Course: Comtemporary Algorithms T.II/2019-20

# Lecture 3: Nearest Neighbours I

13 January 2020

*Lecturer: Dr. Kanat Tangwongsan* *Scribe: Kriangsak T. & Apivich H.*

In the next couple of lectures, we will discuss algorithms for the Nearest Neighbours problem, commonly used in data science and those fancy stuff. However, we first need to discuss some statistics which will be useful in the future lectures.

## 1 Concentration of Measure

To explain the basic idea of concentration of measure, we can start with an example of a fair coin, where you may get heads or tails with an equal probability.

If you toss a fair coin once, it is very hard to determine how many times you will get a heads. In other words, the results is *unpredictable*.

However, if you toss a fair coin 1000 times, you may be able to predict that you will get around 500 heads. The outcome is now more *predictable*.

As you toss more coins, the outcome of the event will be more concentrated around the expected value. In this case, the probability distribution becomes “sharper" and you can be more sure about your predictions.

Some may have seen this phenomena be described as the “Law of Large Numbers" or as the “Central Limit Theorem". However, this only applies in the limit of when the number of events approaches infinity. We would like to be more concrete.

The question we would like to answer is: *Given a random variable $X$ with mean $\mu = \mathbb{E}[X]$, what is the probability that $X$ deviates far from $\mu$?*

### 1.1 Markov’s Inequality

**Theorem 1.1** (Markov’s Inequality)**.** *If $X$ is a non-negative random variable, then, for all $\lambda > 0$,*

$$\mathbb{P}[X > \lambda] \leq \frac{\mathbb{E}[X]}{\lambda}.$$

*Proof.* Since we assume that $X > 0$,

$$\begin{aligned}
\mathbb{E}[X] &= \int_0^\infty xp(x)dx \\
&= \int_0^\lambda xp(x)dx + \int_\lambda^\infty xp(x)dx && \text{by splitting the integral} \\
&\geq \int_\lambda^\infty xp(x)dx \\
&\geq \int_\lambda^\infty \lambda p(x)dx && \text{since under this integral limit } x \geq \lambda \\
&= \lambda \int_\lambda^\infty p(x)dx && \text{since } \lambda \text{ is a constant} \\
&= \lambda \cdot \mathbb{P}[X > \lambda]
\end{aligned}$$

then one can rearrange the above to arrive at $\mathbb{P}[X > \lambda] \leq \mathbb{E}[X]/\lambda$. □

Note that Markov's Inequality is a more general expression, as it makes little assumption on what the actual probability distribution of $X$ actually looks like. As a result, the bound obtained is not as tight as what we will get from Chernov-Hoeffding Bounds in the next section.

Another result that we can prove from Markov's Inequality is *Chebychev's Inequality*. Suppose we define another random variable $Y = (x - \mu)^2$ where $\mu = \mathbb{E}[X]$. We can see that $\mathbb{E}[Y] = \text{Var}[X]$. Since $Y \geq 0$, we can use Markov's Inequality on it. Then, we can see that for some $t > 0$,

$$\begin{aligned}
\mathbb{P}[Y > t^2] &< \frac{\mathbb{E}[Y]}{t^2} && \text{by Markov's Inequality} \\
\mathbb{P}[(X - \mu)^2 > t^2] &< \frac{\text{Var}[X]}{t^2} \\
\mathbb{P}[|X - \mu| > t] &< \frac{\text{Var}[X]}{t^2} && \text{since } (X - \mu)^2 > t^2 \iff |X - \mu| > t
\end{aligned}$$

where the last line is the Chebychev's Inequality. Again, for this result, we have made no assumption about what the probability distribution of $X$ is like (although we do need to be able to find the variance of $X$).

## 1.2 Chernov-Hoeffding Bounds

**Theorem 1.2** (Chernoff-Hoeffding Bounds)**.** *Let $X = \sum_{i=1}^n X_i$ where each $X_i$'s are random variables which are all independently distributed in $[0, 1]$. Let $\mu = \mathbb{E}[X]$. Then,*

1. *For all $t > 0$, $\mathbb{P}[X > \mu + t]$ and $\mathbb{P}[X < \mu - t]$ is at most $\exp(-2t^2/n)$,*

2. *For all* $\lambda > 0$,

$$\mathbb{P}[X > (1+\lambda)\mu] \leq \exp\left\{-\frac{\lambda^2\mu}{3}\right\},$$

*and*

$$\mathbb{P}[X < (1-\lambda)\mu] \leq \exp\left\{-\frac{\lambda^2\mu}{2}\right\}.$$

The theorem will be stated without proof. However, in order to prove this result, a trick which will be needed is something called a *moment-generating function* of $X$. This is defined to be $\mathbb{E}[e^{tX}]$. When this is expanded out, we can see that

$$\mathbb{E}[e^{tX}] = \mathbb{E}\left[\sum_{k\geq 0}\frac{(tX)^k}{k!}\right] = \sum_{k\geq 0}\frac{t^k}{k!}\mathbb{E}[X^k]$$

which is a function that contains $\mathbb{E}[X^k]$ for all $k \geq 0$. These terms are the moments of a random variable. The proof of the Chernov-Hoeffding bound will begin by saying that $\mathbb{P}[X > \lambda] = \mathbb{P}[e^{tX} > e^{t\lambda}] < e^{-t\lambda}\mathbb{P}[e^{tX}]$ followed by a lot of maths which will be skipped.

