Proving Inequalities in Information Theory:

Theory, Computational Challenges and Scalable Solutions

Lin Ling, Ph.D. Candidate

Department of Computer Science, City University of Hong Kong

Table of contents

Information Theoretic Inequalities

Shannon's Information Measures:

- entropy: $H(A)$

Information Theoretic Inequalities

Shannon's Information Measures:

- entropy: $H(A)$
- mutual information: $I(A;B)$

Information Theoretic Inequalities

Shannon's Information Measures:

- entropy: $H(A)$
- mutual information: $I(A;B)$
- conditional entropy: $H(A|B)$
- conditional mutual information: $I(A;B|C)$

Information Theoretic Inequalities

Shannon's Information Measures:

- entropy: $H(A)$
- mutual information: $I(A;B)$
- conditional entropy: $H(A|B)$
- conditional mutual information: $I(A;B|C)$

Information expression: linear combination of information measures involving finite number of random variables.

Information Theoretic Inequalities

Shannon's Information Measures:

- entropy: $H(A)$
- mutual information: $I(A;B)$
- conditional entropy: $H(A|B)$
- conditional mutual information: $I(A;B|C)$

Information expression: linear combination of information measures involving finite number of random variables.

Information inequality: $f \geq c$, where f is an information expression and c is a constant.

Information Theoretic Inequalities

Some information inequalities are always true. e.g.

$$H(A, B, C) \geq I(A; B)$$

, while some of them may not always be true. e.g.

$$H(A, B, C) \geq I(A; B) + I(B; C)$$

Information Theoretic Inequalities

Some information inequalities are always true. e.g.

$$H(A, B, C) \geq I(A; B)$$

, while some of them may not always be true. e.g.

$$H(A, B, C) \geq I(A; B) + I(B; C)$$

Given an information theoretic inequality, is there an algorithm to prove or disprove it automatically?

The ITIP Framework

Information Expressions in Canonical Form

Any information expression can be represented as a linear combination of entropies and joint-entropies. e.g.:

$$
\begin{aligned}
H(A|B) &= H(A,B) - H(B) \\
I(A;B) &= H(A) + H(B) - H(A,B) \\
I(A;B|C) &= H(A,C) + H(B,C) - H(A,B,C) - H(C)
\end{aligned}
$$

For n random variables, there are $k = 2^n - 1$ entropies/joint-entropies.

Information Expressions in Canonical Form

Any information expression can be represented as a linear combination of entropies and joint-entropies. e.g.:

$$
\begin{aligned}
H(A|B) &= H(A,B) - H(B) \\
I(A;B) &= H(A) + H(B) - H(A,B) \\
I(A;B|C) &= H(A,C) + H(B,C) - H(A,B,C) - H(C)
\end{aligned}
$$

For n random variables, there are $k = 2^n - 1$ entropies/joint-entropies.

e.g., for $n = 3$, we have:

$$
H(A),\ H(B),\ H(C),\ H(A,B),\ H(A,C),\ H(B,C),\ H(A,B,C)
$$

Information Expressions in Canonical Form

This inspires us to express information inequalities in form of $b^T h \geq 0$, where $h \in \mathbb{R}^k$. e.g.

$$H(A|B) \geq I(A;B) \Longrightarrow 2H(A,B) - 2H(B) - H(A) \geq 0$$

H(A)	H(B)	H(AB)
-1	-2	2

$$\Longrightarrow b^T h \geq 0,$$

where

$$b = \begin{bmatrix} -1 & -2 & 2 \end{bmatrix}^T$$

Elemental Inequalities

A valid h has to ensure nonnegativity of all information measures.

e.g., for $n = 2$, we must have

$$\begin{aligned} H(A|B) &= H(A,B) - H(B) && \geq 0 \\ H(B|A) &= H(A,B) - H(A) && \geq 0 \\ I(A;B) &= H(A) + H(B) - H(A,B) && \geq 0 \end{aligned}$$

