Information-Theoretic Analysis of MaxCut Algorithms

An Bian, Alexey Gronskiy and Joachim M. Buhmann
Department of Computer Science, ETH Zurich
{ybian, alexeygr, jbuhmann}@inf.ethz.ch

July 12, 2016

Abstract

NP-hard combinatorial optimization algorithms are often characterized by their approximation ratios. In real world applications, the resilience of algorithms to input fluctuations and to modeling errors pose important robustness requirements. Motivated by this problem, we propose an information-theoretic algorithmic regularization and validation strategy based on *posterior agreement*, and further theoretically justify it by presenting the "coding by posterior" framework. The strategy regularizes algorithms and ranks them according to the informativeness of their output given noisy input. To illustrate this strategy, we develop methods to evaluate the posterior distribution of the Goemans-Williamson's MaxCut algorithm using semidefinite programming relaxation (MaxCut-SDP by Goemans and Williamson (1995)). Experimental comparison with representative greedy MaxCut algorithms shows that MaxCut-SDP with the best known approximation ratio generalizes worse than greedy MaxCut algorithms under high noise level.

1. Introduction

Algorithms are usually characterized by time and space complexity in the worst-case setting, and for specific instance distributions, in the average case. The robustness of an algorithm to input fluctuations is rarely investigated although such a property might often be indispensable in applications. Taking the MaxCut problem for example, in practice, instead of having the graph G as input to recover the maximal cut, one usually only have access to multiple noisy observations of the graph G. Assuming for simplicity, there are two noisy observations of the underlying "master" graph G: G' and G'', and we want to recover the maximal cut with respect to G. We call this setting the "two-instance" scenario, and the ability of an algorithm to recover the true solutions given only noisy observations is closed related to the robustness/informativeness of the algorithm. According to Buhmann (2010), the informativeness of algorithms sensitively depends on the input distributions, which controls the precision of the output given the input uncertainty.

For an algorithm $\mathscr{A}$ to survive in this two-instance scenario, e.g., the MaxCut-SDP (Goemans and Williamson, 1995), we propose a general information-theoretic regularization and validation strategy, which is based on a provable analogue of information content for algorithms. Classical algorithms usually search for a unique or a randomized solution in the hypothesis class. Input noise often renders such algorithmic solutions highly unstable. Therefore, we require an algorithm to return a posterior distribution of solutions given the noisy input. Such a posterior should concentrate on few solutions but the posterior must be stable for equally likely inputs. We interpret this tradeoff between precise localization in the hypothesis class and stability of posteriors as the *generalization* property of an algorithm. Under this strategy, an algorithm should stop early to recover the stable solutions (posterior distribution of solutions).

It is well-known that when training machine learning models, e.g., training neural network using the stochastic gradient descent algorithm, one should stop the algorithm early to recover generalizable models, which is called the "early-stopping" strategy (Caruana et al., 2001). With empirical success, few theory has been proposed for this well-utilized strategy. By simple analogue between the generalizable solutions and machine learning models, this work also gives an information-theoretic verification of this "early-stopping" strategy.

Though we use MaxCut algorithms as an illustrating example in this work, it is noteworthy that the strategy applies generally to any algorithms in the two-instance scenario. Previously, generalization of algorithms as a selection principle has been demonstrated for minimum spanning tree algorithms (Gronskiy and Buhmann, 2014) and data clustering (Buhmann, 2010).

1.1 MaxCut as an exemplary problem

Given an undirected graph $G = (V, E)$ with a vertex set $V = \{1, \cdots, n\}$ and an edge set E with non-negative weights $w_{ij}, \forall (i,j) \in E$, MaxCut determines a cut $c := (S, \bar{S})$ of the vertex set $V = S \cup \bar{S}, S \cap \bar{S} = \emptyset$ such that the cut value $\text{cut}(c) := \sum_{i \in S} \sum_{j \in \bar{S}} w_{ij}$ is maximal. MaxCut is a prototypical case of an unconstrained submodular maximization problem, and it is utilized in various applications, such as semisupervised learning (Wang et al., 2013) and social networks (Agrawal et al., 2003). Bian et al. (2015) has investigated the robustness of *greedy* MaxCut algorithms based on the "approximation set coding" framework (Buhmann, 2010; Gronskiy and Buhmann, 2014), the methodology used there can not handle the continuous MaxCut algorithms, e.g., the MaxCut-SDP (Goemans and Williamson, 1995). This work aims to analyze the generalization performance of non-greedy, continuous MaxCut algorithm based on strategy derived from a general framework called "coding by posterior".

1.2 Algorithm analysis by algorithmic information content

We investigate the generalization ability of an algorithm $\mathscr{A}$ under the *two-instance scenario*. Assume there are two noisy instances G', G'' drawn from the same underlying distribution $\mathcal{D}$. The algorithm then calculates a sequence of posteriors $\{\mathbb{P}_t^{\mathscr{A}}(c|G')\}$, $\{\mathbb{P}_t^{\mathscr{A}}(c|G'')\}$ as a function of time t, c is the solution in the hypothesis/solution space $\mathcal{C}$. The *posterior agreement* is defined to measure the overlap between the two posteriors at time t,

$$k_t^{\mathscr{A}}(G', G'') := \sum\nolimits_{c \in \mathcal{C}} \mathbb{P}_t^{\mathscr{A}}(c|G') \mathbb{P}_t^{\mathscr{A}}(c|G'')\,. \qquad \texttt{(posterior agreement)} \tag{1}$$

We define the *information content* of an algorithm $\mathscr{A}$ as the maximal *temporal information content* $I_t^{\mathscr{A}}$ at time t:

$$I^{\mathscr{A}}(G'; G'') := \max_t I_t^{\mathscr{A}}(G'; G'') \;= \max_t \mathbb{E}_{G', G''} \left[\log \left(|\mathcal{C}| k_t^{\mathscr{A}}(G', G'') \right) \right]. \tag{2}$$

It generalizes the algorithmic information content in Gronskiy and Buhmann (2014). $I_t^{\mathscr{A}}$ measures how much information is extracted by $\mathscr{A}$ at time t from the input distribution that is relevant to the output distribution, thus reflecting the generalization ability. It naturally suggests the following algorithmic regularization and validation strategy based on posterior agreement:

- *Regularize* an algorithm $\mathscr{A}$ by stopping it at the optimal time, which is defined as $t^* = \arg\max_t \mathbb{E}_{G', G''}[k_t^{\mathscr{A}}(G', G'')]$. It corresponds to the early-stopping strategy;
- *Validation*: Rank two algorithms $\mathscr{A}_1$ and $\mathscr{A}_2$ by comparing $I^{\mathscr{A}_1}$ and $I^{\mathscr{A}_2}$ under a specific input distribution $\mathcal{D}$.

The rest of this paper is organized as: Section 2 verifies definition of the algorithmic information content (Equation 2) by detailing and proving the "coding by posterior" framework; Section 3 interprets the MaxCut-SDP algorithm; Section 4 describes the provable methods to evaluate MaxCut-SDP posteriors; Section 5 contains experimental results; Section 6 discusses and concludes the paper.

2. The coding by posterior framework

We introduce the "coding by posterior" framework in this section, which is based on an analogue to the noisy communication channel in information theory.

