Approximating maximum satisfiable subsystems of linear equations of bounded width

Zeev Nutov

The Open University of Israel

Daniel Reichman

The Open University of Israel

Abstract

We consider the problem known as MAX-SATISFY: given a system of m linear equations over the rationals, find a maximum set of equations that can be satisfied. Let r be the width of the system, that is, the maximum number of variables in an equation. We give an $\Omega(m^{-1+1/r})$-approximation algorithm for any fixed r. Previously the best approximation ratio for this problem was $\Omega((\log m)/m)$ even for $r = 2$. In addition, we slightly improve the hardness results for MAX-SATISFY.

Key words: Linear equations, Satisfiable systems, Approximation algorithms

1 Introduction

One of the most fundamental computational tasks is solving a set of linear equations over a field. If the whole system is satisfiable (namely, if there exists an assignment of the field elements to the variables that satisfies all equations in the system), then the Gaussian elimination procedure solves the system in polynomial time. If, on the other hand, the system is not satisfiable, then finding an assignment satisfying the maximum number of equations is NP-hard. When the field is the rational numbers Q, this optimization problem is called MAX-SATISFY. To avoid dealing with equivalent equations (namely, when one

Email addresses: nutov@openu.ac.il (Zeev Nutov), daniel.reichman@gmail.com (Daniel Reichman).

equation is obtained by multiplying the other by a non-zero constant), we consider the following more general weighted version of the problem.

MAX-SATISFY:
Instance: A set E of m linear equations on a set V of n variables and positive rational weights $\{w(e) : e \in E\}$.
Objective: Find a maximum weight satisfiable subset $I \subseteq E$ of equations.

MAX-SATISFY has practical importance in various fields such as pattern recognition, operational research, artificial neural networks, and more, c.f., [1]. It also has theoretical importance. In particular, it is related to the Label-Cover [6,9] and Learning-Half-Spaces with Error [9,7]. How well can one approximate this problem? It has been proved in the mid 90's that approximating MAX-SATISFY within $1/m^{\alpha}$ where α is some positive universal constant is NP-hard [1,2]. Furthermore, it was shown that the problem cannot be approximated within $\Omega(1/m^{1-\varepsilon})$ for any constant $\varepsilon > 0$ unless NP$\subseteq$BPP [6].

Definition 1.1 *The* width *of a system of equations is the maximum number of variables in an equation of the system.*

The hardness result $\Omega(1/m^{1-\varepsilon})$ of [6] does not apply for systems of small width. Previously, the best known approximation ratio for systems of width 2 was $\Omega((\log m)/m)$ [11], whereas the hardness result known for this case [5] only rules out an approximation ratio of $\Omega(1/2^{\log^{1-\varepsilon} m})$, assuming NP-hard problems cannot be solved in quasi-polynomial time.

The Unique Games Conjecture of Khot [13], if true, has been shown to imply hardness of approximation results for several important NP-hard problems (e.g., see [4] for more details). Systems of linear equations of width 2 over finite fields are closely related to this conjecture [5,14] and have recently attracted a lot of attention. The papers [4,8,16] considered finite fields and focused on finding a relatively large feasible subsystem conditioned that the system is "almost" satisfiable (namely, a fraction of $1-\varepsilon$ of the equations can be satisfied for some small ε); these results are based on linear and semi-definite programs in which the finiteness of the field is heavily used, and do not seem to apply for infinite fields, such as Q. Trevisan [16] gives a combinatorial algorithm that can be adapted for Q, but it applies for almost satisfiable instances and does not seem to work for the general case.

In this paper we focus on approximating MAX-SATISFY instances with small width. Let r-MAX-SATISFY be the restriction of MAX-SATISFY to instances of width at most r. We give a simple combinatorial algorithm for r-MAX-SATISFY as follows:

Theorem 1.1 r-MAX-SATISFY *admits an* $\Omega(m^{-1+1/r})$*-approximation algorithm with running time* $O(rm^{r+1})$.

Note that for any fixed r the running time is polynomial. Importantly, for system of bounded width it is possible to achieve substantially better approximation ratios than for the general case. In particular, for $r = 2$ the approximation ratio is $\Omega(1/\sqrt{m})$, and for $r = 3$ the approximation ratio is $\Omega(1/m^{2/3})$. Thus unlike finite fields, where the hardest instances to approximate are of width 3 [12], when equations over Q are considered, we can achieve substantially better approximation ratios for systems of bounded width. This result is interesting also in view of the recent hardness result proved in [10], where it is shown that for systems of width 3 it is NP-hard to decide whether the optimum is at least $(1-\varepsilon)m$ or at most δm, for every $0 < \varepsilon, \delta < 1/2$.

