CS5234 - Algorithms at Scale

Liew Zhao Wei

Semester 1, 2023-2024

---

1 Probability and Hashing

1.1 Probability

We start by stating some crucial probability results that will be used throughout the course.

Lemma 1.1 (Union Bound)
For a countable set of events $A_1, A_2, \ldots$, we have

$$\Pr\left[\bigcup_{i=1}^{\infty} A_i\right] \leq \sum_{i=1}^{\infty} \Pr[A_i] \tag{1}$$

Lemma 1.2 (Linearity of Expectation)
For any random variables $X_1, X_2, \ldots, X_n$, we have

$$\mathrm{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathrm{E}[X_i] \tag{2}$$

Lemma 1.3 (Markov's Inequality)
For any *non-negative* random variable X and $t > 0$, we have

$$\Pr[X \geq t] \leq \frac{\mathrm{E}[X]}{t} \tag{3}$$

Lemma 1.4 (Chebyshev's Inequality)
For any random variable X with mean μ and variance σ^2, we have

$$\Pr[|X - \mu| \geq t] \leq \frac{\sigma^2}{t^2} \tag{4}$$

In fact, this holds for any moment p instead of 2.

Lemma 1.5 (Chernoff-Hoeffding Bounds)
Suppose $X_1, \ldots, X_n$ are *independent* random variables with $X_i \in [0,1]$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathrm{E}[X]$ such that $\mu_L \leq \mu \leq \mu_H$.

(Chernoff) For any $0 \leq \delta \leq 1$,

$$\Pr[X \geq (1+\delta)\mu] \leq \exp\left\{-\frac{\delta^2\mu}{3}\right\} \tag{5}$$

$$\Pr[X \leq (1-\delta)\mu] \leq \exp\left\{-\frac{\delta^2\mu}{2}\right\} \tag{6}$$

$$\Pr\left[|X-\mu| \geq \delta\mu\right] \leq 2\exp\left\{-\frac{\delta^2\mu}{3}\right\} \tag{7}$$

$$\tag{8}$$

(Hoeffding) For any $\delta \geq 0$,

$$\Pr[X \geq \mu + \delta] \leq \exp\left\{-\frac{2\delta^2}{n}\right\} \tag{9}$$

$$\Pr[X \leq \mu - \delta] \leq \exp\left\{-\frac{2\delta^2}{n}\right\} \tag{10}$$

$$\Pr\left[|X-\mu| \geq \delta\right] \leq 2\exp\left\{-\frac{2\delta^2}{n}\right\} \tag{11}$$

More generally, if $a_i \leq X_i \leq b_i$, then

$$\Pr[X \geq \mu + \delta] \leq \exp\left\{-\frac{2\delta^2}{\sum_{i=1}^{n}(b_i-a_i)^2}\right\} \tag{12}$$

$$\Pr[X \leq \mu - \delta] \leq \exp\left\{-\frac{2\delta^2}{\sum_{i=1}^{n}(b_i-a_i)^2}\right\} \tag{13}$$

1.2 Hashing

Definition 1.6 (k-Universal Hash)
A hash family $\mathcal{H} = \{h : \mathcal{U} \to S\}$ is *k-universal* if for any distinct k elements $x_1, \ldots, x_k \in \mathcal{U}$ and any k elements $y_1, \ldots, y_k \in S$, we have

$$\Pr_{h \in_R \mathcal{H}}[h(x_1) = y_1 \wedge \ldots \wedge h(x_k) = y_k] = \frac{1}{|S|^k} \tag{14}$$

Such a hash family is also called a *k-wise independent hash family.*

Lemma 1.7 (Construction of k-Universal Hash Family)
There is a construction of a k-universal hash family from $[n]$ to $[n]$ that takes $O(k \log n)$ space.

2 Simple Techniques

2.1 Reservoir Sampling

We can use *reservoir sampling* to uniformly sample an element from a stream without having to store the stream in advance.

Algorithm 1 Reservoir Sampling algorithm

```
1: procedure ReservoirSampling(stream s)
2:     for i = 1, 2, ... do
3:         Flip a coin with probability 1/i
4:         if coin was heads then
5:             x ← s_i
6:         end if
7:     end for
8:     return x
9: end procedure
```

By a simple proof by induction, we can show that the probability of any element x being sampled is $1/n$.

2.2 Mean Trick to Reduce Variance

We can take the average of multiple independent estimators to reduce the variance of the final estimator. This is useful when the variance of the single basic estimator is too large to apply concetration bounds.

Lemma 2.1 (Reduce Variance by Taking Mean of k Samples)
Let $X = \frac{1}{k}\sum_{i=1}^{k} X_i$ where each X_i has mean μ and variance σ^2. Then, $\mathrm{E}[X] = \mu$ and $\mathrm{Var}[X] = \sigma^2/k$.

2.3 Median Trick to Amplify Success Probability

We can use the median of multiple independent estimators to reduce the probability of the final estimator being far from the true value (typically by Chernoff-Hoeffding bounds).

Algorithm 2 Median Trick for Amplifying Sucess Probability

```
1: procedure AmplifyByMedian(algorithm A, integer ℓ)
2:     for i = 1, ..., ℓ do
3:         Let Z_i be the output of A with new independent random values
4:     end for
5:     return Median(Z_1, ..., Z_ℓ)
6: end procedure
```

We commonly combine both the Mean and the Median tricks to form the Median of Means trick. The typical space blow-up of such a technique is $O(\epsilon^{-2}\log(1/\delta))$.

A slightly different version of the Mean Trick is the Min Trick, where we take the minimum instead of the median. This is useful when there is only one-sided error – think of the median as a two-sided error version of the min trick. An example of this is the *Count-Min-Sketch* data structure.

3 Sketches

1. combining sketches, linear sketches 2. Misra-Gries 3. Count-Min-Sketch and Count-Sketch

4 Dimensions and Distances

Lemma 4.1 (Johnson-Lindenstrauss Lemma)
For any set $S \subseteq \mathbb{R}^d$ of n-points, there is an embedding $f\colon \mathbb{R}^d \to \mathbb{R}^m$ for $m = O(\epsilon^{-2} \log n)$ such that

$$\forall u, v \in S \quad (1-\epsilon)\|u-v\|_2^2 \leq \|f(u)-f(v)\|_2^2 \leq (1+\epsilon)\|u-v\|_2^2 \tag{15}$$

In other words, we can embed S into a lower-dimensional space while approximately preserving ℓ_2 norms.

Some observations:

- The embedding has only a logarithmic dependence on n and *no* dependence on d.
- The embedding is can be generated using a Gaussian distribution.
- The embedding can be represented as a linear transformation, or in other words, a matrix.

Definition 4.2 (Locality Sensitive Hash)
A hash family $\mathcal{H} = \{h\colon \mathcal{U} \to S\}$ is a (r_1, r_2, p_1, p_2)-locally sensitive if for all points $p, p' \in \mathcal{U}$,

1. if $d(p,p') \leq r_1$, then $\Pr_{h\in\mathcal{H}}[h(p) = h(p')] \geq p_1$,
2. if $d(p,p') > r_2$, then $\Pr_{h\in\mathcal{H}}[h(p) = h(p')] \leq p_2$.

In other words, a *locality sensitive hash* (LSH) is a hash family where similar items are more likely to collide. Note that the definition makes sense only if $r_1 < r_2$ and $p_1 > p_2$.

WIP. 1. ANN, PLEB, how to solve them