Smoothed Complexity Theory

theroyakash

Indian Institute of Technology Madras

April 20, 2023

Introduction

- Goal of the complexity theory is to understand computational difficulty of engineering problems.

Introduction

- Goal of the complexity theory is to understand computational difficulty of engineering problems.
- So we've developed theory to classify problems according to their worst case behaviour. These classes are P, NP etc.

Introduction

- Goal of the complexity theory is to understand computational difficulty of engineering problems.
- So we've developed theory to classify problems according to their worst case behaviour. These classes are P, NP etc.
- P class contains all the computational problems that in the worst case completes in polynomial time with respect to the size of the input.

Introduction

In our CS6122 Course we've already seen that real world instances for few NP-Complete problems performs **good** in terms of running time.

Introduction

Thus we must develop theory that'll classify problems of their computational difficulty **with respect to real world performance** as well. Thus we develop smooth complexity theory.

Topics we'll look into

In this presentation we'll look into the following

- Basic Definitions and assumptions, Smoothed-P Class.
  - 2-step vs 1-step model, why we are using one step models,
  - Support of the distribution, notion of $N_{x,n}$.
  - Concept of Family of Distribution
  - Definition of smoothed polynomial running time **Definition 2.1**,
  - Definition of Smoothed-P
  - **Theorem 2.3** *An algorithm A has smoothed polynomial running time if and only if there is an $\epsilon > 0$ and a polynomial p such that for all n, x, ϕ and t*

$$\Pr_{y \sim D_{n,\phi,x}} [t_A(y; n, \phi) \geq t] \leq \frac{p(n)}{t^{\epsilon}} N_{n,x} \phi$$

Topics - Continued

- Heurisitic Schemes, error less heuristic schemes in Smoothed-P.
- Notion of Reduciblity, define L_{ds}, notion of completeness.
  - Distributional problems
  - Polynomial time smoothed reductions $\leq_{smoothed}$
  - Transitivity of $\leq_{smoothed}$ [via theorem 3.4]
  - Theorem 3.5 $(L, D) \in \textsf{Smoothed-P}$ if and only if $(L_{ds}, D) \in \textsf{Smoothed-P}_{ds}^{\textsf{obl}}$

Topics - Continued

- Parameterized Distributional NP $\mathsf{Dist\text{-}NP}_{\mathrm{para}}$
  - Introduction
  - One $\mathsf{Dist\text{-}NP}_{\mathrm{para}}$ complete problem to show $\mathsf{Dist\text{-}NP}_{\mathrm{para}}$ has complete problems
  - **Weird looking theorem** $(\mathit{Tiling}, U^{\mathit{Tiling}})$ *is* $\mathit{Dist\text{-}NP}_{para}$*-complete for some* $U^{\mathit{Tiling}} \in \mathit{PComp}_{para}$ *under polynomial time smoothed reductions.*
- Basic Relation to Worst case complexity
- Notion of unsatisfiability and $\mathsf{Smoothed\text{-}RP}$
- Theorem $k\mathsf{UNSAT}_{\beta} \in \mathsf{Smoothed\text{-}RP}$ for $\beta = \Omega(\sqrt{n \log \log n})$
- Concluding remarks

Basic Definitions

- In numerical problems it is easy to let an adversary come up with worst case example and then perturb the instance via some distribution (for example Gaussian Perturbation).

Basic Definitions

- In numerical problems it is easy to let an adversary come up with worst case example and then perturb the instance via some distribution (for example Gaussian Perturbation).
- However for general problems this model can not be used.

