Mean Shift Clustering for Circular-Linear Data

Yufei Zhu
yufeizhu@kth.se

December 2021

1 Introduction

In the unsupervised machine learning area, clustering algorithms can be used to divide unlabeled data into groups based on similarities and patterns. Mean shift is one of clustering algorithms. It is a non-parametric algorithm for clustering and mode detection. Without *a priori* specification of numbers of clusters, the algorithm iteratively shifts data points to a higher density region until convergence. The application of mean shift algorithm includes image segmentation and object tracking.

Mean shift procedure was proposed in 1975 by Fukunaga and Hostetler [1]. In 1995, Cheng generalized and analyzed the procedure in [2] and attracted more attention to mean shift. Comaniciu et al. [3] then extended it for two low-level vision tasks, discontinuity preserving smoothing and image segmentation.

In this report, a mean-shift-cluster tool is developed, which accepts both circular-linear data and linear data. The tool is available at github.

2 Method

Mean shift algorithm is based on Kernel Density Estimation (KDE). KDE, also known as the Parzen window technique in pattern recognition, is a method to estimate the probability density function of a random variable. Given n data points x_i, i=1,...,n in a d-dimensional space R^d, a kernel density estimator, in the point x, is given by

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

where $K(x-x_i)$ is a kernel function. To find the unknown number of modes, mean shift algorithm will maximize the above continuous density function (1).

With kernel function, the sample mean at x is

$$m(x) = \frac{\sum_{i=1}^{n} K(x_i - x)x_i}{\sum_{i=1}^{n} K(x_i - x)} \tag{2}$$

Starting from different data points, modes are found iteratively through gradient ascent. In each iteration, $x \leftarrow m(x)$ is performed for all x simultaneously.

The mean function (5) is computing a weighted average of data points which are close to the current point x. x then is moved to the computed mean, $m(x)$. The iteration will converge to different modes.

Three kernel functions are included in this report:

- flat kernel

$$K(x) = \begin{cases} 1 & \text{if } \|x\| \leq \lambda \\ 0 & \text{if } \|x\| > \lambda \end{cases} \tag{3}$$

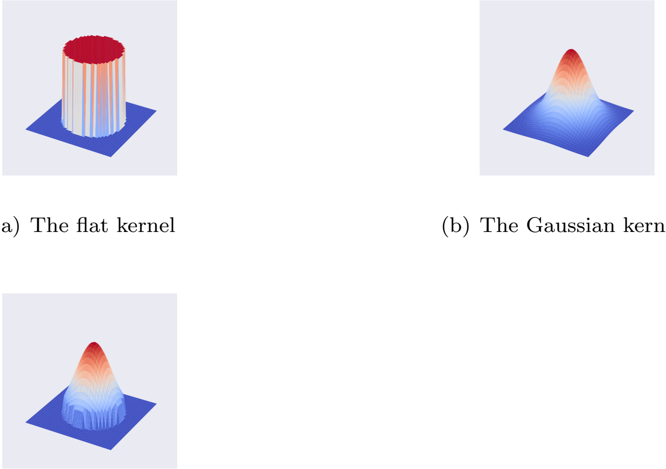
- Gaussian kernel

$$K(x) = e^{-\beta \|x\|^2} \tag{4}$$

- truncated Gaussian kernel

$$K(x) = \begin{cases} e^{-\beta \|x\|^2} & \text{if } \|x\| \leq \lambda \\ 0 & \text{if } \|x\| > \lambda \end{cases} \tag{5}$$

The above three kernels are shown in Figure 1.



(a) The flat kernel

(b) The Gaussian kernel

(c) The Truncated Gaussian kernel

Figure 1: Illustration of three kernels. (a) is the flat kernel with $\lambda = 1.5$. (b) is the Gaussian kernel with $\beta = 0.1$. (c) is the truncated Gaussian kernel with $\lambda = 1.5$, $\beta = 0.1$

3 Implementation

In this project, a mean-shift-cluster is developed. The pseudo-code is shown in Algorithm 1. The input parameters are kernel type, kernel parameters, data type, and the maximum number of iteration times. Supported data types are linear data and circular-linear data. The tool is implemented using Python 3.8.

When the input data is circular-linear data, it is required that data format is [θ, r], where r is in (-inf,+inf) and θ in (0, 2π). The input data will be converted from polar coordinates to Cartesian coordinates.

To run for loops using multiprocessing, a python library, joblib.parallel [4] is used.

