Detection Problems in Assuming Gaussian Distribution for Sum of Independent Random variables

Narendra and Santosh
github link of the Project

Introduction – In estimation and detection theory, many problems have closed form solutions for Gaussian distribution. We also know that distribution of the sum of independent and identically distributed random variables converges towards gaussian by the central limit theorem. The distribution of the underlying random variables doesn't matter. It is assumed that central limit theorem doesn't take sum of too many I.I.D random variables to converge towards a gaussian distribution. In nature, many times we observe the sum of I.I.D random variables. Hence, it is assumed that detection and estimation methods that work for normal distributions can be applicable to many problems involving other types of distributions. Based on the central limit theorem, Gaussian distribution is used in every fields like engineering, economics, statistics and sciences. Central limit theorem is considered central to the statistics and some people even assume that name of the theorem comes from that.

But Normal or Gaussian distribution is a very special type of distribution where probability of random variable goes down as a function of exponential squared. This means that probability of finding a random variable value away from mean is very low i.e., 99.7% times random variable

stays within 3 standard deviations (sigma) away from the mean and probability of 6sigma or 7sigma events is almost zero.

In this project, we will examine consequence of making the Gaussian distribution assumption in detection errors especially for rare events. We will look at the behavior of the normalized sum of random variables and the variables can take any distribution and test framework is built in such a way that Expectation and variance of the distribution will be estimated from the data before continuing to test our hypothesis. We believe our test framework which is written both in Matlab, Python will help anyone who wants to study the central limits and pose questions like how many random variables for a certain distribution are summed up so that distribution converges to Normal distribution. It can test with any heavy tail distribution to see the deviation from the Normal distribution and if the assumption can cause serious errors in detection and others.

$$S_k = \frac{(X_1 + X_2 + \ldots + X_k) - \mathrm{E}[X_i] * \mathrm{K}}{\sqrt{K * \mathrm{Var}[X_i]}}$$

According to the central limit theorem this random variable S_k should converge towards the normal distribution. As we are now going to delve into the science of randomness, first things first. How do some well used scientific software generate random variables and Are they really completely Random?!

How to generate an random number (Pseudo) and Independent Variable!:

As our arguments on central limit theorem are based on empirical results, its good to understand how scientific software generate random variables and how do they make sure (do they?) they are independent on every iteration. The following paragraph is good to know for all those study the science of uncertainty!

Random number generators Generating (pseudo-) random numbers X ~ U(0,1) is fundamental to all experimental statistics, simulation, experiment design, and data analysis. Programming

languages generally use a system-supplied random number generator, like x = rand or x = rand(seed) where seed is an integer a starting value which will be used to "seed" the random number generator. Research on generating good random number generators was recently matured. So, if you're using some good old software and expecting to make strong conclusions, please think again.

Numerical Recipes contains examples of good, easy to port random number generators. But the simply way to generate is to use generators that are part of a well know software packages which are based on the so-called Mersenne Twister (Matsumoto & Nishimura 1997) used by MATLAB [mt19937ar] since about 2008 is considered to be of high quality - it has passed a number of stringent test case for uncertainty, including the 'Diehard' test suite (Marsaglia 1998)

When MATLAB is started, and you ask for to generate four random numbers using the built in function rand(),

you get>> rand(2,2)

ans = *0.814723686393179 0.126986816293506*

0.905791937075619 0.913375856139019

>> rand(1) ans = 0.632359246225410

If you exit MATLAB and restart again,

you get for example: >> rand(2,3)

ans = *0.814723686393179 0.126986816293506 0.632359246225410*

0.905791937075619 0.913375856139019 0.097540404999410

After restarting MATLAB, the default random stream is initialized with the a hard coded seed(=0)

How Python generates random variables:

To compute uniform, normal (Gaussian), lognormal, negative exponential, gamma, and beta distributions. For generating distributions of angles, python packages make use of von Mises distribution. Almost all module functions depend on the basic function random, which generates a random float uniformly in the semi-open range [0.0, 1.0). Python uses the Mersenne Twister like Matlab as the core generator. It produces 53-bit precision floats and has a period of 2**19937 (hinting that it is pseudo random after this runs in the simulation or if restart matlab). The underlying implementation in C is both fast and threadsafe. The Mersenne Twister is one of the most extensively tested random number generators in existence. However, being completely deterministic, it is not suitable for all purposes, and is completely unsuitable for cryptographic purposes.

The functions supplied by this module are actually bound methods of a hidden instance of the random. Random class. Python gives freedom to instantiate own instances of Random to get generators that don't share state.

Simulation Setup*:* A typical detection problem where the observations are distributed for a sum of independent and identically distributed random variables. Our attribute set has two parameters with equal prior probability. Threshold Detection over Likelihood Ratio Test is performed and empirical error rates are compared with Gaussian distribution for the different distributions. Monte carl simulations are ran for 10000000 iterations, varying the numbers of random variables(k=[1,100]) in the sum and also changing the difference between the expected value of the respective conditional probability distributions(μ=[0.5,7]). The following sections share the same simulation setup as explained above

Sum of the random variables are distribute as below.

$$S_k = \frac{(X_1 + X_2 + \ldots + X_k) - \mathrm{E}[X_i] * \mathrm{K}}{\sqrt{K * \mathrm{Var}[X_i]}}$$

In the following section we will explore differently with independent random variables with different distributions like Uniform, Exponential, Power Laws and check empirically limits the Central Limit Theorem. In the each section below we present a set of interesting empirical results.