Basic Definitions

- In numerical problems it is easy to let an adversary come up with worst case example and then perturb the instance via some distribution (for example Gaussian Perturbation).
- However for general problems this model can not be used.
- From Beier Vöcking's model we'll let an adversary choose the whole probability distribution. Let's define the model more formally,

Beier Vöcking's One Step Model

Any input X of length n with $X = (x_1, \dots, x_n) \in F^n$ where F is the domain, with parameter ϕ and an adversary who chooses density functions bounded by ϕ as $\{f_1, \dots, f_n\}$ such that $f_i : F \to [0, \phi]$, an algorithm $\mathcal{A}$'s smoothed performance measure given by the following

$$\textit{smoothed performance } (\mathcal{A}) = \underset{X=(x_1,\dots,x_n),\, x_i \sim f_i : D_{n,x,\phi}}{\mathbb{E}} [\mathcal{A}(X)]$$

Formal Definition of the Model

Our perturbation models are families of distribution $\mathcal{D} = (D_{n,\phi,x})$ where n is the size of the input x, and ϕ is the upper bound on the maximum density of the probability distributions.

Some Properties of Distribution

For every n, x, ϕ the support of the distribution should be of size $\{0,1\}^{\leq \text{poly}(n)}$.

$N_{n,x}$ and $S_{n,x}$

We define $N_{n,x}$ and $S_{n,x}$ here.

$N_{n,x}$ and $S_{n,x}$

$$S_{n,x} = \{y \mid D_{n,x,\phi}(y) > 0 \text{ for some } \phi\}$$
$$N_{n,x} = |S_{n,x}|$$

Some Properties of Distribution

- We say $\mathcal{D}$ is parameterized by n, ϕ, x.

Some Properties of Distribution

- We say $\mathcal{D}$ is parameterized by n, ϕ, x.
- For all n, ϕ, x and y we demand $D_{n,\phi,x}(y) \leq \phi$, We choose $\phi \in [\frac{1}{N_{n,x}}, 1]$ and n is the size of input x.

Some Properties of Distribution

- We say $\mathcal{D}$ is parameterized by n, ϕ, x.
- For all n, ϕ, x and y we demand $D_{n,\phi,x}(y) \leq \phi$, We choose $\phi \in [\frac{1}{N_{n,x}}, 1]$ and n is the size of input x.
- Choice of ϕ determines the strength of perturbation.

Some Properties of Distribution

- We say $\mathcal{D}$ is parameterized by n, ϕ, x.
- For all n, ϕ, x and y we demand $D_{n,\phi,x}(y) \leq \phi$, We choose $\phi \in [\frac{1}{N_{n,x}}, 1]$ and n is the size of input x.
- Choice of ϕ determines the strength of perturbation.
- If we choose $\phi = 1$ this corrsoponds to worst case complexity and setting $\phi = \frac{1}{N_{x,\phi}}$ is average case complexity.

Some Properties of Distribution

- We say $\mathcal{D}$ is parameterized by n, ϕ, x.
- For all n, ϕ, x and y we demand $D_{n,\phi,x}(y) \leq \phi$, We choose $\phi \in [\frac{1}{N_{n,x}}, 1]$ and n is the size of input x.
- Choice of ϕ determines the strength of perturbation.
- If we choose $\phi = 1$ this corrsoponds to worst case complexity and setting $\phi = \frac{1}{N_{x,\phi}}$ is average case complexity.
- The choice of ϕ must be discretized such that it can be represented within polynomial many bits.

Smoothed Polynomial Running Time

Definition 1 *An algorithm $\mathcal{A}$ has smoothed polynomial running time with respect to the distribution family $\mathcal{D}$ if there exists an $\epsilon > 0$ such that, for all n, ϕ, x, we have*

$$\mathbb{E}_{y \sim D_{n,x,\phi}} \left(t_A(y; n, \phi)^{\epsilon} \right) = O(nN_{n,x}\phi)$$

Analysing the Definition

Note that the up-above result do not speak about the expected running time, it takes into account for the ϵ moment of the expected running time.

Analysing the Definition

This is because the expected running time is not robust.