We denote as G, G', $G'' \in \mathcal{G}$ different data instances generated from the underlying distribution $\mathcal{D}$, e.g., G may be a graph instance for the MaxCut problem. Often, a computational problem is associated with some cost function $R(c, G)$, which measures how well a hypothesis c in the hypothesis space $\mathcal{C}$ will solve the problem on input G. An algorithm $\mathscr{A}$ maps the input space to the hypothesis space $\mathscr{A} : \mathcal{G} \to \mathcal{C}$. We introduce parameters $\boldsymbol{\theta} \in \Theta$ to enumerate a set of algorithms. In optimization for example, $\boldsymbol{\theta}$ might denote the approximation precision. In general, we assume that algorithm $\mathscr{A}$ assigns non-negative weights $w_{\boldsymbol{\theta}}(c, G)$ to all hypotheses dependent on the input and the parameters, i.e.,

$$w : \mathcal{C} \times \mathcal{G} \times \Theta \to [0, +\infty), \quad (c, G, \boldsymbol{\theta}) \mapsto w_{\boldsymbol{\theta}}(c, G)\,. \tag{3}$$

Gibbs weights $w_\beta(c, G) = \exp\big(-\beta R(c, G)\big)$, for example, rank different hypotheses according to how well they solve the problem in terms of costs $R(c, G)$.

Such a weighting of hypotheses given data can be interpreted as a posterior distribution $\mathbb{P}_{\boldsymbol{\theta}}(c|G)$ induced by algorithm $\mathscr{A}$, and is defined as

$$\mathbb{P}_{\boldsymbol{\theta}}(c|G) := w_{\boldsymbol{\theta}}(c, G) / \sum\nolimits_{c' \in \mathcal{C}} w_{\boldsymbol{\theta}}(c', G), \quad \forall c \in \mathcal{C}\,. \tag{4}$$

For example, if we choose an indicator function as weights

$$w_{\boldsymbol{\theta}}(c, G) = \mathbb{1}\big\{R(c, G) \leq R(c^{\perp}, G) + \gamma(\boldsymbol{\theta})\big\}, \tag{5}$$

where $\gamma(\boldsymbol{\theta})$ denotes a precision value determined by a specific $\mathscr{A}$ and the empirical risk minimizer $c^{\perp}(G) = \arg\min_{c \in \mathcal{C}} R(c, G)$ centers an "approximation set" of size $\gamma(\boldsymbol{\theta})$ in $\mathcal{C}$. In this manner we can recover the "approximation set coding" in Buhmann (2010).

The posterior $\mathbb{P}_{\boldsymbol{\theta}}(c|G)$ effectively partitions the hypothesis class into statistically equivalent solutions with high weight values and discards hypotheses with vanishing weights. $\mathbb{P}_{\boldsymbol{\theta}}(c|G)$ plays the role of a codebook vector with the associated Voronoi cell. To generate alternative posteriors for a coding protocol we have to use the given data G and have to transform the mapping from $\mathcal{G}$ to $\mathcal{C}$. Such transformations should not change the measurements represented by G but the algorithmic mapping. Given data G and algorithm $\mathscr{A}$ with posterior $\mathbb{P}_{\boldsymbol{\theta}}(c|G)$, we define the *transformation set* $\mathbb{T}$ as a set of mappings $\tau : \mathcal{G} \to \mathcal{G}$ s.t. the following two conditions are satisfied,

1. $\mathscr{A}(\tau \circ G), \tau \in \mathbb{T}$ generates an "approximately uniform cover" of the hypothesis space $\mathcal{C}$, i.e., $\sum_{\tau \in \mathbb{T}} \mathbb{P}_{\boldsymbol{\theta}}(c|\tau \circ G) \in \left[\frac{|\mathbb{T}|}{|\mathcal{C}|}(1-\rho), \frac{|\mathbb{T}|}{|\mathcal{C}|}(1+\rho)\right]$, for $0 < \rho < 1$;

2. For every transformation $\tau \in \mathbb{T}$ there exists an associated transformation $\tau^{\mathcal{C}}$ such that $w_{\boldsymbol{\theta}}(c, \tau \circ G) = w_{\boldsymbol{\theta}}(\tau^{\mathcal{C}} \circ c, G)$.

Given a posterior and transformations, we can define a *virtual communication scenario*. It requires a sender $\mathfrak{S}$, a receiver $\mathfrak{R}$, and a problem generator $\mathfrak{PG}$ as a noisy channel between $\mathfrak{S}$ and $\mathfrak{R}$. Sender and receiver agree on algorithm $\mathscr{A}$ and its induced posterior.

2.1 Code book generation

The communication code is generated by the procedure:

1) Sender $\mathfrak{S}$ and receiver $\mathfrak{R}$ obtain the sample set G' from the problem generator $\mathfrak{PG}$.

2) Sender $\mathfrak{S}$ and receiver $\mathfrak{R}$ calculate the posterior $\mathbb{P}_{\boldsymbol{\theta}}(c|G')$.

3) A set of transformations $T = \{\tau_1, \cdots, \tau_M\} \subseteq \mathbb{T}$ is generated uniformly with associated posteriors $\mathbb{P}_{\boldsymbol{\theta}}(c|\tau_j \circ G'), 1 \leq j \leq M$.

4) $\mathfrak{S}$ and $\mathfrak{R}$ agree on a transformation set T and posteriors $\mathbb{P}_{\boldsymbol{\theta}}(c|\tau_j \circ G'), 1 \leq j \leq M$.

The posteriors $\mathbb{P}_{\boldsymbol{\theta}}(c|\tau_j \circ G'), \tau_j \in T$ play the role of codebook vectors in Shannon's theory of communication.

2.2 Communication protocol

1) The sender $\mathfrak{S}$ selects a transformation $\tau_s \in T$ as message and sends it to the problem generator $\mathfrak{PG}$.

2) $\mathfrak{PG}$ generates the test sample set G'' and applies the transformation τ_s to G'', yielding $\tilde{G} := \tau_s \circ G''$.

3) $\mathfrak{PG}$ sends $\tilde{G}$ to $\mathfrak{R}$ without revealing τ_s.

4) $\mathfrak{R}$ calculates the posterior $\mathbb{P}_{\boldsymbol{\theta}}(c|\tilde{G})$.

5) $\mathfrak{R}$ estimates the message τ_s by using the decoding rule:

$$\hat{\tau} = \arg\max_{\tau \in T} \mathbb{E}_{c \sim \mathbb{P}_{\boldsymbol{\theta}}(c|\tau \circ G')} \mathbb{P}_{\boldsymbol{\theta}}(c|\tau_s \circ G'') = \arg\max_{\tau \in T} \sum\nolimits_{c \in \mathcal{C}} \mathbb{P}_{\boldsymbol{\theta}}(c|\tau \circ G') \mathbb{P}_{\boldsymbol{\theta}}(c|\tilde{G}) \quad (6)$$

2.3 Error analysis of the virtual communication protocol

The probability of a communication error amounts to

$$\begin{aligned}\mathbb{P}(\hat{\tau} \neq \tau_s|\tau_s) &= \mathbb{P}\Big(\max_{\tau_j \in T\setminus\tau_s} \mathbb{E}_{\mathbb{P}(c|\tau_j\circ G')}[\mathbb{P}(c|\tilde{G})] \geq \mathbb{E}_{\mathbb{P}(c|\tau_s\circ G')}[\mathbb{P}(c|\tilde{G})]\Big) \\ &\overset{(a)}{\leq} \sum_{\tau_j \in T\setminus\tau_s} \mathbb{P}\left(\mathbb{E}_{\mathbb{P}(c|\tau_j\circ G')}[\mathbb{P}(c|\tilde{G})] \geq \mathbb{E}_{\mathbb{P}(c|G')}[\mathbb{P}(c|G'')]\right) \qquad (7) \\ &\overset{(b)}{\leq} \sum_{\tau_j \in T\setminus\tau_s} \mathbb{E}_{G',G''} \frac{\mathbb{E}_{\tau_j}\mathbb{E}_{\mathbb{P}(c|\tau_j\circ G')}[\mathbb{P}(c|\tilde{G})]}{\mathbb{E}_{\mathbb{P}(c|G')}[\mathbb{P}(c|G'')]}\,,\end{aligned}$$

by applying the union bound (a) and Markov's inequality (b).

