Probability-based Models

Monday, October 21, 2024 1:30 PM

Probability Distributions Introduction

Probability distributions can form the backbone of simple models and sometimes a simple model is all you need. If you are able to match the data to a known probability distribution, you can gather a lot of information based on that distribution's characteristics.

Bernoulli, Binomial, and Geometric Distributions

The Bernoulli Distribution is a discrete probability distribution and has only two possible values, {0,1}, with 1 often called a success with probability p and 0 often called a failure with probability 1-p.

Let X be a Bernoulli random variable. Then:

$X \sim Bernoulli(p)$

$P(X = x) = \begin{cases} X = 1 : p \\ X = 0 : 1 - p \end{cases}$

- $E[X] = p$
- $Var[X] = P(1 - p)$

The Binomial Distribution is a discrete probability distribution and it counts the number of successes in a sequence of *n* independent Bernoulli trials. The probability p of each Bernoulli trial remains constant for all trials, each trial is independent, and n is a finite number. With large enough number of trials, this distribution converges to the Normal Distribution.

Let X be a Binomial random variable. Then the probability for x successes within n trials is:

$X \sim Binomial(p)$

$P(X = x) = \binom{n}{x} P^x (1 - p)^{n-x}$

- $E[X] = nP$
- $Var[X] = nP(1 - p)$

The Geometric Distribution is a discrete probability distribution for the number of Bernoulli trials it takes to get the first success. This distribution has the special property of being memoryless. This property means that the distribution does not "remember" previous trials; i.e., the outcome of previous trials does not affect subsequent trials and therefore a new trial can be treated like it is the first.

Let X be a Geometric random variable. Then the probability of getting the first success on trial number x is:

$X \sim Geometric(p)$

$P(X = x) = (1 - p)^{X-1} p$

- $E[X] = \frac{1}{p}$
- $Var[X] = {1 - p}/{p^2}$

Poisson, Exponential, and Weibull Distributions

The Poisson Distribution is a discrete probability distribution used to model the occurrences of an event during a fixed interval, where the occurrences in disjoint intervals are independent. Examples include number of emails received in a day, or accidents in a week, or holes in a yard-long tube. The interval is usually some measure of time, but it can be other quantities.

Let X be a Poisson random variable. Then the probability that the event occurs x times is:

$X \sim Poisson(\lambda)$

$P(X = x) = {e^{-\lambda} \lambda^x}/{x!}$

- $E[X] = \lambda$
- $Var[X] = \lambda$

The Exponential Distribution is a continuous probability distribution used to model waiting times between independent events that occur with a constant average rate, like the lifetime of a machine or wait times between bus arrivals. This distribution has the special property of being memoryless. If the distribution of the events is Poisson distributed, then the time between events is Exponentially distributed.

Let X be an Exponential random variable. Then the time until the event occurs is:

$X \sim Exponential(\lambda)$

$f(x) = \lambda e^{-\lambda x}$

- $E[X] = \frac{1}{\lambda}$
- $Var[X] = \frac{1}{\lambda^2}$

The Weibull Distribution is a continuous probability distribution used to model the time to failure or time between failures, with the shape parameter α indicating if the failure rate changes over time. Note that a Weibull random variable with k=1 reduces to an Exponential random variable where the parameter is flipped to $\frac{1}{\lambda}$. Where the Geometric distribution models the number of trials between failures (flipping light switch on and off until the bulb fails), Weibull models the time between failures (leaving the light on until the bulb fails).

