Solutions to the review problems

CS244-Randomness and Computation

May 9, 2014

1. If we sample without replacement, each draw of three cards is a 3-element subset $\{a, b, c\}$, of the 52-cards, so the sample space is

$$\Omega = \{\omega \subseteq \{1, \ldots, 52\} : |\omega| = 3\}.$$

We suppose each of these outcomes is equally likely, so the probability function assigns to each outcome ω

$$P[\omega] = 1/|\Omega| = 1/\binom{52}{3} = 4.52 \times 10^{-5}.$$

The event in question is

$$E = \{\{a, b, c\} : a, b, c \quad \text{have} \quad \text{rank} \quad 3\}.$$

Since there are 4 3's in the deck,

$$|E| = \binom{4}{3} = 4,$$

the probability is

$$4/\binom{52}{3} = 1.8 \times 10^{-4}.$$

If we sample with replacement then every sequence of three cards (repetitions allowed) is an outcome, so

$$\Omega = \{(a, b, c) : 1 \leq a \leq 52\}$$

so $|\Omega| = 52^3$ and P assigns to each $\omega \in \Omega$ the probability

$$P[\omega] = \frac{1}{52^3} = 7.11 \times 10^{-6}.$$

The event in question is

$$E = \{(a, b, c) : a, b, c \quad \text{have} \quad \text{rank} \quad 3\},$$

which has cardinality 4^3, so

$$P[E] = \frac{4^3}{52^3} = 4.55 \times 10^{-4}.$$

It makes sense that the probability is higher in the second scenario, since in the first it becomes much less likely to draw another 3 if the first and second cards drawn are both 3's.

2. The possible values of X are 1, 2 and 3. Each basic outcome is a sequence (a, b, c) of integers in the set $\{1, 2, 3, 4, 5, 6\}$, with probability 6^{-3}. There are 6 outcomes that give $X = 1$, and $6 \times 5 \times 4$ outcomes that give $X = 3$. (In the first case, choosing the first term of the sequence determines the other 2, in the second case, choosing the first term leaves 5 choices for the second, then 4 choices for the third.) Thus

$$P[X = 1] = 6 \times 6^{-3} = 6^{-2} = 0.0278,$$

$$P[X = 3] = 6 \times 5 \times 4 \times 6^{-3} = 0.556.$$

$$P[X = 2]$$

is then the difference between the sum of these two and 1, that is 0.416.

3. The probability of your guessing the outcome of 63 successive games, even with your 65% success rate, is

$$0.65^{63} = 1.635 \times 10^{-12}.$$

The probability that NO ONE of 3×10^8 players with the same success rate wins the contest is thus

$$\begin{aligned}(1 - 1.635 \times 10^{-12})^{3\times10^8} &\approx \exp -1.635 \times 10^{-12} \times 3 \times 10^8 \\ &= \exp -4.9 \times 10^{-4} \\ &= 0.99951.\end{aligned}$$

So the probability that someone wins is 4.9×10^{-4}. (The fact that $exp{-}4.9 \times 10^{-4}$ evaluates to $1 - 4.9 \times 10^{-4}$ is not a coincidence! So close to $x = 0$ the graph of the exponential function is for all practical purposes identical to the line $y = x + 1$.)

4.(a) Part (b) of this problem is a chore, but let's do the (much!) simpler part (a). We'll denote the PDF of X^2 by

$$\begin{aligned} F_{X^2}(x) &= P[X^2 \leq x] \\ &= P[X \leq \sqrt{x}] \\ &= \sqrt{x}. \end{aligned}$$

The density function is then the derivative of this, namely

$$\frac{1}{2\sqrt{x}}.$$

5.

$$\binom{100}{0} = 1, \binom{100}{1} = 100.$$

For the other question we have to pause just a little:

$$\binom{100}{2} = \frac{100 \times 99}{2 \times 1} = 50 \times 99 = 4950.$$

6. Let N be the number of 6-element subsets of the 100 bottles of beer that include one of each kind of beer. The probability we are looking for is $N/\binom{100}{6}$. There are several different ways to determine N. Here is the direct approach:

The 6 beers could split either $(4, 1, 1)$ (four of one kind, 1 of each of the others), $(3, 2, 1)$ or $(2, 2, 2)$.