Abbreviating $Z_{\mathbb{T}} := \mathbb{E}_{\tau_j}\mathbb{E}_{\mathbb{P}(c|\tau_j\circ G')}[\mathbb{P}(c|\tilde{G})]$, we derive

$$\begin{aligned}Z_{\mathbb{T}} &= \mathbb{E}_{\tau_j}\mathbb{E}_{\mathbb{P}(c|\tau_j\circ G')}\mathbb{P}(c|\tilde{G}) = \mathbb{E}_{\tau_j}\sum_{c\in\mathcal{C}} \mathbb{P}(c|\tau_j\circ G')\mathbb{P}(c|\tilde{G}) \\ &= \sum_{c\in\mathcal{C}} \mathbb{P}(c|\tilde{G})\mathbb{E}_{\tau_j}\mathbb{P}(c|\tau_j\circ G') = \sum_{\tau_j\in\mathbb{T}} \mathbb{P}(\tau_j)\mathbb{P}(c|\tau_j\circ G) \leq (1+\rho)|\mathcal{C}|^{-1} \qquad (8)\end{aligned}$$

where Equation 8 arises from the approximately uniform coverage of $\mathcal{C}$ by the posteriors, i.e., $\sum_{\tau\in\mathbb{T}} \mathbb{P}(c|\tau\circ G) \in \big[\frac{|\mathbb{T}|}{|\mathcal{C}|}(1-\rho), \frac{|\mathbb{T}|}{|\mathcal{C}|}(1+\rho)\big]$ and $\mathbb{P}(\tau) = 1/|\mathbb{T}|$. Substituting Equation 8 into Equation 7 we derive the error bound

$$\begin{aligned}\mathbb{P}(\hat{\tau} \neq \tau_s|\tau_s) &\leq \sum_{\tau_j\in T\setminus\tau_s} \mathbb{E}_{G',G''}\left[\big(\tfrac{|\mathcal{C}|}{1+\rho}\mathbb{E}_{\mathbb{P}(c|G')}[\mathbb{P}(c|G'')]\big)^{-1}\right] \\ &= (M-1)\mathbb{E}_{G',G''}\left[\big(\tfrac{|\mathcal{C}|}{1+\rho}k(G',G'')\big)^{-1}\right] \qquad (9) \\ &\leq M\mathbb{E}_{G',G''}\left[\exp\big(-\log(\tfrac{|\mathcal{C}|}{1+\rho}k(G',G''))\big)\right]. \qquad (10)\end{aligned}$$

Where Equation 9 comes from the definition of posterior agreement in Equation 1. We analyze $\hat{I} := \log\big(|\mathcal{C}|k(G',G'')\big)$ in Equation 10 at its expected value

$$I := \mathbb{E}_{G',G''}\left[\log\big(|\mathcal{C}|k(G',G'')\big)\right]. \qquad (11)$$

To control the fluctuations $\Delta_{G',G''} := \hat{I} - I$, we assume that for all $\epsilon > 0, \delta > 0$, there exists $n_0 \in \mathbb{N}$ s.t. for all $n > n_0$

$$\mathbb{P}\left(|\Delta_{G',G''}| \geq \epsilon I\right) < \delta\,. \qquad (12)$$

This assumption of asymptotically vanishing fluctuations yields the following upper bound

$$\mathbb{E}_{G',G''}\big[\exp(-\hat{I})\big] \leq \exp\big(-I(1-\epsilon)\big). \qquad (13)$$

Algorithm 1: MaxCut-SDP (Goemans and Williamson, 1995)

Input: undirected graph $G = (V, E)$ with non-negative weights w
Output: cut $c = (S, \bar{S})$

1 solve (R), obtaining an optimal set of vectors $\boldsymbol{v}_i \in S_{n-1}$;
2 let $\boldsymbol{r}$ be a vector *uniformly* distributed on S_{n-1};
3 **return** $S := \{i \mid \boldsymbol{v}_i \cdot \boldsymbol{r} \geq 0, \forall i \in V\}$ *and* $\bar{S}$

Since ϵ can be chosen arbitrarily small in the asymptotic limit, the error probability is bounded with high probability by

$$\mathbb{P}(\hat{\tau} \neq \tau_s | \tau_s) \leq \exp\big(-I + \log(M(1+\rho))\big)\,. \tag{14}$$

For I exceeding the effective total rate $\log(M(1+\rho))$, the error vanishes asymptotically since $I = O(\log|\mathcal{C}|)$. This bound suggests that we should maximize I (Equation 11) when searching for informative algorithms, thus verifying our definition of algorithmic information content in Equation 2.

2.4 Connection to classical mutual information

We analyse the classical mutual information $\mathcal{I}(G'; G'')$ by expanding the joint distribution $\mathbb{P}(G', G'')$ with cut variables $c \in \mathcal{C}$ and transformations $\tau \in T$:

$$\mathcal{I}(G'; G'') = \mathbb{E}_{G',G''} \log \frac{\mathbb{P}(G', G'')}{\mathbb{P}(G')\mathbb{P}(G'')} = \mathbb{E}_{G',G''} \log \frac{\sum_c \sum_\tau \mathbb{P}(G', G''|c, \tau)\mathbb{P}(c, \tau)}{\mathbb{P}(G')\mathbb{P}(G'')} \tag{15}$$

The conditional distribution $\mathbb{P}(G', G''|c, \tau)$ of G', G'' factorizes due to conditioning on c and τ,

$$\mathbb{P}(G', G''|c, \tau) = \mathbb{P}(G'|c, \tau)\mathbb{P}(G''|c, \tau) \overset{(a)}{=} \frac{\mathbb{P}(c|G', \tau)}{\mathbb{P}(c|\tau)}\mathbb{P}(G'|\tau)\frac{\mathbb{P}(c|G'', \tau)}{\mathbb{P}(c|\tau)}\mathbb{P}(G''|\tau), \tag{16}$$

since G', G'' are drawn i.i.d. from the same distribution $\mathbb{P}(\tau_s \circ G)$. The transformation τ plays the role of a latent variable. Step (a) applies the Bayes rule twice. Substitute Equation 16 into 15 we get (detailed derivation in Appendix A)

$$\mathcal{I}(G'; G'') = \mathbb{E}_{G',G''} \log \sum\nolimits_c \Big[\mathbb{P}(c|G')\mathbb{P}(c|G'')/\mathbb{P}(c)\Big] \leq \mathbb{E}_{G',G''} \log|\mathcal{C}| \sum\nolimits_c \mathbb{P}(c|G')\mathbb{P}(c|G'') = I(G'; G'').$$

With the uniform distribution $\mathbb{P}(c) = |\mathcal{C}|^{-1}$, $\mathcal{I}(G'; G'')$ is maximized and we reach the algorithmic information content in Equation 2.

