36-703 Homework #1 Thursday 26 Jan 2006

1

Pentagon Walk Revisited

Let $X_0, X_1, \ldots$ be random variables that take values in the set $\{A, B, C, D, E\}$. Suppose X_n is the node being visited at time n in the Pentagon walk we discussed in class. That is, we assume that $X_0 = A$ with probability 1 and that X_n moves to each node adjacent to node X_{n-1} with probability 1/2, independently of where it was before.

Find $\mathsf{P}\{X_n = B\}$ for all n. What is this probability in the limit? What do you guess the corresponding limiting probability is for the other nodes and why?

2

Double Heads Revisited

Suppose we flip a coin until two consecutive heads appear. Assume that the coin flips are all independent. and that the coin comes up heads with probability $0 < p < 1$.

Let H_i be the indicator of heads on the ith roll, for $i \in \mathbb{Z}_+$.

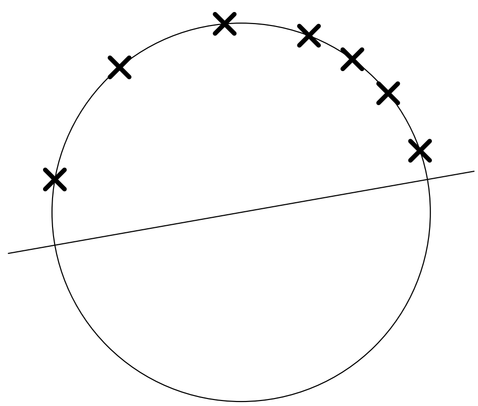
Let N be the number of flips (inclusive) until heads first appears on two consecutive flips.

In class we found p_N when $p = 1/2$. Carry out the same analysis for general $0 < p < 1$.

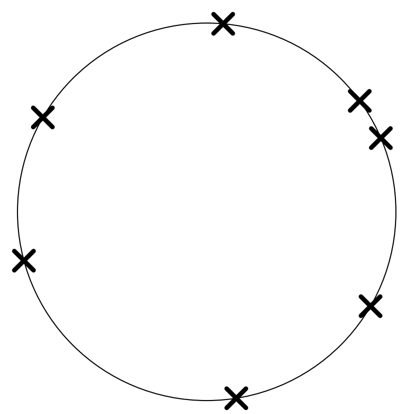
3 Semi-Circle

You choose n random points in succession on the unit circle. (That is, on the perimeter not in the disk.) The position of each point is chosen uniformly over the circle, and the positions of all points are independent.

Find the probability that the n points *all* lie within a semi-circle, or in other words that they all lie within a 180° arc on the circle. (See the figure for examples.)



All points lie within a semi-circle



All points do not lie within a semi-circle

HINT: For each i, consider *separately* the event that all the points fall within a semi-circle clockwise from the ith point chosen. Then, for each i, condition on the position of the ith point. Relate this to whether all n points lie in a semi-circle.

You should do the following in your write up:

- **Describe the Experiment.**
- **Specify your assumptions.**
- **Define relevant random variables.**
- **State what you know.**
- **State what you want to find.**

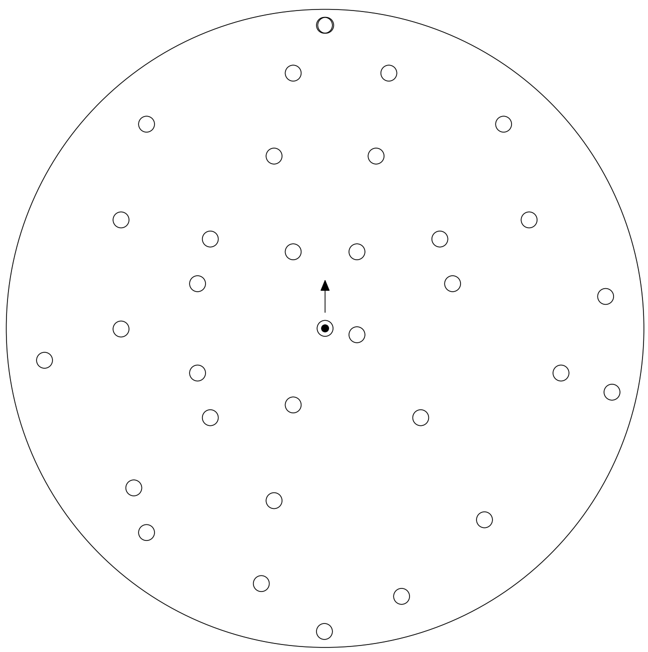
Warning! Set up the problem properly, and express your argument carefully in terms of properly defined random variables or events. Not only will this help you in your solution, but you must show your reasoning explicitly to master this exam.

