Electronic Colloquium on Computational Complexity, Report No. 88 (2006)

Balanced MAX 2-SAT might not be the hardest

Per Austrin*
KTH – Royal Institute of Technology, Stockholm

July 9, 2006

Abstract

We show that, assuming the Unique Games Conjecture, it is NP-hard to approximate MAX 2-SAT within $\alpha^{-}_{LLZ} + \epsilon$, where $0.9401 < \alpha^{-}_{LLZ} < 0.9402$ is the believed approximation ratio of the algorithm of Lewin, Livnat and Zwick [17].

This result is surprising considering the fact that balanced instances of MAX 2-SAT, i.e. instances where each variable occurs positively and negatively equally often, can be approximated within 0.9439. In particular, instances in which roughly 70% of the literals are unnegated variables and 30% are negated appear less amenable to approximation than instances where the ratio is 50%-50%.

1 Introduction

In their classic paper [8], Goemans and Williamson used semidefinite programming to construct 0.8785-approximation algorithms for MAX CUT and MAX 2-SAT, as well as a 0.7960-approximation algorithm for MAX DI-CUT. Since then, improved approximation algorithms based on semidefinite programming have been constructed for many other important NP-hard problems, including coloring of k-colorable graphs [12, 9], and fairly general versions of integer quadratic programming on the hypercube [3].

Meanwhile, the study of *inapproximability* has seen a perhaps even bigger revolution, starting with the discovery of the PCP Theorem [2, 1]. It has led to inapproximability results for a myriad of NP-hard problems, several of them tight (e.g. SET COVER [6], MAX CLIQUE [10], and MAX 3-SAT [11]) in the sense that they match the best known algorithmic results up to lower order terms.

However, for constraint satisfaction problems in which each constraint acts on two variables, tight results, or in some cases even any results, have been more elusive. As a possible means to remedy this, Khot [13] introduced the so called Unique Games Conjecture (UGC), asserting the existence of a very powerful two-prover system with some specific properties. Assuming the UGC, Khot showed superconstant hardness for MIN 2SAT-DELETION.

*E-mail: austrin@kth.se. Research funded by Swedish Research Council Project Number 50394001.

ISSN 1433-8092

Since then, the UGC has been shown to imply hardness for several other problems, including $2-\epsilon$ hardness for VERTEX COVER [15], $\alpha_{GW}+\epsilon$ hardness for MAX CUT [14] (where $\alpha_{GW} \approx 0.8785$ is the approximation ratio of the Goemans-Williamson algorithm), superconstant hardness for SPARSEST CUT [16], coloring 3-colorable graphs with as few colors as possible [5], and approximating MAX CLIQUE within $d/\text{poly} \log d$ in degree-d graphs [19]. Additionally, Khot and Vishnoi [16] used the UGC to show integrality gaps for SDP relaxations of some of the above-mentioned problems (the integrality gap of an instance is the quotient between the optimum value of the instance and the optimum value of the SDP relaxation). They do this by showing an integrality gap for a natural SDP relaxation of Unique Games and then "translating" this instance through the PCP reduction.

It should be noted that the status of the Unique Games Conjecture is highly uncertain; there is no substantial evidence supporting the conjecture (nor, for that matter, any substantial evidence against the conjecture).

For MAX 2-SAT and MAX DI-CUT, Goemans and Williamson's algorithms were improved first by Feige and Goemans [7], and then by Lewin, Livnat and Zwick [17] who obtained a 0.9401-approximation algorithm for MAX 2-SAT, and a 0.8740-approximation algorithm for MAX DI-CUT, and these stand as the current best results for both problems. It should be pointed out that these two ratios arise as the solutions of complex numeric optimization problems and, as far as we are aware of, it has not yet been proved formally that these are the actual optima, though there seems to be little doubt that this is the case.

For both problems, better approximation algorithms are known for the special case of so called *balanced instances*. For MAX 2-SAT this corresponds to the case when every variable occurs negated and unnegated equally often, and for MAX DI-CUT this corresponds to each vertex having the same indegree as outdegree. The approximation ratios achieved are ≈ 0.9439 and α_{GW} respectively, and they match the best known inapproximability ratios under the UGC [14].[1] The best known unconditional hardness for MAX 2-SAT is $21/22 \approx 0.9546$ [11].

It is natural to conjecture, especially considering these results, that balanced instances should be the hardest (and indeed, Khot et al. [14] do that), i.e. that we should always be able to use the presence of any linear terms to our advantage. However, as the main result of our paper shows, this might actually not be the case:

Theorem 1.1. *Assuming the Unique Games Conjecture, for any $\epsilon > 0$ it is NP-hard to approximate* MAX 2-SAT *within $\alpha^-_{LLZ}+\epsilon$, where $\alpha^-_{LLZ} \approx 0.94017$.*

Here, α^-_{LLZ} is the *believed* approximation ratio of Lewin et al.'s MAX 2-SAT algorithm mentioned above. In other words, assuming that their analysis of the algorithm is correct, Theorem 1.1 is tight. The (in our opinion very remote) possibility that their analysis is not correct, i.e. that the approximation ratio of their algorithm is smaller than α^-_{LLZ}, does not

[1] This is not very surprising, since the balanced versions of both problems are equivalent to the MAX CUT problem with a linear transformation on the scoring function.

affect Theorem 1.1, it would just indicate that it might not be tight, i.e. that MAX 2-SAT might be even harder than indicated by our result. The reason that we need to rely on the analysis of Lewin et al. being correct is that our PCP reduction is controlled by a parameter corresponding to a worst-case vector configuration for Lewin et al.'s algorithm. However, the reduction requires this vector configuration to be of a specific form. Fortunately, the worst configurations found by Lewin et al. are of this form.

An in our opinion quite surprising part of our result is the "amount" of imbalance: in our hard instances, every variable occurs positively more than twice as often as negatively (the ratio is roughly $70-30$)!

The proof relies on a careful analysis of the algorithm of Lewin, Livnat and Zwick. This analysis provides the optimal parameters for a PCP reduction which is similar (but slightly more involved) to Khot et al.'s reduction for MAX CUT.

The paper is organized as follows. In Section 2 we set up notation and give some necessary background. In Section 3, we discuss Lewin et al.'s MAX 2-SAT algorithm and its approximation ratio. In Section 4 we reduce UNIQUE LABEL COVER to MAX 2-SAT, establishing Theorem 1.1. In Section 5, we conclude by discussing some related open problems.

2 Preliminaries

We associate the boolean values true and false with -1 and 1, respectively. Thus, a disjunction $x \vee y$ is false iff $x = y = 1$.

2.1 MAX 2-SAT

A MAX 2-SAT instance Ψ on a set of n variables consists of a set of clauses, where each clause $\psi \in \Psi$ is a disjunction $l_1 \vee l_2$ on two literals, where each literal is either a variable or a negated variable, i.e. of the form bx_i for $b \in \{-1, 1\}$ and some variable x_i. Additionally, each clause ψ has a nonnegative weight $\mathrm{wt}(\psi)$ (by [4], weighted and unweighted MAX 2-SAT are equally hard to approximate, up to lower order terms). The MAX 2-SAT problem is to find an assignment $x \in \{-1, 1\}^n$ of the variables such that the sum of the weights of the satisfied clauses is maximised. MAX 2-SAT can be viewed as an integer programming problem by arithmetizing each clause $(b_1x_i \vee b_2x_j)$ as $\frac{3-b_1x_i-b_2x_j-b_1b_2x_ix_j}{4}$. Note that the latter expression is 1 if the clause is satisfied, and 0 otherwise. The value of an assignment $x \in \{-1, 1\}^n$ to Ψ is then

$$\mathrm{Val}_\Psi(x) = \sum_{\psi=(b_1x_i \vee b_2x_j)\in\Psi} \mathrm{wt}(\psi) \cdot \frac{3 - b_1x_i - b_2x_j - b_1b_2x_ix_j}{4}, \tag{1}$$

and we can write a MAX 2-SAT instance Ψ as the (quadratic) integer program

$$\begin{array}{l} \text{Maximize } \mathrm{Val}_\Psi(x) \\ \text{Subject to } x_i \in \{-1,1\} \quad \forall i \end{array} \tag{2}$$

In this paper, we will be especially interested in the family of MAX 2-SAT instances consisting of the following two clauses for every pair of variables x_i, x_j: the clause $(x_i \vee x_j)$ with weight $\mathrm{wt}_{ij} \cdot \frac{1+\Delta}{2}$, and the clause $(-x_i \vee -x_j)$ with weight $\mathrm{wt}_{ij} \cdot \frac{1-\Delta}{2}$, where the nonnegative weight wt_{ij} controls the "importance" of the pair x_i, x_j (we allow $\mathrm{wt}_{ij} = 0$), and $\Delta \in [-1,1]$ is a constant controlling the "imbalance" of the instance. Note that if $\Delta = \pm 1$ every variable occurs only positively/negatively, and the instance is trivially satisfiable, whereas if $\Delta = 0$ the instance is balanced and can be approximated within 0.9439. For our hard instances, we will use a carefully chosen Δ which will be approximately 0.3673 (in other words, the relative weight on the positive clauses will be roughly $\frac{1+0.3673}{2} \approx 68\%$).

We will use the terminology *Δ-mixed clause (of weight wt)* for a pair of clauses $(x_i \vee x_j)$ with weight $\mathrm{wt} \cdot \frac{1+\Delta}{2}$ and $(-x_i \vee -x_j)$ with weight $\mathrm{wt} \cdot \frac{1-\Delta}{2}$. For a MAX 2-SAT instance Ψ of the above form (i.e. an instance that can be viewed as a set of Δ-mixed clauses), $\mathrm{Val}_\Psi(x)$ can be rewritten as (note that the effect of Δ on the integer program simply constitutes a dampening of the linear terms)

$$\mathrm{Val}_\Psi(x) = \sum_{i<j} \mathrm{wt}_{ij} \frac{3 - \Delta x_i - \Delta x_j - x_i x_j}{4}. \tag{3}$$

2.2 Harmonic analysis of Boolean functions

Fourier analysis (of Boolean functions) is a crucial tool in most strong inapproximability results. Here we review some important concepts. We denote by μ_q^n the probability distribution on $\{-1,1\}^n$ where each bit is set to -1 with probability q, independently, and we let B_q^n be the probability space $\left(\{-1,1\}^n, \mu_q^n\right)$. Note that in this paper we work with general distributions rather than the more well-known case of uniform distributions (i.e. where $q = 1/2$).

