COMP 441/COMP 552 — Large Scale Machine Learning Jan 11, 2017

Lecture 2: Tail Inequalities and Reservoir Sampling

Lecturer: Anshumali Shrivastava *Scribe By: Anshumali Shrivastava*

This scribe may contain errors, please do not cite. Please email if you find any errors.

1 Why Random Sampling

In the modern big data world, when the data is exceedingly large, typically approximate estimates suffice for most practical purposes. This is because:

- We never observe the whole world, we only see some observable samples. So our estimates are anyway approximate.
- For most practical decisions, approximate estimates are sufficient.

1.1 A Template for Generic Sampling

Say we wish to estimate the proportion k of elements, with some property P, in a population S with $|S| = N$. Typically, you would take a random sample $\bar{S}$ of size n and observe $\bar{k}$ elements with property P, in the sample. We could then estimate k as $\hat{k} = \frac{N}{n} \times \bar{k}$. It is a good exercise to show that this estimate is sharply concentrated if size of $\bar{S}$ is large enough. In modern setting many times getting the random sample $\bar{S}$ itself is non-trivial. (Ex. Reservoir Sampling).

2 Tail Inequalities

Estimating answers from a random sample is one of the fundamental techniques to get approximate answers. Since the answers are not the exact, we would like to be able to bound the error on our estimates. These bounds are usually achieved using Tail Inequalities, also sometimes known as concentration inequalities.

2.1 Markov's Inequality

Statement: For any positive random variable $X \geq 0$, we have

$$Pr(X \geq a) \leq \frac{\mathbb{E}(X)}{a},$$

where $\mathbb{E}$ is an expected value.

Proof: Consider

$$\begin{aligned} aPr(X \geq a) &= a \sum_{x;x\geq a} Pr(X = x) \\ &= \sum_{x;x\geq a} aPr(X = x) \\ &\leq \sum_{x;x\geq a} xPr(X = x) \quad \text{replace } a \text{ by } x \geq a \\ &\leq \sum_{x;x\geq 0} xPr(X = x) \quad \text{add few positive terms to summation} \\ &= \mathbb{E}(\mathbb{X}) \quad \text{by definition. Note: } X \geq 0 \end{aligned}$$

2.2 Chebyshev's Inequality

Statement:

$$Pr(|X - \mathbb{E}(X)| \geq a) \leq \frac{Var(X)}{a^2},$$

where Var(X) is the variance.

An Interesting Corollary: Substitute $a = \sqrt{\frac{Var(X)}{c}}$. We get with probability $1 - c$ we have

$$X \in \left[\mathbb{E}(X) - \sqrt{\frac{Var(X)}{c}},\ \mathbb{E}(X) + \sqrt{\frac{Var(X)}{c}}\right]$$

.

3 Reservoir Sampling

Problem: How would one get an unbiased random sample from a stream of incoming data where we don't know the size of the population until we've seen the last element of the stream?
Formal Statement: You are given a stream of elements x_t arriving at time t. We only have storage of size k. After some time the stream ends (we don't know when). We want to get a random sample of size k from among all the elements seen, i.e. if we have seen n elements then the probability that any element x_i is in the storage buffer has probability $\frac{k}{n}$. The catch is that we do not know n beforehand, we only know k and at any time t the total elements seen so far.
Use Case in Real World: Suppose, you are monitoring a live twitter feed and you want to generate a perfectly random sample of k (size of memory you have) tweets from the total tweets seen. This sample can be used for estimation. For example, how many tweets are there with a particular sentiments, etc.

The solution is known as Reservoir Sampling and it works as follows.

1. Store the first k elements.
2. Every time we see the i^{th} element we select to keep it with $\frac{k}{i}$ probability. If the element is selected then we randomly knock off any one of the already existing k elements in the storage uniformly and replace it with the selected element.

It can be shown that at every point of time, we have a perfect random sample of size k, i.e. this process ensures that every element seen so far is present in the buffer with equal probability

equal to $\frac{k}{\text{total elements seen}}$.

Proof: We will use induction to prove that we get a random sample:

1. Base case: Place the first k elements of the stream into the buffer. Each element in the buffer was selected with probability $\frac{k}{m} = 1$

2. Inductive hypothesis: probability of each item being in the buffer is $\frac{k}{m}$ where m is the number of elements seen in the stream.

3. Inductive Step: Item x_{m+1} is seen and is accepted with probability $\frac{k}{m+1}$. If it is selected then one of the items currently in the buffer is removed according to a uniform distribution with $p = \frac{1}{k}$ and replaced by the item $m+1$. The probability of any given item $i < m+1$ is in the buffer can be written down as

   $Pr\big(\text{item i was in the buffer}\big) \times Pr(\text{item i is not removed})$
   $Pr\big(\text{item i was in the buffer}\big) \times \big(1 - Pr(x_{m+1}\text{was accepted})Pr(\text{item i was removed from the buffer})\big)$

$$= \frac{k}{m}(1 - \frac{k}{m+1}\frac{1}{k})$$
$$= \frac{k}{m}(1 - \frac{1}{m+1}) = \frac{k}{m+1}$$

   Thus, the probability that any item $i \leq m+1$ is in the buffer is $\frac{k}{m+1}$