On the Approximation Resistance of a Random Predicate

Johan Håstad

Royal Institute of Technology,Stockholm, Sweden

Abstract. A predicate is approximation resistant if no probabilistic polynomial time approximation algorithm can do significantly better then the naive algorithm that picks an assignment uniformly at random. Assuming that the Unique Games Conjecture is true we prove that most Boolean predicates are approximation resistant.

1 Introduction

We consider constraint satisfaction problems (CSPs) over the Boolean domain. In our model a problem is defined by a k-ary predicate P and an instance is given by a list of k-tuples of literals. The task is to find an assignment to the variables such that all the k-bit strings resulting from the list of k-tuples of literals under the assignment satisfy the predicate P. In this paper we focus on Max-CSPs which are optimization problems where we try to satisfy as many constraints as possible.

The most famous such problem is probably Max-3-Sat where $k = 3$ and P is simply the disjunction of the three bits. Another problem that (almost) falls into this category is Max-Cut, in which $k = 2$ and P is non-equality. In traditional Max-Cut we do not allow negations among the literals and if we do allow negation the problem becomes Max-E2-Lin-2, linear equations modulo 2 with exactly two variables in each equation.

It is a classical result that most Boolean CSPs are NP-complete. Already in 1978 Schaefer [12] gave a complete characterization giving only 5 classes for which the problem is in P while establishing NP-completeness in the other cases.

Of course if a CSP is NP-complete, the corresponding Max-CSP is NP-hard. The converse is false and several of Schaefer's easy satisfiability problems are in fact NP-hard as optimization problems. We turn to study approximation algorithms. An algorithm is here considered to be a C-approximation if it, on each input, finds an assignment with an objective value that is within a factor C of the optimal solution. We allow randomized approximation algorithms and in such a case it is sufficient that the expected value, over the random choices of the algorithm, of the objective value satisfies the desired bound.

To define what is non-trivial is a matter of taste but hopefully there is some consensus that the following algorithm is trivial: Without looking at the instance pick a random value for each variable. We say that an approximation ratio is non-trivial if it gives a value of C that is better than the value obtained by this trivial algorithm. We call a predicate *approximation resistant* if it is NP-hard to achieve a non-trivial approximation ratio.

It is perhaps surprising but many CSPs are approximation resistant and one basic example is Max-3-Sat [6]. The famous algorithm of Goemans and Williamson [1] shows that Max-Cut is not approximation resistant and this result can be extended in great generality and no predicate that depends on two inputs from an arbitrary finite domain can be approximation resistant [7].

Zwick [14] established approximability results for predicates that depend on three Boolean inputs and from this it follows that the only predicates on three inputs that are approximation resistant are those that are implied by parity or its negation. A predicate P is implied by a predicate Q iff whenever $Q(x)$ is true so is $P(x)$ and as an example the negation of parity implies disjunction as if we know that an odd number of variables are true they cannot all be false.

Many scattered results on (families of) wider predicates do exist [4, 10] and in particular Hast [5] made an extensive classification of predicates on four inputs. Predicates that can be made equal by permuting the inputs or negating one or more inputs behave the same with respect to approximation resistance and with this notion of equivalence there are 400 different non-constant predicates on 4 Boolean inputs. Hast proved that 79 of these are approximation resistant, established 275 to be non-trivially approximable leaving the status of 46 predicates open. Zwick [13] has obtained numerical evidence suggesting that most of the latter predicates are in fact non-trivially approximable.

The main result of this paper is to give evidence that a random predicate for a large value of k is approximation resistant. The result is only evidence in that it relies on the Unique Games Conjecture (UGC) of Khot [8] but on the other hand we establish that a vast majority of the predicates are approximation resistant under this assumption.

We base our proof on the recent result by Samorodnitsky and Trevisan [11] that establishes that if d is the smallest integer such that $2^d - 1 \geq k$ then there is a predicate of width k that accepts only 2^d of the 2^k possible k-bit strings and which, based on the UGC, is approximation resistant. We extend their proof to establish that any predicate implied by their predicate is approximation resistant.

To establish our main result we proceed to prove that a random predicate is implied by some predicate which is equivalent to the predicate of Samorodnitsky and Trevisan. This is established by a second moment method. A standard random predicate on k bits is constructed by, for each of the 2^k inputs, flipping an unbiased coin to determine whether that input is accepted. It turns out that our results apply to other spaces of random predicates. In fact, if we construct a random predicate by accepting each input with probability k^{-c} for some $c > 0$ we still, with high probability for sufficiently large k, get an approximation resistant predicate. Here c is a number in the range $[1/2, 1]$ that depends on how close k is to the smallest number of the form $2^d - 1$ larger than k.

We make the proof more self contained by reproving one main technical lemma of [11] relating to Gowers uniformity norms and influences of functions.

Our proof is similar in spirit to the original proof but significantly shorter and we hence believe it is of independent interest.