We define a scalar product on the space of functions from B_q^n to $\mathbb{R}$ by

$$\langle f, g \rangle = \mathop{\mathbb{E}}_{x \in B_q^n}[f(x)g(x)], \tag{4}$$

and for each $S \subseteq [n]$ the function $U_q^S : B_q^n \to \mathbb{R}$ by $U_q^S(x) = \prod_{i \in S} U_q(x_i)$ where

$$U_q(x_i) = \begin{cases} -\sqrt{\frac{1-q}{q}} & \text{if } x_i = -1 \\ \sqrt{\frac{q}{1-q}} & \text{if } x_i = 1 \end{cases}.$$

Proposition 2.1. *The set of functions $\{U_q^S\}_{S \subseteq [n]}$ forms an orthonormal basis w.r.t. the scalar product $\langle \cdot, \cdot \rangle$ (i.e. expected value over the distribution μ_q^n).*

A proof of this well-known fact can be found in Appendix C. Thus, any function $f : B_q^n \to \mathbb{R}$ can be written as

$$f(x) = \sum_{S \subseteq [n]} \hat{f}_S U_q^S(x).$$

The coefficients $\hat{f}_S = \langle f, U_q^S \rangle = \mathbb{E}_x[f(x) U_q^S(x)]$ are the Fourier coefficients of the function f. It is a fairly straight-forward exercise to verify the basic identities $\mathbb{E}_x[f(x)] = \hat{f}_\emptyset$, $V_x[f(x)] = \sum_{S \neq \emptyset} \hat{f}_S^2$ and $\langle f, g \rangle = \sum_{S \subseteq [n]} \hat{f}_S \hat{g}_S$.

Definition 2.2. The *influence* of the variable i on the function $f : B_q^n \to \mathbb{R}$ is

$$\mathrm{Inf}_i(f) = \mathop{\mathbb{E}}_x [\, V_{x_i}[f(x) \,|\, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]\,] \tag{5}$$

Intuitively, the influence of the variable i is a measure of how much the variable i is able to change the value of f once we have fixed the other $n - 1$ variables.

Proposition 2.3.

$$\mathrm{Inf}_i(f) = \sum_{\substack{S \subseteq [n] \\ i \in S}} \hat{f}_S^2. \tag{6}$$

A proof is given in Appendix C. Motivated by the Fourier-representation formulation of influence, we define the slightly stronger concept of low-degree influence, crucial to PCP applications.

Definition 2.4. For $k \in \mathbb{N}$, the *low-degree influence* of the variable i on the function $f : B_q^n \to \mathbb{R}$ is

$$\mathrm{Inf}_i^{\leq k}(f) = \sum_{\substack{S \subseteq [n] \\ i \in S \\ |S| \leq k}} \hat{f}_S^2. \tag{7}$$

A nice property of the low-degree influence is the fact that $\sum_i \mathrm{Inf}_i^{\leq k}(f) \leq k$, implying that the number of variables having low-degree influence more than τ must be small (think of k and τ as constants not depending on the number of variables n). Informally, one can think of the low-degree influence as a measure of how close the function f is to depending *only* on the variable i, i.e., for the case of boolean-valued functions, how close f is to being the long code of i (or its negation).

Next, we define the Beckner operator T_ρ on a function $f : B_q^n \to \mathbb{R}$. For the unbiased distribution $q = 1/2$, $T_\rho f(x)$ is simply the expectation of $f(y)$ over a random variable y that is ρ-correlated with x. For biased distributions, the definition is a bit more complicated.

Definition 2.5. Given $\rho \in [-1, 1]$ satisfying $\rho \geq -\frac{\min(q, 1-q)}{\max(q, 1-q)}$, the Beckner operator T_ρ on a function $f : B_q^n \to \mathbb{R}$ is defined by

$$T_\rho f(x) = \mathop{\mathbb{E}}_y [f(y)]. \tag{8}$$

x_i	b	$\Pr[y_i = b]$
1	1	$1 - q(1-\rho)$
1	-1	$q(1-\rho)$
-1	1	$(1-q)(1-\rho)$
-1	-1	$1 - (1-q)(1-\rho)$

Table 1: Distribution of y_i depending on x_i.

where the expectation is over an n-bit string y in which each bit y_i is picked independently as follows: if $x_i = 1$ then $y_i = -x_i$ with probability $q(1-\rho)$, and if $x_i = -1$ then $y_i = -x_i$ with probability $(1-q)(1-\rho)$ (see Table 1).

Note that the lower bound on ρ is needed to make this a valid probability distribution. For $\rho \geq 0$, the probability distribution of y_i can be formulated as follows: with probability ρ, we let $y_i = x_i$, and with probability $1-\rho$, we pick y_i from B_q^1.

The operator T_ρ has the following very nice effect on the Fourier representation of f.

Proposition 2.6.

$$T_\rho f(x) = \sum_{S \subseteq [n]} \rho^{|S|} \hat{f}_S U_q^S(x). \tag{9}$$

Again, a proof is given in Appendix C.

Definition 2.7. The *noise correlation* between $f : B_q^n \to \mathbb{R}$ and $g : B_q^n \to \mathbb{R}$ is given by

$$\mathbb{S}_\rho(f, g) = \langle f, T_\rho g \rangle = \sum_{S \subseteq [n]} \rho^{|S|} \hat{f}_S \hat{g}_S \tag{10}$$

In other words, the well-studied noise stability of a function f, is simply $\mathbb{S}_\rho(f) := \mathbb{S}_\rho(f, f)$.

2.3 Functions in Gaussian space

We denote by $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ the standard normal density function, by $\Phi(x) = \int_{-\infty}^{x} \phi(t)\mathrm{d}t$ the standard normal distribution function, and by Φ^{-1} the inverse of Φ.

As with functions on the hypercube, we define a scalar product on functions $f, g : \mathbb{R}^n \to \mathbb{R}$ by (we abuse notation slightly by using the same notation as for scalar products on functions from the hypercube)

$$\langle f, g \rangle = \mathop{\mathbb{E}}_{x}[f(x)g(x)], \tag{11}$$

where the expectation is over an n-dimensional standard Gaussian, i.e. each component being a standard $N(0,1)$ random variable. The counterpart of the Beckner operator in

Gaussian space is the *Ornstein-Uhlenbeck* operator U_ρ:

$$U_\rho f(x) = \mathop{\mathbb{E}}_{y}\left[f(\rho x + \sqrt{1-\rho^2} y)\right]. \tag{12}$$

Note that $\rho x + \sqrt{1-\rho^2} y$ is an $N(0,1)$ variable with covariance ρ with x. Analogously to the case of Boolean functions, we define the noise correlation between two functions $f, g : \mathbb{R}^n \to \mathbb{R}$ as $\mathbb{S}_\rho(f,g) = \langle f, U_\rho g\rangle$ where $\rho \in [-1,1]$. For $\mu \in [-1,1]$ we denote by $\chi_\mu : \mathbb{R} \to [0,1]$ the indicator function of an interval $(-\infty, t)$, where t is chosen so that $\mathbb{E}[\chi_\mu] = \frac{1-\mu}{2}$ (i.e. $t = \Phi^{-1}\left(\frac{1-\mu}{2}\right)$).

Definition 2.8. For $\rho, \mu_1, \mu_2 \in [-1,1]$, define

$$\Gamma_\rho(\mu_1,\mu_2) = \mathbb{S}_\rho(\chi_{\mu_1}, \chi_{\mu_2}) = \Pr[X_1 \le t_1 \wedge X_2 \le t_2], \tag{13}$$

where $t_i = \Phi^{-1}\left(\frac{1-\mu_i}{2}\right)$ and where $X_1, X_2 \in N(0,1)$ with covariance ρ.

In other words, Γ_ρ is just the bivariate normal distribution function with a transformation on the input. Analogously to noise stability, we define $\Gamma_\rho(\mu) = \Gamma_\rho(\mu,\mu)$. The following nice property of Γ_ρ will be very useful to us.

Proposition 2.9. *For all $\rho \in [-1,1]$, $\mu_1, \mu_2 \in [-1,1]$, we have*

$$\Gamma_\rho(-\mu_1,-\mu_2) = \Gamma_\rho(\mu_1,\mu_2) + \mu_1/2 + \mu_2/2 \tag{14}$$

A proof can be found in Appendix D.

2.4 Thresholds are extremely noise-correlated

For proving hardness of MAX CUT, Khot et al. [14] made a conjecture called Majority Is Stablest, essentially stating that any boolean function with noise stability significantly higher than the majority function must have a variable with high low-degree influence (and thus in a sense be close to a long code). Majority Is Stablest was subsequently proved by Mossel et al. [18], using a very powerful invariance principle which, essentially, allows for considering the corresponding the problem over Gaussian space instead. For our result, we will use a strengthening of Majority is Stablest due to Dinur et al. [5], stated here in a simplified form (Dinur et al. state their theorem for general symmetric Markov operators, but we are only interested in the Beckner operator T_ρ):

Theorem 2.10 (Dinur et al. [5], Theorem 3.1)**.** *For any $\epsilon > 0$, $q \in (0,1)$ and $\rho \in (-1,1)$ there is a $\tau > 0$, $k \in \mathbb{N}$ such that if $f, g : B_q^n \to [0,1]$ with $\mathbb{E}[f] = \frac{1-\mu_f}{2}$, $\mathbb{E}[g] = \frac{1-\mu_g}{2}$, and*

$$\min\left(\mathrm{Inf}_i^{\le k}(f), \mathrm{Inf}_i^{\le k}(g)\right) \le \tau$$

for all $i \in [n]$*, then*

$$\mathbb{S}_\rho(f,g) \leq \left\langle \chi_{\mu_f}, U_{|\rho|}\chi_{\mu_g} \right\rangle + \epsilon \tag{15}$$
$$\mathbb{S}_\rho(f,g) \geq \left\langle \chi_{\mu_f}, U_{|\rho|}(1-\chi_{-\mu_g}) \right\rangle - \epsilon \tag{16}$$

We remark that Dinur et al. state their theorem for functions on the m-ary hypercube $[m]^n$ under *uniform* distribution, rather than the biased hypercube B_q^n. However, their proof does not make use of any properties that only hold for the uniform distribution (in particular, Mossel et al.'s invariance principle is applicable to the case of non-uniform distribution), and so, it is a fairly straight-forward exercise to redo their proof for biased distributions.