Sum of I.I.D Uniform Random Variables/Irwin-Hall Distribution: The Irwin–Hall distribution is the continuous probability distribution for the sum of ***N*** independent and identically distributed Uniform random variables with mean 0.5 and variance 1/12 i.e Uniform distribution between **U[0,1]**

$$X = \sum_{k=1}^{n} U_k$$

The probability density function (pdf) is given by

$$f(x,n) = \frac{1}{2(n-1)!}\sum_{k=0}^{n}(-1)^k {}_k^n C(x-k)^{n-1} sgn(x-k)$$

where ***sgn(x − k)*** denotes the sign function:

$$sgn(x-k) = \begin{cases} 1 & x > k \\ 0 & x = 0 \\ -1 & x < k \end{cases}$$

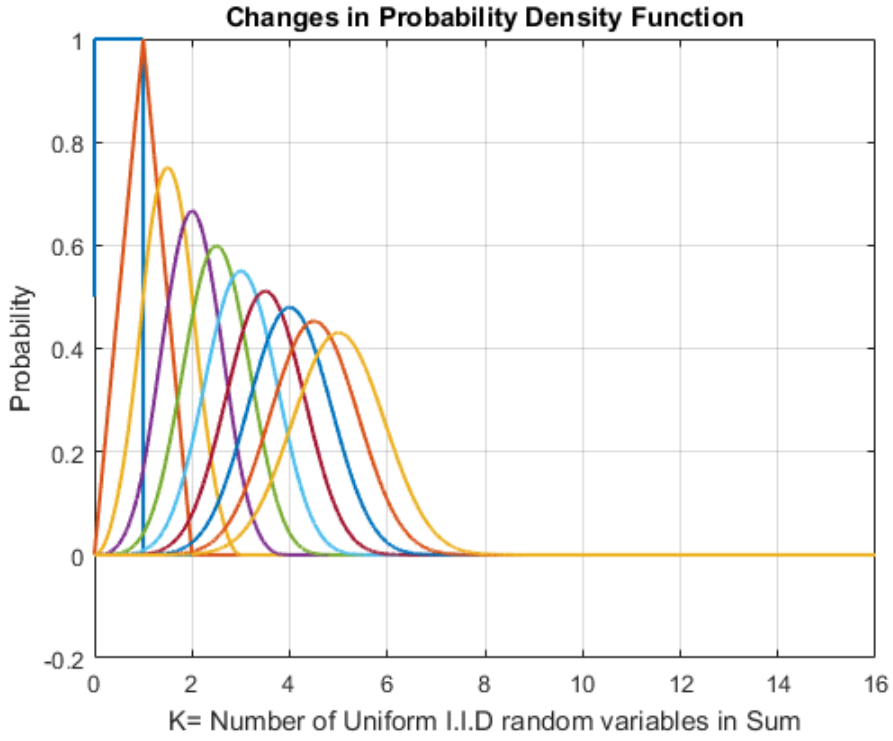
Thus the pdf is a spline (piecewise polynomial function) of degree *n − 1* over the knots 0, 1, ..., *n*. In fact, for *x* between the knots located at *k* and *k + 1*, the pdf is equal to

$$f(x;n) = \frac{1}{2(n-1)!}\sum_{j=0}^{n-1} a_j(k,n)\,(x)^j$$

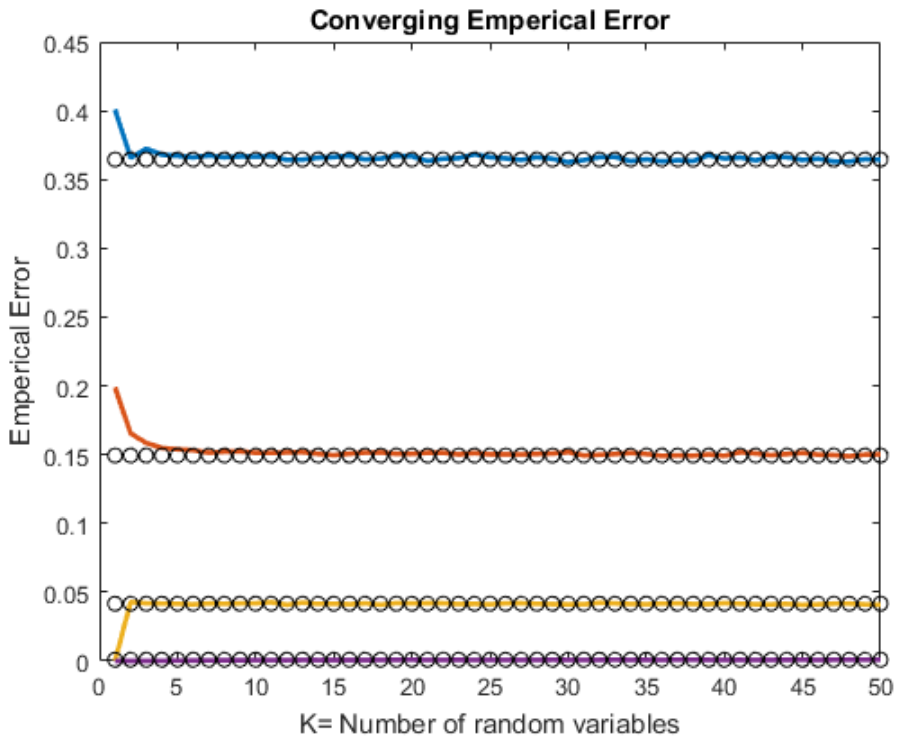
$$a_j(k,n) = \begin{cases} 1 & k = 0, j = n-1 \\ 0 & k = 0, j < n-1 \\ a_j(k-1,n) + (-1)^{n+k-j-1}\binom{n}{k}\binom{n-1}{j}k^{n-j-1} & k > 0 \end{cases}$$

