STATS 4T06 Senior Research Project

By:Justin Chen

Supervisor:Narayanaswamy Balakrishnan

September 2022

Contents

1 Introduction to Poisson Process **2**
1.1 Key Properties 2
1.2 Different Types of Poisson Process 2

2 The Weighted Poisson Distribution **3**

3 The Weighted Poisson Process **4**
3.1 Building a Weighted Poisson Process 5
3.1.1 Step One 5
3.1.2 Step Two 5
3.1.3 Step Three 5
3.1.4 Step Four 5

4 Falling Factorial Weight Function Example **5**
4.0.1 Falling Factorial at l=2 example 7

5 Dispersion Derivation **8**
5.1 Underdispersed Model 8
5.2 Overdispersed Model 9
5.2.1 The general inverse falling factorial starting at (n+1) . . . 12
5.2.2 Proof 13

6 Simulation Study **13**
6.1 Description of Simulation Method 14
6.2 Result 17
6.2.1 Traditional Poisson Dispersion plot 17
6.2.2 Falling Factorial Poisson Dispersion Plots 18
6.2.3 Inverse Rising Factorial Poisson Dispersion Plots 20
6.3 Observations 22

7 Conclusion **22**

8 Bibliography 22

Abstract

The classical Poisson process assumes equidispersion (equal mean and variance) however this is not always the case when we analyze empirical distributions. The weighted Poisson process accounts for this discrepancy by assigning a weight function to the Poisson distribution PMF. This paper will derive the dispersion of two weighted Poisson models. One with a falling factorial weight and another with an inverse rising factorial starting at n+1. The two models will be underdispersed and overdispersed respectively, a simulation study will then be conducted to plot the change in dispersion over time and to compare with the traditional Poisson Process.

1 Introduction to Poisson Process

A Poisson Process is a mathematical object that contains elements that are randomly placed within a mathematical space. The Poisson Process can be used to model random events such as image processing, telecommunications, queuing theory, etc.

1.1 Key Properties

Poisson Property: Points in a Poisson Process have a Poisson distribution:

$$P(N = n) = (\lambda^n/n!)(e^{-\lambda}) \tag{1}$$

Where,
N=Number of occurrences of the event
λ = Density of our points defined as a constant or an integrable function

Complete Independence: Say we have a collection of disjoint subregions that exist in the space we defined. The number of points in each subregion is independent of the other subregions. Meaning that the process is completely random. The points and subregions within the space are independent.
A Poisson Process must have these two properties, complete independence implies the Poisson Property but the converse is not true.

1.2 Different Types of Poisson Process

Homogeneous:Given our density function Λ

$$\Lambda = v\lambda \tag{2}$$

$\lambda = constant$
v = Lebesgue measurement

In this context we can think of λ as the average number of points per unit of the Lebesgue Measurement.
We can define a counting process, in which we define the number of occurrences at time t

$$\{N(t), t \geq 0\} \tag{3}$$

Our distribution at time t would be

$$P(N(t) = n) = ((t\lambda)^n/n!)e^{-t\lambda} \tag{4}$$

We can also define this process on an interval $(a, b] \in R$

$$P(N(a, b] = n) = ((b - a)\lambda)^n/n!)e^{-(b-a)\lambda} \tag{5}$$

Inhomogeneous:We denote the density function over a region B $\subseteq [0, \infty)$
Our Poisson parameter is a locally integrable function defined on the space the Poisson process is defined

$$\Lambda(B) = \int_B \lambda(t)\, dt < \infty \tag{6}$$

We can define our number of occurrences over regions B

$$P\{N(B_i = n_i, i = 1, ..., k\} = \prod_{i=1}^{k} (\Lambda((B_i)^{n_i})/n_i!)e^{-\Lambda(B_i)} \tag{7}$$

Say we want to define these properties on the interval (a,b]

$$P\{N(a, b] = n\} = (\Lambda(a, b)^n/n!)e^{-\Lambda(a,b)} \tag{8}$$

$$\Lambda(a, b) = \int_a^b \lambda(t)\, dt \tag{9}$$

2 The Weighted Poisson Distribution

We should first have a brief discussion on the Weighted Poisson Distribution as it related to the future topics of dispersion and the selection of our weight functions. There is a unifying method for determining weighted distributions in general. Developed by Rao we get the following:
$f(x, \theta)$=pdf of random variable x and θ is a unknown parameter

$$g(x; \theta) = \frac{w(x)f(x; \theta)}{E[w(x)]}, x\epsilon R, \theta > 0$$

This was then applied to a Poisson Distribution. Thus a new family of Poisson Distributions was discovered. The paper by Castillo and Casany gave us the following Weight Poisson Distribution

$$P(k; \lambda, r, a) = \frac{(k + a)^r}{E^\lambda[(k + a)^r]} \frac{\lambda^k e^{-\lambda}}{k!}$$

This is defined as the Weighted Poisson Distribution. The usefulness of this model comes from the idea of the Index of Dispersion.

$$I(x) = V(x)/E(x)$$

where x is a random variable and V(x) and E(x) is the corresponding variance and expected value. A typical Poisson Distribution has a I(x) = 1. Indeed, we can characterize I(x) as a "degree of departure" from the Poisson random variable. This leads to the following theorem

Suppose that the Random Variable is X is distributed with a WPD(λ,r,a), $r\epsilon R$,$\lambda > 0$,$a > 0$. The index of Dispersion is I(x) = 1 iff $r = 0$ or, equivalently, X has a Poisson Distribution. Moreover, $I(x) > 1$ ($I(x) < 1$) iff $r < 0$ ($r > 0$)

This is a fundamental theorem of the Weighted Poisson Distribution that will help us select a weight function depending on whether we observe overdispersion or underdispersion. Which is useful when we observe empirical distributions that aren't equidispersed.

