CS–E4500 Advanced Course in Algorithms

Week 04 – Tutorial

1. Consider an instance of SAT with m clauses, where every clause has exactly k literals from the tutorial sheet of Week 3. The randomized algorithm that yields an assignment that satisfies at least $m(1-2^{-k})$ clauses in expectation is simple: assign the values independently and uniformly at random. Give a derandomization of the randomized algorithm using the method of conditional expectations.

   Solution. Let $\mathcal{A}$ denote an assignment for the SAT problem. Imagine assigning the values to variables $x_1, x_2, \ldots, x_n$ deterministically, one at a time, in the enumerated order. Let a_i be the value x_i is assigned (so a_i is either 0 or 1). Suppose that we have assigned values to the first l variables, and consider the expected number of clauses we satisfy if the remaining variables are assigned 0 or 1 independently and uniformly at random. We write this quantity as $\mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_l)$; it is the conditional expectation of the number of satisfied clauses given the assignments $a_1, a_2, \ldots, a_l$ of the first l variables $x_1, x_2, \ldots, x_l$. We show inductively how to assign the value to the next variable so that

$$\mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_l) \le \mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_{l+1}) \,.$$

   It follows that

$$\mathrm{E}(\mathcal{A}) \le \mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_n) \,.$$

   The right-hand side is the number of satisfied clauses determined by our (deterministic) placement algorithm, since if $a_1, a_2, \ldots, a_n$ are all determined, we have an assignment for the SAT problem. Hence our algorithm returns an assignment which satisfies at least $\mathrm{E}(\mathcal{A}) \ge m(1-2^{-k})$ clauses.

   Both for the base case and the inductive step, consider assigning the value for x_{l+1} randomly ($0 \le l \ge n-1$), so that $a_{l+1} = 0$ and $a_{l+1} = 1$ with probability $1/2$ each, and let Y_{l+1} be a random variable representing the value. Then by the total law of expectation,

$$\begin{aligned}\mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_l) = {} & \frac{1}{2}\mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_l, Y_{l+1} = 0) \\ & + \frac{1}{2}\mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_l, Y_{l+1} = 1) \,.\end{aligned}$$

   It follows that

$$\begin{aligned}\mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_l) \le \max\{ & \mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_l, Y_{l+1} = 0), \\ & \mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_l, Y_{l+1} = 1)\} \,.\end{aligned}$$

   Therefore, all we have to do is compute the two quantities $\mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_k, Y_{l+1} = 0)$ and $\mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_k, Y_{l+1} = 1)$, and then choose value a_{l+1} which satisfies the most clauses in expectation. Once we do this, we will have an assignment satisfying

$$\mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_l) \le \mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_{l+1}) \,.$$

   To compute $\mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_l, Y_{l+1} = 0)$, note that the conditioning gives the value of the first $l+1$ variables. We can therefore compute the number of clauses these variables already satisfy. For the set of all other clauses (consisting of all unsatisfied clauses with at least one unfixed variable), which we denote as C, one can compute the expectation to be

$$\sum_{i \in C} (1 - 2^{-k_i}),$$

where k_i is the number of unfixed variables in clause $i \in C$. The same computations can be made for $\mathrm{E}(\mathcal{A} \mid a_1, a_2, \ldots, a_l, Y_{l+1} = 1)$, concluding the algorithm.