Of course the contribution of this paper heavily depends on how one views the Unique Games Conjecture, UGC. At the least one can conclude that it will be difficult to give a non-trivial approximation algorithm for a random predicate. Our results also point to the ever increasing need to settle the UGC.

An outline of the paper is as follows. We start by establishing some notation and giving some definitions in Section 2. We prove the lemmas relating to Gowers uniformity in Section 3 and proceed to establish that any predicate implied by the predicate used by Samorodnitsky and Trevisan is approximation resistant in Section 4. We then present our applications of this theorem by first establishing that a random predicate is approximation resistant in Section 5 and that all very dense predicates are approximation resistant in Section 6. We end with some concluding remarks in Section 7.

2 Preliminaries

We consider functions mapping $\{-1,1\}^n$ into the real numbers and usually into the interval $[-1,1]$. In this paper we use $\{-1,1\}$ as the value set of Boolean variables but still call the values "bits". For $x, x' \in \{-1,1\}^n$ we let $x \cdot x'$ denote the coordinate-wise product. In $\{0,1\}^n$-notation this is the simply the exclusive-or of vectors.

For any $\alpha \subseteq [n]$ we have the character χ_α defined by

$$\chi_\alpha(x) = \prod_{i \in \alpha} x_i$$

and the Fourier expansion is given by

$$f(x) = \sum_{\alpha \subseteq [n]} \hat{f}_\alpha \chi_\alpha(x).$$

We are interested in long codes coding $v \in [L]$. This is a function $\{-1,1\}^L \rightarrow \{-1,1\}$ and if A is the long code of v then $A(x) = x_v$. We want our long codes to be folded, which means that they only contain values for inputs with $x_0 = 1$. The value when $x_0 = -1$ is defined to be $-A(-x)$. This ensures that the function is unbiased and that the Fourier coefficient corresponding to the empty set is 0.

For two sets α and β we let $\alpha \Delta \beta$ be the symmetric difference of the two sets.

The influence $\text{inf}_i f$ is the expected variance of f when all variables except x_i are picked randomly and uniformly. It is well known that

$$\text{inf}_i = \sum_{i \in \alpha} \hat{f}_\alpha^2.$$

The following lemma from [10] is useful.

Lemma 2.1. *Let $(f_j)_{j=1}^k$, $\{-1,1\}^n \to [-1,1]$ be k functions, and*

$$f(x) = \prod_{j=1}^{k} f_j(x).$$

Then, for every $i \in [n]$, $\inf_i(f) \leq k \sum_{j=1}^k \inf_i(f_j)$.

The pairwise cross influence of a set of functions $(f_j)_{j=1}^k$ is defined to be the maximal simultaneous influence in any two of the functions or more formally

$$\mathrm{cinf}_i(f_j)_{j=1}^k = \max_{j_1 \neq j_2} \min(\inf_i(f_{j_1}), \inf_i(f_{j_2})).$$

Let P be a predicate on k Boolean inputs. An instance of the problem Max-CSP(P) is given by a list of k-tuples of literals. The task is to find the assignment to the variables that maximizes the number of k-tuples that satisfy P.

An algorithm is a C-approximation if it, for any instance I of this problem, produces an assignment which satisfies at least $C \cdot Opt(I)$ constraints where $Opt(I)$ is the number of constraints satisfied by an optimal solution.

Let $d(P)$ be the fraction of k-bit strings accepted by P. The trivial algorithm that just picks a random assignment satisfies, on the average, a $d(P)$-fraction of the constraints and as an optimal solution cannot satisfy more than all the constraints this yields a (randomized) $d(P)$-approximation algorithm. We have the following definition.

Definition 2.1. *A predicate P is* approximation resistant *if, for any $\epsilon > 0$, it is NP-hard to approximate Max-CSP(P) within $d(P) + \epsilon$.*

Some predicates have an even stronger property.

Definition 2.2. *A predicate P is* hereditary approximation resistant *if any predicate Q implied by P is approximation resistant.*

3 Gowers Uniformity and Influence

Gowers [2, 3] introduced the notion of dimension-d uniformity norm $U^d(f)$ which was used in an essential way by Samorodnitsky and Trevisan [11]. Their result says that if a function does not have an influential variable and is unbiased then the dimension-d uniformity norm is small. More importantly for their application, [11] also proved that if a set of functions has small cross influences and at least one function is unbiased then the corresponding product is small. We slightly extend their result by allowing a small bias of the involved functions. Allowing this extension makes it possible to give a short, direct proof.

We want to emphasize that the results obtained by Samorodnitsky and Trevisan are sufficient for us but we include the results of this section since we believe that our proofs are simpler and that the extension might be interesting on its own and possibly useful in some other context.

Theorem 3.1. *Let $f\colon \{-1,1\}^n \to [-1,1]$ be a function with $\max_i \inf_i(f) \le \epsilon$ and $|E[f]| \le \delta$, then*

$$\left| E_{x^1,\ldots x^d} \left[\prod_{S\subseteq[d]} f\left(\prod_{i\in S} x^i\right)\right]\right| \le \delta + (2^{d-1}-1)\sqrt{\epsilon}.$$

Proof. We prove the theorem by induction over d. Clearly it is true for $d=1$ as the quantity to estimate equals $|f(1^n)E[f]|$.

