Constraint Satisfaction Problems II

Victor Vasiliev

April 6, 2016

1 MAX-SAT overview

There are two ways in which the SAT problems can be framed: satisfiability problem and optimization problem. The optimization problem asks what is the largest number of clauses we can satisfy. The satisfiability problem is a special case of the optimization problem: it asks whether

$$OPT(\phi) = n,$$

where n is the number of clauses in ϕ.

It is obvious that MAX-3SAT is NP-hard since 3SAT itself is NP-complete. 2SAT can be solved in linear time. MAX-2SAT is known to be NP-hard. However, we can attempt to approximate those.

We will consider a generalization of MAX-SAT which is called *weighted MAX-SAT*: each clause will have a weight assigned to it. Each clause ψ_i has weight w_i and the goal is to find such assignment σ that

$$W(\phi, \sigma) = \sum_{i=1}^{n} w_i \psi_i(\sigma)$$

This generalization has a potential to make the algorithm much more complicated, but as we will see, since all the tools we are going to use are linear in nature, we get this weighting basically for free. We will normalize the weights so that

$$\sum_{i=1}^{n} w_i = 1,$$

that is, the optimal value is 1 when the formula is satisfiable.

2 Randomized algorithm

Consider a random assignment of variables. For individua clause in k-SAT problem, the expected value of the clause is

$$\mathbb{E}[\psi_i] = 1 - \frac{1}{2^k},$$

since the only cause in which the clause has the value of 0 is when all variable assignments fail. By linearity of expectation, for random σ

$$\mathbb{E}\left[W(\phi, \sigma)\right] = \left(1 - \frac{1}{2^k}\right) \sum_{i=1}^{n} w_i = 1 - \frac{1}{2^k},$$

This has two important implications. The first one is that there exists a variable assignment with at least that many clauses. That is, it is always possible to satisfy at least 3/4 of a 2SAT formula, at least 7/8 of 3SAT formula, and so on. For general SAT this gives a 1/2-approximation, since each clause has to have at least one variable.

Can we do better than that? Well, for most cases, not really:

Theorem 1 (Håstad, 2001). *For any $\varepsilon > 0$, MAX-k-SAT is NP-hard to approximate within a factor better than $1 - 2^{-k} + \varepsilon$.*

This still leaves the question of MAX-2SAT and MAX-SAT in general. For MAX-2SAT specifically, it is possible to reduce the problem to max-cut and use SDP to get a 0.940-approximation (which is fairly close to the theoretical 0.954 maximum after which the problem becomes NP-hard). For MAX-SAT in general, we will see some improved techniques below.

3 Derandomized algorithm

Let us denote the expected value of the algorithm above as E_0. We know it is at least 1/2. Then pick a variable in the formula ϕ, say, X_1. Let ϕ_T be the ϕ with $X_1 = 1$ and ϕ_F be ϕ with $X_1 = 0$. Then, observe by symmetry that

$$\mathbb{E}\left[W(\phi, \sigma)\right] = \frac{1}{2}\mathbb{E}\left[W(\phi_T, \sigma)\right] + \frac{1}{2}\mathbb{E}\left[W(\phi_F, \sigma)\right]$$

That is, E_0 is the average of $\mathbb{E}\left[W(\phi_T)\right]$ and $\mathbb{E}\left[W(\phi_F)\right]$. This means that at least one of them has to be at least as big as E_0. We can compute the expected value for both of those values in linear time, using linearity of expectation. This way, we can fix X_1 to maximize the expected value of the resulting formula. Repeating this procedure n times, we get the formula where the resulting weight is at least E_0, meaning we have successfully derandomized the algorithm.

4 Merging clauses

Observe that our algorithm produces many clauses in the intermediate steps which have only one variable. We can try to be clever with those and merge them:

1. Merge all clauses which are literally the same.

2. If there are clauses $w_1 X_i$ and $w_2 \overline{X}_i$ and $w_1 > w_2$, merge them into $(w_1 - w_2) X_i$, otherwise merge them into $(w_2 - w_1) \overline{X}_i$.

