Mean Shift Summary

Shuo Sun

07/09/2021

1 Introduction

The Mean Shift(MS) algorithm is a kind of clustering algorithm based on density estimation. Given discretely sampled data, the MS algorithm aims to locate the maxima of the density function. To estimate the density function, one simple approach is to smooth the data, for example:

$$f(x) = \sum_{i=1}^{N} K(\frac{x - x_i}{h}) \tag{1}$$

where function K(x) is called the *kernel function* if there exists a profile such that:

$$K(x) = k(||x||^2) \tag{2}$$

the $k(x)$ should have the property:
1) k in non-negative
2) k is non-increasing: $k(a) \geq k(b)$ if $a < b$
3) k is piece-wise continuous and $\int_0^\infty k(r)d(r) < \infty$

The sample mean with kernel K at $x \in X$ can be written as :

$$m(x) = \frac{\sum_{s \in S'} K(s - x)w(s)s}{\sum_{s \in S'} K(s - x)w(s)} \tag{3}$$

where S' is the a finite set(sample); $w(s)$ is the weight function.

Let $T \subset X$ is a finite set, and we use the MS algorithm iteratively $T \leftarrow m(T)$ until converging. If the set $T = S$, we call it a blurring process.

Based on the above information, we can simply summarise the process of the Mean Shift algorithm:

```
Algorithm 1: Mean Shift
Input: sample set: $S'$, point set: $T$, bandwidth: $h$, threshold: $\epsilon$
Result: shifting_points set: $T'$
i = 0
for $p_i$ in $T'$ do
    $p_{shift} \leftarrow p_i$, $d \leftarrow \epsilon + \delta$, $\delta > 0$
    while $d \leq \epsilon$ do
        $p_{old} \leftarrow p_{shift}$
        $p_{new} \leftarrow$ GetNewPoint($p_{old}, T$)
        $d \leftarrow d(p_{old}, p_{new})$
        $p_{shift} \leftarrow p_{new}$
    end
    $T'[i] = p_{shift}$
    i += 1
end
```

In this task, I give two different ways for the Mean-Shift algorithm to process circular-linear data. The difference between the two ways is how to compute and evaluate the distance between two points with (θ, r) format. The first one (Algo1) converts the point in polar coordinate to Cartesian coordinate. The second one (Algo2) directly computes the distance between two points like Euclidean space.

2 Methods

2.1 Distance between points

When we use the kernel function to estimate the density around the point x:

$$f(x) = \sum_{i=1}^{N} K(\frac{x - x_i}{h}) \tag{4}$$

In the above equation, $(x - x_i)$ is used to describe the *distance* between points. Different ways of evaluating the distance can result in different density functions. So the first thing to do is how to evaluate the distance between two points of circular-linear data.

2.1.1 Algo1

The data given is circular-linear data, with the format(θ, r). The variable θ represents direction, and the variable r represents the size in this direction. It is an intuitive idea that we can convert a vector in polar coordinate to a point in Cartesian coordinate:

$$\begin{aligned} x &= r \cos\theta \\ y &= r \sin\theta \end{aligned} \tag{5}$$

We are familiar with the distance between two points in Cartesian coordinate:

$$d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{6}$$

After getting the data with the format of (θ, r), the data should be pre-processed first. The set after processed is called S_p. In the successive steps, the data set S_p will be used to find clusters.

2.1.2 Algo2

We can also compute the distance directly on the raw data in a way similar to Cartesian coordinate:

$$d(p_1, p_2) = \sqrt{(\theta_1 - \theta_2)^2 + (r_1 - r_2)^2} \tag{7}$$

However, the variable θ is angle in $[0, 2\pi]$, the angle is a kind of periodic value. That means the real distance between 0 and 2π is 0 , not 2π. Similarly, the distance between 0 and $\frac{3}{2}\pi$ is $\frac{1}{2}\pi$. As shown below:

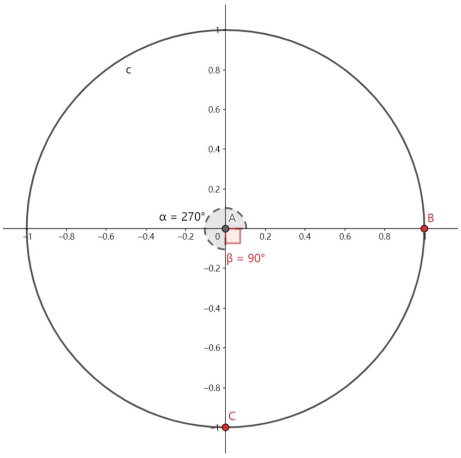


Figure 1: ***Point B represents*** 0***, point C represents*** $3\pi/2$***, the real distance between B and C is*** $\pi/2$

So, the distance between the two angles should be:

$$d(\theta_1, \theta_2) = \begin{cases} 2\pi - |\theta_1 - \theta_2| & |\theta_1 - \theta_2| > \pi \\ |\theta_1 - \theta_2| & else \end{cases} \tag{8}$$

The distance of the circular-linear data is:

$$d(v_1, v_2) = \sqrt{d^2(\theta_1 - \theta_2) + (r_1 - r_2)^2} \tag{9}$$

2.2 Parameters choosing

2.2.1 Bandwidth

The main parameter of the Mean Shift algorithm is the *bandwidth* of the kernel function. If h is small, the point with a larger distance will have a smaller weight. The fewer points will be used to estimate density, the variance will be larger. However, if h is large, this will result in a bigger bias. So, there is a trade-off between bias and variance when choosing bandwidth. Given the size of sampled data N, the bandwidth should meet the requirement that $N \to \infty$, $h \to 0$. By this, we can get h should be :

$$h = cN^{-1/5} \tag{10}$$

where c is a constant.

For the Gaussian distribution, h should be:

$$h = (\frac{4\sigma^5}{3N})^{1/5} \quad = 1.05 * \sigma * N^{-0.2} \tag{11}$$

where σ is the standard deviation for the sample.

2.2.2 Kernel Function

The common kernel function used is ***flat kernel*** :

$$F(x) = \begin{cases} 1 & if||x|| \leq h \\ 0 & if||x|| > h \end{cases} \tag{12}$$

and ***Gaussian kernel***:

$$G(x) = \frac{1}{h\sqrt{2\pi}} e^{-\frac{||x||^2}{2h^2}} \tag{13}$$

The kernel function can be seen as the weight to estimate the point around x. The ***flat kernel*** can be seen as a d-dimensional sphere, only the point in this sphere will be considered. The ***Gaussian kernel*** consider all the points in the set, but closer points will have larger weight.

In this task, we will use the ***Gaussian kernel*** as the kernel function.

2.2.3 Weight Function

Weight function $w(s)$ can be fixed or re-evaluated after each iteration. In this task, the Gaussian kernel has already set the weight for each point, so I just set $w(s) = 1$ in the iteration.

2.2.4 Threshold

The parameter threshold is used to decide when to stop the iteration. The larger threshold will result in more clusters. But smaller thresholds may result in that there is only one mode in the set. In this task, we will set different threshold and analyse the results.

3 Data Analysis

3.1 Computing Means

After the iteration of Mean Shift algorithm, we can get different modes of data. To estimate the basic parameters of different modes, we need to find the centre of each cluster by computing the mean point of each cluster.

The simplest idea is $\bar{x} = \frac{1}{N}\sum_i^N x_i$. For θ, it can be seen as a point on a unit circle. We can cut the circle and make it a line:

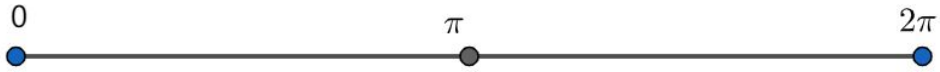


Figure 2: ***Cut from*** π

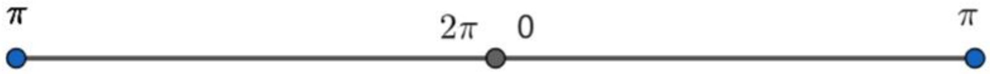


Figure 3: ***Cut from*** 0

By choosing the point where to cut, we can also get different lines. So, the cutting point may have an effect on the result.