For the induction step let $g^{x^d}(x) = f(x)f(x\cdot x^d)$. Then, by Lemma 2.1, $\max_i \inf_i g^{x^d} \le 4\epsilon$. Furthermore

$$E_x[g^{x^d}] = 2^{-n}\sum_x f(x)f(x\cdot x^d) = f * f(x^d)$$

and let us for notational simplicity denote this function by $h(x^d)$. As convolution turns into product on the Fourier transform side we have $\hat{h}_\alpha = \hat{f}_\alpha^2$. For any $\alpha \neq \emptyset$ we have $\hat{f}_\alpha^2 \le \max_i \inf_i(f) \le \epsilon$ and hence

$$\|h\|_2^2 = \sum_\alpha \hat{h}_\alpha^2 = \sum_\alpha \hat{f}_\alpha^4 \le \hat{f}_\emptyset^4 + \epsilon \sum_{\alpha\neq\emptyset} \hat{f}_\alpha^2 \le \delta^4 + \epsilon.$$

This implies, using the Cauchy-Schwartz inequality, that

$$E_{x^d}[|E_x[g^{x^d}(x)]|] \le \sqrt{\delta^4+\epsilon} \le \delta^2 + \sqrt{\epsilon} \le \delta + \sqrt{\epsilon}. \qquad (1)$$

Now

$$\left| E_{x^1,\ldots x^d} \left[\prod_{S\subseteq[d]} f\left(\prod_{i\in S} x^i\right)\right]\right| \le E_{x^d} \left| E_{x^1,\ldots x^{d-1}} \left[\prod_{S\subseteq[d-1]} g^{x^d}\left(\prod_{i\in S} x^i\right)\right]\right|,$$

which, by induction, is bounded by

$$E_{x^d}\left[|E_x[g^{x^d}]| + (2^{d-2}-1)\sqrt{4\epsilon}\right] \le \delta + (2^{d-1}-1)\sqrt{\epsilon}.$$

Note that by doing some more calculations we can get a better bound as a function of δ by not doing the wasteful replacement of δ^2 by δ in (1). We proceed to allow the functions to be different and require the pairwise cross influence to be small.

Theorem 3.2. *Let $(f_S)_{S\subseteq[d]}$ be a set of functions $\{-1,1\}^n \to [-1,1]$, with $\max_i \mathrm{cinf}_i(f_S) \le \epsilon$ and $\min_{S\neq\emptyset} |E[f_S]| \le \delta$, then*

$$\left| E_{x^1,\ldots x^d} \left[\prod_{S\subseteq[d]} f_S(\prod_{i\in S} x^i)\right]\right| \le \delta + (2^d-2)\sqrt{\epsilon}.$$

Proof. We use induction over d. The base case $d = 1$ is straightforward and let us do the induction step.

By a change of variables we can assume that $|E[f_{[d]}]| \le \delta$. Now define a new set of functions by

$$g_S^{x^d}(x) = f_S(x) f_{S\cup\{d\}}(x \cdot x^d),$$

for any $S \subseteq [d-1]$. The cross influence of this set of functions is, by Lemma 2.1, bounded by 4ϵ. Let $h(x^d)$ be the average of $g^{x^d}_{[d-1]}$. Then $h = f_{[d-1]} * f_{[d]}$ and $\hat{h}_\alpha = \hat{f}_{[d-1],\alpha}\hat{f}_{[d],\alpha}$ which yields

$$\|h\|_2^2 = \sum_\alpha \hat{h}_\alpha^2 = \hat{f}^2_{[d-1],\emptyset}\hat{f}^2_{[d],\emptyset} + \sum_{\alpha\neq\emptyset} \hat{f}^2_{[d-1],\alpha}\hat{f}^2_{[d],\alpha} \le$$

$$\delta^2 + \sum_{\alpha\neq\emptyset} \min(\hat{f}^2_{[d-1],\alpha}, \hat{f}^2_{[d],\alpha})(\hat{f}^2_{[d-1],\alpha} + \hat{f}^2_{[d],\alpha}) \le \delta^2 + \sum_{\alpha\neq\emptyset} \epsilon(\hat{f}^2_{[d-1],\alpha} + \hat{f}^2_{[d],\alpha}) \le \delta^2 + 2\epsilon.$$

Using induction we get

$$\left| E_{x^1,\ldots x^d}\left[\prod_{S\subseteq[d]} f_S\left(\prod_{i\in S} x^i\right)\right]\right| \le E_{x^d}\left| E_{x^1,\ldots x^{d-1}}\left[\prod_{S\subseteq[d-1]} g_S\left(\prod_{i\in S} x^i\right)\right]\right| \le$$

$$E_{x^d}\left[|E[g^{x^d}_{[d-1]}]| + (2^{d-1}-2)\sqrt{4\epsilon}\right] \le \delta + 2\sqrt{\epsilon} + (2^d - 4)\sqrt{\epsilon} \le \delta + (2^d-2)\sqrt{\epsilon}.$$