3 The Weighted Poisson Process

The first thing one can think about is the weighted Poisson distribution. Defined as

$$P(N^w = n) = \frac{w(n)p(n)}{E[w(N)]} n = 0, 1, 2, 3, ...$$

(10)

where w(·) is a non-negative weight function (on the set of non-negative integers Z+) with non-zero, finite expectation

$$0 < E[w(N)] = \sum_{n=0}^{\infty} w(n)p(n) < \infty$$

(11)

This can be converted into a stochastic process on the interval [0,1] with intensity λ , $\{N(t), t\epsilon N[0, 1]\}$ into a weighted one $\{N^w(t), t\epsilon N[0, 1]\}$

By lemma 1.1 For each t ϵ (0,1] the variable $N^w(t)$ defined in (1.5) has a weighted Poisson distribution with parameter t and weight function w_t where

$$w_t(k) = \sum_{n=0}^{\infty} \frac{[(1-t)\lambda]^n}{n!} w(n + k). \tag{12}$$

k=0,1,2,3,..

3.1 Building a Weighted Poisson Process

The core issue is how we are able to construct such a process. We will present a step-by-step guide on how one can do this.

3.1.1 Step One

Propose a weight function w(n) where n=0,1,2,3,... and w(n) is non negative

3.1.2 Step Two

By lemma 1.1 $N^w(t)$ has a weighted poisson distribution with parameter and weight function

$$w_t(k) = \sum_{n=0}^{\infty} \frac{[(1-t)\lambda]^n}{n!} w(n+k)$$

One needs to express this explicitly

3.1.3 Step Three

Calculate the expected value of the weight function in terms of the Poisson distribution

$$E[w_t(N)] = \sum_{k=0}^{\infty} p_t(k) w_t(k)$$

3.1.4 Step Four

With these values the following model can be derived

$$P(N^w(t) = n) = \frac{w_t(n)p_t(n)}{E[w_t(N)]}$$

This gives us explicitly, a Weighted Poisson Process

4 Falling Factorial Weight Function Example

The first weight function we will propose is $w(n) = n_l$ where l ϵ Z^+. A falling factorial is defined as,

$$n_l = n(n-1)(n-2)\cdots(n-(l-1))$$

Let us first derive the base case of w(n) = n

w$_t(k) = \sum_{n=0}^{\infty} \frac{[(1-t)\lambda]^n}{n!}(n+k)$

$=\sum_{n=1}^{\infty} \frac{[(1-t)\lambda]^n}{n!}(n) + k\sum_{n=0}^{\infty} \frac{[(1-t)\lambda]^n}{n!}$

$=\sum_{n=1}^{\infty} \frac{[(1-t)\lambda]^n}{(n-1)!} + ke^{(1-t)\lambda}$

$=\sum_{n=1}^{\infty} \frac{[(1-t)\lambda]^{n+1}}{(n)!} + ke^{(1-t)\lambda}$

$=(1\text{-t})\lambda e^{(1-t)\lambda} + ke^{(1-t)\lambda}$

$=\mathrm{e}^{(1-t)\lambda}((1-t)\lambda + k)$

(13)

Now we must calculate the expected value

$\mathrm{E}[\mathrm{w}_t(N)] = \sum_{n=0}^{\infty} \frac{e^{-\lambda t}(\lambda t)^k}{k!}(e^{(1-t)k}((1-t)\lambda + k))$

$=\sum_{n=0}^{\infty} \frac{e^{\lambda-2\lambda t}(\lambda t)^k}{k!}((1-t)\lambda + k))$

$=\mathrm{e}^{\lambda-2\lambda t}[(1-t)\lambda \sum_{n=0}^{\infty} \frac{(\lambda t)^k}{k!} + \lambda t \sum_{n=0}^{\infty} \frac{(\lambda t)^k}{k!}]$

$=\mathrm{e}^{\lambda-2\lambda t}e^{\lambda t}[(1-t)\lambda + \lambda t]$

$=\mathrm{e}^{\lambda(1-t)}$

(14)

We now construct our weighted Poisson Process as;

$$P(N^w(t) = k) = \frac{w_t(k)p_t(k)}{E[w_t(N)]} = \frac{e^{-\lambda t}(\lambda t)^k}{k!}[(1-t)\lambda + k] \tag{15}$$

We can then derive the expected value of this model

$$E[P(N^w = k)] = \sum_{k=0}^{\infty} \frac{ke^{-\lambda t}(\lambda t)^k}{k!}[(1-t)\lambda + k]$$

$$= e^{-\lambda t}[\sum_{k=1}^{\infty} \frac{k(\lambda t)^k[(1-t)\lambda]}{k!} + \sum_{k=1}^{\infty} \frac{k^2(\lambda t)^k}{k!}]$$

$$= e^{-\lambda t}[[(1-t)\lambda]\lambda t e^{\lambda t} + \lambda t[\sum_{k=0}^{\infty} \frac{k(\lambda t)^k}{k!} + \sum_{k=0}^{\infty} \frac{(\lambda t)^k}{k!}]]$$

$$= e^{-\lambda t}[((1-t)\lambda)\lambda t e^{\lambda t} + \lambda t(\lambda t e^{\lambda t} + e^{\lambda t})]$$

$$= \lambda t(\lambda + 1)$$

The variance is calculated in a similar way

$$Var[N^w(t) = k] = E[k^2] - (E[k])^2$$

$$E[k^2] = \sum_{k=0}^{\infty} \frac{k^2 e^{-\lambda t}(\lambda t)^k}{k!}[(1-t)\lambda + k]$$

$$= \lambda^3 t^2 + \lambda^2 t + 2(\lambda t)^2 + \lambda t$$

$$Var[N^w(t) = k] = -\lambda^4 t^2 + 3\lambda^3 t^2 + \lambda^2 t + (\lambda t)^2 + \lambda t$$

