CPSC 500

September 30, 2013

Scribe: Andrey Novitskiy

1 Overview

In this lecture, Coupon Collector's problem was discussed in order to give an example of usage of probability theory for computer science problems. We also used one of the most important tools on probability theory, Chebyshev's inequality, to solve CC problem and took a look at one randomized algorithm – k-Select.

2 Coupon Collector's problem

The main question of this problem is: Suppose we have n different coupons. We can equally likely collect any coupon(with replacement). What is the probability of collecting all n coupons after t trails? Using another definition: How many coupons do you expect you need to draw with replacement, before having drawn each coupon at least once?

The key to solve this problem is to understand that it takes very few trials to get needed coupon in the beginning, but the number of trials rises with the number of collected coupons.

3 Chebyshev's inequality

To solve CC problem, we need to take a look at one of the most important formulas in probability theory: Chebyshev's inequality.

Suppose we have random variable X with expected value $E[X]$ and nonzero variance σ^2 . Then for any positive a we have:

$$P[|X - E[X]| \geq a] \leq \frac{\sigma^2}{a^2}$$

The proof of this statement is easily derived from Markov's inequality, creating new variable *Z*:
$Z = (X - E[X])^2$

4 Coupon Collector's problem solution

Suppose we have *n* random variables X_i, each variable represents an event of collecting *i*-th coupon after *i*-1 coupons have been collected. Then $Y = X_1 + X_2 + ... + X_n$ represent an event, when all n coupons have been collected (our desired event). Observe that probability of collecting new coupon given *i*-1 coupons (variable X_i) $p_i = (n - (i - 1))/n$, therefore, X_i has geometric distribution with expected value $E[X_i]$ equal to $1/p_i$. Expected value of *Y* gives us:

$$E[Y] = E[X_1] + ... + E[X_n] = \frac{1}{p_1} + ... + \frac{1}{p_n} = \frac{n}{n} + \frac{n}{n-1} + ... + \frac{n}{1} = n \cdot H_n = n \ln n + o(n),$$

where H_n is the **harmonic number.**

Now let's evaluate variance of Y. Using same approach, we have:

$$Var[Y] = Var[X_1] + ... + Var[X_n] = \frac{1 - p_1}{p_1^2} + ... + \frac{1 - p_n}{p_n^2} \leq$$

$$\leq \frac{n^2}{n^2} + \frac{n^2}{(n-1)^2} + ... + \frac{n^2}{1^2} = n^2 \cdot (\frac{1}{1^2} + \frac{1}{2^2} + ... \frac{1}{n^2}) = \frac{\pi^2}{6} n^2 .$$

Now we can use Chebyshev's inequality to estimate the probability of Y being greater then it's expected value. Assume b is some positive constant we have:

$$P[Y \geq bnH_n] \leq P[|Y - nH_n| \geq (b-1)nH_n] \leq \frac{n^2 \pi^2 / 6}{(b-1)^2 n^2 H_n^2} \approx \frac{\pi^2}{6(b-1)^2 (\ln n)^2}$$

.

Using asymptotic notation, we can write an expression, which was derived by Paul Erdos and Alfred Renyi:

$$P[Y \geq n \ln n + cn] \xrightarrow[n \longrightarrow \infty]{} 1 - e^{-e^{-c}}$$

5 k-Select algorithm

One of the best example how randomization can benefit algorithm's properties is k-Select. As input we have array of *n* distinct real numbers *s*[1..*n*]. As output we need to find k-th smallest number in *s*.

Here is the algorithm:

```
Select(s, k){
        If(|s|=1) return s[1]
        p=GoodPivot(s)
        L={S[i]|S[i]<p}
        R={S[i]|S[i]>p}
If |L|>k return Select(L,k)
else if |L|=k-1 return p
else return select(R,k+|L|-1)
}
```

The main idea is pretty similar to quicksort: having a pivot p, we recursively call Select() for different pieces of input array, until desired value is found. Algorithm's speed is mainly dependent on choice of p, i.e. function GoodPivot();

One of the best implementations for GoodPivot() was proposed by Blum and Floyd:

1)GoodPivot() picks randomly from *s.*

2)Then we split *s* into *n*/5 small arrays of 5 elements and take as a pivot result of Select(*M*, *n*/10), where *M* is a set of medians of these small arrays, $|M|=n/5$.

Such a pick of GoodPivot() gives very good performance for k-th Select, showing faster results, compared to deterministic analogues:

$$T(n) \leq T(n/5) + n + T(7n/10)$$

6 Conclusion

This lecture gave a good overview of how probability theory is used in computer science. Introduction to k-Select algorithm showed an example of randomized algorithm. The next lecture will give more detailed look at properties of k-Select and show deeper analysis of it's properties.

7 References

Wikipedia, "Coupon Collector Problem", http://en.wikipedia.org/wiki/Coupon_collector's_problem

Wikipedia, "Chebyshev's Inequality" ,http://en.wikipedia.org/wiki/Chebyshev's_inequality

Wikipedia,"Selection algorithm", http://en.wikipedia.org/wiki/Selection_algorithm