4 The ST-predicate

Assume that $2^{d-1} \le k \le 2^d - 1$. For any integer i with $1 \le i \le 2^d - 1$ let $\hat{i} \subseteq [d]$ be the set whose characteristic vector is equal to the binary expansion of i. We define $P_{ST}(x)$, a predicate on k-bit strings, to be true if for all triplets i_1, i_2, and i_3 such that $\hat{i}_1 \Delta \hat{i}_2 = \hat{i}_3$ we have $x_{i_1}x_{i_2} = x_{i_3}$. Of course the predicate depends on k but as k (and d) remains fixed we suppress this dependence.

It is not difficult to see that the accepted strings form a linear space of dimension d. In fact the following procedure for picking a random string accepted by P_{ST} is a good way to visualize the predicate. For each i that is a power of two set x_i to a random bit. For other values of i set

$$x_i = \prod_{j\in\hat{i}} x_{2^j}.$$

Now consider Max-CSP(P_{ST}) and the following theorem is from [11].

Theorem 4.1. *Assuming the UGC, for any $\epsilon > 0$, it is NP-hard to approximate Max-CSP(P_{ST}) within $2^{d-k} + \epsilon$.*

Equivalently, the theorem says that P_{ST}, assuming UGC, is approximation resistant, but we need more.

Theorem 4.2. *Assuming UGC, P_{ST} is hereditary approximation resistant.*

It is satisfying to note that for $k = 3$ the predicate P_{ST} is simply parity and hence this instance of the theorem was proved in [6] without using the UGC.

Proof. Let Q be any predicate of arity k implied by P_{ST}. Our proof is very similar to the proof of [11] but we use a slightly different terminology. We assume that the reader is familiar with Probabilistically Checkable Proofs (PCPs) and their relation to inapproximability result for Max-CSPs. Details of the connection can be found in many places, one possible place being [6]. The short summary is that for any $\gamma > 0$ we need to define a PCP where the acceptance condition is given by the predicate Q and such that it is hard to distinguish the case when the maximal acceptance probability is $1 - \gamma$ and the case when the maximal acceptance probability is $d(Q)+\gamma$. It is also needed that the verifier uses $O(\log n)$ random bits when checking proofs of statements of size n. The latter property implies that the proof is of polynomial size.

As in [11] we use a form of the UGC which, using the terminology of [11], is called the k-ary unique games. We have variables $(v_i)_{i=1}^n$ taking values in a finite domain of size L, which we assume to be $[L]$. A constraint is given by a k-tuple, $(v_{i_j})_{j=1}^k$ of variables and k permutations $(\pi_j)_{j=1}^k$. An assignment V strongly satisfies the constraint iff the k elements $\pi_j(V(v_{i_j}))$ are all the same and the assignment weakly satisfies the constraint if these values are not all distinct. The following result, originally by Khot and Regev [9] is stated in [11].

Theorem 4.3. *If the UGC is true then for every k and ϵ there is a $L = L(k, \epsilon)$ such that, given a k-ary unique game problem with alphabet size L, it is NP-hard to distinguish the case in which there is an assignment that strongly satisfies at least a $(1 - \epsilon)$-fraction of the constraints from the case where every assignment weakly satisfies at most a fraction ϵ of the constraints.*

We proceed to construct a PCP based on the k-ary unique game problem. The test is as described in [11] but slightly reformulated.

The written proof is supposed to be coding of an assignment which satisfies a $(1 - \epsilon)$-fraction of the constraints. For each v_i the proof contains the long code A_i of $V(v_i)$. We access these long codes in a folded way as described in the preliminaries. This folding gives rise to negations in the resulting instance of Max-CSP(Q). We let permutations act on vectors by $\pi(x)_j = x_{\pi(j)}$.

As in many PCPs we use noise vectors $\mu \in \{-1, 1\}^L$ which has the property that μ_v is picked randomly and independently and for each $v \in [L]$ it equals 1 with probability $1 - \delta$ and -1 with probability δ, where δ is a parameter to be determined. It is an important parameter of the test and hence we include it explicitly. The verifier of the PCP works as follows.

Q-test(δ).

1. Pick a random k-ary constraint, given by variables $(v_{i_j})_{j=1}^k$, and permutations $(\pi_j)_{j=1}^k$.

2. Pick d independent random unbiased unbiased $x^i \in \{-1,1\}^L$ and k independent noise functions $\mu^j \in \{-1,1\}^L$.
3. Let $y^j = \prod_{i \in \hat{j}} x^i$ and $b_j = A_{i_j}(\pi_j(y^j) \cdot \mu^j)$.
4. Accept if $Q(b) = Q(b_1, b_2, \ldots b_k)$ is true.