The weighted poisson process can also be derived for $n_{(2)}$

4.0.1 Falling Factorial at l=2 example

$$w(n) = n(n-1)$$

By the steps we outlined

$$w_t(k) = \sum_{n=0}^{\infty} \frac{[(1-t)\lambda]^n (n+k)((n-1)+k)}{n!}$$

$$= \sum_{n=0}^{\infty} \frac{n(n-1) + kn + k(n-1) + k^2}{n!}[(1-t)\lambda]^n$$

$$= \sum_{n=2}^{\infty} \frac{[(1-t)\lambda]^n}{(n-2)!} + 2k \sum_{n=0}^{\infty} \frac{[(1-t)\lambda]^k n}{n!} - k \sum_{n=0}^{\infty} \frac{[(1-t)\lambda]^n}{n!} + k^2 \sum_{n=0}^{\infty} \frac{[(1-t)\lambda]^n}{n!}$$

$$= e^{(1-t)\lambda}[((1-t)\lambda)^2 + 2k(1-t)\lambda - k + k^2]$$

The expected value of the weight function is as follows

$$E[w_t(N)] = \sum_{n=0}^{\infty} \frac{e^{-\lambda t}(\lambda t)^k}{k!} e^{(1-t)\lambda}[e^{(1-t)\lambda}[((1-t)\lambda)^2 + 2k(1-t)\lambda - k + k^2]$$

$$= e^{\lambda - 2\lambda t}[((1-t)\lambda)^2 \sum_{n=0}^{\infty} \frac{(\lambda t)^k}{k!} + 2(1-t)\lambda \sum_{n=0}^{\infty} \frac{k(\lambda t)^k}{k!} - \sum_{n=0}^{\infty} \frac{k(\lambda t)^k}{k!} + \sum_{n=0}^{\infty} \frac{k^2(\lambda t)^k}{k!}]$$

$$= e^{\lambda - 2\lambda t}[((1-t)\lambda)^2 e^{\lambda t} + 2(1-t)\lambda(\lambda t)e^{\lambda t} - (\lambda t)e^{\lambda t} + (\lambda t)[(\lambda t)e^{\lambda t} + e^{\lambda t}]]$$

$$= e^{\lambda - \lambda t}[((1-t)\lambda)^2 + 2(1-t)\lambda^2 t + (\lambda t)^2]$$

$$= e^{\lambda(1-t)}[(1-t)^2\lambda^2 + 2\lambda^2 t]$$

$$= e^{\lambda(1-t)}[1 - \lambda^2 t^2]$$

We get the following weighted poisson process

$$P(N^W(t)) = \frac{[((1-t)\lambda)^2 + 2k(1-t)\lambda - k + k^2]}{[1 - (\lambda t)^2]}(\frac{e^{-\lambda t}(\lambda t)^k}{k!})$$

5 Dispersion Derivation

5.1 Underdispersed Model

We will first derive the derivation of the falling factorial weighted Poisson model

$$P(N^w = n) = \frac{w(n)p(n)}{\sum_{n=0}^{\infty} w(n)p(n)}$$

(16)

First, consider

$$w(n)p(n) = \frac{e^{-\lambda}\lambda^n}{n!}(n_l) = \frac{e^{\lambda}\lambda^n}{n!}(n(n-1)\cdots(n-(l-1)) \tag{17}$$

Now we can take the expected value of w(n) in terms of p(n)

$$\begin{aligned}
E[w(N)] &= \sum_{n=0}^{\infty} \frac{e^{-\lambda}\lambda^n}{n!}(n(n-1)\cdots(n-(l-1)) \\
&= \sum_{n=l}^{\infty} \frac{e^{-\lambda}\lambda^n}{n!}(n(n-1)\cdots(n-(l-1)) \\
&= \sum_{n=l}^{\infty} \frac{e^{-\lambda}\lambda^n}{(n-l)!} \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda}\lambda^{n+l}}{(n)!} \\
&= \sum_{n=0}^{\infty} \frac{e^{-\lambda}\lambda^{n+l}}{(n)!} \\
&= e^{-\lambda}(\lambda^l(e^{\lambda})) \\
&= \lambda^l
\end{aligned}$$

From here we can derive the weighted probability mass function

$$\begin{aligned}
P(N^w = n) &= \frac{\frac{e^{-\lambda}\lambda^n}{n!}(n(n-1)\cdots(n-(l-1))}{\lambda^l} \\
&= \frac{e^{-\lambda}\lambda^{n-l}}{(n-l)!}
\end{aligned}$$

We end with a typical Poisson process but shifted up by l.
Using proposition 1.3 we can actually determine the dispersion of the weighted

Poisson Process using the expected value and the variance of the weighted Poisson Distribution. Consider:

$$E[N^w] = \sum_{n=l}^{\infty} \frac{ne^{-\lambda}\lambda^{n-l}}{(n-l)!}$$

$$= e^{-\lambda} \sum_{n=0}^{\infty} \frac{(n+l)\lambda^n}{(n)!}$$

$$= e^{-\lambda}[\sum_{n=0}^{\infty} \frac{n\lambda^n}{(n)!} + le^{\lambda}]$$

$$= e^{-\lambda}[\lambda \sum_{n=0}^{\infty} \frac{\lambda^n}{(n)!} + le^{\lambda}]$$

$$= e^{-\lambda}[\lambda e^{\lambda} + le^{\lambda}]$$

$$= \lambda + l$$

Since we just proved the proposed distribution is just the classical Poisson distribution but shifted up by l. This means that the variance is consistent with the classic Poisson Distribution:

$$Var(N^w) = \lambda \tag{18}$$

The resulting fisher index of dispersion is:

$$\frac{Var(N^w)}{E(N^w)} = \frac{\lambda}{\lambda + l} \tag{19}$$

So the resulting distribution is underdispersed, meaning that the corresponding weighted Poisson Process is underdispersed. In fact, this dispersion decreases monotonically as l increases.

