JOURNAL OF SCIENCE
QUY NHON UNIVERSITY

Một kỹ thuật xây dựng tập dữ liệu huấn luyện dựa trên tiếp cận gom cụm

Lê Thị Kim Nga[1,*], Đinh Thị Mỹ Cảnh[1]

[1]Khoa Công nghệ thông tin, Trường Đại học Quy Nhơn

Ngày nhận bài: 27/03/2019; Ngày nhận đăng: 16/07/2019

TÓM TẮT

Đi cùng với quá trình phát triển các hệ thống nhận dạng đó là việc xây dựng những bộ dữ liệu huấn luyện không những cần thể hiện tốt về đối tượng được quan tâm mà còn cần hiệu quả, phù hợp với mô hình học máy được lựa chọn. Bài báo này trình bày một kỹ thuật xử lý lựa chọn tập dữ liệu theo tiếp cận gom cụm nhằm loại bỏ bớt những mẫu rất giống nhau. Kỹ thuật được cài đặt thử nghiệm để xây dựng tập dữ liệu đầu vào cho mô hình K-láng giềng gần nhất và đã chứng tỏ sự hiệu quả với nhiều bộ dữ liệu, cụ thể là dữ liệu sinh ngẫu nhiên theo phân phối chuẩn, bộ dữ liệu chữ số viết tay MNIST và bộ dữ liệu mặt YawDD.

Từ khóa: *Xây dựng mẫu huấn luyện, Thuật toán K-láng giềng gần nhất, Thuật toán K-means, nhận diện khuôn mặt, phát hiện khuôn mặt.*

[*] *Tác giả liên hệ chính.*
Email: kimnga78@gmail.com

TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC QUY NHƠN

A Technique For Training Data Selection Based On Clustering

Le Thi Kim Nga[1,*], Dinh Thi My Canh

[1]*Faculty of Information Technology, Quy Nhon University*

Received: 27/03/2019; Accepted: 16/07/2019

ABSTRACT

Along with the development of recognition systems, buiding training data sets not only needs to express well on the object of interest but also needs to be effective, consistent with the selected machine learning model. This article presents a processing technique for selecting data sets basing on clustering approach to reduce the very similar samples. This technology was installed, tested on trial to select input data for K-nearest neighbors model and proved its effectiveness with many data sets, namely the data generated randomly in standard distribution, MNIST database- data sets of handwritten digits and YawDD face data sets.

Keywords: *Training sample selection, K_Nearest Neighbors, K means, Face recognition, Face Detection*.

1. INTRODUCTION

The training data sets always play a vital role in every recognition system. For each system, the quality of recognition not only depends on the classification method and the parameter selection but also depends directly on the quality of the training data sets. Basically, learning of classification model is the corrective process of the model that is the best fit for input training data. The quality of the training data sets can be decreased because of many reasons such as the errors in data labeling, the imbalance of quantity and the diversity of observed patterns.[3]

For example, the implementation of the data collection for a face recognition system may appear many problems and challenges that result in the imbalance of the data sets. In general, the diversity of variants of the face image data depends on many factors including lighting, pose, individuals, facial expressions. In other words, personal factors are only a part of the face recognition system. In fact, two images of two people taken at the same time are more similar than that of a person taken at two different times.[2] Concern with this difference, when collecting a person's data at the same time, the face patterns are often very similar. Hence, a person can have the images of multiple faces but a lot of them are very similar. Multiple classifiers based on statistics, many patterns group similar can affect adversely learning result. For example, learning process based on minimum objective function is calculated by mean error of the samples, or case of K nearest neighbors algorithm, when the result was effected base of majority statistics, any sample is near to the similar group, then the high probability classification result will belong to that sample group. Other way, in some cases, a lot of similar samples have influence to the processing time and storage memory. Besides, because the conditions of meeting each one are

[*] *Corresponding author.*
Email: kimnga78@gmail.com

different that can lead to less data or more data of people.[5]

One way to handle data problems with similarity multiple data sets is to select a subset of patterns to learn. This can directly reduce the effect of the similar patterns. The approach to selecting this subset of data is also applied by some research groups in their specific problems. In a recent study, the authors analyze the effect of selecting data from the classifier Random Forest on remote sensing data on peat.[3] Abe author group used the Mahalanobis distance to estimate the boundary points to peed up the training of vector support machines by just learning these data.[1] Jigang Wang author group proposed two methods of data selection also serve for SVM learning[6]. The first method is based on similarity measure calculating by erecting (construct) spheres around the samples; the second method is based on Hausdorff distance, according that will use the minimum distance from the sample set to samples that belong to other class and use it as criteria for selected samples near boundary.