As a simple Corollary to Theorem 2.10, we have

Corollary 2.11. *Let* $\epsilon > 0$, $q \in (0,1)$ *and* $\rho \in (-1,0)$. *Then there is a* $\tau > 0$, $k \in \mathbb{N}$ *such that for every function* $f : B_q^n \to [-1,1]$ *satisfying* $\mathbb{E}[f] = \mu$ *and* $\mathrm{Inf}_i^{\leq k}(f) \leq \tau$ *for all* i*, we have*

$$\mathbb{S}_\rho(f) \geq 4\Gamma_\rho(\mu) + 2\mu - 1 - \epsilon \tag{17}$$

Proof. Set $\tilde{f} = \frac{1-f}{2}$, $\tilde{\mu} = \mathbb{E}[\tilde{f}] = \frac{1-\mu}{2}$. Thus, $\mathbb{S}_\rho(f) = 4\,\mathbb{S}_\rho(\tilde{f}) - 4\tilde{\mu} + 1 = 4\,\mathbb{S}_\rho(\tilde{f}) + 2\mu - 1$. By Theorem 2.10,

$$\mathbb{S}_\rho(\tilde{f}) \geq \left\langle \chi_\mu, U_{|\rho|}(1-\chi_{-\mu}) \right\rangle - \epsilon/4 \tag{18}$$

for any f where every variable has sufficiently small low-degree influence. Now, note that

$$\begin{aligned}(U_{|\rho|}(1-\chi_{-\mu}))(x) &= \Pr_y\left[|\rho|x + \sqrt{1-\rho^2}y \geq \Phi^{-1}(1-\tilde{\mu})\right] \\ &= \Pr_y\left[-|\rho|x + \sqrt{1-\rho^2}y \leq \Phi^{-1}(\tilde{\mu})\right] = U_{-|\rho|}\chi_\mu(x).\end{aligned}$$

Combining this with Equation (18) and the definition of Γ_ρ, we get

$$\mathbb{S}_\rho(f) \geq 4\Gamma_{-|\rho|}(\mu) + 2\mu - 1 - \epsilon. \tag{19}$$

Finally, using that $\rho < 0$, we obtain the desired form. □

2.5 The Unique Games Conjecture (UGC)

The Unique Games Conjecture was introduced by Khot [13] as a possible means to obtain inapproximability for constraint satisfaction problems in which each constraint acts over two variables. As is common, we will formulate it in terms of a Label Cover problem.

Definition 2.12. An instance $X = (V, E, \mathrm{wt}, [L], \{\sigma_e^v, \sigma_e^w\}_{e=\{v,w\}\in E})$ of UNIQUE LABEL COVER is defined as follows: given is a weighted graph $G = (V, E)$ (which may have multiple edges) with weight function $\mathrm{wt} : E \to (0,1]$, a set $[L]$ of allowed labels, and for each edge $e = \{v, w\} \in E$ two permutations $\sigma_e^v, \sigma_e^w \in \mathfrak{S}_L$ such that $\sigma_e^w = (\sigma_e^v)^{-1}$, i.e. they are each others inverse. We say that a function $l : V \to [L]$ (called a labelling of the vertices) satisfies an edge $e = \{v, w\}$ if $\sigma_e^v(l(v)) = l(w)$, or equivalently, if $\sigma_e^w(l(w)) = l(v)$. The value of l is the total weight of edges satisfied by it, i.e.

$$\mathrm{Val}_X(l) = \sum_{\substack{e \\ l \text{ satisfies } e}} \mathrm{wt}(e) \tag{20}$$

The value of X is the maximum fraction of satisfied edges for any labelling, i.e.

$$\mathrm{Val}(X) = \max_l \mathrm{Val}_X(l). \tag{21}$$

WLOG, we will always assume that $\sum_e \mathrm{wt}(e) = 1$, i.e. that wt is in fact a probability distribution over the edges of X. We denote by $E(v)$ the subset of edges adjacent to v, i.e. $E(v) = \{\, e \,|\, v \in e \,\}$. The probability distribution wt induces a natural probability distribution on the vertices of X where the probability of choosing v is $\frac{1}{2}\sum_{e\in E(v)} \mathrm{wt}(e)$, and wt also induces a natural distribution on the edges of $E(v)$ where the probability of choosing e is $\frac{\mathrm{wt}(e)}{\sum_{e\in E(v)} \mathrm{wt}(e)}$.

Whenever we speak of choosing a random element of V, E or $E(v)$, it will be according to these probability distributions, but to simplify the presentation, we will simply refer to it as a random element. For the same reason we will refer to a fraction c of the elements of V, E or $E(V)$ when in fact we mean a set of vertices/edges with probability mass c.

A UNIQUE LABEL COVER problem where G is bipartite can be viewed as a two-prover (one-round) game in which the acceptance predicate of the verifier is such that given the answer for one of the provers, there is always a unique answer from the other prover such that the verifier accepts. The probability that the verifier accepts assuming that the provers use an optimal strategy is then $\mathrm{Val}(X)$. Hence the terminology "Unique Games". We will be interested in the gap version of the UNIQUE LABEL COVER problem, which we define as follows.

Definition 2.13. GAP-UNIQUE LABEL COVER$_{\eta,\gamma,L}$ is the problem of, given a UNIQUE LABEL COVER instance X with label set $[L]$, determine whether $\mathrm{Val}(X) \geq 1-\eta$ or $\mathrm{Val}(X) \leq \gamma$.

Khot's Unique Games Conjecture (UGC) then asserts that the gap version is hard to solve for arbitrarily small η and γ, provided we take a sufficiently large label set.

Conjecture 2.14 (Unique Games Conjecture [13])**.** *For every $\eta > 0$, $\gamma > 0$, there is a constant $L > 0$ such that* GAP-UNIQUE LABEL COVER$_{\eta,\gamma,L}$ *is NP-hard.*

Thus, if the UGC is true, then any problem $\mathcal{P}$ such that GAP-UNIQUE LABEL COVER$_{\eta,\gamma,L}$ can be reduced to $\mathcal{P}$ in polynomial time for all constants η, γ, L is NP-hard. Note that even if the UGC turns out to be false, it might still be the case that GAP-UNIQUE LABEL COVER$_{\eta,\gamma,L}$ is hard in the sense of not being solvable in polynomial time, and such a (weaker) hardness would also apply to MAX 2-SAT and (as far as we are aware, all) other problems for which hardness has been shown under the UGC.

3 Approximating MAX 2-SAT

To approximate MAX 2-SAT, the common approach is to relax the integer program Equation (2) to a semidefinite program by relaxing each variable x_i to a vector $v_i \in \mathbb{R}^{n+1}$. In addition, we introduce the variable $v_0 \in \mathbb{R}^{n+1}$, which is supposed to encode the value "false". The constraint $x_i \in \{-1, 1\} = S^0$ translates to the constraint that $v_i \in S^n$, i.e. that each vector v_i should be a unit vector. The value of an assignment $v = (v_0, \ldots, v_n) \in (S^n)^{n+1}$ to the relaxation is then

$$\text{SDP-Val}_\Psi(v) = \sum_{\psi=(b_1x_i \vee b_2x_j)\in\Psi} \text{wt}(\psi) \cdot \frac{3 - b_1v_i \cdot v_0 - b_2v_j \cdot v_0 - b_1b_2v_i \cdot v_j}{4}, \tag{22}$$

where $v_i \cdot v_j$ is the standard inner product on vectors in $\mathbb{R}^n$.

This semidefinite relaxation was studied by Goemans and Williamson [8]. For their improved approximation algorithm, Feige and Goemans [7] considered a strengthening of this semidefinite program, by adding, for each triple $\{v_i, v_j, v_k\} \subseteq \{v_0, \ldots, v_n\}$ the triangle inequalities

$$\begin{aligned} v_i \cdot v_j + v_i \cdot v_k + v_j \cdot v_k &\geq -1 & \qquad -v_i \cdot v_j + v_i \cdot v_k - v_j \cdot v_k &\geq -1 \\ v_i \cdot v_j - v_i \cdot v_k - v_j \cdot v_k &\geq -1 & \qquad -v_i \cdot v_j - v_i \cdot v_k + v_j \cdot v_k &\geq -1. \end{aligned}$$

These are equivalent to inequalities of the form $||v_i - v_j||^2 + ||v_j - v_k||^2 \geq ||v_i - v_k||^2$, which clearly hold for the case that all vectors lie in a one-dimensional subspace of S^n (so this is still a relaxation of the original integer program), but may not necessarily be true otherwise. They are also equivalent to requiring that the value of any possible clause on the variables $x_1, \ldots, x_n$ is at most 1 which, again, clearly holds for a boolean solution.

In general, we cannot find the exact optimum of a semidefinite program. It is however possible to find the optimum to within an arbitrarily small additive factor. We ignore this point for notational convenience and assume that we can solve the semidefinite program exactly.

Given solution vectors $(v_0, \ldots, v_n)$ maximizing $\text{SDP-Val}_\Psi(v)$, we will produce a solution $(x_1, \ldots x_n) \in \{-1, 1\}^n$ using some rounding method, which will typically be randomized. For consistency, we require that this rounding method always rounds v_i and $-v_i$ to opposite values. To determine the approximation ratio of the algorithm, we analyze the worst possible approximation ratio on the clause $(x_i \vee x_j)$ for any vector configuration.[2] This gives the a lower bound on the approximation ratio:

$$\min_{v\in(S^n)^{n+1}} \frac{\mathbb{E}[3 - x_i - x_j - x_ix_j]}{3 - v_0 \cdot v_i - v_0 \cdot v_j - v_i \cdot v_j}, \tag{23}$$

[2]Note that because of the consistency requirement, the approximation ratio of the clause $(-x_i \vee x_j)$ equals the approximation ratio of the clause $(x_i \vee x_j)$ with v_i negated, and similarly for other clauses with negated variables.

where the minimum is over all *feasible* vector solutions to the SDP, and the expected value is over the randomness of the rounding method. Typically, the rounding of the vector v_i will only depend on v_0 and v_i, and so the minimum in Equation (23) only needs to be taken over the three vectors v_0, v_i and v_j.

3.1 The LLZ algorithm

The best approximation algorithm known for MAX 2-SAT (hereafter referred to as the LLZ algorithm) is due to Lewin, Livnat and Zwick [17]. It uses the SDP relaxation described above, including the triangle inequalities. In order to describe the rounding method, it is convenient to define some notation. Given a solution $(v_0, \ldots, v_n)$ to the SDP, we define $\xi_i = v_0 \cdot v_i$ and $v_i = \xi_i v_0 + \sqrt{1-\xi_i^2}\tilde{v}_i$, i.e. $\tilde{v}_i$ is the part of v_i orthogonal to v_0, normalized to a unit vector.

