Electronic Colloquium on Computational Complexity, Report No. 63 (2006)

Approximation Algorithm for the Max k-CSP Problem

Moses Charikar* Konstantin Makarychev† Yury Makarychev‡

Princeton University

Abstract

We present a $\frac{ck}{2^k}$ approximation algorithm for the Max k-CSP problem (where $c > 0.44$ is an absolute constant). This result improves the previously best known algorithm by Hast, which has an approximation guarantee of $\Omega(\frac{k}{2^k \log k})$. Our approximation guarantee matches the upper bound of Samorodnitsky and Trevisan up to a constant factor (their result assumes the Unique Games Conjecture).

1 Introduction

In this paper we present an approximation algorithm for the Max k-CSP problem.

Definition 1.1 (Max k-CSP Problem). *Given a set of boolean variables and constraints, where each constraint depends on k variable. The goal is to find an assignment so as to maximize the number of satisfied constraints.*

Note that a random assignment satisfies each constraint with probability at least $\frac{1}{2^k}$ (except those constraints which cannot be satisfied). Therefore, the random assignment algorithm gives a $\frac{1}{2^k}$ approximation. Recently, Hast [2] proposed an algorithm with a much better approximation guarantee $\Omega\left(\frac{k}{2^k \log k}\right)$. We further improve the approximation ratio to $\Omega\left(\frac{k}{2^k}\right)$. This result is also applicable to Max-k CSP with a larger domain[1], it gives a $\Omega\left(\frac{k \log d}{d^k}\right)$ approximation for instances with domain size d. Our approach is similar to that of Hast.

Samorodnitsky and Trevisan [5] proved that it is hard to approximate Max k-CSP within $\frac{2k}{2^k}$ for every $k \geq 3$), and within $\frac{k+1}{2^k}$ for infinitely many k assuming the Unique Games

*http://www.cs.princeton.edu/~moses/ Supported by NSF ITR grant CCR-0205594, NSF CAREER award CCR-0237113, MSPA-MCS award 0528414, and an Alfred P. Sloan Fellowship.

†http://www.cs.princeton.edu/~kmakaryc/ Supported by a Gordon Wu fellowship.

‡http://www.cs.princeton.edu/~ymakaryc/ Supported by a Gordon Wu fellowship.

[1]To apply the result to an instance with a larger domain, we just encode each domain value with $\log d$ bits.

ISSN 1433-8092

Conjecture of Khot [3]. This shows that our algorithm is asymptotically optimal within a factor of $\approx 1/0.44 \approx 2.27$.

We use Hast's reduction of the Max k-CSP problem to the Max k-AllEqual problem.

Definition 1.2 (Max k-AllEqual Problem). *Given a set S of clauses of the form $l_1 \equiv l_2 \equiv \cdots \equiv l_k$, where each literal l_i is either a boolean variable x_j or its negation $\bar{x}_j$. The goal is to find an assignment to the variables x_i so as to maximize the number of satisfied clauses.*

The reduction works as follows. First, we write each constraint $f(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$ as a CNF formula. Then we consider each clause in the CNF formula as a separate constraint; we get an instance of the Max k-CSP problem, where each clause is a conjunction. The new problem is equivalent to the original problem: each assignment satisfies exactly the same number of clauses in the new problem as in the original problem. Finally, we replace each conjunction $l_1 \wedge l_2 \wedge \ldots \wedge l_k$ with the constraint $l_1 \equiv l_2 \equiv \cdots \equiv l_k$. Clearly, the value of this instance of Max k-AllEqual is at least the value of the original problem. Moreover, it is at most two times greater then the value of the original problem: if an assignment $\{x_i\}$ satisfies a constraint in the new problem, then either the assignment $\{x_i\}$ or the assignment $\{\bar{x}_i\}$ satisfies the corresponding constraint in the original problem. Therefore, a C approximation guarantee for Max k-AllEqual translates to a $C/2$ approximation guarantee for the Max k-CSP. Below we consider only Max k-AllEqual problem.

For brevity, we denote $\bar{x}_i$ by x_{-i}. We think of each clause C as a set of indices: clause C defines the constraint "(for all $i \in C$, x_i is true) or (for all $i \in C$, x_i is false)". Without loss of generality we assume that there are no unsatisfiable clauses in S, *i.e.* there are no clauses that have both literals x_i and $\bar{x}_i$.