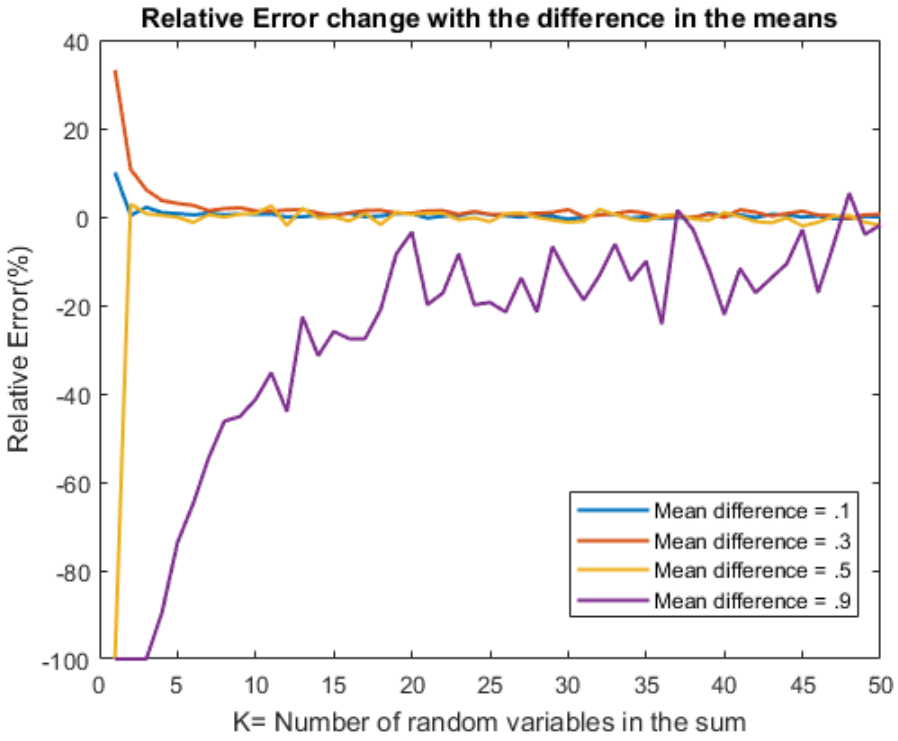
The mean and variance are ***N/2*** and ***N/12***, respectively.

CDF is given by: $\frac{1}{(n)!}\sum_{j=0}^{\lfloor x \rfloor} -1^k\binom{n}{k}(x-k)^n$



The above plot shows that with increase in the numbers of I.I.D uniform random variables in the sum distribution the probability density function becomes bell shaped, shorter at the mean and the mean and variance of the respective distribution changes. Therefore, the following plots must show a converging behavior and so is the case below.


Converging Emperical Error
Emperical Error
K= Number of random variables
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
35
40
45
50


Relative Error change with the difference in the means
Relative Error(%)
K= Number of random variables in the sum
Mean difference = .1
Mean difference = .3
Mean difference = .5
Mean difference = .9
40
20
0
-20
-40
-60
-80
-100
0
5
10
15
20
25
30
35
40
45
50

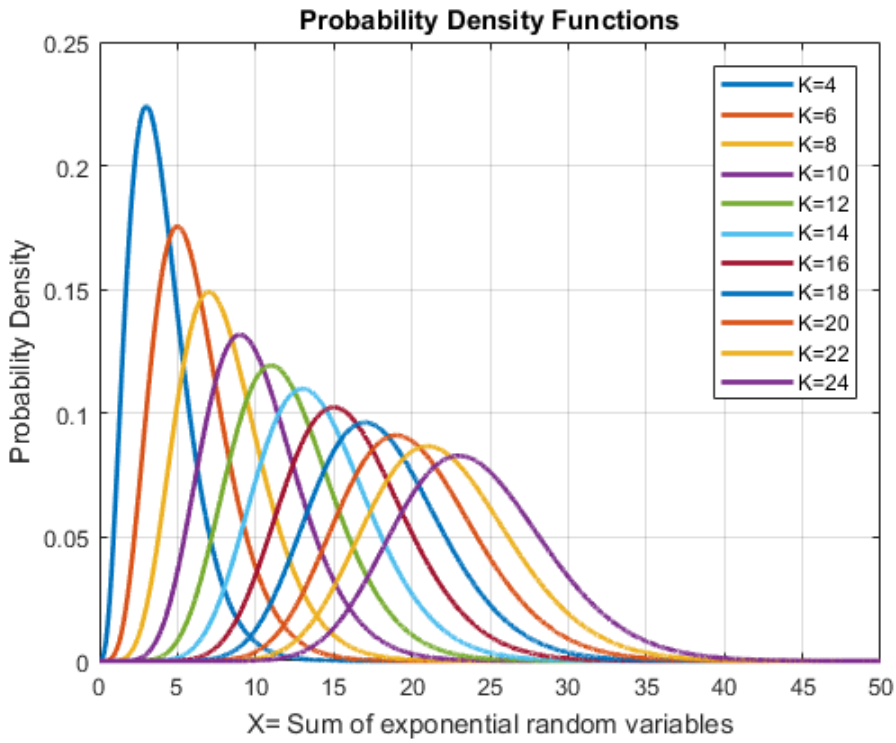
Empirical detection error converges to the error of the Gaussian distribution and also the interesting part in the behavior is that the more tail spread is in the detection region, the converges rate is higher, implies that heavy tails are more error prone and Gaussian assumption for the application like big data will result in good of false positives.