Our algorithm uses a greedy strategy, but it is not straightforward. The difficulty is that, even for $r = 2$, choosing one "bad" equation into a partial solution may prevent from adding any other remaining equation. We use decomposition tools to overcome this difficulty.

In addition, we slightly improve the hardness results for the problem. In [6] it is shown that MAX-SATISFY cannot be approximated within $1/m^{1-\varepsilon}$ for any $\varepsilon > 0$ unless NP $\subseteq$ BPP. The systems constructed in [6] have width $\Theta(\ln m)$. Under the assumption that P$\neq$NP, it is proved in [2] that r-MAX-SATISFY with $r = \Theta(\ln m)$ cannot be approximated within $1/m^{\alpha}$ for some $\alpha > 0$; the value of α in [2] is not computed explicitly, but it is known to be strictly smaller than 1/2, see [6]. We improve the hardness result of [2] as follows:

Theorem 1.2 *r-MAX-SATISFY with $r = \Theta(\ln m)$ cannot be approximated within $O(1/m^{1/2-\varepsilon})$ for any constant $\varepsilon > 0$, unless* P=NP .

Our proof is simpler than the proofs in [6,2]. It is based on a recent construction of Zuckerman [17]. Currently, this is the best known hardness result under the assumption that P$\neq$NP.

Preliminaries:

Let E be a system of linear equations with variable set V over the field Q of rationals. Let $m = |E|$ and $n = |V|$. An equation $e \in E$ *contains a variable* $v \in V$ if the coefficient of v in e is not zero. Let $\delta_E(v)$ denote the set of equations in E containing a variable v. A set of equations is *linearly independent* if the coefficients vectors of the equations in the set are linearly independent. The *rank* of a linear system is the maximum number of linearly independent equations in the system. Two equations $\vec{a}'\cdot\vec{x} = b'$ and $\vec{a}''\cdot\vec{x} = b''$, where $\vec{a}', \vec{a}'' \in Q^V$ and $b', b'' \in Q$, are *equivalent* if $\vec{a}'' = c\vec{a}'$ and $b'' = cb'$ for some constant $c \neq 0$. Given an instance of MAX-SATISFY we may assume that it is *proper*, namely that no two equations are equivalent; otherwise, for every maximal set of equivalent equations we keep one and set its weight to be the total weight of this set. Let opt be an optimal solution value for an instance at hand. For a maximization problem, we say that an algorithm has approximation ratio ρ, or that it is a ρ-approximation algorithm, where $0 < \rho \leq 1$, if it runs in polynomial time and delivers a solution of value at least ρ times the value of an optimal solution. When considering running times, it is assumed that basic arithmetic operations between rational numbers (addition, subtraction, multiplication and division) take $O(1)$ time.

2 The algorithm

The key statement toward proving Theorem 1.1 is the following.

Theorem 2.1 r-MAX-SATISFY *admits an algorithm with running time* $O(rm^{r+1})$ *and approximation ratio*

$$\frac{1}{r \cdot \binom{n}{r-1}} \geq \frac{1}{r} \cdot \left(\frac{r-1}{ne}\right)^{r-1} \equiv \rho(r,n) \ .$$

Given Theorem 2.1, whose proof is deferred to the next section, the algorithm is as follows. Let ℓ be a parameter, to be determined later. The following algorithm starts with $I_1 = I_2 = \emptyset$ and returns a satisfiable set $I_1 + I_2 \subseteq E$.

Phase 1
While there is $v \in V$ with $|\delta_E(v)| \leq \ell$ do:
  $I_1 \leftarrow I_1 \cup e_v$, where e_v is the maximum weight edge in $\delta_E(v)$.
  $E \leftarrow E \setminus \delta_E(v)$, $V \leftarrow V \setminus \{v\}$.
EndWhile

Phase 2
Compute a set $I_2 \subseteq E$ of equations using the algorithm as in Theorem 2.1.
Return $I \leftarrow I_1 \cup I_2$.

We claim that the algorithm returns a feasible solution. Indeed, I_2 is clearly satisfiable. The coefficient matrix of I_1 is upper triangular with no zero rows. Hence I_1 is satisfiable. As every $e \in I_1$ contains a variable not appearing in any other equation in I_2, we have that $I_1 \cup I_2$ is also satisfiable.

