The Poisson Tree

Bill Huang
Oberlin College
chuang@oberlin.edu

Abstract

Inspired by Tyler Hobbss artworks on probability distribution [1], in this paper I will discuss how to generate two dimensional Poisson point process and construct minimal spanning tree from the generated points. Then, I extend the algorithm by incorporating the Monte Carlo integration to generate the "Heart tree" and the "Batman tree".

Introduction

The Poisson process is a widely used for counting the occurrences of a certain event happening in a certain rate but completely random in timings. For example, imagine that you are the owner of a grocery store and you observe that customers visit the store with a rate of 15 people per hour, but you do not know what time in the hour the customer actually visits. The Poisson process would be a good model for that. Before giving the definition for Poisson process, I need to introduce the idea of counting process. Again, imagine you are the owners of a grocery store and you record the time every time a consumer comes in. Let $X(t)$ be the number of customers comes in up to and including time t. Then, $\{X(t) : t \in [0, \infty)\}$ is a stochastic process with state space $\mathcal{X} = \{0, 1, \ldots\}$, and this is a counting process. A Poisson process is a counting process with rate λ that satisfy the following conditions:

- $X(0) = 0$, which no events happen at time 0.
- $X(t)$ has independent and stationary increments.
- $\mathbb{P}(X(t+h) - X(t) = 1) = \lambda h + o(h)$ and $\mathbb{P}(X(t+h) - X(t) \geq 2) = o(h)$ [1], which the probability of an event in $[t, t+h]$ is approximately λh while the probability of more than one events is very small.

The above conditions imply that $X(s+t) - X(s) \sim \text{Poisson}(\lambda t)$, meaning that the number of events in any interval of length $t > 0$ follows a Poisson distribution with parameter λt. Let W_n denote the time at which the n^{th} event happens and set $W_0 = 0$. Now let $S_n = W_{n+1} - W_n$, where S_n is the length of the time between the n^{th} and $(n+1)^{th}$ events. One important property about $S_0, S_1, \ldots$ is that they follow a exponential distribution with mean $1/\lambda$ [4]. With this property, I can generate one dimensional homogeneous [2] Poisson point process (Figure 1). The horizontal direction is time, and each vertical line represents an event. The distance between any two consecutive lines indicates the time between any two consecutive events, which follows an exponential distribution. In the next section, I will present the different algorithm of generating Poisson point process in two dimensional space and construct minimal spanning tree. In the next section, I include Monte Carlo integration to create the "Heart Tree" and the "Batman Tree". Last, I draw conclusion and discuss future direction.

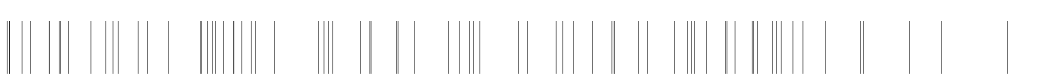

Figure 1 : *One Dimensional Homogeneous Poisson Process with* $t \in [0, 200]$ *and* $\lambda = 0.2$

[1]If $f(h)/h \to 0$ as $h \to 0$, then we write $f(h) = o(h)$, meaning that $f(h)$ approaches to 0 faster than h does. E.g. $h^3 = o(h)$.
[2]When λ is a constant, the Poisson process is homogeneous. Otherwise, it is a non-homogeneous Poisson process.

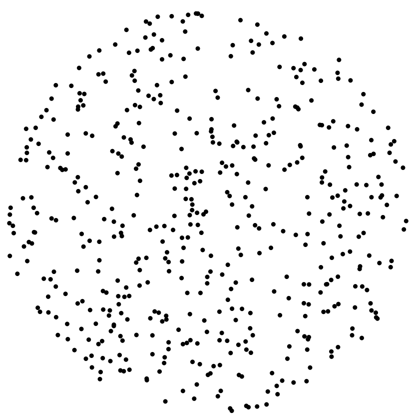

Figure 2: *Two Dimensional Homogeneous Poisson Point Process with $n = 500$ and $\lambda = 0.02$*