Sum of Exponential Independent Random Variables:

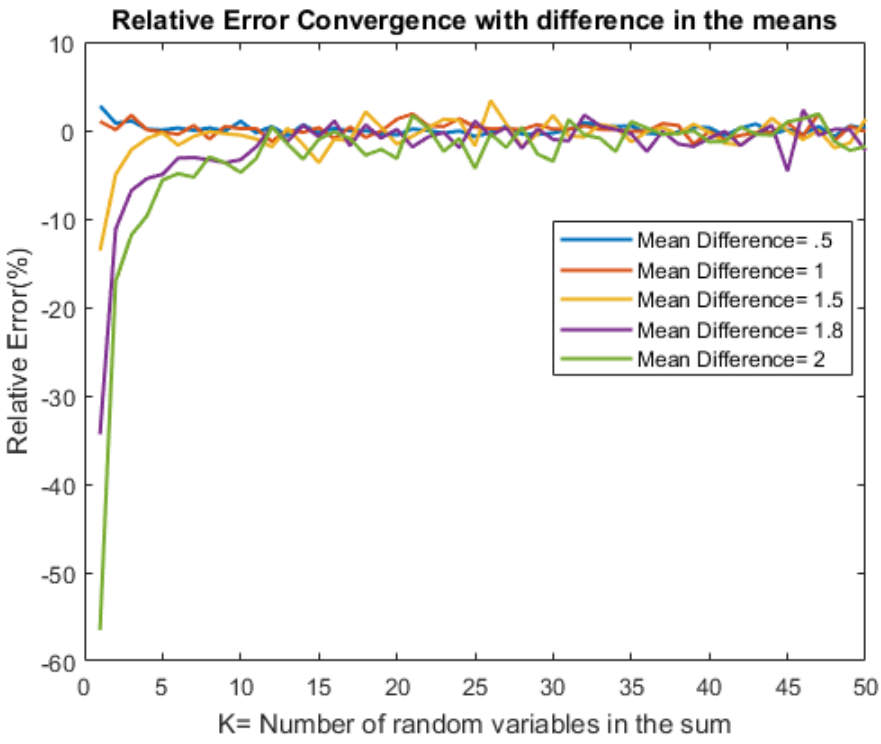
Erlang Distribution: Sum of k independent and identically distributed exponential random variables with parameter μ are erlang distributed and probability density function is given below

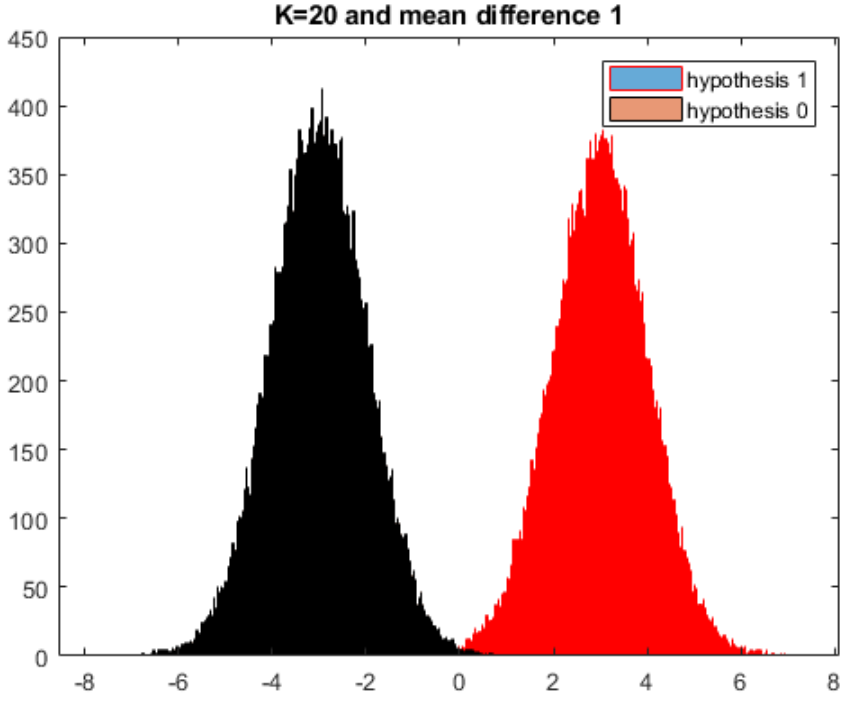
$$f(x, k, \mu) = \mu^k x^{k-1} e^{-\mu x} \quad x, \mu \geq 0$$

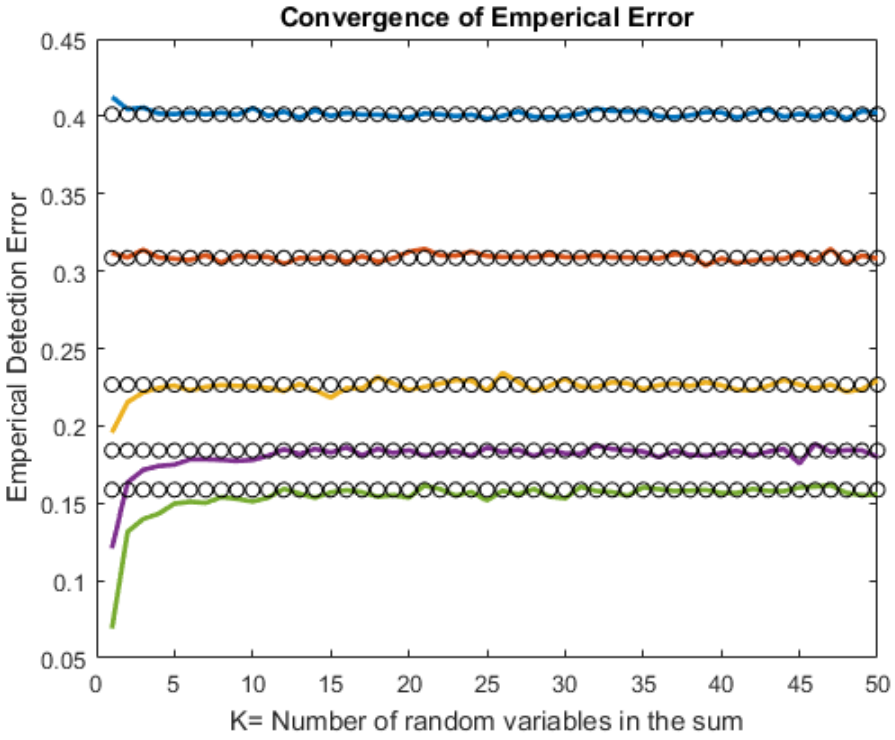
From the below plot we can observe that with increase in k , tail is increasing and density and the center/mean is decreasing and after certain k the shape of distribution becomes bell shaped, so the probability of error also converges towards what we can estimate with gaussian noise.