We now prove the approximation ratio. Let (V_2, E_2) denote the instance at the beginning of Phase 2, and let $n_2 = |V_2|$ and $m_2 = |E_2|$. We claim that the algorithm has approximation ratio $\min\{1/\ell, \rho(r, n_2)\}$. Let F be an optimal solution, let F_1 be the edges in F incident to nodes deleted at Phase 1, and let

$F_2 = F - F_1$. It is easy to see that $w(I_1) \geq w(F_1)/\ell$ and $w(I_2) \geq \rho(r, n_2) \cdot w(F_2)$. Thus

$$\frac{w(I)}{w(F)} = \frac{w(I_1) + w(I_2)}{w(F_1) + w(F_2)} \geq \min\left\{\frac{w(I_1)}{w(F_1)}, \frac{w(I_2)}{w(F_2)}\right\} \geq \min\left\{\frac{1}{\ell}, \rho(r, n_2)\right\} .$$

We have $r \cdot m_2 \geq \ell \cdot n_2$, since $|\delta_E(v)| \geq \ell$ for all $v \in V_2$ and since every equation contains at most r variables. Thus $n_2 \leq rm_2/\ell \leq rm/\ell$. Consequently, the approximation ratio in terms of m and ℓ is $\min\{1/\ell, \rho(r, rm/\ell)\}$.

Note that

$$\rho(r, rm/\ell) = \ell^{r-1} \cdot \frac{1}{r} \cdot \left(\frac{r-1}{e \cdot r}\right)^{r-1} \cdot m^{1-r} .$$

Solving the equation $1/\ell = \rho(r, rm/\ell)$ gives

$$\ell = \left[r^{1/r} \cdot \left(\frac{e \cdot r}{r-1}\right)^{1-1/r}\right] \cdot m^{-1+1/r} = \Theta(m^{-1+1/r})$$

Substituting this ℓ in $\min\{1/\ell, \rho(r, rm/\ell)\}$ gives the approximation ratio as in Theorem 1.1.

The running time spent for computing I_1 at Phase 1 is $O(m^2)$, hence the total time is as in Theorem 2.1.

3 Proof of Theorem 2.1

Theorem 2.1 will follow from the following three statements.

Lemma 3.1 MAX-SATISFY *can be solved (exactly) in time $O(nm^{R+1})$, where R is the rank of the system.*

Proof: Note that if $I \subseteq E$ is an inclusion maximal satisfiable sub-system of E, then I has rank R. Otherwise, there is an equation $e \in E \setminus I$ so that e is linearly independent of the equations in I. But then $I \cup \{e\}$ is a satisfiable system, contradicting the maximality of I. As the weights are positive, every optimal solution is inclusion maximal, and thus has rank R.

The algorithm is as follows. Find the rank R of the system using Gaussian elimination. Then, for every subset $I \subseteq E$ of size R do the following. First, using Gaussian elimination, find a satisfying assignment to the system (V, I), or determine that such does not exist; note that if I has rank R, then every variable $v \in V$ is contained in some equation in I. Second, if I is satisfiable and has rank R, substitute the assignment computed into all equations in E

and return the set E_I which this assignment satisfies. Finally, among the sets E_I computed output one of the maximum weight.

The correctness of the algorithm is straightforward. We show the time complexity. Finding the rank of the system can be done in $O(n^2m)$ time. The number of iterations is $\binom{m}{R}$ and the time complexity per iteration is $O(R^2n+nm) = O(n(R^2+m))$; indeed, solving each system I can be done in $O(nR^2)$ time, while substituting this solution into all equations requires $O(nm)$ time. Hence the overall time complexity is $O\left(n^2m + \binom{m}{R}n(R^2+m)\right) = O(nm^{R+1})$. $\square$

Note that r-MAX-SATISFY instances can have rank n, hence Lemma 3.1 does *not* imply that r-MAX-SATISFY can be solved in $O(nm^{r+1})$ time. However, r-MAX-SATISFY instances with r variables have rank at most r, and thus can be solved in $O(rm^{r+1})$ time.

