Assignment #2, Module: MA5634

Gustavo Ramirez

December 7, 2016

1 POISSON-DISTRIBUTED BANK WITHDRAWALS

REVIEW OF CONCEPTS:

First, it's important to recall that a random variable is "some numerical value determined by the result of an experiment".

Both the *cumulative distribution function* and the *probability mass function* are defined over that random variable:

$$F(x) = P\{X \le x\} \tag{1.1}$$

$$p(x) = P\{X = x\} \tag{1.2}$$

For a continuous random variable:

$$P\{X \in C\} = \int_C f(x)dx \tag{1.3}$$

where f is the *probability density function*.

The *cumulative distribution* and the *probability density* can be connected:

$$F(a) = P\{X \in (-\infty, a)\} = \int_{-\infty}^{a} f(x)dx \Rightarrow \frac{d}{da}F(a) = f(a) \tag{1.4}$$

Another perspective: $f(a)$ is the measure of how likely it is that the random variable will be near a.

Also, it's important to recall the expectation (or expected value) of a random variable: if X is a continuous random variable having probability density function f, then:

$$E[X] = \int_{-\infty}^{\infty} xf(x)dx \tag{1.5}$$

and also:

$$E[g(X)] = \int_{-\infty}^{\infty} g(x)f(x)dx \tag{1.6}$$

Finally, let's review the concept of variance:

$$\mu = E[g(X)] \Rightarrow Var(X) = E[(X - \mu)^2] \tag{1.7}$$

ON THE POISSON AND THE EXPONENTIAL DISTRIBUTIONS:

- **Poisson (discrete):**

$$p_i = P\{X = i\} = e^{-\lambda}\frac{\lambda^i}{i!}, \quad i = 0, 1, \dots \tag{1.8}$$

$$\Rightarrow P\{X < j\} = \sum_{i=0}^{j} p_i, \quad E[X] = \lambda \tag{1.9}$$

- **exponential (continuous):**

$$f(x) = \lambda e^{-\lambda x}, \quad 0 < x < \infty \tag{1.10}$$

$$\Rightarrow P\{X < a\} = \int_{0}^{a} f(x)dx, \quad E[X] = \frac{1}{\lambda} \tag{1.11}$$

ON THE POSED PROBLEM:

A Poisson (discrete) distribution is given for the number of withdrawls (X) in a single month. On the other hand, for the amount of each withdrawal (Y), a exponential (continuous) distribution is given.

The final goal is: *calculate the probability that the total sum of withdrawals in a given month exceeds 50 000, i.e.:*

$$P(XY > 50000) =? \tag{1.12}$$

According to the data given in the problem (here a slight change in notation is made: i refers to number of withdrawals, and x refers to the amount per withdrawal):

$$p_i = P\{X = i\} = e^{-50}\frac{50^i}{i!}, \quad i = 0, 1, \dots \tag{1.13}$$

$$f(x) = \frac{e^{-\frac{x}{800}}}{800}, \quad 0 < x < \infty \tag{1.14}$$

Superficial solution:

In the case of a discrete distribution, obtaining a probability is simple, as it's just p_i, while in the continuous case, $f(x)$ is a probability density, and $f(x)dx$ is the corresponding probability (of being around x in an interval of width dx).

Then, in the specific case of study here, and for equations 1.13 and 1.14, the probability of having withdrawn i times, with a x amount per each withdrawal is:

$$P(i, x) = p_i f(x)dx \tag{1.15}$$

For each number i, there is a probability that $ix < 50000$ (or equivalently, that $x < \frac{50000}{i}$). Then (for fixed i):

$$P(i)|_{ix<50000,\ fixed\,i} = \int_0^{50000/i} p_i f(x)\,dx \tag{1.16}$$

Those probabilities have to be added, for every possibly i, which implies:

$$P_{total} = P|_{ix<50000} = p_0 \int_0^\infty f(x)dx + \sum_1^\infty \int_0^{50000/i} p_i f(x)dx = p_0 + \sum_1^\infty \int_0^{50000/i} p_i f(x)dx \quad (1.17)$$

