Undergraduate Journal of Mathematical Modeling: One + Two

Volume 11 | 2020 Fall 2020 Article 1

2020

Probabilistic Machine Learning Using Bayesian Inference

Mayank Pandey
University of South Florida, mayankpandey@usf.edu

Advisors:
Arcadii Grinshpan, Mathematics and Statistics
James Anderson, Computer Science and Engineering

Problem Suggested By: Mayank Pandey

Recommended Citation
Pandey, Mayank (2020) "Probabilistic Machine Learning Using Bayesian Inference," *Undergraduate Journal of Mathematical Modeling: One + Two*: Vol. 11: Iss. 1, Article 1.
DOI: https://doi.org/10.5038/2326-3652.11.1.4920
Available at: https://scholarcommons.usf.edu/ujmm/vol11/iss1/1

Probabilistic Machine Learning Using Bayesian Inference

Abstract

Machine Learning is a branch of AI (Artificial Intelligence) which expands on the idea of a computational system extending its knowledge about set methodical behaviors from the data that is fed to it to essentially develop analytical skills that can help in identifying patterns and making decisions with little to no participation of a real human being. Computer algorithms help in gaining experience to improve the facility over time for use by both consumers and corporations. In today's technologically advanced world, Machine Learning has given us self-driving cars, speech recognition software, and AI agents like Siri and Google assistant. This project evaluates how the Beta function came to be and how Stirling's formula is implemented in calculating the magnitude of this function for large input values. The Beta function can then be used to produce a Beta distribution of probabilities to find whether people will actually watch a video they come across on their recommendations feed or search feed and then using Bayesian inference update the prior set predictions.

Keywords

Bayes' theorem, Bayesian inference, machine learning, beta function, beta distribution, Stirling's formula

PROBLEM STATEMENT

Show how Bayesian inference is used in Machine Learning which enables corporations like YouTube to predict how many people will actually watch a video they come across on YouTube's recommendations feed or search feed and how this data develops an intelligent sorting algorithm.

MOTIVATION

Big corporations like YouTube make use of programs and algorithms that can predict the outcome of appreciation of a particular video or post on their sites. The results are probabilities of univariate cases. The Beta function is used for generating a Beta distribution. Once essential data is generated for sampling, an algorithm implements the Bayesian inference to successfully predict the percentage of people that will possibly watch the video. Furthermore, with the help of a binomial distribution, YouTube can even figure out the number of views they will fetch for every possible user that comes across a particular video which can help their sorting algorithm based on Machine Learning to target specific audiences or demonetize videos based on a violation of rules which adds to the experience for the selection algorithm that can potentially restrict future content of such type. Therefore, it is incredibly important and useful for companies to develop Machine Learning algorithms that can improve over time and reduce labor-intensive work for the company and require minimal human intervention. Machine Learning and Artificial Intelligence in itself really interest me so I decided to research more about how it's implemented at a large scale.

MATHEMATICAL DESCRIPTION AND SOLUTION APPROACH

In order to target specific audiences for their future videos and content for maximum efficiency and profit, video streaming companies like YouTube, Netflix, and Hulu gather large amounts of user feedback on some "prior" content. Essentially, they are using the Bayesian inference which comes from Bayes' theorem to model a probability of the event where a specific criterion bounded audience likes their content based on similar content views from the past. To model this probability, they use a probability distribution like Beta or Dirichlet. The Beta distribution is univariate whereas Dirichlet is for multivariate. Though any distribution of probabilities over (0,1) will work. Simply take any function that does not blow up between 0 and 1 everywhere and remains stable, then add it from 0 to 1, and simply split the function with the result. The reason Beta Distribution is special is that it is the prior conjugate in Bayesian inference for the Bernoulli, binomial, negative binomial, and geometric distributions. It is very easy to calculate a posterior using a prior conjugate since you can skip the expensive computational calculation involved in Bayesian inference.

Since the Beta function is related to the Gamma function and it is itself a factorial function generalization. Many complex integrals can be reduced to expressions involving the Beta function.