Partition the set E of equations as follows. For every equation e choose an arbitrary set $X_e \subseteq V$ of size r that contains every variable contained in e. Let $E(X) = \{e \in E : X_e = X\}$. Consider the r-uniform hypergraph $\mathcal{H} = (V, \mathcal{E})$, where $\mathcal{E} = \{X \subseteq V : |X| = r, E(X) \neq \emptyset\}$. Note that $\{E(X) : X \in \mathcal{E}\}$ is a partition of E, thus $|\mathcal{E}| \leq |E| = m$. For $X \in \mathcal{E}$, let F_X be an optimal solution to the instance $E(X)$, and let $p(X) = w(F_X)$. For every $X \in \mathcal{E}$, $p(X)$ can be computed in $O(r|E(X)|^{r+1})$ time, by Lemma 3.1; hence the total time complexity for computing $\mathcal{H}, p$ is $O(rm^{r+1})$. Clearly,

$$\mathsf{opt} \leq p(\mathcal{H}) = \sum_{X \in \mathcal{E}} p(X) = \sum_{X \in \mathcal{E}} \sum_{e \in F_X} w(e) \ . \tag{1}$$

The following statement is immediate.

Lemma 3.2 *Let $\mathcal{M} \subseteq \mathcal{E}$ be a matching in $\mathcal{H}$, that is, $\mathcal{M}$ is a subset of pairwise disjoint sets from $\mathcal{E}$. Then the subset $F_{\mathcal{M}} = \bigcup_{X \in \mathcal{M}} F_X \subseteq E$ of equations in E that corresponds to $\mathcal{M}$ is a satisfiable system.*

Lemma 3.3 *Let $\mathcal{H} = (V, \mathcal{E})$ be any simple r-uniform hypergraph with hyperedge weights $\{p(X) : X \in \mathcal{E}\}$. Then one can find in $O(m^2)$ time a matching $\mathcal{M}$ in $\mathcal{H}$ of weight at least*

$$p(\mathcal{M}) \geq \frac{p(\mathcal{H})}{r \cdot \binom{n}{r-1}} \ . \tag{2}$$

Proof: Starting with $\mathcal{M} = \emptyset$, the algorithm iteratively finds the heaviest hyperedge X in $\mathcal{E}$, adds X to $\mathcal{M}$, and removes from $\mathcal{E}$ all the hyperedges intersecting X. This procedure is repeated until no edges are left in $\mathcal{H}$. The analysis of the running time is straightforward. Inequality (2) follows from the observation that in a simple r-uniform hypergraph the degree of every node is at most $\binom{n}{r-1}$; thus the number of hyperedges intersecting a single hyperedge is at most $r\binom{n}{r-1}$. Thus when X is added to $\mathcal{M}$, the weight of the hyperedges deleted from $\mathcal{H}$ is at most $p(X) \cdot r\binom{n}{r-1}$. The statement follows. $\square$

Theorem 2.1 now follows easily by combining Lemmas 3.2 and 3.3. After computing the hypergraph $\mathcal{H} = (V, \mathcal{E})$ and the weights $\{p(X) : X \in \mathcal{E}\}$, we compute a matching $\mathcal{M}$ as in Lemma 3.3, and output the subset $F_{\mathcal{M}} \subseteq E$ of equations that corresponds to $\mathcal{M}$. The dominating time is spent for computing the weight function p, which is $O(rm^{r+1})$.

The proof of Theorem 2.1 is complete.

4 Proof of Theorem 1.2

In this section we give a proof sketch of Theorem 1.2, namely, that r-MAX-SATISFY with $r = \Theta(\ln m)$ cannot be approximated within $O(1/m^{1/2-\varepsilon})$ for any constant $\varepsilon > 0$, unless P=NP. We assume familiarity of the reader with proof systems. For more on this subject see [3]. We use the following result due to Zuckerman [17]:

Lemma 4.1 ([17]) *For any $\varepsilon > 0$, $\mathrm{NP} \subset \mathrm{FPCP}_{2^{(\varepsilon-1)R}}(R, \varepsilon R)$, where $R = O(\log m)$ (m is the size of the input). Further, the query complexity of the above proof system is $O(R)$.*

The main idea is to arithmetize the above proof system in a similar way to [6].

Create a linear set of equations over the rationals as follows. For the proof system above, we give a variable for any position in the proof that has positive probability of being queried. Assume our query complexity is q. If the values $b_1, ..., b_q$ cause the verifier to accept, we add the ℓ equations $\sum_{i=1}^{q} (x_i - b_i) = 0$, $\sum_{i=1}^{q} (x_i - b_i)2^i = 0, \ldots, \sum_{i=1}^{q} (x_i - b_i)\ell^i = 0$, where $\ell = 2^R$. (Note – the indices of the variables should correspond to the queried positions and not to $1, 2...q$. We write it like we did to avoid notational difficulties). Clearly if $x_i = b_i$ for every i, then all ℓ equations are satisfied. If $x_i - b_i$ is nonzero for some i then at most q out of the ℓ equations are satisfied.