3. MaxCut algorithm using SDP relaxation

In this section we give a geometric interpretation of Goemans-Williamson's MaxCut algorithm using semidefinite programming relaxation (Goemans and Williamson

(1995), abbreviated as MaxCut-SDP), which will facilitate deriving methods to calculate the posterior of cuts. Algorithm 1 summarizes the MaxCut-SDP algorithm: It rounds the solution to a non-linear programming relaxation, which can be interpreted as SDP, then it solves the SDP using standard algorithms, such as interior-point methods (Helmberg et al., 1996), bundle method or block coordinate descent (Waldspurger et al., 2015). Concretely, MaxCut is formulated as the NP-complete integer program:

$$\begin{array}{ll} & \max \frac{1}{2} \sum_{i<j} w_{ij}(1 - v_i v_j) \\ (Q) & \text{s.t. } v_i \in \{-1, 1\} \qquad \forall i \in V \end{array} \tag{17}$$

then (Q) is relaxed to define the following non-linear problem,

$$\begin{array}{ll} & \max \frac{1}{2} \sum_{i<j} w_{ij}(1 - \boldsymbol{v}_i \cdot \boldsymbol{v}_j) \\ (R) & \text{s.t. } \boldsymbol{v}_i \in S_{n-1} \qquad \forall i \in V \end{array} \tag{18}$$

where S_{n-1} is the $(n-1)$-dimensional unit sphere, i.e., $S_{n-1} = \{\boldsymbol{v} \in \mathbb{R}^n \mid \|\boldsymbol{v}\|_2 = 1\}$. Arrange the n vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n$ to be the n columns of a $n \times n$ matrix D, that is, $D = (\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n)$. Let $X := D^\top D$, then the ij-th entry of X is $x_{ij} = \boldsymbol{v}_i \cdot \boldsymbol{v}_j$. One can observe that (R) equals to the following SDP problem with only equality constraints:

$$\begin{array}{ll} & \max \frac{1}{2} \sum_{i<j} w_{ij}(1 - x_{ij}) \\ \text{(SDP)} & \text{such that } x_{ii} = 1, \forall i \in V,\ X \text{ is symmetric positive semidefinite.} \end{array} \tag{19}$$

We use one classical interior-point method (Helmberg et al., 1996) to solve the SDP problem in (19).

A geometric view in Figure 1 explains the essense of Algorithm 1: It maps vertices to vectors on the unit sphere. A feasible SDP solution corresponds to a point configuration on the unit sphere, while a feasible solution to MaxCut assigns a sign variable $\{\pm 1\}$ with every graph vertex. An optimal solution of SDP tends to send adjacent vertices with heavy edges to antipodal points, thereby maximizing $(1 - \boldsymbol{v}_i \cdot \boldsymbol{v}_j)/2$. A rounding technique is required that separates most far away pairs, and hence keeps close pairs together. *Random hyperplane rounding* works gracefully: A random hyperplane through the origin partitions the sphere into two halves, which correspond to cut parts (see Figure 1). The ratio between the expected cut value over the maximum cut value is never worse than $\alpha \approxeq .87856$, which is the expected approximation guarantee.

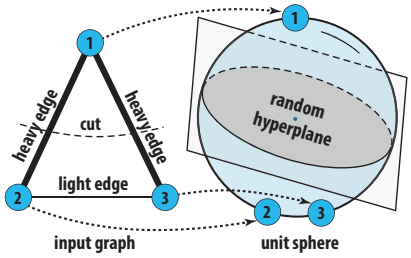

Figure 1: A geometric view of Algorithm 1

4. Calculate posterior probability of cuts

Given the geometric interpretation of the MaxCut-SDP algorithm in Section 3, we can derive the scheme to calculate posterior probability of cuts here, which will be used to evaluate the posterior agreement in Equation 1.

In step t, the relaxed SDP problem (18) outputs the n intermediate vectors $O_t = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n\}$. Each cut $c := (S, \bar{S})$, where $S = \{1, \ldots, \ell\}$, $\bar{S} = \{\ell + 1, \ldots, n\}$, induces a set,

$$B(c) = \{\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n\} := \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_\ell, -\boldsymbol{v}_{\ell+1}, \cdots, -\boldsymbol{v}_n\}.$$

$B(c)$ is used to define a polygonal intersection cone:

Definition 1 (Polygonal *intersection* cone)**.** *The cone C determined by cut $c = (S, \bar{S})$ is the intersection of n half-spaces:*

$$C = C(c) := \{\boldsymbol{x} \in \mathbb{R}^n, \|\boldsymbol{x}\|_2 \leq 1 \mid \boldsymbol{x} \cdot \boldsymbol{b}_i \geq 0, \forall i \in V\}. \tag{20}$$

It determines the posterior of the corresponding cut by Lemma 1 (all proof of lemmas are in Appendix A) in the following,

Lemma 1. *The posterior $\mathbb{P}(c|G)$ of a cut c is*

$$\mathbb{P}(c|G) = \frac{2 * \text{unit spherical area of } C(c)}{\text{area of unit sphere}} = \frac{2 * \text{volume of } C(c)}{\text{volume of unit ball}} = \frac{2 * \text{solid angle of } C(c)}{\text{solid angle of unit sphere}}. \tag{21}$$

Ensured by Lemma 1, the cut probabilities are measured either by spherical area, by volume or by solid angle. Without loss of generality, we calculate the solid angle to derive $\mathbb{P}(c|G)$. Since it is convenient to express the method of calculating solid angle in terms of the spanning cone definition, let us transform the intersection cone C into the spanning cone,

Definition 2 (Polygonal *spanning* cone)**.** *According to Tiel (1984), a polygonal spanning cone is spanned by a set of n linearly independent unit vectors $A = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$ in $\mathbb{R}^n$:*

$$C' := \{\boldsymbol{x} \in \mathbb{R}^n, \|\boldsymbol{x}\|_2 \leq 1 \mid \boldsymbol{x} = \sum\nolimits_{i=1}^{n} \lambda_i \boldsymbol{a}_i, \lambda_i \geq 0, 1 \leq i \leq n\}$$

Given an intersection cone C, one can get an equivalent spanning cone C' by taking $A^\top = B^{-1}$, which is ensured by,

Lemma 2. *Given one intersection cone C (Definition 1) and one spanning cone C' (Definition 2), if $\exists k_1, \cdots, k_n > 0$, s.t., $A^\top = diag(k_1, \cdots, k_n)B^{-1}$, then $C' = C$.*

Algorithm 2: Calculate posterior of each cut

Input: independent vectors $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n\}$ on S_{n-1}
Output: posterior of each cut $\mathbb{P}(c|G), \forall c \in \mathcal{C}$

for each cut $c \in \mathcal{C}$ **do**
  get the cut induced set B
  $A^\top \leftarrow B^{-1}$; `//ensured by Lemma 2`
  compute $\mathbb{P}(c|G)$ by Equations 23 and 22;
return $\mathbb{P}(c|G), \forall c \in \mathcal{C}$

Now we have the spanning cone C' associated with the cut c, we borrow the results of n-dimensional solid angle calculating (Hajja and Walker, 2002; Ribando, 2006): The solid angle of a spanning cone C' from Definition 2 is given by:

$$E = |\det(A)| \int_S \|A\boldsymbol{s}\|_2^{-n} dS, \tag{22}$$

where the integral is calculated over a unit sphere $\|\boldsymbol{s}\|_2 = 1$ in the positive orthant given by $s_i \geq 0$. Combined with the fact that the solid angle subtended by S_{n-1} is $\Omega_n = \frac{2\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2})}$ ($\Gamma(\cdot)$ is the Gamma function), according to Equation 21,

$$\mathbb{P}(c|G) = 2E/\Omega_n = E \cdot \Gamma(n/2)/\pi^{\frac{n}{2}}. \tag{23}$$

The complete procedure[1] to calculate the posterior probability of each cut is summarized in Algorithm 2.

The way to exactly evaluate the surface integral (Equation 22) is in Appendix B. It involves a $(n-1)$-variate integral, which is computationally intractable, we only use it in the low dimensional case as ground truth.