We can then derive explicitly the fisher index of dispersion of the process using proposition 1.3

$$\frac{Var(N^w(t))}{E(N^w(t))} = 1 + t(\frac{-l}{\lambda + l}) \tag{20}$$

for tϵ(0,1] We have therfore proven this model is indeed underdispersed

5.2 Overdispersed Model

Now we will consider a different weight function with the goal of creating an overdispersed model. Let's first consider

$$w(n) = \frac{1}{n+1} \tag{21}$$

We can derive the expected value of the weight function

$$E(w(n)) = \sum_{n=0}^{\infty} \frac{1}{n+1} \frac{e^{-\lambda}\lambda^n}{n!}$$

$$= \sum_{n=1}^{\infty} \frac{e^{-\lambda}\lambda^n}{(n)!}$$

$$= \frac{e^{-\lambda}}{\lambda}(e^{\lambda} - 1)$$

We can now derive the Probability mass function

$$\frac{w(n)p(n)}{E(w(N))}$$

$$= \frac{1}{e^{\lambda} - 1}(\frac{\lambda^{n+1}}{(n+1)!})$$

We, therefore, have a Poisson Distribution that is shifted down one, and we have a modification to the $e^{-\lambda}$ term. Lets now derive the expected value

$$E[N^w] = \sum_{n=0}^{\infty} \frac{n}{e^{\lambda} - 1}(\frac{\lambda^{n+1}}{(n+1)!})$$

$$= \sum_{n=1}^{\infty} \frac{n-1}{e^{\lambda} - 1}(\frac{\lambda^n}{(n)!})$$

$$= \frac{1}{e^{\lambda} - 1}[\sum_{n=1}^{\infty} \frac{\lambda^n n}{n!} - \sum_{n=1}^{\infty} \frac{\lambda^n}{n!}]$$

$$= \frac{1}{e^{\lambda} - 1}[\lambda e^{\lambda} - (e^{\lambda} - 1)]$$

$$= \frac{\lambda e^{\lambda} - e^{\lambda} + 1}{e^{\lambda} - 1}$$

One can now derive the variance of the distribution:

$$Var(N^w) = E((N^w)^2) - (E[N^w])^2$$

$$E((N^w)^2) = \frac{1}{e^{\lambda} - 1} \sum_{n=0}^{\infty} \frac{n^2\lambda^{n+1}}{(n+1)!}$$

$$= \frac{1}{e^{\lambda} - 1} \sum_{n=1}^{\infty} \frac{(n-1)^2\lambda^n}{(n)!}$$

$$= \frac{1}{e^{\lambda} - 1} \sum_{n=1}^{\infty} \frac{(n^2 - 2n + 1)\lambda^n}{(n)!}$$

$$= \frac{1}{e^\lambda - 1}[\sum_{n=1}^{\infty} \frac{(n^2)\lambda^n}{(n)!} - 2\sum_{n=1}^{\infty} \frac{(n)\lambda^n}{(n)!} + \sum_{n=1}^{\infty} \frac{\lambda^n}{(n)!}]$$

$$= \frac{1}{e^\lambda - 1}[\sum_{n=1}^{\infty} \frac{(n)\lambda^n}{(n-1)!} - 2\sum_{n=1}^{\infty} \frac{(n)\lambda^n}{(n)!} + (e^\lambda - 1)]$$

$$= \frac{1}{e^\lambda - 1}[\lambda(\sum_{n=0}^{\infty} \frac{(n+1)\lambda^n}{(n)!}) - 2\sum_{n=0}^{\infty} \frac{\lambda^{n+1}}{(n)!} + (e^\lambda - 1)]$$

$$= \frac{1}{e^\lambda - 1}[\lambda(\sum_{n=0}^{\infty} \frac{n\lambda^n}{(n)!} + \sum_{n=0}^{\infty} \frac{(\lambda^n}{(n)!}) - 2\sum_{n=0}^{\infty} \frac{\lambda^{n+1}}{(n)!} + (e^\lambda - 1)]$$

$$= \frac{1}{e^\lambda - 1}[\lambda^2 e^\lambda - \lambda e^\lambda + e^\lambda - 1]$$

From here we can calculate the variance

$$Var(N^w) = \frac{1}{e^\lambda - 1}[\lambda^2 e^\lambda - \lambda e^\lambda + e^\lambda - 1] - \frac{\lambda^2 e^{2\lambda} - 2\lambda e^{2\lambda} + 2\lambda e^\lambda + e^{2\lambda} - 2e^\lambda + 1}{(e^\lambda - 1)^2}$$

$$= \frac{(e^\lambda - 1)[\lambda^2 e^\lambda - \lambda e^\lambda + e^\lambda - 1] - [\lambda^2 e^{2\lambda} - 2\lambda e^{2\lambda} + 2\lambda e^\lambda + e^{2\lambda} - 2e^\lambda + 1]}{(e^\lambda - 1)^2}$$

$$= \frac{\lambda e^\lambda (e^\lambda - \lambda - 1))}{(e^\lambda - 1)^2}$$

Now we can calculate the fisher index of dispersion

$$\frac{Var(N^w)}{E(N^w)} = \frac{(\frac{\lambda e^\lambda - e^\lambda + 1}{e^\lambda - 1})}{(\frac{\lambda e^\lambda (e^\lambda - \lambda - 1))}{(e^\lambda - 1)^2})}$$

$$= \frac{\lambda e^\lambda (e^\lambda - \lambda - 1)}{\lambda e^{2\lambda} - e^{2\lambda} + e^\lambda - \lambda e^\lambda + e^\lambda - 1}$$

$$= \frac{\lambda e^{2\lambda} - e^\lambda (\lambda)(\lambda - 1)}{e^{2\lambda}(\lambda - 1) + e^\lambda (2 - \lambda) - 1}$$

We know this equation overdispersed if the following holds

$$\lambda e^{2\lambda} - e^\lambda (\lambda)(\lambda - 1) > e^{2\lambda}(\lambda - 1) + e^\lambda (2 - \lambda) - 1$$