Important Results - Complexity Class Smoothed-P

- In classical complexity theory we only consider decision problems,

Complexity Class Smoothed-P

- In classical complexity theory we only consider decision problems,
- In average case complexity we consider a decision problem along with a distribution D

Complexity Class Smoothed-P

- In classical complexity theory we only consider decision problems,
- In average case complexity we consider a decision problem along with a distribution D
- Similary here for smoothed complexity we'll consider a decision problem L along with a distribution D where $L \subseteq \{0,1\}^*$

Complexity Class Smoothed-P

Smoothed-P is the class of all $(\mathcal{L}, \mathcal{D})$ such that there is a deterministic algorithm $\mathcal{A}$ with smoothed polynomial running time that decides $\mathcal{L}$.

Complexity Class Smoothed-P

Theorem 1 *An algorithm $\mathcal{A}$ has smoothed polynomial running time if and only if there is an $\epsilon > 0$ and a polynomial p such that for all n, ϕ, x and t*

$$\Pr_{y \sim D_{n,\phi,x}} [t_A(y; n, \phi) \geq t] \leq \frac{p(n)}{t^{\epsilon}} N_{n,x}\phi$$

Proof of Theorem 1

($\implies$) Forward Direction
Let A be an algorithm whoose running time t_A fulfills Definition 1:

$$\mathbb{E}_{y \sim D_{n,x,\phi}} (t_A(y; n, \phi)^{\epsilon}) = O(nN_{n,x}\phi)$$

Via Markov's inequality we can say that

$$\begin{aligned} \Pr[t_A(y; n, \phi) \geq t] &= \Pr[t_A(y; n, \phi)^{\epsilon} \geq t^{\epsilon}] \\ &\geq \frac{\mathbb{E}_{y \sim D_{n,x,\phi}} (t_A(y; n, \phi)^{\epsilon})}{t^{\epsilon}} = O(nN_{n,x}\phi t^{-\epsilon}) \end{aligned}$$

Proof Continued

($\Longleftarrow$) Backward Direction
Assume that

$$\Pr_{y\sim D_{n,\phi,x}}[t_A(y;n,\phi)\geq t]\leq \frac{n^c}{t^\epsilon}N_{n,x}\phi$$

for some constant c,ϵ. Let $\epsilon'=\frac{\epsilon}{c+2}$. Then we have

$$\begin{aligned}\mathbb{E}_{y\sim D_{n,x,\phi}}\left(t_A(y;n,\phi)^{\epsilon'}\right) &= \sum_t \Pr\left[\left(t_A(y;n,\phi)^{\epsilon'}\right)\geq t\right]\\ &\leq n+\sum_{t\geq n}\Pr\left[(t_A(y;n,\phi))\geq t^{\frac{1}{\epsilon'}}\right]\\ &\leq n+\sum_{t\geq n} t^{-2}N_{n,x}\phi = n+O(N_{n,x}\phi)=O(nN_{n,x}\phi)\end{aligned}$$

Heuristic Schemes

A different way to think about efficiency in the smoothed setting is via Heuristic Schemes.

Heuristic Schemes

The notion of Heuristic Schemes comes from the observation that we might be able to run the algorithm for polynomially many steps and if it does not succeed within that time it will return failure.

Heuristic Schemes

Definition 2 *Let* $(\mathcal{L}, \mathcal{D})$ *be a smoothed distributional problem. An algorithm* A *is an error less Heuristic Scheme for* $(\mathcal{L}, \mathcal{D})$ *if there exists a polynomial* q *such that*

- *For every* $n, x, \phi, \delta > 0, y \in suppD_{n,x,\phi}$ *we have* $A(y; n, \phi, \delta)$ *outputs either* $\mathcal{L}(y)$ *or* $\perp$*,*
- *For every* $n, x, \phi, \delta > 0, y \in suppD_{n,x,\phi}$ *we have* $t_A(y; n, \delta) \leq q(n, N_{n,x}, \phi, \frac{1}{\delta})$*,*
- *For every* $n, x, \phi, \delta > 0, y \in suppD_{n,x,\phi}$ *we have* $\Pr_{y \sim D_{n,x,\phi}}[A(y; n, \phi, \delta) = \perp] \leq \delta$*.*

Smoothed-P with respect to Heuristic Schemes

With the definition from the last slide for Heuristic Scheme,
We state the following theorem

Theorem: Heuristic Schemes and Smoothed-P

Theorem 2: $(\mathcal{L}, \mathcal{D}) \in \textsf{Smoothed-P}$ *if and only if* $(\mathcal{L}, \mathcal{D})$ *has an error less Heuristic Scheme* $\mathcal{H}$.

