Probability Density Estimation: Finding the PDF of a Random Variable

TOGAN TLIMAKHOV

Corresponding author: Togan Tlimakhov (e-mail: t0gan@icloud.com).

ABSTRACT Parametric probability density estimation is a technique that involves selecting a common distribution and estimating the parameters for the density function from a data sample. In this paper, we are dealing with an experiment in which it is required to find a probability density function. We start by simulating the experiment considered in order to obtain information about the behavior of the system and further assign certain values such as mean and standard derivation for the sample's distribution, which will be obtained based on certain assumptions rather than randomly selecting trivial values. After obtaining a relatively rational sample of 100 experiments, we assume no previous knowledge of the data sample and try to estimate its probability density function using parametric density estimation. Finally, we give some perspective for a further comprehensive approach and end with a conclusion of the work.

INDEX TERMS Density Estimation, Probability, Random Variable, Simulation.

I. INTRODUCTION

THE probability density function (PDF) is one of the most important concepts in probability theory, it is used to define a continuous random variable's probability coming within a distinct range of values, as opposed to taking on any single value. A continuous random variable is a random variable that has a real numerical value where each numerical outcome of a continuous random variable can be assigned a probability. The relationship between the events for a continuous random variable and their probabilities is called the continuous probability distribution and is summarized by a probability density function. The function can explain the PDF of normal distribution and how mean and deviation exists. There are many other common continuous probability distributions such as the simplest uniform distribution. The most common is the normal (Gaussian) probability distribution. Practically all continuous probability distributions of interest belong to the so-called exponential family of distributions, which are just a collection of parameterized probability distributions (e.g. distributions that change based on the values of parameters) [1].

Knowing the probability distribution for a random variable can help calculate moments of the distribution, like the mean and variance, but can also be useful for other considerations, such as determining whether an observation is very unlikely and might be an outlier or anomaly. The problem is, we may not know the probability distribution for a random variable, as we rarely know the distribution since we don't have access to all possible outcomes for a random variable. In fact, all we have access to is a sample of observations. As such, we must select a probability distribution. This problem is referred to as probability density estimation, or simply "density estimation", as we are using the observations in a random sample to estimate the general density of probabilities beyond just the sample of data we have available [4].

In this paper, we will try to find the probability density function of a given random variable using density estimation. The random variable is not provided per se, however an experiment setup is given where a random variable is constructed from 100 samples that we will be generating using certain assumptions about the considered experiment after preforming a simulation of similar system.

II. A PROBLEM DEFINITION

A problem to consider is defined as follows; given a ball with diameter about 5 cm. Consider the experiment that a ball is dropped from 40 cm above the ground to a specific point and measuring the range from the center point to the last location that ball reached. Repeat this experiment 100 times and save all the range measurements. Now, suppose that range between center point and the last location of the ball is a random variable. Find the probability density function of the defined random variable. In the next sections, we will try answering the equation of finding the PDF of the random variable.

III. MODELING A FREE-FALLING BALL

As we have a limited amount of information about the parameters that affect the behavior of the system, we aim in this section to model a similar system in order to understand how falling beads behave and deduce a mean value that we will be using later to construct the dataset with 100 sample experiments.

We simulate two different objects with different masses and weights in a vacuum where there is no air, therefore the opposite resistive force of the air is not presented. In this case, we realize that the two objects are falling at the same rate and they reach the ground at the same time no matter from which height they were dropped, i.e. both objects fall at the same rate regardless of their masses, cross-sectional area, and height. The motion equations regarding this experiment are given as follow:

$$d = \frac{1}{2}gt^2 \ , \quad t = \sqrt{\frac{2d}{g}} \ , \quad v = \sqrt{2gd} \tag{1}$$

Where d is the distance traveled in t seconds, t is the time taken for an object to fall distance d and v is the instantaneous velocity of a falling object. Note that all three equations can be deduced from Newton's laws of motion. F=P, where P=mg, and g is the acceleration of gravity on earth calculated approximately 9.807 m / s² [5].