The notation for the Gamma function is:

$$\Gamma(z) = \int_0^\infty x^{z-1}e^{-x}dx \tag{1}$$

https://scholarcommons.usf.edu/ujmm/vol11/iss1/1
DOI: https://doi.org/10.5038/2326-3652.11.1.4920

If the real part of the complex number z in (1) is positive, then the integral converges absolutely, see, e.g., Davis (1959, pp. 849-869). Equation (1) is known as the **Euler integral of the second kind.** Using integration by parts, one sees that:

$$\Gamma(z+1) = \int_0^{\infty} x^z e^{-x} dx$$

$$= [-x^z e^{-x}]_0^{\infty} + \int_0^{\infty} z x^{z-1} e^{-x} dx$$

$$= \lim_{x \to \infty} (-x^z e^{-x}) - (0e^{-0}) + z \int_0^{\infty} x^{z-1} e^{-x} dx \qquad (2)$$

It is easy to recognize that in Equation (2): $-x^z e^{-x} \to 0$ as $x \to \infty$,

$$\Gamma(z+1) = z \int_0^{\infty} x^{z-1} e^{-x} dx \qquad (3)$$

$$= z\Gamma(z)$$

Equation (3) is the formal definition of the Gamma Function.

For the purpose of this project, we need the Beta Function as a function of the Gamma Function. To derive the integral representation of the Beta function, write the product $\Gamma(x)\Gamma(y)$ as:

$$\Gamma(x)\Gamma(y) = \int_{u=0}^{\infty} e^{-u} u^{x-1} du \cdot \int_{v=0}^{\infty} e^{-v} v^{y-1} dv$$

$$= \int_{v=0}^{\infty} \int_{u=0}^{\infty} e^{-u-v} u^{x-1} v^{y-1} du\, dv \qquad (4)$$

Equation (4) leads to a simple derivation of the relationship between the Beta and Gamma functions, see, e.g., Artin (2015).

Changing variables by u = f(z,t) = zt and v = g(z,t) = z(1-t) shows that this is:

$$\begin{aligned}\Gamma(x)\Gamma(y) &= \int_{z=0}^{\infty}\int_{t=0}^{1} e^{-z}(zt)^{x-1}(z(1-t))^{y-1}|J(z,t)|\,dt\,dz \\ &= \int_{z=0}^{\infty}\int_{t=0}^{1} e^{-z}(zt)^{x-1}(z(1-t))^{y-1} z\,dt\,dz \\ &= \int_{z=0}^{\infty} e^{-z}z^{x+y-1}\,dz \cdot \int_{t=0}^{1} t^{x-1}(1-t)^{y-1}\,dt \\ &= \Gamma(x+y)B(x,y)\end{aligned} \tag{5}$$

Representing Equation (5) in the standard form we get:

$$B(x,y) = \int_{t=0}^{1} t^{x-1}(1-t)^{y-1}\,dt = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)} \tag{6}$$

Equation (6) is the formal definition of the Beta function and the relationship between Gamma and Beta functions (see, e.g., Artin, 2015). The integral in (6) is known as **Euler's integral of the first kind.** Now, for generating probability distribution of probabilities we use the Probability density function of Beta Distribution, which is as follows:

$$\frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)} \tag{7}$$

Equation (7) is the Probability Density Function of Beta Distribution and it is consistent with several authors like (George, 1960; Gerald, 1994; Keeping, 2010; Norman, 1995).

https://scholarcommons.usf.edu/ujmm/vol11/iss1/1
DOI: https://doi.org/10.5038/2326-3652.11.1.4920

Since the Beta function, like the Gamma function, is based on $n!$ therefore, it is helpful then for cases with large values of n to approximate the value of this function.

In fact, programs written in the Python language or JAVA language implement Stirling's approximation to give an approximation for calculating factorials. It is also useful for approximating the log of a factorial.

There is a quick way to derive Stirling's formula. We can approximate the sum:

$$\ln (n!) = \sum_{j=1}^{n} \ln j \tag{8}$$

Equation (8) is consistent with several sources like (Wikipedia-Stirling's approximation, n.d.; Conrad, n.d.)

Now, using an integral:

$$\sum_{j=1}^{n} \ln j \approx \int_{1}^{n} \ln x \, dx = n \cdot \ln n - n + 1 \tag{9}$$

So instead of approximating $n!$ we will consider its natural logarithm as this is a slowly increasing or varying function:

$$\ln n! = \ln 1 + \ln 2 + \cdots + \ln n \tag{10}$$

$$\frac{1}{2}(\ln 1 + \ln n) = \frac{1}{2}\ln n \tag{11}$$

The right-hand side of Equation (10) minus Equation (11) is the approximation by the trapezoid rule of the integral (Conrad, n.d.):

$$\ln (n!) - \frac{1}{2}\ln n \approx \int_1^n \ln x \, dx = n\cdot\ln n - n + 1 \tag{12}$$

and the error in this approximation as suggested in Equation (12) is given by the Euler-Maclaurin formula (Conrad, n.d.):