Elemental Inequalities

A valid h has to ensure nonnegativity of all information measures.

e.g., for $n = 2$, we must have

$$\begin{aligned} H(A|B) &= H(A,B) - H(B) && \geq 0 \\ H(B|A) &= H(A,B) - H(A) && \geq 0 \\ I(A;B) &= H(A) + H(B) - H(A,B) && \geq 0 \end{aligned}$$

This is equivalent to $Dh \geq 0$, where

$$D = \begin{bmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & -1 \end{bmatrix}$$

Elemental Inequalities

For n random variables, there are

$$m = n + \binom{n}{2} 2^{n-2}$$

elemental inequalities to satisfy, thus $D \in \mathbb{R}^{m \times n}$

Linear Programming Formulation

From the above discussion, we notice that to prove or disprove a given information inequality, we can calculate the lower bound of $b^T h$. This can be formulated into a simple LP:

$$\begin{aligned} \min \quad & p = b^T h \\ \text{s.t.} \quad & Dh \geq 0 \end{aligned}$$

If $p^* \geq 0$, we know the corresponding inequality is always true, otherwise we conclude that it does not always hold.

Proof/Disproof Construction

Wait a minute... What this does is just verification, not proving. What we want is human-readable proof/disproof.

Proof/Disproof Construction

Consider the dual problem:

$$\begin{aligned} \max \quad & 0 \\ \text{s.t.} \quad & b - D^T \lambda = 0 \\ & \lambda \geq 0 \end{aligned}$$

Assume the problem is feasible. Let the optimal solution be λ^*, we know $b = D^T \lambda^*$

Proof/Disproof Construction

Consider the dual problem:

$$\begin{aligned} \max \quad & 0 \\ \text{s.t.} \quad & b - D^T\lambda = 0 \\ & \lambda \geq 0 \end{aligned}$$

Assume the problem is feasible. Let the optimal solution be λ^*, we know $b = D^T\lambda^*$

Notice that, for **ANY** feasible h, we have

$$b^T h = \lambda^{*T} D h \geq 0$$

Proof/Disproof Construction

Consider the dual problem:

$$\begin{aligned} \max \quad & 0 \\ \text{s.t.} \quad & b - D^T\lambda = 0 \\ & \lambda \geq 0 \end{aligned}$$

Assume the problem is feasible. Let the optimal solution be λ^*, we know $b = D^T\lambda^*$

Notice that, for **ANY** feasible h, we have

$$b^T h = \lambda^{*T} D h \geq 0$$

This is our proof!

Toy Example

Let's prove $2H(A,B) >= H(A) + H(B) \Longrightarrow b = \begin{bmatrix} -1 & -1 & 2 \end{bmatrix}^T$,

$$D = \begin{bmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & -1 \end{bmatrix}$$

$$H(A|B) = H(A,B) - H(B) \geq 0$$
$$H(B|A) = H(A,B) - H(A) \geq 0$$
$$I(A;B) = H(A) + H(B) - H(A,B) \geq 0$$

Toy Example

Let's prove $2H(A,B) >= H(A) + H(B) \Longrightarrow b = \begin{bmatrix} -1 & -1 & 2 \end{bmatrix}^T$,

$$D = \begin{bmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & -1 \end{bmatrix}$$

$$H(A|B) = H(A,B) - H(B) \geq 0$$
$$H(B|A) = H(A,B) - H(A) \geq 0$$
$$I(A;B) = H(A) + H(B) - H(A,B) \geq 0$$

Solve that $\lambda^* = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^T$

Toy Example

Let's prove $2H(A,B) >= H(A) + H(B) \Longrightarrow b = \begin{bmatrix} -1 & -1 & 2 \end{bmatrix}^T$,

$$D = \begin{bmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & -1 \end{bmatrix}$$

$$H(A|B) = H(A,B) - H(B) \geq 0$$
$$H(B|A) = H(A,B) - H(A) \geq 0$$
$$I(A;B) = H(A) + H(B) - H(A,B) \geq 0$$

Solve that $\lambda^* = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^T$

Proof:

$$\begin{aligned} & 2H(A,B) - H(A) - H(B) \\ = & (H(A,B) - H(B)) + (H(A,B) - H(A)) \\ = & H(A|B) + H(B|A) \geq 0 \end{aligned}$$

Disproof Construction

The primal would be unbounded below if the inequality is not provable.