Algorithm 1: Mean shift Algorithm

```
validate_inputs(dataset, kernel_parameter);
initial_start_points = generate_initial_start_points();
for point $\in$ initial_start_points do
    mean $\leftarrow$ point;
    while shift_mean $>$ convergence_threshold do
        last_mean $\leftarrow$ mean;
        mean $\leftarrow$ compute_mean_value(mean);
        shift_mean $\leftarrow$ mean - last_mean;
    clusters $\leftarrow$ clusters $\cup$ { mean };
clusters $\leftarrow$ eliminate_near_duplicates_points(clusters)
```

When developing the mean-shift-cluster, we use scikit-learn library [5] as a reference and follow its naming convention. One limitation of the scikit-learn library mean shift cluster is that it is developed only based on flat kernel usage. We refactor mean calculation part and change the class to a more generalized pattern that can integrate with different kernels.

In mean-shift-cluster, three kernels are provided, which are flat kernel, Gaussian kernel and truncated Gaussian kernel. The kernel info validation and circular-linear data validation are implemented.

Mean-shift-cluster provides three functions:

- Fit input dataset using mean shift algorithm. Cluster centers and labels for each sample can be accessed.
- Provide basic information of each cluster(mode), using input dataset.
- Predict the cluster results of any dataset, using the fitted mean-shift-cluster.

4 Results

In this section, we present the behaviors of mean-shift-cluster when utilizing different kernels. Then a special case of data shape is discussed.

4.1 Clustering results with different kernels

Here we discuss the behaviors of mean-shift-cluster tool when utilizing each kernel. Three aspects will be presented, which are clustering results, computation cost, and parameter tuning. Four circular-linear datasets are used.

4.1.1 Flat Kernel

With flat kernel, the clustering results of each dataset are shown in Figure 2. Samples are shown in different colors, by different clusters. For each cluster(mode), the mean value of samples is marked as a red spot. The convergence threshold is set to $\lambda * 10^{-3}$.

The bandwidth parameter, λ can affect clustering results. The influence of bandwidth is presented in Table 1. The main time-consuming part of the algorithm is the convergence for each starting point. When decreasing the bandwidth, it will take more iterations to converge and result in a larger number of clusters. So computation time will increase. And when bandwidth is too large, the clustering result becomes inaccurate and some clusters can be lost. In Figure 3, it can be seen that when bandwidth is 2, two modes are detected. And when the bandwidth is 6, all samples are grouped in one mode.

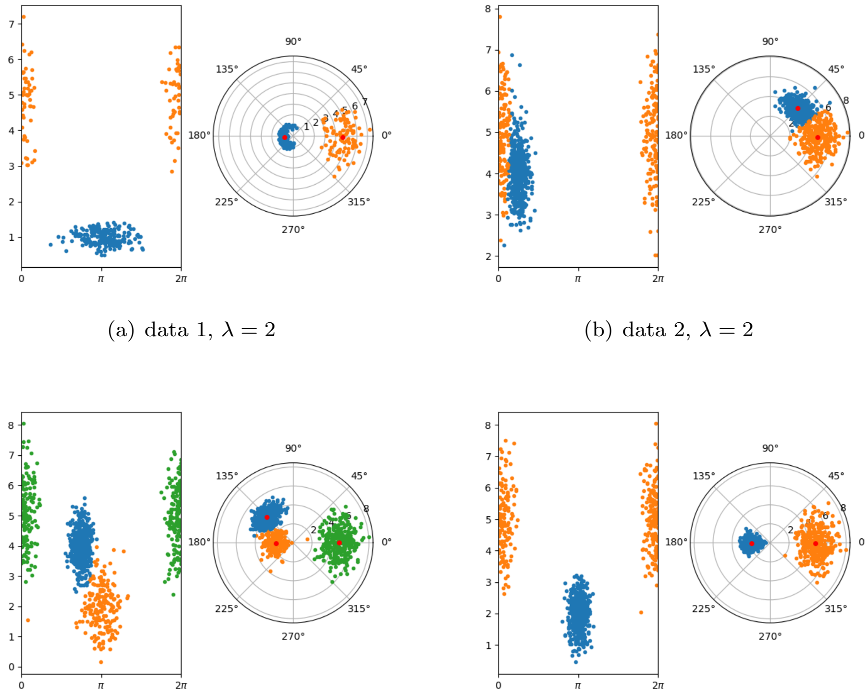
Bandwidth	Computation Time (s)	Max Iterations
1	13.28	35
1.5	10.00	19
2	10.73	15
10	9.06	3
16	8.46	2

Table 1: Algorithm performance with different bandwidth settings when using flat kernel.

