A Trivial Randomized Approximation Algorithm for MAX-E3SAT

Arjun P

5 September, 2020

The Problem

Given: 3SAT instance where each clause has exactly three literals.
Example:

$$(X \vee Y \vee Z) \wedge (X \vee Y \vee \overline{Z}) \wedge (X \vee \overline{Y} \vee Z) \wedge (X \vee \overline{Y} \vee \overline{Z}) \wedge$$
$$(X \vee Y \vee Z) \wedge (X \vee Y \vee \overline{Z}) \wedge (X \vee \overline{Y} \vee Z) \wedge (X \vee \overline{Y} \vee \overline{Z})$$

Goal: Find an assignment to satisfy as many clauses as possible.

The Algorithm

The Algorithm

Choose each variable to be true or false uniformly at random.

Analysis

- Let $C_1, \ldots C_m$ be indicator RVs, with C_i denoting whether the ith clause is satisfied.

Analysis

- Let $C_1, \ldots C_m$ be indicator RVs, with C_i denoting whether the ith clause is satisfied.
- $\mathbb{E}[C_i] = 1 - 1/2^{l_i}$, where l_i is the number of literals in the clause.

Analysis

- Let $C_1, \dots C_m$ be indicator RVs, with C_i denoting whether the ith clause is satisfied.
- $\mathbb{E}[C_i] = 1 - 1/2^{l_i}$, where l_i is the number of literals in the clause.
- Expected number of satisfied clauses is $\mathbb{E}[\sum_i C_i] = 7m/8$.

The Probabilistic Method

- Can there be an input with m clauses, such that we can't even satisfy $\lceil 7m/8 \rceil$ clauses at once?

The Probabilistic Method

- Can there be an input with m clauses, such that we can't even satisfy $\lceil 7m/8 \rceil$ clauses at once?
- Our algorithm satisfies at least $7m/8$ clauses on average.

The Probabilistic Method

- Can there be an input with m clauses, such that we can't even satisfy $\lceil 7m/8 \rceil$ clauses at once?
- Our algorithm satisfies at least $7m/8$ clauses on average.
- This would not be possible if there didn't exist at least one solution that satisfies at least $7m/8$ clauses. **Although the algorithm is randomized, our conclusion is deterministic: there always exists such a solution.**

The Probabilistic Method

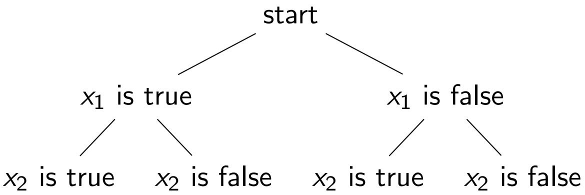
- Can there be an input with m clauses, such that we can't even satisfy $\lceil 7m/8 \rceil$ clauses at once?
- Our algorithm satisfies at least $7m/8$ clauses on average.
- This would not be possible if there didn't exist at least one solution that satisfies at least $7m/8$ clauses. **Although the algorithm is randomized, our conclusion is deterministic: there always exists such a solution.**
- Non-constructive: We don't know what the solution is, but we know one always exists.

A Deterministic Algorithm

- Consider the brute force algorithm that tries all possibilities.

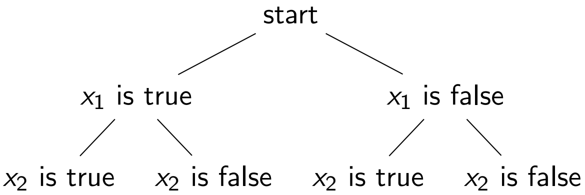
A Deterministic Algorithm

- Consider the brute force algorithm that tries all possibilities.
- At the ith level, we assign the ith variable.



A Deterministic Algorithm

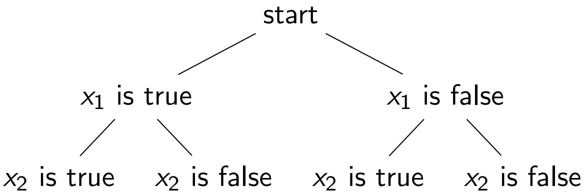
- Consider the brute force algorithm that tries all possibilities.
- At the ith level, we assign the ith variable.



- Use the conditional expectation as a heuristic to **derandomize** the algorithm.

A Deterministic Algorithm

- Consider the brute force algorithm that tries all possibilities.
- At the ith level, we assign the ith variable.



- Use the conditional expectation as a heuristic to **derandomize** the algorithm.
- **Optimal:** unless $\mathrm{P} = \mathrm{NP}$, this is the best approximation ratio a polytime algorithm can achieve.

Conclusion

- Trivial randomized algorithms can do surprisingly well.
- Randomized algorithms can give deterministic conclusions.
- Some kinds of randomized algorithms can be easily derandomized.

Bonus: E3SAT is NP-hard

- Reduction from 3SAT: given a 3SAT instance, construct equisatisfiable E3SAT instance.

Bonus: E3SAT is NP-hard

- Reduction from 3SAT: given a 3SAT instance, construct equisatisfiable E3SAT instance.
- X becomes $(X \vee T) \wedge (X \vee \overline{T})$.
- $X \vee Y$ becomes $(X \vee Y \vee T) \wedge (X \vee Y \vee \overline{T})$.

Bonus: E3SAT is NP-hard

- Reduction from 3SAT: given a 3SAT instance, construct equisatisfiable E3SAT instance.
- X becomes $(X \vee T) \wedge (X \vee \overline{T})$.
- $X \vee Y$ becomes $(X \vee Y \vee T) \wedge (X \vee Y \vee \overline{T})$.
- Why can't we use this to approximate 3SAT?