We first address completeness.

Lemma 4.1. *For any $\gamma > 0$ there exists $\delta > 0$, $\epsilon > 0$ such that if there is an assignment that strongly satisfies a fraction $1-\epsilon$ of the constraints in the k-ary unique game problem then the verifier in Q-test(δ) can be made to accept with probability $1-\gamma$.*

Proof. Assume that each A_j is the correct long code of the value $V(v_j)$ for an assignment V that satisfies at least a $(1-\epsilon)$-fraction of the constraints. Then assuming that $\mu^j_{V(v_{i_j})} = 1$ and $\pi_j(V(v_{i_j})) = v$ for all j we have

$$b_j = y^j_{\pi_j(V(v_{i_j}))} \cdot \mu^j_{V(v_{i_j})} = y^j_v = \prod_{i \in \hat{j}} x^i_v.$$

Recalling the description of the accepted inputs of P_{ST} it follows that b satisfies P_{ST} and hence also Q. The completeness is hence at least $1-\epsilon-k\delta$ and choosing ϵ and δ sufficiently small this is at least $1-\gamma$.

Let us turn to the more challenging task of analyzing the soundness.

Lemma 4.2. *For any $\gamma > 0$, $\delta > 0$ there exist $\epsilon = \epsilon(k, \delta, \gamma) > 0$ such that if the verifier in Q-test(δ) accepts with probability at least $d(Q) + \gamma$ there exists an assignment that weakly satisfies at least a fraction ϵ of the constraints in the k-ary unique game problem.*

Proof. We assume that the verifier accepts with probability $d(Q) + \gamma$ and turn to define a (randomized) assignment that weakly satisfies a fraction of the constraints that only depends on k, δ and γ.

We use the multilinear representation of Q (which is in fact identical to the Fourier transform)

$$Q(b) = \sum_\beta \hat{Q}_\beta \prod_{j \in \beta} b_j.$$

Note that the constant term $\hat{Q}_\emptyset$ is exactly $d(Q)$ and hence if the verifier accepts with probability $d(Q) + \gamma$ there must be some nonempty β such that

$$|E[\prod_{j \in \beta} b_j]| \geq 2^{-k}\gamma, \tag{2}$$

where the expectation is taken over a random constraint of the k-ary unique game and random choices of x^i and μ^j.

Let us first study expectation over the noise vectors and towards this end let

$$B_j(y) = E_\mu[A_j(y \cdot \mu)],$$

which gives $E_{\mu^j}(b_j) = B_{i_j}(\pi_j(y^j))$. It is a standard fact (for a proof see [6]) that

$$\hat{B}_{j,\beta} = (1-2\delta)^{|\beta|}\hat{A}_{j,\beta}$$

and hence

$$\sum_{|\beta| \geq t} \hat{B}^2_{j,\beta} \leq (1-2\delta)^{2t} \tag{3}$$

for any t. Now set $\mathit{\Gamma} = 2^{-2(d+k+2)}\gamma^2$ and let $t = O(\delta^{-1} \log \mathit{\Gamma}^{-1})$ be such that

$$(1-2\delta)^{2t} \leq \mathit{\Gamma}/2,$$

and define

$$T_j = \{i \mid \mathrm{inf}_i B_j \geq \mathit{\Gamma}\}.$$

As

$$\mathrm{inf}_i B_j = \sum_{i \in \beta} \hat{B}^2_{j,\beta}, \tag{4}$$

by (3) and the definition of t, if $i \in T_j$ then we must have at least a contribution of $\mathit{\Gamma}/2$ from sets of size at most t in (4). Using this it follows that $|T_j| \leq 2t/\mathit{\Gamma}$ for any j.

Consider the probabilistic assignment that for each v_j chooses a random element of T_j. If T_j is empty we choose an arbitrary value for v_j.

By (2) we know that for at least a fraction $2^{-k}\gamma/2$ of the constraints we have

$$\left| E_{x^i,\mu^j}\left[\prod_{j\in\beta} b_j\right]\right| \geq 2^{-k}\gamma/2. \tag{5}$$

Fix any such constraint and define the following family of functions.

For any $j \notin \beta$ or $k < j \leq 2^d - 1$ set $h_{\hat{j}}$ to be identically one while if $j \in \beta$ we define $h_{\hat{j}}$ by

$$h_{\hat{j}}(y) = B_{i_j}(\pi_j(y)).$$

These definitions imply that

$$E_\mu\left[\prod_{j\in\beta} b_i\right] = \prod_{S\subseteq[d]} h_S\left(\prod_{i\in S} x^i\right) \tag{6}$$

and hence we are in a position to apply Theorem 3.2. Note first that, by folding, each h that is non-constant is in fact unbiased and hence, as β is non-empty, the minimum bias of the set of functions is 0.