Lewin et al. consider the following general class of rounding methods, which they call $\mathcal{THRESH}^-$: First, a standard normal random vector r is chosen in the n-dimensional subspace of $\mathbb{R}^{n+1}$ orthogonal to v_0. Then, the variable x_i is set to true iff $\tilde{v}_i \cdot r \leq T(\xi_i)$, where the threshold function $T(\cdot)$ is (almost) arbitrary, and it is convenient for us to have it on the form

$$T(x) = \Phi^{-1}\left(\frac{1-a(x)}{2}\right), \tag{24}$$

where $a : [-1, 1] \to [-1, 1]$ is some function which is (almost) arbitrary.[3] The consistency requirement on the rounding method translates to requiring that T is an odd function (or equivalently, that a is an odd function).

The reason that it is natural to formulate T in terms of the function a becomes evident when we analyze the performance ratio of the algorithm. Note that $\tilde{v}_i \cdot r$ is a standard $N(0, 1)$ variable, implying that x_i is set to true with probability $\frac{1-a(\xi_i)}{2}$. In other words, the expected value of x_i is simply $\mathbb{E}[x_i] = a(\xi_i)$, and thus, we can think of the function a as controlling exactly how much we lose on the linear terms when we round the solution to the semidefinite program.

In order to evaluate the performance of the algorithm, we also need to analyze performance on the quadratic terms, which we do by analyzing the probability that two variables x_i and x_j are rounded to the same value. Let $\rho := v_i \cdot v_j$ and $\tilde{\rho} := \tilde{v}_i \cdot \tilde{v}_j = \frac{\rho - \xi_i\xi_j}{\sqrt{(1-\xi_i^2)(1-\xi_j^2)}}$. It is readily verified that the scalar products $\tilde{v}_i \cdot r$ and $\tilde{v}_j \cdot r$ are standard $N(0, 1)$ variables with covariance $\tilde{\rho}$, and thus, the probability that $\tilde{v}_i \cdot r \leq T(\xi_i)$ and $\tilde{v}_j \cdot r \leq T(\xi_j)$ is simply $\Gamma_{\tilde{\rho}}(a(\xi_i), a(\xi_j))$. By symmetry, the probability that both x_i and x_j are set to false is

[3] In the notation of [17], this corresponds to setting $S(x) = T(x)\sqrt{1-x^2}$, or $a(x) = 1 - 2\Phi\left(S(x)/\sqrt{1-x^2}\right)$ (we may, w.l.o.g., assume that $\xi_i \neq \pm 1$ for all i).

$\Gamma_{\tilde{\rho}}(-a(\xi_i), -a(\xi_j))$. Thus, the expected value of the term $x_i x_j$ is

$$\begin{aligned} 2\Pr[x_i = x_j] - 1 &= 2\left(\Gamma_{\tilde{\rho}}(a(\xi_i), a(\xi_j)) + \Gamma_{\tilde{\rho}}(-a(\xi_i), -a(\xi_j))\right) - 1 \\ &= 4\Gamma_{\tilde{\rho}}(a(\xi_i), a(\xi_j)) + a(\xi_i) + a(\xi_j) - 1 \end{aligned} \tag{25}$$

(where we used Proposition 2.9), and the expected value of the clause $x_i \vee x_j$ becomes

$$\frac{3 - \mathbb{E}[x_i] - \mathbb{E}[x_j] - \mathbb{E}[x_i x_j]}{4} = \frac{2 - a(\xi_i) - a(\xi_j) - 2\Gamma_{\tilde{\rho}}(a(\xi_i), a(\xi_j))}{2} \tag{26}$$

For their best algorithm, Lewin et al. choose $a(x) := \beta \cdot x$ to be a linear function, where $\beta \approx 0.94016567$, i.e. the apparent approximation ratio. Note that this choice [20] is not the same as that described in [17] but is more natural and (apparently) achieves the same approximation ratio since its behavior around the worst case configurations is the same. See Appendix B for details on the difference between these two rounding functions. Define

$$\alpha_\beta(\xi_i, \xi_j, \rho) = \frac{4 - 2\beta(\xi_i + \xi_j) - 4\Gamma_{\tilde{\rho}}(\beta\xi_i, \beta\xi_j)}{3 - \xi_i - \xi_j - \rho}, \tag{27}$$

i.e. the expected approximation ratio of the configuration (ξ_i, ξ_j, ρ), using a specific choice of β. Let

$$\alpha(\beta) = \min_{\xi_i, \xi_j, \rho} \alpha_\beta(\xi_i, \xi_j, \rho), \tag{28}$$

i.e. a lower bound on the approximation ratio achieved for a specific β, where (ξ_i, ξ_j, ρ) ranges over all configurations satisfying the triangle inequalities. Finally, let

$$\alpha_{LLZ} = \max_{\beta \in [-1,1]} \alpha(\beta), \tag{29}$$

i.e. a lower bound on the best possible approximation ratio when letting a be a linear function.

3.2 Simple configurations

We represent a configuration for the SDP by the three scalar products (ξ_i, ξ_j, ρ), where $\rho = v_i \cdot v_j$. When showing hardness of MAX 2-SAT, we will reduce UNIQUE LABEL COVER to MAX 2-SAT. The reduction is parametrized by a configuration (ξ_i, ξ_j, ρ) of the SDP. However, the reduction needs the configuration to be of a specific form.

First, it needs the configuration to satisfy $\xi_i = \xi_j$, in other words, that both v_i and v_j have the same angle to v_0. This restriction is quite natural; considering the symmetry of the linear terms in the quadratic program, it seems intuitive that the weight on the two linear terms should be distributed fifty-fifty for a worst case configuration, i.e. that $\xi_i = \xi_j$.

Second, the reduction needs the configuration to satisfy $-2|\xi_i| + \rho = -1$, in other words, that we have equality in one of the triangle inequalities. This restriction is also quite natural; the triangle inequalities cut away a part of the configuration space in which there are extremely bad configurations, and sticking as close as possible to this part of the configuration space would intuitively seem like a good approach for finding bad configurations.

We will refer to a configuration satisfying the two criterions, i.e. a configuration of the form $(\xi, \xi, -1 + 2|\xi|)$ for some $\xi \in [-1, 1]$, as a *simple configuration* ξ. The worst configurations found by Lewin et al. are roughly $(0.169, 0.169, -0.662)$ and $(-0.169, -0.169, -0.662)$, both being simple configurations.

Motivated by this restriction to simple configurations, we define

$$\alpha_\beta^-(\xi) = \alpha_\beta(\xi, \xi, -1 + 2|\xi|) = \frac{2 - 2\beta\xi - 2\Gamma_{\tilde{\rho}}(\beta\xi)}{2 - \xi - |\xi|} \tag{30}$$

to be the expected approximation ratio on a specific simple configuration ξ, where $\tilde{\rho} = \frac{-1+2|\xi|-\xi^2}{1-\xi^2} = \frac{|\xi|-1}{|\xi|+1}$ is the value of $\tilde{\rho}$ for the simple configuration ξ. Analogously to $\alpha(\beta)$ and α_{LLZ}, let

$$\alpha^-(\beta) = \min_{\xi \in [-1,1]} \alpha_\beta^-(\xi) \tag{31}$$

$$\alpha_{LLZ}^- = \max_{\beta \in [-1,1]} \alpha^-(\beta), \tag{32}$$

i.e. lower bounds on the approximation ratio for a specific choice of β and the best approximation ratio for any choice of β, when only considering simple configurations. Clearly, we have $\alpha_{LLZ} \leq \alpha_{LLZ}^-$, and unless Lewin et al.'s analysis is wrong, we have equality. In Appendix A.1, we briefly discuss the actual numeric value of α_{LLZ}^-.

It is possible to show that the right hand side of Equation (32) is indeed maximised by setting $\beta = \alpha_{LLZ}^-$ (see Appendix A.2 for a proof), and in fact, we need this for obtaining an expression for α_{LLZ}^- that matches the inapproximability yielded by the reduction from UNIQUE LABEL COVER.

3.3 Approximation of Δ-mixed clauses

In order to be able to show matching inapproximability, we give another formulation of α_{LLZ}^-. We show that for an appropriately chosen configuration ξ and imbalance Δ, we can not get a better approximation ratio than α_{LLZ}^- on the configuration ξ for a Δ-mixed clause even if we change the value of β. This is not quite as trival as it may sound—for the regular clause $(x_i \vee x_j)$, we can do better on any specific configuration ξ by adjusting the value of β—if $\xi < 0$ then increasing β will give a better approximation ratio, and if $\xi > 0$ then decreasing β will give a better approximation ratio—but in doing so we decrease the performance on the configuration $-\xi$, i.e. on the clause $(-x_i \vee -x_j)$. Formally, we show (see Appendix A.3 for a proof):

i:th bit		Probability
x_1	x_2	
1	1	$(\|\xi\| + \xi)/2 = 0$
-1	1	$(1 - \|\xi\|)/2 = (1+\xi)/2$
1	-1	$(1 - \|\xi\|)/2 = (1+\xi)/2$
-1	-1	$(\|\xi\| - \xi)/2 = -\xi$

Table 2: Distribution of the i:th bit of x_1 and x_2 (recall that $\xi < 0$).

Proposition 3.1. *There are $\xi \in (-1,0)$ and $\Delta \in (-1,1)$ such that*

$$\alpha^-_{LLZ} = \max_{\mu \in [-1,1]} \frac{2 - (1+\Delta)\mu - 2\Gamma_{\tilde{\rho}}(\mu)}{2 - \Delta\xi - |\xi|}, \tag{33}$$

where $\tilde{\rho} = \frac{|\xi|-1}{|\xi|+1}$.

The values of ξ and Δ given by Proposition 3.1 are roughly $\xi \approx -0.1625$, $\Delta \approx 0.3673$. The difference in ξ from the previously mentioned worst case value ± 0.169 from [17] is due to the fact that we use a slightly different rounding function. Again, see Appendix B for details on the difference.

We remark that the choice of sign for ξ is arbitrary (essentially, it corresponds to the choice of whether most of the variable occurences in our hard MAX 2-SAT instance should be positive or negative), the proposition holds for $\xi \in (0,1)$ as well.

