Uncovering Probability Models:

A Cross-Disciplinary Exploration

Volume 01

Edited By:

I.M.L. Nadeesha Jayaweera

Welcome!

Welcome to the mini journal, where we embark on an illuminating exploration of probability modeling and its transformative applications in various domains. In this concise journey, I endeavor to shed light on the multifaceted nature of statistical concepts acquired from our Probability for Application class. These concepts, far from being confined to the realms of theory, emerge as dynamic tools with tangible implications across a spectrum of fields, including but not limited to Computer Science, Engineering, Marketing, and Biology.

At the heart of probability lies the science of uncertainty; a fundamental pillar for quantifying the likelihood of events in an unpredictable world. Through meticulous study and practical application, we've sharpened our skills to develop models that not only forecast outcomes but also serve as guiding beacons amid the complexities of decision-making in ambiguous scenarios.

As avid students of Probability for Applications, we recognize the paramount importance of data in shaping our perceptions and actions. Yet, within the vast expanse of information, uncertainty looms ominously. Herein lies the power of probability models—they offer clarity amidst the fog of unpredictability, providing actionable insights and paving the way forward in the face of uncertainty.

Throughout this journal, we'll delve into concrete examples that showcase the invaluable role of probability modeling in our respective fields. From assessing risk in financial markets to predicting patient outcomes in healthcare settings, the applications are as diverse as they are impactful. Through meticulous analysis and insightful interpretation, we'll witness firsthand how probability models empower us to navigate the intricate landscapes of our domains, fostering informed decision-making and driving meaningful change.

I invite you to join us on this captivating journey as we unravel the mysteries of probability modeling. Together, let's embark on a quest for not only theoretical enlightenment but also tangible insights with real-world implications, forging a path towards innovation and excellence in our chosen fields.

Thank You!

I.M.L. Nadeesha Jayaweera
Co-authors: Probability for Application Students (Spring D term, 2024)
Department of Mathematical Sciences
Worcester Polytechnic Institute (WPI)

Acknowledgment

I would like to express my deepest gratitude to everyone who contributed to the successful completion of the Women's Impact Network (WIN)-funded Open Educational Resources (OER) Development grant project at WPI, particularly in the creation of the mini journal *"Uncovering Probability Models: A Cross-Disciplinary Exploration"*, based on the mini projects of Probability for Applications students.

First and foremost, I am immensely thankful to all my students in MA2621-D24-DL03 Probability for Applications at WPI for their hard work and for presenting several examples from their fields to demonstrate the importance of using probability models in real applications. Their hard work and dedication were instrumental in shaping the content of this mini journal.

I am also indebted to Prof. Marja Bakermans and Ms. Lori Ostapowicz-Critz, the mentors of the OER project, whose constructive feedback and dedicated mentorship were pivotal in refining the educational resources. Their commitment to fostering open education initiatives has been truly inspiring. I extend my appreciation to Vikranth Vilas, our diligent computer assistant, for his technical expertise and assistance with the project.

Furthermore, I extend my appreciation to all the individuals who provided encouragement, feedback, and assistance at various stages of this endeavor. Your collective efforts have played an integral role in making this project a success.

Lastly, I would like to acknowledge the funding support provided by the EMPOwER (Engaging More Powerfully, Openly with Educational Resources) grant at WPI, which made this project possible and contributed to the broader goal of advancing open educational resources in our academic community.

Thank you all for your unwavering support, guidance, and commitment to enhancing educational resources for our students.

I.M.L. Nadeesha Jayaweera
Department of Mathematical Sciences
Worcester Polytechnic Institute (WPI)

Content

Content Cont...

Content Cont...

Quality Assurance in Software Engineering Using Binomial Distribution

Tanya Ali

Introduction:

Ensuring the quality of software products is paramount in software engineering. A useful statistical approach in this domain is the binomial distribution. This distribution serves as a robust tool for modeling the occurrence of defects within samples extracted from software releases. In this project, we aim to construct a probability model employing the binomial distribution to scrutinize quality assurance in software engineering practices.

Background Information:

Let's envision a scenario where a software development team is working on a new application. To uphold software quality standards, a subset of features or lines of code is systematically chosen from each release for meticulous inspection. The presence or absence of defects within each feature or line of code can be effectively represented using the binomial distribution. This distribution characterizes the count of successful outcomes (defective features or lines of code) within a predetermined number of independent trials (individual inspections), where each trial possesses an identical probability of success (probability of encountering a defective feature or line of code). Probability Model: Consider X as the variable denoting the count of defective features or lines of code within a sample of size n, each with a probability p of being defective.

Probability Model:

The probability mass function (PMF) of the binomial distribution is articulated as follows:

$$P(X = k) = \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k}; k = 0, 1, \ldots, n.$$

where:

- k represents the count of defective features or lines of code.
- n signifies the sample size.
- p indicates the probability of a feature or line of code being defective.

Probability Calculation:

Let's assume we randomly select a sample of 50 lines of code from a software release, and the known probability of a line of code being defective is 0.02. Our objective is to determine the probability of exactly 3 lines of code in the sample being defective. Utilizing the binomial distribution, we can compute this probability utilizing the PMF:

$$P(X = 3) = \binom{50}{3} \cdot 0.02^3 \cdot (1-0.02)^{50-3} = 0.1857.$$

Conclusion:

This project endeavors to establish a probability model leveraging the binomial distribution to scrutinize quality assurance practices in software engineering, particularly in assessing the likelihood of encountering a specific count of defective features or lines of code within a sample. By employing the binomial distribution, we computed the probability of observing precisely 3 defective lines of code within a sample of 50 lines. Such a probability model holds significant implications for software development teams in enhancing product quality and reliability.

Reference:

Montgomery, D. C., & Runger, G. C. (2013). Applied Statistics and Probability for Engineers (6th ed.). John Wiley & Son.

Predicting Cyber Attacks Using Binomial Distribution

Jivan Baghsarian

Introduction:

Every year, IBM produces a cyber threat intelligence report that shows statistics about real cyber-attacks that occurred during the reported year. These reports are statistics from the past and they are not predictive. However, they are one of the best tools we have to predict what the future may look like. In this project, we will calculate the probability that future attacks are specifically phishing, to give us some insight into predicting future attacks. We will use the Binomial model to achieve this.

Information:

Based on the IBM report, 30% of all cyber attacks last year were phishing. A company uses this information and needs to know the probability of this same outcome occurring for the following year, so they can adjust their cyber defense infrastructure cost effectively. The binomial distribution shows the probability of getting a phishing attack given that 30% statistic.

Probability Formula:

$$P(X = k) = \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k}; \quad k = 0, 1, \ldots, n.$$

where n is the sample size, p is the probability of a phishing attack, and k is the number of phishing attacks.

Probability Calculation:

A company has been attacked 100 times in the past month by various cybercrime. 30% of those attacks were phishing. What is the probability that exactly 30 of those attacks were phishing.

$$P(X = 10) = \binom{100}{30} \cdot (0.3)^{30} \cdot (1-0.3)^{(100-30)} = 0.08678$$

Conclusion:

Our binomial distribution calculations showed that there is an 8.68% chance that there will be exactly 30 phishing attacks. Companies can use this information to more accurately design their cyber security infrastructure in a more cost-effective manner based on the probability of this exact scenario occurring.

References:

1. https://en.wikipedia.org/wiki/Binomialdistribution
2. https://www.ibm.com/reports/threat-intelligence

Probability Model for Electronic Devices

Claire Bitner

Introduction and Background Information:

Electronic devices play a significant role in everyday life for many people. These devices allow us to continue making progress in the fields of science, engineering, business, etc. Since these devices are so important to our society, it is also important to know what their lifetimes look like as well as their probability of failure. A model that is commonly used to study and summarize these ideas is the exponential model. Looking at the exponential model and other probability distributions along with their modifications allows for an analysis of the reliability for each different electronic device based on its own unique characteristics. The exponential distribution model with no modifications is not entirely accurate for each electronic device since it does not include increasing and decreasing failure rates. The exponential distribution can still provide data that can help give a rough summarization of reliability for these cases though.

Probability Model:

The exponential distribution will model the time between when the electronic device was manufactured and when it fails. Let X, a continuous random variable, represent the time between the creation and failure of the electronic devices. The average time of failure will be $1/\lambda$, and λ represents the rate parameter. That is $X \sim Exp(\lambda)$: Then the pdf of X is:

$$f(x) = \begin{cases} \lambda \cdot e^{-\lambda \cdot x} & ; x \geq 0 \\ 0 & ; \text{otherwise} \end{cases}$$

Probability Calculation:

Suppose the time between when a phone is manufactured and when it dies is modeled using the exponential distribution model, and the average time between those two events is 5 years. To find the probability that the phone will last 4 to 6 years can be calculated using the PDF of the exponential distribution.

$E(X) = 5 = 1/\lambda \implies \lambda = 0.2$. So,

$$P(4 \leq X \leq 6) = \int_4^6 0.2 \cdot e^{-0.2x} = 0.1481.$$

Reference:

Ali, S., Ali, S., Shah, I., Siddiqui, G. F., Saba, T., & Rehman, A. (2020). Reliability Analysis for electronic devices using generalized exponential distribution. IEEE Access, 8, 108629–108644.

Analysis item acquisition rate in gacha games using geometric distribution.

Qiushi Chen

Introduction:
A gacha game is a video game that implements the gacha mechanism. Similar to loot boxes, gacha games entice players to spend in-game currency to acquire random in-game items. In this case, the acquisition rate of rare items is often very low, but the low single extraction price will drive the player to spend money to extract, and in the end, it will often cost a lot of money to obtain rare items.

Background Information: In certain gacha games, players can "pull" from a character pool to obtain virtual characters of varying rarity. One of the most sought-after categories are characters rated as the highest rarity, which are typically the rarest and most powerful. The mechanics of these pulls can be modeled using a geometric distribution. In this model, each pull is an independent trial with a fixed probability of success, which is the chance of drawing a character of the highest rarity. This probability is usually low, for example, 1%. The geometric distribution effectively captures the number of trials needed to achieve the first success in such scenarios, making it a suitable mathematical representation for analyzing the likelihood of obtaining rare characters in gacha games.