## 1.3 Examples Using Chernov-Hoeffding Bounds

### 1.3.1 Skip Lists

The idea of skip lists has been discussed in a previous lecture. Here, we will prove that searching for a key in a skip-list of size $n$ takes $O(\lg n)$ time with high probability using Chernov-Hoeffding Bounds.

**Lemma 1.3.** *Searching for a key in a skip-list of size* $n$ *takes* $O(\lg n)$ *time w.h.p.*

*Proof.* Let $Z$ be the number of steps needed to take to reach a certain key in a skip-list. We can see that there are two types of steps in the search traversal, either you move across (in the same level of the skip-list), or you move down (to a lower level of the skip-list). Let the number of steps moving across be $H$ and the number of steps moving down be $V$. By our definition, $Z = H + V$.

We want to know what is the probability that $Z$ is less than $c\lg n$, for some sufficiently large $c$. In mathematical notation, we want to find out what is $\mathbb{P}[Z \leq c\lg n]$. This is the probability of some “good" event occurring (since it is the event where we don’t take too many steps in the traversal).

We can see that $\mathbb{P}[Z \leq c\lg n]$ is actually equal to $1 - \mathbb{P}[Z > c\lg n]$. The term $\mathbb{P}[Z > c\lg n]$ can be approximated. We will use the fact that

$$\mathbb{P}[B] = \mathbb{P}[B|A]\mathbb{P}[A] + \mathbb{P}[B|\bar{A}]\mathbb{P}[\bar{A}]$$

to rewrite our expression as

$$\mathbb{P}[Z > c\lg n] = \mathbb{P}[Z > c\lg n|V \geq k]\mathbb{P}[V \geq k] + \mathbb{P}[Z > c\lg n|V < k]\mathbb{P}[V < k].$$

Since the probability of all events must be less than one, we can see that $\mathbb{P}[Z > c\lg n | V \geq k] \leq 1$ and $\mathbb{P}[V < k] \leq 1$. We can reduce the expression above further to

$$\mathbb{P}[Z > c\lg n] \leq \mathbb{P}[V \geq k] + \mathbb{P}[Z > c\lg n | V < k].$$

From a previous lecture, we have shown that $\mathbb{P}[V \geq k] \leq 1/n^{k-1}$. This leaves the second expression. If we have $V < k$, then $Z > c\lg n$ implies that $H = Z - V > c\lg n - k$. We can therefore say that

$$\begin{aligned}\mathbb{P}[Z > c\lg n | V < k] &\leq \mathbb{P}[H > c\lg n - k] \\ &\leq \exp\left\{-\frac{2k^2}{c\lg n}\right\}\end{aligned}$$

where we have used the Chernov-Hoeffding Bounds to get from the first line to the second. We can see that if we pick some small enough $k$ (maybe $k = c\lg n/10$ for example), then we can say that $\mathbb{P}[Z > c\lg n | V < k] \leq \exp\{-\epsilon \lg n\} = 1/n^\epsilon$ for some $\epsilon > 0$. We can therefore see that

$$\mathbb{P}[Z > c\lg n] = \frac{1}{n^{k-1}} + \frac{1}{n^\epsilon} \leq \frac{1}{n^{(\text{some constant})}}$$

which means that

$$\mathbb{P}[Z \geq c\lg n] \geq 1 - \frac{1}{n^{(\text{some constant})}}$$

which is sufficient to show that our skip list takes $O(\lg n)$ steps to search for a key *w.h.p.* □

### 1.3.2 Height of Treap

In the previous lecture, we have proven that the height of a treap to be fairly balanced with the expected depth of $\lg n$. Here is another derivation of the height of a treap using Chernoff-Hoeffding Bounds.

**Lemma 1.4.** *The height of a treap of size $n$ is $O(\lg n)$ time w.h.p.*

*Proof.* Let $A_{i,j}$ be an indicator random variable where,

$$A_{i,j} = \begin{cases} 1 & \text{if } j \text{ is an ancestor of } i \\ 0 & \text{otherwise} \end{cases}$$

Note that for a fixed $i$, all $A_{i,j}$'s are independent, meaning Chernoff-Hoeffding bounds apply.