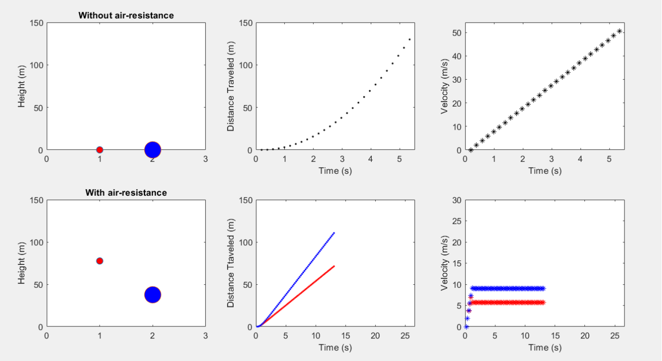

FIGURE 1. An instant of the trajectory where both objects are still falling at a different rate with air-resistance.

The simulation with air resistance naturally implies that the beads will fall on different rates which makes the problem more complex. For simplicity one might proceed with the first simulation. The reason we model the falling bead in the first place is to understand how different unknown parameters such as air resistance can affect the experiment outcome. After the bead reaches the ground for the first time it will rebounce again losing some of its initial energy. The amount of energy loss might depend on multiple factors, for simplicity, we assume that the bead falls and rebounds to 3/4 of the initial height.

Based on this assumption, we now calculate the total vertical distance traveled by the bead up to the third bounce as follow: $D = h(0) + (3/4).h(0) + (3/4).(3/4).h(0)$ which is *2.3125 h(0)* and for $h(0) = 40$ *cm* the total distance traveled is *92.5 cm*. This value can be considered the max distance or within the longest 10% (i.e. top 90%-99% on the distribution) the bead can travel. Hence a good mean distance can be around $D = 45$ *cm*, where we assume the horizontal distance is marginally close to the vertical distance traveled.

It is important to notice that the value D obtained is based on a set of assumptions such as the number of bounces till rest, horizontal distance deduced from the vertical distance, etc. that might not be necessarily true for every real-life experiment. We use these assumptions in order to deduce a rational value rather than merely randomly picking one. The values obtained can also be calculated using geometric progression equations which leads to similar results [2].

IV. CONSTRUCTING THE SAMPLE DATA

Since we now have a mean value for the random variable, we can construct 100 sample data using random number generators such as the *rand* function in MATLAB. However, before doing so, we still need to determine two properties, the standard derivation and the type of distribution.

Regarding the choice of distribution, we will draw a rational conclusion on which type to chose. There are many different classifications of continuous probability distributions. Some of them include the normal (Gaussian) distribution, uniform distribution, Chi-square, etc. Based on our experiment, choosing a normal distribution seem to be the best option. For instance, if we choose a uniform distribution, it will imply that the probability of reaching 45 cm will be the same as reaching 100 cm. which is not true as in reality the bead will fall less in a further area than in the mid-point area due its energy limitation.

On the other hand, if we choose an exponential distribution, it will imply that the further/closer the distance range is the higher the probability which is also not true and disobeys the laws of thermodynamics. Hence a normal distribution seems the best option as it implies that there is a certain range (i.e. mean) where the bead falls on the most, and probability decreases as we move further from this range. Which is a rational outcome that can be obtained in a real-life experiment.

The standard deviation on the other hand is a measure of the amount of variation or dispersion in the sample values. i.e. a low standard deviation indicates that the values tend to be close to the mean of the set, while a high standard deviation indicates that the values are spread out over a wider range. Determining the standard deviation might be more tricky since we have few clues about the experiment setup. Hence we will choose a generic variance that might be comprehensive for most natural experiments that don't involve complex external effects. Since we have a mean of 45 cm and the longest 10% is around *92.5 cm* which will help us determine the standard deviation, and from the following standard deviation formula:

$$\sigma = \sqrt{\frac{\sum (x_i - \overline{x})^2}{n-1}} \tag{2}$$

Where x_i is the value in the data distribution, $\overline{x}$ is the sample mean and n is the total number of observations.

Trying different values for σ, we can observe as in Figure 2 that a standard deviation of 15 produces a distribution where the values are spread out over a wider range reaching 90 cm on the right which was around the max range. The histogram of the sample data with a mean of 45 cm and a standard deviation of 15 is shown in Figure 2.