2 Approximation Algorithm

2.1 Relaxation

We consider the following SDP relaxation of Max k-AllEqual problem.

$$\max \frac{1}{k^2} \sum_{C \in S} \left\| \sum_{i \in C} v_i \right\|^2$$
$$\|v_i\|^2 = 1$$
$$v_i = -v_{-i}.$$

This is indeed a relaxation of the problem: in the intended solution $v_i = e$ if x_i is true, and $v_i = -e$ if x_i is false (where e is a fixed unit vector). Then the value of the SDP is greater than or equal to the value of the Max k-AllEqual problem.

2.2 Algorithm

We use the following theorem of Nesterov [4].

Theorem 2.1. *There exists an efficient algorithm that given a positive semidefinite matrix $A = (a_{ij})$, and a set of unit vectors v_i, assigns ± 1 to variables z_i, s.t.*

$$\sum_{i,j} a_{ij}\, z_i z_j \geq \frac{2}{\pi} \sum_{i,j} a_{ij}\, \langle v_i, v_j \rangle.$$

Observe that the quadratic form

$$\frac{1}{k^2} \sum_{C \in S} \left(\sum_{i \in C} z_i \right)^2$$

is positive semidefinite. Therefore we can use the algorithm from Theorem 2.1. Given vectors v_i as in the SDP relaxation, it yields numbers z_i s.t.

$$\frac{1}{k^2} \sum_{C \in S} \left(\sum_{i \in C} z_i \right)^2 \geq \frac{2}{\pi} \frac{1}{k^2} \sum_{C \in S} \left\| \sum_{i \in C} v_i \right\|^2$$
$$z_i \in \{\pm 1\}$$
$$z_i = -z_{-i}$$

(Formally, v_{-i} is a shortcut for $-v_i$; z_{-i} is a shortcut for $-z_i$).

In what follows, we assume that $k \geq 3$ — for $k = 2$ we can use the MAX CUT algorithm by Goemans and Williamson [1] to get a better approximation[2].

We now present the approximation algorithm.

Approximation Algorithm for the Max k-AllEqual Problem

1. Solve the semidefinite relaxation. Get vectors v_i.
2. Apply Theorem 2.1 to vectors v_i as described above. Get values z_i.
3. Let $\delta = \sqrt{\frac{2}{k}}$. For each $i \geq 1$ assign $x_i = 1$ (true) w.p. $\frac{1 + \delta z_i}{2}$, assign $x_i = -1$ (false) w.p. $\frac{1 - \delta z_i}{2}$.

2.3 Analysis

Theorem 2.2. *The approximation algorithm finds an assignment satisfying at least $c\,\frac{k}{2^k}\,OPT$ clauses (where $c > 0.88$ is an absolute constant), given that OPT clauses are satisfied in the optimal solution.*

[2] Our algorithm works for $k = 2$ with a slight modification: δ should be less than 1.

Proof. Denote $Z_C = \frac{1}{k}\sum_{i\in C} z_i$. Then Theorem 2.1 guarantees that

$$\sum_{C\in S} Z_C^2 = \frac{1}{k^2}\sum_{C\in S}\left(\sum_{i\in C} z_i\right)^2 \geq \frac{2}{\pi}\frac{1}{k^2}\sum_{C\in S}\left\|\sum_{i\in C} v_i\right\|^2 = \frac{2}{\pi}SDP \geq \frac{2}{\pi}OPT,$$

where SDP is the SDP value.