In this case, the depth of the treap is,

$$\begin{aligned}\text{depth}(X_i) &= \sum_{j=1}^{n} A_{i,j} \\ &= 1 + \underbrace{\sum_{j=1}^{i-1} A_{i,j}}_{L} + \underbrace{\sum_{j=i+1}^{n} A_{i,j}}_{R}\end{aligned}$$

To prove our claim above, we have to show that

$$\mathbb{P}[\text{depth} \le k \ln n\,] = 1 - \underbrace{\mathbb{P}[\text{depth} > k \ln n\,]}_{*} \ge 1 - \frac{1}{n^\alpha}$$

where the term $\frac{1}{n^\alpha}$ is the error probability mentioned in the previous lecture. We can bound $(*)$ using union bound. Union bound (a.k.a. Boole's Inequality) states that for a set of events $A_1, A_2, \ldots$,

$$\mathbb{P}\left(\bigcup_i A_i\right) \le \sum_i \mathbb{P}\left(A_i\right).$$

In our problem, we can let $A_1$ be the event when $L > k \ln n$ and $A_2$ be the event when $R > k \ln n$, so that $A_1 \cup A_2$ be the event when either $L > k \ln n$ or $R > k \ln n$ (hence depth $> k \ln n$). From here, we show that

$$\mathbb{P}[\text{ depth} > k \ln n\,] \le \mathbb{P}[L > k \ln n\,] + \mathbb{P}[R > k \ln n\,].$$

Applying the powerful Chernoff-Hoeffding bounds, we then have

$$\begin{aligned}\mathbb{P}[L > (1 + k - 1) \ln n\,] &\le \exp\{-\frac{(k-1)^2}{3} \times \underbrace{2 \ln n}_{**}\} \\ &\le \frac{1}{n^\alpha} \quad \text{where } \alpha = \frac{(k-1)^2}{3}.\end{aligned}$$

Note that $(**)$ is bounded by the deepest height of a treap. The same taken to the second term, we will have

$$\mathbb{P}[\text{depth} \le k \ln n] \ge 1 - \frac{2}{n^\alpha} \ge 1 - \frac{1}{n^{\text{(some constant)}}}.$$

□

# 2 Closest Pairs Problem

In this last bit of the lecture, we will explore the a simple idea whereby the space is sub-divided into grid cells. By doing so, it turns out to be extremely powerful.

## 2.1 Closest Pair in 2D

Given the input as a set ***P*** of *n* points in 2D, and the goal is to compute the closest pair of points. There are a few ways to solve such a problem. One may try to go about writing two for-loops but of course the algorithm delivers with $O(n^2)$. Another approach to this problem is to solve it using divide-and-conquer yielding $O(n \log n)$ running time. However, there is a faster, and simpler, algorithm that delivers the closest pair in $O(n)$, with allowance of additional operations.

## 2.2 Verification Problem

Let $CP(Q)$ denote the distance between the closest pair of points of $Q$, where $Q$ is a set of points. If someone claims $CP(Q)$ is *r*. How do we verify this quickly?

We can in fact do this in linear time:

- Build a grid of size *r* by *r*
- Dump all the points into the cells by bucketing/hashing. In fact, $p = (x, y)$ goes to the coordinate ($\lfloor x/r \rfloor$,$\lfloor y/r \rfloor$).
- If any cell has more than 9 points, $CP(Q) < r$. Note that if the cell is sub-divided into a 3-by-3 sub-cells, Pigeonhole Principle says one of them has at least two points, but then this sub-cell's diameter is strictly less than $r$

For each operation, we will look at at most 81 points, that is, we will be looking at the neighboring sub-cells plus itself, each containing at most 9 points. The amount of time spent on each point is therefore constant time, and so the algorithm overall will take linear time.

## 2.3 A Closest Pair Algorithm

The algorithm will be an edited version of the verification algorithm. It will as follows.

1. Permute the given input set ***P*** of points randomly, say ***P*** = $\langle p_1, p_2, ..., p_n \rangle$ is a permutation of the given input.
2. start with $r_2 = ||p_1 - p_2||_2$. In this case, the above grid will be sub-divided into an $r$-by-$r$ grid where $r = r_2$
3. $\forall i \in \{3, 4, 5, ..., n\}, p_i$ will be added . When added, there algotithm shall:
   - Check whether $p_i$ forms a new closest pair.This requires the checking of $p_i$ to its original and other 8 neighboring cells, all of which contain at most 9 points.
   - If $p_i$ becomes the new closest pair, rebuild the grid using the new closest pair distance.
   - Continue with this grid otherwise.

We claim that te algorithm above takes linear time. To show this, let $X_i$ be a indicator random variable where

$$X_i = \begin{cases} 1 & \text{if } p_i \text{ forms a new closest pair} \\ 0 & \text{otherwise} \end{cases}$$

The total cost is then given by the resurrance,

$$T(n) = O(1) + \sum_{i=3}^{n} (O(1) + X_i \cdot i)$$

We claim that $\mathbb{P}[X_i = 1] \leq \frac{2}{i}$, where $p_i$ is randomly drawn from the permuted input. The probability of a point randomly drawn forms the new closest pair is given by two cases:

- The closest pair is unique and they are x–y. Then either one of them will be picked as $p_i$ with $2/i$ probability.
- The closest pair is not unique, then the probability that $p_i$ becomes the *new* closest pair is 0.

Hence, the expected running time is then

$$\begin{aligned}
\mathbb{E}[T(n)] &\leq O(1) + n + \sum_{i=3}^{n} [\underbrace{X_i}_{2/i} \cdot i] \\
&= O(n) + \sum_{i=3}^{n} [\frac{2}{i} \cdot i] \\
&= O(n).
\end{aligned}$$