We get a total of $2^{2R+\varepsilon}$ equations. The width of the system is the query complexity which is $O(R)$. If we have a proof that is accepted with probability 1, then we can satisfy at least 2^{2R} equations. If every proof is accepted with probability at most $2^{(\varepsilon-1)R}$, then we can satisfy at most $\frac{2^{2R}}{2^{(1-\varepsilon)R}} + q2^{R+\varepsilon R}$ equations. For large enough R this is no larger than $2^{(1+2\varepsilon)R}$ (recall that $q = O(R)$). Hence for a system containing $2^{2R+\varepsilon}$ equations, it is NP-hard to distinguish between the case in which we can satisfy 2^{2R} equations, to the case in which we can satisfy at most $2^{(1+2\varepsilon)R}$ equations. The result follows.

5 Conclusions

We have shown that r-MAX-SATISFY admits an $\Omega(m^{-1+1/r})$-approximation algorithm with running time $O(rm^{r+1})$. Our algorithm is combinatorial in nature and easy to implement. There is still a big gap, even for $r = 2$, between the approximation guarantee provided by our algorithm and the lower bound known [5]. Narrowing this gap is a challenging open problem even for $r = 2$, as improving the approximation ratio of $O(1/\sqrt{m})$ for MAX-2-SATISFY implies improving the best known $\Omega(1/\sqrt{m})$-approximation [15] to LABEL-COVER, see [5]. We also note that for $r = 2$, no hardness for the $(1-\varepsilon)$-satisfiable version is known. Over finite fields, this is equivalent to the Unique Games Conjecture; a similar statement is not known for the field Q of rational numbers. Finally, observe that our algorithm applies for any field for which Lemma 3.1 is valid; in particular, it applies to every finite field.

Acknowledgment: We thank anonymous referees for their useful comments on a previous version of this paper.

References

[1] E. Amaldi and V. Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theor. Comput. Sci.*, 147(1-2):181–210, 1995.

[2] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997.

[3] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs, and nonapproximability - towards tight results. *SIAM J. Comput.*, 27(3):804–915, 1998.

[4] M. Charikar, K. Makarychev, and Y. Makarychev. Near optimal algorithms for unique games. In *Proc. Symposium on the Theory of Computing (STOC)*, pages 205–214, 2006.

[5] U. Feige and D. Reichman. On systems of linear equations with two variables per equation. In *Proc. Workshop on Approximation algorithms (APPROX)*, pages 117–127, 2004.

[6] U. Feige and D. Reichman. On the hardness of approximating max-satisfy. *Inf. Process. Lett.*, 97(1):31–35, 2006.

[7] V. Feldman, P. Gopalan, S. Khot, and A. K. Ponnuswami. New results for learning noisy parities and halfspaces. In *Proc. Symposium on the Foundations of Computer Science (FOCS)*, pages 563–574, 2006.

[8] A. Gupta and K. Talwar. Approximating unique games. In *Proc. Symposium on Discrete Algorithms (SODA)*, pages 99–106, 2006.

[9] V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. In *Proc. Symposium on the Foundations of Computer Science (FOCS)*, pages 543–552, 2006.

[10] V. Guruswami and P. Raghavendra. A 3-query pcp over integers. In *Proc. Symposium on the Theory of Computing (STOC)*, pages 198–206, 2007.

[11] M. M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *J. Graph Algorithms Appl.*, 4(1), 2000.

[12] J. Hástad. Some optimal inapproximability results. *Journal of the ACM*, 48(1), 2001.

[13] S. Khot. On the power of unique 2-prover 1-round games. In *Proc. Symposium on the Theory of Computing (STOC)*, pages 767–775, 2002.

[14] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? In *Proc. Symposium on the Foundations of Computer Science (FOCS)*, pages 146–154, 2004.

[15] D. Peleg. Approximation algorithms for the label-covermax and red-blue set cover problems. In *Proc. Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 220–230, 2000.

[16] L. Trevisan. Approximation algorithms for unique games. In *Proc. Symposium on the Foundations of Computer Science (FOCS)*, pages 197–205, 2005.

[17] D. Zuckerman. Linear degree extractors and the inapproximability of max-clique and chromatic number. In *Proc. Symposium on the Theory of Computing (STOC)*, pages 681–690, 2006.