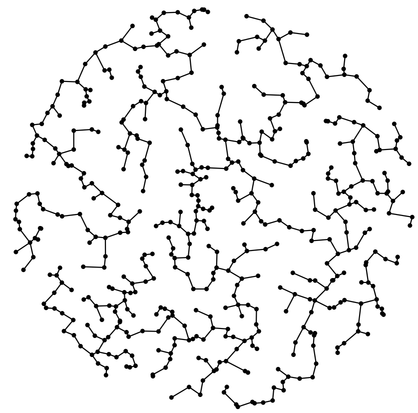

Figure 3: *Minimal Spanning Tree Generated from Points in Figure 2*

Two Dimensional Poisson Point Process

In generating one-dimensional Poisson point process, the key is that the distance between any two consecutive lines (points) follows a exponential distribution with mean $1/\lambda$. Then, the key for generating two-dimensional Poisson point process is that the area between two consecutive points follows a exponential distribution, but how can we identify the area between two points in two dimensional space?

First, I introduce the method of generating two dimensional Poisson point process in polar coordinate. For two points in polar coordinate (r_1, θ_1) and (r_2, θ_2) where $r_2 > r_1$, I will identify the area between these two points as the area of the region bounded by the circle with radius r_1 and circle with radius r_2, that is $\pi(r_2^2 - r_1^2)$. Then, the steps of generating two dimensional Poisson point process are the following:

- Pick λ, the rate of Poisson process, and n, the number of points to generate.
- draw a number from Exponential(λ) as A_1, the area between the first point and the origin.
- Calculate $r_1 = \sqrt{A_1/2\pi}$.
- draw a number from Uniform$(0, 360)$ as θ_1 to get the coordinate of the first point (r_1, θ_1).
- draw another number from Exponential(λ) as A_2, the area between the second point and the first point.
- Calculate $r_2 = \sqrt{(A_1 + A_2)/2\pi}$.
- draw a number from Uniform$(0, 360)$ as θ_2 to get the coordinate of the first point (r_2, θ_2).
- Repeat the process until we get the coordinate for all n points.

This produce Figure 2. Having points randomly placed over the plane does not produce much aesthetic value, so I will use the points in Figure 2 as vertexes to construct a minimal spanning tree in the following way:

- Every two vertexes form a edge and calculate the length of all the edges.
- Find the shortest edge and connect the two vertexes on that edge.
- Find the next shortest edge and connect the two vertexes on that edge.
- Repeat the above step. If connecting two vertexes forms a cycle, do not connect and find the next shortest edge until all vertexes are connected.

The above steps generate Figure 3. Using the same method, we can also generate non-homogeneous two-dimensional Poisson point process with $n = 500$ in Figure 4 and the corresponding minimal spanning tree