This paper presents the method for data processing that is based on clustering. The patterns with the same role will form a cluster that has a represented pattern. The design of the specific technique should be based on the method of recognition used since the nature of clustering is to create the data sets for identification. We applied the clustering technique that is based on K means algorithm. However, the distance thresholds for clustering have been used rather than the cluster numbers. The obtained results were tested with the K Nearest Neighbors method.

2. SOME RELATED RESEARCH

2.1. K_means

K_means is an important algorithm and used popular in cluster approach.[7] The main idea of the K_means algorithm is to classify objects that was put down into k cluster (K is the number of clusters that is predefined, K is a positive integer) so that the sum of the squares distance between objects and the group centroid is minimal.

The K means algorithm is described in the following figure:

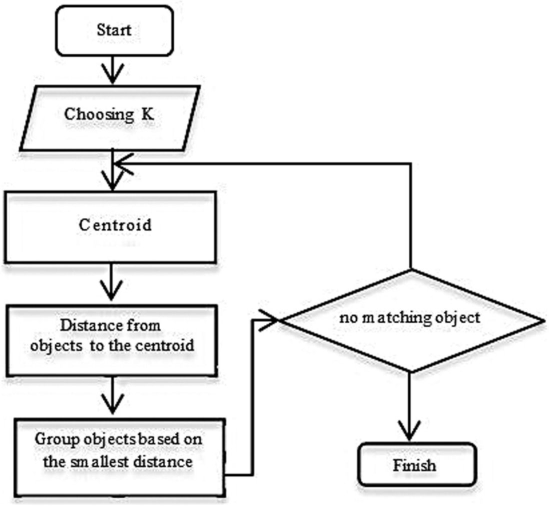
Start
Choosing K
Centroid
Distance from objects to the centroid
Group objects based on the smallest distance
no matching object
Finish

Fig 1. Describe the K means algorithm

The K means algorithm is implemented through the following steps:

1) Random selection K centroid for K cluster. Each centroid defines one of the cluster;

2) To calculate the distance between the objects and K centroid (offen use Euclidean distance);

3) Group the objects into the nearest group;

4) Update the new centroid for the groups;

5) Repeat step 2 until there are no group changes of objects.

The K means algorithm is applied in WeKa tool software (Waikato Environment for Knowledge Analysis), that is a machine learning software, developed by the University of Waikato New Zealand.

2.2. K_Nearest Neighbors

K_Nearest Neighbors (K_NN) is used common in the Data Mining. K_NN is a method for clustering objects based on the nearest distance between the objects need to be classified and all object in the training data set.

Each object is clustered based on its neighbors K. K is positive integer and prechosen when the algorithm is executed. User usually use Euclidean distance to calculate distance between objects.

The K_NN algorithm is described below:

6) Initial estimates for the K centroids (the number of Nearest Neighbors);

7) Calculate the distance between the object to be classified (Query Point) and all object in the training data set (often use Euclidean distance);

8) Sort the distance in ascending order and identify K-Nearest Neighbors to Query Point;

9) Takes all classes of K_Nearest Neighbors defined;

10) Based on the most class of K_Nearest Neighbors to determine the class for Query Point.

In the figure below, the training data set is represented by a plus sign (+) and a minus sign (–), the object needs to be defined for it (the Query Point) is a black circle. Our task is to estimate (or predict) the class of Query Point based on the selection number of the nearest neighbor to it. In other words, we want to know whether Query Point will be assigned to the class (+) or class (–):

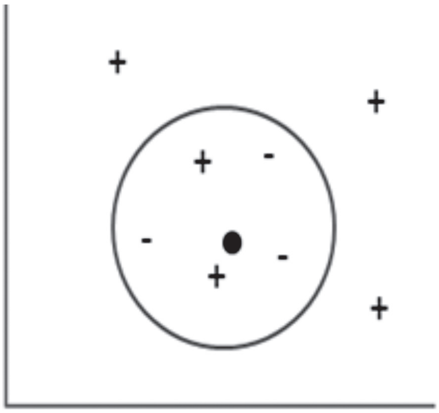

Fig 2. For example, the training data set

We see that:

1 Nearest Neighbors: The result is + (Query Point is assigned into class plus sign (+));

2 Nearest Neighbors: No class defined for Query Point because the number of the Nearest Neighbors of it is two, in that 1 is plus class and 1 is minus class (no class has more objects than the other);

5 Nearest Neighbors: Result is – (the Query Point is assigned into minus sign class because in 5 Nearest Neighbors of it, there are 3 objects belong to minis sign class, 2 object belong to plus sign class, the minis sign has more class than the plus sign).

3. ALGORITHM

Our method use to process local for each class in data set. Accordingly, each class of data will be clustered. Each cluster will be extracted a representative sample that was used to the result set. Thus, the data use to learn for each class will have the same number as clusters that is representative samples of classes. This method also toward the possibility of adding up online sample to the data set and serve the actual requirement is the samples may need to be gradually incorporated into the system data. Such as, input data set has many classes, we process each class with multiple clusters, and each cluster is a set of sample with a certain similarity to each other.

3.1. Construct each cluster

A cluster include a set of samples belong to a class. For the convenience in calculation 1 cluster, we construct set M. This set is described below:

1) If the cluster has 1 sample, M is empty.

2) If the cluster has 2 sample, M = {{m_{10}}}.

3) If the cluster has 3 sample, M = {{m_{10}}, {m_{20}, m_{21}}}.

4) If the cluster has 4 sample, M = {{m_{10}}, {m_{20}, m_{21}}, {m_{30}, m_{31}, m_{32}}}.

In there, m_{ij} is the distance between sample i and sample j.

InnerMetrics set allows to be constructed sequentially when it is incremented the sample into cluster, and cluster also to be constructed by incremented each sample. The cluster is represented by couple <U, M>, U is the sample set of cluster. The incremental sample process is described below:

```
Sign: enroll (s, <U, M>)
Input: sample s, cluster <U, M>
Output: new cluster <U, M>
Process:
size = |U|
U ← s
if size > 0 then
    V = empty
    foreach i in [0, size)
        V ← distance (s, $u_i$)
    endfor
    M ← V
endif
```

The representative sample of the cluster was chosen which have total distance to the remaining samples is minimal. The function calculates the total distance of a sample to the remaining samples as follows:

```
Sign: sumInner(idx, <U, M>)
Input: idx order of the sample, <U, M> cluster
Output: total distance value
Process:
sum = 0
foreach i in [0, idx)
    sum = sum + M[idx -1][i];
endfor
foreach i in [idx, |M|)
    sum = sum + M[i][idx];
endfor
return sum
```

3.2. Building class

Essentially of construct a class is create its clusters. The construction is also designed according to the criteria to incremented each sample into the class. New sample will be considered for put into the cluster already in the class. If we can't find it, we set up a new cluster with the only element that is the current sample. A sample is considered belong to the cluster if the sample is the most nearest to the cluster and the distance is less than a given threshold. The distance between the sample and the cluster is the distance of that sample to the representative of the cluster.

Sign C= {<U_0, M_0>, <U_1, M_1>,…, <U_{m-1}, M_{m-1}>}is current class and had m cluster.

The new admissions are described below:

```
Sign: enroll (s,C)
Input: sample s, class C
Output: new class C
Process:
clusID = – 1
minMetric = – 1
if |C| > 0then
    clusID =  0
    minMetric = distance (<$U_0$, $M_0$>, s)
    foreach i in [1, |C|)
        d = distance (<$U_i$, $M_i$>, s)
        if d < minMetricthen
            minMetric = d
            clusID = i
        endif
    endfor
endif
if clusID = – 1 OR minMetric>clusterThreshold then
    init (<$U_m$, $M_m$>)
    C ← <$U_m$, $M_m$>
    clusID = m
endif
enroll (s, <$U_m$, $M_m$>)
```

4. TESTING

The algorithm was tested to the data processing to make learning data for K_Nearest Neighbors classifier. The testing is also performed with threshold levels different cluster that will be get different levels of data selection. On the basis of conducting with many levels, we can compare the variance of the quality of classification, the test time corresponds to the sample rate used for training. The algorithm was tested in the different data set. That is randomized data, album of handwritten digits and YawDD surface video data set.