Note that the number of z_i equal to 1 is $\frac{1+Z_C}{2}k$, number of z_i equal to -1 is $\frac{1-Z_C}{2}k$. The probability that a constraint C is satisfied equals

$$\begin{aligned}
\Pr\left(C \text{ is satisfied}\right) &= \Pr\left(x_i = 1 \text{ for all } i\in C\right) + \Pr\left(x_i = -1 \text{ for all } i \in C\right)\\
&= \prod_{i\in C}\frac{1+\delta z_i}{2} + \prod_{i\in C}\frac{1-\delta z_i}{2}\\
&= \frac{1}{2^k}\left((1+\delta)^{\frac{1+Z_C}{2}k}\cdot(1-\delta)^{\frac{1-Z_C}{2}k} + (1-\delta)^{\frac{1+Z_C}{2}k}\cdot(1+\delta)^{\frac{1-Z_C}{2}k}\right)\\
&= \frac{1}{2^k}\left(1-\delta^2\right)^{k/2}\left(\left(\frac{1+\delta}{1-\delta}\right)^{Z_Ck/2} + \left(\frac{1-\delta}{1+\delta}\right)^{Z_Ck/2}\right)\\
&= \frac{1}{2^k}2(1-\delta^2)^{k/2}\cosh\left(\frac{1}{2}\ln\frac{1+\delta}{1-\delta}\cdot Z_Ck\right).
\end{aligned}$$

Here, $\cosh t \equiv \frac{e^t+e^{-t}}{2}$. Let α be the minimum of the function $\frac{\cosh t}{t^2}$. Numerical computation shows that $\alpha > 0.93945$. We have,

$$2(1-\delta^2)^{k/2}\cosh\left(\frac{1}{2}\ln\frac{1+\delta}{1-\delta}\cdot Z_Ck\right) > 2(1-\delta^2)^{k/2}\alpha\left(\frac{1}{2}\ln\frac{1+\delta}{1-\delta}\cdot Z_Ck\right)^2.$$

We now lower bound the right hand side. Recall that $\delta = \sqrt{2/k}$ and $k \geq 3$. We have

$$(1-\delta^2)^{k/2} = (1-\frac{2}{k})^{k/2} \geq (1-\frac{2}{k})\frac{1}{e}.$$

Then

$$\left(\ln\frac{1+\delta}{1-\delta}\right)^2 \geq (2\delta)^2 = \frac{8}{k}.$$

Combining these bounds we get,

$$2\alpha(1-\delta^2)^{k/2}\left(\frac{1}{2}\ln\frac{1+\delta}{1-\delta}\cdot Z_Ck\right)^2 \geq (1-\frac{2}{k})\frac{4\alpha k}{e}Z_C^2.$$

However, it turns out that the factor $1-\frac{2}{k}$ is not necessary; the following bound holds,

$$2\alpha(1-\delta^2)^{k/2}\left(\frac{1}{2}\ln\frac{1+\delta}{1-\delta}\cdot Z_Ck\right)^2 \geq \frac{4\alpha k}{e}Z_C^2.$$

We get,

$$\Pr(C \text{ is satisfied}) \geq \frac{4\alpha}{e} \frac{k}{2^k} Z_C^2.$$

So the expected number of satisfied constraints is

$$\sum_{C \in S} \Pr(C \text{ is satisfied}) \geq \sum_{C \in S} \frac{4\alpha}{e} \frac{k}{2^k} Z_C^2 = \frac{4\alpha}{e} \frac{k}{2^k} \sum_{C \in S} Z_C^2 \geq \frac{4\alpha}{e} \frac{k}{2^k} \cdot \frac{2}{\pi} OPT.$$

We conclude that the algorithm finds an $\frac{8\alpha}{\pi e} \frac{k}{2^k} > 0.88 \frac{k}{2^k}$ approximation with high probability. □

References

[1] M. Goemans and D. Williamson. *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming.* Journal of the ACM, vol. 42, no. 6, pp. 1115–1145, Nov. 1995.

[2] G. Hast. *Approximating Max kCSP — Outperforming a Random Assignment with Almost a Linear Factor.* In Proceedings of the 32nd International Colloquium on Automata, Languages and Programming, pp. 956–968, 2005.

[3] S. Khot. *On the power of unique 2-prover 1-round games.* In Proceedings of the 34th ACM Symposium on Theory of Computing, pp. 767–775, 2002.

[4] Y. Nesterov. *Quality of semidefinite relaxation for nonconvex quadratic optimization.* CORE Discussion Paper 9719, March 1997.

[5] A. Samorodnitsky and L. Trevisan. *Gowers Uniformity, Influence of Variables, and PCPs.* To appear in Proceedings of the 38th ACM Symposium on Theory of Computing, 2006.

ECCC ISSN 1433-8092

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