学校代码： 10246
学　　号： 20210860084

Machine Learning Fundamentals
Week 11 Experimental Class

K-means implements image clustering

院　　系： 工程与应用技术研究院

姓　　名： 杨仲伟

学　　号： 20210860084

指 导 教 师： 范佳媛老师

完 成 日 期： 2020 年 12 月 11 日

Contents

I. Experiment Purpose

K-means implements image clustering.

II. Data Set Source

CIFAR-10 training set data, take 1/5 of the training set data to complete the clustering.

In fact, for performance reasons, only one-tenth of the data is used.

III. Implementation Tools

Platform and Notebook：Anaconda Navigator & JupyterLab

Programming Language：python 3.8

IV. Experiment Procedure

Step 1.

Import binary format data, visually display sample of 10 labels. 5 images are randomly selected for each label.

Step 2.

Use the K-means algorithm to achieve image clustering. The parameters include k and Lp norms, which represent clustering of images with different k values, and compare them with L1 and L2 norms.

Step 3.

Calculate the DBI, and use DBI to evaluate the clustering effect of the K-means algorithm when k and p are different.

Step 4.

Select the k with the minimum DBI as the optimal number of clusters. And choose the better Lp norm by comparing DBI.

Step 5.

Visualize the images after clustering. Five images are randomly selected for each cluster.

V. Data Processing Method

1. Download the CIFAR-10 data set from the official website and save it to the local path: D:\Projects\CIFAR-10DataSet\cifar-10-batches-py.

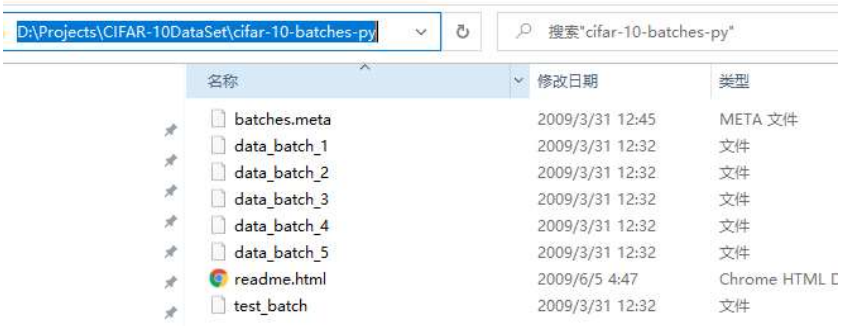


FIG. 1 Local path of data set binary files

2. Read binary files.

```python
# 一、读取二进制文件，获得数据集
def unpickle(file):
    # 导入pickle库将字节流转化为对象
    import pickle
    # 打开本地的二进制文件
    with open(file, 'rb') as fo:
        # 读取字节流，保存为dict格式
        dict = pickle.load(fo, encoding='bytes')
    # 返回dict
    return dict
```

FIG. 2 Python API of reading binary files

3. According to the data format given by the official website, segment the dictionary to obtain the images and labels.

VI. Algorithm

1. Clustering and k-means

Clustering is a kind of unsupervised learning. The goal is to explain the inherent properties and laws of the data through the learning of unlabeled training samples, and provide a basis for further data analysis.

By dividing the data set, a cluster structure is formed. And the experiment is using k-means algorithm to divide the data set.

Given the sample set D, the "k-means" algorithm minimizes the square error

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} \left|\left|x - \mu_i\right|\right|_2 \text{^} 2$$

for the cluster partition $C = \{C_1, C_2, \ldots, C_k\}$ obtained by clustering, where $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$ is the mean vector of the cluster C. This formula

describes the certain extent of how closely the samples in the cluster surround the cluster mean vector. The smaller the E value, the higher the similarity of the samples in the cluster.

k-means uses a greedy strategy to approximate this formula through iterative optimization. During the iterative update process, the distance between the samples in each cluster and the mean vector will be shorter. This distance can be calculated by different norms.