***Implementation of solution*:**

Finally, plugging the distributions information:

$$P_{total} = e^{-50} + \sum_1^\infty \left(e^{-50}\frac{50^i}{i!} \right) \int_0^{50000/i} \left(\frac{e^{-\frac{x}{800}}}{800} \right) dx = e^{-50} + \sum_1^\infty \left(\left(e^{-50}\frac{50^i}{i!} \right)\left(1 - e^{-50000/(800i)} \right) \right)$$

$$= e^{-50} + \sum_1^\infty \left(e^{-50}\frac{50^i}{i!} \right) - \sum_1^\infty \left(e^{-50}\frac{50^i}{i!} \right) e^{-50000/(800i)} = 1 - e^{-50} \sum_1^\infty \left(e^{-62.5i}\frac{50^i}{i!} \right)$$

$$(1.18)$$

The goal is to obtain P such that $ix > 50000$. Then:

$$P = 1 - P_{total} = e^{-50} \sum_1^\infty \left(e^{-62.5i}\frac{50^i}{i!} \right) \quad (1.19)$$

The sum can be performed numerically, and in this case it was done with the use of a Wolfram|Alpha's online sum calculator. The result is:

$$P(XY > 50000) = e^{-50} \sum_1^\infty \left(e^{-62.5i}\frac{50^i}{i!} \right) = 6.93 \cdot 10^{-48} \quad (1.20)$$

2 Monte Carlo integration

- **uniform random numbers:**

  The given integral is:

$$I = \int_0^3 \frac{e^{-s}}{1 + \frac{s}{9}} ds \quad (2.1)$$

  Making the change of variable $u = s/3$, the integral takes the form:

Table 2.1: results for MC integration with uniformly distributed random variable.

N	Result	Standard Deviation
100	0.7322465567101966	0.6348587381229975
1000	0.838864109739082	0.7697936527603505
10000	0.8660623409784465	0.7680669900080613
100000	0.873798713723586	0.7699488672469019

$$I = \int_0^1 3\frac{e^{-3u}}{1 + \frac{u}{3}} du \tag{2.2}$$

A numerical simulation of the integral in equation 2.2 was implemented in script prob/uniform_integration.py. In that script, a uniformly distributed random variable was used.

For the various numbers of generated random numbers, the results are presented in table 2.

- **exponentially distributed random numbers:**

As seen in table 2, the standard deviations are very high (on the order of the solution of the integral). To be able to reduce these errors, an exponentially distributed random variable is used.

Before this, let's review quickly how to change the form in which the random variables are distributed.

According to the inverse transform algorithm: *if U is a continuous random variable over $(0, 1)$, and $X = F^{-1}(U)$, then X has distribution function $F(x)$.* Then, the idea is to generate a set of random numbers U over $(0, 1)$, and then set: $X = F^{-1}(U)$.

In the case of interest in generating an exponentially distributed random variable, as seen in class: $X = -\frac{1}{\lambda}\log(1 - U)$, with $f(x) = \lambda e^{-\lambda x}$. In the posed problem, we choose: $f(x) = 3e^{-3x}$.

The idea of using an exponentially distributed random variable is the following:

$$I = \int_0^1 3\frac{e^{-3u}}{1 + \frac{u}{3}} du = \int_0^1 \frac{1}{1 + \frac{u}{3}}(3e^{-3u}) du = \int_0^1 \frac{1}{1 + \frac{u}{3}} f(u) du = E_X\left[\frac{1}{1 + \frac{x}{3}}\right] \tag{2.3}$$

Table 2.2: results for MC integration with exponentially distributed random variable.

N	Result	Standard Deviation
100	0.8941728578421076	0.08457872752248434
1000	0.902889021116287	0.08135148342547543
10000	0.9069282989007331	0.07902699766430729
100000	0.9076925594774227	0.07867954211948758

Important: the random numbers generator here is RANLUX by Martin Lüscher generate numbers which include 0 but exclude 1.

In table 2 is the result of the Monte Carlo integration using the exponentially distributed random variables.

3 STUDENT'S DIET: MARKOV MATRIX

- **States and Markov matrix**:

For the vector of states:

$$\vec{v} = \begin{bmatrix} P_p \\ P_b \end{bmatrix} \tag{3.1}$$

where P_p is the probability of eating pizza, and P_b is the probability of eating burrito, for a certain day.

Then, for the Markov matrix:

$$P = \begin{pmatrix} P_{p \to p} & P_{b \to p} \\ P_{p \to b} & P_{b \to b} \end{pmatrix} = \begin{pmatrix} 0.15 & 0.7 \\ 0.85 & 0.3 \end{pmatrix} \tag{3.2}$$

Where, for example, $P_{10} = 0.85$ represents the probability of eating pizza in a day, and in the next day eating burrito.