A $(2, 2, 2)$ split contributes

$$N_1 = \binom{40}{2}\binom{30}{2}\binom{30}{2}$$

subsets.

A $(4, 1, 1)$ split could mean either 4 pilseners, 4 stouts, or 4 ales, so this contributes

$$N_2 = \binom{40}{4}\binom{30}{1}\binom{30}{1} + \binom{40}{1}\binom{30}{4}\binom{30}{1} + \binom{40}{4}\binom{30}{1}\binom{30}{4}$$

subsets.

A $(3,2,1)$ split could happen 6 different ways, contributing

$$2\binom{40}{3}\binom{30}{1}\binom{30}{2}+2\binom{40}{2}\binom{30}{1}\binom{30}{3}+2\binom{40}{1}\binom{30}{1}\binom{30}{2}$$

subsets. You have to evaluate all those binomial coefficients to compute N_1, N_2, N_3 and then set N to the sum. I evaluated these on the computer and got

$$N = 884782500,$$

so the probability is 0.7422.

Another way to do it is to count those selections that *don't* contain all three kinds of beer: The number of subsets with no pale ale is $\binom{60}{6}$, with no pilsener $\binom{70}{6}$ and with no stout $\binom{70}{6}$. The problem is that in counting this way, we have counted subsets that contain only one kind of beer twice, so we have to subtract these off. The result is

$$2\binom{70}{6}+\binom{60}{6}-\binom{40}{6}-2\binom{30}{6}=307269900$$

so N is the difference of this number from $\binom{100}{6}$, which miraculously is 884782500, as we found before.

7. $P[X=2]$ (you draw 2 3's) and $P[Y=2]$ (you draw 2 hears) are obviously both positive, but $P[X=2,Y=3]=0$, since there is only one three of hearts in the deck! So

$$P[X=2,Y=3] \neq P[X=2]P[Y=3],$$

and thus the events are not independent, so X, Y are not independent. Observe that in this problem and the next, any counterexample proves non-independence.

8. By the same token, $P[X \geq 0.9]$ is nonzero, as it represents the relative area of a little tip of the circle near the point $(1,0)$. $P[Y \geq 0.9]$ is the same. Thus the product of these two is nonzero, but $P[X \geq 0.9, Y \geq 0.9] = 0$, as no point in the circle has both coordinates greater than 0.9.

9. Let E_1, E_2, F denote, respectively, the events 'the beers were taken from Pantry 1', 'the beers were taken from Pantry 2', and 'all three types of beers were chosen'. The problem is to compute $P[E_1|F]$. We compute $P[F|E_1]$ and $P[F|E_2]$ by the methods of Problem 6 and then get

$$P[E_1|F] = \frac{P[F|E_1]P[E_1]}{P[F]}.$$

We assume $P[E_1] = 0.5$ and thus

$$P[F] = 0.5(P[F|E_1] + P[F|E_2]).$$

Now you just have to plug in the numbers found by the methods of Problem 6.

10. $E(X) = 1/2$, as we have found many times before, and its density is the function that is constant and equal to 1 on the interval $[0, 1]$.

$$E(Z) = E(X^2) + E(Y^2) = 2E(X^2).$$

We can compute $E(X^2)$ using the density for X^2 computed in problem 4, but we also can use the density p_X of X. This gives

$$E(X^2) = \int_{-\infty}^{\infty} x^2 p_X(x)dx = \int_0^1 x^2 dx = 1/3.$$

Thus $E(Z) = 2/3$.