When the threshold of the adjustment range is reached, the iteration is stopped, and an approximately optimal cluster classification result is output. In this experiment, this threshold is set to 0.0001.

2. K-means algorithm

Input:

Data set: $D = \{x_1, x_2, \dots, x_m\}$

Number of clusters: k.

Lp norm: p (L1 or L2).

Process:

1: Randomly select k samples from D as the initial mean vector. $\{\mu_1, \mu_2, \dots, \mu_k\}$

2: **repeat**

3: $C_i = \emptyset (1 \leq i \leq k)$

4: **for** j = 1, 2, …, m **do**

5: Calculate the distance between the sample x_j and each mean vector $\mu_i(1 \le i \le k)$: $d_{ji} = \left|\left|x_j - \mu_i\right|\right|_p$;

6: Determine the cluster label of x_i according to the nearest mean vector: $\lambda_j = arg\ min_{i\in\{1,2,\ldots,k\}} d_{ji}$;

7: Divide sample x_j into corresponding clusters: $C_{\lambda_j} = C_{\lambda_j} \cup \{x_j\}$;

8: **end for**

9: **for** $i = 1,2,\ldots,k$ **do**

10: Calculate the new mean vector: $\mu_i' = \frac{1}{|C_i|}\sum_{x\in C_i} x$;

11: **if** $|\mu_i' - \mu_i| > 0.0001$ **then**

12: Update μ_i to μ_i'

13: **else**

14: Keep the μ_i not changed

15: **end if**

16: **end for**

17: **until** μ_i is not updated

Output:

Cluster $C = \{C_1, C_2, \ldots, C_k\}$

3. DBI indicator

For the clustering results, we need to evaluate its quality through some performance metric. This experiment uses the internal index DBI to measure the performance of clustering.

The idea of DBI is to hope that the sum of the average distances between samples in cluster C is as small as possible, and the distance between the center points of different clusters is as large as possible.

This means that the samples after cluster classification are more concentrated. So that the result of clustering has a higher "intra-cluster similarity" and a lower "inter-cluster similarity".

The DBI algorithm is as follows:

Davies-Bouldin Index:

$$DBI = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i}\left(\frac{avg(C_i) + avg(C_j)}{d_{cen}(C_i, C_j)}\right)$$

Average distance between samples in cluster C:

$$avg(C) = \frac{2}{|C|(|C|-1)} \sum_{1 \leq i < j \leq |C|} dist(x_i, x_j)$$

The distance between the center point of cluster C_i and cluster C_j:

$$d_{cen}(C_i, C_j) = dist(\mu_i, \mu_j)$$

$dist(\cdot,\cdot)$ is used to calculate the distance between two samples, and

The distance calculation here uses Euclidean distance, that is p=2;

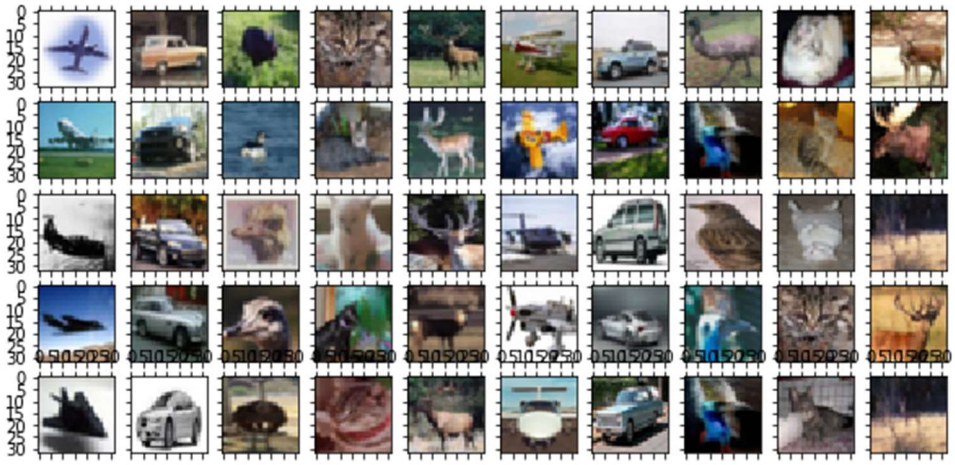
$$dist_{ed}(x_i, x_j) = \left|\left|x_i - x_j\right|\right|_2$$