The above plot shows that with increase in the numbers of I.I.D exponential random variables in the sum distribution the probability density function becomes bell shaped, shorter at the mean and the mean and variance of the respective distribution changes. Therefore, the following plots must show a converging behavior and so is the case below. Empirical detection error converges to the error of the Gaussian distribution and also the interesting part in the behavior is that the more tail spread is in the detection region, the convergence rate is higher, implies that heavy tails are more error prone and Gaussian assumption for the application like big data will result in good of false positives. Also, with increase in difference between Expected values of the conditional density functions more number of random variables in the sum distribution are needed to converge to the Gaussian error.

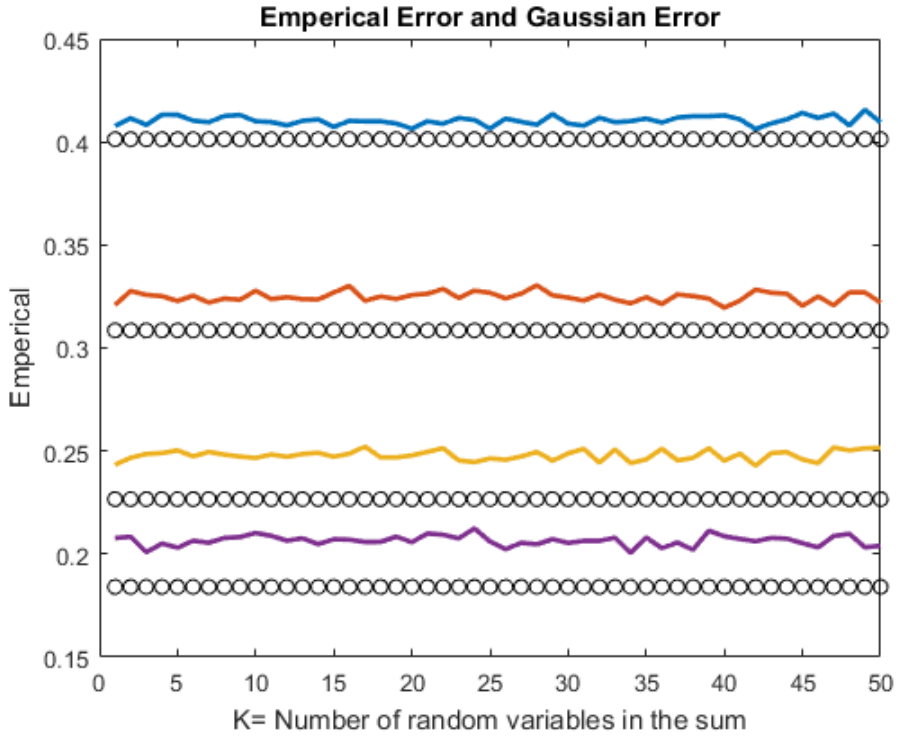

Relative Error Convergence with difference in the means
Relative Error(%)
K= Number of random variables in the sum
Mean Difference= .5
Mean Difference= 1
Mean Difference= 1.5
Mean Difference= 1.8
Mean Difference= 2
10
0
-10
-20
-30
-40
-50
-60
0
5
10
15
20
25
30
35
40
45
50

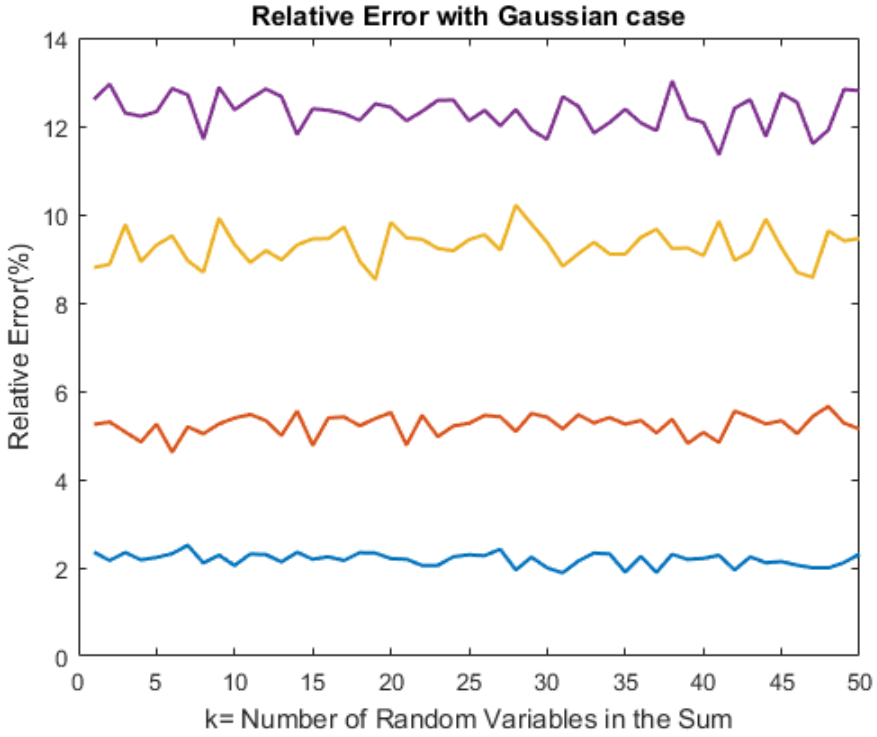

K=20 and mean difference 1
hypothesis 1
hypothesis 0
450
400
350
300
250
200
150
100
50
0
-8
-6
-4
-2
0
2
4
6
8



Sum of Correlated Gaussian Random Variables: We generated correlated normal random variables with 2 tap filter with configurable coefficients. So, code framework can be reused by anyone who wants to study the central limit theorem empirically and gain some insights.

