

Cluster Finder for 1D and 2D Histogram Data

Xiwen Gong^{1,†}, Zilin Zhou^{2,*,†}

¹ Greenwich High School, Greenwich, 06830, United States

² La Salle College Preparatory, Pasadena, 91107, United States

¹ xiwengong111@gmail.com

² zilinzhou08@gmail.com

*corresponding author

†co-first authors

Abstract. Clustering is a fundamental unsupervised task in machine learning. It involves grouping a set of objects into distinct clusters based on their inherent properties. Clustering finds applications in various domains, such as image segmentation, customer segmentation, document categorization, anomaly detection, and social network analysis. In this paper, we investigate several clustering algorithms applied to 1D and 2D histogram data. In particular, we try the Center of Gravity, Gaussian Mixture Model, and Neural Network and conclude that in low-dimensional cases, simple methods can yield good performance.

Keywords: 1D, 2D, Mean Squared Error.

1. Introduction

In this paper we analyzed 1D and 2D multiple clustering datasets using different types of clustering models. The datasets that we are given contain multiple clusters. Our goal is to evaluate the results and performances of different models. The methods we used includes the Center of Gravity, Neural Networks, and Gaussian Mixture Model. We begin by describing the datasets separately and the pre-processing steps we did, then we detailed the setup and resolutions of each model applied to the datasets. Lastly, we concluded our findings and discussed the implications of our results. Throughout this study our goal is to determine which model is best at identifying the clusters in the dataset and provide the most accurate results.

There are two types of datasets, namely 1D and 2D, used throughout the study. Each entry contains the number of points lying in the bin (1D case) or the cell (2D case) respectively. The bin boundaries are consecutive from 0 to 48 with step 1. The cells are generated from a uniform mesh grid with both x and y coordinates delimited by 0 and 64 with gap 1.

2. Method

2.1. Center of Gravity

For a bin $[i - 1, i]$ and its associated bin count c_i , we determine that c_i is a local maxima if c_i is larger than $c_{i-\sigma}, \dots, c_{i-1}, c_{i+1}, \dots, c_{i+\sigma}$, where σ is a tunable hyperparameter by which we can adjust the

magnitude of neighborhood. To some extent and especially in multiple-cluster case, it can help us handle the noisy data.

Similarly for a cell (i, j) and its associated bin count $c_{i,j}$ we expect this count to be larger than that of its neighbors surrounding it in two directions. That is, $c_{i,j}$ should be larger than $c_{i-\sigma,j}, \dots, c_{i+\sigma,j}, \dots, c_{i,j-\sigma}, \dots, c_{i,j+\sigma}$ (neglecting crossing the boundaries).

The intuition behind the center of gravity is that, the cluster center should be a local maxima in terms of the frequency of points. But to avoid the effect brought by noise, we expect the value of local maxima to be larger than other numbers within a small window around itself [1].

2.2. Gaussian Mixture Model

Gaussian mixture model assumes that the data points are generated with a weighted sum of K Gaussian distributions. That is, for each data point x , its probability is

$$p(x) = \sum_{i=1}^K \pi_i N(x; \mu_i, \sigma_i^2)$$

where π_i is the probability that x is drawn from i -th Gaussian distribution and $N(x; \mu_i, \sigma_i^2)$ is the probability of obtaining x with the i -th Gaussian distribution.

The Gaussian mixture model can be modelled with maximum likelihood estimation. Under the assumption identically-and-independently-distributed samples (data points), the likelihood function can be written as:

$$L(\theta) = \prod_{i=1}^n p_{\theta}(x_i) = \prod_{i=1}^n \sum_{j=1}^K \pi_j N(x_i; \mu_j, \sigma_j^2)$$

The objective of Gaussian mixture model is to maximize the above likelihood function with respect to parameters $\theta = (\pi, \mu_1, \dots, \mu_K, \sigma_1, \dots, \sigma_K)$:

$$\theta^* = \arg \max_{\theta} L(\theta)$$

subject to $\sum_{i=1}^K \pi_i = 1$

2.3. Neural Network

Neural network is a universal function approximator (given enough artificial neuron cells, the neural network is theoretically capable of approximating any function).

Typically, during neural network training, we choose a loss function ℓ can compute the total loss $C(\theta)$ incurred on the whole dataset, where θ is the neural network parameter:

$$C(\theta) = \sum_{i=1}^n \ell(\text{NN}_{\theta}(x_i), y_i).$$

The objective is apparently to minimize the total loss:

$$\theta^* = \arg \min_{\theta} C(\theta).$$

Though we can explicitly write down the formula of C w.r.t. θ , there is no easy closed-form solution to the optimization problem above.