4.1. Randomized data

We were tested in randomized data based on standard distribution, the value on each dimension is created in [0,1]. Whereby, we were prechosen the dimension number of sample, the number of class will test and the sample number of each class. The data set generated that will be tested by cross-validation.

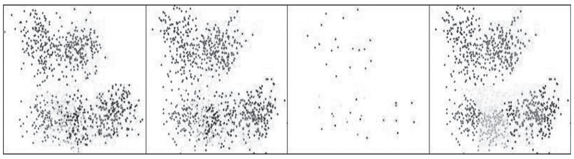

Fig 3. For test example, from left to right: the training data, the testing data, the sample set from the clusters, the result.

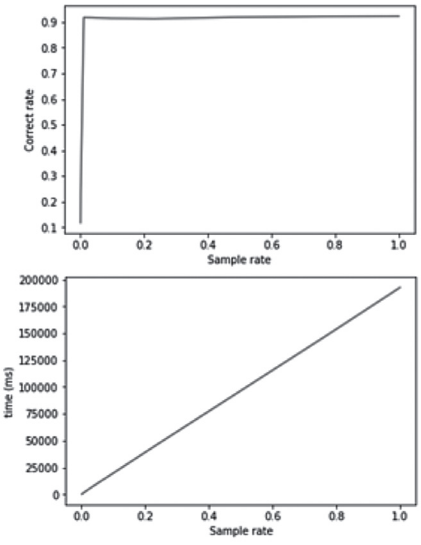

Fig 4. The testing in randomized data: cross-validation 2 fold, data generate 100 classes, each class has 2000 sample, characteristic length 4, K-Nearest Neighbors with k=9.

The experiment was carried out in a way of cross-validation 2 folds, 100 classes of data and 2000 samples per class. Both charts in accurately represent the prediction results by the suggested method (Fig 4). Because the implementation will need to calculate the distance to the samples in the data set, the processing time will increase linearly with the number of samples and more importantly, the idea of the algorithm has been confirmed when the accuracy increases very fast in a range of sample rates which closes to 0 and stable after that interval to 1. That is, we can use a small number of samples to produce quality equivalent to a large samples. In that case, time advantage will be evident on the time chart.

4.2. MNIST

MNIST is a database of handwritten digits that contains a training set of 60,000 examples and a test set of 10,000 examples^3. The MNIST database is commonly used for testing handwritten digits identification system, especially, classification technique use deep learning. In this test scope, techniques in the paper will be applied to the training data set and this result will be used to learning in K Nearest Neighbors. Thence, we evaluate the change of accuracy and testing time (Fig 5).

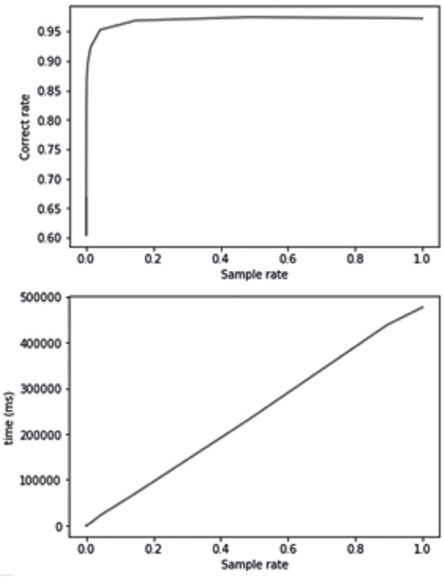

Fig 5. Testing with MNIST: K_Nearest Neighbors with k=3

The idea of the algorithm is also clearly shown with the experiment on MNIST and is similar to random data. Accuracy increases rapidly and reaches the highest level in a small range of sample rates while processing time still increases linearly with the sample rate. This test has also proved the judgment of the idea in the article.

