;



An MAHRS Polynomials-based S-FG

6. The **local function** $f_{\mathrm{l},i}$ on the **LHS** is defined to be the mapping:

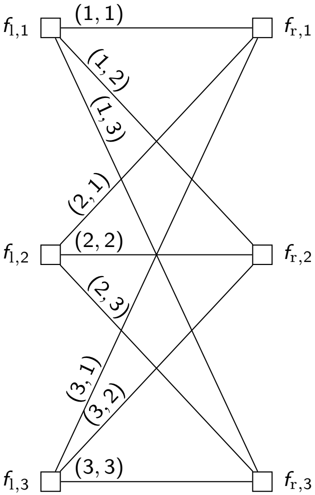
$$\{0,1\}^3 \to \mathbb{R}_{\geq 0}, \quad \boldsymbol{x}(i,:) \mapsto f_{\mathrm{l},i}\big(\boldsymbol{x}(i,:)\big)$$

such that it corresponds to an **MAHRS polynomial**.

7. The **support** of $f_{\mathrm{l},i}$:

$$\mathcal{X}_{f_{\mathrm{l},i}} \triangleq \left\{ \boldsymbol{x}(i,:) \in \{0,1\}^3 \;\middle|\; f_{\mathrm{l},i}\big(\boldsymbol{x}(i,:)\big) > 0 \right\}.$$

8. A **similar idea** in the definitions of $f_{\mathrm{r},j}$ and $\mathcal{X}_{f_{\mathrm{r},j}}$ on the **RHS**.



An MAHRS Polynomials-based S-FG

9. The **nonnegative-valued global function**:

$$g(\boldsymbol{x}) \triangleq f_{\mathrm{l},1}\big(\boldsymbol{x}(1,:)\big) \cdot f_{\mathrm{l},2}\big(\boldsymbol{x}(2,:)\big) \cdot f_{\mathrm{l},3}\big(\boldsymbol{x}(3,:)\big) \cdot f_{\mathrm{r},1}\big(\boldsymbol{x}(:,1)\big) \cdot f_{\mathrm{r},2}\big(\boldsymbol{x}(:,2)\big) \cdot f_{\mathrm{r},3}\big(\boldsymbol{x}(:,3)\big).$$

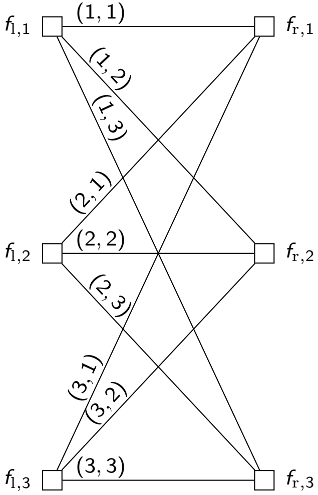
10. The set of **valid configurations**:

$$\mathcal{C} \triangleq \left\{ \boldsymbol{x} \in \{0,1\}^{3\times 3} \;\middle|\; g(\boldsymbol{x}) > 0 \right\},$$

which is also the **support** of the **global function**.

11. The **partition function**:

$$Z(\mathsf{N}) \triangleq \sum_{\boldsymbol{x}\in\mathcal{C}} g(\boldsymbol{x}).$$



Outline

Known results

Consider an **S-FG** N where **each local function** is defined based on a (possibly different) **MAHRS** polynomial.

Remarks

- Exactly computing $Z(\mathsf{N})$ is a **#P-complete problem** in general. (**#P-complete problem** is the set of the **counting problems** associated with the **decision problems** in the class **NP**.)

- **Run the SPA** to find the value of the **Bethe partition function** $Z_{\mathrm{B}}(\mathsf{N})$ that **approximates** $Z(\mathsf{N})$.

- [Theorem 3.2, SV19][12]: $Z_{\mathrm{B}}(\mathsf{N}) \leq Z(\mathsf{N})$.

[12] D. Straszak and N. K. Vishnoi, "Belief propagation, Bethe approximation, and polynomials," IEEE Trans. Inf. Theory, 2019.

Known results

Consider an **S-FG** N where **each local function** is defined based on a (possibly different) **MAHRS** polynomial.

Remarks

- **Other real-stable-polynomial-based approximation of $Z(\mathsf{N})$** [Gur15][13] and [Bra23][14].

[13] L. Gurvits, "Boolean matrices with prescribed row/column sums and stable homogeneous polynomials: Combinatorial and algorithmic applications," Inform. and Comput., 2015.

[14] P. Brändén, J. Leake, and I. Pak, "Lower bounds for contingency tables via Lorentzian polynomials," Israel J. Math., 2023.

Main results

Consider an **S-FG** N where **each local function** is defined based on a (possibly different) **MAHRS** polynomial.

