CSE 103: Probability and statistics **Fall 2010**

Homework 5

Many of these questions are taken from Grinstead and Snell or from Feller; some have been lightly edited.

1. Suppose the probability that a car accident occurs on Monday is 5%, Tuesday 10%, Wednesday 10%, Thursday 20%, Friday 20%, Saturday 30%, and Sunday 5%. Suppose a series of 200 accidents occur at random, independently. Let X denote the number of them that occur on Sunday.

   (a) What is the expected value and variance of X?

   (b) What is the probability that X is exactly its expected value?

2. Suppose you throw m balls into n bins, where $m \geq n$. For the following questions, give answers in terms of m and n.

   (a) Let X_i be the number of balls that fall into bin i. What is $\Pr(X_i = 0)$?

   (b) What is $\Pr(X_i = 1)$?

   (c) What is $\mathbb{E}(X_i)$?

   (d) What is $\mathrm{var}(X_i)$?

3. Give an example of random variables X and Y such that $\mathbb{E}(XY) = \mathbb{E}(X)\mathbb{E}(Y)$ even though X and Y are not independent.

4. Give an example of random variables X and Y such that $\mathrm{var}(X + Y) \neq \mathrm{var}(X) + \mathrm{var}(Y)$.

5. In a sequence of coin tosses, a *run* is a series of consecutive heads or consecutive tails. For instance, the longest run in $HTHHHHTTHHTHH$ consists of three heads. We are interested in the following question: when a fair coin is tossed n times, how long a run is the resulting sequence likely to contain?

   To study this, pick any k between 1 and n, and let R_k denote the number of runs of length exactly k (for instance, a run of length $k+1$ doesn't count). In order to figure out $\mathbb{E}(R_k)$, we define the following random variables: $X_i = 1$ if a run of length exactly k begins at position i, where $i \leq n - k + 1$.

   (a) What are $\mathbb{E}(X_1)$ and $\mathbb{E}(X_{n-k+1})$?

   (b) What is $\mathbb{E}(X_i)$ for $1 < i < n - k + 1$?

   (c) What is $\mathbb{E}(R_k)$?

   (d) What is, roughly, the largest k for which $\mathbb{E}(R_k) \geq 1$?

6. Consider the following process by which a social network evolves.

```
Initially there is just one person $P_1$
For $i = 2$ to $n$:
    A new person $P_i$ arrives
    $P_i$ randomly chooses one of the $i - 1$ existing people as a friend
```

   At the end of the this process, there are n people. Let F_i be the number of friends of person P_i. In both of the following problems, ignore the first person (P_1).

(a) Everyone has at least one friend. What is the probability that P_i has exactly one friend? That is, compute $\Pr(F_i = 1)$ as a function of i and n.

(b) What is the expected number of people with exactly one friend?

7. There are n processes $P_1, \ldots, P_n$ that all want access to a single database. The problem is that if more than one process tries to access the database at the same time, none of them will succeed due to a particular locking mechanism. So instead, we use the following randomized protocol:

```
For time $t = 1, 2, \ldots$:
    Each processor flips a coin with heads probability $1/n$
    If heads, it attempts to access the database at time $t$
```

(a) Fix any time t, and let $E_{t,i}$ be the event that process P_i successfully accesses the database at that time. This means that P_i's coin flip turns up heads, whereas the coin flip of every other processor turns up tails. What is $\Pr(E_{t,i})$ exactly? Use the approximation $1 - x \sim e^{-x}$ to show that if n is large, this probability is approximately $1/(en)$. Henceforth pretend that this approximation is exact, and use it in the next two questions.

(b) Suppose that at time t, there are still k processes that have not yet been able to access the database. What is the expected additional time it takes until one of them gets access? Give your answer in terms of k and n. *Hint:* Start by figuring out the probability that one of these k processes gets access at time t.

(c) Roughly what is the expected time by which *all* processes will have been able to access the database?

8. Here is a very general balls-in-bins problems. Suppose m balls are thrown into n bins, but that the bins are not equally likely to be chosen. Each time a ball is thrown, it goes into bin 1 with probability p_1, bin 2 with probability p_2, and so on. The numbers $p_1, p_2, \ldots, p_n$ are nonnegative and sum to 1.

(a) Let X_i be the number of balls that fall into bin i. What is the probability that X_i is exactly k?