Disproof Construction

The primal would be unbounded below if the inequality is not provable.

There exists a subset of entries of h^* that's $\infty \Longrightarrow$ Add an upper bound to h

e.g. let $H(X_1, X_2, \cdots, X_n) = 1$.

$$\begin{aligned} \min \quad & b^T h \\ \text{s.t.} \quad & Dh \geq 0 \\ & e^T h = 1 \end{aligned}$$

, where $e = \begin{bmatrix} 0, 0, \cdots, 1 \end{bmatrix}^T$

$$\begin{aligned} \max \quad & -\mu \\ \text{s.t.} \quad & b - D^T \lambda + \mu e = 0 \\ & \lambda \geq 0 \end{aligned}$$

The Computational Challenge

D Matrix Size

$D \in \mathbb{R}^{m \times k}$, where $m = n + \binom{n}{2} 2^{n-2}$ and $k = 2^n - 1$

D Matrix Size

$D \in \mathbb{R}^{m \times k}$, where $m = n + \binom{n}{2} 2^{n-2}$ and $k = 2^n - 1$

The bad news: The size of D grows exponentially!

D Matrix Size

$D \in \mathbb{R}^{m \times k}$, where $m = n + \binom{n}{2} 2^{n-2}$ and $k = 2^n - 1$

The bad news: The size of D grows exponentially!

The good news: D is very sparse

D Matrix Size

$D \in \mathbb{R}^{m \times k}$, where $m = n + \binom{n}{2} 2^{n-2}$ and $k = 2^n - 1$

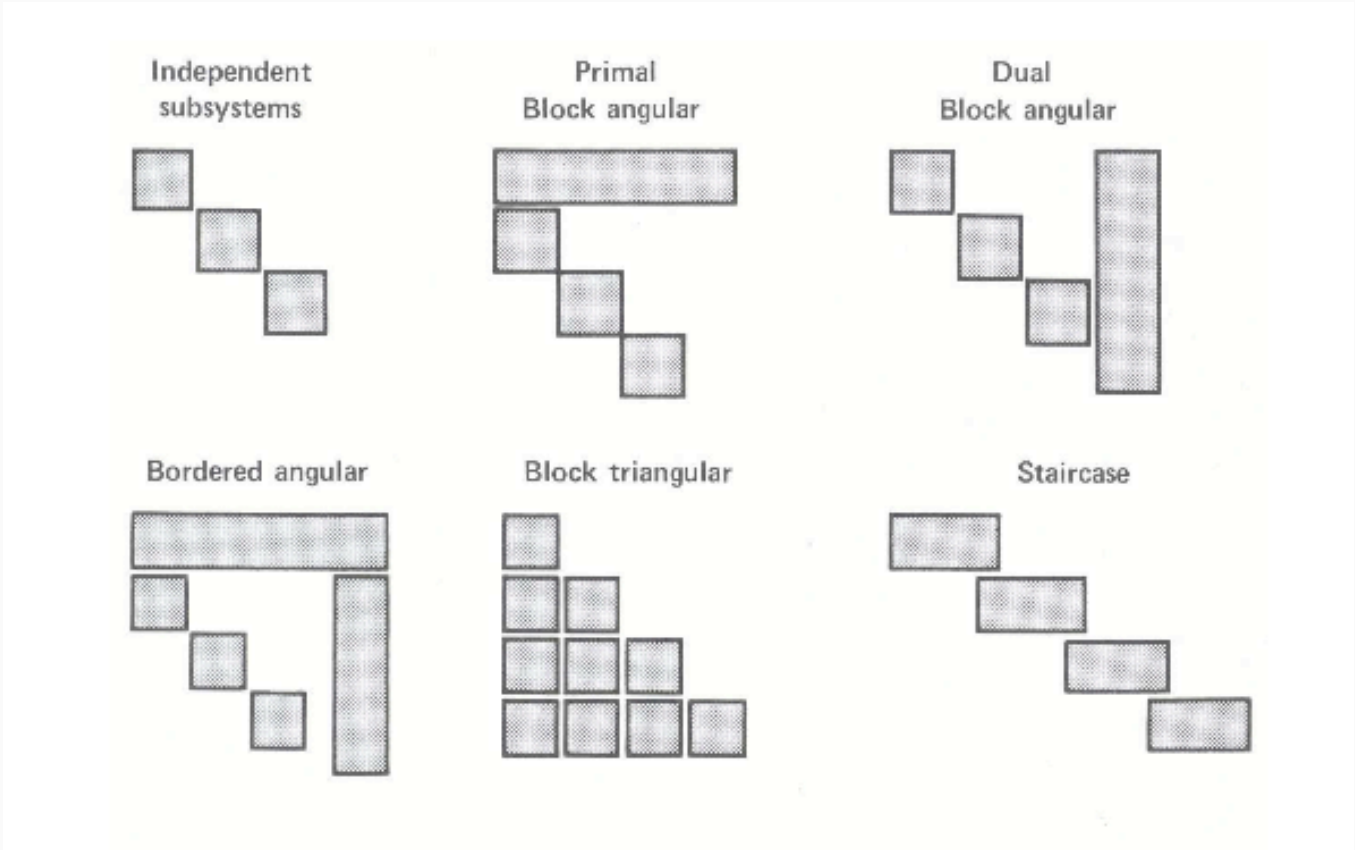
The bad news: The size of D grows exponentially!

