The task of Mean Shift Algorithm

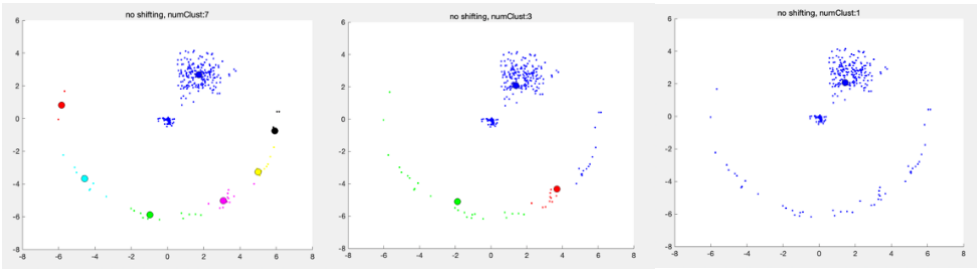
I feel excited about this project. Firstly, I would like to point out one question, which confuses me until this afternoon. The format should be [r, theta] rather than [theta, r] and I make it after I solve this question.

From email: In the attached files you can find data sampled from multimodal distributions (the number of modes is unknown) following the format [theta, r], where r is in (-inf,+inf) and theta in (0, 2pi).

(1) Analyze the data

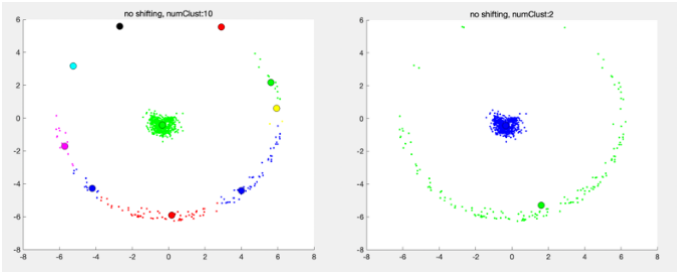
We use the traditional Mean Shift to cluster the data and locate the problem. (The Mean Shift has the kernel so the optional parameter is the bandwidth for the kernel).

Data1:



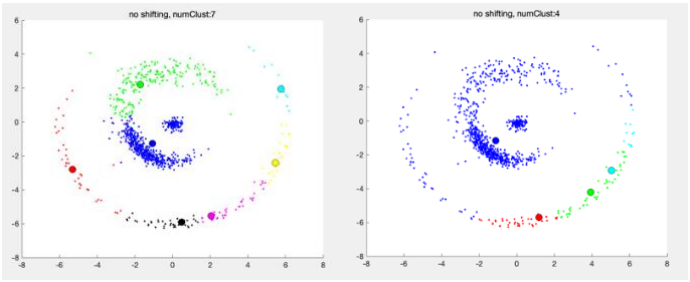
Bandwidth =2 Bandwidth =4 Bandwidth =6

Data2:



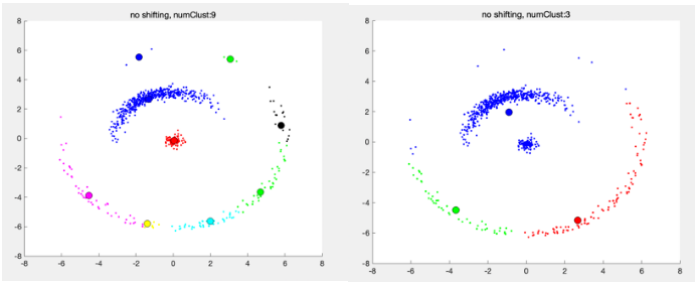
Bandwidth=2 Bandwidth=4

Data3:



Bandwidth = 2 Bandwidth = 3

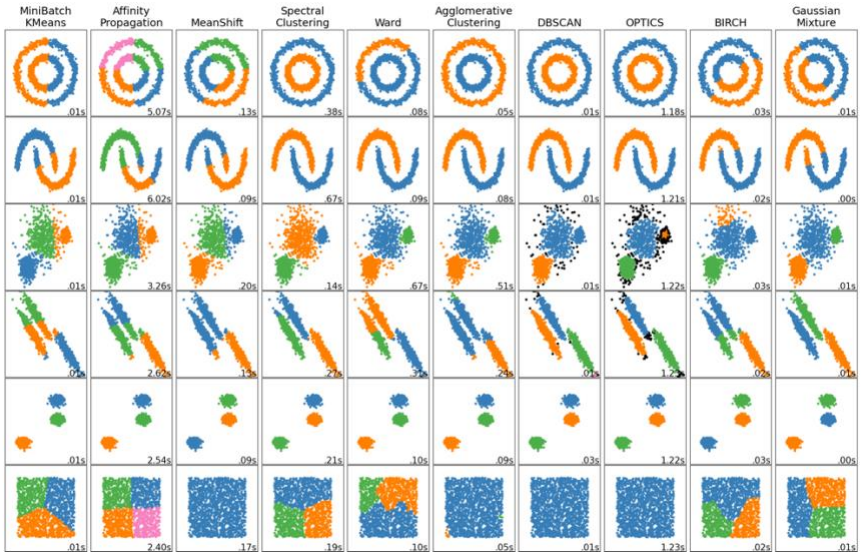
Data4:



Bandwidth = 2 Bandwidth = 3

From the results of the 4 datasets, we could conclude that traditional Mean Shift can't deal with the circular-linear data. It's because MS uses the distance to measure the density but it fails when facing directional data such as circular-linear data. For example, two points locate in the same circle, one is [r=3, theta=0], another is [r=3, theta=3.14]. The distance between the two points is far but they are in the same circle. Thus, we have to polish the MS and make it work here.

(2) How do we deal with circular-linear data



From the above image, we could see that spectral clustering, agglomerative clustering, DBSCAN and OPTICS can deal with circular-linear data. Although I make it using DBSCAN, how to use MS algorithm here is the task.

(3) The improvement of Mean Shift

To visualize the data, we transfer the [r, theta] to [x, y]. Traditional MS measure the distance from a point to another point. I refer to some papers such as "On mean shift-based clustering for circular data" and "On Mean Shift Clustering for Directional Data on a Hypersphere" but it seems too complex.

I just want to try some simple but effective ideas of my own. In practice, I only measure the distance of "r" without considering the "theta", which has good performance in the circular-linear data. Next, I will demonstrate the results.

Data1:

There are 3 clusters, which have 53, 48, 199 points separately.

For cluster 1: the mean value is (0.021394，-0.109178),

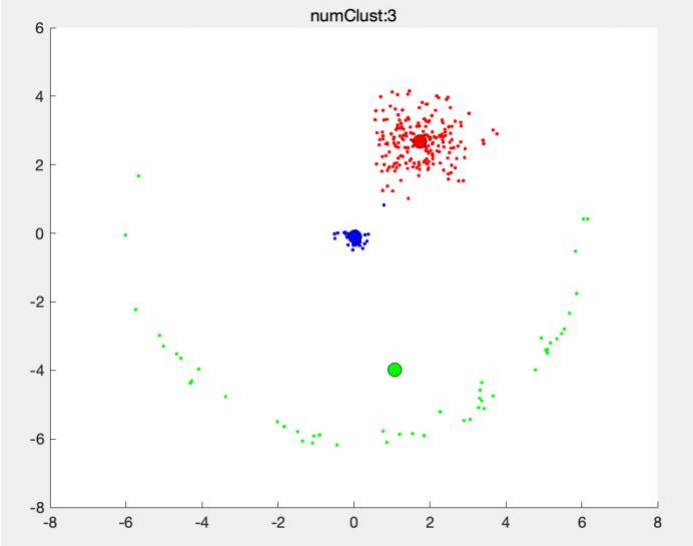
the covariance is
```
0.0434    0.0067
0.0067    0.0348
```

For cluster 2: the mean value is (1.071562，-3.977674),

the covariance is
```
15.9137    0.7139
 0.7139    3.7454
```

For cluster 3: the mean value is (1.745260，2.681609),

the covariance is
```
0.4368    0.0076
0.0076    0.3933
```



Bandwidth = 1

Data2:

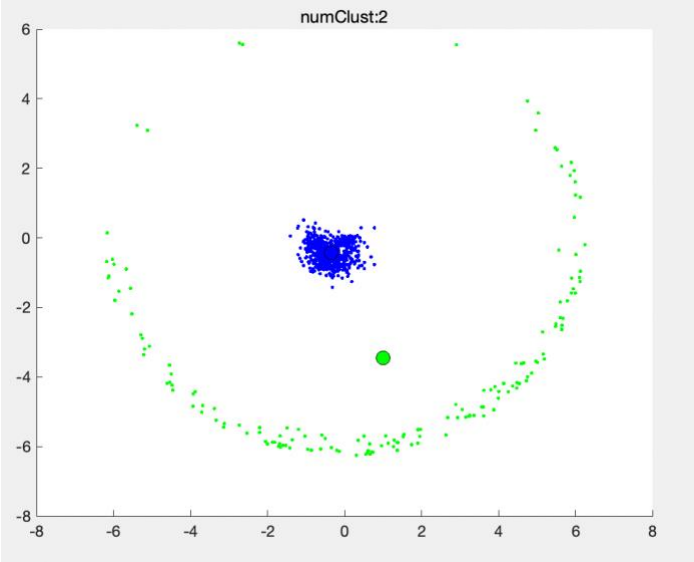
There are 2 clusters, which have 644, 156 points separately.