11. The probability that the game lasts 2 rounds is 1/6. The probability that the game lasts exactly 3 rounds is $5/6 \times 2/6 = 10/36$, since we can have anything the first roll, have 5 allowable values for the second, and then the third roll has to match one of the first 2. We continue in this manner, and find the following values for PMF f_X:

2	$1/6$
3	$10/36$
4	$5/6 \times 4/6 \times 3/6 = 10/36$
5	$5/6 \times 4/6 \times 3/6 \times 4/6 = 40/6^3$
6	$5/6 \times 4/6 \times 3/6 \times 2/6 \times 5/6 = 100/6^4$
7	$1 - s$

where s is the sum of the 5 preceding probabilities. The expected value is then

$$\sum_{j=2}^{7} j f_X(j).$$

12. Because X^2 and Y^2 are independent and identically distributed,

$$var(Z) = var(X^2) + var(Y^2) = 2 \cdot var(X^2).$$

$$var(X^2) = E(X^4) - E(X^2)^2.$$

We can proceed as in Problem 10 and use the fact that the density for X is the constant function 1 on the interval $[0, 1]$, so

$$E(X^4) = \int_0^1 x^4 dx = 1/5.$$

Since we know $E(X^2) = 1/3$, we get

$$var(Z) = 2(1/5 - 1/9) = 8/45,$$

and the standard deviation is the square root of this (0.4216).

13. (No solution provided.)

14. The number of raindrops hitting the area during a one-second period is a Poisson-distributed random variable X. We have that $P[X > 0] = 1/3$ from the information in the problem, so $P[X = 0] = 2/3$. Thus we can find the parameter λ of the Poisson distribution:

$$e^{-\lambda} = 2/3,$$

so

$$\lambda = \ln 1.5 = 0.405.$$

This is the expected value of the Poisson distribution, and is thus the average number of raindrops per second. The waiting time is exponentially distributed with CDF $1 - e^{-\lambda x}$, and this distribution has expected value $1/\lambda = 2.466$. For the last question,

$$P[X \geq 2] = 1 - P[X < 2] = 1 - e^{-\lambda}(1 + \lambda) = 0.063.$$

15. We already found $E(X^2) = 1/3$, and $var(X^2) = 1/5 - 1/9 = 0.08889$. Thus the standard deviation of X^2 is the square root of this last, or 0.2981. Let U be the sum of 100 independent copies of X^2. U has mean 33.333 and standard deviation $\sqrt{100} \times 0.2981 = 2.981$. The problem asks for

$$P[U \geq 30].$$

We normalize to

$$V = \frac{U - 33.33}{2.981},$$

which has mean 0 and standard deviation 1, and apply the Central Limit Theorem,

$$\begin{aligned} P[U \geq 30] &= P[V \geq \frac{30 - 33.33}{29.81}] \\ &= P[V > -1.118] \\ &\approx \Phi(1.118). \end{aligned}$$

where Φ is the cumulative standard normal distribution.