We now claim that the maximal cross influence of the function set h_S is at least Γ. Indeed suppose that this is not the case. Then, by Theorem 3.2, the expectation of (6), over the choice of vectors x^i, is at most

$$(2^d - 2)\sqrt{\Gamma} < 2^d 2^{-(d+k+2)}\gamma \leq 2^{-k}\gamma/2$$

contradicting (5).

Thus we have $j_1, j_2 \in \beta$ and an i such that $\mathrm{inf}_i h_{\hat{j}_1} \geq \Gamma$ and $\mathrm{inf}_i h_{\hat{j}_2} \geq \Gamma$. Now, by definition, $\mathrm{inf}_i h_{\hat{j}_1}$ is the same as $\mathrm{inf}_{\pi^{-1}_{j_1}(i)}(B_{i_{j_1}})$. We conclude that there is a common element in $\pi_{j_1}(T_{i_{j_1}})$ and $\pi_{j_2}(T_{i_{j_2}})$ and our probabilistic assignment weakly satisfies the constraint with probability at least

$$\frac{1}{|T_{i_{j_1}}|} \cdot \frac{1}{|T_{i_{j_2}}|} \geq \frac{\Gamma^2}{4t^2}.$$

As this happens for at least a fraction $2^{-k}\gamma/2$ of the constraints our probabilistic assignment weakly satisfies, on the average, a fraction at least

$$\frac{2^{-k}\gamma\Gamma^2}{8t^2}$$

of the constraints. Clearly there exists a standard, deterministic assignment that satisfies the same fraction of the constraints. This finishes the proof of Lemma 4.2.

As stated before Lemma 4.1 and Lemma 4.2 together with the fact that the acceptance criteria of Q-test(δ) is given by Q is sufficient to prove Theorem 4.2. Note that the randomness used by the verifier is bounded by $O(\log n)$ and most of the randomness is used to choose a random constraints as all other random choices only require $O(1)$ random bits.

We do not give the details of these standard parts of the proof here. In short, an approximation algorithm for Max-CSP-(Q) can be used to solve the problem established to be hard in Theorem 4.3.

5 Random Predicates

Remember that we allow negation of inputs and permutation of input variables and hence two predicates that can be obtained from each other by such operations are equivalent. Thus Theorem 4.2 does not only apply to P_{ST} but also to any predicate which is equivalent to it.

Consider the following space of random predicates.

Definition 5.1. *Let $Q_{p,k}$ be the probability space of predicates in k variables where each input is accepted with probability p.*

A uniformly random predicate corresponds to a predicate from $Q_{1/2,k}$ but we will consider also smaller values of p. Whenever needed in calculations we assume $p \leq 1/2$.

We want to analyze the probability that a random predicate from $Q_{p,k}$ is implied by a negated and/or permuted variant of P_{ST} and let us just check that it is reasonable to believe that this is the case.

We have $k!$ permutations of the inputs and 2^k possible ways to negate the inputs. Thus the expected number of P_{ST}-equivalent predicates that imply a random predicate from $Q_{p,d}$ is

$$p^{2^d} 2^k k!.$$

There is hope if this number is at least one, which, ignoring low order terms, happens as soon as

$$p \geq k^{-k2^{-d}}.$$

This lower bound is between k^{-1} and $k^{-1/2}$ and in particular it is smaller than any constant. In fact this rough estimate turns out to be rather close to the truth and the proof is an application of the second moment method. A problem to be overcome is that some pairs of P_{ST}-equivalent predicate have large intersection of their accepted sets. To address this problem we pick a large subset of the P_{ST}-equivalent predicates with bounded size intersections.

Theorem 5.1. *Assuming UGC and $2^{d-1} \leq k \leq 2^d - 1$ then, there is a value c of the form $c = k2^{-d}(1 - o(1))$, such that, with probability $1 - o(1)$, a random predicate chosen according to $Q(p, k)$ with $p = k^{-c}$ is approximation resistant.*

Proof. In view of Theorem 4.2 we need only prove that a random predicate from $Q_{p,k}$ with high probability is implied by some predicate which can be obtained from P_{ST} by negations and/or permutations of inputs.

Let us denote the set accepted by P_{ST} by L. It is a linear space of dimension d. Negating one or more inputs gives an affine space that is either L or completely disjoint from L. We get 2^{k-d} disjoint affine spaces denoted by $L + \alpha$ where α ranges over a suitable set of cardinality 2^{k-d}. We can also permute the coordinates and this gives a total of $k!2^{k-t}$ sets

$$\pi(L + \alpha)$$

Consider

$$\pi(L + \alpha) \cap \pi'(L + \beta).$$

It is an affine space which is either empty or of dimension of the linear space

$$\pi(L) \cap \pi'(L)$$

The below lemma is useful towards this end. Due to space limitations the proof of the lemma will only appear in the full version.