$$\ln (n!) - \frac{1}{2}\ln n = \frac{1}{2}\ln 1 + \ln 2 + \ln 3 + \cdots + \ln(n-1) + \frac{1}{2}\ln n$$

$$= n \ln n - n + 1 + \sum_{k=2}^{m} \frac{(-1)^k B_k}{k(k-1)}\left(\frac{1}{n^{k-1}} - 1\right) + R_{m,n} \tag{13}$$

where B_k is a Bernoulli number and $R_{m,n}$ is the remainder in the Euler-Maclaurin formula. Now, we take limits of Equation (13) to find that:

$$\lim_{n\to\infty} [\ln (n!) - n\ln n + n - \frac{1}{2}\ln n] = 1 - \sum_{k=2}^{m} \frac{(-1)^k B_k}{k(k-1)} + \lim_{n\to\infty} R_{m,n} \tag{14}$$

Denote this Equation (14) as y. Because the remainder $R_{m,n}$ in the Euler-Maclaurin formula satisfies (Wikipedia-Stirling's approximation, n.d.):

$$R_{m,n} = \lim_{n\to\infty} R_{m,n} + O\left(\frac{1}{n^m}\right) \tag{15}$$

where the Big-O notation is used, combining Equation (14) and Equation (15) from above yields the approximation formula in its logarithmic form:

https://scholarcommons.usf.edu/ujmm/vol11/iss1/1
DOI: https://doi.org/10.5038/2326-3652.11.1.4920

$$\ln(n!) = n\ln\left(\frac{n}{e}\right) + \frac{1}{2}\ln n + y + \sum_{k=2}^{m} \frac{(-1)^k B_k}{k(k-1)n^{k-1}} + O\left(\frac{1}{n^m}\right) \quad (16)$$

Taking the exponential of both sides and choosing any positive integer m, one obtains a formula involving an unknown quantity e^y. For $m = 1$, the formula is:

$$n! = e^y\sqrt{n}\left(\frac{n}{e}\right)^n\left(1 + O\left(\frac{1}{n}\right)\right) \quad (17)$$

The quantity e^y can be found by taking the limit on both sides as n tends to infinity and using Wallis' product, which shows that $e^y = \sqrt{2\pi}$.

Therefore, one obtains Stirling's formula from Equation (17) (Wikipedia-Stirling's approximation, n.d.):

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1 + O\left(\frac{1}{n}\right)\right) \quad (18)$$

For large values of n Stirling's formula proves to be extremely useful since the usual notation of factorial is defined as:

$$n! = n\cdot(n\text{-}1)\cdot(n\text{-}2)\cdot(n\text{-}3)\cdots 1 \quad (19)$$

If applied to cases or situations where n is extremely large it would be a tedious and expensive approach since a lot of memory capacity is going to be used up in storing each consecutive result, therefore, the approach in Equation (19) is inefficient for large values of n.

The Gamma function is based on an integral so it is a better approach but since Stirling's formula is an asymptotic approximation, it is the most efficient method because it requires the least computational complexity for execution and from Big-O-notation, it has a decent run-time with sufficient memory use. This way Stirling's formula in Equation (18) can be easily used in Equation (7) to find the Beta Distribution. Beta Distribution gives a probability, so its domain is bounded between 0 and 1.

I believe YouTube exploits the Beta Distribution for generating a probability distribution of probabilities by using feedback from previous content as a sample. They actually make use of the conjugate prior method which can be inferred from the model for Bayes' Theorem in Equation (20) below:

$$P(\Theta|\text{data}) = \frac{P(\text{data}|\Theta) \times P(\Theta)}{P(\text{data})} \tag{20}$$