- k = 1: failure rate is constant
- k<1: failure rate decreases over time; worst things fail first
- k>1: failure rate increases over time; things that wear out

Let X be a Weibull random variable. Then the time to failure is:

$X \sim Weibull(k, \lambda)$

$f(x) = \frac{k}{\lambda} (\frac{x}{\lambda})^{k-1} e^{-(x/\lambda)^k}$

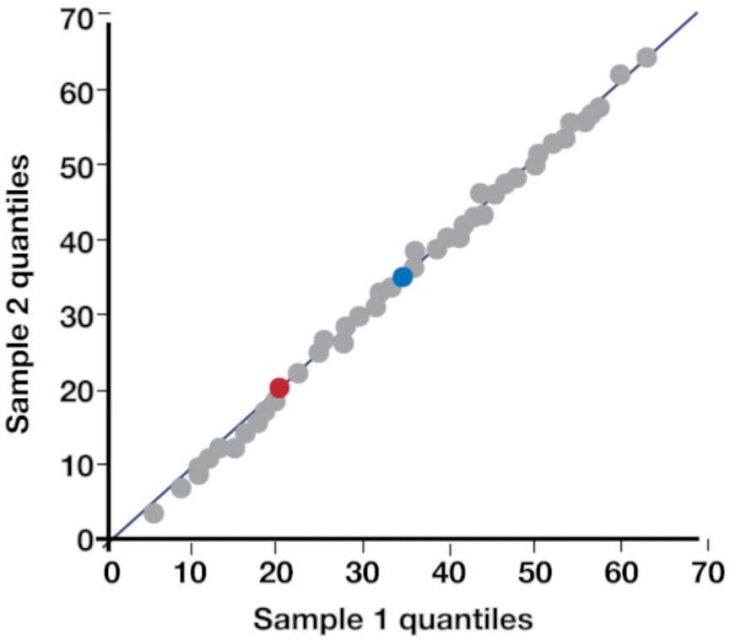
There is software that can tell you what distribution your data best fits, but only use it as guidance.

QQ Plots

QQ Plots test whether a single data set is a good fit to a probability distribution:

- Horizontal axis: data
- Vertical axis: theoretical values of percentiles in a perfect distribution

The idea behind a QQ plot is that, whatever variation in the data there might be and even if there is a disparate number of data points, two similar distributions should have about the same value at each quantile. Thus the quantiles of one data set are plotted against the quantiles of another data set, and the more similar their distributions, the more the QQ plot looks like a 45 degree line. Seeing this information visually can inform us better than a simple output from a statistical test.



Queueing

Suppose we run a telemarketing business with an auto-dialer that automatically calls phone numbers. If the call is answered, it is put into a queue to then be picked up by an employee. So: how many employees should we have, based on how many people answer the auto-dialer or based on the duration of the call once the employee is on the phone?

In the simplest case, suppose the distribution of the calls is $Poisson(\lambda)$ and the call length is $Exponential(\mu)$. Then:

- Arrival Rate (calls) = λ
- Service Rate (calls) = μ > λ
- Transition Equations (≥ 1 calls in the queue)
  - $P(next\ event\ is\ an\ arrival) = \frac{\lambda}{\lambda+\mu}$
  - $P(next\ event\ is\ a\ finished\ call) = \frac{\mu}{\lambda+\mu}$
- We can then calculate:
  - Expected fraction of time employee is busy: $\frac{\lambda}{\mu}$
  - Expected waiting time before talking to an employee: $\frac{\lambda}{\mu(\mu-\lambda)}$
  - Expected number of calls waiting in queue: $\frac{\lambda^2}{\mu(\mu-\lambda)}$

Thus we can make a queueing model with certain potential parameters:

- General arrival distribution [A]
- General service distribution [S]
- Number of employees [c]
- Size of the queue [K]
- Population size - potential people called by auto-dialer [N]
- Queueing discipline - how the queueing is done [D]

A standard way of notating this is known as Kendall Notation. The model can be extended with certain possibilities, like a caller hanging up in the queue, balking, etc. Additional complexities to the model means that the math behind it gets very difficult.

Due to the Exponential Distribution's memoryless property, we know that the distribution of the remaining call time is the same as the initial distribution of call time. If our data fits the Exponential distribution, then we know it is memoryless. If the data is not memoryless, it cannot fit the exponential distribution.