Therefore, we turn to iterative algorithms, namely the gradient descent family (e.g. stochastic gradient descent, Adam solver), to solve the optimization problem above.

3. Experiment

3.1. Data

The 1-dimensional dataset consists of 100000 rows, each containing 55 values. The values represent, in order, ID/index of data (1), number of clusters (2), centers of clusters, count of data points in each cluster (2), and the histogram bin counts for each cluster (50).

The 2-dimensional dataset consists of 100 entries, each containing 10 ground-truth cluster centers and a 64x64 array that contain the number of points in each cell.

3.2. D 2-cluster Case

In this case, we only have two clusters in each entry. The objective is to predict the distance between the two clusters [2].

3.2.1. Center of Gravity

In Center of Gravity, we simply set $\sigma = 2$ and check the local maxima in a single loop. After finding the cluster centers, we subtract them to give the distance. The distribution of absolute difference between the predicted distance and the ground-truth distance is depicted in Fig. 1.

3.2.2. Gaussian Mixture Model

The data is read in chunks of 1000 which are then processed in parallel to improve the speed of the program. We fit a Gaussian Mixture Model with 2 components to the bin centers. The absolute difference between the means of the two Gaussian distributions is then computed. We compute the error between the predicted and actual differences of the cluster centers, and plot a histogram of the errors.

FIG. 2 shows the result of the Gaussian Mixture Model.

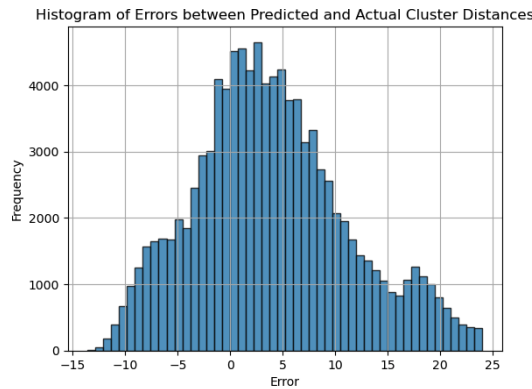


Figure 1. Histogram of Errors between Predicted and Actual Cluster Distances

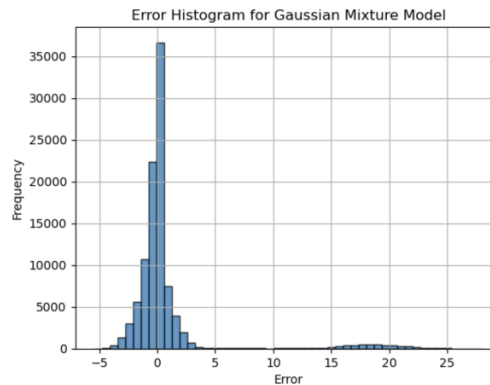


Figure 2. 1D Error Histogram for 2-cluster Gaussian Mixture Model

3.2.3. Neural Network

In the Neural Network model, we take the input data to be the histogram bin counts for each cluster [3]. The neural network is a Multi-Layer Perceptron (MLP) with input layer of 50 neurons (corresponding to 50 histogram bin counts), 3 hidden layers with 48, 100, and 100 neurons, respectively, and an output layer of 1 neuron (predicting the distance between the two cluster centers). The network is trained using the adam solver, and the loss function used is the Mean Squared Error (MSE), which measures the average of squares of the errors between the predicted and actual values.

The figure below shows the result of the Neural Network.

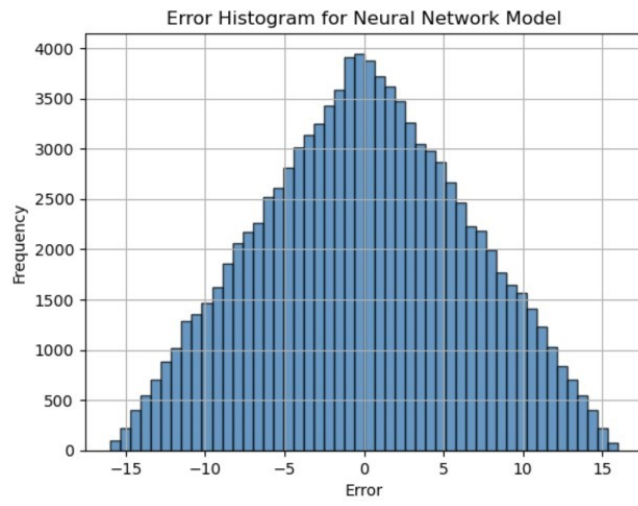


Figure 3. 1D Error Histogram for 2-cluster Neural Network

3.3. D Multiple-cluster Case

Result of different models on the 2D multiple-cluster dataset, shown by figures.