For cluster 1: the mean value is (-0.340951，-0.430915),

the covariance is
```
0.1331    0.0036
0.0036    0.1035
```

For cluster 2: the mean value is (0.998672，-3.456956),

the covariance is
```
15.8096    2.0754
 2.0754    8.2109
```



Bandwidth = 1

Data3:

There are 3 clusters, which have 700, 148, 152 points separately.

For cluster 1: the mean value is (-1.245077，-0.501192),

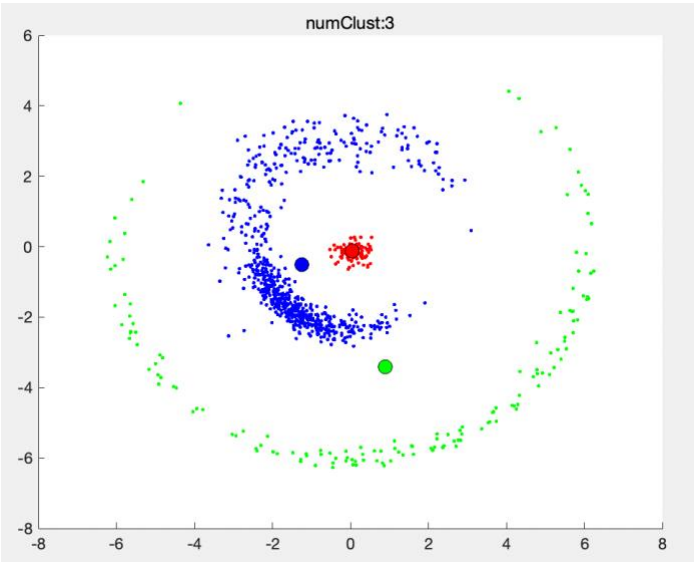
the covariance is
```
1.3603    0.2061
0.2061    3.6809
```

For cluster 2: the mean value is (0.901619，-3.411387),

the covariance is
```
17.1633    1.3591
 1.3591    7.2252
```

For cluster 3: the mean value is (0.051268，-0.119289),

the covariance is
```
0.0391    0.0004
0.0004    0.0267
```



Bandwidth = 1

Data4:

There are 3 clusters, which have 158, 142, 500 points separately.

For cluster 1: the mean value is (1.089410，-3.437098),

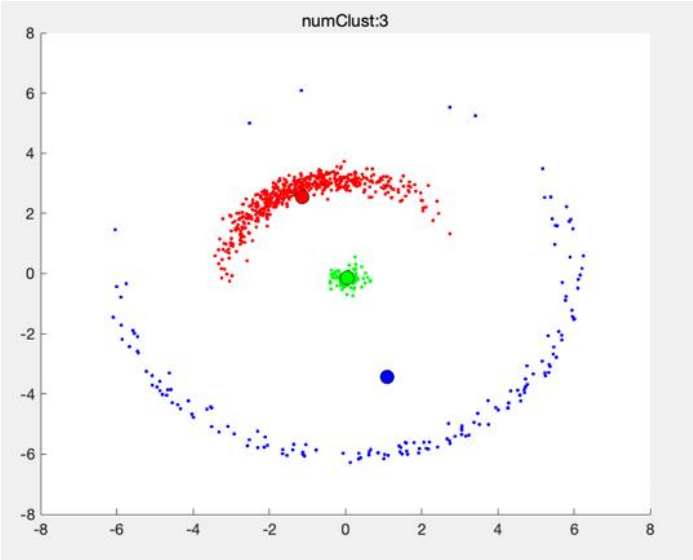
the covariance is
```
16.2010    2.6128
 2.6128    7.5875
```

For cluster 2: the mean value is (0.047525，-0.130013),

the covariance is
```
0.0389    0.0009
0.0009    0.0323
```

For cluster 3: the mean value is (-1.129794，2.540492),

the covariance is
```
1.7247    0.5868
0.5868    0.4897
```



Bandwidth = 1

(4) The future

At present, we just use the strategy to deal with the circular-linear data. It's better to combine the concept of DBSCAN or spectral clustering with MS to make it more robust. In the future, more effort should be paid to make the MS useful for any directional data.

Thanks for your attention. Please contact me if you have any questions.

Baichuan Huang
huangbaichuan@whu.edu.cn
https://whubaichuan.github.io/