4.3. YawDD

YawDD (Yawning Detection Dataset) is a video database that contains two data sets mirror and dash, capture face data of the driver with different facial features[4]. Capture data consisting of both male and female drivers, with and without glasses, from different ethnicities, mouth conditions such as normal, talking/singing, and yawning while driving. The video are taken in varying illumination conditions. This data set includes 349 video of 57 men and 50 woman, resolution of 640×480, 30 frames per second, AVI format without audio. Testing is done on the mirror data set, consisting of 47 man and 43 women, the total is 320 video. For each video, we extract each facial image region in the frames and regroup to make the testing data. The face was detected by Haar Adaboost technique and then the face data regions were extracted based on AAM technique[2,5]. Later, we have a data set of 90 people with 65595 samples, a length of each sample is 4573. That data set was tested with the K Nearest Neighbors technique based on cross-validation. Thus, techniques in the paper will also be applied to the training data set and its result used to learning data for the K Nearest Neighbors, on that basis, we evaluate the change of a accuracy and testing time.

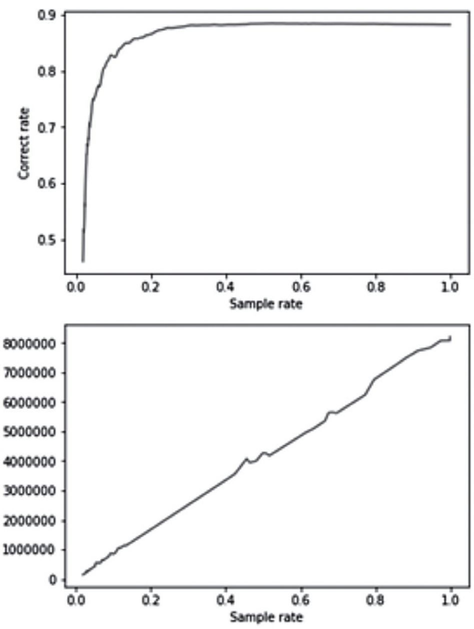

Fig 6. Testing with YawDD: cross-validation 2 fold, the K_Nearest Neighbors with k=3

In Fig 6, it is easy to see the similarity of the two charts in the experiment with YawDD with previous tests. This also once again demonstrates the idea of the algorithm. By these tests, we have absolutely the basis to build an identifier with a small enough sample number but still achieve the same accuracy with a large data set with reasonable sample selection. This will create a discrete time advantage for the identification system.

From the tests, it can be seen that the test time always varies with the sample rate used for the training. It is true that the nearest K neighbor technique is performed on the basis of a comparison of each sample and the nearest statistic. In addition, in the diagrams depicting the correlation between the precision and the sample rate used. It can be seen that even with many levels of sample usage, there is no significant change in accuracy when compared to using full data to learn. Thus, with the tried-and-tested data sets, according to the approach of the proposed technique and the optimum sampling rate, we can significantly reduce the test time as well as the memory used for storage.

5. CONCLUSION

In fact, building a recognition system involves many steps in which building a training data set is one of the decisive steps. Building a training data set not only needs to well representation observed variants of the object of interest but also needs to avoid duplication and needs to be consistent with the machine learning method used. This paper presents the method for data processing based on clustering. This method was designed to be able to add online patterns to a data set, serving the actual demand of gradually adding patterns to the system data. The technique has been tested and demonstrated to be effective with the K_Nearest Neighbors model.

REFERENCES

1. Abe, S., & Inoue, T. *Fast training of support vector machines by extracting boundary data*, In International Conference on Artificial Neural Networks, Springer, Berlin, Heidelberg, 2001, 308-313.

2. Cootes, T. F., Edwards, G. J., & Taylor, C. J. Active appearance models, *IEEE Transactions on pattern analysis and machine intelligence*, **2001,** *23(6),* 681-685.

3. Millard, K., & Richardson, M. On the importance of training data sample selection in random forest image classification: A case study in peatland ecosystem mapping, *Remote sensing,* **2015**, *7*(7), 8489-8515.

4. S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri. "YawDD: A Yawning Detection Dataset", *Proc. ACM Multimedia Systems, Singapore,* **2014**, 24-28.

5. Viola, P., & Jones, M. Rapid object detection using a boosted cascade of simple features, In Computer Vision and Pattern Recognition, *Proceedings of the 2001 IEEE Computer Society Conference,* **2001**, 1, I-I.

6. Wang, J., Neskovic, P., & Cooper, L. N. *Training data selection for support vector machines,* In International Conference on Natural Computation, Springer, Berlin, Heidelberg, **2005**, 554-564.

7. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition", *Proceedings of the IEEE,* **November 1998**, *86*(11), 2278-2324.