The good news: D is very sparse

n	m	k	sparsity
2	3	3	0.222222
5	85	31	0.878558
10	11530	1023	0.996095
15	860175	32767	0.999878
20	49807380	1048575	0.999996

D Matrix Spatial Structure

One of the standard tricks to deal with large-scale LPs is to look for spatial structures in the constraint matrix (*D* in our case) and decompose the LP into a number of smaller LPs.



D Matrix Spatial Structure

Unfortunately this cannot be done for our D matrix.

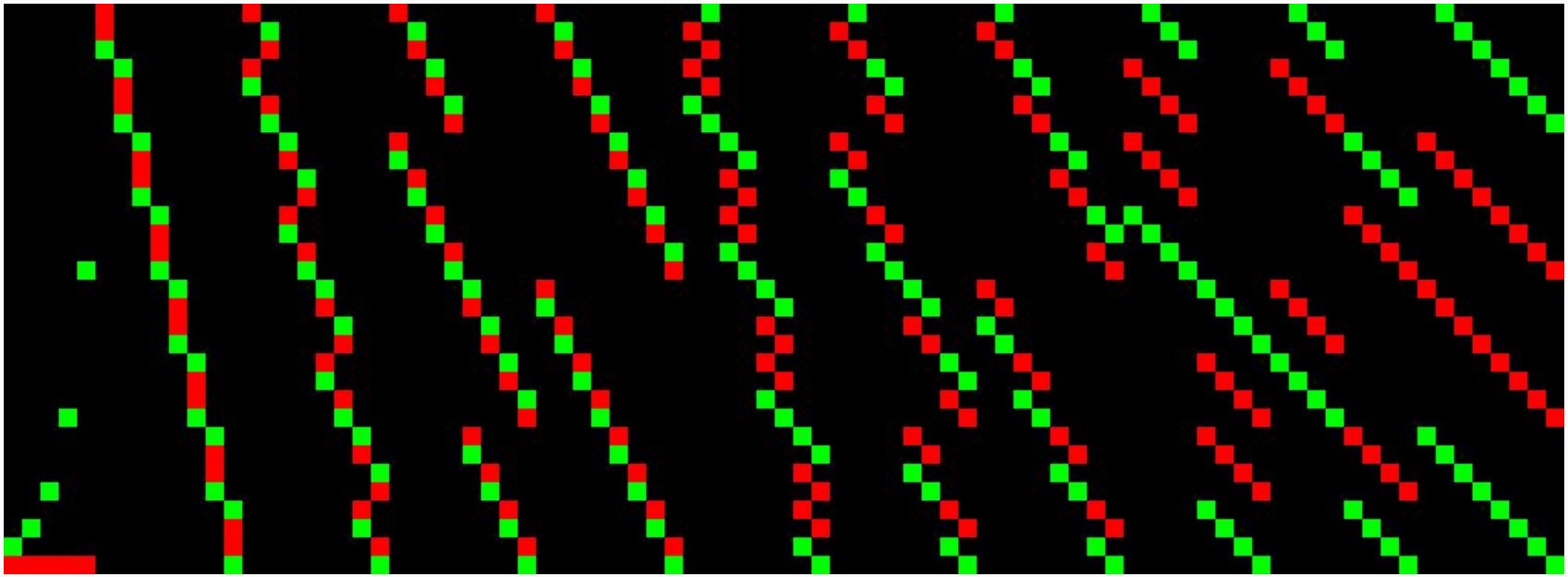


Figure 1: Sparsity map for D^T when $n = 5$

No spatial structure that we can utilize :(

Degeneracy

It's easy to see that in the dual problem

$$\begin{aligned} \max \quad & 0 \\ \text{s.t.} \quad & b - D^T \lambda = 0 \\ & \lambda \geq 0, \end{aligned}$$

there's an infinite number of optimal solution λ^*, which means the problem is highly degenerate.

Degeneracy

It's easy to see that in the dual problem

$$\begin{aligned} \max \quad & 0 \\ \text{s.t.} \quad & b - D^T \lambda = 0 \\ & \lambda \geq 0, \end{aligned}$$

there's an infinite number of optimal solution λ^*, which means the problem is highly degenerate.

The simplex method would struggle on degenerate problems.

Degeneracy

It's easy to see that in the dual problem

$$\begin{aligned} \max \quad & 0 \\ \text{s.t.} \quad & b - D^T \lambda = 0 \\ & \lambda \geq 0, \end{aligned}$$