Empirical detection error converges to the remains constant as expected but is never equal to a Gaussian error in this case. Even though the Central Limit Theorem is still valid for weakly dependent Gaussian Identically distributed random variables the error never converges to the Gaussian case as it is dependent.


Emperical Error and Gaussian Error
Emperical
K= Number of random variables in the sum
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0
5
10
15
20
25
30
35
40
45
50


Relative Error with Gaussian case
Relative Error(%)
k= Number of Random Variables in the Sum
14
12
10
8
6
4
2
0
0
5
10
15
20
25
30
35
40
45
50

Power Law distributed random variables: A power law is the form of an important relationship taken by two quantities and is a relation of the type $= ax^{a-1}$, where Y and X are interested quantities, ***a*** is called the power law exponent. In nature, many relationships follow power laws. Surface area to volume, the inverse-square laws of Newtonian gravity, fractals, earthquake intensity, the sizes of wars, etc. Most of the economic relationships also take the form of power laws like the distribution of income, wealth, size of cities and firms, distribution of financial variables such as returns and trading volume. Power laws are also the heavy tail distributions, meaning that frequency of rare events is much higher than that of the known events. We are particularly interested in the distribution of the sum of independent and identically distributed power laws and we presented number results in the following section and found the key issues in assuming such distribution as Gaussian either knowingly or unknowingly through a detection problem. Rigorous simulations for multiple scenarios are performed to detect signals from a sum of I.I.D distributed power laws and detection errors are compared with that of the Gaussian distributed to argue on our hypothesis.

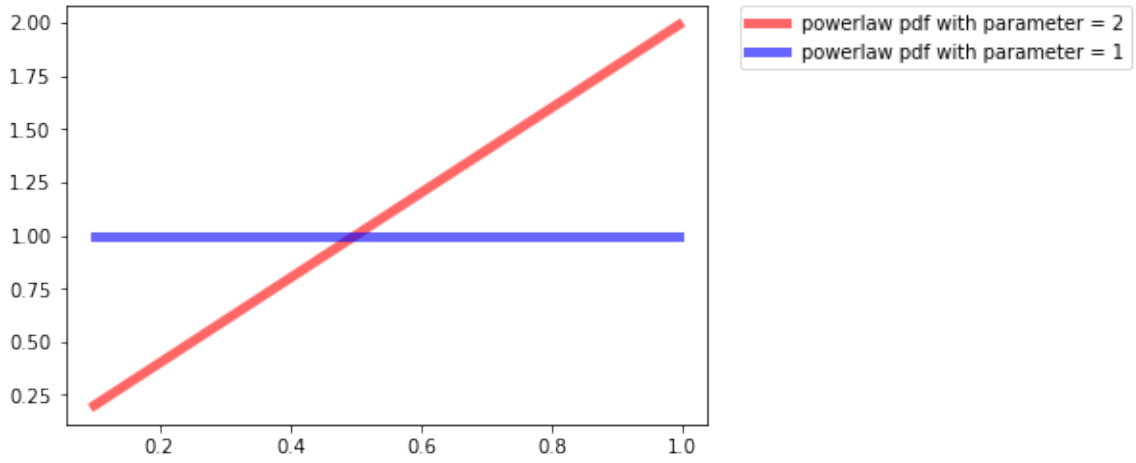
Experiments –

In all our experiments we assume that there are two separate hypotheses 0 and 1 which generate two different normalized sums. The normalized sum are separated from each other by adding a fixed mean difference to hypothesis 0. The detector is designed by assuming both hypothesis 0 and 1 behave like Gaussian. We compare the empirical error and Gaussian error of the detector.

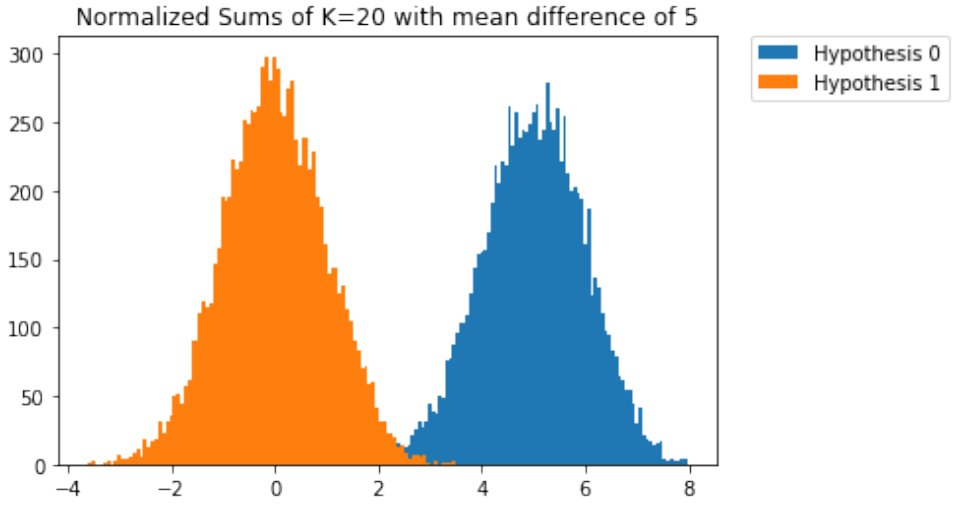
$$Detector\ error\ gaussian = Q(\frac{\mu 0 + \mu 1}{2})$$

$$Relative\ \%\ Error = \frac{100 * (Detector\ empirical\ error - Detector\ error\ gaussian)}{Detector\ error\ gaussian}$$

Experiments with power law – In this set of experiments we assume that all Hypothesis 1 random variables are generated from a power law distribution with parameter $a = 1$. This is same as uniform distribution. Hypothesis 0 random variables were generated from power law distribution with parameter 0.9, 2, 4, 10 and 40. Figure 1 shows the power law pdf with parameters 1 and 2. The higher parameter value in power law makes the distribution heavy tail.