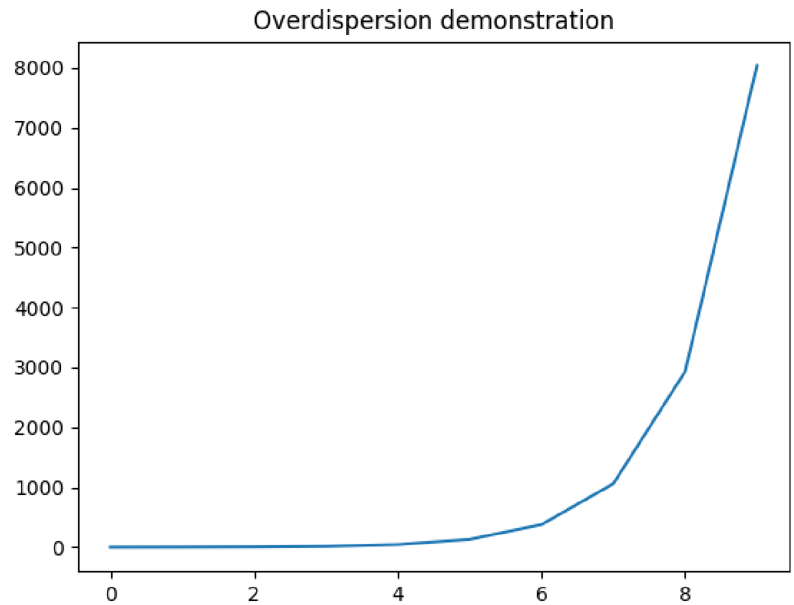
We can rearrange this inequality as:

$$-e^\lambda \lambda(\lambda - 1) > -e^{2\lambda} + 2e^\lambda - \lambda e^\lambda - 1$$

$$-\lambda^2 + \lambda > -e^\lambda + 2 - \lambda - \frac{1}{e^\lambda}$$

$$e^\lambda - \lambda^2 + 2\lambda - 2 + \frac{1}{e^\lambda} > 0$$

We can now graph this function using python



We see this function is monotonically increasing to infinity as lambda increases but the main takeaway is that the inequality always holds for $\lambda >0$.
So the proposed Weighted Poisson process is indeed overdispersed

$$\frac{Var(N^w)}{E(N^w)} = \frac{\lambda e^{2\lambda} - e^{\lambda}(\lambda)(\lambda-1)}{e^{2\lambda}(\lambda-1)+e^{\lambda}(2-\lambda)-1} > 1 \tag{22}$$

We can derive the explicit dispersion in terms of t

$$\frac{Var(N^w(t))}{E(N^w(t))} = 1 + t[\frac{\lambda e^{2\lambda} - e^{\lambda}(\lambda)(\lambda-1)}{e^{2\lambda}(\lambda-1)+e^{\lambda}(2-\lambda)-1} - 1] \tag{23}$$

5.2.1 The general inverse falling factorial starting at (n+1)

We can even derive the fisher index of dispersion in general for

$$w(n) = \frac{1}{(n+1)\cdots(n+l)} \tag{24}$$

Calculating the expected value and variance of the weighted Poisson model for this weight function will be very tedious and cumbersome in terms of computation. However, there are several useful theorems we can invoke to make this easier

Theorem 1 (logconvavity and logconvexity theorem) *let N be a Poisson random variable with $\lambda > 0$ and w(n) be a weight function not depending on λ. Then the Expected value of the weight function is logconvex (logconcave) if and only if the weighted version N^w is overdispersed (underdispersed)*

Given the conditions we have specified we can also use the following corollary

Corollary 1.1 *If w(n) is logconvex (logconcave) then its expected value is also logconvex (logconcave)*

Claim:N^w is overdispersed with

$$w(n) = \frac{1}{(n+1)\cdots(n+l)} \tag{25}$$

as its weight function.

5.2.2 Proof

Proof this by proving w(n) is logconvex

$$log(w(n)) = log(\frac{1}{(n+1)\cdots(n+l)})$$

$$\frac{d}{dn} log(w(n))$$

$$= \frac{d}{dn} log(\frac{1}{(n+1)} \cdots \frac{1}{(n+l)})$$

$$= \frac{d}{dn}[log(\frac{1}{n+1}) + \cdots + log(\frac{1}{n+l})]$$

$$= \frac{-(n+1)}{(n+1)^2} + \cdots + \frac{-(n+l)}{(n+l)^2}$$

$$= \frac{-1}{n+1} - \cdots - \frac{-1}{n+l}$$

$$\frac{d^2}{dn^2} log(w(n))$$

$$= (n+1)^{-2} + \cdots + (n+l)^{-2}$$

Thus $\frac{d^2}{dn^2} log(w(n)) > 0 \ \forall n \epsilon N$ proving its logconvex, this implies the mean weight function is also logconvex. Therefore, N^w is overdispersed

6 Simulation Study

First, we should identify an effective way to simulate Poisson Process Data. We are dealing with a PMF with independent events. We will us the 'Binning Method' to simulate the Overdispersed Weighted Poisson Process. The simulation will be done in python.

6.1 Description of Simulation Method

We first want to derive the PMf of the proposed process. The underdispersed and classic Poisson Models can be simulated using built-in generators (numpy). Luckily we have already derived this from the dispersion derivation. After we find this PMF f(x), we will create 100 instances of f(x) at x=0,1,2,....,100. This gives us probability values for x arrivals. Since we are dealing with independent, discrete random variables we can find the CDF of each x by adding the previous occurrences. $F(x) = f(x) + f(x-1) + \cdots + f(0)$. We then get a probability array that looks like [F(0), F(1),....,F(100)]. We then produce a uniform random variable from (0,1), we get probability p from this. We take p and match it to the CDF array such that

$$max\{F(x)|p \leq F(x)\} \tag{26}$$

We take the x value from this F(x) and make that our number of occurrences. We repeat this process in accordance with how many iterations we specified.