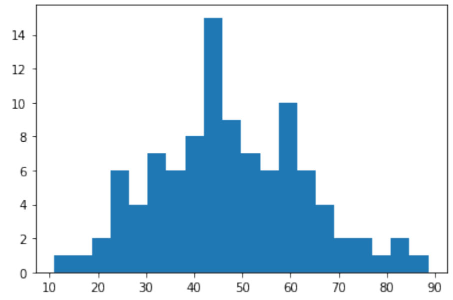

FIGURE 2. Histogram of the generated sample data with N(45, 15)

V. PARAMETRIC PROBABILITY DENSITY ESTIMATION

After we have constructed our 100 sample dataset with the experiment outcome. We now assume that we have no information regarding the dataset as if it was given to us without any prior knowledge. We start by observing the histogram plot of the dataset in order to get information about the type of distribution.

We can observe that the distribution is Gaussian-like as expected. The Gaussian distribution has two parameters: the mean and the standard deviation. Given these two parameters, we can determine the probability distribution function. These two parameters can be estimated from data by calculating the sample mean and sample standard deviation. The process of estimating these parameters is referred to as parametric density estimation.

Using the Python library Numpy's *numpy.mean* and *numpy.std* functions, we calculate the mean and standard deviation of the sample dataset, which is the estimate for the parameters of the normal probability distribution. and then print the results as follow:

$$Mean = 47.41, \; Standard\ Deviation = 15.62$$

After estimating the parameters, we fit the distribution with these parameters using the parametric density estimation of our data sample. We use the Python library SciPy's *scipy.stats.norm* function, which provides a normal continuous random variable.

Afterward, we sample the probabilities from this distribution for a range of values in our domain, in this case between $\{10, 90\}$ and plot a histogram of the data sample with an overlapping line plot of the probabilities calculated for the range of values from the probability density function shown in Figure 3.

It is important to notice that we can convert the counts or frequencies in each bin of the histogram to a normalized probability in order to ensure the y-axis of the histogram matches the y-axis of the line plot. This can be achieved by setting the "density" argument to "True" in the call to hist() function [4].

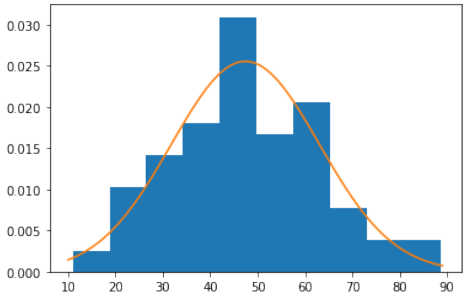

FIGURE 3. Histogram of the generated sample data with N(45, 15)

In order to calculate the probability of bead falling in certain range, for example between 30 and 60, we can use the PDF estimated to preform the calculations as follow:

$$P(30 < X < 60) \; = \; P(30 - 45 < X - \mu < 60 - 45)$$

$$= \; P(\frac{30-45}{15} < \frac{X-\mu}{\sigma} < \frac{60-45}{15})$$

Where

$$Z = \frac{x-\mu}{\sigma} \;\Rightarrow\; \frac{30-45}{15} = -1 \; and \; \frac{60-45}{15} \; = 1$$

$$\Rightarrow P(30 < X < 60) \; = \; P(-1 < Z < 1) \; = 0.6826$$

Giving a 68.26% probability of the bead stopping on the range of 30 and 60 cm. The final probability is calculated from the linear transformation by standardizing the random variable X, and calculating the area with Z values from the standard normal table [1].

We can compare the probability obtained from the estimated PDF with the initial sample data from the histogram by summing all frequency values from 30 to 60 and divide by the total 100 as follow:

$$X = \{"range\ between\ 30\ and\ 60"\} \; = \; \{30, 31, ..., 60\},$$

for the histogram in Figure 2 with 20 bins;

$$P(X) \; = \; \frac{7+6+8+15+9\;+7\;+6+10}{100} \; = \; 0.68$$

VI. A MORE COMPREHENSIVE APPROACH WITH TWO RANDOM VARIABLES

Since the solution provided above only assume the position with one random variable -namely the distance R from the center, this approach does not provide a completely accurate description as the final position of the bread requires at least two variable x and y. And for more complicated systems one might require to introduce the z-dimension as well, yet for our problem, that might cause unnecessary complexity for modeling the system.