Probability Model: Let X represent the number of trials required to achieve the first success in a sequence of independent Bernoulli trials. Each trial has a fixed probability p of success, representing an event such as drawing a character of the highest rarity from a gacha pool. The probability that the first success occurs on or before the k trial is given by the formula:

$$P(X \leq k) = 1 - (1-p)^k$$

- k is the number of trials until the first success
- p is the probability of pulling a character of the highest rarity from the available pool of characters in a single draw

Probability Calculation:

Suppose probability of pulling a character of the highest rarity from the available pool of characters in a single draw is 2%. We want to find the probability we gain a character of the highest rarity in 50 pulls. We can calculate this probability using the equation below:

$$P(X \leq 50) = 1 - (1-0.02)^{50} = 0.6358.$$

Conclusion:

In this study, we developed a probability model using the geometric distribution to analyze the dynamics of rare event occurrences in gacha games, specifically the drawing of characters of the highest rarity from a pool of characters. We calculated the probability of drawing at

least one character of the highest rarity within 50 pulls, using a success probability of 2%. This probability model provides valuable insights for players and game developers alike, aiding in understanding and optimizing the gacha mechanics to balance player satisfaction and game profitability.

Reference:

Toto, Serkan. "Gacha: Explaining Japan's Top Money-Making Social Game Mechanism". Serkan Toto: CEO Blog. Kantan Games. Retrieved 10 April 2020

Probability Model in the Stock Market

Gabriel D'Amour

Introduction:

Analysis of the stock market is all about trends and probability. As a data scientist, this is where I will have to thrive. Investing in the stock market is a great way to let your money work for you over time. One of two things can happen in the market. The market can either have a bullish trend (upward trend), or a bearish trend (downward trend). The more ideal situation would be the bullish trend, signifying that your investment is increasing in value while bullish trend signifies a decrease in value. While there are many factors at play to whether a stock increases in value or decreases in value, we will not worry about them for the sake of this project. It has been proven multiple times by many investors that the best way to increase your odds in gaining profit from the market is to have portfolio diversification. This essentially means that you do not put all your money into one stock and instead spread your wealth.

Example / Probability Model

Let's consider this example: An investor is diversifying their portfolio between Stocks and Bonds. They put 50% of their portfolio capital into Stocks and 50% portfolio capital into Bonds. Given some historical data about the market, there is a 60% chance that the STOCK market will perform bullishly and a 40% chance that it performs bearishly. Additionally, there is a 70% chance that the BOND market performs bullishly and a 30% chance that it performs bearishly. What is the probability that the investor's portfolio performs bullishly?

Probability Model:

Let $A_1, A_2, \ldots, A_n$ be a partition of Ω. Then for any event B,

$$P(B) = \sum_{i=1}^{n} P(B|A_i) \cdot P(A_i)$$

where

- B is any event
- n is the sample size
- A_i represents the different events forming a partition of the sample space

Probability Calculation:

In this case the sample size is 2, representing investing in Stocks or Bonds. The conditional probability that the portfolio performs bullishly given that the investor has chosen to allocate their portfolio to Stocks is 0.60. The conditional probability that the portfolio performs bullishly

given that the investor has chosen to allocate their portfolio to Bonds is 0.70.

Let W = Bullish, S = Stocks, and B = Bonds

$$\begin{aligned} P(W) &= P(S) \cdot P(W|S) + P(B) \cdot P(W|B) \\ &= 0.50 \cdot 0.60 + 0.50 \cdot 0.70 = 0.65 \end{aligned}$$

Conclusion:

In this project we developed a probability model using the Total Probability Theorem to analyze the change of success in an investors investment portfolio. We calculated the probability that the investor's portfolio performs bullishly. This probability model provides valuable insights for investors investing.

Reliability Analysis in Electrical and Computer Engineering
Jeremy De La Cruz

Background of Electrical and Computer Engineering

Electrical and Computer Engineering (ECE) encompasses a broad spectrum of disciplines, focusing on the principles and applications of electricity, electronics, and electromagnetism. This field is instrumental in designing and optimizing electrical systems, electronic devices, and computer technologies. Probability is a critical aspect of ECE, as engineers need to ensure sustained functionality and performance of systems over time.

Probability Model

The following probability model I developed centers on evaluating the reliability of systems comprising multiple independent components. The model uses foundational probability concepts and Bayes' theorem to facilitate the computation of the system's probability of functioning given the individual probabilities of component reliability.

Key components of the probability model include:

- *Component Reliability Probabilities:* These probabilities represent the likelihood of each system component functioning or failing independently, where the reliability of one component does not influence the reliability of another. In the model, we assign reliability probabilities to the CPU, memory module, and power supply. It's important to mention that real-world scenarios may exhibit inter-dependencies among components.

- *Bayes' Theorem Application:* Bayes' theorem serves as the mathematical framework for computing the conditional probability of system functionality based on the states of individual components. It facilitates the integration of new information, such as component states, into the probability assessment. At its core, Bayes' theorem relates the conditional probability of an event A given event B, $P(A|B)$ to the conditional probability of event B given event A, $P(B|A)$. This is represented in the following formula:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

In the subsequent section, we will illustrate the application of this probability model through a specific example involving a computer system with multiple critical components.

Model:

You have a computer system with three critical components: CPU, memory module, and power supply. The probabilities of each component being functional (F) or nonfunctional (NF) are as follows:

- $P(F_{CPU}) = 0.8$ (CPU functioning)
- $P(F_{Memory}) = 0.85$ (Memory module functioning)
- $P(F_{Power}) = 0.9$ (Power supply functioning)

Given that all three components (events) are independent and functioning, we want to calculate the overall probability that the system is not functioning: $P(NF_{System}) \cdot P(A|B)$.

- First, we calculate $P(B)$, the probability that all three components are functioning simultaneously.

$$P(B) = P(F_{CPU}) \cdot P(F_{Memory}) \cdot P(F_{Power}) = 0.8 \cdot 0.85 \cdot 0.9 = 0.612.$$

- Next, we calculate P(A), the probability that the system is not functioning, which is the complement of the system functioning:

$$P(A) = 1 - P(B) = 1 - 0.612 = 0.388.$$

- Now, we use Bayes' Theorem to find $P(A|B)$, the probability that the system is not functioning given that all 3 components are functioning:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} = \frac{1 \cdot 0.388}{0.612} = 0.633$$

since if all the components are functioning, then the system is functioning is $P(B|A) = 1$.

Therefore, given that all three components are functioning and independent, the probability that the system is not functioning is approximately 0.633 or 63.3%.

Conclusion:

This project delved into probability models and Bayes' theorem's application in Electrical and Computer Engineering's reliability analysis. Through this study, we've gained insights into assessing system reliability, emphasizing the value of probabilistic approaches in engineering decision-making and system design.

Reference:

Devore, J. L. (2018). Probability and statistics for engineering and the Sciences. Nelson.

A probability model for determining fraudulent bank transactions

Vivek Jagadeesh

Introduction:

One application of probability in the computing industry comes in the form of computer-generated models which determine wheter or not a credit card or banking transaction is fraudulent. Banks utilize a binomial distribution with a particular parameter p, which is determined based on a massive amount of data collected by both Banks and government officals on bank fraud. They then apply the distribution to understand how often it is for a transaction to be fraudulent, allowing the bank to perform audits accordingly, and to recuperate any money lost due to fraudulent activity.

Background Information:

Consider a bank with 25customers, each performing 2 bank transactions on an ordinary day. In an effort to reduce fraud, the bank wants to understand the probability of there being a fraudulent transaction. Each bank transaction can be represented as a binomial distribution, with each transaction either ebing fraudulent, or legitimate. The distribution models the number of fraudulent transactions in a given day, assuming that 1% of transactions at the bank are fraudulent based on historical data. All of the bank's customers are of the same character, and each trial is identical to the others.

Probability model:

Let the discrete random variable X be the number of fraudulent transactions in a sample size of 400 transactions, and P be the probability that any one of the transactions is fraudulent. This yields the following probability mass function for the distribution:

$$P(X = x) = \binom{n}{x} \cdot p^x \cdot (1-p)^{n-x}; \quad x = 0, 1, \ldots, 400.$$

- n is the number of transactions in a single day
- x is the number of fraudulent transactions in a single day
- p is the probability of a fraudulent transaction.

Calculation:

Suppose the bank wants to find the probability that there were exactly 4 fraudulent transactions in an ordinary day where there were 50 transactions(so that it can report it to the authorities).

$$P(X = 4) = \binom{50}{4} \cdot 0.01^4 \cdot (1 - 0.01)^{50-4} = 0.00145$$

Therefore, the probability that four out of 50 bank transactions are fraudulent is, using a binomial distribution with the probability of success set to 0.001450.

Conclusion:

In this project, I developed a probability model which models the number of fraudulent bank transactions using the binomial distribution. The binomial distribution was an easy choice, since each trial is identical and has the same chance of success. We calculated the probability of exactly four out of fifty bank transactions being fraudulent. This probability model will provide critical information to both banks and authorities to understand the nature of fraudulent transactions.

Reference:

Bobbitt, Zach. "5 Real-Life Examples of the Binomial Distribution." Statology, 14 July 2021, www.statology.org/binomial-distribution-real-life-examples/

Probabilistic models in the neural action potential timing

Sophia Kouznetsov

Introduction

In the field of neuroscience, one application of probabilistic models is in the neural action potential timing. Neurons communicate with each other through electrical impulses called action potentials or spikes. These spikes are brief, rapid changes in membrane potential (rapid depolarization) that propagate along the neuron's axon. The timing of these spikes carries information about the particular stimulus the neuron is responding to, as well as the neuron's role in neural circuits.

Probabilistic application

The timing of neural spikes follows a Poisson distribution, particularly in spontaneous firing rates. Models of neural coding often incorporate Poisson distributions to describe the stochastic nature of spike timing (that is, involving randomness or variability). The stochastic nature arises from various sources, including the random arrival of synaptic inputs, the stochastic opening and closing of ion channels, and intrinsic noise within the neuron.