4 Reduction from UNIQUE LABEL COVER

In this section, we reduce UNIQUE LABEL COVER to MAX 2-SAT. Let $\epsilon > 0$. We will show hardness of approximating MAX 2-SAT within $\alpha^-_{LLZ} + \mathcal{O}(\epsilon)$. Let $\eta > 0$ and $\gamma > 0$ be parameters which will be chosen sufficiently small and let L be the corresponding label size given by the UGC. We will reduce GAP-UNIQUE LABEL COVER$_{\eta,\gamma,L}$ to the problem of approximating MAX 2-SAT via a PCP verifier whose queries correspond to checking a Δ-mixed MAX 2-SAT clause.

The reduction is controlled by a parameter $\xi \in (-1,0)$ and an imbalance parameter $\Delta \in (-1,1)$. The values of these will be chosen later.

Given is a UNIQUE LABEL COVER instance $X = (V, E, [L], \{\sigma^v_e\}_{e=\{v,w\}\in E})$. A proof Σ that X is $(1-\eta)$-satisfiable will consist of supposed long codes of the labels of all $v \in V$. Denote by $f_v : \{-1,1\}^L \to \{-1,1\}$ the purported long code of the label of vertex v. For a permutation $\sigma \in \mathfrak{S}_L$ and $x = x_1 \dots x_L \in \{-1,1\}^L$, we let $\sigma x = x_{\sigma(1)} \dots x_{\sigma(L)}$. The PCP verifier $\mathcal{V}$ is described in Algorithm 1.

Algorithm 1: The verifier $\mathcal{V}$

$\mathcal{V}(X, \Sigma = \{f_v\}_{v \in V})$

(1) Pick a random $v \in V$.

(2) Pick $e_1 = \{v, w_1\}$ and $e_2 = \{v, w_2\}$ randomly from $E(v)$.

(3) Pick $x_1, x_2 \in \{-1, 1\}^L$ such that each bit of x_j is picked independently with expected value ξ and that the i:th bits of x_1 and x_2 are $(-1 + 2|\xi|)$-correlated (see Table 2).

(4) For $i = 1, 2$, let $b_i = f_{w_i}(\sigma^v_{e_i} x_i)$.

(5) With probability $\frac{1+\Delta}{2}$, accept iff $b_1 \vee b_2$.

(6) Otherwise, i.e. with probability $\frac{1-\Delta}{2}$, accept iff $-b_1 \vee -b_2$.

We now analyze the properties of $\mathcal{V}$. Arithmetizing the acceptance predicate of $\mathcal{V}$, we get that the probability that $\mathcal{V}$ accepts a proof is

$$\mathop{\mathbb{E}}_{v,e_1,e_2,x_1,x_2}\left[\frac{3 - \Delta(f_{w_1}(\sigma^v_{e_1} x_1) + f_{w_2}(\sigma^v_{e_2} x_2)) - f_{w_1}(\sigma^v_{e_1} x_1) f_{w_2}(\sigma^v_{e_2} x_2)}{4}\right], \tag{34}$$

where v, e_1, e_2, x_1, x_2 are picked with the same distribution as they are picked by the verifier.

4.1 Completeness

The completeness of $\mathcal{V}$ is as follows:

Lemma 4.1 (Completeness). *If $\mathrm{Val}(X) \geq 1 - \eta$, then there is a proof Σ that makes $\mathcal{V}$ accept with probability at least*

$$(1 - 2\eta)\frac{2 - \Delta\xi - |\xi|}{2} \tag{35}$$

Proof. Suppose there is an assignment of labels to the vertices of X such that the fraction of satisfied edges is at least $1 - \eta$. Fix such a labelling, and let $f_v : \{-1, 1\}^L \to \{-1, 1\}$ be the long code of the label of v. Note that for a satisfied edge $e = \{v, w\}$, $f_w(\sigma^v_e x_i)$ equals the value of the l_v:th bit of x_i (where l_v is the label of vertex v)

By the union bound, the probability that any of the two edges e_1 and e_2 are not satisfied is at most 2η. For a choice of edges e_1, e_2 that *are* satisfied, the expected value of $f_{w_i}(\sigma^v_{e_i} x_i)$ is simply the expected value of the l_v:th bit in x_i, i.e. ξ, and the expected value of $f_{w_1}(\sigma^v_{e_1} x_1) f_{w_2}(\sigma^v_{e_2} x_2)$ is the expected value of the l_v:th bit of $x_1 x_2$, i.e. $-1+2|\xi|$. Thus, for such a choice of edges, the acceptance probability becomes

$$\frac{3 - 2\Delta\xi - (-1 + 2|\xi|)}{4} = \frac{2 - \Delta\xi - |\xi|}{2} \tag{36}$$

□

4.2 Soundness

The soundness of $\mathcal{V}$ is as follows:

Lemma 4.2 (Soundness). *For any $\epsilon > 0$, $\eta \in (-1, 0)$ and $\Delta \in (-1, 1)$ there exists a $\gamma > 0$, such that if $\mathrm{Val}(X) \leq \gamma$, then for any proof Σ, the probability that $\mathcal{V}$ accepts is at most*

$$\max_{\mu \in [-1,1]} \frac{2 - (1 + \Delta)\mu - 2\Gamma_{\tilde{\rho}}(\mu)}{2} + \epsilon, \tag{37}$$

where $\tilde{\rho} = \frac{|\xi|-1}{|\xi|+1}$.

Proof. As is common, the proof is by contradiction. Assume that the value of X is at most $\mathrm{Val}(X) \leq \gamma$. Take any proof $\Sigma = \{f_v\}_{v \in V}$. Define $g_v(x) := \mathbb{E}_{e=\{v,w\} \in E(v)}[f_w(\sigma_e^v x)]$, and $\mu_v := \mathbb{E}_x[g_v(x)]$. Assume that the probability that the verifier accepts this proof is at least

$$\Pr[\mathcal{V} \text{ accepts } \Sigma] \geq \mathop{\mathbb{E}}_v \left[\frac{2 - (1 + \Delta)\mu_v - 2\Gamma_{\tilde{\rho}}(\mu_v)}{2} + \epsilon \right]. \tag{38}$$

We will show that in that case, it is possible to satisfy a constant (that depends only on ξ and ϵ) fraction of the edges of X. Setting γ smaller than this constant will yield the desired result.

Note that the probability distribution of x_1, x_2 is the same as that induced by first picking x_1 at random in B_q^n and then constructing x_2 from x_1 in the same way y is constructed from x in the Beckner operator $T_{\tilde{\rho}}$, for $q = \frac{1-\xi}{2}$ and $\tilde{\rho} = -\frac{1-q}{q} = \frac{|\xi|-1}{|\xi|+1}$. Thus, the expected value of $g_v(x_1)g_v(x_2)$ equals $\mathbb{S}_{\tilde{\rho}}(g_v)$. So by the definition of g_v and μ_v, we can rewrite the probability that the verifier accepts as

$$\begin{aligned} \Pr[\mathcal{V} \text{ accepts } \Sigma] &= \mathop{\mathbb{E}}_{v,x_1,x_2} \left[\frac{3 - \Delta(g_v(x_1) + g_v(x_2)) - g_v(x_1)g_v(x_2)}{4} \right] \\ &= \mathop{\mathbb{E}}_v \left[\frac{3 - 2\Delta\mu_v - \mathbb{S}_{\tilde{\rho}}(g_v)}{4} \right] \end{aligned}$$

Plugging in Equation (38), this gives

$$\begin{aligned} \mathop{\mathbb{E}}_v \left[\frac{3 - 2\Delta\mu_v - \mathbb{S}_{\tilde{\rho}}(g_v)}{4} \right] &\geq \mathop{\mathbb{E}}_v \left[\frac{2 - (1 + \Delta)\mu_v - 2\Gamma_{\tilde{\rho}}(\mu_v)}{2} + \epsilon \right] \\ \mathop{\mathbb{E}}_v \left[4\Gamma_{\tilde{\rho}}(\mu_v) + 2\mu_v - 1 - \mathbb{S}_{\tilde{\rho}}(g_v) \right] &\geq 4\epsilon \end{aligned}$$

Note that $4\Gamma_{\tilde{\rho}}(\mu_v) + 2\mu_v - 1 - \mathbb{S}_{\tilde{\rho}}(g_v) = 2(\Gamma_{\tilde{\rho}}(\mu_v) + \Gamma_{\tilde{\rho}}(-\mu_v)) - 1 - \mathbb{S}_{\tilde{\rho}}(g_v) \leq 2 - 1 - (-1) = 2$, so it must be the case that for a fraction of at least $\frac{3\epsilon}{2-\epsilon} \geq \epsilon$ of the vertices $v \in V$, we have

$$\mathbb{S}_{\tilde{\rho}}(g_v) \leq 4\Gamma_{\tilde{\rho}}(\mu_v) + 2\mu_v - 1 - \epsilon. \tag{39}$$

Let V_{good} be the set of all such v. Since $\tilde{\rho} < 0$ we have by (extended) Majority Is Stablest (Corollary 2.11) that for all $v \in V_{\text{good}}$ there must be some $i \in [L]$ such that $\mathrm{Inf}_i^{\le k}(g_v) \ge \tau$, where τ and k are constants depending only on ϵ and ξ.[4] Thus, for any $v \in V_{\text{good}}$, we have

$$\tau \le \mathrm{Inf}_i^{\le k}(g_v) = \mathop{\mathbb{E}}_{e=\{v,w\}}[\mathrm{Inf}_{\sigma_e^v(i)}^{\le k}(f_w)], \tag{40}$$

This, and the fact that $\mathrm{Inf}_{\sigma_e^v(i)}^{\le k}(f_w) \le 1$ for all i, implies that for a fraction of at least $\frac{\tau - \tau/2}{1-\tau/2} \ge \frac{\tau}{2}$ of the edges $e = \{v, w\} \in E(v)$, we have $\mathrm{Inf}_{\sigma_e^v(i)}^{\le k}(f_w) \ge \tau/2$.

For $v \in V$, let

$$C(v) = \{\, i \in L \mid \mathrm{Inf}_i^{\le k}(f_v) \ge \tau/2 \vee \mathrm{Inf}_i^{\le k}(g_v) \ge \tau \,\}. \tag{41}$$

Intuitively, the criterion $\mathrm{Inf}_i^{\le k}(f_v) \ge \tau/2$ means that the purported Long Codes of the label of v suggests i as a suitable label for v, and the criterion $\mathrm{Inf}_i^{\le k}(g_v) \ge \tau$ means that many of the purported Long Codes for the neighbours of v suggests that v should have the label i. By the fact that $\sum_i \mathrm{Inf}_i^{\le k}(f_w) \le k$, we must have $|C(v)| \le 2k/\tau + k/\tau = 3k/\tau$.

