Image Compression Using Partitioning Around Medoids Clustering Algorithm

[1]V.B. Nikam [2]Vinod J. Kadam, , [3]B. B. Meshram

[1]Research Scholar, Department of Computer Technology,

[2]Assistant Professor, Department of Information Technology,
Dr.Babasaheb Ambedkar Technological University, Lonere,(Maharashtra)

[3]Professor, Department of computer Technology

[2,3]Veermata Jijabai Technological Institute, Mumbai, (Maharashtra).

Abstract

Clustering is a unsupervised learning technique. This paper presents a clustering based technique that may be applied to Image compression. The proposed technique clusters all the pixels into predetermined number of groups and produces a representative color for each group. Finally for each pixel only clusters number is stored during compression. This technique can be obtained in machine learning which one of the best methods for clustering is. The *k*-medoids algorithm is a clustering algorithm related to the K-means algorithm and the medoidshift algorithm.

Keywords: *Data Mining, K-medoids clustering, Machine learning, Image compression,*

1. Introduction

Clustering in data mining is a discovery process that groups similar objects into the same cluster[1]. Various clustering algorithms have been designed to fit various requirements and constraints of application[1]. Machine learning has been applied to many problems, and has demonstrated their superiority over classical methods when dealing with noisy or incomplete data. One such application is for data compression. A useful tool for determining *k* is the silhouette. Silhouette refers to a method of interpretation and validation of clusters of data. The technique provides a succinct graphical representation of how well each object lies within its cluster. It was first described by Peter J. Rousseeuw in 1986[2]. It is more robust to noise and outliers as compared to K-means because it minimizes a sum of dissimilarities instead of a sum of squared Euclidean distances. This clustering seem to be well suited to this particular function, as they have an ability to preprocess input patterns to produce simpler patterns with fewer components . This compressed information (stored in a hidden layer) preserves the full information obtained from the external environment. The compressed features may then exit the network into the external environment in their original uncompressed form.

For data compression, the image or data is broken down into smaller vectors for use as input. For each input vector presented, the Euclidean distances to all the output points are computed. The weights of the point with the minimum distance, along with its neighboring points are adjusted. This ensures that the outputs of these points are slightly enhanced. This process is repeated until some criterion for termination is reached. After a sufficient number of input vectors have been presented, each output point becomes sensitive to a group of similar input vectors, and can therefore be used to represent characteristics of the input data. This means that for a very large number of input vectors passed into the model,(uncompressed image or data), the compressed form will be the data exiting from the output points of the model (considerably smaller number). This compressed data may then be further decompressed by another model.

2. The Proposed Method

The basic concept of this method is as follows:

1. Clustering all the pixels in to predetermined number of groups.
2. Producing a representative color for each group.
3. For each pixels storing only cluster number during compression.
4. During decompression restoring cluster number and storing representative color of that cluster.

A brief description of the method is shown in Fig.1.

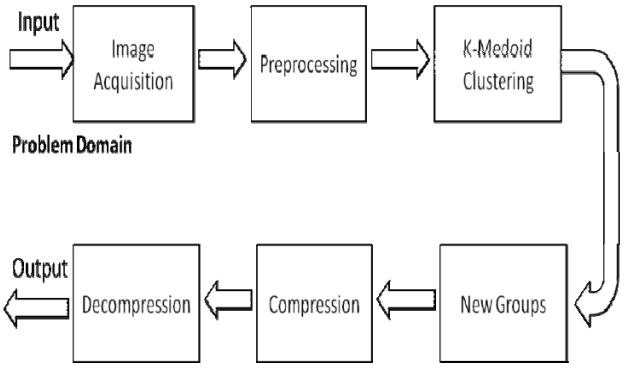

Fig.1 Design Steps for Image Compression

Step1: Image Acquisition and Preprocessing : These are preliminary steps required to feed input to the network. Problem domain is set of images received from user as a part of image acquisition. Preprocessing is required to remove noise from images using low pass filter.

Step2: Clustering : Since the basic colors are Red, Green and Blue, each vector have three parts as Red , Green and Blue quantity. Total number of feature in input pattern will be exactly three. For particular pixel input feature one will accept amount of Red quantity, feature two will accept amount of Green quantity and , feature three will accept amount of Blue quantity .The number of groups in the model will vary based on total number of clusters eg., if number of cluster required is 256 then we will have 256 (k*) representatives.

1. Initialize: randomly select (*k**) of the *total* data points as the medoids
2. Associate each data point to the closest medoid. ("closest" here is defined using any valid distance metric, most commonly Euclidean distance, Manhattan distance or Minkowski distance). We here used Euclidean Distance.
3. For each medoid *m*
    1. For each non-medoid data point *o*
        1. Swap *m* and *o* and compute the total cost of the configuration
4. Select the configuration with the lowest cost.
5. repeat steps 2 to 5 until there is no change in the medoid.