16. The mean of each row is 3, so the mean of the 4 columns is the column vector $[3 \quad 3 \quad 3]^T$. This is more of a computer problem than a do-by-hand problem. The code below shows the remaining computations for this problem, carried out in Python:

```
>>> X=array([[1,3,-2,10],[2,5,-1,6],[-3,7,4,4]])
>>> Y=X-3 #in numpy this subtracts 3 from every entry
>>> print Y
[[-2  0 -5  7]
 [-1  2 -4  3]
 [-6  4  1  1]]
>>> Z=Y.dot(Y.T)
>>> M=cov(Y)
>>> print M
[[ 26.          14.33333333   4.66666667]
 [ 14.33333333  10.           4.33333333]
 [  4.66666667   4.33333333  18.        ]]
>>> print Z
[[78 43 14]
 [43 30 13]
 [14 13 54]]
>>> #Z is exactly 3 times M
>>> #diagonalize covariance:
>>> u,d,ut=linalg.svd(M)
>>> #verify that u and ut are transposes
>>> print u.dot(T)-ut
>>> print u.T-ut
[[ -1.11022302e-16  -2.22044605e-16  -1.11022302e-16]
 [  0.00000000e+00   2.49800181e-16   2.22044605e-16]
 [  5.55111512e-17   3.33066907e-16  -1.66533454e-16]]
```

```
>>> #verify that u is orthogonal
>>> u.dot(u.T)
array([[  1.00000000e+00,   1.11022302e-16,   1.11022302e-16],
       [  1.11022302e-16,   1.00000000e+00,   1.38777878e-16],
       [  9.71445147e-17,   1.38777878e-16,   1.00000000e+00]])
>>> #verify that the columns of u are eigenvectors of M with
>>> #eigenvalues given in d:
>>> print d
[ 36.50986857  16.0278244    1.46230702]
>>> M.dot(u)/u
array([[ 36.50986857,  16.0278244 ,   1.46230702],
       [ 36.50986857,  16.0278244 ,   1.46230702],
       [ 36.50986857,  16.0278244 ,   1.46230702]])
>>> #projections onto two principal components:
>>> #first form the matrix from the first two rows of u transpose
>>> v=ut[[0,1],:]
>>> print v
[[-0.81080606 -0.4905736  -0.31926648]
 [ 0.32060986  0.08411702 -0.94346894]]
>>> projections=v.dot(Y)
>>> #now express projections back in original coordinates
>>> Yreconst=v.T.dot(projections)
>>> #add back the mean
>>> Xreconst=3+Yreconst
>>> print Xreconst
[[ 1.31661044  3.67496856 -2.54352314  9.55194414]
 [ 1.43923283  3.8045239  -0.03733453  6.79357781]
 [-2.94240577  7.12278273  3.90112835  3.91849469]]
>>> #find difference from original X
>>> print Xreconst-X
[[ 0.31661044  0.67496856 -0.54352314 -0.44805586]
 [-0.56076717 -1.1954761   0.96266547  0.79357781]
 [ 0.05759423  0.12278273 -0.09887165 -0.08150531]]
>>> #the differences are fairly (although not very) small
>>> #relative to the entries in the original matrix
```

17. The state-transition matrix, with the ordering of the states as given in the

problem, is

$$\begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The matrix Q is the 3×3 portion in the upper left, corresponding to the non-absorbing states. The principal fact about Q is that the ij-entry of $(I - Q)^{-1}$ is the expected number of times the chain is in state j, given that it starts in state i. Computing the inverse matrix entails solving a system of linear equations. This can be done, a bit tediously, by hand. I used the computer and found

$$(I - Q)^{-1} = \begin{bmatrix} 4 & 4 & 2 \\ 2 & 4 & 2 \\ 2 & 2 & 2 \end{bmatrix}.$$

The row sums give the expected time to absorption from different start states. The expected time to absorption from the 'None' state is therefore 10.

18. We begin by computing the transition matrix. The probability of a transition from state 1 to state 2 in the 'follow the link mode' is 0.5, because there are two different links. The probability of a transition to state 2 in the 'random surfing mode' is 1/3. (We are allowing a random surfing step to land the surfer on the same page.) Thus the $1, 2$ entry of the transition matrix is

$$0.8 * 0.5 + 0.2/3 = 0.4667.$$

The remaining entries of the matrix are computed the same way. All the entries can be written as fractions with a denominator of 30. This gives the transition matrix

$$M = \frac{1}{30} \cdot \begin{bmatrix} 2 & 14 & 14 \\ 2 & 2 & 26 \\ 2 & 26 & 2 \end{bmatrix}.$$

To find the stationary distribution, we need a solution to

$$[a \quad b \quad c]M = [a \quad b \quad c],$$

Python can do it for you if you remember that we have to find a left eigenvector of M, which is an ordinary eigenvector of M^T. Because of the nice integer entries, though, it's not too hard to come up with a solution by hand:

$$[2 \quad 14 \quad 14].$$

We need to normalize this so that the sum of the components is 1. This gives

$$[\frac{1}{15} \quad \frac{7}{15} \quad \frac{7}{15}]$$

as the result. You can see from the description of the system at the outset that the surfer is more likely to wind up in states 2 or 3 than in state 1, and that the probabilities ought to be the same for states 2 and 3.