IMAGE SEGMENTATION USING K-MEANS CLUSTERING ALGORITHM

Ankit Bisla, 2012UCP1684   Peeyush Yadav, 2012UCP1687

Abstract— Image segmentation is the division of an image into regions or categories, which correspond to different objects or parts of objects. Every pixel in an image is allocated to one of a number of these categories. Clustering analysis method is one of the main analytical methods in data mining, the method of clustering algorithm will influence the clustering results directly. Data Clustering refers to the unsupervised classification of a dataset into groups, such that elements within a group are similar to each other. Data clustering is a data exploration technique that allows objects with similar characteristics to be grouped together in order to facilitate their further processing. This paper discusses the standard k- means clustering algorithm and analyzes the shortcomings of standard k- means algorithm, such as the k-means clustering algorithm has to calculate the distance between each data object and all cluster centers in each iteration, which makes the efficiency of clustering is not high. This paper proposes an improved k-means algorithm in order to solve this question, requiring a simple data structure to store some information in every iteration, which is to be used in the next interation. The improved method avoids computing the distance of each data object to the cluster centers repeatly, saving the running time. Experimental results show that the improved method can effectively improve the speed of clustering and accuracy, reducing the computational complexity of the k-means.

Keywords- clustering analysis; k-means algorithm; distance; computational complexity, filtering algorithm

Introduction - Clustering is the process of organizing data objects into a set of disjoint classes called clusters. Clustering is an example of unsupervised classification. Classification refers to a procedure that assigns data objects to a set of classes. Unsupervised means that clustering does not depends on predefined classes and training examples while classifying the data objects. Cluster analysis seeks to partition a given data set into groups based on specified features so that the data points within a group are more similar to each other than the points in different groups . Therefore, a cluster is a collection of objects that are similar among themselves and dissimilar to the objects belonging to other clusters. Clustering is an crucial area of research, which finds applications in many fields including bioinformatics, pattern recognition, image processing, marketing, data mining, economics, etc.

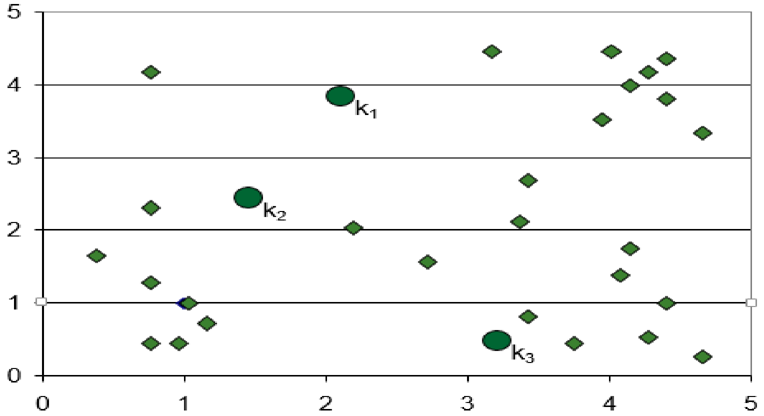
Cluster analysis is a one of the primary data analysis tool in the data mining. Clustering algorithms are mainly divided into two categories: Hierarchical algorithms and Partition algorithms. A hierarchical clustering algorithm divides the given data set into smaller subsets in hierarchical fashion. A partition clustering algorithm partition the data set into desired number of sets in a single step.

K-Means Clustering

The objective of the *K*-means clustering algorithm is to divide an image into *K* segments minimizing the total within-segment variance. The variable *K* must be set before running the algorithm. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. Thus, the purpose of K-mean clustering is to classify the data.

1. Initialize cluster centers

The first step of algorithm is to initialize k random cluster centers z1, z2,...., zk from the input data x1, x2,....,xn.