4 — Mean Free Path

We consider an idealized, two-dimensional gas with molecules of radius $r > 0$ within the disk or radius $R >> r$. (See the figure below.)

Assume that the number of molecules within the disk of radius R has a Poisson$\langle\lambda\rangle$ distribution and that the positions of all molecules are independent and uniform over the disk. Ignore the motion of these molecules; that is, treat the positions as fixed throughout the experiment.

Another molecule, identical to the others, is inserted at the origin and sent along the y axis at some velocity (presumably much faster than the other molecules since we ignore their motion).



Find the expected value of the free path distance, defined as the distance the new molecule travels before hitting another molecule or the edge of the disk, whichever comes first. (If there is already a molecule within $2r$ of the origin, you this distance is 0.)

NOTE: Two molecules collide if their center are ever $\leq 2r$ apart.

HINT: Start by computing the probability that the free path distance is greater than a particular value ℓ; that is, find its survival function (SF). Since the free path distance is a non-negative random variable, its expected value can be computed by integrating the SF from zero to infinity.

5 Random Permutations

Suppose your computer has a random number generator that can produce a sequence of independent random numbers with a Uniform$\langle 0,1\rangle$ distribution. In this exercise, you will analyze an algorithm for using this to generate a random permutation of the labels $1,\ldots,n$. For example, the $5,3,1,4,2$ is a permutation of $1,2,3,4,5$.

We will make use of the following random varibles:

- Let $U_2,\ldots,U_n$ denote independent Uniform$\langle 0,1\rangle$ random variables. (This is the continuous uniform distribution.) These are the values generated on your computer.
- Let $I_2,\ldots,I_n$ denote independent random variables where, for $k = 2,\ldots,n$, I_k has a DiscreteUniform$\langle 1,\ldots,k\rangle$ distribution. We can generate I_k from U_k as follows:

$$I_k = 1 + \lfloor kU_k \rfloor,$$

  where $\lfloor x \rfloor$ is the biggest integer less than or equal to x.
- Let $X_1,\ldots,X_n$ denote the random permutation where X_i denotes the label from $1,\ldots,n$ that ends up in position i. These are the random variables being generated by the algorithm. For example, if $(5,3,1,4,2)$ is the result of the algorithm when $n = 5$, then $X_1 = 5$, $X_2 = 3$, $X_3 = 1$, $X_4 = 4$, and $X_5 = 2$.

The algorithm is as follows:

1. Let $a_1,\ldots,a_n$ be a scratch array of numbers that holds the permutation while we build it. We initialize this array by setting $a_i = i$ for $i = 1,\ldots,n$.
2. Let j be an index variable keeping track of our iteration. We initially set $j = n$.
3. Generate U_j and set I_j as described above.
4. Swap the values of a_j and a_{I_j} in our scratch array.
5. Decrement j by 1.
6. If $j > 1$, goto step 3. If $j = 1$, stop and set $X_i = a_i$ for $i = 1,\ldots,n$.

Thus, $X_1,\ldots,X_n$ represents the permutation output by the algorithm.

Your goal is as follows:

Show that the algorithm below generates a random permutation of n labels in such a way that all $n!$ permutations are equally likely.