4.1.2 Gaussian Kernel

The clustering results of Gaussian kernel are shown in Figure 4. The convergence threshold is set to 10^{-6}.

The parameter, β can be seen as bandwidth, and also affect clustering results. The computation time and max iterations with different bandwidths are shown in Table 2. When β parameter increases, it has a similar effect of decreasing bandwidth, which leads to a larger number of max iterations to converge and longer computation time. The clustering result with different bandwidth is shown in Figure 5. When β increases, the number of detected modes increases.



(a) data 1, $\lambda = 2$

(b) data 2, $\lambda = 2$

(c) data 3, $\lambda = 1.5$

(d) data 4, $\lambda = 2$

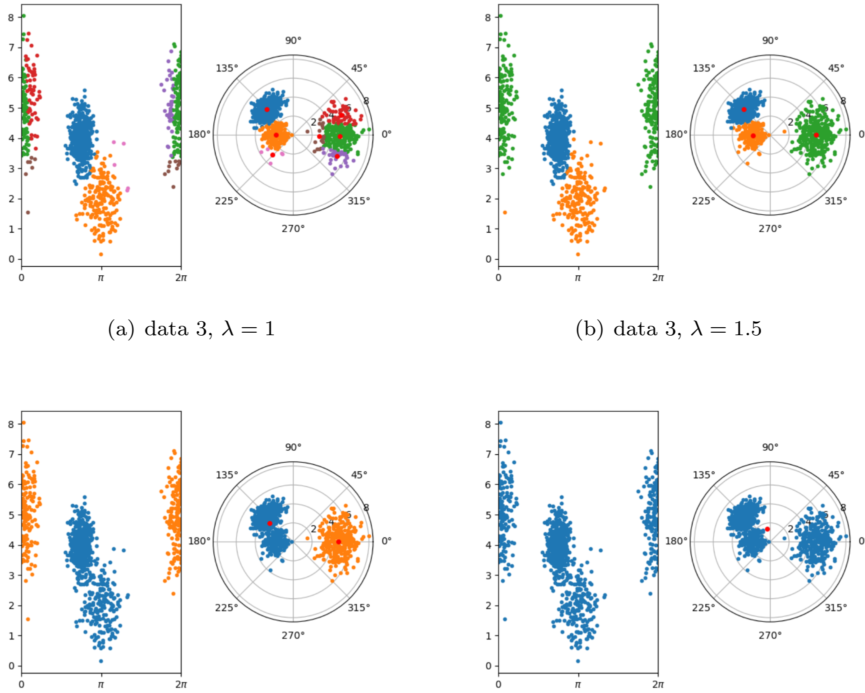
Figure 2: Clustering results of flat kernel

Bandwidth	**Computation Time (s)**	**Max Iterations**
0.1	6.82	20
0.5	7.47	38
1	8.67	67
2	9.97	71
8	12.79	272

Table 2: Algorithm performance with different bandwidth settings when using Gaussian kernel.

4.1.3 Truncated Gaussian Kernel

The clustering results of truncated Gaussian kernel are presented in Figure 6. Both λ and β affect the clustering results. Within a range, the number of modes increases when λ decreases or β increases. If β keeps decreasing, λ will be the main factor.



