Journal of Physics:
Conference Series



**PAPER** • **OPEN ACCESS**

# Cluster analysis of MNIST data set

To cite this article: Yuanzhao Pei and Linlin Ye 2022 *J. Phys.: Conf. Ser.* **2181** 012035



# Cluster analysis of MNIST data set

**Yuanzhao Pei[1], Linlin Ye[1*]**

[1]School of Applied Science and Technology, Hainan University, Danzhou, Hainan Province, China

*Corresponding author's e-mail: 2485205041@qq.com

**Abstract.** The full name of MNIST is Mixed National Institute of Standards and Technology database. This is a very large database of handwritten digits. The data set is divided into two parts: the first part is 60,000 training set images, the second part is 10,000 test set images, each image is composed of 28×28 pixels, and the value of each pixel between 0 and 1, based on this data set, not only can a judgment method for handwritten numbers be designed, but also the accuracy of the recognition algorithm can be compared through the test set. At present, machine learning experts around the world use different learning methods to test the data set. At present, the mainstream methods with higher accuracy are: support vector machines, convolutional neural networks, and naive Bayes classification. This article tries to use the clustering method to train the MNIST data set. First, the image data of the training set is converted into 60000×785 rows of two-dimensional matrix data, one of which is the real value of the image, and then through K-means the algorithm divides the data into 10 categories, uses the T-SNE algorithm to reduce the dimensionality, and visualizes by randomly selecting 1000 training data. It is found that the clustering effect of the numbers 0, 6, and 8 is the best, while the number 1 is completely invisible, the accuracy of K-means algorithm clustering is 83.33%. In order to further explore the test effects of different clustering methods on the data set, other clustering methods are introduced: MiniBatchKMeans algorithm. When the cluster is set to 10, the clustering of data 6 and 8 in MiniBatchKMeans the effect is good. At this time, the accuracy of the MiniBatchKMeans algorithm clustering is 87.67%. By comparing the test results of other algorithms, it is found that the MiniBatchKMeans algorithm is better than the K-means algorithms and the clustering result of is better than some supervised learning methods.

## 1. Introduction

The full name of MNIST is Mixed National Institute of Standards and Technology database[1]. This is a very large database of handwritten digits . The data set is divided into two parts: the first part is 60,000 images in the training set, and the second part is 10,000 images in the test set. Each image is composed of 28×28 pixels. These numbers have been standardized and are The fixed-size image (28x28 pixels) is the center. Based on this data set, not only can the judgment method of handwritten numbers be designed, but also the accuracy of the recognition algorithm can be compared through the test set. At present, machine learning experts around the world use different learning methods to test the data set. At present, the mainstream methods with higher accuracy are: support vector machines, convolutional neural networks, naive Bayes classification, etc.These are supervised learning methods. For unsupervised learning methods, there is no suitable algorithm to deal with the data set well. This paper attempts to test the data set by using classical clustering methods, and the final results are more objective the methods and accuracy of some scholars' processing of this data set are given below[2].

Table 1. Current scholars' methods and accuracy of processing this data set

| CLASSIFIER | PREPROCESSING | TEST ERROR RATE (%) | Reference |
|---|---|---|---|
| K-nearest-neighbors, Euclidean | deskewing | 2.4 | LeCun et al. 1998 |
| product of stumps on Haar f. | Haar features | 0.87 | Kegl et al., ICML 2009 |
| Virtual SVM deg-9 poly | none | 0.8 | LeCun et al. 1998 |
| 2-layer NN, 800 HU, MSE | none | 0.9 | Simard et al., ICDAR 2003 |
| Convolutional net Boosted LeNet-4 | none | 0.7 | LeCun et al. 1998 |
| pairwise linear classifier | deskewing | 7.6 | LeCun et al. 1998 |

**2.Symbol Description**

| Symbolic variable | Express meaning |
|---|---|
| $x_i$ , $x_j$ | Point in high-dimensional space |
| $d_{ij}$ | Euclidean distance of $x_i$ , $x_j$ |
| $p_{j\|i}, q_{j\|j}$ | $x_j$ is the probability of $x_i$ field |
| $SS_B$ is the between-class variance, $SS_W$ is the within-class variance | Between-class variance |
| $SS_W$ | Within-class variance |
| $c_E$ | The central point of all data |
| $n_q$ | Total number of data points |
| $m_i$ | Cluster marker array |
| $X$ | Data collection |

See the text for detailed symbol description.

