COMP 6730 Advanced Database Systems Homework 1

Fall 2016

Due Wed. Oct. 5, in Class

Problem 1 (10 points)

Suppose we roll a fair k-sided die with the numbers 1 through k on the die's faces. If X is the number that appears, what is E[X]?

Answer: $E[X]=1\cdot\frac{1}{k}+2\cdot\frac{1}{k}+...+k\cdot\frac{1}{k}=\frac{1}{k}(1+2+...+k)=\frac{1}{k}\cdot\frac{k(k+1)}{2}=\frac{k+1}{2}.$

Problem 2 (10 points)

A monkey types on a 26-letter keyboard that has lowercase letters only. Each letter is chosen independently and uniformly at random from the alphabet. If the monkey types 1,000,000 letters, what is the expected number of times that sequence "proof" appears?

Answer: The word "proof" possibly starts from position 1, position 2, …, and position 1,000,000 − 4 = 999,996. Define 999,996 random variables $X_i=\begin{cases}1, & if\ "proof"\ occurs\ from\ position\ i\\ 0, & otherwise\end{cases}$ $(1\le i\le 999,996).$ Clearly, E(X_i) = Pr(X_i = 1) = $1/26^5$. Define another random variable $X=\sum_{i=1}^{999,996}X_i$. Then X is the number of times that sequence "proof" appears in the monkey's whole writing experience. From the linearity of expectation, $E(X)=\sum_{i=1}^{999,996}E(X_i)=\frac{999,996}{26^5}$. Note that the linearity of expectation holds even though the X_i's are correlated.

Problem 3 (25 points)

The following approach is often called *reservoir sampling*. Suppose that we have a sequence of items, passing by one at a time. We want to maintain a sample of one item that has the property that it is uniformly distributed over all the items that we have seen at each step. Moreover, we want to accomplish this without knowing the total number of items in advance or storing all of the items that we see.

Consider the following algorithm, which stores just one item in memory at all times. When the first item appears, it is stored in the memory. When the k-th item appears, it replaces the item in memory with probability $1/k$. Explain why this algorithm solves the problem.

Answer: Suppose we pause at time step n and we need to prove that all n items that we have seen have an equal probability $\frac{1}{n}$ to remain as the single sample in memory. In general, consider the item that comes in at time step k $(1\le k\le n)$. The probability that it stays in memory at time step k is $\frac{1}{k}$. Conditioned on it's in memory, at time step $k+1$ it remains in memory with probability $\frac{k}{k+1}$. Conditioned on it's still in memory, the probability that it remains in memory at step $k+2$ is $\frac{k+1}{k+2}$, and so on. Thus,

$\Pr[it\ survives\ at\ time\ n]=\Pr[it\ survives\ at\ time\ k]\cdot\Pr[it\ survives\ at\ k+1\mid it\ survives\ at\ k]\cdot$
$\ldots\cdot\Pr[it\ survives\ at\ n\mid it\ survives\ at\ k,k+1,\ldots,n-1]$
$=\frac{1}{k}\cdot\frac{k}{k+1}\cdot\frac{k+1}{k+2}\cdot\ldots\cdot\frac{n-1}{n}=\frac{1}{n}.$

Problem 4 (15 points)

Suppose that we roll a standard fair die 100 times. Let X be the sum of the numbers that appear over the 100 rolls. Use Chebyshev's inequality to bound Pr(|X - 350| ≥ 50).

Answer: Let X_i be the number for each roll. E(X_i) = 7/2. E(X_i^2) = 1/6*(1+4+9+16+25+36) = 91/6. Then, Var(X_i) = E(X_i^2) - E(X_i)^2 = 35/12. From the linearity of expectation, E(X) = 100*7/2 = 350. As X_i's are independent, Var(X) = 100 * Var(X_i) = 3500/12. Now, from Chebyshev's inequality,

Pr(|X - 350| ≥ 50) < Var(X) / 2500 = 7/60.

Problem 5: Exercise 2.3.1 (page 58) (25 points)

Design MapReduce algorithms to take a very large file of integers and produce as output:
(a) The largest integer.
(b) The average of all the integers.
(c) The same set of integers, but with each integer appearing only once.
(d) The count of the number of distinct integers in the input.

Answer: (a). Map the file into several chunks. For each chunk calculate the maximum value. Then map maximum value of each chunk, and reduce to get the largest value of the file.
Map(file, chunks)
Reduce(chunks, max(value))
Map(key, max(value))
Reduce(key, max(value))

(b). Similar to (a)
Map(file, chunks)
Reduce(chunks, avg(value))
Map(size(chunk), avg(value))
Reduce(key, avg(value))
Emit(1, $\sum_i avg(value) * sizeof(chunks)$ /total_number)

(c)
Map(file, content)
Emit(integer,1)
Reduce(integer, List(count))
Emit(integer,1)

(d)
Map(file, content)
Emit(integer,1)
Reduce(integer, List(count))
Emit(integer,1)
Map(integer, count)
Emit(1,1)
Reduce(1,sum(value))

Problem 6: Exercise 3.1.1 (page 95) (15 points)

Compute the Jaccard similarities of each pair of the following three sets: {1, 2, 3, 4}, {2, 3, 5, 7}, and {2, 4, 6}.

Answer: S1={1,2,3,4} s2={2,3,5,7}, s3={2,4,6}
$\text{Jaccard}(A,B)=|A\cap B|/|A\cup B|$
Jaccard(s1,s2)=2/6=1/3
Jaccard(s1,s3)=2/5
Jaccard(s2,s3)=1/6