there's an infinite number of optimal solution λ^*, which means the problem is highly degenerate.

The simplex method would struggle on degenerate problems.

Can we use numerical algorithms (e.g. interior-point method)? Yes, but there's a catch...

An ADMM-Based Scalable Solution

Generic ADMM

For the following optimization problem:

$$\begin{aligned} \min \quad & f(x) + g(y) \\ \text{s.t.} \quad & Ax + By = c, \end{aligned}$$

with ρ-augmented Lagrangian:

$$L_\rho = f(x) + g(y) + \lambda^T(Ax + By - c) + \frac{\rho}{2}||Ax + By - c||^2,$$

Generic ADMM

For the following optimization problem:

$$\begin{aligned} \min \quad & f(x) + g(y) \\ \text{s.t.} \quad & Ax + By = c, \end{aligned}$$

with ρ-augmented Lagrangian:

$$L_\rho = f(x) + g(y) + \lambda^T(Ax + By - c) + \frac{\rho}{2}||Ax + By - c||^2,$$

a generic ADMM algorithm is given as follows:

ALGORITHM 2: Generic ADMM Algorithm

repeat

1. x-update: $x^{k+1} = \textbf{arg min}\{L_\rho(x, y^k, \lambda^k)\}$
2. y-update: $y^{k+1} = \textbf{arg min}\{L_\rho(x^{k+1}, y, \lambda^k)\}$
3. λ-update: $\lambda^{k+1} = \lambda^k + \rho(Ax^{k+1} + By^{k+1} - c)$

until *Stopping criteria is met*;

Problem Reformulation

Consider the following reformulation:

$$\begin{aligned} \min \quad & b^T h \\ \text{s.t.} \quad & 0 \leq Ah \leq 1, \end{aligned}$$

where $A = \begin{bmatrix} D \\ I \end{bmatrix}$.