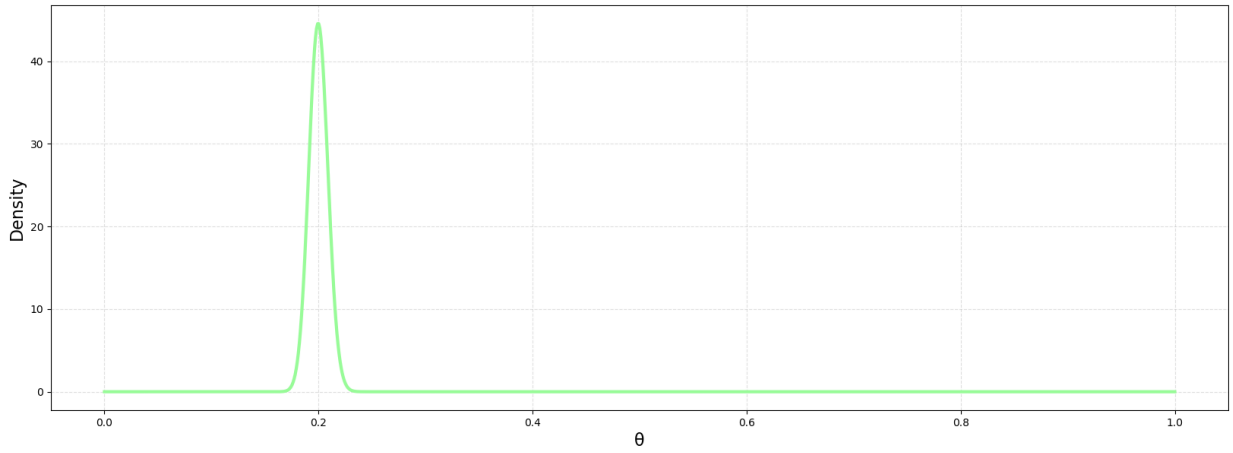
Conjugate prior essentially means for some likelihood functions, if you choose a certain prior $P(\Theta)$, the posterior $P(\Theta|\text{data})$ ends up being in the same distribution as the prior. Such a prior $P(\Theta)$ then is called a Conjugate Prior. Θ is the probability of success (most likes or most number of views) and our goal is to pick the Θ that maximizes the posterior probability which is calculated by numerical optimization, such as the Gradient Descent or Newton method. The $P(\text{data}|\Theta)$ denotes feedback from previous content from the source as a sample distribution.

Since I do not have access to such large scale sample data, for my analysis I am going to assume that if an average music video has about 4,000 interactions on YouTube (people came across this

https://scholarcommons.usf.edu/ujmm/vol11/iss1/1
DOI: https://doi.org/10.5038/2326-3652.11.1.4920

video) and if we know that 800 people watched the video then let α, the number of people that watched the video be 800 and let β, the number of people that didn't watch the video, be 4000-800 = 3,200 in Equation (7). These parameters — how big or small α & β are — will determine the shape of the distribution.

Using some Python code and using the algorithm for Beta distribution, we have now acquired the graph of this probability density for the prior distribution:



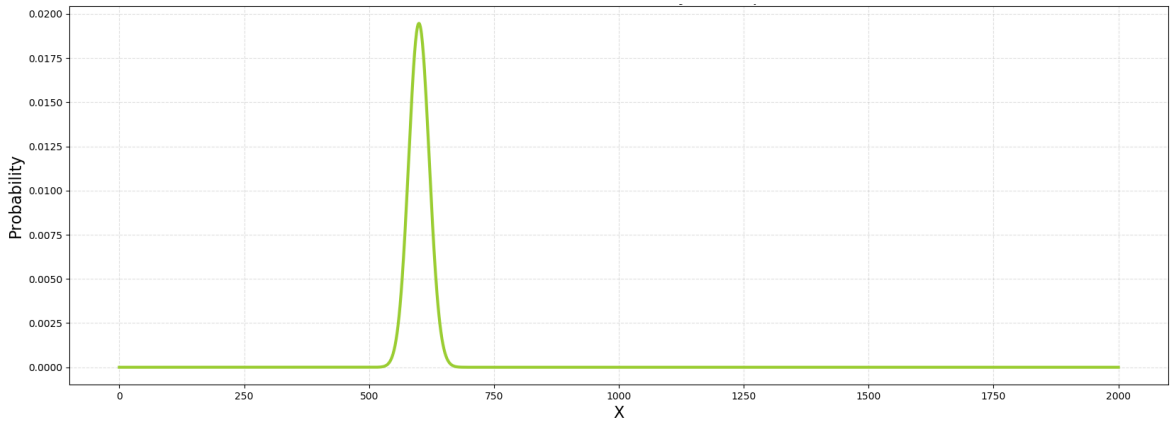
Aerin Kim. (2020). [Graph showing the probability density of about 20% for the selected values]. *Towards Data Science*. Retrieved from https://towardsdatascience.com/bayesian-inference-intuition-and-example-148fd8fb95d6

The Probability Density rises to 20% (800 views/4000 interactions) as expected. Four thousand points tends to be a solid prior in this case, judging from the plot. This is the $P(\Theta)$ part of Bayes' Theorem.

Now for the Likelihood $P(\text{data}|\Theta)$ we can assume that we have the same values for the data variable in the form of a binary array, say:

data is a binary array [0,1,0,1,...,0,0,0,1].