In a Poisson distribution, events occur randomly and independently over time, with a constant average rate of occurrence defined as parameter λ. In the context of neural spike timing, the Poisson distribution can be used to model the probability distribution of spike times, where λ corresponds to the average firing rate of the neuron in response to sensory stimuli. It is important to note that neural spike trains often exhibit deviations from strict Poisson statistics due to various factors such as refractory periods, burst firing behavior, and circuit interactions.

Probabilistic Model using Poisson Distribution

Probability Mass Function (PMF):

$$P(X = x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}; \quad x = 0, 1, 2, \ldots.$$

- $P(X = x)$: probability of observing 'x' spikes in a given time interval.
- λ: average rate of spikes per unit time interval.
- x: number of spikes observed in the given time interval.

Example

Suppose a neuron has an average firing rate of 50 spikes per second (50 Hz). The probability of observing a spike within a small time interval 't' is λ. Consider a specific trial where the neuron fires spikes at the following times (in seconds) after stimulus onset: 0.02, 0.06, 0.1, 0.15, 0.2, 0.24, and 0.28. We can analyze these spike times using a Poisson model to estimate the neuron's firing rate and assess whether the observed spike train is consistent with

a Poisson distribution with $E(X) = \lambda = 50Hz$. By calculating the inter-spike intervals and comparing the observed distribution to the expected distribution from a Poisson distribution with the expected firing rate, we can evaluate the goodness-of-fit of the Poisson model and gain insights into the neural coding properties of the neuron. λ= 50 Hz (spikes/second),$t = 1$ second.

Inter-spike intervals (ISIs): $ISI_i = t_{i+1} - t_i$

$$
\begin{aligned}
ISI_1 &= t_{1+1}–t_1 = 0.06–0.02 = 0.04 \text{ seconds} \\
ISI_2 &= t_{2+1}–t_2 = 0.1–0.06 = 0.04 \text{ seconds} \\
ISI_3 &= t_{3+1}–t_3 = 0.15–0.1 = 0.05 \text{ seconds} \\
ISI_4 &= t_{4+1}–t_4 = 0.2–0.15 = 0.05 \text{ seconds} \\
ISI_5 &= t_{5+1}–t_5 = 0.24–0.2 = 0.04 \text{ seconds} \\
ISI_6 &= t_{6+1}–t_6 = 0.28–0.24 = 0.04 \text{ seconds}
\end{aligned}
$$

Estimated firing rate of this sample,

$$\hat{\lambda} = 1/mean(ISIs) = 1/0.0417 = 24Hz$$

$24Hz < 50Hz$ ($\hat{\lambda} < \lambda$) indicates that the observed spike train is not consistent with a Poisson distribution with $\lambda = 50Hz$. Specific implications of slowed neural spike time encompass:

1. Altered information encoding and processing: The stochastic nature of spike timing allows neurons to encode information in an efficient and effective manner. By modulating the timing and rate of their spikes – temporal coding -, neurons can convey a wide range of sensory, cognitive, and motor information efficiently. Slower firing of action potentials may reduce the rate at which information is transmitted within neural circuits, thereby impacting the speed and accuracy of sensory processing and cognitive function (i.e., attention, memory, decision-making abilities).

2. Impaired motor function: A decreased firing rate of neurons in motor pathways can result in impaired movement coordination and slower reaction times. This can manifest as difficulties in fine motor tasks, decreased muscle responsiveness, and overall deficits in motor performance.

3. Reduced plasticity: Neural plasticity is the ability of the brain to adapt and change in response to experience and relies on activity-dependent modulation of synaptic strength. Slower firing rates may reduce the frequency of synaptic activation, which may result in decreased levels of synaptic plasticity and, consequently, deficits in learning and memory.

4. Neurological disorders: Abnormalities in neural action potential firing are implicated in various neurological and psychiatric disorders, including epilepsy, neurodegenerative diseases, schizophrenia, and depression, by way of alterations in neuronal excitability and firing patterns.

Conclusion

This example demonstrates how probabilistic models, such as the Poisson distribution, can be applied to analyze and interpret neural spike timing data, providing valuable information about the underlying neural mechanisms and information processing in the brain.

References

1. Grider MH, Jessu R, Kabir R. Physiology, Action Potential. [Updated 2023 May 8]. StatPearls Publishing. https://www.ncbi.nlm.nih.gov/books/NBK538143/

2. Softky, W. R., & Koch, C. (1993). The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. Journal of Neuroscience, 13(1), 334-350.

Probability Model in Data Science/ Business Analytics

Luz Joseph

Introduction:

The analysis and decision-making based on the information analyzed is a major portion of data science and its conjunction with business analytic. In the context of business, the use of probability modeling can be complex, requiring the analysis of large amounts of data and its translation into business value. These translations can come in the form of model improvement and forecasting, which we will focus on. In the event of predicting or preparing a business for the situation of allocating a reasonable amount of product or marketing efforts for a new product, forecasting is used, which is connected to the Poisson Distribution definition of finding the probability of an event happening within an interval of times for a certain number of times.

Background Information:

Consider a music company like Spottily coming out with a new feature for its Premium users. To ensure that the number of Premium user account activation may increase for that quarter, a possible projection of the number of acquired users may be asked of the deployment team to ensure that its deployment is accurately planned.

Probability Model:

Let X represent the number of acquired Premium users for that quarter given the average, 1, increases in users after deployment. The Poisson Distribution of X would be:

$$P(x = x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}; \quad x = 0, 1, 2, \ldots.$$

where λ is the average number of (parameter) and x is the actual number of (random variables).

Probability Calculation:

Suppose the average number of new users after a feature deployment is 50. Let X be the number of acquired users in the quarter after a new feature deployment is 50.

$$P(x = 50) = \frac{e^{-50} \cdot 50^{50}}{50!} = 0.056325.$$

Conclusion:

For this project, we had to develop a model and scenario to of how the Poisson distribution is used in the context of Data Science/ Business Analytic, specifically for the scenario of Spotify's deployment of a new feature. Assuming that all aspects are set for a successful deployment, we found that there is a 6% probability that there will be exactly 50 newly acquired Premium users after a month of the new feature being deployed. This probability model provides powerful

insight into possible customer projections in assessment of their product deployments.

Reference:

Montgomery, D. C., & Runger, G. C. (2013). Applied Statistics and Probability for Engineers (6th ed.). John Wiley & Sons.

Modeling Nonlinear Wind Turbine Oscillations Under Non-stationary Wind Excitation

Komninos Georgios

Introduction:

Wind turbines, which use wind energy to create electricity, are crucial parts of renewable energy systems. However, because of the nonstationary nature of the wind and possible nonlinearities in the system, wind turbines are susceptible to a variety of oscillations. The performance and lifespan of the turbines may be impacted by these oscillations. In order to study the oscillations of a nonlinear wind turbine, we will create a probability model in this project by applying the ideas of equivalent linearization and nonstationary random excitation.

Background Information:

Nonlinearities in the structural dynamics of the wind turbine and the random and non-stationary character of wind direction and speed are the main causes of oscillations in wind turbines. Using Priestley's description of an evolutionary process, we shall characterize the wind excitation as a nonstationary random process and use equivalent linearization to account for the nonlinearities.

Probability Model:

Consider the equation of motion for a nonlinear wind turbine model with one degree of freedom:

$$\ddot{X} + \beta\dot{X} + \omega_n^2 X + \epsilon h(X, \dot{X}, t) = F(t)$$

where:

- X is the displacement of the wind turbine's nacelle.
- β is the linear damping coefficient.
- ω_n is the natural frequency.
- ϵ is a small parameter.
- $h(X, \ddot{X}, t)$ is a nonlinear function.
- $F(t)$ is the non-stationary wind excitation.

We define F(t) as an evolutionary process using the Fourier-Stieltjes integral:

$$F(t) = \int_{-\infty}^{\infty} A_F(t, \omega) \cdot \exp(i\omega t) dZ(\omega)$$

The frequency-time modulating function is denoted by $A_F(t, \omega)$, and the orthogonal-increment process with certain parameters is represented by $dZ(\omega)$.

Equivalent Linearization:

We apply the analogous linearization technique to account for the nonlinearity. An equivalent linear term, $\alpha_e X + \beta_e \dot{U}$, replaces the nonlinear component $\epsilon h(X, \ddot{w}, t)$. The determination of α_e and β_e is achieved by minimizing the mean square error between the original and equivalent systems.

Response Characterization:

The response of the equivalent linear system to the non-stationary excitation can be obtained using the Duhamel integral:

$$X(t) = \int_0^t g(t-\tau)F(\tau)d\tau$$

where $g(t)$ is the impulse response function of the equivalent linear system. The response's evolutionary power spectral density can be calculated by utilizing the excitation's frequency-time modulating function and the analogous linear system's transfer function.

Conclusion:

In this study, I analyzed the oscillations of a nonlinear wind turbine subjected to non-stationary wind loads by developing a probability model based on the ideas of equivalent linearization and non-stationary random excitation. An evolutionary process was used to describe the wind excitation, and equivalent linearization was used to account for the nonlinearity. For wind turbine builders and operators, this probability model offers important insights into how the system behaves oscillatorily under actual wind conditions and nonlinearities.

Reference:

1. Ahmadi, G. (1980). Mean square response of a Duffing oscillator to a modulated white noise excitation by the generalized method of equivalent linearization. Journal of Sound and Vibration, 71(1), 9-15.

2. Probability Models in Engineering and Science, Haym Benaroya, Seon Mi Han, Mark Nagurka CRC Press. (ISBN 0824723155, 9780824723156).

Normal Distribution in Robot Forward Kinematic

Quincy Laflin

Introduction:

Forward kinematics in robotics is the process of calculating the position and orientation of a robot based on given parameters. This is known as a pose. Given two poses, one being the current pose and one being a pose in the future, a course can be plotted between the two poses. For instance, given that our robot starts at $(x = 0, y = 0, heading = North)$, and our desired pose is $(x = 4, y = 5, heading = South)$, our robot needs to figure out a path from the pose to the second pose. The process is summarized as,

1. Plan motion from A to B
2. Calculate target wheel speeds
3. Control the wheel speeds
4. Track progress by updating the pose
5. Repeat

Background:

Sensitivity analysis applies specifically to wheeled robots. It is used to understand how variation in input parameters impacts the output (robot trajectory). Using sensitivity analysis we can predict the behavior of a wheeled robot moving along a path. To start, sensitivity analysis helps us understand which variables impact the robot's final position the most. This allows us to better model robotic systems, which leads to better robotic designs overall. The process helps us understand sources of error that would cause a robot to misbehave. These sources include a stepper motor being faster than the OEM specs, a malfunctioning sensor returning value outside of the expected range, or even a wheel slipping. Sensitivity analysis can also help predict how setting the wheels to different velocities affects the robot's path.