- **Eating pizza following Tuesday:**

If the student has pizza:

$$\vec{v} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.3}$$

and for 2 days after:

$$\vec{v}_{Tuesday} = P^2 \vec{v} = \begin{pmatrix} 0.15 & 0.7 \\ 0.85 & 0.3 \end{pmatrix}^2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.6175 \\ 0.3825 \end{bmatrix} \tag{3.4}$$

which implies that $P_p = 0.6175$ for Tuesday.

- **Long-term behavior:**

  In the long term ($n = 50$ is used just as an example here, but the result for that case was compared to $n = 40$ and $n = 60$, and it agrees):

$$P^{50} = \begin{pmatrix} 0.15 & 0.7 \\ 0.85 & 0.3 \end{pmatrix}^{50} \approx \begin{pmatrix} 0.4516 & 0.4516 \\ 0.5484 & 0.5484 \end{pmatrix} \tag{3.5}$$

  which implies:

  – probability of having pizza = 0.4516

  – probability of having burrito = 0.5484

  i.e. those two probabilities stabilized, and (in the long-term) the probability of having pizza is a little bit lower than that of having burrito.

4 MARKOV CHAIN: PREDATOR-PREY SYSTEM

- **states of the system:**

  for an instant in time, there is a probability for evolving the system, initially in the state (x, y), to one of the following states:

- (x, y)

- $(x, y+1)$

- $(x, y-1)$

- $(x+1, y)$

- $(x-1, y)$

- **on irreducibility:**

  The system is not irreducible, as there isn't a probability to go from any one of the possible states to another. The set of probabilities is limited, as there is a zero probability (from the statement of the problem), of going from any of the states that is not (x, y), to another.

- **normalization factor:**

  As the probability of something happening is 1, then:

  $$\frac{1}{c}(s + \alpha y + \beta xy + \gamma xy + \delta x) = 1$$
  $$\Rightarrow c = \frac{1}{s + \alpha y + \beta xy + \gamma xy + \delta x} \tag{4.1}$$

- **long-term when $\beta = \gamma = 0$:**

  in case those two coefficients are set to zero, then no interaction term is present. From the usual way of imlpementing the prey-predator problem (with the use of differential equations), the mathematical way of representing an interaction between those two objects, is through the use of terms of the form xy (i.e. a product of the current number of elements in each set: predator and prey sets).

- **implementation of the Markov chain:**

  - no reduction in any species:

    $P = \frac{s}{c}$ if:

    $$\vec{v}_{t+1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \vec{v} \tag{4.2}$$

– +1 only in prey:

$P = \frac{\alpha y}{c}$ if:

$$\vec{v}_{t+1} = \begin{pmatrix} 1 & 0 \\ \frac{1}{x} & 1 \end{pmatrix} \vec{v} \qquad (4.3)$$

– reduction only in prey:

$P = \frac{\beta x y}{c}$ if:

$$\vec{v}_{t+1} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{x} & 1 \end{pmatrix} \vec{v} \qquad (4.4)$$

– +1 only in predator:

$P = \frac{\gamma x y}{c}$ if:

$$\vec{v}_{t+1} = \begin{pmatrix} 1 & \frac{1}{y} \\ 0 & 1 \end{pmatrix} \vec{v} \qquad (4.5)$$

– reduction only in predator:

$P = \frac{\delta x}{c}$ if:

$$\vec{v}_{t+1} = \begin{pmatrix} 1 & -\frac{1}{y} \\ 0 & 1 \end{pmatrix} \vec{v} \qquad (4.6)$$

Then, after 1 step of the Markov chain process:

$$\vec{v}_{t+1} = \frac{1}{c} \begin{pmatrix} s + \alpha y + \beta xy + \gamma xy + \delta x & \gamma x - \frac{\delta x}{y} \\ \frac{\alpha y}{x} - \beta y & s + \alpha y + \beta xy + \gamma xy + \delta x \end{pmatrix} \vec{v}$$

$$\Rightarrow \vec{v}_{t+1} = \begin{pmatrix} 1 & \frac{\gamma x - \frac{\delta x}{y}}{s + \alpha y + \beta xy + \gamma xy + \delta x} \\ \frac{\frac{\alpha y}{x} - \beta y}{s + \alpha y + \beta xy + \gamma xy + \delta x} & 1 \end{pmatrix} \vec{v} \tag{4.7}$$

Previous matrix is not a Markov matrix, clearly, as it doesn't obey the the restriction $\sum_{j=1}^{N} P_{ij} = 1$, but the implementation corresponds to the evolution of the predator-prey system.

The previous evolutionary system was implemented in Python, in the script located at prob4/markov_chain_sim.py.

In the end, after 2011 iterations, all the predators die, and there are only preys left (a total of 1094 preys left).