## 3. Methods and Materials

### *3.1. K-means algorithm*

K-means algorithm is the most classic partition-based clustering method, and it is one of the ten classic data mining algorithms[3]. The basic idea of the K-means algorithm is: clustering around k points in the space, and classifying the objects closest to them. Through the iterative method, the value

of each cluster center is updated successively until the best clustering result is obtained. The k-means algorithm accepts the parameter k, and then divides the previously input n data objects into k clusters so that the obtained clusters are satisfied. The similarity of objects in the same cluster is higher, but the objects in different clusters The similarity is small. Cluster similarity is calculated by using the mean value of the objects in each cluster to obtain a "central object" (gravitation center) . The basic steps are:

**Step1:** Select the number k of categories to be clustered (k=3 categories in the above example), and select k center points.

**Step2:** For each sample point, find the center point closest to it (search for organization), and the point closest to the same center point is a class, thus completing a clustering.

**Step3:** Determine whether the categories of the sample points before and after the clustering are the same. If they are the same, the algorithm terminates, otherwise enter step4.

**Step4:** For the sample points in each category, calculate the center points of these sample points as the new center points of the category, and continue to step 2.

*3.2. t-SNE algorithm*

t-SNE (t-distributed stochastic neighbor embedding) is a machine learning algorithm for dimensionality reduction, which was proposed by Laurens van der Maaten and Geoffrey Hinton in 2008. t-SNE is a nonlinear dimensionality reduction algorithm, which is very suitable for dimensionality reduction of high-dimensional data to 2 or 3 dimensions for visualization[4]. In practical applications, t-SNE is seldom used for dimensionality reduction, mainly for visualization. The basic idea is that if two data are similar in a high-dimensional space, they should be very separated when they are reduced to a 2-dimensional space.

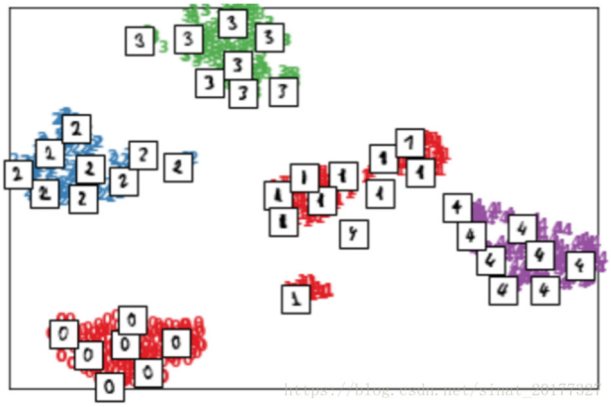

Figure 1. SNE algorithm visualization effect

K-means and t-SNE algorithms are used to train 600,000 data sets, and 1,000 data are randomly selected and reduced to 2 using T-SNE as the visualization result. Since the mnist data set has 10 types, the parameter cluster n_clusters of k-means is set to 10. It can be seen from the figure that, except for the number 1 which is clearly divided into 2 categories, the clustering results of other numbers are better. Among them, the clustering results of the four numbers of 0, 2, 6, and 8 have almost no other numbers. , The numbers 7,9,4 are mixed together because they are relatively similar in shape, and the two numbers 3 and 5 are also mixed together because they are relatively similar in shape. The specific clustering situation is as follows:

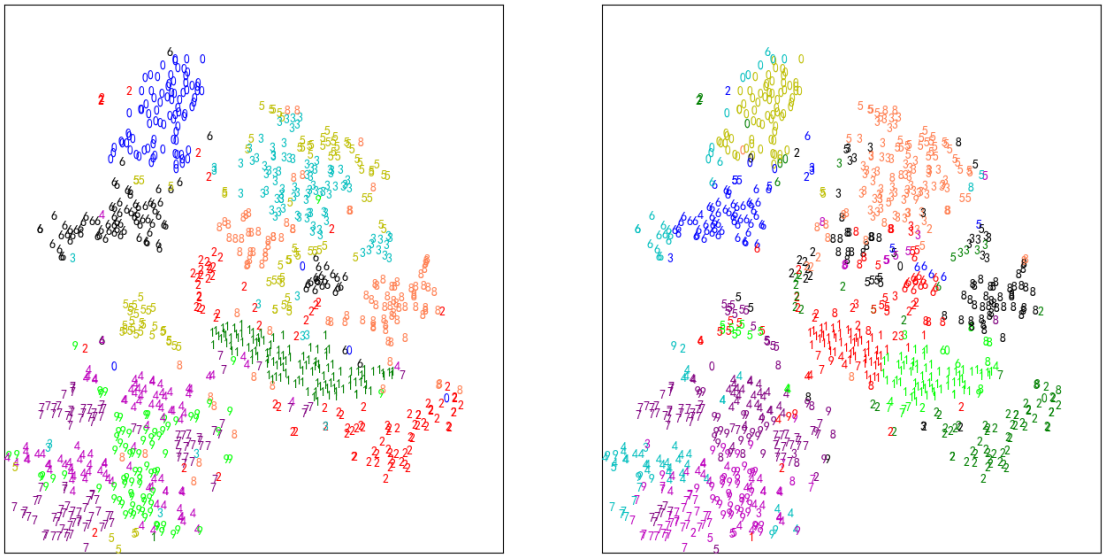

Figure 2. Comparison of original data before and after processing using K-means and t-SNE

In order to analyze the effect of the K-means algorithm on clustering when the cluster is 10, the following figure shows the correct number of 1000 pictures for each number (if the class contains more numbers, it is considered Numbers are correctly classified), the classification accuracy rate reaches 83.3%:

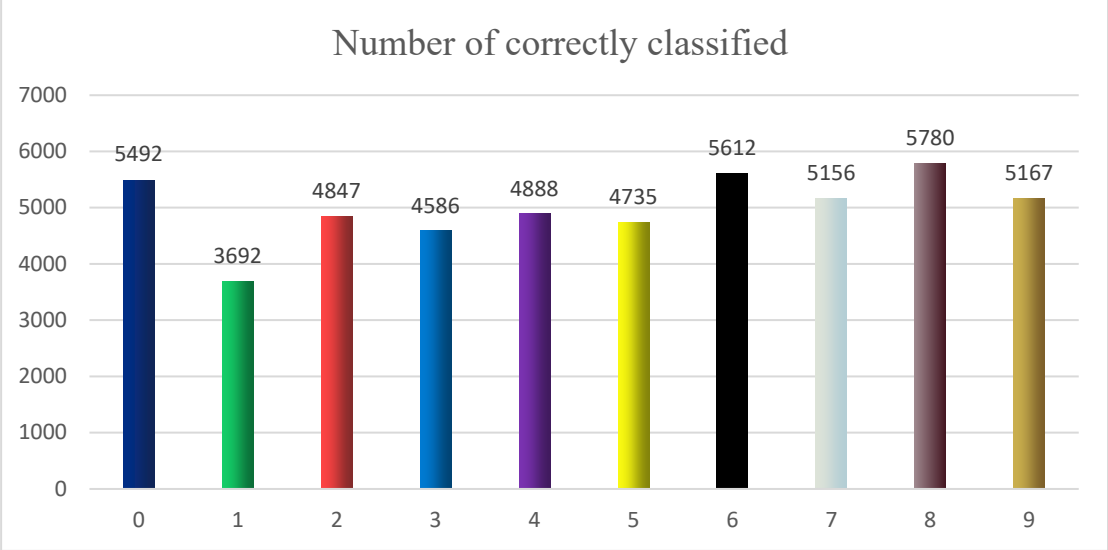


Figure 3. Using the K-means algorithm to correctly classify the number of each number