Conducting Sensitivity analysis:

1. Define something that needs to be modeled
2. Identify key parameters
3. Vary parameters
4. Observe changes in output
5. Using the data collected model the variation in a normal distribution
6. Run a Monte Carlo simulation: Randomly sample wheel speed from the normal distribution and observe the effect on output.

7. Perform an error propagation analysis. Calculate how errors in velocity analysis affect the controlling equation.

Probability model:

- V_i = velocity of the wheel I
- $\mu(i)$ = average velocity of wheel I
- $\Sigma(i)$ = standard of deviation of wheel i, represents variability PMF: For each wheel $V_i \sim N(\mu_i, \sigma_i^2)$
  - $X(t) = X(t) = X(0) + \int 0t(V1(s) + V2(s))ds$
  - $X(0)$ is the initial position, and $V1(s)$ and $V2(s)$ are the velocities of the wheels at times s.

Example:

Problem: Imagine a differential drive robot (a type of mobile robot with two separately driven wheels on either side). We want to analyze the effect of the wheel velocities on the robot's straight-line travel over a period of time.

- Left wheel (VL): mean (μ_L) of 5 m/s and a standard deviation (σ_L) of 0.1 m/s.
- Right wheel (VR): mean (μ_R) of 5.2 m/s and a standard deviation (σ_R) of 0.1 m/s.

Calculate the expected position of the robot along a straight path after 10 seconds, taking into account the variability in wheel velocities. Additionally, estimate the probability that the difference in velocities leads to the robot deviating from its intended straight path. Calculate the probability that the difference in velocities exceeds a certain threshold, say 0.2 m/s, which we assume could significantly affect the robot's path.

- Foreward kinematics: $V_{avg} = (V_L + V_R)/2$
- Average $V = \mu_{avg} = (\mu_L + \mu_R)/2$
- Expected position after 10 sec: $X(10) = V_{avg} \times 10$
- Deviation: $\Delta V = V_R - V_L$
- Variance: $Var_{avg} = (Var(V_L) + Var(V_R)/2, \sigma_{avg} = (\sigma_L^2 + \sigma_R^2)/4$

Solution:

- The expected position of the robot along a straight path after 10 seconds, based on the average velocities of the wheels, is 50.5 meters.
- The probability that the difference in velocities between the left and right wheels exceeds 0.2 m/s (which could significantly affect the robot's path) is approximately 23.98% 24% chance that the robot might deviate significantly from a straight path due to the difference in wheel velocities.

$$\text{Mean of variance: } \mu\Delta V = \mu_R - \mu_L, \sigma\Delta V = \sigma_L^2 + \sigma_R^2$$

$$P(\Delta V > \text{ threshold}) = 1 - CDF(\text{ threshold}; \mu\Delta V, \sigma\Delta V)$$

Where: $\Delta V = 0.2m/s$ and CDF of a normal distribution with the given parameter.

References:

1. Altuzarra, Óscar, and A. Kecskeméthy, eds. Advances in Robot Kinematics 2022. Cham, Switzerland: Springer, 2022. Print.
2. Di Gregorio, Raffaele, and Raffaele Di Gregorio. Kinematics and Robot Design II (KaRD2019) and III (KaRD2020). Basel: MDPI - Multidisciplinary Digital Publishing Institute, 2022. Print.
3. Yan, Ying. "Error Recognition of Robot Kinematics Parameters Based on Genetic Algorithms." Journal of ambient intelligence and humanized computing 11.12 (2020): 6167–6176. Web

Developing a Probability Model for Cyber-security Risk Assessment using Bernoulli Trials

Alexander Lap

Introduction:

In cybersecurity, probability is essential for evaluating threats and arriving at wise conclusions. The Bernoulli trial, which is used to model binary outcomes like success or failure in a single experiment, is one of the basic ideas in probability theory. In this project, we will use Bernoulli trials to construct a probability model for cybersecurity risk assessment, with a particular focus on the likelihood of a successful cyberattack under specified circumstances.

Background Information:

In the field of cybersecurity, it is critical for businesses to evaluate the likelihood of a successful cyberattack in order to put in place appropriate security measures. Bernoulli trials are a useful tool for modeling the likelihood of a successful cyberattack since they represent individual attacks with a predetermined success rate. Numerous variables, including attacker sophistication, security protocols, and system vulnerabilities, can affect this chance.

Probability Model:

Let's refer to the successful cyberattack that occurs during a specific time frame as event A. With a success chance of p, we can use a Bernoulli trial to model occurrence A. For a Bernoulli trial, the probability mass function (PMF) is provided by:

$$P(A) = p \text{ if } A \text{ occurs }, P(A) = 1 - p \text{ if } A \text{ does not occur}$$

System vulnerabilities, threat intelligence, and historical data can all be used to calculate the success probability p.

Probability Calculation:

Assume that past data indicates a monthly probability of $p = 0.03$ for a successful cyberattack on a system. We wish to determine the likelihood that within the next month, there will be at least one successful cyberattack.. Using the Bernoulli trial model, we can calculate this probability as follows:

$$P(\text{at least one successful cyberattack}) = 1 - P(\text{No successful cyber attack}) = 1 - (1 - p)^n$$

where n is the number of trials (months) considered.

For example, if we consider $n = 30$ days in a month, the probability of at least one successful cyberattack in the next month is:

$$P(\text{at least one successful cyberattack}) = 1 - (1 - 0.03) \cdot 0.30 = 0.785$$

Conclusion:

For the purpose of this study, we created a probability model that employs Bernoulli trials to evaluate cybersecurity risk. Specifically, we concentrated on the likelihood that a cyberattack will be effective given a particular success rate. Organizations are better able to assess and reduce cybersecurity risks by using historical data and probabilistic modeling. This improves security protocols and increases resistance to cyberattacks.

Reference:

Montgomery, D. C., & Runger, G. C. (2013). Applied Statistics and Probability for Engineers (6th ed.). John Wiley & Sons.

Products' quality control in APPLE company
Yahong Liu

Introduction: APPLE is an American computer and consumer electronics company famous for creating the iPhone, iPad, and Macintosh computers etc. One common statistical tool used in quality control is the binomial distribution. This distribution is frequently used to model the number of defective items in a sample from a production batch, such as products of iPhone, iPad, and Macintosh computers.

Background Information: Consider an APPLE company producing Mac computers in a company. To ensure product quality, A sample of Mac computers is selected from each production batch and inspected for defects. Quality of production batch can be described by using a binomial distribution because there are two possibilities: 1) defective 2) good. The binomial distribution describes the number of successes (defective items) in a fixed number of independent Bernoulli trials (individual inspections), where each trial has the same probability of success (probability of a defective item).

Probability Model:

Let X represent the number of defective components in a sample of size n, each with a probability p of being defective. The probability mass function (PMF) of the binomial distribution is given by:

$$P(X = x) = \binom{n}{x} \cdot p^x \cdot (1-p)^{n-x}; \quad x = 0, 1, \ldots, n.$$

where:

- x is the number of defective components.
- n is the sample size.
- p is the probability of a defective component.

Probability Calculation:

Suppose a company produces only Mac computers in this situation. The probability that Mac computers produced by the company are defective is 0.01. If 10 Mac computers are produced, what is the probability that exactly one of them will be defective?

$$P(X = 1) = \binom{10}{1} \cdot (0.01)^1 \cdot (1-0.01)^9 = 0.09135 = 9.135\%$$

Conclusion:

In this project, we developed a probability model by binomial distribution to analyze the quality of a Mac computer in a manufacturing environment. specifically, the probability of finding a certain number of defective Mac computers in a sample. We calculated the probability of exactly

1 defective Mac computer in a sample of 10 Mac computers using the binomial distribution. In sum, this probability model provides insight for APPLE company faculty in assessing the Mac computer only. If the APPLE company faculty wanted to assess only the other kind of product such as iPad or Macintosh computers.

Reference:

https://clubztutoring.com/ed-resources/math/binomial-distribution- definitions-examples-6/

Data Corruption From Faulty RAM Using Exponential Distribution
Andrew Melton

Introduction:

Random Access Memory (RAM) is a piece of hardware inside a computer made up of millions of tiny capacitors and transistors. RAM serves as a short term memory system to the computer as it is used to process and save data that is actively being used by the Central Processing Unit (CPU). Unfortunately, when RAM fails, the system's memory and data is corrupted and the CPU suffers because of this. One of the leading causes of faulty RAM is age. In this model we will use an Exponential Distribution to model the amount of time (in years) it takes for a stick of RAM to fail/break.

Background Information:

Let's consider normal RAM for our example i.e. not performance/gaming RAM, but rather RAM that would be in a normal office computer. Let's assume we have a company that produces this RAM. We want to look into the details of the lifetime of this RAM (how much time the user has until the RAM fails). An exponential distribution is often used to model the time until an event occurs in a continuous time frame given a constant failure rate. For our failure rate, we use λ. $E(X)$ is calculated as such for the exponential distribution, $1/\lambda$. Similarly, $VAR(X)$ is given by $1/\lambda^2$.

Probability Model:

Let T represent the time it takes for a stick of RAM from our company to fail and let λ represent the inverse of the mean lifetime of a stick of RAM from our company. Thus $T \sim Exp(\lambda)$. The probability density function (PDF) for T is as follows:

$$f(t) = \begin{cases} \lambda e^{-\lambda t} & ; t \geq 0 \\ 0 & ; \text{otherwise.} \end{cases}$$

where

- t is an input representing the time (in years) since the RAM stick was put into use.
- λ represents the inverse of the mean lifetime of a stick of RAM from our company.