We might assume that only two variables describe the position; x and y. In case we want to add the z-direction, this will imply that the ground isn't flat and there are hole or/and incline with a certain slope. The issue with introducing the z-dimension is that the slope of the ground will also affect the position of x and y i.e. the variable x and y will become dependent on z, for example, if the ball falls on a downhill slope, this will cause the bread to further accelerates downwards due to gravitational force. Since we don't have any information about the slope of the ground and in order to simplify the model, we can assume a flat ground, there the z-dimension is ignored. The density analysis might require the usage of joint PDF for the two random variable.

An alternative for the Cartesian coordinate system (x, y) could be the polar coordinate system (r, θ), which will describe the position in terms of distance r and angle θ. Since a function of a random variable is another random variable, we can easily convert between the two coordinate systems. We can proceed with the Cartesian coordinates for our model.

It might also be useful to notice that the sample data set i.e the 100 observations in our experiment might have a unimodal distribution, such as the familiar bell shape of the normal, the flat shape of the uniform, or the descending or ascending shape of an exponential or Pareto distribution. We might also observe complex distributions, such as multiple peaks that don't disappear with different numbers of bins, referred to as a bimodal distribution, or multiple peaks referred to as a multi-modal distribution. We might also see a large spike in density for a given value or a small range of values indicating outliers, often occurring on the tail of distribution far away from the rest of the density [4].

VII. CONCLUSION

In conclusion, we have started by introducing a simulation of a falling ball in order to understand the behavior of the experiment. Then we try to construct a set of 100 sample data using a set of rational assumptions rather than randomly deducing a sample data which might not be realistic leading to trivial results. After we construct the sample data, we assume no prior knowledge and try to estimate the mean and standard deviation parameters using the parametric probability density estimation technique with the help of python libraries Numpy and SciPy. Finally, we plot both the histogram of the data sample with an overlapping line plot of the probabilities calculated for the range of values from the probability density function. In the end, we provide a possible further extension of this work, where we suggest using two random variables to specify the location of the ball, giving a possibly more accurate yet more complex estimation of the probability density function.

ACKNOWLEDGMENT

The work in this paper is not officially published. The code for Probability Density Estimation is included in Appendix A. For MATLAB files of falling bead simulation and other related documents visit the GitHub repo.

REFERENCES

[1] Bertsekas, D. and Tsitsiklis, J., 2008. Introduction to probability. 2nd ed. Belmont, Mass.: Athena Scientific.

[2] Kay, S., 1993. Fundamentals of statistical signal processing Vol. I : Estimation Theory. Englewood Cliffs, NJ: Prentice-Hall.

[3] Law, A., 2015. Simulation modeling and analysis. New York: McGraw-Hill Education.

[4] Brownlee, J., 2020. Probability for Machine Learning - Discover How To Harness Uncertainty With Python, Edition: v1.9. Machine Learning Mastery.

[5] Velten, Kai. (2009). Mathematical Modeling and Simulation: Introduction for Scientists and Engineers. 10.1002/9783527627608.

APPENDIX A

The following Python Notebook code includes the algorithm for the Parametric Probability Density Estimation [4].

```python
import matplotlib
import numpy as np
import scipy as sp
from matplotlib import pyplot
from numpy.random import normal
from scipy.stats import norm

# Generate random sample of 100 experiments from a
# normal distribution with a mean of 45 and a
# standard deviation of 15.
sample = normal(loc=45, scale=15, size=100)

# Plot a histogram of the sample
pyplot.hist(sample, bins=20)
pyplot.show()
```

The above snippet outputs Figure 2.

```python
# Calculate the parameters from the sample data
cal_mean = np.mean(sample)
cal_std = np.std(sample)
print(’Mean = %.2f, Standard Deviation = %.2f’
% (cal_mean, cal_std))
```

The above snippet outputs the following:
Mean = 47.41, Standard Deviation = 15.62

```python

# Using calculated parameters, define distribution

# A normal continuous random variable.
dist = norm(cal_mean, cal_std)

# Sample the probabilities for a spesific range
# (10 to 90) of experiment outcomes
values = [value for value in range(10, 90)]
prob = [dist.pdf(value) for value in values]

# Plot the histogram and probability density fun.
pyplot.hist(sample, bins=10, density=True)
pyplot.plot(values, prob)
pyplot.show()
```

The above snippet outputs Figure 3.