- The **support** $\mathcal{X}_{f_{\mathrm{l},i}}$ on the LHS corresponds to **a set of bases of a matroid** [Brä07][15].
- The support of the **product** of the **local functions** on the **LHS** is $\{\mathcal{X}_{f_{\mathrm{l},1}} \times \mathcal{X}_{f_{\mathrm{l},2}} \times \cdots \times \mathcal{X}_{f_{\mathrm{l},n}}\}$.
- Similarly for the local functions and the **support** on the **RHS**.
- The support of the **global function** equals the **intersection** of the bases of **matroids**:

$$\mathcal{C} = \{\mathcal{X}_{f_{\mathrm{l},1}} \times \mathcal{X}_{f_{\mathrm{l},2}} \times \cdots \times \mathcal{X}_{f_{\mathrm{l},n}}\} \bigcap \{\mathcal{X}_{f_{\mathrm{r},1}} \times \mathcal{X}_{f_{\mathrm{r},2}} \times \cdots \times \mathcal{X}_{f_{\mathrm{r},m}}\}$$

[15] P. Brändén, "Polynomials with the half-plane property and matroid theory," Adv. Math., 2007.

Main results

1. The **convex hull** $\text{conv}(\mathcal{C})$ is the **projection of the LMP** on the **edges**. (Based on results on **intersection of matroids** (see, *e.g.*, [Oxl11][16]).)

2. For the typical case where the S-FG has an SPA **fixed point** consisting of **positive-valued messages only**, the SPA finds the value of $Z_{\mathrm{B}}(\mathsf{N})$ **exponentially fast**.
   (Based on the properties of **real stable polynomials** in [Brä07][17].)

[16] J. Oxley, Matroid Theory, Oxford University Press, 2011.

[17] P. Brändén, "Polynomials with the half-plane property and matroid theory," Adv. Math., 2007.

Main results

3. The **Bethe free energy function** F_{B} has some **convexity properties**.

The proof of the convexity is **new**.

This result is based on the **dual** form of $Z_{\mathrm{B}}(\mathsf{N})$ in the following two papers:

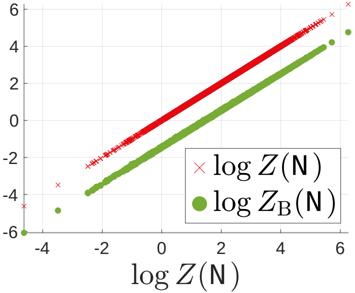
- D. Straszak and N. K. Vishnoi, "Belief propagation, Bethe approximation, and polynomials," IEEE Trans. Inf. Theory, 2019.
- N. Anari and S. O. Gharan, "A generalization of permanent inequalities and applications in counting and optimization," Adv. Math., 2021.

Outline

Numerical results

Setup

- We first consider the case $n = m = 6$ **and** $r_i = c_j = 2$.
- We independently randomly generate 3000 **instances** of N.



Observation

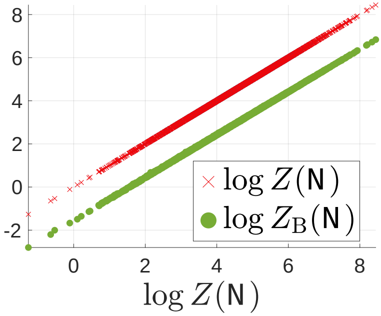
- $Z_{\mathrm{B}}(\mathsf{N}) \leq Z(\mathsf{N})$ ([Theorem 3.2, SV19][18]).
- $Z_{\mathrm{B}}(\mathsf{N})$ provides a **good estimate** of $Z(\mathsf{N})$ in this case.

[18] D. Straszak and N. K. Vishnoi, "Belief propagation, Bethe approximation, and polynomials," IEEE Trans. Inf. Theory, 2019.

Numerical results

Setup

Consider **the same setup** as the previous case, but with $n = m = 6$ **replaced** by $n = m = 7$.



Observation

We can make **similar observations**.

Outline

Motivation

A setup based on binary matrices with prescribed row sums and column sums

Graphical-model-based approximation method

Main results

A more general setup

Main results for a more general setup

Numerical results

- **Future works and connection to other works**

Future work

- Consider a **more general** S-FG, where each local function corresponds to a **more general** polynomial.
- Prove the **convergence** of the SPA for a **more general** S-FG.

Connection to other works

Works on **polynomial approaches** to approximate **partition functions**:

- L. Gurvits, "Unleashing the power of Schrijver's permanental inequality with the help of the Bethe approximation," Elec. Coll. Comp. Compl., 2011.
- D. Straszak and N. K. Vishnoi, "Belief propagation, Bethe approximation, and polynomials," IEEE Trans. Inf. Theory, 2019.
- N. Anari and S. O. Gharan, "A generalization of permanent inequalities and applications in counting and optimization," Adv. Math., 2021.

Connection to other works

Works on the properties of **real stable** polynomials and the **partition functions**.

- P. Brändén, "The Lee-Yang and Pólya-Schur programs. I. Linear operators preserving stability," Amer. J. Math., 2014.
- J. Borcea and P. Brändén, "The Lee-Yang and Pólya-Schur programs. II. Theory of stable polynomials and applications," Commun. Pure Appl. Math., 2009.
- J. Borcea, P. Brändén, and T. M. Liggett, "Negative dependence and the geometry of polynomials," J. Amer. Math. Soc., 2009.

Thank you!

Yuwen Huang gratefully acknowledges useful discussions with Prof. Jonathan Leake from the University of Waterloo.