Since we know the number of views (n) and the number of interactions/visitors (p) and the prior probability $P(\Theta)$, we can use the binomial distribution for all possible values of **data** and a specific value of $\Theta=0.3$ (30%) to predict the number of views.



Aerin Kim. (2020). [Graph showing the probability of seeing X successes in n trials for Θ=30%]. *Towards Data Science*. Retrieved from https://towardsdatascience.com/bayesian-inference-intuition-and-example-148fd8fb95d6

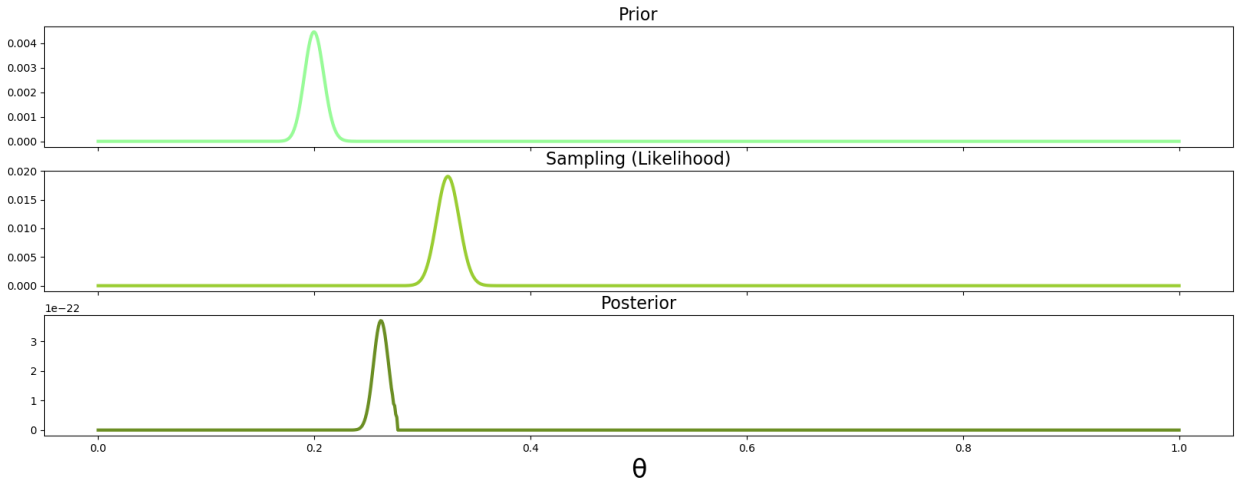
This is the probability of seeing successes for the data points of **data** in **n** trials, assuming a specific parameter $\Theta=0.3$ (30%) which is close to 0.02 (from the graph) so it is a very small probability.

Finally, the posterior function $P(\Theta|\text{data})$ is the crux of this approach. YouTube implements Bayesian inference using Bayes' Theorem. Again, the Bayesian inference is used to find Θ that best fits the **data**. The principle is simple, $P(\Theta)$ was the original definition of parameters. Now after acquiring some data we are upgrading a simple $P(\Theta)$ to something more informative —

DOI: https://doi.org/10.5038/2326-3652.11.1.4920

$P(\Theta|data)$ and it becomes smarter as smarter with more supply of **data** or experience. $P(\Theta|data)$ is still the probability of Θ and so is $P(\Theta)$. However, $P(\Theta|data)$ is a smarter version of $P(\Theta)$. Kim (2020) explains in her article "Bayesian Inference -- Intuition and Example" that:

> Although there are thousands of data points, we can transform them into a single scalar — the probability P(data|θ) — by plugging data into the model you want (in this case, the binomial distribution), then measure and multiply P(θ) & P(data|θ) for a particular model. If you do this for every possible resort you can select between different θ's the highest P(θ)*P(data|θ). We can essentially multiply P(θ) with P(data|θ) by conditioning on θ. (paras. 35-37)



Aerin Kim. (2020). [Graph showing the probability for prior, Likelihood, and Posterior]. *Towards Data Science*. Retrieved from https://towardsdatascience.com/bayesian-inference-intuition-and-example-148fd8fb95d6

The probability in the posterior graph can be observed to have shifted to the prior one. For the prior, the views' probability was 20%. The views' probability from the data points was taken to be 30%. Now, posterior has its value around 0.25%.

Since we added more information/data, the range of possible parameters is now smaller/narrower. Therefore, the width of the graph has reduced. The more data you collect, the posterior graph will look more like the Likelihood/sampling distribution graph and less like the prior. In other words, the original pre-distribution becomes less relevant when you get more info. In general, however, after we determine the posterior, the posterior is the fresh prior before the next data batch is entered. We can update our prior and posterior iteratively in this way. That is how YouTube's algorithm for recommending videos to target audiences learns from experience for the profit of the company.