Problem Reformulation

Consider the following reformulation:

$$\begin{aligned} \min \quad & b^T h \\ \text{s.t.} \quad & 0 \leq Ah \leq 1, \end{aligned}$$

where $A = \begin{bmatrix} D \\ I \end{bmatrix}$.

1. If the inequality is provable, the extra constraint $Ah \leq 1$ would be redundant. In this case, the reformulated problem is equivalent to the original one in both primal and dual.

Problem Reformulation

Consider the following reformulation:

$$\begin{aligned} \min \quad & b^T h \\ \text{s.t.} \quad & 0 \leq Ah \leq 1, \end{aligned}$$

where $A = \begin{bmatrix} D \\ I \end{bmatrix}$.

1. If the inequality is provable, the extra constraint $Ah \leq 1$ would be redundant. In this case, the reformulated problem is equivalent to the original one in both primal and dual.
2. If the inequality is not provable, we are still upper-bounding the entries of h as the original problem does, and we can still construct disproof using the optimal dual values.

Problem Reformulation

Consider the following reformulation:

$$\begin{aligned} \min \quad & b^T h \\ \text{s.t.} \quad & 0 \leq Ah \leq 1, \end{aligned}$$

where $A = \begin{bmatrix} D \\ I \end{bmatrix}$.

1. If the inequality is provable, the extra constraint $Ah \leq 1$ would be redundant. In this case, the reformulated problem is equivalent to the original one in both primal and dual.
2. If the inequality is not provable, we are still upper-bounding the entries of h as the original problem does, and we can still construct disproof using the optimal dual values.

We can just solve this one problem!

Problem Reformulation

Now we reformulate the problem to ADMM form by adding slack variables u and v:

$$\begin{aligned} \min \quad & b^T h \\ \text{s.t.} \quad & Bh + z = c, \end{aligned}$$

where $B = \begin{bmatrix} A \\ A \end{bmatrix}$, $z = \begin{bmatrix} -u \\ v \end{bmatrix}$, $c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $u, v \geq 0$

ADMM Subproblems Solution

The ρ-augmented Lagrangian of the problem is:

$$L_\rho = b^T h + \lambda^T (Bh + z - c) + \frac{\rho}{2} ||Bh + z - c||^2$$

ADMM Subproblems Solution

The ρ-augmented Lagrangian of the problem is:

$$L_\rho = b^T h + \lambda^T (Bh + z - c) + \frac{\rho}{2} ||Bh + z - c||^2$$

h-update:

$$h^{k+1} = \textbf{arg min}\, L_\rho(h, z^k, \lambda^k)$$

ADMM Subproblems Solution

The ρ-augmented Lagrangian of the problem is:

$$L_\rho = b^T h + \lambda^T (Bh + z - c) + \frac{\rho}{2} ||Bh + z - c||^2$$

h-update:

$$h^{k+1} = \textbf{arg min}\, L_\rho(h, z^k, \lambda^k)$$

This is an unconstrained QP, and we have a closed-form solution:

$$h^{k+1} = -\frac{1}{\rho}(B^T B)^{-1}(b + B^T \lambda^k + \rho B^T z^k - \rho B^T c).$$

It's easy to prove that $(B^T B)^{-1}$ exists.