*3.3. Ablation experiment based on Calinski-Harabasz index*

The CH index measures the tightness within the class by calculating the sum of squares of the distances between each point in the class and the center of the class, and measures the separation of the data set by calculating the sum of the squares of the distances between various center points and the center of the data set. The CH index is determined by the degree of separation. The ratio to the tightness is obtained. Therefore, the larger the CH, the tighter the cluster itself, and the more scattered between the clusters, that is, the better clustering result. In this paper, the CH index can be used to judge the optimal number of clusters.

$$s = \frac{\frac{SS_B}{k-1}}{\frac{SS_W}{N-k}} \tag{1}$$

Where k represents the number of clusters, N represents the total number of data, $SS_B$ is the variance between clusters, $SS_W$ is the variance within clusters, and the calculation method of $SS_B$ is:

$$SS_B = tr(B_k) \tag{2}$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T \tag{3}$$

Trace only considers the elements on the diagonal of the matrix, that is, the Euclidean distance from all data points in the class $q$ to the class. The calculation method of $SS_W$ is:

$$SS_W = tr(W_k) \tag{4}$$

$$W_k = \sum_{q=1}^{k} \sum_{x_q} (x - c_q)(c_q - c_q)^T \tag{5}$$

Where $C_q$ is the collection of all data in class $q$, $C_q$ is the mass point of class $q$, $c_E$ is the center point of all data, and $n_q$ is the total number of data points of class $q$.

By changing the number of clusters to calculate the value of the CH indicator, the results of CH are as follows:

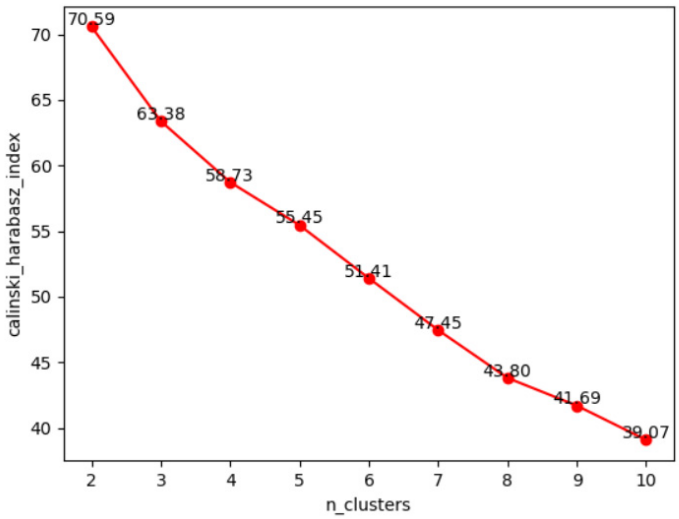


Figure 4. Selecting the CH value of different cluster numbers under the K-means algorithm

Select n_clusters from 2 to 10 to perform ablation experiments on Kmeans clusters. The CH index is that the larger the score, the better the clustering effect. The smaller the covariance of the intra-category data, the better, the larger the covariance between categories, the better, and the higher the CH index score will be. It can be seen from the figure that as the cluster n_clusters increases, the Calinski-Harabasz (CH) index score is lower, indicating that the K-means algorithm has the best clustering effect when the cluster is selected as 2 clusters. When the value is 10, the clustering effect of the K-means algorithm is the worst. This is contrary to the expected clustering effect of the K-means

algorithm when the cluster is 10, which also indirectly shows that it is based on K-means. It can be seen that the difference between these picture data is not so huge, of course, this may also be related to the selected distance index and other factors, so I will not discuss it here.

*3.4. Mini Batch K-Means Algorithm*

The Mini Batch KMeans algorithm is a clustering model that can maintain the accuracy of clustering as much as possible but can greatly reduce the calculation time . It uses a small batch of data subsets to reduce the calculation time while still trying to optimize the objective function. Mini Batch refers to a subset of data randomly selected each time the algorithm is trained. Using these randomly selected data for training greatly reduces the calculation time and reduces the convergence time of the KMeans algorithm , but it is better than The standard algorithm is slightly worse. It is recommended that when the sample size is greater than 10,000 for clustering, the Mini Batch KMeans algorithm needs to be considered. Due to the large data set, the algorithm is theoretically more suitable for learning from this data, The algorithm steps are as follows:

**Step1:** First extract part of the data set, and use the K-Means algorithm to build a model of K clustering points

**Step2:** Continue to extract part of the data set sample data in the training data, add it to the model, and assign it to the closest cluster center point

**Step3:** Update the center point value of the cluster (only use the extracted part of the data set for each update)

**Step4:** The second and third steps of loop iteration, know that the center point is stable or reach the number of iterations, stop the calculation operation.