DISCUSSION

Through this project, I wanted to demonstrate the use of special functions like Beta function, Gamma function, Beta Distribution, Stirling's formula, and Bayes' Theorem in the field of Machine Learning. I chose to evaluate how the YouTube algorithm decides what content is suitable (or profitable from the company's perspective) for what audience. Researching about Machine Learning introduced me to the concept of Bayesian inference. Bayesian inference is a method of statistical inference in which Bayes' theorem is used to update the probability for a hypothesis as more evidence or information becomes available. For the purpose of this project, I used my own small set of sample data for plotting the probability using the Beta Distribution, However, YouTube - being one of the most successful companies out there - has access to copious amounts of data that helps their Machine Learning algorithm to expand its vision from the experience. Calculating the probability using Bayes' Theorem led me to a probability of about 20% for the number of views for the prior. The number of views' probability was taken to be 30% for the **data**. Finally, the posterior had a peak around 0.25%.

https://scholarcommons.usf.edu/ujmm/vol11/iss1/1
DOI: https://doi.org/10.5038/2326-3652.11.1.4920

This result demonstrated the use of Bayesian inference as the posterior function learned from the predicted value for the probability and the expected value to shift towards the prior. Furthermore, when this method is repeated for new data, this posterior will become the new prior distribution and then the probability keeps getting updated at the end of each cycle. The endpoint of one cycle is the starting prior for the next. Therefore, more data means better results.

This small example goes on to show that Bayesian inference is an extremely powerful tool for the field of Machine Learning and Artificial Intelligence in general and can result in the advancement of technology effectively. The results of this project align with my initial beliefs about how YouTube uses user data to effectively manipulate content their audience receives upfront, However, it was only after much effort and research did I realize that this approach is used by several companies in today's world. It does make sense to use Bayesian inference in such a manner because while it benefits the companies and corporations with millions of dollars of profit, it is beneficial to the general consumer as they get recommendations on products that they actually are interested in which saves much of their time and effort. So, it is a win-win for both parties. But serious allegations can be charged against the company for data privacy if proof is found for a potential data breach.

CONCLUSION AND RECOMMENDATIONS

This project evaluated how Beta function came to be and how Stirling's formula is implemented in calculating the magnitude of this function. The Beta function was then used to produce Beta distribution for finding the probability of whether people will watch a particular YouTube video and that probability was calculated using a computer algorithm that used the Beta distribution by implementing Stirling's approximation. The result was then used in unison

with Bayesian inference to update the system if the prior beliefs (probability data) turned out to be inaccurate or inconsistent with the actual trends and data values.

The findings of this project fit with my original assumption that YouTube is utilizing user data to successfully exploit media censorship, but it was only after a lot of work and study that I discovered that this strategy is being utilized by a variety of businesses in today's world. Using my own small data package to track the probability with the Beta distribution I calculated the likelihood using Bayes' Theorem. This led me to a 20 percent chance for the number of views and utilizing the Beta Distribution through a computer algorithm I arrived at a similar outcome in this case.

I would recommend a revision of this project and this problem with a real set of data (from any company) to evaluate how the functions on the computer handle these equations in real life and that would give us a better idea about how Bayesian inference is implemented to optimize recommendation patterns and how it contributes to the advancement of Artificial Intelligence. I also recommend that a research study should be conducted studying the potential drawbacks of Machine Learning for our future generations. Data might be misused by corporations for gaining an edge over both political and socio-economic sectors and so it is important to be aware of the worst possible scenario.

https://scholarcommons.usf.edu/ujmm/vol11/iss1/1
DOI: https://doi.org/10.5038/2326-3652.11.1.4920

NOMENCLATURE

Symbols	Description
Γ	Notation for Gamma Function
α	Parameter of Beta Function (Represents Success of event)
β	Parameter of Beta Function (Represents Failure of event)
Θ	Probability of our event occurring
B	Notation for Beta Function
$n!$	Notation for n factorial
$R_{m,n}$	Notation for the remainder term in the Euler–Maclaurin formula
B_k	Notation for Bernoulli's Number
$P(\Theta)$	Prior Distribution (prediction before seeing the actual data)
$P(data\|\Theta)$	Likelihood/Sampling Distribution (model of data given the parameter Θ
$P(\Theta\|data)$	Posterior Distribution (upon updating becomes new prior)

REFERENCES

Askey R. A., Roy, R. NIST Digital Library of Mathematical Functions,

Chapter 5: *Gamma Function*; 5.12: *Beta Function*; https://dlmf.nist.gov/5.12.