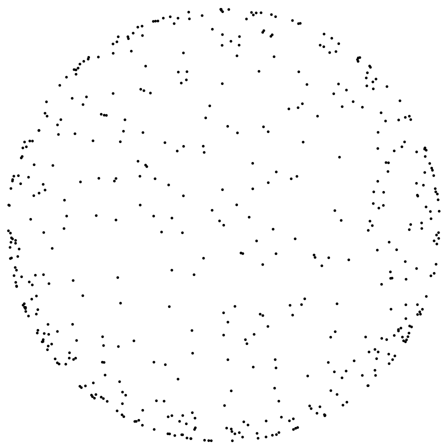

Figure 4 : *Two Dimensional Non-Homogeneous Poisson Point Process*

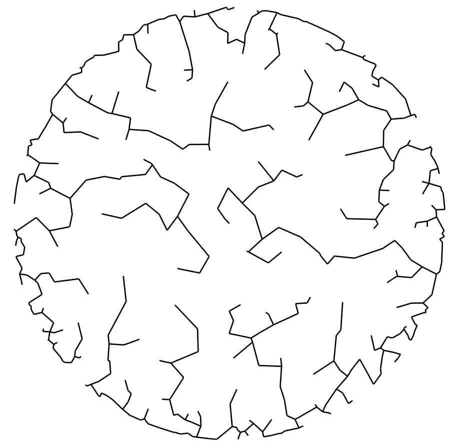

Figure 5 : *Minimal Spanning Tree Generated from Points in Figure 4*

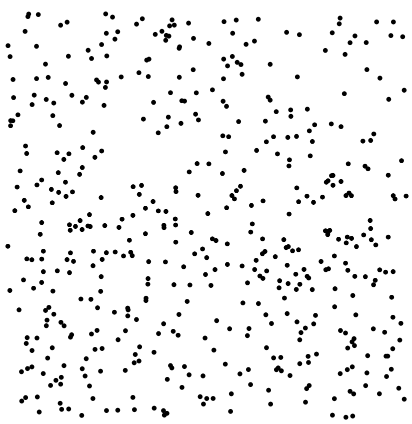

Figure 6 : *Two Dimensional Homogeneous Poisson Point Process with* $n = 500$ *and* $\lambda = 0.008$

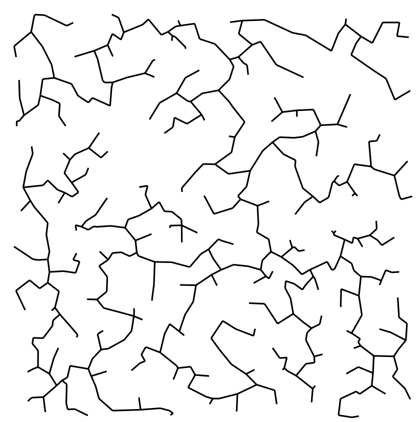

Figure 7 : *Minimal Spanning Tree Generated from Points in Figure 6*

in Figure 5. Here, the rate $\lambda = 500/(y-500)^2$ where y is the number of points that are already drew, so λ is not a constant. It is a non-homogeneous Poisson point process.

After discussing the method used in polar coordinate system, now I will talk about the method used in Cartesian coordinate system, and we only focus on the first quadrant. Suppose we want to generate two dimensional Poisson point process over the square $[0, 250] \times [0, 250]$. For two points (x_1, y_1) and (x_2, y_2), I will define the area between the two points as the rectangle with vertexes $(x_1, 0)$, $(x_1, 250)$, $(x_2, 250)$ and $(x_2, 0)$, which has area of $250|x_1 - x_2|$. Then, the algorithm is similar to the one for the polar coordinate but different in the following way:

- λ is not randomly chosen. It is determined by $\lambda = 1/A$ where $A = 250 \times 250/n$, the average of all the areas between any two consecutive points, for example, point (x_{i-1}, y_{i-1}) and point (x_i, y_i)
- Draw a number from Exponential(λ) as A_i, but the coordinate is determined by $x_i = x_{i-1} + \frac{A_i}{250}$ and $y_i \sim$ Uniform$(0, 250)$.

The above algorithm generates Figure 6, and Figure 7 shows the minimal spanning tree constructed from those points. With the algorithm, we can also generate Poisson point process over any rectangle.

Figure 8 : *Homogeneous Poisson Point Process Under* $f(x) = x^{0.75} + 50$

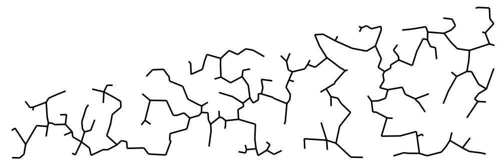

Figure 9 : *Minimal Spanning Tree Generated from Points in Figure 8*

Poisson Point Process Under the Curve

Having discussed the method of generating Poisson point process over the region of any rectangle, now we can extend the algorithm further to generate it over the area under any curve in the first quadrant of the Cartesian coordinate system. Let $f(x)$ be the curve. For any two pints (x_1, y_1) and (x_2, y_2) under curve where $x_1 < x_2$, $y_1 < f(x_1)$ and $y_2 < f(x_2)$, I will define the area between these two point as the area under the curve between $x = x_1$ and $x = x_2$, that is $\int_{x_1}^{x_2} f(x)\,dx$. Suppose that we want to generate Poisson point process over the region under the curve $f(x)$ between $x = a$ and $x = b$ where $a < b$ in the first quadrant, the algorithm is the following:

- $\lambda = 1/A$ where $A = (\int_a^b f(x)\,dx)/n$, the average of all the areas between any two consecutive points
- Draw a number from exponential(λ) as A_1. The coordinate of the first point is determined by x_1 where $\int_a^{x_1} f(x)\,dx = A_1$ and $y_1 \sim$ Uniform$(0, f(x_1))$
- Draw a number from exponential(λ) as A_i. The coordinate of the i^{th} point is determined by x_i where $\int_{x_{i-1}}^{x_i} f(x)\,dx = A_i$ and $y_i \sim$ Uniform$(0, f(x_i))$

If $f(x)$ is complicated, then we cannot determine the x-coordinate of the points easily, but there is a way to deal with this problem: Monte Carlo Integration. The idea is to write the integral as the following:

$$I = \int_a^b f(x)\,dx = \int_a^b w(x)h(x)\,dx$$

where $w(x) = f(x)(b - a)$ and $h(x) = 1/(b - a)$. Notice that h is the probability density function for Uniform(a, b), so the above integration is equivalent to $I = \mathbb{E}_h[w(X)]$ where $X \sim$ Uniform(a, b). If we generate $X_1, \dots, X_k \sim$ Uniform(a, b), then by the weak law of large numbers

$$\widehat{I} = \frac{1}{k}\sum_{i=1}^{k} w(X_i) \xrightarrow{P} \mathbb{E}[w(X)] = I \quad [4]$$

Then, the steps for approximating x_1 where $\int_a^{x_1} f(x)\,dx = A_1$ by Monte Carlo integration are

- Set the increment step as d, a small number, and let $x_1 = a + d$
- Calculate $B = \int_a^{x_1} f(x)\,dx$ by Monte Carlo integration. If $B < A$, increase x_1 by d, then calculate B again. Keep increasing x_1 by d until we have a x_1 that makes $B \geq A$.

After figuring out x_1, we can use the same method to approximate x_2, then $x_3, \dots, x_n$. Though the Monte Carlo integration cannot give actual value for x_i, but the approximation under $k = 10000$ is precise. Figure 8 is generated by the above algorithm and Figure 9 shows the corresponding minimal spanning tree. In the next section, I will present the ”Heart tree” and the ”Batman tree” generated by this algorithm.

Heart Tree

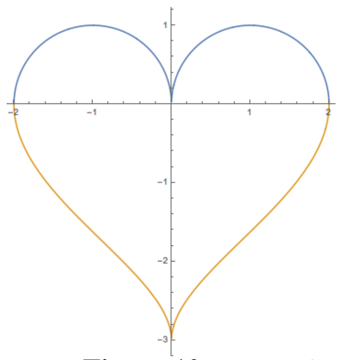

Figure 10 : *Heart Curve*

Before I present the "Heart Tree", I need to introduce the "Heart curve", a combination of two curves that give the heart shape (Figure 10). The function for the upper part is

$$f(x) = \sqrt{1-(|x|-1)^2}, \ -2 \leq x \leq 2 \quad [3]$$

and the function for the lower part is

$$h(x) = -3\sqrt{1-(|x|/2)^{0.5}}, \ -2 \leq x \leq 2 \quad [3]$$

The algorithm described in the last section is limited for the curve in the first quadrant, so we need to transform the function first. From Figure 10, we can see that it is symmetry across the y-axis, so we only need to generate Poisson point process for the right side ($x > 0$). The steps are the following:

- Generate Poisson point process under $f'(x) = \sqrt{1-(x-1)^2}$ for $0 \leq x \leq 2$. Save all the points and replicate them with changing all the x-coordinate to $-x$. Now we have the upper part done.
- Generate Poisson point process under $h'(x) = 3\sqrt{1-(x/2)^{0.5}}$ for $0 \leq x \leq 2$. Save all the points with y-coordinate as $-y$. Then replicate all the points and change x-coordinate to $-x$, and we have the lower part.

With the generated points, we can construct a minimal spanning tree, which is the "Heart Tree" (Figure 11).