μ represents the center point of cluster C: $\mu = \frac{1}{|C|}\sum_{1 \leq i \leq |C|} x_i$

VII. Experiment Results

Code running results display:

```
From label 0-9, show 5 images for each label from CIFAR-10DataSet in data_batch_1:
```



```
k is 6 ,norm Lp is L1
DBI of k-means algorithm is 8.640330
k is 6 ,norm Lp is L2
DBI of k-means algorithm is 8.323485
k is 7 ,norm Lp is L1
DBI of k-means algorithm is 9.319597
k is 7 ,norm Lp is L2
DBI of k-means algorithm is 8.573347
k is 8 ,norm Lp is L1
DBI of k-means algorithm is 9.050936
k is 8 ,norm Lp is L2
DBI of k-means algorithm is 8.592764
k is 9 ,norm Lp is L1
DBI of k-means algorithm is 9.379981
k is 9 ,norm Lp is L2
DBI of k-means algorithm is 8.588100
k is 10 ,norm Lp is L1
DBI of k-means algorithm is 9.390740
k is 10 ,norm Lp is L2
DBI of k-means algorithm is 9.290835
The minimum DBI : 8.323485
k of the minimum DBI : 6
Norm of the minimum DBI : L2 norm
The optimal clusters are shown below, and each column is a cluster:
```

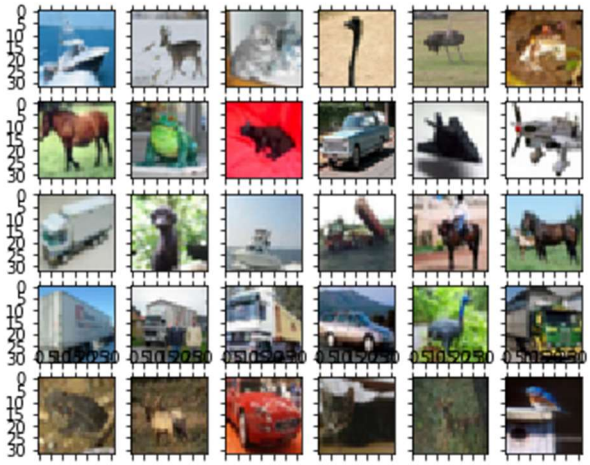


FIG. 3 Output of python code in JupyterLab

From the picture, the information shows here:

First, the test images are shown. Each label shows 5 images, there are 10 columns for 10 labels.

Then the information shows in order:

The different DBI when k is 6 to 10 and norm is L1 or L2.

The minimum DBI and the optimal cluster partition.

Finally, show the images of cluster with the minimum DBI. Each label shows 5 images, there are k columns for k labels.

Note: Since the initial mean vector is randomly selected, the DBI results are slightly different. And the clustering result is not influenced because clustering result of the minimum DBI is the same. When the initial mean vector is selected the same, the DBI results are the same.

VIII. Performance

1. Running time

The running time of the program is about 5 minutes, which only used 1000 samples of CIFAR-10 training set data.

The part of code using too much time is calculating the average distance between samples in cluster C when get the DBI. The time complexity is $O(n^2)$, and n is the number of samples of cluster C.

When using more than 10000 samples and different k, the running time is too long to be tolerated.

2. Memory space

The maximum list is arr_images[] that stores 10000 image samples

and the space is about 10000*32*32*3 = 30Mb.

IX. References

1. CIFAR-10 database download:

   http://www.cs.toronto.edu/~kriz/cifar .html

2. Machine Learning-Zhou Zhihua