Artin, E. (2015). *The gamma function*. Dover Publications, Inc.

Bernoulli, Dutka, Euler, Gussov, Schneider, & Brunel. (1769, January 1). *The early history of the factorial function*. Archive for History of Exact Sciences. https://link.springer.com/article/10.1007/BF00389433.

Brooks-Bartlett, J. (2018, September 7). *Probability concepts explained: Bayesian inference for parameter estimation.* Medium. https://towardsdatascience.com/probability-concepts-explained-bayesian-inference-for-parameter-estimation-90e8930e5348.

Conrad, K. *Stirling's Formula*. Retrieved May 7, 2020, from https://kconrad.math.uconn.edu/blurbs/analysis/stirling.pdf

Davis, P. J. Leonhard Euler's Integral: A Historical Profile of the Gamma Function: In Memoriam: Milton Abramowitz,

The American Mathematical Monthly (1959), V.66 (10), 849-869;

https://www.jstor.org/stable/2309786 .

Gamma function. (n.d.) Gamma function - Rosetta Code. https://rosettacode.org/wiki/Gamma_function.

DOI: https://doi.org/10.5038/2326-3652.11.1.4920

Hahn, G. J., & Shapiro, S. S. (1994). *Statistical models in engineering*. John Wiley & Sons.

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). Continuous univariate distributions. In *Continuous univariate distributions*. essay, John Wiley and Sons.

Keeping, E. S. (1995). *Introduction to statistical inference*. Dover Publications.

Kim, A. (2019, December 22). *Gamma Function-Intuition, Derivation, and Examples*. Medium. https://towardsdatascience.com/gamma-function-intuition-derivation-and-examples-5e5f72517dee.

Kim, A. (2020, January 23). *Bayesian Inference-Intuition and Example*. Medium. https://towardsdatascience.com/bayesian-inference-intuition-and-example-148fd8fb95d6.

Kim, A. (2020, January 31). *Beta Distribution-Intuition, Examples, and Derivation*. Medium. https://towardsdatascience.com/beta-distribution-intuition-examples-and-derivation-cf00f4db57af.

Wadsworth, G. P. (1960, January 1). *Introduction to probability and random variables*: *Wadsworth, George P. (George Proctor), 1906- : Free Download, Borrow, and Streaming*. Internet Archive. https://archive.org/details/introductiontopr0000wads.

Wikimedia Foundation. (2020, April 3). *Stirling's approximation*. Wikipedia. https://en.wikipedia.org/wiki/Stirling's_approximation.

APPENDIX

SOURCE CODE:

IMPLEMENTATION OF STIRLING's FORMULA

Stirling's Approximation implementation Source Code [Source Code]. Retrieved from:

https://rosettacode.org/wiki/Gamma_function#Java

```
public class GammaFunction {


      public double st_Gamma(double x){
            return Math.sqrt(2*Math.PI/x)*Math.pow((x/Math.E), x);
      }


      public double la_Gamma(double x){
            double[] p = {0.99999999999980993, 676.5203681218851, -
1259.1392167224028,
                          771.32342877765313, -176.61502916214059,
12.507343278686905,
                          -0.13857109526572012, 9.9843695780195716e-6,
1.5056327351493116e-7};
            int g = 7;
            if(x < 0.5) return Math.PI / (Math.sin(Math.PI * x)*la_Gamma(1-
x));


            x -= 1;
            double a = p[0];
            double t = x+g+0.5;
```


DOI: https://doi.org/10.5038/2326-3652.11.1.4920

```
            for(int i = 1; i < p.length; i++){
                a += p[i]/(x+i);
            }

            return Math.sqrt(2*Math.PI)*Math.pow(t, x+0.5)*Math.exp(-t)*a;
    }

    public static void main(String[] args) {
            GammaFunction test = new GammaFunction();
            System.out.println("Gamma \t\tStirling \t\tLanczos");
            for(double i = 1; i <= 20; i += 1){
                System.out.println("" + i/10.0 + "\t\t" +
test.st_Gamma(i/10.0) + "\t" + test.la_Gamma(i/10.0));
            }
    }
}
```