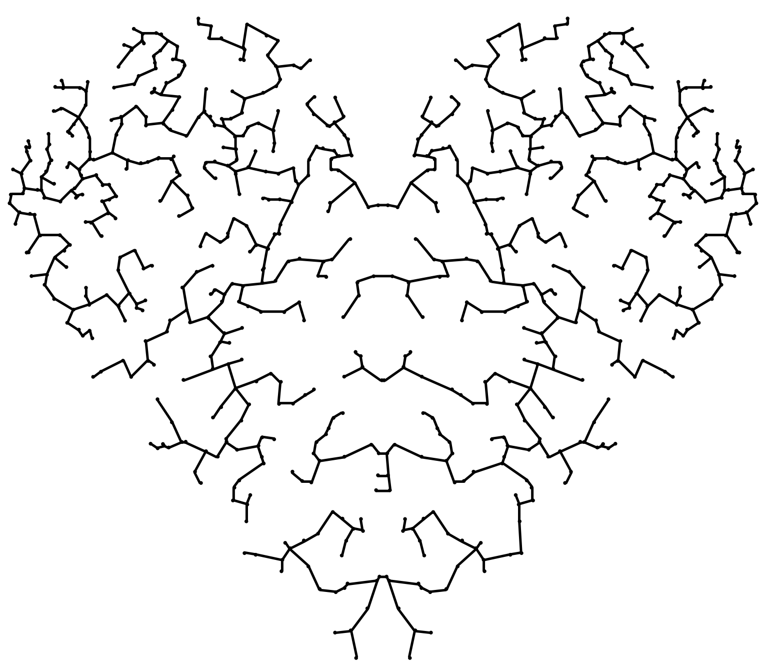

Figure 11 : *Heart Tree*

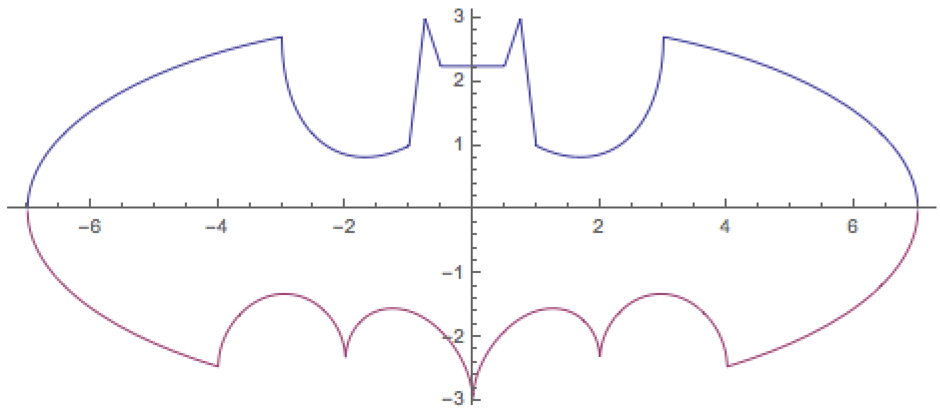

Figure 12 : *Batman Curve*

Batman Tree

Figure 12 shows us the Batman curve. The upper part is constructed by the following function:

$$f(x) = \begin{cases} 2.25 & |x| \leq 0.5 \\ 3|x| + 0.75 & 0.5 < |x| \leq 0.75 \\ -8|x| + 9 & 0.75 < |x| \leq 1 \\ \frac{6\sqrt{10}}{7} + (1.5 - 0.5|x|) - \frac{6\sqrt{10}}{14}\sqrt{4 - (|x| - 1)^2} & 1 < |x| \leq 3 \\ \sqrt{9 - 9(\frac{x}{7})^2} & 3 < |x| \leq 7 \end{cases} \quad [2]$$

and the lower part is constructed by

$$h(x) = \begin{cases} |\frac{x}{2}| - \frac{3\sqrt{33}-7}{112}x^2 - 3 + \sqrt{1 - (||x| - 2| - 1)^2} & |x| \leq 4 \\ -\sqrt{9 - 9(\frac{x}{7})^2} & 4 < |x| \leq 7 \end{cases} \quad [2]$$

Using the same algorithm for the "Heart Tree", I can also generate the "Batman Tree" (Figure 13).