We now define a labelling by picking independently for each $v \in V$ a (uniformly) random label $i \in C(v)$ (or an arbitrary label in case $C(v)$ is empty). For a label $v \in V_{\text{good}}$ with $\mathrm{Inf}_i^{\le k}(g_v) \ge \tau$, the probability that v is assigned label i is $1/|C(v)| \ge \tau/3k$. Furthermore, by the above reasoning and the definition of C, at least a fraction $\tau/2$ of the edges $e = \{v, w\}$ from v will satisfy $\sigma_e^v(i) \in C(w)$. For such an edge, the probability that w is assigned the label $\sigma_e^v(i)$ is $1/|C(w)| \ge \tau/3k$. Thus, the expected fraction of satisfied edges adjacent to any $v \in V_{\text{good}}$ is at least $\tau/2 \cdot (\tau/3k)^2$, and so the expected fraction of satisfied edges in total[5] is at least $\epsilon \cdot \frac{\tau^3}{18k^2}$ (note that this is a positive constant that depends only on ϵ and ξ) and thus there is an assignment satisfying at least this total weight of edges. Making sure that $\gamma < \frac{\epsilon\tau^3}{18k^2}$, we get a contradiction on the assumption of the acceptance probability (Equation (38)), implying that the soundness is at most

$$\begin{aligned} \Pr[\mathcal{V} \text{ accepts } \Sigma] &\le \mathop{\mathbb{E}}_{v}\left[\frac{2 - (1+\Delta)\mu_v - 2\Gamma_{\tilde{\rho}}(\mu_v)}{2} + \epsilon\right] \\ &\le \max_{\mu \in [-1,1]} \frac{2 - (1+\Delta)\mu - 2\Gamma_{\tilde{\rho}}(\mu)}{2} + \epsilon, \end{aligned} \tag{42}$$

as desired. □

[4]The dependency on ξ stems from the fact that g_v is a function from B_q^n to $\mathbb{R}$, where $q = \frac{1-\xi}{2}$.

[5]We remind the reader of the convention of Section 2.5 that the choices of random vertices and edges are according to the probability distributions induced by the weights of the edges, and so choosing a random $v \in V$ and then a random $e \in E(v)$ is equivalent to just choosing a random $e \in E$.

4.3 Wrapping it up

Combining the completeness and soundness Lemmas and picking η small enough, we get that, assuming the UGC, it is NP-hard to approximate MAX 2-SAT within a factor

$$\max_{\mu \in [-1,1]} \frac{2-(1+\Delta)\mu - 2\Gamma_{\tilde{\rho}}(\mu)}{2-\Delta\xi - |\xi|} + \mathcal{O}(\epsilon)\,. \tag{43}$$

Setting ξ and Δ to the values given by Proposition 3.1, this equals $\alpha^-_{LLZ} + \mathcal{O}(\epsilon)$, proving Theorem 1.1. Again, we would like to emphasize the values given by Proposition 3.1. The large value of $\Delta \approx 0.3673$ in particular is interesting, since the weights on positive and negative occurences of variables are $\frac{1+\Delta}{2}$ and $\frac{1-\Delta}{2}$, which is roughly 68% vs. 32%. We find it remarkable that so greatly imbalanced instances should be the hardest to approximate.

5 Concluding remarks

We have shown that it is hard to approximate MAX 2-SAT within $\alpha^-_{LLZ}+\epsilon$, where $\alpha^-_{LLZ} \approx 0.94017$ is the guaranteed performance ratio of the LLZ algorithm on vector configurations v_0, v_i, v_j such that $v_i \cdot v_0 = v_j \cdot v_0$ and such that one of the two triangle inequalities $v_i \cdot v_0 + v_j \cdot v_0 + v_i \cdot v_j \geq -1$ and $-v_i \cdot v_0 - v_j \cdot v_0 + v_i \cdot v_j \geq -1$ holds with equality.

5.1 Open problems and further work

Beside the obvious importance of resolving the Unique Games Conjecture, there are a few other, quite possibly easier, questions that would be nice to settle.

- Given the result in this paper and previous works on integrality gap for e.g. MAX CUT [16], it seems likely that we should be able to show a matching integrality gap for the SDP relaxation of MAX 2-SAT (since otherwise, the UGC would be false, and it seems unlikely that a careful analysis of the MAX 2-SAT SDP should be enough to disprove the conjecture), but so far, our attempts at showing this has been elusive.
- It would be nice to have a proof that there are worst configurations for the LLZ algorithm that are simple, i.e. that the performance ratio is indeed α^-_{LLZ}.
- Given that progress for MAX 2-SAT has tended to go hand in hand with progress for MAX DI-CUT, it seems natural to conjecture that imbalanced instances should be harder also for the latter problem. However, our reduction relies heavily on the special structure of the worst-case configuration for MAX 2-SAT, in particular, that $\xi_i = \xi_j$ (this causes the PCP verifier's queries to be random variables from the same distribution, something that is critical for the soundness analysis). This is not the case for the worst-case configurations for Lewin et al.'s MAX DI-CUT algorithm—indeed, for configurations where $\xi_i = \xi_j$ the linear terms cancel out and the problem

behaves as its balanced version. Thus, it seems some (perhaps small) new idea would be needed for the PCP verifier in order to show improved hardness for MAX DI-CUT.

- It would be interesting to determine how the hardness of approximating MAX 2-SAT depends on the imbalance of the instances considered (for a suitable definition of imbalance for general instances and not just instances consisting only of Δ-mixed clauses). For instance, how large can we make the imbalance and still have instances that are hard to approximate within, say, 0.95?

5.2 Acknowledgements

I would like to thank Johan Håstad for suggesting that I work with the problem in the first place, as well as for his patience and many insightful comments along the way. I would also like to thank Uri Zwick for discussing the LLZ algorithm with me.

References

[1] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof Verification and the Hardness of Approximation Problems. *Journal of the ACM*, 45(3):501–555, 1998.

[2] Sanjeev Arora and Shmuel Safra. Probabilistic Checking of Proofs: A New Characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[3] Moses Charikar and Anthony Wirth. Maximizing Quadratic Programs: Extending Grothendieck's Inequality. In *FOCS 2004*, pages 54–60, 2004.

[4] Pierluigi Crescenzi, Riccardo Silvestri, and Luca Trevisan. On Weighted vs Unweighted Versions of Combinatorial Optimization Problems. *Information and Computation*, 167(1):10–26, 2001.

[5] Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring. In *STOC 2006*, pages 344–353, 2006.

[6] Uriel Feige. A threshold of ln *n* for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[7] Uriel Feige and Michel Goemans. Aproximating the Value of Two Prover Proof Systems, With Applications to MAX 2SAT and MAX DICUT. In *ISTCS 1995*, pages 182–189, 1995.

[8] Michel X. Goemans and David P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *Journal of the ACM*, 42:1115–1145, 1995.

[9] Eran Halperin, Ram Nathaniel, and Uri Zwick. Coloring k-colorable graphs using smaller palettes. In *SODA 2001*, pages 319–326, 2001.

[10] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 48(4):105–142, 1999.

[11] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[12] David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM*, 45(2):246–265, 1998.

[13] Subhash Khot. On the power of unique 2-prover 1-round games. In *STOC 2002*, pages 767–775, 2002.

[14] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal Inapproximability Results for Max-Cut and Other 2-Variable CSPs? In *FOCS 2004*, pages 146–154. IEEE Computer Society, 2004.

[15] Subhash Khot and Oded Regev. Vertex Cover Might be Hard to Approximate to within $2-\epsilon$. In *IEEE Conference on Computational Complexity*, pages 379–. IEEE Computer Society, 2003.

[16] Subhash Khot and Nisheeth K. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into l_1. In *FOCS 2005*, pages 53–62, 2005.

[17] Michael Lewin, Dror Livnat, and Uri Zwick. Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems. In *IPCO 2002*, volume 2337 of *Lecture Notes in Computer Science*, pages 67–82, 2002.

[18] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. Preprint, 2005.

[19] Alex Samorodnitsky and Luca Trevisan. Gowers uniformity, influence of variables, and PCPs. In *STOC 2006*, pages 11–20, 2006.

[20] Uri Zwick. Personal communication, 2005.

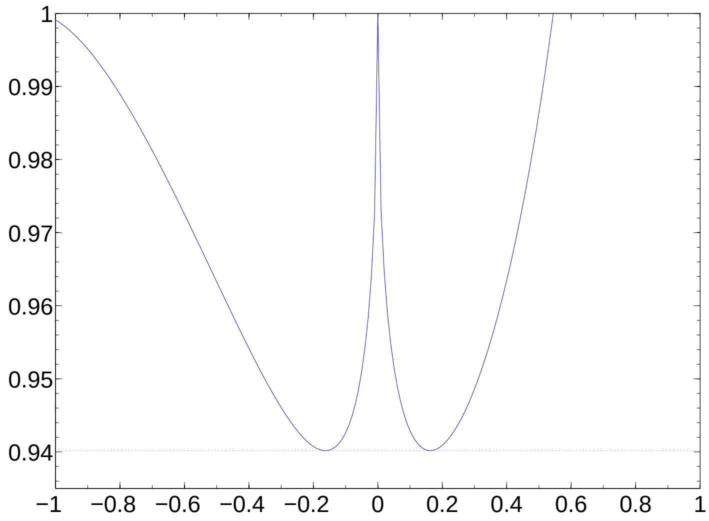

Figure 1: $\alpha^-_{0.94016567248}(\xi)$

A Analysis of α^-_{LLZ}

In this section we show some properties of α^-_{LLZ}.

A.1 The numeric value of α^-_{LLZ}

First, we will (very briefly) discuss the actual numeric value of α^-_{LLZ}. Let $B = 0.9401656724$. To give a feel for $\alpha^-_B(\xi)$, Figure 1 gives a plot of this function in the interval $\xi \in [-1, 1]$, along with the line $y = B$ (dashed). The one-dimensional optimization problem

$$\min_{\xi} \alpha_B(\xi) \tag{44}$$

can be solved numerically to a high level of precision. This gives a lower bound $\alpha^-_{LLZ} \geq 0.9401656724$. The two minima seen in Figure 1 turn out to be roughly $\xi_1 = -0.1624783294$ and $\xi_2 = 0.1624783251$. In order to obtain an upper bound on α^-_{LLZ}, we can then solve the one-dimensional optimization problem

$$\max_{\beta} \min \left(\alpha^-_{\beta}(\xi_1), \alpha^-_{\beta}(\xi_2) \right) \tag{45}$$

numerically to a high level of precision. This results in an upper bound of $\alpha^-_{LLZ} \leq 0.9401656725$. In conclusion, we have $|\alpha^-_{LLZ} - 0.94016567245| \leq 5 \cdot 10^{-11}$.