(a) data 3, $\lambda = 1$

(b) data 3, $\lambda = 1.5$

(c) data 3, $\lambda = 2$

(d) data 3, $\lambda = 6$

Figure 3: Clustering results of flat kernel with different bandwidth

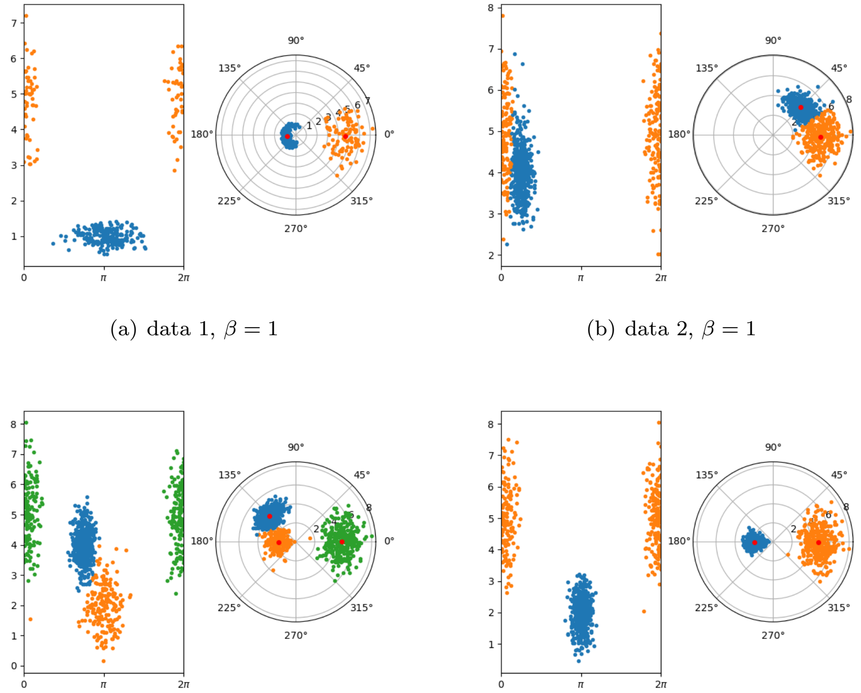
4.1.4 Kernel Comparison

By applying three kernels to four datasets, we can compare the behavior of kernels. From the aspect of clustering results, with proper settings of parameters, all kernels can be used to detect modes successfully.

With convergence threshold set to 10^{-6}, the computation time is listed in Table 3, where flat kernel has the lowest computation time. The time complexity of mean shift algorithm is $\mathcal{O}(n^2)$, where n is the size of the dataset.

Kernel	**Data 1 (s)**	**Data 2 (s)**	**Data 3 (s)**	**Data 4 (s)**
flat	0.97	7.14	10.38	6.35
Gaussian	1.34	8.59	12.86	7.87
truncated Gaussian	1.40	8.44	13.00	8.18

Table 3: Computation time when using different kernels.



(a) data 1, $\beta = 1$

(b) data 2, $\beta = 1$

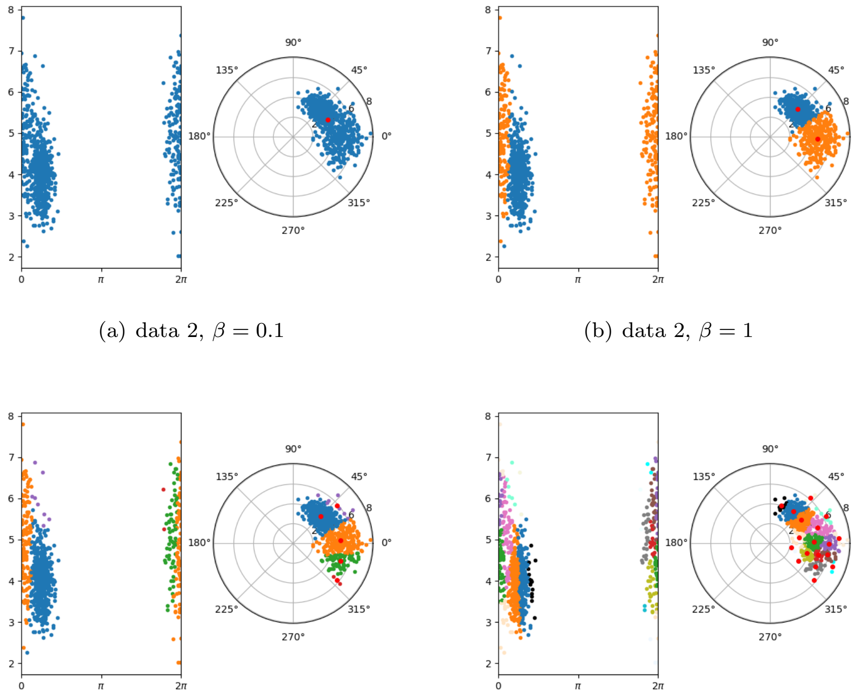
(c) data 3, $\beta = 1$

(d) data 4, $\beta = 1$

Figure 4: Clustering results of Gaussian kernel

4.2 One Special Case of Circular-Linear Data

Figure 7 presents one special case of circular-linear data. When dataset is in a circular shape, converting data to Cartesian coordinates will lead to the clustering results in Figure 7(b). In this case, we can use data in polar coordinates directly and the mean shift algorithm will detect the modes.