Proof of Theorem 2

($\implies$) Forward Direction,
Let $\mathcal{A}$ be an algorithm for $(\mathcal{L}, \mathcal{D})$. By Theorem 1 we can say

$$\Pr_{y \sim D_{n,\phi,x}} [t_A(y; n, \phi) \geq t] = O(nN_{n,x}\phi t^{-\epsilon})$$

Now all is left, is to construct $\mathcal{H}$.

Constuction of $\mathcal{H}$

Algorithm 1: $\mathcal{H}$

Run algorithm A for $(n \cdot \frac{N_{n,x} \cdot \phi}{\delta})^{\frac{1}{\epsilon}}$ steps.

if *If A stops within this many steps* **then**

  Output whatever A Outputs;

else

  Output $\perp$.

Properties of $\mathcal{H}$

- By the choice of the parameter $t = (n \cdot \frac{N_{n,x} \cdot \phi}{\delta})^{\frac{1}{\epsilon}}$ probability that $\mathcal{H}$ outputs $\perp$ is at most δ

Properties of $\mathcal{H}$

- By the choice of the parameter $t = (n \cdot \frac{N_{n,x} \cdot \phi}{\delta})^{\frac{1}{\epsilon}}$ probability that $\mathcal{H}$ outputs $\perp$ is at most δ
- $\mathcal{H}$ is correct and is a Heuristic Scheme according to the Definition 2.

Proof: Continued

($\Longleftarrow$) Backward Direction For the other direction suppose we have $\mathcal{H}$ an errorless heuristic scheme, we need to find a smoothed polynomial time algorithm $\mathcal{A}$.

Constuction of $\mathcal{A}$

Algorithm 2: Construction of Algorithm $\mathcal{A}$

i $\leftarrow$ 1;
while *True* **do**
  Run $\mathcal{H}$ with $\delta = \frac{1}{2^i}$;
  if $\mathcal{H}$ *does not output* $\perp$ **then**
    return whatever $\mathcal{H}$ says.
  i $\leftarrow$ i + 1;

Analysis of Algorithm $\mathcal{A}$

Heuristic Scheme $\mathcal{H}$ will eventually stop at some i with delta being set to $\frac{1}{2^i}$.
Then from **Definition 2** $\exists$ a polynomial q such that

$$\begin{aligned} t_{\mathcal{H}} &\leq \sum_{j=1}^{i} q(n, N_{n,x}\phi, 2^j) \\ &\leq \text{Poly}(n, N_{n,x}\phi) \cdot 2^{ci} \end{aligned}$$

Heuristic Scheme $\mathcal{H}$ will eventually stop when $\delta < D_{n,x,\phi}(y)$ from definition of Heuristic Scheme. Thus Algorithm $\mathcal{A}$ has smoothed polynomial time algorithm (it has a pseudo polynomial time algorithm).

Disjoint Support

Let's define pair $\langle x, y \rangle$ as "y was drawn according to $D_{n,x,\phi}$". For a parameterized distributional problem $(\mathcal{L}, \mathcal{D})$ we define

$$L_{\text{ds}} = \{\langle x, y \rangle \mid y \in \mathcal{L} \text{ and } |\, y \,| \leq \text{poly}(|\, x \,|)\}$$

Disjoint Support Continued, Reducibility

With this notion of L_{ds} we define the notion of reducibility.