ADMM Subproblems Solution

z-update:

$$z^{k+1} = \textbf{arg min}\, L_\rho(h^{k+1}, z, \lambda^k)$$

Recall that $z = \begin{bmatrix} -u & v \end{bmatrix}^T$, so we can split this subproblem into u-update and v-update:

$$u^{k+1} = \textbf{arg min}\{L_\rho(h^{k+1}, u, \lambda^k) | u \geq 0\}$$
$$v^{k+1} = \textbf{arg min}\{L_\rho(h^{k+1}, v, \lambda^k) | v \geq 0\}$$

ADMM Subproblems Solution

z-update:

$$z^{k+1} = \textbf{arg min}\, L_\rho(h^{k+1}, z, \lambda^k)$$

Recall that $z = \begin{bmatrix} -u & v \end{bmatrix}^T$, so we can split this subproblem into u-update and v-update:

$$u^{k+1} = \textbf{arg min}\{L_\rho(h^{k+1}, u, \lambda^k) | u \geq 0\}$$
$$v^{k+1} = \textbf{arg min}\{L_\rho(h^{k+1}, v, \lambda^k) | v \geq 0\}$$

They are constrained QPs, but luckily their KKT systems can be solved directly, giving us closed-form solutions:

$$u^{k+1} = (Ah^{k+1} + \frac{1}{\rho}\lambda_u^k)_+$$
$$v^{k+1} = (1 - Ah^{k+1} - \frac{1}{\rho}\lambda_v^k)_+$$

ADMM Algorithm

ALGORITHM 3: ITIP ADMM Algorithm

repeat

1. h-update: $h^{k+1} = -\frac{1}{\rho}(B^T B)^{-1}(b + B^T \lambda^k + \rho B^T z^k - \rho B^T c)$
2. u-update: $u^{k+1} = (Ah^{k+1} + \frac{1}{\rho}\lambda_u^k)_+$
3. v-update: $v^{k+1} = (1 - Ah^{k+1} - \frac{1}{\rho}\lambda_v^k)_+$
4. λ-update: $\lambda^{k+1} = \lambda^k + \rho(Bh^{k+1} + z^{k+1} - c)$

until *Stopping criterion is met*;

Why This Algorithm?

The advantages:

1. We have closed-form solution for every subproblem, which means all we need to do is a series of Linear Algebra computation (distributed systems, GPUs, Cloud Computing, etc.)

Why This Algorithm?

The advantages:

1. We have closed-form solution for every subproblem, which means all we need to do is a series of Linear Algebra computation (distributed systems, GPUs, Cloud Computing, etc.)
2. Not simplex-based, so degeneracy is not a problem

Why This Algorithm?

The advantages:

1. We have closed-form solution for every subproblem, which means all we need to do is a series of Linear Algebra computation (distributed systems, GPUs, Cloud Computing, etc.)
2. Not simplex-based, so degeneracy is not a problem
3. Makes good use of sparsity of D

Why This Algorithm?

The advantages:

1. We have closed-form solution for every subproblem, which means all we need to do is a series of Linear Algebra computation (distributed systems, GPUs, Cloud Computing, etc.)
2. Not simplex-based, so degeneracy is not a problem
3. Makes good use of sparsity of D
4. Move the computation burden from solving optimization problem to inverting $B^T B$, which we can calculate beforehand and cache

Why This Algorithm?

The advantages:

1. We have closed-form solution for every subproblem, which means all we need to do is a series of Linear Algebra computation (distributed systems, GPUs, Cloud Computing, etc.)
2. Not simplex-based, so degeneracy is not a problem
3. Makes good use of sparsity of D
4. Move the computation burden from solving optimization problem to inverting $B^T B$, which we can calculate beforehand and cache

The disadvantage:

- We need to do “crossover” to obtain “elegent” proofs/disproofs

A Scalable Web Service

The ITIP Web Service (WiP)

`https://itip.algebragame.app`

Q&A

References i


S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al.
Distributed optimization and statistical learning via the alternating direction method of multipliers.
Foundations and Trends® in Machine learning, 3(1):1–122, 2011.


S.-W. Ho, C. W. Tan, and R. W. Yeung.
Proving and disproving information inequalities.
In *2014 IEEE International Symposium on Information Theory (ISIT)*, pages 2814–2818. Citeseer, 2014.


R. W. Yeung.
***Information theory and network coding*.**
Springer Science & Business Media, 2008.