The general function we use for this process is as follows:

```
#Simulating Weighted Poisson Process
import random as rd
import math as mt
import matplotlib.pyplot as plt
import statistics as stat
import numpy as np

def over_poisson(lamb , iterations ,f):
    #Create an empty list to store the various cdf values and the number of arrivals
    arrivals =[]
    cdf_set = []
    fisher_dispersion = []
    mean_arrival = []
    x = 0
    #We need 100 instances of the pdf, then we increment each time
    for i in range(0 , 100):
        y = f(i , lamb)
        x += y
        cdf_set.append(x)

    #Create a uniform random variable outputs values from 0 to 1np.random.seed()
    np.random.seed(300)
    prob = np.random.uniform(0 , 1 , iterations)
    for i in range(iterations):
        #Output random probability from 0 to 1s

        prob_value = prob[i]
        #Match this probability the accumlated CDF list
```

```
        #Brute force way of doing it
        for j in range(len(cdf_set)):
            if prob_value < cdf_set[j]:
                arrivals.append(j)
                break
    #Calculate dispersion over time
    for i in range(len(arrivals)):
        if stat.mean(arrivals[0:i+1]) == 0 or i == 0 :
            fisher_dispersion.append(0)
        else:
            fisher_dispersion.append(stat.variance(arrivals[0:i+1])
            /stat.mean(arrivals[0:i+1]))

    for i in range(len(arrivals)):
        mean_arrival.append(stat.mean(arrivals[0:i+1]))

    return arrivals , fisher_dispersion ,mean_arrival
```

For simulating the classic and the falling factorial weighted Poisson Processes the code is as follows:

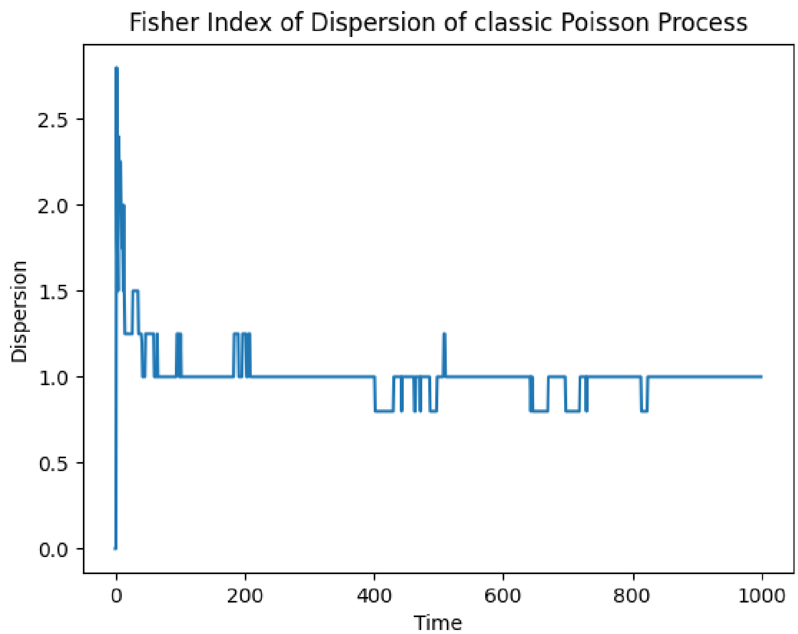
```
    #Use numpy to simulate a regular poisson process at lambda=5
rng = np.random.default_rng(300)
sim_poisson = rng.poisson(lam=5 , size= 1000)
print(sim_poisson)

#Calculate how the mean and variancechanges overtime
def fisher_calculator(sim):
    val_list=[]
    mean_arrival = []
    var_arrival = []
    fisher_index_classic =[]
    for i in range(len(sim)):
        val_list.append(sim[i])
        mean_arrival.append(stat.mean(val_list))
        if i == 0:
            var_arrival.append(0)
        else:
            var_arrival.append(stat.variance(val_list))
        fisher_index_classic.append(var_arrival[i]/mean_arrival[i])
    return mean_arrival , fisher_index_classic


fisher_poisson = fisher_calculator(sim_poisson)
#plot the mean
plt.plot(fisher_poisson[0])
```

```
plt.title("Mean plot of classic Poisson Process")
plt.xlabel("Time")
plt.ylabel("Mean")
plt.show()

#plot the fisher index
plt.plot(fisher_poisson[1])
plt.title("Fisher Index of Dispersion of classic Poisson Process")
plt.xlabel("Time")
plt.ylabel("Dispersion")
plt.show()

#To simulate the underdispersed model we simply shift our poisson out put up by l
def under_dispersed(list,l):
    under_disp_data = list + l
    return under_disp_data

under_sim = under_dispersed(sim_poisson , l=1)

#Use the fisher_calculator function to get the dispersion plot
under_fisher = fisher_calculator(under_sim)
plt.plot(under_fisher[1])
plt.title("Fisher Index of Dispersion of Falling Factorial Poisson Process ")
plt.xlabel("Time")
plt.ylabel("Dispersion")
plt.show()
#We observe slight under dispersion.  We can run this experiment with l=2
under_sim_2= under_dispersed(sim_poisson , l=2)
under_fisher_2 = fisher_calculator(under_sim_2)

plt.plot(under_fisher_2[1])
plt.title("Fisher Index of Dispersion of Falling Factorial Poisson Process at l=2")
plt.xlabel("Time")
plt.ylabel("Dispersion")
plt.show()

#We can also do the same for l=5
under_sim_5= under_dispersed(sim_poisson , l=5)
under_fisher_5 = fisher_calculator(under_sim_5)

plt.plot(under_fisher_5[1])
plt.title("Fisher Index of Dispersion of Falling Factorial Poisson Process at l=5")
plt.xlabel("Time")
plt.ylabel("Dispersion")
plt.show()
```