Lemma 5.1. *Let d_0 and k_0 be sufficiently large constants and let r be a number such that $2^{d-r} \geq d_0$ and assume that $k \geq k_0$. Then, if π and π' are two random permutations we have*

$$Pr[dim(\pi(L) \cap \pi'(L)) \geq r] \leq 2^{(2-r)k}.$$

Set $R = 2^{k(r-2)}$ and let us see how to use Lemma 5.1 to choose R different permutations π_i such that

$$dim(\pi_i(L) \cap \pi_j(L)) \le r$$

for any $i \neq j$. First pick $2R$ random permutations. The expected number of pairs (i, j), $i < j$, with

$$dim(\pi_i(L) \cap \pi_j(L)) > r$$

is bounded by $2R^2 2^{(1-r)k} \le R$ and hence there is a choice of $2R$ permutations such that the number of such pairs is bounded by R. Erase one of the two permutations in each such pair and we have the desired set. Let us fix this set $(\pi_i)_{i=1}^R$ once and for all.

Let $X_{i,\alpha}$ be the indicator variable for the event a random predicate from $Q_{k,p}$ is identically one on the set

$$\pi_i(L + \alpha).$$

Set

$$X = \sum_{i,\alpha} X_{i,\alpha}.$$

The probability of the event that the random predicate is not identically one on any $\pi_i(L + \alpha)$ is now exactly $Pr[X = 0]$ and we estimate the probability of this event. Clearly

$$E[X] = p^{2^d} 2^{k-d} R. \tag{7}$$

The variance of X equals

$$E\left[\sum_{i_1,i_2,\alpha_1,\alpha_2} (X_{i_1,\alpha_1} - p^{2^d})(X_{i_2,\alpha_2} - p^{2^d})\right]. \tag{8}$$

We have the following lemma.

Lemma 5.2. *We have* $E[(X_{i_1,\alpha_1} - p^{2^d})(X_{i_2,\alpha_2} - p^{2^d})] = 0$ *if* $\pi_{i_1}(L + \alpha_1)$ *and* $\pi_{i_2}(L+\alpha_2)$ *are disjoint while if the size of the intersection is* K *it is bounded by*

$$p^{2^{d+1}-K}.$$

Proof. In fact

$$E[(X_{i_1,\alpha_1} - p^{2^d})(X_{i_2,\alpha_2} - p^{2^d})] = E[X_{i_1,\alpha_1} X_{i_2,\alpha_2}] - p^{2^{d+1}} = p^{2^{d+1}-K} - p^{2^{d+1}}.$$

Let us now estimate (8). Terms with $i_1 = i_2$ are easy as the corresponding sets either have full intersection or are disjoint. These give a contribution that is upper bounded by $E[X]$. Now for $i_1 \neq i_2$ let us fix α_1 and consider

$$\sum_{\alpha_2} E\left[(X_{i_1,\alpha_1} - p^{2^d})(X_{i_2,\alpha_2} - p^{2^d})\right]. \tag{9}$$

It is the case that for some $r' \leq r$ we have $2^{d-r'}$ terms with set intersection size $2^{r'}$ while all other intersections are empty leading to the upper estimate

$$2^{d-r'} p^{2^{d+1}-2^{r'}} \leq 2^{d-r} p^{2^{d+1}-2^{r}}$$

(using the assumption $p \leq 1/2$) for the sum (9). Summing over all i_1, i_2 and α_1 we get

$$\sigma^2(X) \leq E[X] + R^2 2^{k-d} 2^{d-r} p^{2^{d+1}-2^r} = E[X] + R^2 2^{k-r} p^{2^{d+1}-2^r}. \qquad (10)$$

We have

$$Pr[X=0] \leq \frac{\sigma^2(X)}{E[X]^2} \leq \frac{1}{E[X]} + \frac{R^2 2^{k-r} p^{2^{d+1}-2^r}}{R^2 2^{2(k-d)} p^{2^{d+1}}} \leq \frac{1}{E[X]} + 2^{2d-(k+r)} p^{-2^r} (11)$$

We need to choose p and r to make this probability $o(1)$. Set $p = k^{-c}$ for some $c \leq 1$. Then provided

$$2^r \log k < (k+r) - 2d - \omega(1)$$

the second term of (11) is small. This is possible to achieve with $r = d - \Theta(\log d)$. Note that this choice also ensures $d - r \in \omega(1)$ as required by Lemma 5.1.

Fixing this value of r the first term of (11) is $o(1)$ provided that

$$p^{2^d} \geq 2^{(2-r)k}$$

which with, $p = k^{-c}$, is equivalent to

$$c \leq k 2^{-d} \cdot \frac{r-2}{\log k}. \qquad (12)$$

As the second factor of the bound in (12) is $(1 - o(1))$ we have proved Theorem 5.1.

Apart from adjustments of the error terms this is the best that can be obtained by the current methods. Namely setting $p = k^{-(k2^{-d}+\epsilon)}$ for $\epsilon > 0$ the probability of a random predicate being implied by some P_{ST}-equivalent predicate goes to 0 as can be seen from calculating the expected value of the number of such predicates.