HINT: A very useful fact is the multiplication rule for probabilities. Let $\mathcal{E}_1, \mathcal{E}_2, \ldots$, denote events. We have $\mathsf{P}(\mathcal{E}_1 \cap \mathcal{E}_2) = \mathsf{P}(\mathcal{E}_1)\,\mathsf{P}(\mathcal{E}_2 \mid \mathcal{E}_1)$ and that $\mathsf{P}(\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3) = \mathsf{P}(\mathcal{E}_1)\,\mathsf{P}(\mathcal{E}_2 \mid \mathcal{E}_1)\,\mathsf{P}(\mathcal{E}_3 \mid \mathcal{E}_1 \cap \mathcal{E}_2)$. This generalizes:

$$\mathsf{P}(\mathcal{E}_1 \cap \cdots \cap \mathcal{E}_m) = \mathsf{P}(\mathcal{E}_1)\,\mathsf{P}(\mathcal{E}_2 \mid \mathcal{E}_1)\,\cdots\,\mathsf{P}(\mathcal{E}_m \mid \mathcal{E}_1 \cap \cdots \cap \mathcal{E}_{m-1}).$$

6 Rejection Sampling

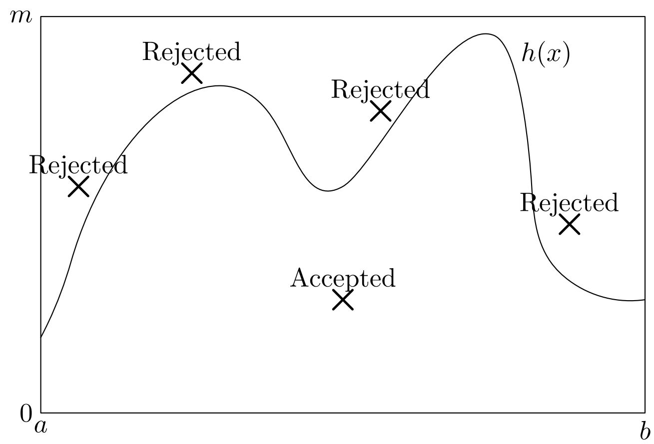
A useful method for generating random numbers with a particular distribution is called the *rejection method*. In this exercise, we show that this method works.

Suppose that $h(x)$ is a PDF that is non-zero only on an interval $]a, b[$ for some numbers $a < b$. Let $H(x)$ be the CDF corresponding to the PDF $h(x)$; that is, $H(x) = \int_{-\infty}^{x} h(u)\,du$. Suppose also that the maximum value of $h(x)$ is less than $m > 0$.

To generate a random variable X with CDF $H(x)$ (and thus PDF $h(x)$), run the following procedure:

0. Initialize an index of iteration i, setting $i = 1$.
1. Generate a random pair (X_i, Y_i) where X_i and Y_i are independent, X_i has a Uniform$\langle a, b\rangle$ distribution, and Y_i has a Uniform$\langle 0, m\rangle$ distribution. Assume that (X_i, Y_i) is independent of all other generated random pairs.
2. If $Y_i \le h(X_i)$, we "accept" the pair, and goto step 4.
3. Otherwise $Y_i > h(X_i)$ and the random pair (X_i, Y_i) is "rejected". Increment i by one and goto step 1.
4. Set $X = X_i$ and return the random value X.

The figure below illustrates the method.



> Show that X, the value returned by the above procedure, has CDF equal to H, that is, that $\mathsf{F}_X(u) = H(u)$ for all values u.

Hints:

1. Be careful in defining random variables to distinguish between the number returned by the procedure and the random pairs used at each iteration.
2. The algorithm generates a sequence of random pairs, stopping with the first pair to be accepted. This acceptance can happen on one and only one iteration.
3. Think carefully about how to write the event that the ith pair is accepted in terms of the X's and Y's.
4. The distribution of the first component of a pair given that that that pair was accepted *need not be the same* as the distribution of the first component of the pair without that information.