Output:

```
Gamma               Stirling            Lanczos
0.1         5.697187148977169       9.513507698668734
0.2         3.3259984240223925      4.590843711998803
0.3         2.3625300362696198      2.9915689876875904
0.4         1.8414763359362354      2.218159543757687
0.5         1.5203469010662807      1.7724538509055159
0.6         1.307158857448356       1.489192248812818
0.7         1.15905329211392        1.2980553326475577
0.8         1.0533709684256085      1.1642297137253035
0.9         0.9770615078776954      1.0686287021193193
1.0         0.9221370088957891      0.9999999999999998
```

1.1	0.8834899531687038	0.9513507698668735
1.2	0.8577553353965909	0.9181687423997607
1.3	0.8426782594483921	0.8974706963062777
1.4	0.8367445486370817	0.8872638175030757
1.5	0.8389565525264963	0.8862269254527586
1.6	0.8486932421525738	0.8935153492876909
1.7	0.865621471793884	0.9086387328532916
1.8	0.8896396352879945	0.9313837709802425
1.9	0.9208427218942293	0.9617658319073877
2.0	0.9595021757444916	1.0000000000000002

GRAPH PLOT CODE:

Aerin, Kim (2020) Probability Graph Plot Source Code [Source Code] Retrieved from:

https://towardsdatascience.com/bayesian-inference-intuition-and-example-148fd8fb95d6.

```
import numpy as np
np.set_printoptions(threshold=100)

# Generating 2,000 readers' reponse.
# Assuming the claps follow a Bernoulli process - a sequence of binary (success/failure)
random variables.
# 1 means clap. 0 means no clap.

# We pick the success rate of 30%.
clap_prob = 0.3

# IID (independent and identically distributed) assumption
clap_data = np.random.binomial(n=1, p=clap_prob, size=2000)



import scipy.stats as stats
import matplotlib.pyplot as plt
```


DOI: https://doi.org/10.5038/2326-3652.11.1.4920

```
a = 400
b = 2000 – a

# domain θ
theta_range = np.linspace(0, 1, 1000)

# prior P(θ)
prior = stats.beta.pdf(x = theta_range, a=a, b=b)

# Plotting the prior distribution
plt.rcParams['figure.figsize'] = [20, 7]
fig, ax = plt.subplots()
plt.plot(theta_range, prior, linewidth=3, color='palegreen')

# Add a title
plt.title('[Prior] PDF of "Probability of Claps"', fontsize=20)

# Add X and y Label
plt.xlabel('θ', fontsize=16)
plt.ylabel('Density', fontsize=16)

# Add a grid
plt.grid(alpha=.4, linestyle='--')

# Show the plot
plt.show()



# The sampling dist P(X|θ) with a given clap_prob(θ)

likelihood = stats.binom.pmf(k = np.sum(clap_data), n = len(clap_data), p = clap_prob)


# Domain (# of claps)
X = np.arange(0, len(clap_data)+1)

# Likelihood P(X|θ) for all X's
likelihood = stats.binom.pmf(k = X, n = len(clap_data), p = clap_prob)

# Create the plot
fig, ax = plt.subplots()
plt.plot(X, likelihood, linewidth=3, color='yellowgreen')

# Add a title
plt.title('[Likelihood] Probability of Claps' , fontsize=20)
```

```
# Add X and y Label
plt.xlabel('X', fontsize=16)
plt.ylabel('Probability', fontsize=16)

# Add a grid
plt.grid(alpha=.4, linestyle='--')

# Show the plot
plt.show()



# (cont.)

theta_range_e = theta_range + 0.001

prior = stats.beta.cdf(x = theta_range_e, a=a, b=b) - stats.beta.cdf(x = theta_range, a=a, b=b)
# prior = stats.beta.pdf(x = theta_range, a=a, b=b)

likelihood = stats.binom.pmf(k = np.sum(clap_data), n = len(clap_data), p = theta_range)

posterior = likelihood * prior
normalized_posterior = posterior / np.sum(posterior)



fig, axes = plt.subplots(3, 1, sharex=True, figsize=(20,7))
plt.xlabel('θ', fontsize=24)

axes[0].plot(theta_range, prior, label="Prior", linewidth=3, color='palegreen')
axes[0].set_title("Prior", fontsize=16)

axes[1].plot(theta_range, likelihood, label="Likelihood", linewidth=3, color='yellowgreen')
axes[1].set_title("Sampling (Likelihood)", fontsize=16)

axes[2].plot(theta_range, posterior, label='Posterior', linewidth=3, color='olivedrab')
axes[2].set_title("Posterior", fontsize=16)
plt.show()
```

https://scholarcommons.usf.edu/ujmm/vol11/iss1/1
DOI: https://doi.org/10.5038/2326-3652.11.1.4920