Sampling to approximate posterior of cut. For the high dimensional case, ensured by Lemma 1, we propose one simple and efficient sampling method in Algorithm 3 to approximate the posterior: In each iteration it uniformly samples one hyperplane with normal vector $\boldsymbol{r}$ and records the cut c separated by that hyperplane, then it estimates $\mathbb{P}(c|G)$ by the statistics of each cut's frequency of occurrence. Theoretical analysis of approximation guarantee of Algorithm 3 and space-efficient implementation of it is in Appendix C and D, respectively.

5. Experiments

We compare the MaxCut-SDP algorithm (abbreviated as "`SDP`" in the following) with two representative greedy MaxCut algorithms: The double greedy `D2Greedy`

1. If the n intermediate vectors $O_t = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n\}$ have mutual dependencies, one can add small perturbations to them in order to make them independent, and the perturbation would still be insignificant w.r.t. vector positions.

Algorithm 3: Approximate cut's posterior by sampling

Input: $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n\}$ on S_{n-1}, #samplings
Output: approximate posterior of each cut

initialize $\text{count}(c) \leftarrow 0, \forall c \in \mathcal{C}$;
for each $\boldsymbol{r}$ *uniformly* sampled from S_{n-1} **do**
  $\tilde{c} \leftarrow (S, \bar{S})$, where $S = \{i \mid \boldsymbol{v}_i \cdot \boldsymbol{r} \geq 0, \forall i \in V\}$;
  $\text{count}(\tilde{c}) \leftarrow \text{count}(\tilde{c}) + 1$;
return $\mathbb{P}(c|G) \cong \text{count}(c)/\#\text{samplings}, \forall c \in \mathcal{C}$

(Deterministic Double `Greedy` algorithm in Buchbinder et al. (2012)), and the backward greedy `EC` (Edge Contraction algorithm in Kahruman et al. (2007)). The way to evaluate their posteriors (approximation sets) can be found in Bian et al. (2015). Let $W_{\mathscr{A}}(G)$ be the cut value generated by an algorithm $\mathscr{A}$ on graph G, $W_*(G)$ be the optimal cut value of G. The approximation ratio of an algorithm $\mathscr{A}$ is the worst-case bound $\min_G \frac{W_{\mathscr{A}}(G)}{W_*(G)}$, which ranks the three algorithms as `SDP` $\succ$ `D2Greedy` $\succ$ `EC`. Since finding $W_*(G)$ for NP-complete problem is non-trivial, we use $\frac{W_{\mathscr{A}}(G)}{W(G)}$ ($W(G)$: total weight of G) as a natural lower bound of $\frac{W_{\mathscr{A}}(G)}{W_*(G)}$.

Experimental setting. We experiment with the Gaussian edge weights model (Gronskiy and Buhmann, 2014): The graph instances are generated in a two-step fashion: Firstly, a random "master" graph G is generated with Gaussian distributed edge weights $w_{ij} \sim N(\mu, \sigma_m^2)$, $\mu = 300, \sigma_m = 50$, negative edges are set to be μ. Secondly, noisy graphs G', G'' are obtained by adding Gaussian distributed noise $n_{ij} \sim N(0, \sigma^2)$, negative edges are set to be zero. We perform 1000 repeated noisy samplings to estimate the expectation over (G', G'') in Equation 2.

Results and analysis. Figure 2 shows the temporal information content ($I_t^{\mathscr{A}}$ in Equation 2) for two σ values: 10 and 58. For all the algorithms, $I_t^{\mathscr{A}}$ increases at the beginning. After reaching some optimal step t^*, where the highest $I_t^{\mathscr{A}}$ ($I^{\mathscr{A}}$ in Equation 2) is achieved, it decreases and finally vanishes. This observation confirms the principle of regularization by early stopping at time t^* when maximum $I_t^{\mathscr{A}}$ is reached.

Figure 3 shows (a) the information content, and (b) the fully overlapping curves $W_{\mathscr{A}}(G')/W(G')$, $W_{\mathscr{A}}(G'')/W(G'')$, respectively. σ controls the noise level, larger σ means larger noise. In the noiseless case, $G' = G''$, so $\mathbb{P}(c|G') = \mathbb{P}(c|G'')$, and $I_t^{\mathscr{A}} = \mathbb{E}_{G',G''}[\log(|\mathcal{C}| \sum_{c \in \mathcal{C}} \mathbb{P}^2(c|G'))]$. All algorithms start with uniform distribution of solutions when $t = 0$; as the algorithm proceeds, the distribution of solutions concentrates more and more on a small support, $\sum_{c \in \mathcal{C}} \mathbb{P}^2(c|G')$ increases and reaches a maximum in the final step, so all algorithms reach the maximum $I_t^{\mathscr{A}}$ in the final step. For greedy algorithms (`D2Greedy` and `EC`), there is only one final solution

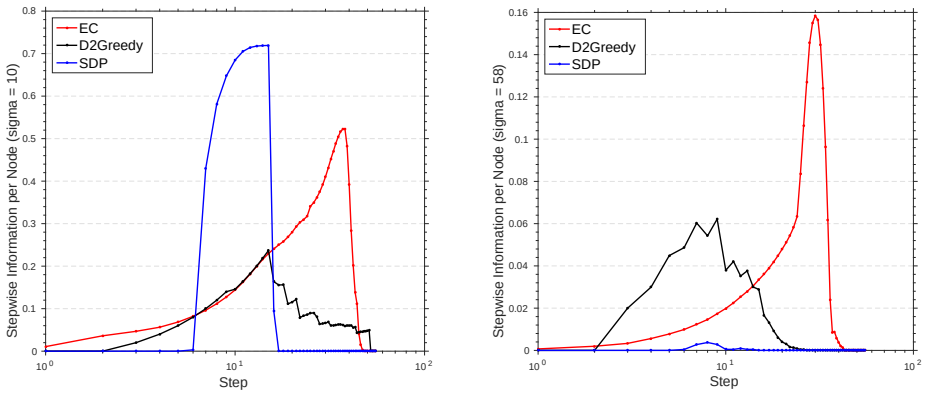

Figure 2: $I_t^{\mathscr{A}}$ *per vertex* w.r.t. t (a) $\sigma = 10$ (b) $\sigma = 58$ ($n = 50$)

with probability 1 in the last step, so $\sum_{c\in\mathcal{C}} \mathbb{P}^2(c|G') = 1$ and the maximum $I_t^{\mathscr{A}}$ is $\log(|\mathcal{C}|) = \log(2^{n-1} - 1)$, as shown by Figure 3(a). For `SDP`, however, when $\sigma = 0$, `SDP` can only approximately solve the input graphs, in the last step there are several solutions with non-zero probability, which renders its information content less than $\log(|\mathcal{C}|)$.

It is worth noting that for greedy algorithms (`D2Greedy` and `EC`), the higher the approximation ratio is for noisy graphs, the lower is the information content achieved by the algorithm. This behavior is quite intuitive since high approximation ratio means better adaptation to empirical fluctuations and, therefore, overfitting to noisy graphs. Consequently, there will be less agreement between the solutions of the two noisy graphs and the information content of the algorithm drops. A similar conclusion has also been drawn in Bousquet and Bottou (2008).

However, for non-greedy algorithm `SDP`, there are two factors affecting its information content: The approximation ratio and its probabilistic weighting strategy to down-weight solutions without discarding them. `SDP` keeps all the possible solutions, instead of removing the bad solutions as greedy algorithms do, it assigns less probabilistic weights to them, so it can capture some uncertainty in the input.

The information content of `SDP` shows the influence of both factors: For low noise, the probabilistic weighting strategy dominates, `SDP` outperforms greedy algorithms in information content; while in high noise level, the influence of approximation ratio dominates, and `SDP` is inferior to greedy algorithms.