3.3.1. Center of Gravity

Describe the setup of the model in 2D case. The Center of Gravity model calculates the central point of each cluster based on the provided 2D coordinates in a text file. In this case, the “true clusters” and “found clusters” are compared to assess the model’s performance. The script processes the clusters, calculates the differences.

In the 2D multiple-cluster model, the Center of Gravity (CoG) model identifies and locate clusters by determining the local maxima within a 2D histogram. The 2D histogram’s result is generated from the provided data text file. Each cell in the histogram grid represents a bin. To determine the local maxima, we set up a hyper- parameter, which defines the neighborhood window. For a cell (i, j) with a count $c_{i,j}$ we can identify the local maxima if

$$c_{i,j} > c_{i-\sigma,j}, \dots, c_{i+\sigma,j}, c_{i,j-\sigma}, \dots, c_{i,j+\sigma}$$

This can help us reduce noise by making sure that the identified cluster centers are at peaks compared to the surroundings. We processed the 2D dataset to evaluate the gap between true and predicted clusters. The overall performance of the Center of Gravity model is evaluated by comparing those differences.

3.3.2. Neural Network

Describe the setup of the model in 2D case.

In case of the 2D clusters we used a Neural Network model to predict the cluster centers that’s extracted from the 2D histogram data. And we used a Multi-Layer Perceptron (MLP) for our neural network model [4] [5].

The MLP includes the following structures; Input Layer: Flattened 64x64 grid (4096 input neurons) Hidden Layer 1: 256 neurons Hidden Layer 2: 128 neurons Hidden Layer 3: 64 neurons Output Layer:

2 neurons (predicting the x and y coordinates of the cluster center). The input of this model is the 2D histogram, and the output was the predicted coordinates and location of the cluster centers. In order to train the neural network model, we used methods such as the Adam optimizer and the Mean Squared Error (MSE) loss function. The MSE calculates the average of the squares of the errors that's in between the predicted and true values. The overall performance of the Neural Network Model was evaluated by comparing the predicted cluster centers and the true centers.

4. Conclusion

Summarize the performances of different models (e.g. the error distribution) and discuss which is better and why. In the 1D models, the Gaussian Mixture Model performed the best. It had the lowest error variance and most accurate predictions, as indicated by the concentration of errors around 0 with a small group near 15 and 20. The Center of Gravity method centered around 0, but had a much larger spread of errors. This suggests that it is more susceptible to noise and less precise compared to the Gaussian Mixture Model. The Neural Network also showed a peak at 0 with a relatively symmetrical spread around it. It is less accurate than the Gaussian Mixture Model, but showed a smaller range of errors compared to the Center of Gravity Model. The Center of Gravity method can provide a reasonable cluster center prediction. It can do that by setting an appropriate, this model can identify local maxima as cluster centers effectively, it indeed is an effective method but it might be sensitive to noise and depends on the choice of. On the other hand, The Neural Network model showed an overall better performance by learning more complex spatial patterns and data from the 2D histogram data. It shows more accurate predictions of cluster centers compared to the Center of Gravity method. However, the Neural Network model requires more computational resources and more careful tuning of parameters.

5. Acknowledgement

Xiwen Gong and Zilin Zhou contributed equally to this work and should be considered co-first authors.

References

- [1] Hejazi, M. M., Golabi, F., Bahrami, M., Kahroba, H., & Hejazi, M. (2022). Fmsclusterfinder: a new tool for detection and identification of clusters of sequential motifs with varying characteristics inside genomic sequences. *bioRxiv*.
- [2] Feldbrugge, J., & Weygaert, R. V. D.. (2023). Cosmic web & caustic skeleton: non-linear constrained realizations 2d case studies. *Journal of Cosmology and Astroparticle Physics*.
- [3] Moayedi, H., Salari, M., Dehrashid, A. A. , & Le, B. N. . (2023). Groundwater quality evaluation using hybrid model of the multi-layer perceptron combined with neural-evolutionary regression techniques: case study of shiraz plain. *Stochastic environmental research and risk assessment*(8), 37.
- [4] Kang, H., Xu, Q., & Duofang ChenShenghan RenHui XieLin WangYuan GaoMaoguo GongXueli Chen. (2024). Assessing the performance of fully supervised and weakly supervised learning in breast cancer histopathology. *Expert Systems with Application*, 237(Mar. Pt.B), 121575.1-121575.13.
- [5] Wu, S., Du, Z., Zhang, F. , Zhou, Y. , & Liu, R. . (2023). Time-series forecasting of chlorophyll-a in coastal areas using lstm, gru and attention-based rnn models. *Journal of Environmental Informatics*.