Note that our worst configurations $\xi \approx \pm 0.1625$ differs slightly from the worst configurations $\xi \approx \pm 0.169$ found by Lewin et al.. This is because of the small difference in behaviour of the two rounding functions (see Appendix B); the approximation ratio is marginally worse when using the original function of [17] rather than the one used in this paper [20].

A.2 The optimal choice of β

Proposition A.1. $\alpha^-(\alpha^-_{LLZ}) = \alpha^-_{LLZ}$. *In other words, the function $\alpha^-(\beta)$ is maximized by setting $\beta = \alpha^-_{LLZ}$.*

Furthermore, if ξ satisfies $\alpha^-_{\alpha^-_{LLZ}}(\xi) = \alpha^-_{LLZ}$ then so does $-\xi$. In other words, if ξ is a worst-case configuration for $\beta = \alpha^-_{LLZ}$, then so is $-\xi$.

Proof. Define

$$\begin{aligned} \mathrm{gain}(\beta, \xi) &= (1 - \beta\xi - \Gamma_{\tilde{\rho}}(\beta\xi)) - \alpha^-_{LLZ} \cdot \left(1 - \frac{\xi}{2} - \frac{|\xi|}{2}\right) && (46) \\ &= \frac{\alpha^-_{LLZ} - \beta}{2}\xi + \frac{2 - \Gamma_{\tilde{\rho}}(\beta\xi) - \Gamma_{\tilde{\rho}}(-\beta\xi)}{2} - \alpha^-_{LLZ}\frac{2 - |\xi|}{2} && (47) \end{aligned}$$

to be the advantage over the guaranteed approximation ratio of the LLZ algorithm when rounding the configuration $(\xi, \xi, -1 + 2|\xi|)$ using a particular value of β (where we used Proposition 2.9 to get Equation (47)). The first part of the proposition amounts to showing that

$$\mathrm{gain}(\alpha^-_{LLZ}, \xi) \geq 0 \tag{48}$$

for all $\xi \in [-1, 1]$.

Let β^* be such that $\alpha^-(\beta^*) = \alpha^-_{LLZ}$, i.e. an optimal choice of β. By definition, we have that $\mathrm{gain}(\beta^*, \xi) \geq 0$ for all simple configurations ξ. Also, from Equation (47) we see that $\mathrm{gain}(\beta, -\xi) - \mathrm{gain}(\beta, \xi) = (\beta - \alpha^-_{LLZ})\xi$ for all β and ξ. Let

$$\mathrm{gain}_\beta(\beta, \xi) = \frac{\partial\, \mathrm{gain}}{\partial \beta}(\beta, \xi) = -\xi\left(1 + \Gamma'_{\tilde{\rho}}(\beta\xi)\right) \tag{49}$$

be the derivative of gain (in the form of Equation (46)) with respect to β. Note that by Corollary D.2 we have $1 + \Gamma'_{\tilde{\rho}}(\beta\xi) \in [0, 1]$. In particular, the sign of the derivative depends only on the sign of ξ.

Consider an arbitrary configuration ξ. Let $\xi' = \xi$ if $(\beta^* - \alpha^-_{LLZ})\xi \geq 0$, and $\xi' = -\xi$ otherwise. It might help to think of ξ' the following way: if ξ has the right sign for the function $\mathrm{gain}(\cdot, \xi)$ to be increasing from β^* to α^-_{LLZ}, we are happy, but otherwise we flip ξ, thereby also flipping the sign of gain_β.

Note that since $\xi' = \pm\xi$, we have by the observation above that $\mathrm{gain}(\alpha^-_{LLZ}, \xi') = \mathrm{gain}(\alpha^-_{LLZ}, \xi)$. By the Mean Value Theorem, there is a β' between β^* and α^-_{LLZ} such that

$$\begin{aligned} \mathrm{gain}(\alpha^-_{LLZ}, \xi) &= \mathrm{gain}(\beta^*, \xi') + (\alpha^-_{LLZ} - \beta^*)\,\mathrm{gain}_\beta(\beta', \xi') \\ &\geq (\beta^* - \alpha^-_{LLZ})\xi'\left(1 + \Gamma'_{\tilde\rho}(\beta'\xi')\right) \geq 0, \end{aligned} \tag{50}$$

as desired.

The second part of the proposition follows from $\mathrm{gain}(\alpha^-_{LLZ}, \xi) = \mathrm{gain}(\alpha^-_{LLZ}, -\xi)$. In particular, if ξ is a worst case configuration, they both equal 0. $\square$

Analyzing this a bit further will (unsurprisingly) show that $\beta = \alpha^-_{LLZ}$ is indeed the *only* maximum of the function $\alpha^-(\beta)$. In order to show this, it suffices to realize that $\xi = 0$ can never be a worst-case configuration.

A.3 Proof of Proposition 3.1

In this section, we prove Proposition 3.1, which states that for an appropriately chosen configuration ξ and imbalance Δ, we can not get a better approximation ratio than α^-_{LLZ} on the configuration ξ for a Δ-mixed clause even if we change the value of β.

Proposition 3.1 restated. *There are $\xi \in (-1, 0)$ and $\Delta \in (0, 1)$ such that*

$$\alpha^-_{LLZ} = \max_{\mu \in [-1,1]} \frac{2 - (1+\Delta)\mu - 2\Gamma_{\tilde\rho}(\mu)}{2 - \Delta\xi - |\xi|}. \tag{51}$$

where $\tilde\rho = \frac{|\xi|-1}{|\xi|+1}$.

Proof. Let $\beta^* = \alpha^-_{LLZ}$, and let $\xi \approx -0.1625$ be a worst configuration, i.e. such that $\alpha^-_{\beta^*}(\xi) = \alpha^-_{LLZ}$. Note that for $\Delta \in [-1, 1]$, the quantity

$$\frac{2 - (1+\Delta)\beta^*\xi - 2\Gamma_{\tilde\rho}(\beta^*\xi)}{2 - \Delta\xi - |\xi|} \tag{52}$$

is the approximation ratio of a Δ-mixed clause on the simple configuration ξ. And since $-\xi$ is also a worst configuration (Proposition A.1), we have that Equation (52) equals α^-_{LLZ}.

Now, similarly to the proof of Proposition A.1, let

$$\mathrm{gain}(\beta) = \left(1 - \frac{1+\Delta}{2}\beta\xi - \Gamma_{\tilde\rho}(\beta\xi)\right) - \alpha^-_{LLZ} \cdot \frac{2 - \Delta\xi - |\xi|}{2} \tag{53}$$

be the advantage over the approximation ratio α^-_{LLZ} on a Δ-mixed clause when rounding the configuration ξ using a particular value of β. Again, since both ξ and $-\xi$ are worst case configurations, we have that $\mathrm{gain}(\beta^*) = 0$. We want to show that there is a suitable choice of Δ such that $\mathrm{gain}(\beta) \leq 0$ for all β. Substituting μ for $\beta\xi$ then gives the desired result. Let

$$\mathrm{gain}_\beta(\beta) = -\xi \left(\frac{1+\Delta}{2} + \Gamma'_{\tilde{\rho}}(\beta\xi) \right) \tag{54}$$

be the derivative of gain. Solving $\mathrm{gain}_\beta(\beta^*) = 0$ for Δ gives

$$\Delta = -2\Gamma'_{\tilde{\rho}}(\beta^*\xi) - 1 = 2\Phi\left(\Phi^{-1}\left(\frac{1-\beta^*\xi}{2}\right) / \sqrt{|\xi|}\right) - 1 \tag{55}$$

(see Corollary D.2 for the derivative of $\Gamma_{\tilde{\rho}}$). Note that clearly $\Delta < 1$, and since $\xi < 0$ we also have $\Delta > 0$, as promised. Furthermore, we have that the second derivative $\mathrm{gain}_{\beta\beta}(\beta) = -\xi^2\Gamma''_{\tilde{\rho}}(\beta\xi)$ which, since $\Gamma_{\tilde{\rho}}(\cdot)$ is a convex function (see Corollary D.2), implies that gain is a concave function, and thus

$$\max_\beta \mathrm{gain}(\beta) = \mathrm{gain}(\beta^*) = 0, \tag{56}$$

and we are done.

□

B The tale of the two rounding functions

The rounding function of the LLZ algorithm used in this paper differs from the rounding function used by Lewin et al. [17]. The rounding function used in this paper is $a_1(x) = \beta \cdot x$, where $\beta = \alpha^-_{LLZ} \approx 0.94016567$ (see Section 3.1 for further details). The rounding function used in [17] is $a_2(x) = 1 - 2\Phi(S(x)/\sqrt{1-x^2})$. Here, $S(x) = -2\cot(f(\arccos x))\sqrt{1-x^2}$ where f is the linear rotation function given by

$$f(\theta) \approx 0.588314458\theta + 0.64667394. \tag{57}$$

$a_2(x)$ can be simplified to

$$a_2(x) = 1 - 2\Phi(-2\cot(f(\arccos x))) = 2\Phi(2\cot(f(\arccos x))) - 1. \tag{58}$$

Figure 2 gives plots of the functions $a_1(x)$ and $a_2(x)$ for the interval $x \in [0, 1]$ (since both functions are odd we restrict our attention to positive x). As can be seen, the functions are fairly close to each other. Most importantly, the functions behave almost the same in the critical interval $x \in [0.1, 0.2]$. Nevertheless, there is a small difference between the functions in this interval as well, and as noted in Appendix A.1, this causes the worst configuration when using $a_1(x)$ to be slightly different from the worst configuration when using $a_2(x)$. This small difference in fact causes the (apparent) approximation ratio when using $a_1(x)$ to be marginally better than when using $a_2(x)$.

For large x, the functions $a_1(x)$ and $a_2(x)$ differ noticeably, but using *the* best rounding does not matter there; these are configurations that are in some sense easy to round, and any function with a reasonable behaviour suffices to get a decently good approximation ratio.