Simliar to the idea of *Algo1*, we can also convert the point on the unit circle to Cartesian coordinate : $[x, y]^T = [\cos(\theta), \sin(\theta)]^T$. So we can compute the mean value of each axis separately by :

$$\begin{aligned}\bar{c} &= \frac{1}{N}\sum_{i}^{N}\cos\theta_i \\ \bar{s} &= \frac{1}{N}\sum_{i}^{N}\sin\theta_i\end{aligned} \tag{14}$$

Assuming the mean value of the θ is θ_c, it has the property:

$$\begin{aligned}\cos(\theta_c) = \bar{c} &= \frac{1}{N}\sum_{i}^{N}\cos\theta_i \\ \sin(\theta_c) = \bar{s} &= \frac{1}{N}\sum_{i}^{N}\sin\theta_i\end{aligned} \tag{15}$$

Converting it back to the θ:

$$\theta_c = \begin{cases}\arctan\left(\frac{\sin\theta_c}{\cos\theta_c}\right) & if\cos\theta_c > 0 \\ arctan\left(\frac{\sin\theta_c}{\cos\theta_c}\right) + \pi & if\cos\theta_c < 0\end{cases} \tag{16}$$

For the radius r, we can just compute it by :

$$\bar{r} = \sum_{i}^{N} r_i \tag{17}$$

3.2 Computing Variance

Variance is a measure of dispersion, based on the meaning of distance of two circular-linear points, we compute the variance in the distance space:

$$\begin{aligned}\sigma^2 &= \frac{1}{N-1}\sum_{i=1}^{N}(p_i - p_c)^2 \\ &= \frac{1}{N-1}\sum_{i=1}^{N}d(p_i, p_c)^2\end{aligned} \tag{18}$$

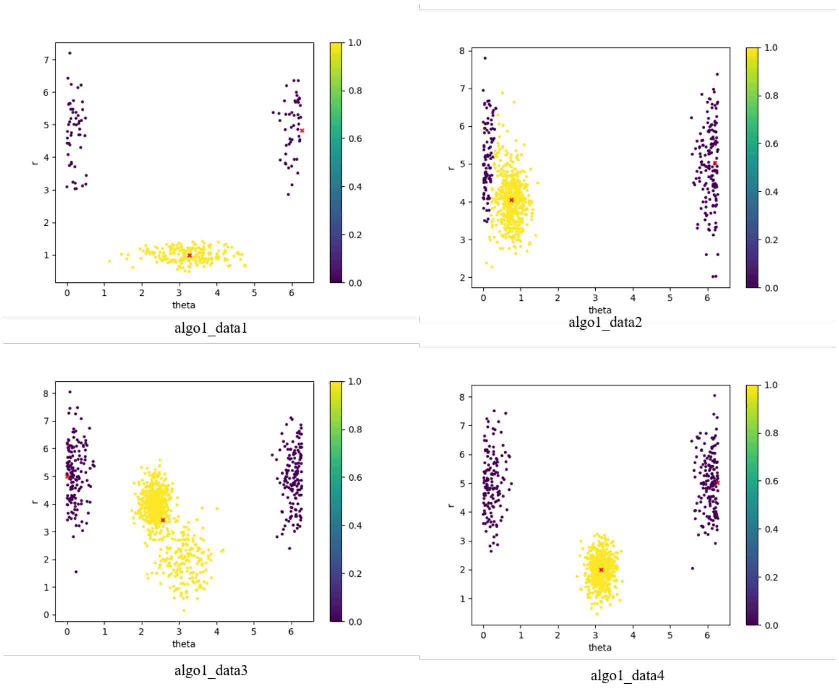
Where d is the equation(9).

