CS6122 - Probabilistic and smoothed analysis of Algorithms

Lecture 24: Introduction to smoothed analysis

March 11, 2023

Lecturer: Raghavendra Rao, B. V. *Scribe: Roy, Akash*

Topics: Introduction to smoothed analysis, smoothed analysis of 2-OPT algorithm.

1 Introduction to smoothed analysis

In the last class we discussed why worst-case and average case analysis fails to analyse the behaviour of some *algorithms* run on real-world data. We'll do smoothed analysis in more detail on TSP, Knapsac, Network Flows, KMeans and also on some data structures like Binary Search Tree.

We'll discuss TSP's 2-OPT algorithm, partitioning algorithm, and at the end make a general framework (quantaization) to analyze smoothed analysis. We'll discuss basic models of Smoothed Analysis Techniques below.

2 Models of Smoothed Analysis

2.1 Two step model

Definition 2.1: 2-Step Model

Any input domain $\mathcal{I}$ with parameter π and some $x \in \mathcal{I}$ with $N_\pi(x)$ being a neighborhood or a distribution, then for algorithm $\mathcal{A}$ smoothed performance of $\mathcal{A}$ is defined as following

$$\textit{smoothed performance } (A) = \max_{x \in \mathcal{I}} \quad \mathbb{E}_{Y \sim N_\pi(x)}[A(Y)]$$

The model is called 2-step model because in step 1, adversary chooses X with this $N_\pi(X)$ get's choosen. After that as defined neighborhood we take the expectation of the performance of the algorithm. In short adversary chooses an arbitrary instance, then the instance are perturbed with random variables from any specific distribution. The smoothed performance is the maximum expected performance over all the adversarial choices.

2.2 One step model of Beier and Vöcking

Definition 2.2: One Step Model

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

$$\textit{smoothed performance } (A) = \underset{X=(x_1,\ldots,x_n),\, x_i\sim f_i\ \textit{independently at random}}{\mathbb{E}}[A(X)]$$

Here adversary specifies probability density functions for each of the $x_i \in X$ then all numbers are drawn independently according to their respective density function. This one step model is equivalent to two step model, it also finds out the maximum expected performance. Adversary implicitly chooses maximum over all the points then expectation is taken.

3 Analysis of 2-OPT algorithm for TSP

Let's recall the 2-OPT algorithm for TSP.

```
Algorithm 1: 2-OPT Algorithm
  Input: Graph $\mathbb{G}$
  Start with some arbitrary tour $\tau$
  Perform 2-exchanges of $\tau$ until no more improvement is possible.
```

2-OPT algorithm is unlikely to find the most optimal TSP tour, but will find a local minimum such that doing 2 exchanges no longer reduce the tour cost. It is proven in the worst case this runs in exponential time $O(2^n)$ with worst case approximation ratio $\frac{\lg n}{\lg \lg n}$. But this worst case analysis do not reflect the widespread real world usage of this algorithm on the real world data. In the coming parts we'll try to understand what's going on.
Let's look at the following theorem.

Theorem 3.1: Longest Path Theorem

Expected length of the longest path in any 2-OPT configuration graph is bounded by

- $O(n^4\phi)$ for ϕ perturbed L_1 metric instances.
- $O(n^{4+\frac{1}{3}}\phi^{\frac{8}{3}}\lg n\phi)$ for ϕ perturbed L_2 metric instances.
- $O(m^{1+\epsilon}n\phi)$ for ϕ perturbed graph with n vertices and m edges.

Let's understand what the theorem is saying. First let's define 2-OPT configuration graph and ϕ perturbed graph.

Definition 3.1: 2-OPT configuration graph

2-OPT configuration graph is graph with $n!$ many vertices each representing one permutation of tour of the main graph, and this configuration graph has edge (u, v) if and only if from tour u, tour v is one 2-exchange away.

Definition 3.2: ϕ perturbed graph

A ϕ perturbed graph is a graph with n vertices and m edges with weight function $w : E \to [0, 1]$, such that $\forall\ e \in E\ \exists\ f_e : [0, 1] \to [0, \phi]$ and wt.(e) is drawn according to f_e.

The runtime of our 2-OPT algorithm will depend upon the longest path in the 2-OPT configuration tree by intuition. Our theorem 3.1 captures the expected size of such longest paths. This will upperbound our running time for the 2-OPT algorithm. We'll prove a weaker version of theorem 3.1 for ϕ perturbed graph with n vertices and m edges.

Proof. Suppose G is a fixed graph on n vertices and m edges.
Weight function $w : E \to [0, 1]$ be ϕ perturbed weights such that $w(e) \sim f_e : [0, 1] \to [0, \phi]$.
Consider any 2-edges e_1 and e_2. Suppose after performing a 2-exchange the cost of the tour reduces and the new edges are e_3 and e_4. $(e_i)_{i \in 1,2,3,4}$ creates a four-cycle.
Say the change in weight is $\Delta(e_1, \ldots, e_4)$. We define the following

$$\Delta(e_1, \ldots, e_4) = \sum_{i \in 1,2} e_i - \sum_{i \in 3,4} e_i$$

Let Δ be minimum across all such 4-cycles $\Delta = \min_{(e_1,\ldots,e_4),\, \Delta(e_1,\ldots,e_4) \geq 0} \Delta(e_1, \ldots, e_4)$. Then the running time of the algorithm is $O(\frac{n}{\Delta})$.

Lemma 3.1: Δ do not have large values

Δ do not have large values is captured by the following probability measure.

$$\forall\, \epsilon > 0\ \exists\, \Pr[\Delta \leq \epsilon] \leq m^2 \epsilon \phi$$