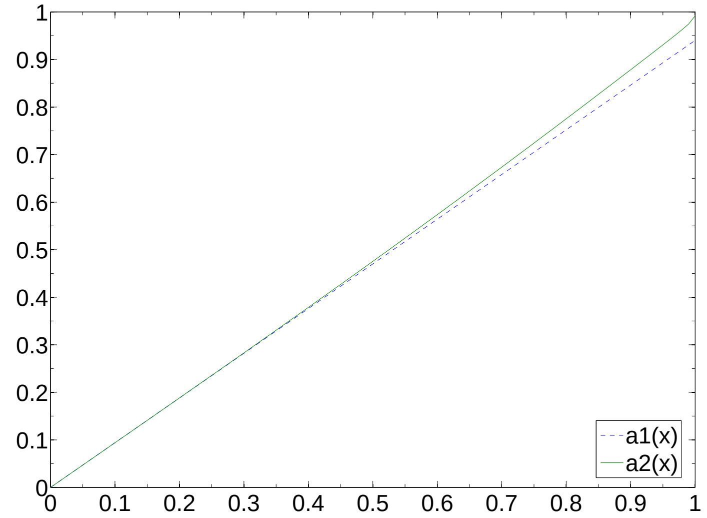

Figure 2: $a_1(x)$ vs. $a_2(x)$

C Proofs of some Fourier-analytic identities

In this section, we prove some basic Fourier-analytic identities which were given in Section 2.2.

Proposition 2.1 restated. *The set of functions $\{U_q^S\}_{S\subseteq[n]}$ forms an orthonormal basis w.r.t. the scalar product $\langle\cdot,\cdot\rangle$.*

Proof. First, note that the function U_q satisfies $\mathbb{E}[U_q(x_i)] = 0$ and $\mathbb{E}[U_q(x_i)^2] = 1$. Take any $S, T \subseteq [n]$, and denote by $m(i) = |S \cap \{i\}| + |T \cap \{i\}| \in \{0, 1, 2\}$ the number of occurences of i in the two sets. We have

$$\begin{aligned}\left\langle U_q^S, U_q^T\right\rangle &= \mathbb{E}\left[\prod_{i\in S} U_q(x_i) \prod_{i\in T} U_q(x_i)\right] \\ &= \prod_{i=1}^{n} \mathbb{E}[U_q(x_i)^{m(i)}], \end{aligned} \tag{59}$$

where the last equality uses that x_i and x_j are independent for $i \neq j$. If $S = T$ we have that $m(i) \in \{0, 2\}$ for all i and thus $\left\langle U_q^S, U_q^T\right\rangle = 1$. If $S \neq T$ there is some i such that $m(i) = 1$ and thus $\left\langle U_q^S, U_q^T\right\rangle = 0$. ☐

Proposition 2.3 restated.

$$\mathrm{Inf}_i(f) = \sum_{\substack{S\subseteq[n]\\ i\in S}} \hat{f}_S^2. \tag{60}$$

Proof. For $x \in \{-1,1\}^n$, let $x \cup i$ denote the bit string $(x_1, \dots, x_{i-1}, -1, x_{i+1}, \dots, x_n)$ (i.e. we set the i:th bit), and let $x \setminus i$ denote the bit string $(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ (i.e. we unset the i:th bit).

Split f into $f(x) = f_0(x) + f_1(x)$, where

$$\begin{aligned} f_0(x) &= \sum_{i \notin S} \hat{f}_S U_q^S(x) && (61) \\ f_1(x) &= \sum_{i \in S} \hat{f}_S U_q^S(x). && (62) \end{aligned}$$

Note that f_0 does not depend on the variable x_i. We have

$$\begin{aligned} &\mathop{\mathbb{E}}_{x_i}[f(x) \,|\, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n] \\ &\quad = q(f_0(x) + f_1(x \cup i)) + (1-q)(f_0(x) + f_1(x \setminus i)) \\ &\quad = f_0(x), && (63) \end{aligned}$$

where we use the fact that $f_1(x \cup i) = -f_1(x \setminus i)\frac{1-q}{q}$. This implies

$$\begin{aligned} &V_{x_i}[f(x) \,|\, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n] \\ &\quad = q f_1(x \cup i)^2 + (1-q) f_1(x \setminus i)^2 \\ &\quad = \mathop{\mathbb{E}}_{x_i}[f_1(x)^2 \,|\, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n]. && (64) \end{aligned}$$

Averaging over all x, we get that

$$\mathrm{Inf}_i(f) = \mathop{\mathbb{E}}_{x}[f_1(x)^2] = \langle f_1, f_1 \rangle = \sum_{i \in S} \hat{f}_S^2, \tag{65}$$

and we are done. $\square$

Proposition 2.6 restated.

$$T_\rho f(x) = \sum_{S \subseteq [n]} \rho^{|S|} \hat{f}_S U_q^S(x). \tag{66}$$

Proof. Clearly, it suffices to show that for every i, $\mathbb{E}_{y_i}[U_q(y_i)] = \rho \cdot U_q(x_i)$. If $x_i = 1$, we have

$$\begin{aligned} \mathop{\mathbb{E}}_{y_i}[U_q(y_i)] &= q(1-\rho)U_q(-1) + (1 - q(1-\rho))U_q(1) \\ &= -(1-q)(1-\rho)U_q(1) + (1 - q(1-\rho))U_q(1) = \rho U_q(1), && (67) \end{aligned}$$

where we used $U_q(-1) = -U_q(1)\frac{1-q}{q}$. The case $x_i = -1$ is handled analogously. $\square$

D Proofs of some properties of Γ_ρ

In this section we prove some useful properties about Γ_ρ. We start with the proof of Proposition 2.9, which we restate for convenience.

Proposition 2.9 restated. *For all* $\rho \in [-1,1]$, $\mu_1, \mu_2 \in [-1,1]$, *we have*

$$\Gamma_\rho(-\mu_1, -\mu_2) = \Gamma_\rho(\mu_1, \mu_2) + \mu_1/2 + \mu_2/2 \tag{68}$$

Proof. Let $t_i = \Phi^{-1}\left(\frac{1-\mu_i}{2}\right)$, and let X and Y be two ρ-correlated $N(0,1)$ variables. Clearly, $\Gamma_\rho(-\mu_1,-\mu_2) = \Pr[X \le -t_1 \wedge Y \le -t_2]$. Assume that $\mu_1 < 0, \mu_2 < 0$ (implying $t_1 > 0$ and $t_2 > 0$). We have

$$\begin{aligned}
\Gamma_\rho(\mu_1,\mu_2) - \Gamma_\rho(-\mu_1,-\mu_2) &= \Pr[X \le t_1 \wedge Y \le t_2] - \Pr[X \le -t_1 \wedge Y \le -t_2] \\
&= \Pr[X \le 0 \wedge |Y| \le t_2] + \\
&\quad \Pr[0 \le X \le t_1 \wedge -t_2 \le Y \le 0] + \\
&\quad \Pr[0 \le X \le t_1 \wedge 0 \le Y \le t_2] + \\
&\quad \Pr[|X| \le t_1 \wedge Y \le -t_2].
\end{aligned} \tag{69}$$

Note that $\Pr[0 \le X \le t_1 \wedge 0 \le Y \le t_2] = \Pr[-t_1 \le X \le 0 \wedge -t_2 \le Y \le 0]$ and that $\Pr[X \le 0 \wedge |Y| \le t_2] = \Pr[X \ge 0 \wedge |-Y| \le t_2] = \Pr[|Y| \le t_2]/2 = -\mu_2/2$. Thus,

$$\begin{aligned}
\Gamma_\rho(\mu_1,\mu_2) - \Gamma_\rho(-\mu_1,-\mu_2) &= \Pr[X \le 0 \wedge |Y| \le t_2] + \\
&\quad \Pr[|X| \le t_1 \wedge -t_2 \le Y \le 0] + \\
&\quad \Pr[|X| \le t_1 \wedge Y \le -t_2] \\
&= -\mu_1/2 - \mu_2/2,
\end{aligned} \tag{70}$$

as desired. The other three sign combinations for μ_1 and μ_2 are handled analogously. $\square$

Next, we compute the derivative of Γ_ρ. For the rest of this section, let $t(x) = \Phi^{-1}\left(\frac{1-x}{2}\right)$.

Proposition D.1. *For* $\rho \in (-1,1)$, *we have*

$$\frac{\partial \Gamma_\rho}{\partial \mu_1}(\mu_1,\mu_2) = -\frac{1}{2}\Phi\left(\frac{t(\mu_2) - \rho t(\mu_1)}{\sqrt{1-\rho^2}}\right) \tag{71}$$

Proof. This follows from the fact that $\Gamma_\rho(\mu_1,\mu_2)$ can be written as

$$\Gamma_\rho(\mu_1,\mu_2) = \int_{x=-\infty}^{t(\mu_1)} \phi(x)\Phi\left(\frac{t(\mu_2) - \rho x}{\sqrt{1-\rho^2}}\right) \mathrm{d}x, \tag{72}$$

giving

$$\frac{\partial \Gamma_\rho}{\partial \mu_1}(\mu_1, \mu_2) \quad = \quad t'(\mu_1)\phi(t(\mu_1))\Phi\left(\frac{t(\mu_2) - \rho t(\mu_1)}{\sqrt{1-\rho^2}}\right). \tag{73}$$

Using $t'(x) = -\frac{1}{2\phi(t(x))}$, the result follows. □

As a simple corollary, we get

Corollary D.2. *For* $\rho \in (-1, 1)$, *we have*

$$\frac{\partial \Gamma_\rho}{\partial \mu}(\mu) = -\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t(\mu)\right) \tag{74}$$

Note that Corollary D.2 implies that $\frac{\partial^2 \Gamma_\rho}{\partial \mu^2}(\mu) > 0$ for all μ, i.e. that Γ_ρ is a convex function.

Proof. Indeed,

$$\begin{aligned} \frac{\partial \Gamma_\rho}{\partial \mu}(\mu) \quad &= \quad \frac{\partial \Gamma_\rho}{\partial \mu_1}(\mu, \mu) + \frac{\partial \Gamma_\rho}{\partial \mu_2}(\mu, \mu) \\ &= \quad 2 \cdot \left(-\frac{1}{2}\Phi\left(\frac{(1-\rho)t(\mu)}{\sqrt{1-\rho^2}}\right)\right) \\ &= \quad -\Phi\left(\sqrt{\frac{1-\rho}{1+\rho}}t(\mu)\right). \end{aligned} \tag{75}$$

Here, we used the fact that $\Gamma_\rho(\mu_1, \mu_2) = \Gamma_\rho(\mu_2, \mu_1)$, so the derivative of Γ_ρ with respect to μ_2 can also be computed using Proposition D.1. □

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/