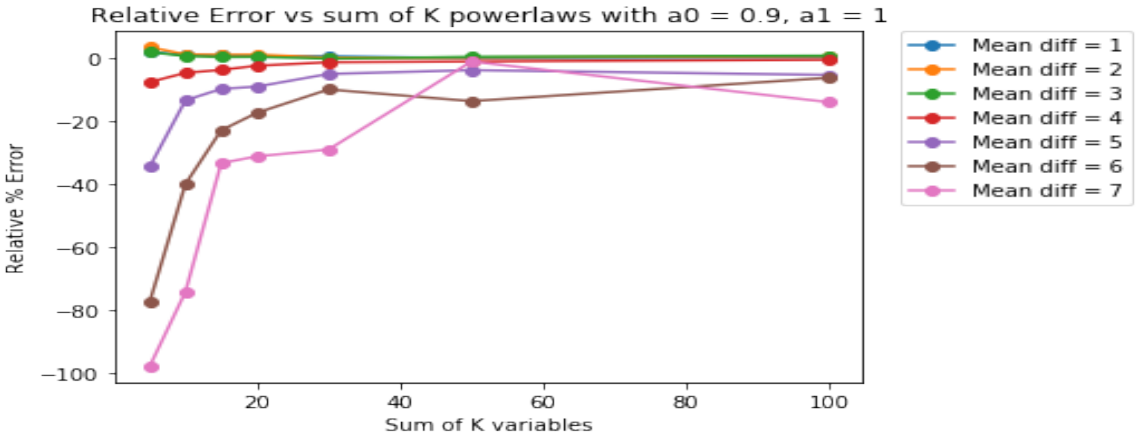


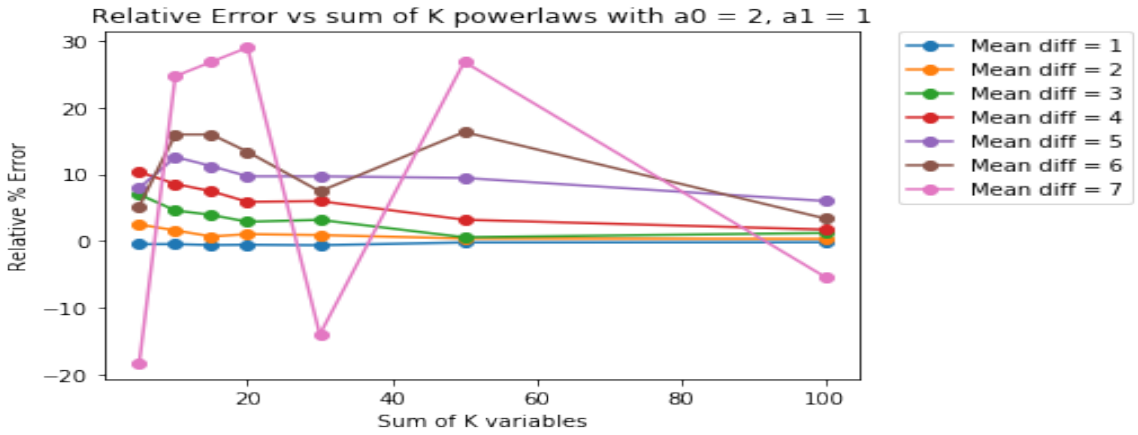
Our observations are for the normalized sum of K random variables separated by a given mean difference. Figure 2 shows how the observations associated with different hypothesis look like. We use the threshold detector assuming Gaussian distribution. After that we empirically try to determine the probability of error for this detector and compare it to the theoretical Gaussian error. We do this comparison for different value of K (number of random variable that we sum) and mean difference.