(b) What is $\mathbb{E}(X_i)$ and $\mathrm{var}(X_i)$?

(c) Give an upper bound on the probability that there is an empty bin. The larger m is, the smaller this probability. Roughly how large should m be for this probability to be less than 1/2? (So that with probability at least 1/2, every bin will contain one or more balls: a generalization of the *coupon collector* problem.)

(d) Give an upper bound on the probability that there is a bin with ≥ 2 elements. The larger m is, the larger this probability. Roughly how small must m be for this probability to be less than 1/2? (So that with probability at least 1/2, every bin has at most one ball: a generalization of the *birthday paradox*.)

(e) Check that your answers to (c) and (d) agree with what we already know for the uniform case, that is, $p_1 = \cdots = p_n = 1/n$.

9. **(Challenging)** Suppose a fair coin is tossed repeatedly until the same outcome occurs twice in a row (that is, two heads in a row or two tails in a row). What is the expected number of tosses?

10. **(Challenging)** Recall the *quicksort* algorithm:

```
Function quicksort(S)
Pick an element $v$ from $S$ at random
Split $S$ into three pieces:
   $S_L$, elements less than $v$
   $S_v$, elements equal to $v$
   $S_R$, elements greater than $v$
Return quicksort($S_L$) $\circ S_v \circ$ quicksort($S_R$) // $\circ$ is concatenation
```

In this problem, we will analyze the expected running time of this algorithm.

First, notice that all the work is done in splitting the various arrays: the original array and the many subarrays generated by recursive invocations. Each split operation involves *comparing* all elements of the array to the *split element* v. Therefore

$$\text{total running time of quicksort} \quad = \quad \sum_{i,j} X_{ij}$$

where X_{ij} is the number of times $S[i]$ and $S[j]$ are compared; such comparisons occur only when one of them is the split element.

It will be convenient to let $S_{(i)}$ denote the ith largest element in the array; and to instead define X_{ij} as the number of times $S_{(i)}$ and $S_{(j)}$ get compared. The formula for the running time remains the same.

The second useful observation is that X_{ij} is never more than 1: if $S_{(i)}$ and $S_{(j)}$ are compared at some point, it means one of them is the split element; thus they will end up in separate arrays and never again be compared. Let p_{ij} be the probability that $S_{(i)}$ and $S_{(j)}$ are compared at some stage. Thus $\mathbb{E}(X_{ij}) = p_{ij}$.

(a) Suppose $j > i$. Consider the elements $S_{(i)}, S_{(i+1)}, \ldots, S_{(j-1)}, S_{(j)}$ in the original array. At some point, this group of elements will be split: they will no longer remain within a single subarray. This happens when the split element v ends up being one of these $j - i + 1$ elements. With this in mind, show that $p_{ij} = 2/(j - i + 1)$.

(b) Show that the expected running time is $O(n \log n)$.

Answer key 1 (a) $\mathbb{E}(X) = 10$, $\text{var}(X) = 9.5$ (b) $\binom{200}{10}(\frac{1}{20})^{10}(\frac{19}{20})^{190}$. 2 (a) $(1 - \frac{1}{n})^m$ (b) $\frac{m}{n}(1 - \frac{1}{n})^{m-1}$ (c) $\frac{m}{n}$ (d) $\frac{m}{n}(1 - \frac{1}{n})$. 4. Take $X = Y$. 5 (a) $1/2^k$ (b) $1/2^{k+1}$ (c) $(n - k + 3)/2^{k+1}$ (d) $\log n$. 6 (a) $(i - 1)/(n - 1)$ (b) $n/2$. 7 (a) $\Pr(E_{t,i}) = \frac{1}{n}(1 - \frac{1}{n})^{n-1}$ (b) en/k (c) $en \ln n$. 8 (a) $\binom{m}{k} p_i^k (1 - p_i)^{m-k}$ (b) $\mathbb{E}(X_i) = mp_i, \text{var}(X_i) = mp_i(1 - p_i)$ (c) $\sum_i (1 - p_i)^m$, take $m \geq (\ln 2n)/(\min_i p_i)$, for instance (d) $\binom{m}{2} \sum_i p_i^2$, take $m \leq 1/\sqrt{\sum_i p_i^2}$. 9. 3.