Similarly, the Cumulative Density Function (CDF) is as follows:

$$F(t) = P(T \leq t) = 1 - e^{-\lambda t}$$

Probability Calculation (PDF):

Suppose we know that a stick of RAM from our company lasts, on average, 5 years. Thus and if then it follows that $E(X) = 5 = 1/\lambda \implies \lambda = 0.2$. We want to know the probability that a given stick of RAM lasts for exactly 6 years. We calculate this using our PDF:

$$P(X = 6) = f(6) = 0.2 \cdot e^{-0.2 \times 6} = 0.060238.$$

Probability Calculation (CDF):

Using the same situation from the calculation above, we now want to know the probability that a given stick of ram lasts for at most 10 years. We calculate this using our CDF;

$$P(T \leq 10) = F(10) = 1 - e^{-0.2 \times 10} = 0.864664.$$

Conclusion:

In this project, we developed a probability model using an exponential distribution to analyze the lifetimes for RAM sticks inside of a computer. Specifically, calculating these lifetimes in years from the first time the RAM stick was put into use. We made two calculations using our PDF and CDF. PDF: Found the probability of a stick of RAM from our company lasting exactly 6 years (≈6%). CDF: Found the probability of a stick of ram from our company lasting at most 10 years (≈86%). This model serves to provide information to not just big companies and industries, but computer users all over the world.

Probability Modeling used in Manufacturing
Savannah Miller

Introduction and Background

In manufacturing, specificity is important. Product designs need to meet specific requirements in order to pass assessments. These products are considered and assessed individually to ensure all products meet the standard. Therefore, if a product that is being manufactured does not meet the requirements, it will not be considered or will be rejected. One tool used to help control these rejections is the Third Axiom of Probability. This is often used in manufacturing to predict the probability of these defects if the events that cause the defect are independent. In this project, we will make a probability model using The Third Axiom of Probability. This probability will look at independent events as diameters of a shaft being greater than or less than the desired or nominal value. In the end, we will be able to determine the probability that the shaft will be rejected.

Probability Model

(The Third Axiom of Probability) For two events E_1, E_2, that are mutually exclusive, that is, $E_1 \cap E_2 = \emptyset$, the probability of the occurrence of either or both events is given by

$$Pr(E_1 \cup E_2) = Pr(E_1) + Pr(E_2)$$

Probability Calculation

Machine shafts are being manufactured. The shaft can be rejected if the diameter is less than 98% or greater than 102% of its nominal value. The "nominal" value is the desired or design value. The probability that a shaft is being rejected because the diameter is less than 98% of its nominal value is given as 0.02 and the probability that the shaft is being rejected because the diameter is greater than 102% of its nominal value is given as 0.015. What is the probability that a shaft will be rejected?

Solution: Let E_1be the event that the diameter of the shaft is less than 98% of its nominal value, and E_2 be the event that the diameter of the shaft is greater than 102% of its nominal value. The probability of each event is $Pr(E_1) = 0.02, Pr(E_2) = 0.015$. A shaft will be rejected if either E_1 or E_2 occurs. That is, the probability that a shaft will be rejected equals the probability of the union E1 and E2, that is, $Pr(E_1 \cup E_2)$. Since the shaft diameter cannot be too small and too small at the same time, the events E1 and E2 are mutually exclusive, and using the third axiom of probability,

$$Pr(E_1 \cup E_2) = Pr(E_1) + Pr(E_2) = 0.02 + 0.015 = 0.035$$

Reference

Probability Models in Engineering and Science, Haym Benaroya, Seon Mi Han, Mark Nagurka CRC Press. (ISBN 0824723155, 9780824723156)

Clustering in Machine Learning Using Gaussian Mixture Model

Matthew Montero

Introduction/Background:

A useful probabilistic model to cluster intricate data patterns is the Gaussian Mixture Model (GMM). GMMs use multiple Gaussian distributions to cluster – that is, grouping similar data points in a dataset – to provide probabilistic insights into how different data is spread out. For example, imagine a two-dimensional graph with a length and width as axes, representing a map of Earth with individuals as data points. Each person represents a single interest in a social network. In the real world, people would have different interests in various social networks, but assume the former for the sake of simplicity and explanation. On the map, many individuals are spread out in various regions, and it seems visually apparent that various sections of the Earth represent groups of individuals with the same interest. If a company wanted to effectively market its product to people whose interests align with their product, instead of marketing to everyone which may incur more losses due to overextended marketing, they may employ a Gaussian Mixture Model to cluster individuals by their interests and help the company determine the best areas to market and reach more customers.

Probability Model:

Consider the earlier example with a company selling gaming personal computers (PCs) and the dataset of people with only two interests: PCs and gaming consoles. Assume those who want gaming consoles will not buy PCs; the company wants to market in areas of the Earth that are likely – that is, have a higher density of – to buy their PCs. To use a Gaussian Mixture Model, a probability density function (PDF) must be set up, in which the following parameters are initialized.

- mean (μ): mean of the distribution
- covariance (Σ): covariance matrix of the distribution

From these values, create a PDF of a multivariate Gaussian distribution, $X \sim N(\mu, \Sigma)$,

$$f(x; \mu, \Sigma) = \frac{1}{(2\pi)^{k/2} \cdot |\Sigma|^{1/2}} \cdot e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

where:

- x: represents data points
- K: number of components or dimensions of each data point

e.g., when $K = 2$, each data point has two components. Once this is set up, someone may use this Gaussian Mixture model to find the probability of a data point that belongs to a cluster, such as the strength of whether it belongs to PC or console gamers. For example, refer to the previous example of the company wanting to find people interested in PCs. Assume $K = 2$; the following parameters would be assigned these values:

- $\mu = [\mu_1, \mu_2]$
- $\Sigma = \begin{bmatrix} \sigma_{1,1}^2 & \sigma_{1,2}^2 \\ \sigma_{1,2}^2 & \sigma_{2,2}^2 \end{bmatrix}$

where:

- μ_i: mean of X_i
- $\sigma_{i,j}^2$: covariance of X_i and X_j

After this, the company may use this PDF to find the probability of data points that come from each distribution or cluster. Note that after the setup of the PDF, it is necessary to refine the parameters to find the optimal values for the PDF to find more accurate probabilities of the data points. This requires differentiation of the equation with respect to the mean and covariance, found in a step named the Maximum Likelihood Estimation (MLE). Otherwise, using the PDF only after the setup would yield inaccurate probabilities.

Conclusion:

This project gives a general overview of the setup for a Gaussian Mixture Model. GMMs can be used to cluster data points by how probabilistic a data point is associated with a specific element of a cluster. Given this, GMMs are useful at associating data points sorted in complex arrangements or shapes, and they can help remove bias when associating data points to a cluster by giving a probability than a label to any data point, such as in K-means clustering. GMMs are useful in many applications such as in medical diagnosis, climate analysis, and image processing.

References:

1. Carrasco, O. C. (2019, June 3). Gaussian Mixture Models Explained. Medium.
2. https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95
3. McGonagle, J., Pilling, G., Dobre, A., Tembo, V., Kurmukov, A., Chumbley, A., Ross, E., & Khim, J. (n.d.). Gaussian Mixture Model. Brilliant. https://brilliant.org/wiki/gaussian-mixture-model/
4. (2024, March 20). Clustering in Machine Learning. GeeksforGeeks.
5. https://www.geeksforgeeks.org/clustering-in-machine-learning/

Store Returns in Retail Using Binomial Distribution

Carter Moore

Introduction:

For retail stores around the world, the number of returns they receive can negatively impact their pro;it. To help test the ef;icacy of their products, we can use the binomial distribution to model the number of returns customers make. In this project, I will develop a probability model using the binomial distribution to analyze returns at a retail store.

Background Information:

Consider a large department store that sells a wide range of products. To ensure customers are happy with their purchases, a sample of products is selected from each week's sales data and tracked to see how many are returned. The decision to return an item or not can be modeled using a binomial distribution. The binomial distribution describes the number of successes (returned products) in a fixed number of independent Bernoulli trials (individual products), where each trial has the same probability of success (probability of a return being made).

Probability Model:

Let X represent the number of products sold in a sample of size n, each with a probability p of being returned. The distribution can be written as:

$$X \sim Binomial(n, p)$$

$$P(X = x) = \binom{n}{x} \cdot p^x \cdot (1-p)^{n-x}; \quad x = 0, 1, 2, \ldots, n.$$

The probability mass function (PMF) of the binomial distribution is given by: Where: x is the number of returned products. n is the sample size. p is the probability of a returned product.

Probability Calculation:

Suppose a sample of 50 products from the store are randomly selected from a week of sales data, and the probability of a product being returned is known to be 2%. We want to ;ind the probability that exactly 3 products in the sample were returned. Using the binomial distribution, we can calculate this probability using the PMF:

$$P(X = x) = \binom{50}{x} \cdot 0.02^x \cdot (1-0.02)^{50-x}; \quad x = 0, 1, 2, \ldots, 50.$$

Then, $P(X = 3) = \binom{50}{3} \cdot 0.02^3 \cdot (1-0.02)^{50-3} = 0.0607$.

Conclusion:

In this project, we developed a probability model using the binomial distribution to analyze product return data in a retail setting, specifically the probability of ;inding a certain number of returned products in a sample. We calculated the probability of 3 returned products in a sample of 50 retail products using the binomial distribution. This probability model provides valuable insights for retailers in analyzing how product returns affect their business model.

Reference:

Bobbitt, Zach. "5 Real-Life Examples of the Binomial Distribution." Statology, 14 July 2021, www.statology.org/binomial-distribution-real-life- examples

Probability Model for Anomaly Detection in Network Traffic

Viet Thanh Nguyen

Introduction:

Anomaly detection is critical to ensuring the security and reliability of computer networks, especially in an era of increasing cyber threats. Detecting unusual patterns or behaviors in network traffic can aid in the identification of potential security breaches, malicious activities, or system failures. In this paper, we present a probability model for detecting anomalies in network traffic, which employs data science techniques to analyze and identify abnormal network behavior.