6. Discussion and conclusion

$I^{\mathscr{A}}$ in Equation (2) measures the information content of an algorithm given a noisy source of instances. We theoretically justify this criterion, and apply it to study the robustness of MaxCut algorithms with different approximation ratios. Of particular interest is the SDP based algorithm by Goemans and Williamson (1995), since it pursues a non-greedy strategy for MaxCut.

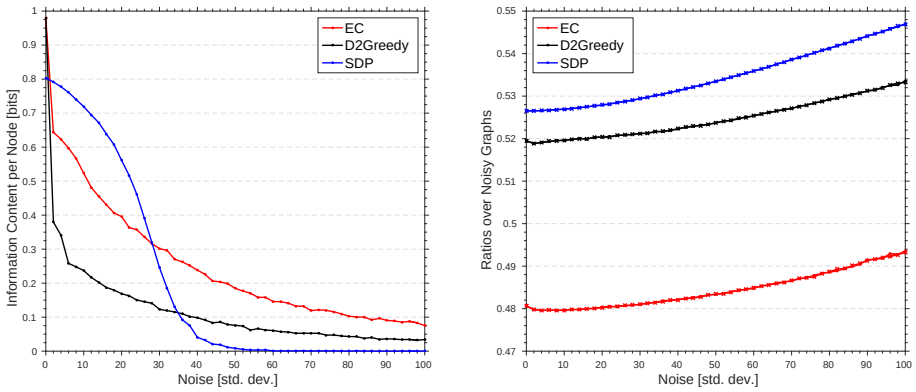

Figure 3: (a) $I^{\mathscr{A}}$ *per vertex* w.r.t. σ (b) $\frac{W_{\mathscr{A}}(G')}{W(G')}$, $\frac{W_{\mathscr{A}}(G'')}{W(G'')}$

Comparison of `SDP` with two representative greedy MaxCut algorithms (`D2Greedy` and `EC`) demonstrates that the ability of this approximation algorithm to achieve a high approximation ratio might decrease its generalization ability. The property of an algorithm to efficiently find a good empirical minimum might increase its fragility due to noise adaptation. This observation could be generalized or even proved for general approximation algorithms provided that the algorithms operate in a similar settings or use similar optimization strategies.

The posterior agreement based criterion also enables a meta-algorithm to search for more informative algorithms. Algorithms are usually tuned by parameter adaptation or by modifying the algorithmic strategy in the spirit of genetic programming. Thereby, the meta-algorithm will search through the space of algorithms guided by maximal gradient ascent on posterior agreement. With a validation criterion as posterior agreement, we enable algorithm engineering to explore multi-objective optimization of algorithms with respect to time, space and robustness.

Acknowledgment

This research was partially supported by the Max Planck ETH Center for Learning Systems.

References

Rakesh Agrawal, Sridhar Rajagopalan, Ramakrishnan Srikant, and Yirong Xu. Mining newsgroups using networks arising from social behavior. In *WWW*, pages 529–535, 2003.

Yatao Bian, Alexey Gronskiy, and Joachim M. Buhmann. Greedy maxcut algorithms and their information content. In *IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015.

Olivier Bousquet and Léon Bottou. The tradeoffs of large scale learning. In *NIPS*, pages 161–168, 2008.

Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *FOCS, 2012*, pages 649–658. IEEE, 2012.

Joachim M. Buhmann. Information theoretic model validation for clustering. In *ISIT*, pages 1398–1402, 2010.

Rich Caruana, Steve Lawrence, and Lee Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *NIPS*, volume 13, page 402, 2001.

Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

Alexey Gronskiy and Joachim M. Buhmann. How informative are minimum spanning tree algorithms. In *ISIT*, 2014.

Torben Hagerup and Christine Rüb. A guided tour of chernoff bounds. *Information processing letters*, 33(6):305–308, 1990.

Mowaffaq Hajja and Peter Walker. The measure of solid angles in n-dimensional euclidean space. *International Journal of Mathematical Education in Science and Technology*, 33(5):725–729, 2002.

Christoph Helmberg, Franz Rendl, Robert J Vanderbei, and Henry Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6(2):342–361, 1996.

Sera Kahruman, Elif Kolotoglu, Sergiy Butenko, and Illya V Hicks. On greedy construction heuristics for the maxcut problem. *International Journal of Computational Science Engineering*, 3(3):211–218, 2007.

Jason M Ribando. Measuring solid angles beyond dimension three. *Discrete & Computational Geometry*, 36(3):479–487, 2006.

Jan van Tiel. *Convex analysis*. John Wiley, 1984.

Irène Waldspurger, Alexandre d'Aspremont, and Stéphane Mallat. Phase recovery, maxcut and complex semidefinite programming. *Mathematical Programming*, 149 (1-2):47–81, 2015.

Jun Wang, Tony Jebara, and Shih-Fu Chang. Semi-supervised learning using greedy max-cut. *JMLR*, 14(1):771–800, March 2013. ISSN 1532-4435.

Appendix

A. Proofs

A.1 Detailed proof in Section 2.4

The classical mutual information determines a lower bound on information content defined in Equation 2. The classical mutual information $\mathcal{I}(G';G'')$ is defined as

$$\mathcal{I}(G';G'') = \mathbb{E}_{G',G''} \log \frac{\mathbb{P}(G',G'')}{\mathbb{P}(G')\mathbb{P}(G'')} \tag{24}$$

From the definition of virtual communication scenario, the data instances G', G'' can be treated to be drawn from a mixture distribution, as illustrated in Figure 4.

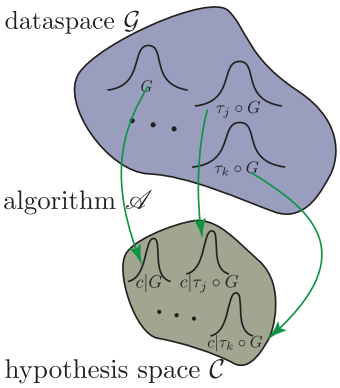

Figure 4: Illustration of the mixture distribution

We consider the special transformations that map from input space to output spaces, then the joint probability can be factorized in the following way,

$$\begin{aligned}
\mathbb{P}(G',G'') &= \sum_{c\in\mathcal{C}} \sum_{T\in\mathbb{T}^M} \mathbb{P}(G',G''|c,T)\mathbb{P}(c,T) \\
&= \sum_{j=1}^{M} p_j \sum_{\tau_j\in\mathbb{T}} \sum_{c\in\mathcal{C}} \mathbb{P}(G',G''|\tau_j,c)\mathbb{P}(\tau_j,c) \\
&\overset{(a)}{=} \sum_{j=1}^{M} p_j \sum_{\tau_j\in\mathbb{T}} \sum_{c\in\mathcal{C}} \mathbb{P}(G'|\tau_j,c)\mathbb{P}(G''|\tau_j,c)\mathbb{P}(c|\tau_j)\mathbb{P}(\tau_j) \\
&\overset{(b)}{=} \sum_{j=1}^{M} p_j \sum_{\tau_j\in\mathbb{T}} \mathbb{P}(\tau_j) \sum_{c\in\mathcal{C}} \frac{\mathbb{P}(c|G',\tau_j)}{\mathbb{P}(c|\tau_j)}\mathbb{P}(G'|\tau_j)\frac{\mathbb{P}(c|G'',\tau_j)}{\mathbb{P}(c|\tau_j)}\mathbb{P}(G''|\tau_j)\mathbb{P}(c|\tau_j) \\
&\overset{(c)}{=} \sum_{j=1}^{M} p_j \sum_{\tau_j\in\mathbb{T}} \mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_j)\mathbb{P}(\tau_j) \underbrace{\sum_{c\in\mathcal{C}} \frac{\mathbb{P}(c|G',\tau_j)\mathbb{P}(c|G'',\tau_j)}{\mathbb{P}(c|\tau_j)}}_{=\tilde{k}(\tau_j\circ G',\tau_j\circ G'')}
\end{aligned}$$