As an example, say we want to model the distribution of tire failures at 10,000 miles. Tires are more likely to fail the more worn out they are, thus the distribution of the remaining tire lifetime is not the same as the initial tire lifetime and therefore this model cannot be Exponentially distributed. Perhaps it can be modeled with the Weibull distribution with k>1.

Simulation Basics

Simulation is the process of building a model to observe its behavior. Simulations can be continuous (with changes happening continuously) or discrete (changes occur only at discrete points in time). There are several different types of simulations:

- Deterministic simulations: Given the same inputs, the outputs will always stay the same, so there is no random variation.
- Stochastic simulations: Output may vary each time the model is run because the system includes randomness that we've modeled.

Discrete Stochastic Simulations are very valuable when analyzing systems that have high variability and when using average values isn't good enough.

Simulation software generally allows you to build several things into the simulation:

- Entities: things that move through the simulation (people, bags, etc.)
- Modules: parts of the process (queues, storage, etc.)
- Actions
- Resources
- Decision Points
- Statistical tracking

Simulations ought to be run many times, since one run / replication is a single data point which may be unrepresentative of the outcomes. With many simulation replications, we can get a distribution of the outcomes. The output distribution can be used to evaluate the reasonableness of the simulation output:

- If the real and simulated averages don't match, that indicates a problem with the simulation
- If the averages match, but the variances don't match, that indicates a problem with the simulation

It's difficult to validate a simulation of something that doesn't exist because there's no way to compare the real and simulated statistics. It is important to properly validate the simulations because otherwise you may make decisions based off simulations that lead you entirely down the wrong and potentially costly path.

Prescriptive Simulation

Simulation is great for asking "what if" questions or to explore what the best potential solution may be for a situation. By running the same simulation with slightly different variations, we are able to compare the simulations to see which performed better. Keep in mind that one set of 'random' observations may be better than another purely by chance and this can be avoided by using the same random numbers for both simulations.

Note that simulation can only be as good as the quality of the input, so missing or incorrect information may lead to incorrect answers.

Markov Chain Model

Markov Chain Models are based on states of a system.

Say we want to make a Markov Chain model of the state of the weather - cloudy, rainy, sunny. For each state *i* in the model:

- $p_{ij} = transition\ probability\ from\ state\ i\ to\ state\ j$
- $P = \{p_{ij}\}$ is the transition matrix

To

From / Transition Probabilities	Sunny	Cloudy	Rainy
Sunny	.75	.15	.10
Cloudy	.20	.40	.40
Rainy	.40	.30	.30

With the transition probabilities, we can answer questions like "what is the long-run probability of rainy days"?

- Given today's probabilities of sunny, cloudy, rainy: $\pi = (.5, .25, .25)$
- We can estimate the probabilities of tomorrow: $\pi P = (.525, .25, .225)$
- And we can estimate the probabilities of the day after tomorrow: $(\pi P)P = (.53375, .24625, .22)$

The long-run probability of rainy days is πP^{∞}, which is difficult to calculate. Instead, we can use steady state, where the states have gotten so mixed that the initial conditions no longer matter and thus the probability of being in state i is the same every day:

- Apply P to get the initial vector back: $\pi P = \pi^*$
- We can solve the system of equations: $\pi P = \pi^*$ and $\sum_i^{\square} \pi_i^* = 1$
- This gives the steady state probability vector π^*
  - π^* may not always exist since we can't have cyclic behavior between states and every state much be reachable from all other states.

A key assumption when dealing with Markov Chains is that they are Memoryless; state transitions only depend on the most recent state. Most systems don't exhibit the Memoryless property. For systems that do exhibit this behavior, these models are powerful and effective.

Some example of Markov Chain Models are Google's Page Rank system, models for urban sprawl, population dynamics, and disease propagation, and systems that aren't memoryless in the short term but can still get meaningful models of long-run system dynamics (since the Memoryless property becomes less important in the long run).