Background Information:

Network traffic refers to the vast amount of data exchanged between devices in a network, including communication between servers, clients, routers, and other network devices. This traffic can be monitored and analyzed to extract a variety of information, including source and destination IP addresses, port numbers, packet sizes, protocols, and times. Anomalies in network traffic can appear as deviations from expected patterns in these features, indicating potential security risks, network failures, or performance problems. Creating an effective probability model allows for automated detection of such anomalies, which improves network security and operational efficiency.

Probability Model:

The probability model for detecting anomalies in network traffic assumes that normal network behavior follows a specific statistical distribution, whereas anomalies deviate significantly from this distribution. We use a multivariate Gaussian distribution to simulate the probability density function (PDF) of network traffic features under typical operating conditions. The PDF is defined as follows:

$$f(x;\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}\cdot|\Sigma|^{1/2}}\cdot e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

Where:

- X represents the vector of network traffic features.
- μ is the mean vector of the features.
- Σ is the covariance matrix capturing the relationships between different features.
- d is the dimension of the feature space.

Using this PDF, we can calculate the probability of observing a specific network traffic pattern X under normal conditions and identify deviations from expected behavior that may indicate anomalies.

Probability Calculation:

Given a new observation x that represents a specific network traffic pattern, we can calculate its probability using the normal operating conditions model. If this probability falls below a predetermined threshold, the observation is classified as anomalous. The threshold can be adjusted to achieve the desired balance of false positives and false negatives, depending on the network environment.

Conclusion:

In this paper, we have proposed a probability model for anomaly detection in network traffic based on a multivariate Gaussian distribution. By analyzing network traffic features and comparing them to the modeled normal behavior, anomalies can be detected and flagged for further investigation or mitigation. This approach leverages data science techniques to enhance network security, reliability, and performance in modern computer networks.

Reference:

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

Optimal Maintenance Using Weibull Distribution

Lulu Ouyang

Introduction:

Optimal maintenance strategies are critical in the later stages of mechanical engineering to ensure reliable and efficient operation of machinery and equipment and to save costly losses due to inappropriate maintenance schedules. Probabilistic models are crucial in determining appropriate maintenance intervals and predicting the likelihood of failure or breakdown. A widely used probabilistic model in this regard is the Weibull distribution. Modeling with it allows for more efficient maintenance scheduling and intervals.

Background Information:

Consider a manufacturing factory that produces industrial machinery and equipment. To ensure the damage rate and longevity of these machines. From the first batch of machines in use, a record of the time of each damage is kept. The data is used to make two probability curves, β, and α, as well as a histogram. Finally, the pattern of damage can be identified from the distribution of the data in order to develop an optimal maintenance schedule.

Probability Model:

The Weibull distribution is characterized by two parameters: the shape parameter (β) and the scale parameter (η). The shape parameter determines the behavior of the failure rate over time, while the scale parameter represents the characteristic life or the time at which 63.2% of the components or systems have failed.

$$f(t) = (\beta/\eta) \cdot (t/\eta)^{(\beta-1)} \cdot e^{-(t/\eta)^{\beta}}$$

where:

- t is the time or age of the component or system
- β is the shape parameter ($\beta > 0$)
- η is the scale parameter ($\eta > 0$)

Probability Calculation:

Suppose we have a factory with a Weibull distribution shape parameter $\beta = 3$ and scale parameter $\eta = 5000$ hours. We want to calculate the probability that the component will fail before 4000 hours. The cumulative distribution function (CDF) of the Weibull distribution is given by:

$$F(t) = 1 - e^{-(t/\eta)^{\beta}}$$

Use the data: $F(4000) = 1 - e^{-(4000/5000)^3} = 1 - e^{-0.512} = 0.401$.

Therefore, the probability that the component will fail before 4000 hours is approximately 0.401 or 40.1%.

Conclusion:

In this calculation, we find that the probability of a machine breaking down is 40.1% when used for less than four thousand hours. This probability gives the factory the likelihood that the machine will continue to work and also gives the factory continuous data on the machine's usage so that the factory can decide in real-time if repairs are needed. This will improve the efficiency of the factory.

Reference:

Modarres, M., Kaminskiy, M., & Krivtsov, V. (2017). Reliability Engineering and Risk Analysis: A Practical Guide (2nd ed.). CRC Press.

Predicting Software Bug Counts Using Deep Learning

Aditya Patel

Introduction/Background:

The BCV-Predictor, as described by Sushant Kumar Pandey and Anil Kumar Tripathi, utilizes deep learning to predict the bug count vector of a software system's successive version. The predictor employs metadata created by concatenating different versions of the same software system. This project aims to provide a simplified probability model using this context, focusing on the distribution of bug occurrences in the software system.

Probability Model:

Model Setup: Let variable X represent the number of bugs in a given software module. Assume X follows a Poisson distribution model, which is typical for modeling the count of occurrences (bugs) that happen independently within a fixed interval (module).

Parameters: λ: the rate parameter, which represents the average number of bugs per module. This can be estimated from historical big data.

Model Definition: The probability mass function (PMF) for the Poisson distribution is given by:

$$P(X = x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}; \quad x = 0, 1, 2, \ldots.$$

where x is the number of bugs, and e is the base of the natural logarithm.

Probability Calculation:

Suppose historical data suggests an average of 3 bugs per version ($\lambda = 3$). We can calculate the probability of observing exactly 4 bugs in the next version as follows:

$$P(X = 4) = \frac{(e^{-3} \cdot 3^4)}{4!} = 0.168$$

which means there's about a 16.8% chance of encountering exactly four bugs.

Practical Application:

- *Risk Assessment:* Helps managers decide where to focus testing efforts by showing which parts of the software might have more bugs.
- *Planning:* Assists in scheduling the right amount of time for fixing bugs and maintaining the software, based on predicted bug counts.
- *Quality Assurance:* Enhances software quality by directing quality checks towards the areas most likely to have issues.

Conclusion:

This project developed a probability model using the Poisson distribution to predict the number of bugs in a software system's next version. The model provides a framework that software development teams can use to anticipate potential issues and allocate resources efficiently.

References:

Pandey, S. K., & Tripathi, A. K. (2020). BCV-Predictor: A bug count vector predictor of a successive version of the software system. Knowledge-Based Systems, 197, 105924. ISSN 0950-7051. https://doi.org/10.1016/j.knosys.2020.105924.

Modeling Genetic Variation through discrete random variables

Akansha Pathak

Introduction:

To study biological diversity and inheritance of certain traits, bioinformaticians and computational biologists have studied components of genetic variants in order to actively model genome-wide genetic variation among individuals. Using Discrete Random Variables to analyze and quantify genetic variation through differences in alleles provides a powerful framework for bioinformatics and computational biology.

Background:

Genetic variation encompasses a wide variety of mutations and genotypic/phenotypic differences with traits, such as height, weight, and biochemical concentration. Some of these variations include single nucleotide mutations (such as SNP's), which involve changing one single gene as a mutation. Others include indel that encompass insertions/deletions within a genomes and copy number variations (CNV's) which is when there is a duplicate copy made of a single gene, that can cause a frameshift mutation during translation. SNP's involve single base pair mutations and are commonly studied due to their association with phenotypic traits and diseases. For this specific probability model we will be looking at a specific population to map a genome of specific allele frequencies. Developing a probability model centered around discrete random variables will help researchers analyze distinct allele frequencies and their probabilities for a specific population represent by n, or the sample size of people in a population.

Probability Model:

Let a genetic variation in this case be at a specific genomic locus, denoted as discrete random variable X, and all of the possible outcomes are $x_1, x_2, \ldots, x_k$, which represents different alleles that are present in the population. In this case, the probability mass function (PMF) would describe the likelihood of observing an allele. This model can also be extended to encompass the analysis of joint distributions through linkage disequilibrium.

Probability calculation:

Suppose we have the genotype for a population of genes at a specific genomic locus. By utilizing multinomial distribution, which is the binomial distribution for multiple outcomes, we can compute probabilities relating to mutated allele outcomes in the population. The multinomial distribution for this situation is:

$$P(n_1, n_2, \ldots, n_k) = \frac{n!}{n_1! n_2! \ldots n_k!} \cdot p_1^{n_1} \cdot p_2^{n_2} \cdots p_k^{n_k}$$

where:

- n represents the total number of people with either this allele or a different allele in place

- n_k represents the count of the k^{th} allele
- p represents the probability of an allele being present among the population

Conclusion:

By representing genetic variants using this probability model involving multinomial distribution by representing genetic variants in a specific population using discrete random variables. This probability model model can help with getting a bigger picture of variant's implications on disease prevention.

References:

The Mathematical Theory of Probabilities: and Its Application to Frequency Curves and Statistical Methods, Arne Fisher (1915). Macmillan Co., New York. Academic.oup.com. (n.d.). https://academic.oup.com/bioinformatics/article/32/5/713/1744055.

Predicting Cybersecurity Breaches Using Poisson Distribution

Brendan Reilly

Introduction:

Cybersecurity breaches pose significant threats to organizations worldwide, with potential consequences ranging from financial loss to reputational damage. Predicting the occurrence of such breaches is essential for proactive risk management. The Poisson distribution offers a statistical framework for modeling rare events, making it applicable to cybersecurity breach prediction. In this project, we explore the application of the Poisson distribution in forecasting cybersecurity breaches.

Background Information:

Cybersecurity breaches encompass a wide range of unauthorized activities, including data theft, malware infections, and denial-of-service attacks. The Poisson distribution, commonly used to model the occurrence of rare events, is suitable for estimating the frequency of cybersecurity incidents within a given time interval. This distribution assumes that events occur independently and at a constant average rate.

Probability Model:

Let X denote the number of cybersecurity breaches occurring within a specified time period t. The Poisson probability mass function (PMF) is given by:

$$P(X = k) = \frac{e^{-\lambda} \cdot \lambda^k}{k!}; \quad k = 0, 1, 2, \ldots.$$

Where:

- k is the number of breaches.
- λ is the average rate of breaches per unit time.

Probability Calculation:

Suppose an organization experiences an average of 3 cybersecurity breaches per month. We want to calculate the probability of exactly 2 breaches occurring in the next month using the Poisson distribution:

$$P(X = 2) = \frac{e^{-3} \cdot 3^2}{2!} = 0.224$$

Conclusion:

The Poisson distribution serves as a valuable tool for predicting cybersecurity breaches based on historical data and average breach rates. By understanding the probability of breach occurrences, organizations can implement preemptive measures and allocate resources effectively to enhance their cybersecurity posture.