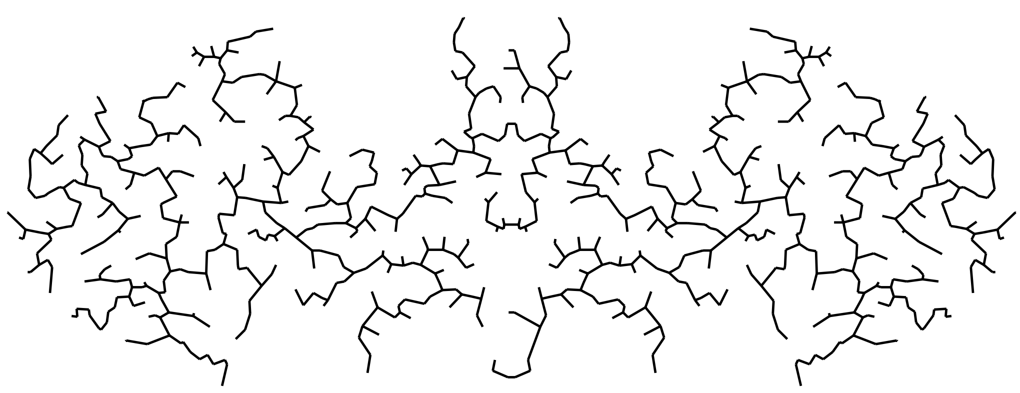

Figure 13 : *Batman Tree*

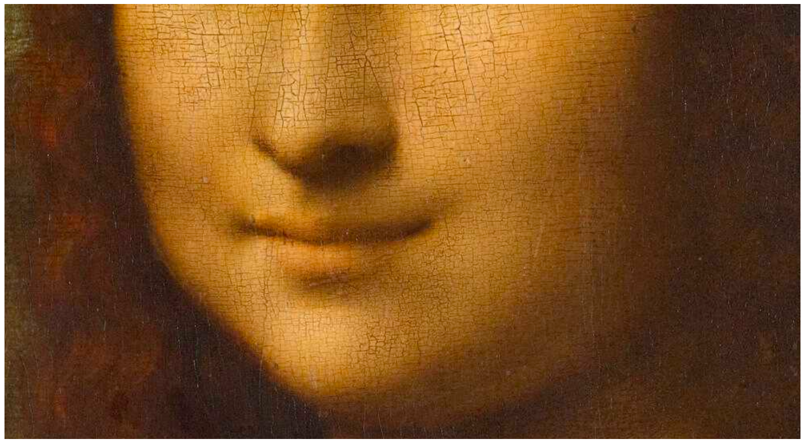

Figure 14 : *Texture on Mona Lisa, adapted from Site officiel du muse du Louvre*

Summary and Conclusion

In the first part of the paper I presented an algorithm of generating two dimensional Poisson point process in various way and construct minimal spanning tree from the generated points. In the second part, I incorporate the Monte Carlo integration method to extend the algorithm for generating Poisson point process under any curve in the first quadrant. Moreover, if we can present an object by some functions, we can always generate Poisson point process over that region and construct minimal spanning tree.

However, the method I use to compute the x-coordinate for the points under a curve by Monte Carlo integration is a naive approach. It takes fair amount of time to compute. Therefore, one future direction is to investigate a more efficient algorithm for computing the x-coordinate. One idea I have is to incorporate the geometrical perspective by constructing trapezoidal and triangle to reduce number of steps the algorithm takes to compute x-coordinate.

On the other hand, the minimal spanning generated from the Poisson point process is similar to the texture in the oil painting, such as the texture on Mona Lisa (Figure 14). Therefore, another future approach will be to develop a way to generate Poisson point process and build minimal spanning tree that add interesting texture on a photograph.

References

[1] Hobbs, Tyler. ”Probability Distributions for Algorithmic Artists.” *Algorithmic Art*. `http://www.tylerlhobbs.com/writings/probability-distributions-for-artists` (as of May. 13, 2016)

[2] “Is This Batman Equation for Real?” *Geometry*. `http://math.stackexchange.com/questions/54506/is-this-batman-equation-for-real` (as of May. 13, 2016).

[3] “One Man’s Funnies.” *Mathematical Equations of Love*. `http://onemanadreaming.blogspot.com/2014/08/mathematical-equation-of-love-heart.html` (as of May. 13, 2016).

[4] Wasserman, Larry. *All of Statistics: A Concise Course in Statistical Inference*. New York: Springer, 2004. 394-404. Print.