4 Conclusion

4.1 Clustering results of given data

Here, I set some basic parameters as the baseline:
1) h (bandwidth) is set by the equation(11);
2) ϵ (threshold) = 0.00001
3) $cluster\_distance$ (to identify points) = 0.5

The results are shown as below,



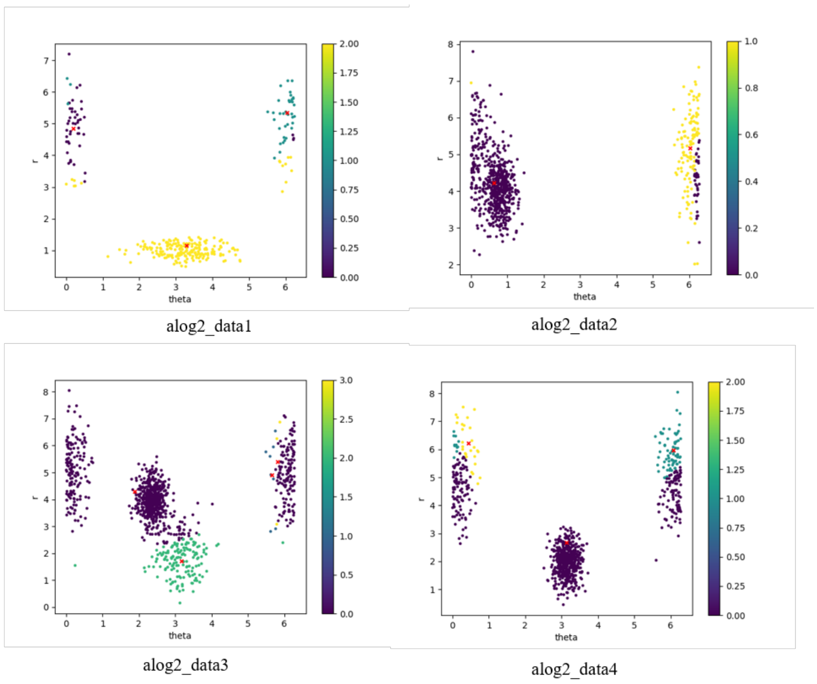
algo1_data1

algo1_data2

algo1_data3

algo1_data4

Figure 4: ***The clustering results used algorithm1***



alog2_data1

alog2_data2

alog2_data3

alog2_data4

Figure 5: ***The clustering results used algorithm2***

The corresponding parameters are:

For *Algo1*:

1)*Algo1_data1*:

Mean1, variance1 = [6.264581278374883, 4.835123552413281], 0.939911335274201

Mean2, variance2 = [3.260652115284019, 0.9951555487518141], 0.4847930143675687

2)*Algo1_data2*:

Mean1, variance1 = [6.195806779743288, 5.038739483173911], 1.0144883862305363

Mean2, variance2 = [0.7578102016545607, 4.056523895407684], 0.43649180883735894

3)*Algo1_data3*:

Mean1, variance1 = [0.0011053485412668106, 5.008956198106666], 1.1546428427750621

Mean2, variance2 = [2.561818743211738, 3.4347695060328554], 1.345255056792747

4)*Algo1_data4*:

Mean1, variance1 = [6.275052766052679, 5.0138292362733345], 1.126485967720785

Mean2, variance2 = [3.1462162642512377, 1.985775081232], 0.28859923244214547

For *Algo2*:

1)*Algo2_data1*:

Mean1, variance1 = [0.20222136352012238, 4.854327167542238], 0.6226003937141277

Mean2, variance2 = [6.042231988296247, 5.349348037649856], 0.43750127152496493

Mean3, variance3 = [3.2814843637511655, 1.1633712602153288], 1.4042334165883408

2)*Algo2_data2*:

Mean1, variance1 = [0.6241342718010923, 4.22775436941789], 0.7075564329703974

Mean2, variance2 = [6.017494334803616, 5.172806610915254], 1.1435871536830982

3)*Algo2_data3*:

Mean1, variance1 = [1.8852498215398534, 4.291848884517322], 2.301119305564973

Mean2, variance2 = [5.660576532243779, 4.90552073475], 1.923843711654354

Mean3, variance3 = [3.1683116862320397, 1.7059184362171056], 0.537530802469436

Mean4, variance4 = [5.82388720943778, 5.401409710333334], 4.150890959672452

4)*Algo2_data4*:

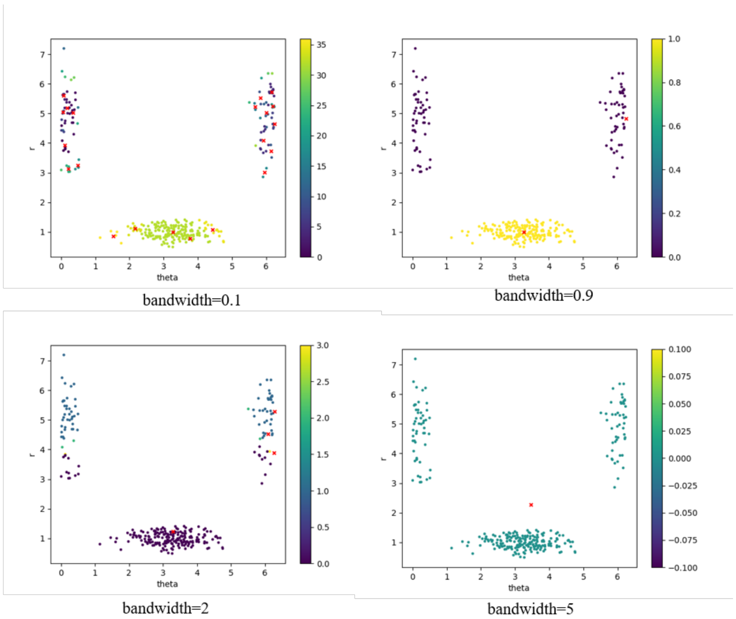
Mean1, variance1 = [3.138574735333872, 2.667478984917632], 3.954162764034368

Mean2, variance2 = [6.0711038101168135, 5.9668292419135796], 0.3574547782738188

Mean3, variance3 = [0.43445628507695205, 6.215840197777778], 0.6251928397153621

4.2 Parameters effects

Use data1 as the sampled set, Algo1 as the algorithm, threshold = 0.0001; The results with different bandwidth are shown below:



bandwidth=0.1

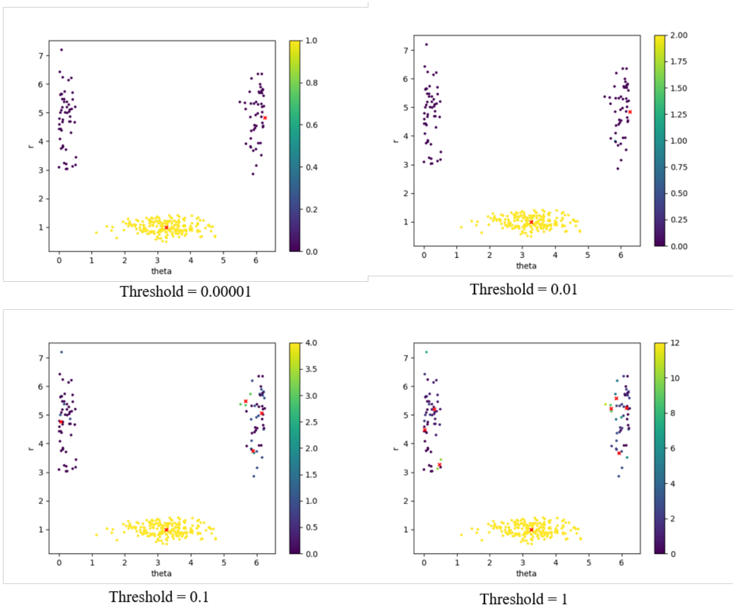
bandwidth=0.9

bandwidth=2

bandwidth=5

Figure 6: ***Different bandwidth***

Use data1 as the sampled set, Algo1 as the algorithm, bandwidth = 0.9; The results with different threshold are shown below:



Threshold = 0.00001

Threshold = 0.01

Threshold = 0.1

Threshold = 1

Figure 7: ***Different bandwidth***