Step (a) exploits the fact that conditioning on the transformation τ_j and hypothesis c renders G', G'' statistically independent since the two instances are drawn i.i.d. from the same component of the mixture distribution; (b) applies Bayes rule twice. In step (c) we define the *generalized posterior agreement* as

$$\tilde{k}(G',G'') := \sum_{c\in\mathcal{C}} \frac{\mathbb{P}(c|G')\mathbb{P}(c|G'')}{\mathbb{P}(c)}.$$

From condition 2) of the definition of the transformation set $\mathbb{T}$, one can get that

$$\tilde{k}(\tau_j \circ G', \tau_j \circ G'') = \tilde{k}(G',G'') \tag{25}$$

Combining Equation 25 with (c) one can get,

$$\frac{\mathbb{P}(G',G'')}{\mathbb{P}(G')\mathbb{P}(G'')} = \tilde{k}(G',G'') \underbrace{\frac{\sum_{j=1}^{M} p_j \sum_{\tau_j \in \mathbb{T}} \mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_j)\mathbb{P}(\tau_j)}{\sum_{j=1}^{M} p_j \sum_{\tau_j \in \mathbb{T}} \mathbb{P}(G'|\tau_j)\mathbb{P}(\tau_j) \sum_{l=1}^{M} p_l \sum_{\tau_l \in \mathbb{T}} \mathbb{P}(G''|\tau_l)\mathbb{P}(\tau_l)}}_{\text{to be proved} = |\mathbb{T}|} \tag{26}$$

$$= \tilde{k}(G',G'')|\mathbb{T}| \tag{27}$$

Let us prove Equation 27 first of all. We simplify the term,

$$\frac{\sum_{j=1}^{M} p_j \sum_{\tau_j \in \mathbb{T}} \mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_j)\mathbb{P}(\tau_j)}{\sum_{j=1}^{M} p_j \sum_{\tau_j \in \mathbb{T}} \mathbb{P}(G'|\tau_j)\mathbb{P}(\tau_j) \sum_{l=1}^{M} p_l \sum_{\tau_l \in \mathbb{T}} \mathbb{P}(G''|\tau_l)\mathbb{P}(\tau_l)} =$$

$$\frac{\sum_{j=1}^{M} p_j \sum_{\tau_j \in \mathbb{T}} \mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_j)\mathbb{P}(\tau_j)}{\sum_{j=1}^{M} \sum_{l=1}^{M} \sum_{\tau_j \in \mathbb{T}} \sum_{\tau_l \in \mathbb{T}} p_j p_l \mathbb{P}(\tau_j)\mathbb{P}(\tau_l)\mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_l)} =$$

$$\frac{\sum_{j=1}^{M} p_j \sum_{\tau_j \in \mathbb{T}} \mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_j)\mathbb{P}(\tau_j)}{\sum_{j=1}^{M} \sum_{\tau_j \in \mathbb{T}} p_j^2 \mathbb{P}(\tau_j)^2 \mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_l) + \sum_{j \neq l} \sum_{\tau_j \in \mathbb{T}} \sum_{\tau_l \in \mathbb{T}} p_j p_l \mathbb{P}(\tau_j)\mathbb{P}(\tau_l)\mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_l)} \tag{28}$$

We further make the simplifying assumption that $\mathbb{P}(\tau_j) = \mathbb{P}(\tau_l) = 1/|\mathbb{T}|$, then

$$\Rightarrow \frac{\frac{1}{|\mathbb{T}|} \sum_{j=1}^{M} p_j \sum_{\tau_j \in \mathbb{T}} \mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_j)}{\frac{1}{|\mathbb{T}|^2} \left(\sum_{j=1}^{M} \sum_{\tau_j \in \mathbb{T}} p_j^2 \mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_l) + \sum_{j \neq l} \sum_{\tau_j \in \mathbb{T}} \sum_{\tau_l \in \mathbb{T}} p_j p_l \mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_l) \right)}$$

$$= |\mathbb{T}| \frac{\sum_{j=1}^{M} p_j \sum_{\tau_j \in \mathbb{T}} \mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_j)}{\left(\sum_{j=1}^{M} p_j^2 \sum_{\tau_j \in \mathbb{T}} \mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_l) + \sum_{j \neq l} p_j p_l \sum_{\tau_j \in \mathbb{T}} \sum_{\tau_l \in \mathbb{T}} \mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_l) \right)}$$

$$\overset{p_j = \Delta_{js}}{=} |\mathbb{T}| \frac{\sum_{\tau_s \in \mathbb{T}} \mathbb{P}(G'|\tau_s)\mathbb{P}(G''|\tau_s)}{\sum_{\tau_s \in \mathbb{T}} \mathbb{P}(G'|\tau_s)\mathbb{P}(G''|\tau_s) + \sum_{j \neq l} \underbrace{\Delta_{js}\Delta_{ls}}_{=0 \text{ for } l \neq j} \sum_{\tau_j \in \mathbb{T}} \sum_{\tau_l \in \mathbb{T}} \mathbb{P}(G'|\tau_j)\mathbb{P}(G''|\tau_l)}$$

$$= |\mathbb{T}|$$

Inserting Equation (27) into Equation (24) proves the claim that the mutual information is a lower bound on the information content:

$$\begin{aligned}\mathcal{I}(G';G'') &= \mathbb{E}_{G',G''} \log \frac{\mathbb{P}(G',G'')}{\mathbb{P}(G')\mathbb{P}(G'')} \\ &= \mathbb{E}_{G',G''} \log \tilde{k}(G',G'') \\ &= \mathbb{E}_{G',G''} \log \sum_{c\in\mathcal{C}} \frac{\mathbb{P}(c|G')\mathbb{P}(c|G'')}{\mathbb{P}(c)} \\ &\leq \mathbb{E}_{G',G''} \log |\mathcal{C}| \sum\nolimits_{c\in\mathcal{C}} \mathbb{P}(c|G')\mathbb{P}(c|G'') \\ &= I(G';G'').\end{aligned}$$

With the uniform distribution $\mathbb{P}(c) = |\mathcal{C}|^{-1}$, $\mathcal{I}(G';G'')$ is maximized and we derive the information content in Equation 2.