References:

1. Stallings, W., & Brown, L. (2017). "Computer Security: Principles and Practice" (4th ed.). Pearson.
2. Pfleeger, C. P., & Pfleeger, S. L. (2015). "Security in Computing" (5th ed.). Pearson.

Probability Model in Health

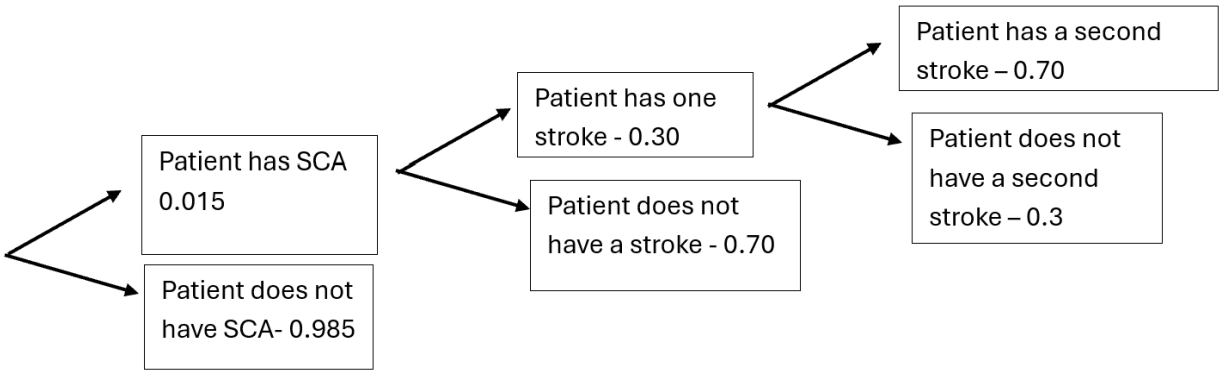
Emily Rivers

Introduction/Background:

Sickle cell anemia (SCA) is a disease controlled by a single gene. If the gene functions properly then you will not have any issues producing hemoglobin and have normal shaped red blood cells. If that gene is non-functional, any red blood cells produced will lack hemoglobin and take the shape of a sickle. SCA puts its patients at risk for different cardiovascular events such as strokes. When children develop SCA, their risk of a cardiovascular event such as a stroke is increased significantly.

Probability model:

About 1.5% of children born in the US have sickle cell anemia (SCA). Their risk of having a stroke is 30%, and if the patient has already had one stroke, then the chances of having another stroke are raised to about 70%. The probability of having sickle cell anemia and then one stroke and then another stroke can be modeled by the following tree diagram.



- Probability of having SCA: $0.015 = 1.5\%$
- Probability of having one stroke due to SCA: $0.015 \times 0.30 = 0.0045 = 0.45\%$
- Probability of having two strokes due to SCA: $0.015 \times 0.30 \times 0.70 = 0.00315 = 0.315\%$

References:

- "About Sickle Cell Disease." National Human Genome Research Institute, 26 May 2020, https://www.genome.gov/Genetic-Disorders/Sickle-Cell-Disease.
- CDC. "Incidence of Sickle Cell Trait in the US — CDC." Centers for Disease Control and Prevention, 4 May 2018, https://www.cdc.gov/ncbddd/sicklecell/features/keyfinding-trait.html.
- "Sickle Cell Disease and Stroke." Nationwide Children's Hospital, 2024, https://www.nationwidechildrens.org/family-resources-education/health-wellness-and- safety-resources/helping-hands/sickle-cell-disease-and-stroke.

Multinomial Distribution in Mechanical Engineering

Katrina Russell

Introduction/Background Information

Probability plays a fundamental role in various aspects of mechanical engineering, and allows engineers to tackle uncertainty and make informed decisions. As a mechanical engineer, one of the most important parts of the job is being able to detect when a product is defective. Mechanical engineers are often dealing with systems that have uncertain inputs and outputs. Using probability we are able to assess the risk associated with different scenarios, such as failure of components under various operating conditions. One example of this is multinomial distribution. As demonstrated in the problem below, multinomial distribution can be very useful in manufacturing and machine use. In manufacturing processes, products often need to meet certain specifications or quality standards. Multinomial distribution deals with experiments where there are more than two possible outcomes, and each outcome has a specific probability of occurring in each trial. These outcomes are mutually exclusive, meaning that only one of them can happen in each trial. This can be used to model the probability of products falling into different quality categories during production. By analyzing the probability of defects occuring, engineers can identify areas for improvement in the manufacturing process, as well as better identify the likelihood of a product being defective.

Probability Model: Multinomial Distribution:

If we have k possible outcomes for our experiment with probabilities $p_1, \ldots, p_k$, then the probability of getting exactly ni outcomes of type i in $n = n_1 + \cdots + n_k$ trials is:

$$P(n_1, n_2, \ldots, n_k) = \frac{n!}{n_1! \cdot \cdots \cdot n_k!} \cdot p_1^{n_1} \cdot p_2^{n_2} \ldots p_k^{n_k}$$

Probability Calculation

The output of a machine is graded excellent 70% of the time, good 10% of the time, and defective 10% of the time. What is the probability that a sample of size 15 has 10 excellent, 3 good, and 2 defective items?

The total number of trials is $n = 15$. there are $k = 3$ categories: excellent, good, and defective. We are interested in outcomes with $n_1 = 10, n_2 = 3$, and $n_3 = 2$. Using the formula we get:

$$P(n_1, n_2, \ldots, n_k) = \frac{15!}{10!3!2!} \cdot 0.7^{10} \cdot 0.2^3 \cdot 0.1^2$$

This gives us the exact probability of having the given number of each different quality of product.

Conclusion

In this problem, multinomial distribution is used to determine the probability that in a sample of 15 products, there will be 10 excellent, 3 good, and 2 defective items. This is just one example of how probability can be used to help engineers determine the quality of products and the effectiveness of of machines.

References

Elementary Probability for Applications, Rick Durrett (2009) (ISBN 978-0-521-86756-6).

Application of Probability in Data Science: Count Data Analysis

Ian Schneider

Introduction:

Counting is an important source of information and way of data collection. Data science is a multidisciplinary field, and counting techniques are used in every field. Once you get the data from the counting, you need to be able to analyze it. A common probabilistic model used in count analysis is the Poisson distribution, which is employed when the focus of count analysis is the number of events occurring in a fixed interval of time or space. In this project, we will develop a Poisson model to model the distribution of the number of patient arrivals at a hospital every hour.

Background:

Consider a busy suburban hospital that wants to minimize the amount of workers that need to be in the hospital at one time. One factor contributing to the minimization of workers is the amount of expected patients in any given hour. We can model the probability distribution of this statistic using a Poisson distribution. First, we will take counts of patients into the hospital per hour for every hour, every day of the week. Using this data, optimally from multiple different weeks throughout the year, we can determine if the model follows a Poisson distribution, and, if it does, determine an estimate for the mean number of incoming patients per hour (λ).

Probability Model:

Let X represent the number of patients arriving at the hospital in 1 hour. Then, the number of patients per hour could be modeled by the function,

$$P(X = x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}; \quad x = 0, 1, 2, \ldots.$$

Where λ is the observed average number of patients arriving at the hospital in 1 hour.

Example: Lets say we want to know the probability that 100 patients arrive in an hour where we have determined the average number of patients per hour to be 87. So, $\lambda = 87$ and $x = 100$, would find 0.01578707534. There would be around

$$P(X = 100) = \frac{e^{-87} \cdot 87^100}{100!} = 0.01578707534$$

1.58% chance of this happening according to the model.

Conclusion:

A data scientist would likely write a program capable of calculating for every λ hour (or any other desired time period), and then automatically update as time passes and more data is collected in order to keep changing and improving the prediction power. Using this estimation, a hospital can determine how many workers it needs for each hour of the day. For example, a 1:00 AM shift on a Tuesday is likely to need much less workers than a 1:00 AM shift on a Friday

would need because of the lesser amount of incoming patients. Ultimately, however, the hospital would also likely consider many other factors, such as patient discharges, specific department data (beds available, food available, etc.), etc.

Probability Model: Kalman Filters in Robotics

Zachary Serocki

Introduction & Background:

Kalman filters are pivotal tools in robotics. They are used to address sensor noise. By using the inherent probabilistic characteristics of sensor data, these filters play a crucial role in augmenting the reliability of sensor information. For instance, within a robotic system equipped with diverse localization sensors such as odometry and lidar, the Kalman filter adeptly amalgamates data from these sources to precisely estimate the robot's position. An illustrative scenario involves robots outfitted with a blend of localization sensor pairs comprising relative and absolute sensors. This selection is strategic: absolute sensors tend to exhibit lower confidence in their measurements yet are impervious to drift, whereas relative sensors offer more precise measurements that will drift as time goes on due to noise. Here, the relative sensors constitute the prediction stage, while the absolute sensors constitute the observational stage. Leveraging their independence, the optimal estimate is derived by combining their probability distributions as both variables follow a Gaussian distribution.

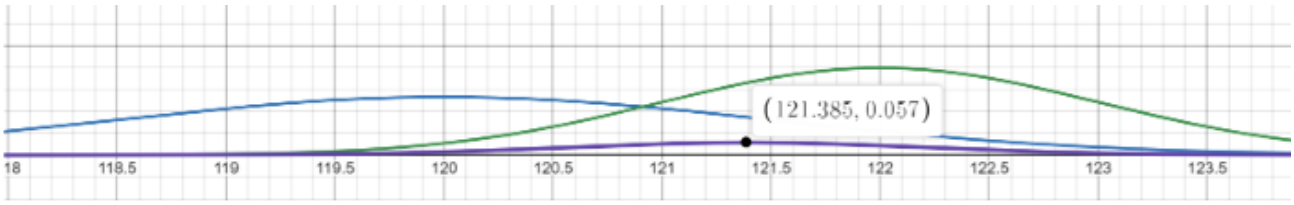
$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{1}{2\sigma^2}(x-\mu)^2}; -\infty < x < \infty, -\infty < \mu < \infty$$

where μ is the mean value and σ is the standard deviation.