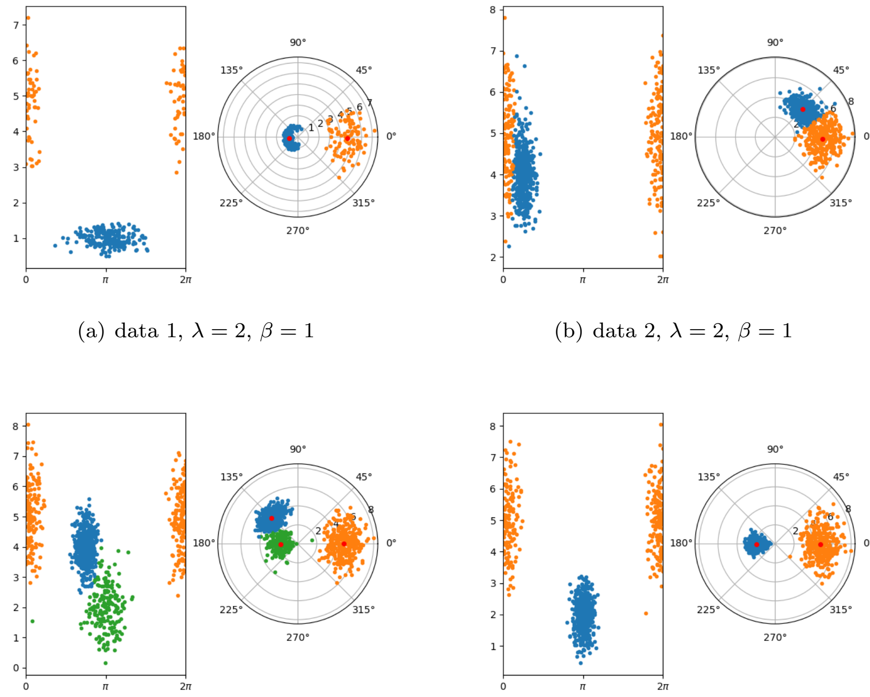
(a) data 2, $\beta = 0.1$

(b) data 2, $\beta = 1$

(c) data 2, $\beta = 4$

(d) data 2, $\beta = 8$

Figure 5: Clustering results of Gaussian kernel with different bandwidth



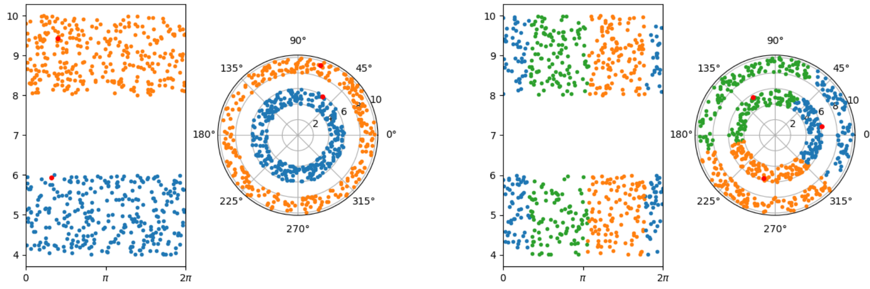
(a) data 1, $\lambda = 2$, $\beta = 1$

(b) data 2, $\lambda = 2$, $\beta = 1$

(c) data 3, $\lambda = 2$, $\beta = 1$

(d) data 4, $\lambda = 2$, $\beta = 1$

Figure 6: Clustering results of truncated Gaussian kernel



(a) data 5, flat kernel, $\lambda = 3$

(b) data 5, flat kernel, $\lambda = 6$

Figure 7: Clustering results of circular shape dataset

5 Conclusions

In this project, a mean-shift-cluster tool is developed and evaluated using five datasets. Clustering results and computation cost are presented and discussed. Three kernels are supported by mean-shift-cluster, which are flat, Gaussian and truncated Gaussian. For each kernel, clustering results, computation time, and max iteration numbers are presented and compared with different settings of parameters. By comparing behaviors of all three kernels, we conclude that all kernels can achieve expected clustering results and flat kernel has the least computation time.

One special case of circular-linear data is discussed. In 4.2), one extra dataset for circular shape data in generated. In this case, converting circular-linear data to Cartesian coordinates will lead to unexpected clustering results.

After evaluation and results discussion, we can conclude the advantages of mean shift algorithms:

- Does not need cluster numbers as input. Only needs bandwidth.
- Robust to outliers. By using KDE, the outliers will not affect cluster centers much.
- Can be efficient with complex shape of input data.

And the disadvantages of mean shift algorithms are:

- Scalability can be an issue. The time complexity is $\mathcal{O}(n^2)$. Both to large datasets and to high dimensions.
- Need tuning the parameter to select proper bandwidth.
- In some cases, outliers can be detected as a cluster. Need post-processing of the results.

References

[1] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975. [Page 1.]

[2] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995. [Page 1.]

[3] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002. [Page 1.]

[4] Joblib Development Team. Joblib: running python functions as pipeline jobs, 2021. [Page 3.]

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. [Page 3.]