A.2 Proof of Lemma 1

Proof. For a specific cut $c := (S, \bar{S})$, assume the collections of normal vectors of all hyperplanes that give the cut c is $R(c)$, according to the random hyperplane rounding technique,

$$\begin{aligned}R(c) =& \{\boldsymbol{r} \in S_{n-1} \mid \boldsymbol{r}\cdot\boldsymbol{b}_i \geq 0, \forall i \in V\} \\ &\cup \{\boldsymbol{r} \in S_{n-1} \mid \boldsymbol{r}\cdot\boldsymbol{b}_i \leq 0, \forall i \in V\}\end{aligned} \tag{29}$$

So $R(c)$ is two times the unit spherical surface of $C(c)$. Considering the fact that normal vectors of all hyperplanes constitute the surface of unit sphere, we get the first equality in (21). Using simple geometrical knowledge, we can get the second and third equalities. ■

A.3 Proof of Lemma 2

Proof. $C' = C \Leftrightarrow$ any point in C' must be in $C \Leftrightarrow (\sum_1^n \lambda_i \boldsymbol{a}_i)\cdot\boldsymbol{b}_j \geq 0, 1 \leq i,j \leq n$ holds $\forall \lambda_i \geq 0 \Leftrightarrow (\lambda_1,\cdots,\lambda_n)\cdot A^T B \geq 0$ holds $\forall(\lambda_1,\cdots,\lambda_n) \geq 0$

So if $\exists k_1,\cdots,k_n > 0$, s.t. $A^{\mathrm{T}} = \mathrm{diag}(k_1,\cdots,k_n)B^{-1}$, one can get that $(\lambda_1,\cdots,\lambda_n)\cdot A^T B \geq 0$ holds $\forall(\lambda_1,\cdots,\lambda_n) \geq 0$, so $C' = C$. ■

B. Exactly evaluate the surface integral (Equation 22)

Exactly calculating probability of cuts involves evaluating the high dimensional surface integral in Equation 22, To do this, we first of all parametrize it using spherical polar coordinates, then transform it to be a multivariate integral. Writing $\boldsymbol{s} = \sum_1^n s_i \boldsymbol{e}_i$, we get:

$$\|A\boldsymbol{s}\|_2^2 = \sum_{i=1}^n \boldsymbol{a}_i\cdot\boldsymbol{a}_i s_i^2 + 2\sum_{i<j} \boldsymbol{a}_i\cdot\boldsymbol{a}_j s_i s_j = 1 + 2\sum_{i<j} \boldsymbol{a}_i\cdot\boldsymbol{a}_j s_i s_j \tag{30}$$

Plugging Equation 30 into 22 one can express the surface integral in a more manageable form

$$E = |\det(A)| \int_S (1 + 2\sum_{i<j} \boldsymbol{a}_i \cdot \boldsymbol{a}_j s_i s_j)^{-n/2} dS = |\det(A)| \int_S f^{-n/2}(\boldsymbol{s}) dS \tag{31}$$

where $f(\boldsymbol{s}) = 1+2\sum_{i<j} \boldsymbol{a}_i \cdot \boldsymbol{a}_j s_i s_j$. Then parametrizing by spherical polar coordinates $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_{n-1})$:

$$s_i = \cos(\theta_i) \prod_{j=1}^{i-1} \sin(\theta_j), i = 1, \cdots, n-1; s_n = \prod_{i=1}^{n-1} \sin(\theta_i) \tag{32}$$

for $0 \le \theta_i \le \pi/2, 0 \le i \le n-1$, considering that the Jacobian is $\prod_{i=1}^{n-2} \sin^{n-1-i}(\theta_i)$, substitute Equation 32 to Equation 31 it reaches the multivariate integral:

$$E = |\det(A)| \int_{\theta_1} \cdots \int_{\theta_{n-1}} \frac{\prod_{i=1}^{n-2} \sin^{n-1-i}(\theta_i)}{f^{n/2}(\boldsymbol{\theta})} d\theta_1 \cdots d\theta_{n-1} \tag{33}$$

C. Theoretical analysis of Algorithm 3

We will show that for a cut c with high ground truth probability $p_c := \mathbb{P}(c|V)$, the estimated cut probability $\hat{p}_c$ by uniform sampling in Algorithm 3 will be close to p_c with high probability.

Let $k = \#$samplings, random variable $X_i = 1$ means recovering cut c in the i^{th} sampling, $X_i = 0$ means not recovering c in the i^{th} sampling. So $\hat{p}_c = \sum_{i=1}^{k} X_i/k$, from the Chernoff-Hoeffding theorem (Hagerup and Rüb, 1990), for $\epsilon > 0$,

$$\begin{aligned}
\mathbb{P}(\hat{p}_c \ge p_c + \epsilon) &\le \left[\left(\frac{p_c}{p_c + \epsilon} \right)^{p_c+\epsilon} \left(\frac{1-p_c}{1-p_c-\epsilon} \right)^{1-p_c-\epsilon} \right]^k = e^{-D(p_c+\epsilon||p_c)k} \\
&\le \left(\frac{p_c}{p_c + \epsilon} \right)^{k(p_c+\epsilon)} \cdot e^{k\epsilon} \\
\mathbb{P}(\hat{p}_c \le p_c - \epsilon) &\le \left[\left(\frac{p_c}{p_c - \epsilon} \right)^{p_c-\epsilon} \left(\frac{1-p_c}{1-p_c+\epsilon} \right)^{1-p_c+\epsilon} \right]^k \\
&= e^{-D(p_c-\epsilon||p_c)k} \le \left(\frac{p_c}{p_c - \epsilon} \right)^{k(p_c-\epsilon)} \cdot e^{-k\epsilon}
\end{aligned}$$

where $D(\cdot)$ is the Kullback-Leibler divergence between two Bernoulli random variables.

So to ensure that with probability at most $\delta < 1$, the estimated probability $\hat{p}_c$ is at most ϵ-distant from the true probability p_c, one need to ensure that:

$$\max(e^{-D(p_c+\epsilon||p_c)k}, e^{-D(p_c-\epsilon||p_c)k}) \le \delta$$

Algorithm 4: Pseudo-code to calculate estimate of $\sum_c \mathbb{P}(c|G')\mathbb{P}(c|G'')$ when $k < |\mathcal{C}|$

Input: $\texttt{cutIndices}(G')$, $\texttt{cutIndices}(G'') \in \mathbb{R}^k$, wherein the indices are in ascending order

Output: estimate of $\sum_c \mathbb{P}(c|G')\mathbb{P}(c|G'')$

```
initialize idx1 = idx2 = 1, sum = 0;
while idx1 ≤ k && idx2 ≤ k do
    if cutIndices(G')_idx1 == cutIndices(G'')_idx2 then
        commonIdx = cutIndices(G')_idx1;
        cutNum1 = cutNum2 = 1;
        idx1++, idx2++ ;
        while idx1 ≤ k && commonIdx == cutIndices(G')_idx1 do
            cutNum1++, idx1++;
        while idx2 ≤ k && commonIdx == cutIndices(G'')_idx2 do
            cutNum2++, idx2++;
        sum+ = cutNum1 * cutNum2;
    else if cutIndices(G')_idx1 < cutIndices(G'')_idx2 then
        idx1++;
    else
        idx2++;
return ∑_c ℙ(c|G')ℙ(c|G'') ≈ sum/k^2
```

which is equivalent to:

$$k \geq \max\left(\frac{-\ln\delta}{D(p_c+\epsilon||p_c)}, \frac{-\ln\delta}{D(p_c-\epsilon||p_c)}\right) \tag{34}$$

which gives the lower bound of the sampling number k required to recover the ground truth p_c with probability δ at a specific error level ϵ.

D. Space-efficient implementation of Algorithm 3

When sampling number $k \geq |\mathcal{C}|$, use array $\texttt{cutFrequency} \in \mathbb{R}^{|\mathcal{C}|}$ to record cuts' frequency of occurrence, and the posterior agreement $\sum_c \mathbb{P}(c|G')\mathbb{P}(c|G'')$ is estimated as the inner product $\langle \texttt{cutFrequency}(G'), \texttt{cutFrequency}(G'')\rangle$.

When $k < |\mathcal{C}|$, use array $\texttt{cutIndices} \in \mathbb{R}^k$ to record indices of sampled cuts in each sampling, note that there would be duplicated cuts in $\texttt{cutIndices}$. Then sort the array $\texttt{cutIndices}$ to make the indices in it be in ascending order. Finally, use the way described by the pseudo-code in Algorithm 4 to calculate estimate of posterior agreement $\sum_c \mathbb{P}(c|G')\mathbb{P}(c|G'')$.