Using Mini Batch K-Means and t-SNE algorithm to train 600000 data sets, and randomly select 1000 data and use T-SNE to reduce to 2 as the visualization result. Since the mnist data set has 10 types, the parameter cluster n_clusters of kmeans is set to 10. It can be seen from the figure that, except for the number 1 which is clearly divided into 2 categories, the clustering results of other numbers are better. Among them, the clustering results of the four numbers of 0, 2, 3, and 6 are almost free of other numbers. , The numbers 7,9,4 are still mixed together because they are relatively similar in shape, and the two numbers 5 and 8 are also mixed together because they are relatively similar in shape. The specific clustering situation is as shown in the figure below:

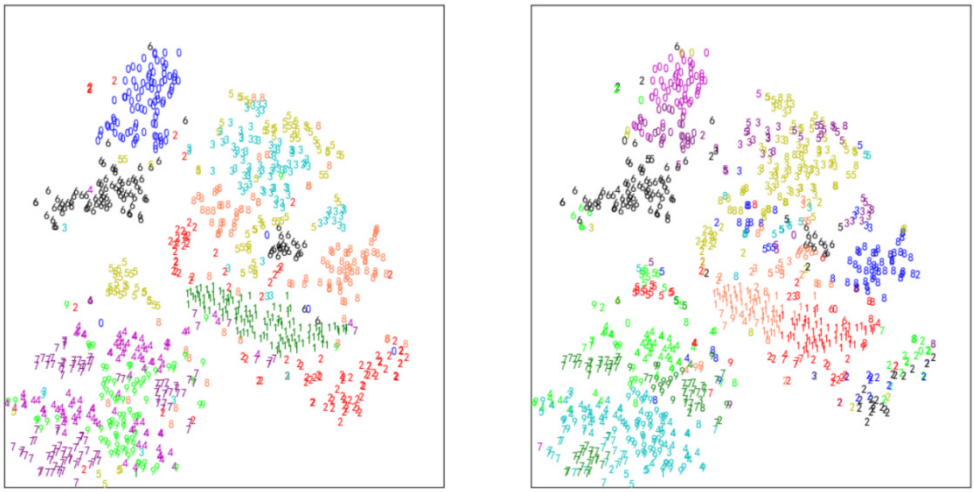

Figure 5. Comparison of original data before and after processing using Mini Batch KMeans and t-SNE

In order to analyze the effect of the Mini Batch K-Means algorithm on the clustering when the cluster is 10, the following figure shows the correct number of the 1000 pictures of each number (if the class contains more numbers, then It is believed that the number is correctly classified), and its

classification accuracy rate reaches 87.67%:

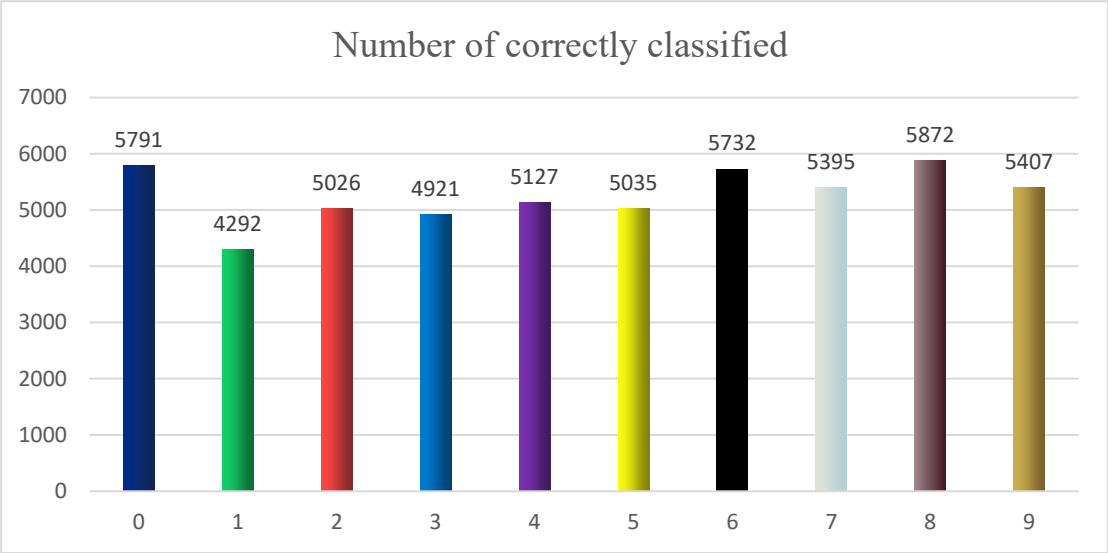


Figure 6. The number of correct classifications of each number using the Mini Batch KMeans algorithm

Select n_clusters from 2 to 10 to perform ablation experiments on Mini Batch K-Means clustering. The CH index is that the larger the score, the better the clustering effect. The smaller the covariance of the intra-category data, the better, the larger the covariance between categories, the better, and the higher the CH index score will be. It can be seen from the figure that as the cluster n_clusters increases, the Calinski-Harabasz (CH) index score is lower, indicating that when 2 or 3 clusters are selected for clustering, the Mini Batch K-Means algorithm has the best clustering effect. Well, when the cluster is 6, the clustering effect of Mini Batch K-Means algorithm is also better. You can make a cluster of 6 for a round of clustering. Of course, when the cluster is 10, the clustering of Mini Batch K-Means algorithm The effect is not ideal, and the reasons will not be discussed in detail.

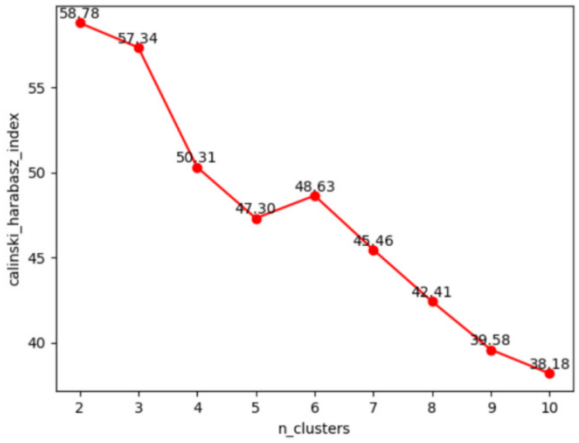


Figure 7. Selecting the CH value of different cluster numbers under the Mini Batch KMeans algorithm

## 4. Conclusions

This paper attempts to use the clustering algorithm to cluster MINST data sets. It is found that the relative part of this algorithm is unsupervised. When the set cluster is 10, the clustering accuracy of K-means algorithm is 83.33%. At this time, the clustering accuracy of minibatchkmeans algorithm is

87.67%. By comparing the test results of these two algorithms, it is found that minibatchkmeans algorithm is better. However, For today's mainstream unsupervised learning algorithms, the accuracy of the clustering algorithm used in this paper is still low. In order to improve the accuracy of the clustering algorithm, we can try DBSCN and other clustering algorithms.

**Acknowledgments**
The content filled in is not false and the content is not omitted. If there is any violation, we are willing to accept criticism and give up our rights to undertake the project. The project research results are marked as "Hainan University Research Fund Project (Project No.KYQD(ZR)-21082)".

**References**
[1] Deng L 2014 A tutorial survey of architectures, algorithms, and applications for deep learning APSIPA Transactions on Signal and Information Processing 3
[2] Wu Z, Shen C and van den Hengel A 2019 Wider or Deeper: Revisiting the ResNet Model for Visual Recognition Pattern Recognition 90 119–33
[3] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2015 Going deeper with convolutions 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
[4] He K, Zhang X, Ren S and Sun J 2016 Deep Residual Learning for Image Recognition 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)