Step3: Compression : During this process firstly medoid details obtained in previous step are stored ie., cluster number of closest medoid is stored.

3. Analysis

In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points that one would measure with a ruler, and is given by the Pythagorean formula[3]. By using this formula as distance, Euclidean space (or even any inner product space) becomes a metric space. The associated norm is called the Euclidean norm. In one dimension, the distance between two points on the real line is the absolute value of their numerical difference[3]. Thus if *x* and *y* are two points on the real line, then the distance between them is computed as

$$\sqrt{(x-y)^2} = |x-y|.$$

In one dimension, there is a single homogeneous, translation-invariant metric (in other words, a distance that is induced by a norm), up to a scale factor of length, which is the Euclidean distance. In higher dimensions there are other possible norms. [3]In three-dimensional Euclidean space, the distance is

$$d(p,q) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + (p_3-q_3)^2}.$$

In our example, Here P1 red feature, P2 green feature and P3 is Blue feature. The representative who will give minimum d(p,q) will be the winner. A medoid can be defined as the object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the given data set[4].

If *j* is the medoid of Cluster *C*, the average distance of all objects of *C* to *j* is calculated using the formula

$$\frac{\sum_{i \in C} d_{ij}}{N_j}$$

where *N* is the number of object *minus* other than *j*. PAM is a partitioning method which operates on a distance matrix. The core objective of a learner is to generalize from its experience.[5] Medoids are representative objects of a data set or a cluster with a data set whose average dissimilarity to all the objects in the cluster is minimal. Medoids are similar in concept to means or centroids, but medoids are always members of the data set. Medoids are most commonly used on data when a mean or centroid cannot be defined such as 3-D trajectories or in the gene expression context[6]. Note that a medoid is not equivalent to a median or a geometric median. A median is only

defined on 1-dimensional data, and it only minimizes dissimilarity to other points for a specific distance metric (Manhattan norm). A geometric median is not necessarily a point from within the original dataset[6]. This algorithm basically works as follows. First, a set of medoids is chosen at random. Second, the distances to the other points are computed. Third, data are clustered according to the medoid they are most similar to. Fourth, the medoid set is optimized via an iterative process.

Fig.2 Output of Algorithm with PAM K=12

4. Conclusions

In this paper we have proposed a method for Image Compression using a K-medoids clustering. A superior training time and compression could be achieved with our method. It is more robust, because it minimizes a sum of dissimilarities instead of a sum of squared Euclidean distances.

References

[1] Shu-Chuan Chu, John F. Roddick and J. S. Pan, “An Efficient K -Medoids-Based Algorithm Using Previous Medoid Index, Triangular Inequality Elimination Criteria, and Partial Distance Search” , Data Warehousing and Knowledge Discovery, Lecture Notes in Computer Science, 2002, Volume 2454/2002, 301-311, DOI: 10.1007/3-540-46145-0_7.

[2] Peter J. Rousseeuw, "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis", Computational and Applied Mathematics **20**: 53–65, (1987).

[3] Elena Deza, Michel Marie Deza, “Encyclopedia of Distances”, page 94, (2009), Springer.

[4] Sergios Theodoridis, Konstantinos Koutroumbas, Pattern Recognition 3rd ed.. (2006), pp. 635.

[5] Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer ISBN 0-387-31073-8, (2006).

[6] Van Der Lann, Mark J; Pollard, Katherine S; Bryan, Jennifer; E, "A New Partitioning Around Medoids Algorithm". Journal of Statistical Computation and Simulation (Taylor & Francis Group) **73** (8): 575–584, (2003).

Vinod J Kadam is a faculty in Information Technology Deaprtment at Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad District, Maharashtra, India. He Holds BE, and MTech in Information Technology. He has seven years of academic experience. His research interests are in the domain of Data Mining, Image Processing.

Valmik B Nikam is a Ph.D. candidate in the Department of Computer Technology at Veermata Jijabai Technological Institute, Matunga, Mumbai. He holds B.E. in Computer Science and Engineering from Government College of Engineering Aurangabad and M.E. in Computer Technology from Veermata Jijabai Technological Institute, Mumbai. He worked at Dr. Babasaheb Ambedkar Technological University, Lonere for several years before joining PhD program. His research interests are data mining, scalable computing, image processing. He is a member of CSI, ACM and IEEE.

B.B.Meshram is a Professor and Head of Department of Computer Technology Department of Veermata Jijabai Technological Institute, Matunga, Mumbai. His current research includes database technologies, data mining, securities, forensic analysis, video processing, distributed computing. His has authored over 150 research publications. He has given numerous invited talks at various conferences, workshops, training programs and also served as chair/co-chair for many conferences/workshops in the area of computer science and engineering.