Example:

Envision the task of estimating the position of a car. You can collect velocity data from the speedometer, indicating a speed of 60 mph (constant speed). Additionally, the car is equipped with a GPS unit capable of estimating position with a standard deviation of 1 mile, while the speedometer's standard deviation is 0.5 mph. After 2 hours, the GPS indicates the car is 122 miles from the starting point.

The car's position is initially approximated using the speedometer data: (60 mph) $\times$ (2hours) = 120 miles, with the standard deviation propagating to 1 mile. By integrating the distributions from both sources, the most probable position of the car, factoring in the variability of each variable, is determined to be approximately 121.385 miles from the starting point.



Conclusion:

In conclusion, probability distributions serve as invaluable tools in mitigating the impact of sensor noise, facilitating robust sensor fusion processes that yield high accuracy outcomes. Beyond this, alternative noise reduction techniques such as running average filters offer additional avenues for refining data, these work because they keep a large sample of the previous data and whenever new data comes in it has a minuscule effect on the output of the filter. Furthermore, Bayesian Probabilistic Inferences provide yet another dimension, leveraging past state information to estimate current states with precision. These multifaceted approaches collectively contribute to advancing the reliability and efficacy of sensor-based systems in various domains, including robotics and beyond.

Probability Model for Malicious Network Traffic Detection using Anomaly Algorithms Detection

Spencer Trautz

Introduction:

Network security is a critical concern in computer science, particularly with the increasing frequency and sophistication of cyber attacks. Anomaly detection algorithms play a crucial role in identifying unusual patterns or behaviors in network traffic that may indicate malicious activity.

Background Information:

Anomaly detection algorithms analyze network traffic data to identify deviations from normal behavior. These algorithms utilize probabilistic models to distinguish between normal and malicious network traffic. One widely used approach is based on Gaussian distributions, where normal network traffic follows a certain distribution, and deviations from this distribution may signal an anomaly.

Probability Model:

Let X represent a feature vector representing network traffic characteristics. We assume that X follows a multivariate Gaussian distribution, where μ represents the mean vector and Σ represents the covariance matrix:

$$f(x;\mu,\Sigma) = \frac{1}{(2\pi)^{d/2} \cdot |\Sigma|^{1/2}} \cdot e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}; \quad -\infty < x, \mu < \infty, \sigma > 0$$

Probability Calculation:

Suppose we have collected network traffic data and estimated the mean vector μ and the covariance matrix Σ from a training dataset. We want to calculate the probability of observing a particular network traffic instance X, given our learned model.

Conclusion:

In this project, we developed a probability model for detecting malicious network traffic using anomaly detection algorithms based on Gaussian distributions. By analyzing deviations from normal network traffic behavior, these algorithms can help identify potential security threats in computer networks.

Reference:

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern Classification (2nd ed.) Wiley-Interscience.

Probability in Robotics

Lehong Wang

Introduction:

A Markov chain is a stochastic process that describes a sequence of events in which the probability of each event depends only on the state of the system at the previous event. Markov chains are used to model a wide variety of phenomena, such as weather, stock prices, and traffic patterns. In Robotics, the Markov chain is used to model a markov process. For example, when we would like to estimate the current state of a robot, we could use the previous state and the action taken to predict the next state. This is an essential, and this theory leads to the famous Kalman filter, that is used widely.

Example:

Here, we will demonstrate the simplest form of this theory, where we have the PMF of current state, and the PMF of the result of the action, which, combined, will give us the PMF of the next state.

Problem setup:

A robot (dot) is moving along the integer number line, and it could attempt to move forward a step (+1), but this action (a) have 30% probability of failing. The PMF of the current state is: $P(X = x) = 0.8x = 0, 0.2x = 1$where x is the position that the robot is at.
Question is, what is the PMF of the next state of the robot.

$$x = 0 \rightarrow x = 0, a = 0 \rightarrow 0.8 * 0.3 = 0.24$$
$$x = 1 \rightarrow x = 0, a = 1 \text{ or } x = 1, a = 0 \rightarrow 0.2 * 0.3 + 0.8 * 0.7 = 0.62$$
$$x = 2 \rightarrow x = 1, a = 1 \rightarrow 0.2 * 0.7 = 0.14$$

So, PMF: $P(X = x) = 0.24x = 0, 0.62x = 1, 0.14x = 2$ is the next state of the robot.

Reference:

A New Approach to Linear Filtering and Prediction Problems. (1960). Transactions of the ASME–Journal of Basic Engineering, 82(Series D), 35–45.

Probability Model in Computer Science

Evelyn Yee

Introduction and Background:

In software development, estimating the time it may take in completing a programming project is vital in making sure it can be completed in a timely manner. By using previous data, the normal distribution can be used in order to predict how long a project may take to complete based on a programmer's average time to finish projects in the past. In this project, we will use the normal distribution to estimate the completion time of a programming project.

Probability Model and Calculation

Let X represent the time in hours it will take a programmer to finish one programming project.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{1}{2\sigma^2}(x-\mu)^2}; -\infty < x < \infty, -\infty < \mu < \infty$$

where μ is the mean value and σ is the standard deviation.

Example:

A software development company would like to assign a programming project to an employee who will take no more than a certain amount of time to complete it. A certain employee's completion time is a normal distribution with a mean value of 32 hours and a standard deviation of 2.4 hours. What is the probability that this employee will, at maximum, take 30 hours to complete this project?

Using the normal distribution, we can calculate this probability using the probability model.

$$P(X \leq 30) = \Phi(30 - 32/2.4) = \Phi(-0.83) = 0.20327.$$

Conclusion:

The probability that this employee will take, at maximum, 30 hours to complete this particular project is 20.3%. Using this probability model, the company can decide which employee may be the best fit for any particular project and improve their project completion efficiency.

Probability Model in Computer Science Software Engineering and Video Game Design.

Joseph Caproni

Introduction/Background:

Software applications, especially video games use probability to simulate the chance of certain events occurring. Games often use Random Number Generators or RNG to determine when certain events occur, like what might be inside a random loot box that the user purchases for example. The way that these loot boxes work is that when the player opens them, the computer rolls electronic dice, which return a number. The numbers on the dice correspond with various items that the loot box then rewards the player with. This has revolutionized the mobile gaming world, as many companies have designed models around chance, assigning various items in games different rarities (probabilities of opening in loot boxes) to motivate customers to play and purchase these boxes. To increase the rarity of certain items, designs may use multiple electronic dice and tie the outcome of the item being dispensed to the combinations of each.

Example/Probability Model:

Say there is a video game in which a player gets a treasure chest and has the chance to get either a wooden stick, silver, gold, or a diamond from it. Assume that there are two electronic dice.

$$P(W) = 70\%, P(S) = 15\%, P(G) = 10\%, P(D) = 5\%$$

When the player opens the chest, they get two items. Whatever each die decides is what each item the player gets. Since there are two dice the player will get two random combinations of items based upon the probabilities listed above.

$$P(\text{Two Item Output}) = P(\text{die 1}) \times P(\text{die 2})$$

This is done to ensure that a variety of items are dispensed, with the probability being tied to multiple independent events, instead of a single one.

Probability model in Computer Science

Martin Kalo

Introduction:

Sorting algorithms are a very well-known thing in the world of computer science. We use sorting algorithms all the time in order to handle data in an efficient way. Depending on the different ways algorithms operate, they may have different completion times and probability distributions for achieving those times.

Background:

Consider the sorting algorithm Quick Sort. This algorithm sorts the data by choosing a random value from the data and turning it into a "pivot". Then, it puts all of the values in the array which are greater than the pivot to the right, and the values which are smaller to the left. This process is repeated until the array is sorted. The time to complete a successful sort can be modeled with a normal distribution, because some values result in a more efficient iteration of the program and the value chosen as the pivot is random.

Probability Model:

The PMF of a normal distribution can be given as the following:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{1}{2\sigma^2}(x-\mu)^2}; -\infty < x < \infty, -\infty < \mu < \infty$$

where X is the variable which represents the time taken to complete the sorting, μ is the mean time taken to complete the operation, and σ is the standard deviation of time taken to complete the operation.

Probabilistic Latent Factor Models in Recommendation Systems

Abbas Jivan

Introduction/Background:

In many online streaming services such as YouTube, Netflix, and Spotify, recommendation systems are the core selling point that keep users on the service for hours at a time, providing them with content that is likely to appeal to the user. Thus, these companies are always looking to improve their recommendations processes and in recent years, with breakthroughs in machine learning, recommendation systems have only grown become more robust and intuitive. While traditional recommendation systems use deterministic models such as collaborative and content-based filtering, these do not account for the uncertainty of user preferences. Instead, a better method is the use of probabilistic latent factor models such as probabilistic matrix factorization to account for uncertainties and thus create models that provide more accurate recommendations.

Example:

To demonstrate, consider an example where a user on Netflix continuously watches romance movies and generally rates them with a high score. However, another analysis shows that within the romance genre, the user tends to watch movies where the main love interests end up together and does not highly rate movies where the main love interests are separated in the end. To better determine movies that the user may enjoy, we can find the minimum value of the square of the distance between the original scoring matrix and the scoring matrix of the potential factors that could impact recommendations. This will help us determine the best scoring matrix of factors and thus give us more accurate recommendations for the user. This is usually accomplished through a machine learning model that utilizes a variety of matrix factorization technologies to best determine movies for the user.

Conclusion:

In conclusion, probabilistic latent factor models such as probabilistic matrix factorization are better utilizations recommendation systems as they provide more accurate recommendation due to being able to consider more nuanced factors and uncertainties compared to deterministic models.

References:

Zhang, Z., Wu, Q., Zhang, Y., & Liu, L. (2023, April 21). Movie recommendation model based on probabilistic matrix decomposition using hybrid AdaBoost integration. PeerJ. Computer Science. https://doi.org/10.7717/peerj-cs.1338

"Probability is not just a math concept; it's a way of thinking about the world."

"In every situation, you have a choice—what you do with it is your probability of success."

"Probability is the very guide of life." – Cicero

"The best way to predict the future is to create it." – Peter Drucker

****