3. Renormalize the weights.

Now, consider the case where we pick every variable which has its own single-variable clause, and assign probability $h > \frac{1}{2}$ to it instead of $1/2$. Then for each single-variable clause

$$\mathbb{E}\left[\psi\right] = h$$

and for multivariable case (with k variables), the probability that all assignments fail is at most h^k, hence

$$\mathbb{E}\left[\psi\right] \geq 1 - h^k \geq 1 - h^2,$$

meaning that we have to find the value of h which minimizes both h and $1 - h^2$. Because they are both monotonic functions, we get the optimum when $h = 1 - h^2$, that is,

$$h = \frac{\sqrt{5} - 1}{2} \approx 0.618$$

This improves our approximation algorithm efficiency for general MAX-SAT from 0.5.

5 ILP relaxation

We can also naturally reformulate that problem as an integer LP. Let Z_j denote whether jth clause is satisfied in the assignment, and X_i denote whether the variable is satisfied. Under the constraints that

$$Z_j \leq \sum_{i \in \psi_j^+} X_i + \sum_{i \in \psi_j^-} (1 - X_i)$$

$$0 \leq X_i \leq 1$$

$$0 \leq Z_j \leq 1$$

and goal

$$\max w \cdot Z$$

we get the ILP formulation of MAX-SAT. Naturally, ILP is NP-complete, but we can do an LP relaxation and get some vector P_i as the result. Let us randomly assign values to the resulting vector X_i using P_i as probability that the variable is true. Then

$$\mathbb{E}\left[1 - \psi\right] = \prod_{i \in \psi^+} (1 - P_i) \prod_{i \in \psi^-} P_i$$

From AM-GM inequality

$$\mathbb{E}\left[1 - \psi\right] \leq \left(\frac{\sum_{i \in \psi^+} (1 - P_i) + \sum_{i \in \psi^-} P_i}{k}\right)^k$$

3

From constraints,

$$\mathbb{E}\left[1-\psi\right] \leq \left(\frac{k-Z_\psi}{k}\right)^k = \left(1-\frac{Z_\psi}{k}\right)^k$$

$$\mathbb{E}\left[\psi\right] \geq 1 - \left(\frac{k-Z_\psi}{k}\right)^k = \left(1-\frac{Z_\psi}{k}\right)^k$$

Claim 2. *For* $Z \in [0,1]$,

$$1 - \left(1 - \frac{Z_j}{k}\right)^k \geq \left(1 - (1-k^{-1})^k\right) \cdot Z$$

Let $\beta_k = 1 - \left(1 - \frac{1}{k}\right)^k$ and $\beta = 1 - \frac{1}{e} \approx 0.63$. Note that $\beta_k \geq \beta$. Combining that with fact above, we get

$$\mathbb{E}\left[\psi\right] \geq \beta Z_\psi$$

Observe that

$$\mathbb{E}\left[W\right] \geq \sum_j w_j \mathbb{E}\left[\psi_j\right] \geq \beta \sum_j w_j Z_j \geq \beta\,\mathrm{OPT}_{LP} \geq \beta\,\mathrm{OPT}_{ILP}$$

This shows that we have a β-approximation of MAX-SAT.

6   With our algorithms combined

Consider an algorithm which flips a coin in the beginning and uses random assignment or LP relaxation depending on the outcome of the coin flip. Then

$$\mathbb{E}\left[W\right] = \frac{1}{2}\mathbb{E}\left[W_{LP}\right] + \frac{1}{2}\mathbb{E}\left[W_R\right]$$

$$= \frac{1}{2}\beta_k \sum_j w_j Z_j + \frac{1}{2} \sum_j w_j \left(1 - \frac{1}{2^k}\right)$$

$$\geq \left(\frac{1}{2}\beta_k + \frac{1}{2}\left(1 - \frac{1}{2^k}\right)\right) \sum_j w_j Z_j$$

It can be shown that for all integer values of k,

$$\frac{1}{2}\beta_k + \frac{1}{2}\left(1 - \frac{1}{2^k}\right) \geq \frac{3}{4},$$

implying that

$$\mathbb{E}\left[W\right] \geq \frac{3}{4}\,\mathrm{OPT}$$

Thus, combining a 0.618- and 0.63-approximation algorithms yields a 0.75-approximation algorithms. The intuition here is as follows: random variable assignment works well in cases when there are a lot of large clauses, and LP relaxation works well when the clauses are mostly small. When combined together, they compliment each other.