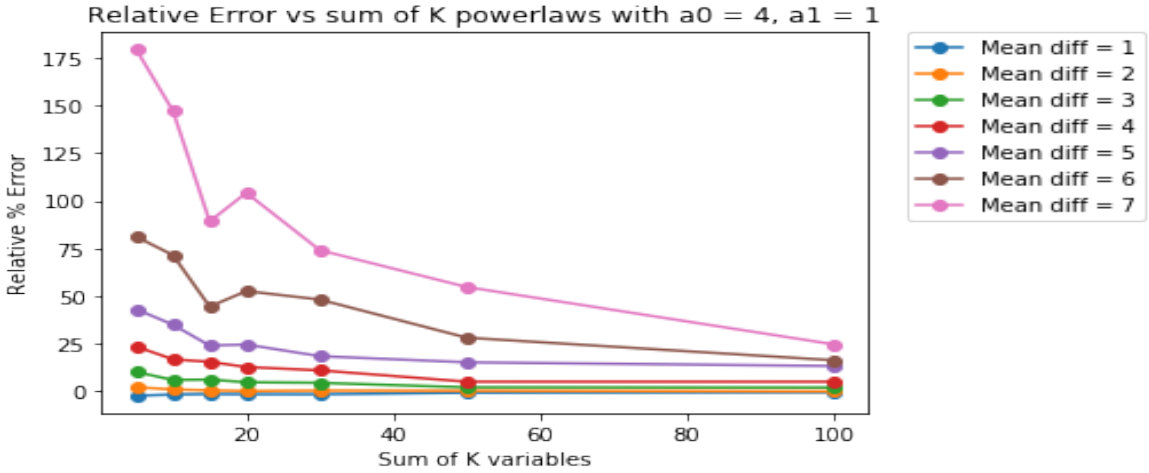


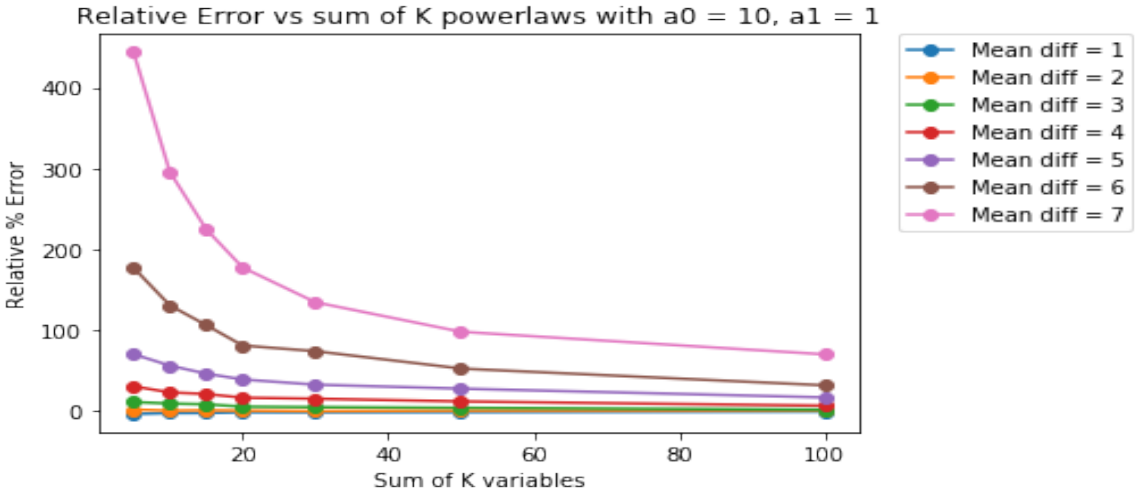
Results –

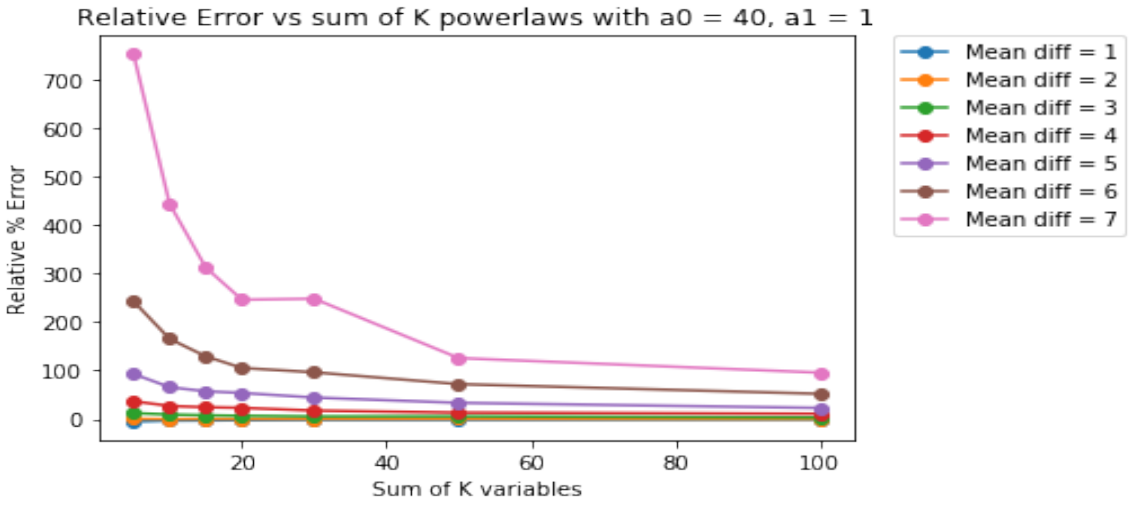
Figure 3-7 show the plot of relative error with respect to sum of ***K*** random variables in which hypothesis 0 has parameter ***a0*** and hypothesis 1 has parameter a1 = 1.




Relative Error vs sum of K powerlaws with a0 = 2, a1 = 1
Relative % Error
Sum of K variables
Mean diff = 1
Mean diff = 2
Mean diff = 3
Mean diff = 4
Mean diff = 5
Mean diff = 6
Mean diff = 7


Relative Error vs sum of K powerlaws with a0 = 4, a1 = 1
Relative % Error
Sum of K variables
Mean diff = 1
Mean diff = 2
Mean diff = 3
Mean diff = 4
Mean diff = 5
Mean diff = 6
Mean diff = 7


Relative Error vs sum of K powerlaws with a0 = 10, a1 = 1
Relative % Error
Sum of K variables
Mean diff = 1
Mean diff = 2
Mean diff = 3
Mean diff = 4
Mean diff = 5
Mean diff = 6
Mean diff = 7



As expected the relative error improves as we take the sum of larger number of random variables. What is unexpected is that Relative error is large when mean difference is large. Also, relative error is very high when power law parameter is 40 as compared to when it is 2 or 4. Even with sum of 100 IID random variables the relative error stays large. The reasons for these are effect of tail probabilities. The relative error explodes as Gaussian tail probability goes down much faster than actual tail probability. This behavior cannot be easily seen as low values of probabilities in both Gaussian and non-Gaussian distribution hide the actual relative difference.

Conclusions:

1. We have shown from our experiments that Gaussian approximation for sum of independent and identically distributed variables only holds closer to the mean. These assumptions break down for tail probabilities.
2. The central limit theorem only works closer to the mean, hence the name central. The use of Gaussian distribution in detection of rare events or risk management can be highly flawed if underlying distribution has heavy tail.

3. With increase in the number of identical and independent random variables in the sum of random variables the detection error converges to the detection error made for the Gaussian distributed.
4. Initially with small ***k,*** the detection errors may be better or worse than that of Gaussian distribution, but later detection errors converge to Gaussian case.
5. The assumption of Gaussian distribution is dangerous for heavy tailed distributions and relative error percentage is huge.
6. We think that Gaussian distribution should not be used for failure analysis or any type of risk management even if the assumption of IID random variable holds.

With pleasure we convey our most of the references are Lectures and the Notes and extended arguments (sometimes little heated!) between both of us. Apart from that here and there of Wikipedia and documentation of Python, Matlab.