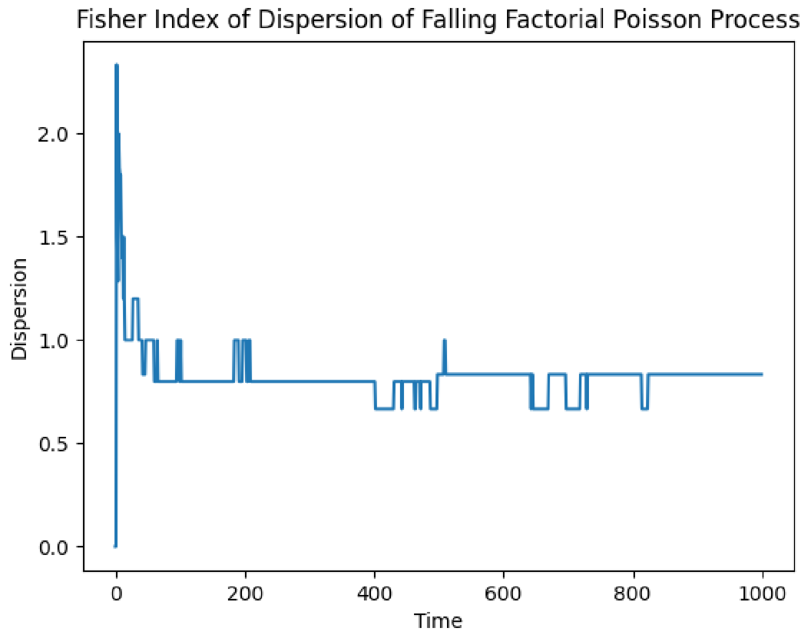
6.2 Result

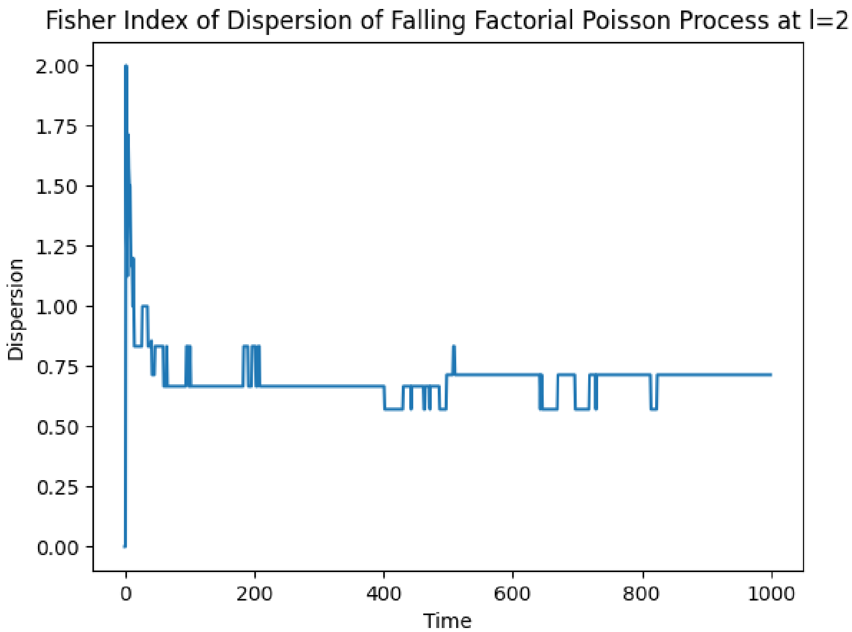
We set the following parameters: $\lambda = 5$ with 1000 iterations

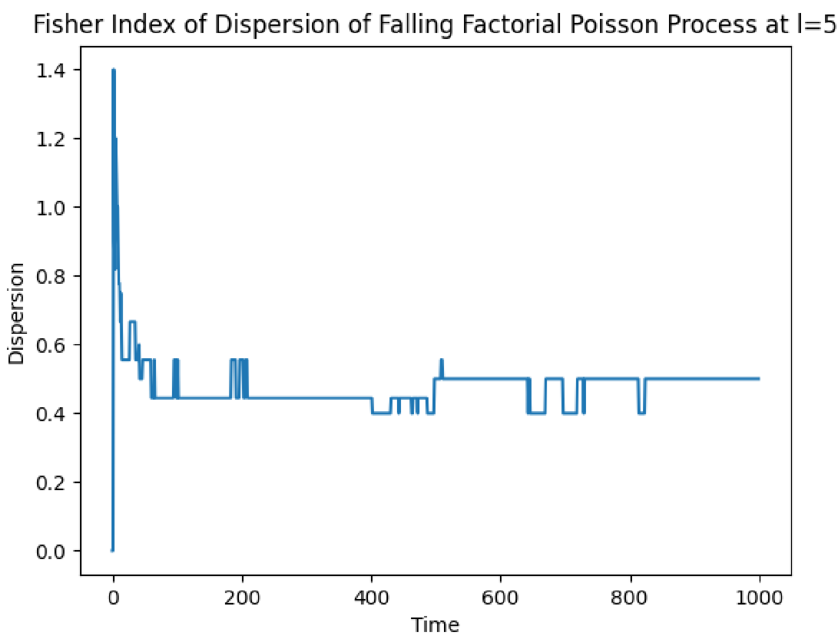
6.2.1 Traditional Poisson Dispersion plot



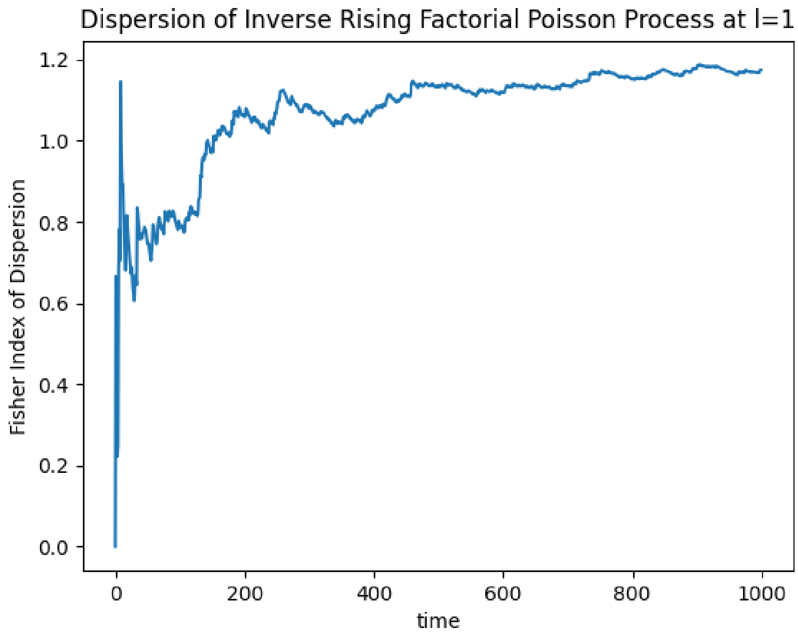
6.2.2 Falling Factorial Poisson Dispersion Plots