One can always wonder about reasonable values for p for small values of k. Particularly good values for k are numbers of the form $2^d - 1$ as this gives an unusually sparse predicate P_{ST}. Numerical simulations suggests that a random predicate on 7 bits that accepts M of the 128 inputs has a probability at least 1/2 of being implied by a P_{ST}-equivalent predicate iff $M \geq 60$. Thus it seems like the asymptotic bound of density essentially k^{-1} is approached slowly.

6 Very Dense Predicates

As P_{ST} only accepts 2^d inputs we can derive approximation resistance of many predicates but let us here give only one immediate application.

Theorem 6.1. *Let $2^{d-1} \leq k \leq 2^d - 1$ and P be any predicate that accepts at least $2^k + 1 - 2^{k-d}$ inputs, then, assuming the UGC, P is approximation resistant.*

Proof. We use the same notation as used in the proof of Theorem 5.1.

We need to prove that any such predicate is implied by a P_{ST}-equivalent predicate. This time we need only apply negations and look at $L + \alpha$ for all the 2^{k-d} different representatives α. As P only rejects $2^{k-d} - 1$ different inputs and the sets $L + \alpha$ are disjoint, one such set is included in the accepted inputs of P. The corresponding suitable negated form of P_{ST} hence implies P and Theorem 6.1 follows from Theorem 4.2.

It is an interesting question how dense a non-trivially approximable predicate can be. Let d_k be the maximum value of $d(P)$ for all predicates on k variables which are not approximation resistant. We have $d_2 = d_3 = 3/4$ and Hast [5] proved that $d_4 = \frac{13}{16}$ and, as we can always ignore any input, d_k is an increasing function of k. It is not obvious whether d_k tends to one as k tends to infinity.

Our results show that dense predicates which can be non-trivially approximated need to be extremely structured as they cannot be implied by any P_{ST}-equivalent predicate.

7 Concluding Remarks

The key result in the current paper is to prove that P_{ST} is hereditary approximation resistant. This is another result indicating that the more inputs accepted by the predicate P, the more likely it is to be approximation resistant. One could be tempted to conclude that all approximation resistant predicates are in fact hereditary approximation resistant. We would like to point that this is false and Hast [5] has an example of two predicates P and Q where P is approximation resistant, P implies Q and Q is not approximation resistant.

That a predicate is approximation resistant is almost the ultimate hardness. There is a stronger notion; approximation resistance on satisfiable instances. In such a case no efficient algorithm is able to do significantly better than picking a random assignment even in the case when the instance is satisfiable.

An example of a predicate which is approximation resistant but not approximation resistant on satisfiable instances is Max-E3-Lin-2, linear equations modulo 2 with three variables in each equation. In this case, for a satisfiable instance, it is easy to find an assignment that satisfies all constraints by Gaussian elimination.

In most cases, however, approximation resistant predicates have turned out to be approximation resistant also on satisfiable instances and it would seem reasonable to conjecture that a random predicate is indeed approximation resistant on satisfiable instances. If true it seems hard to prove this fact using the Unique Games Conjecture in that the non-perfect completeness of UGC would tend to produce instances of the CSP which are not satisfiable. There are variants of the unique games conjecture [8] which postulate hardness of label cover problems with perfect completeness but it would be much nicer to take a different route not relying on any conjectures.

Another open problem is of course to establish approximation resistance in absolute terms and not to rely on the UGC or, more ambitiously, to prove the UGC.

Acknowledgment: I am grateful to Per Austrin for useful comments on the current manuscript.

References

1. M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.
2. T. Gowers. A new proof of Szemerédi's theorem for progressions of length four. *Geometric and Functional Analysis*, 8:529–551, 1998.
3. T. Gowers. A new proof of Szemerédi's theorem. *Geometric and Functional Analysis*, 11:465–588, 2001.
4. V. Guruswami, D. Lewin, M. Sudan, and L. Trevisan. A tight characterization of NP with 3 query PCPs. In *Proceedings of 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 8–17, Palo Alto, 1998. IEEE.
5. G. Hast. *Beating a random assignment.* KTH, Stockholm, 2005. Ph.D Thesis.
6. J. Håstad. Some optimal inapproximability results. *Journal of ACM*, 48:798–859, 2001.
7. J. Håstad. Every 2-CSP allows nontrivial approximation. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computation*, pages 740–746, 2005.
8. S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of 34th ACM Symposium on Theory of Computating*, pages 767–775, 2002.
9. S. Khot and O. Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. In *Proc. of 18th IEEE Annual Conference on Computational Complexity (CCC)*, pages 379–386, 2003.
10. A. Samorodnitsky and L. Trevisan. A PCP characterization of NP with optimal amortized query complexity. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 191–199, 2000.
11. A. Samorodnitsky and L. Trevisan. Gowers uniformity, influence of variables and PCPs. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 11–20, 2006.
12. T. Schaefer. The complexity of satisfiability problems. In *Conference record of the Tenth annual ACM Symposium on Theory of Computing*, pages 216–226, 1978.
13. U. Zwick. Personal Communication.
14. U. Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 201–210. ACM, 1998.