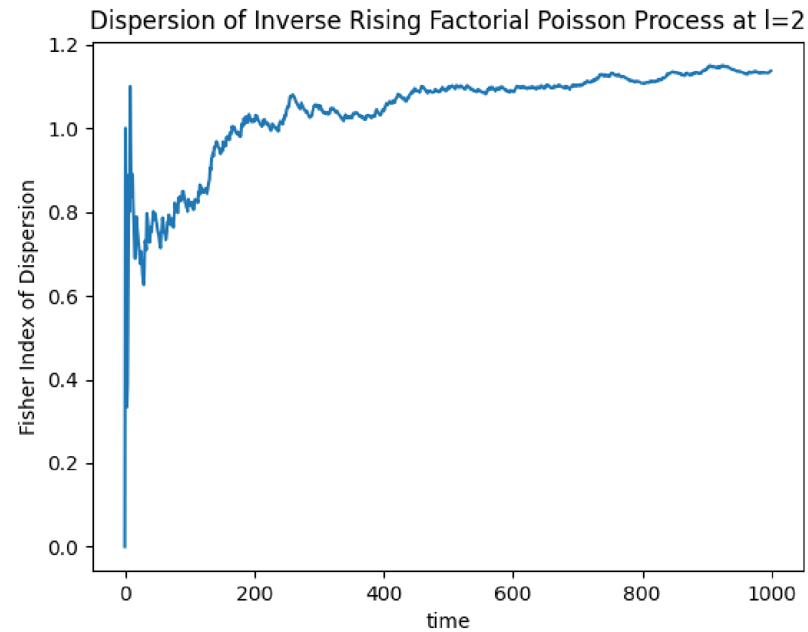


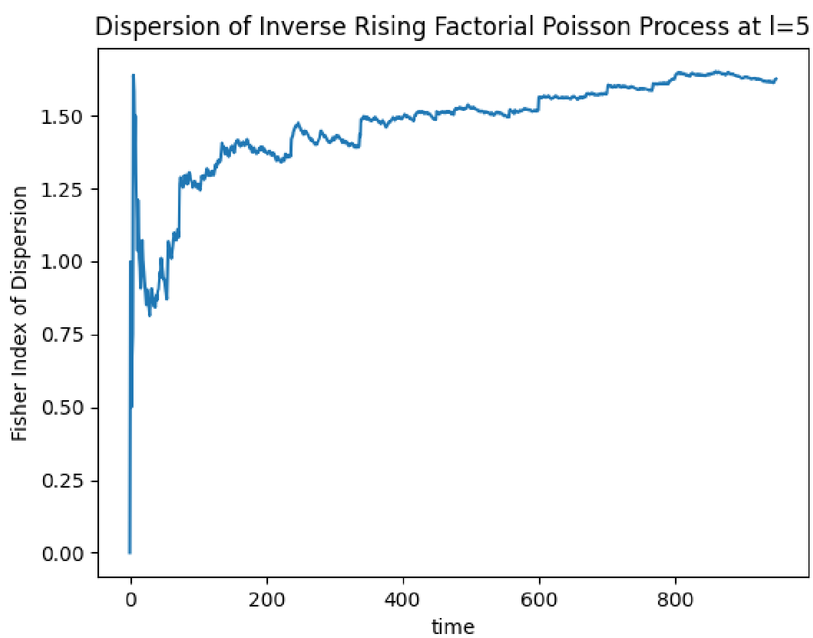

Fisher Index of Dispersion of Falling Factorial Poisson Process at l=2
Dispersion
Time
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
200
400
600
800
1000


Fisher Index of Dispersion of Falling Factorial Poisson Process at l=5
Dispersion
Time
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000

6.2.3 Inverse Rising Factorial Poisson Dispersion Plots




Dispersion of Inverse Rising Factorial Poisson Process at l=2
Fisher Index of Dispersion
time
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
200
400
600
800
1000


Dispersion of Inverse Rising Factorial Poisson Process at l=5
Fisher Index of Dispersion
time
0.00
0.25
0.50
0.75
1.00
1.25
1.50
0
200
400
600
800

6.3 Observations

We see that the dispersion plots line up with our hypothesis. One can see that the classic Poisson Distribution converges to equidispersion as time increases. The Falling factorial model clearly demonstrated underdispersion. We observe that as we increase our l value the model decreases in dispersion, which lines up exactly with the fisher index of dispersion formula we came up with previously. We can observe that the inverse rising factorial weighted poisson process demonstrated overdispersion. We did not see a dramatic change from l=1 to l=2, however, the difference becomes more clear when we compare the l=1,2 and l=5 plots.

7 Conclusion

The implications of such models are that we can create a modified Poisson process depending on the dispersion of our empirical data. We maintain the simplicity and familiarity of the Poisson process, with the flexibility to change our model depending on the type of dispersion we observe. We have presented a clear methodology in place for when we encounter under or over dispersed data. We can increase the l values of the falling factorial depending on how underdispersed the data is. The same can be said for the rising factorial in regard to overdispersed data.

8 Bibliography

1.Rao, C.R. (1965) ON Discrete Distributions Arising out of Methods of Ascertainment. Sankhya; The Ind.J. Statist, Series A, 27, 311-324.
2.Castillo, J., Perez-Casany, M., 1998. Weighted Poisson distributions for overdispersion and underdispersion situations. Ann. Inst. Statist. Math. 50 (3), 567–585
3.Balakrishnan, N., amp; Kozubowski, T. J. (n.d.). A class of weighted Poisson processes. Statistics and Probability Letters 78 (2008) 2346–2352
4.Balakrishnan Poisson UDOD powerpoint