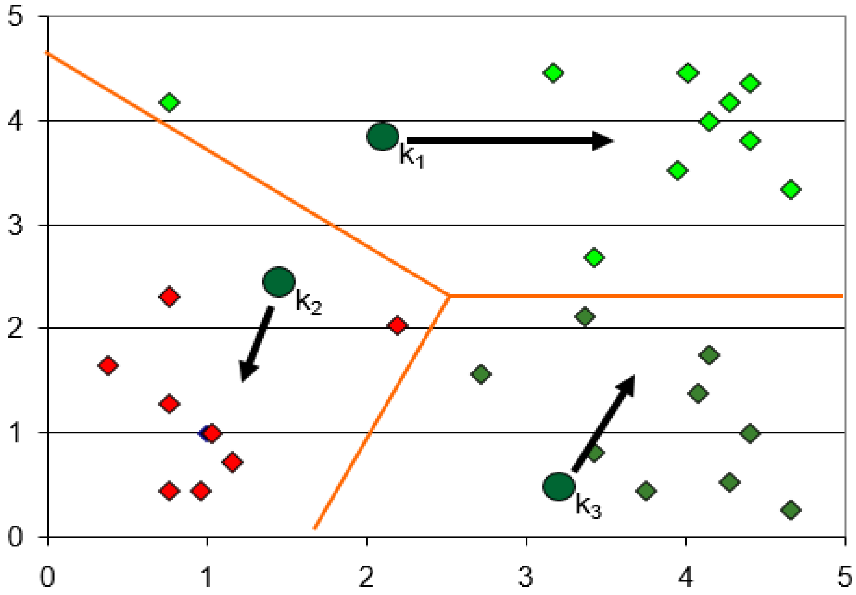


Input data dimension - n_samples x n_features
Cluster Centers dimension - n_clusters x n_features

2. Assign labels to input data

In this step distance of all the cluster centers z_1, z_2,...., z_k initialized in last step from each input data point x_1, x_2,....,x_n is computed and stored in a dist matrix. After computing this matrix, each data point is assigned[4] a cluster center whose distance from this data point is least and is stored in a labels vector. An algorithm for partitioning (or clustering) N data points into *K* disjoint subsets S_j containing N_j data points so as to minimize the sum-of-squares criterion

dist dimension - n_samples x n_clusters
labels dimension - n_samples x 1

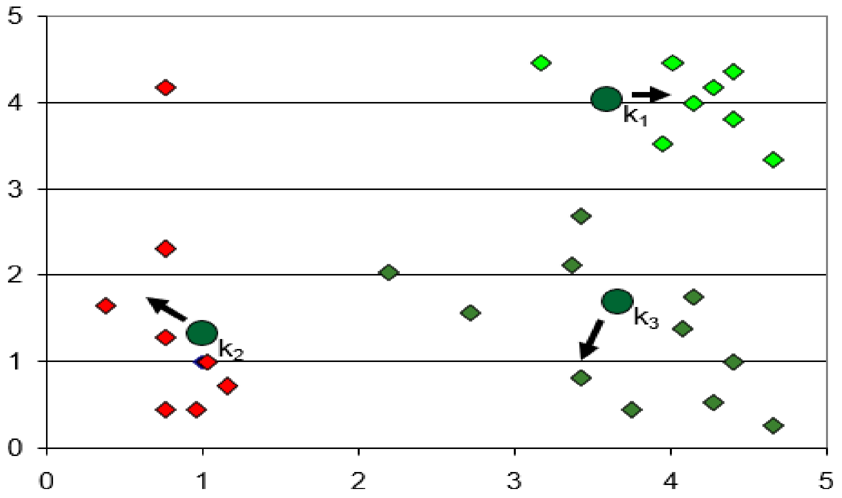


Mathematically, this step can be formulated as

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} \|x_n - \mu_j\|^2,$$

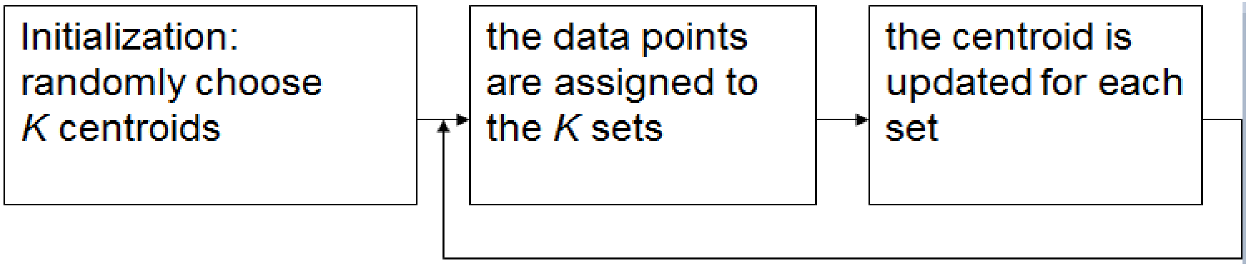
3. Update the cluster centers

Compute the new cluster centers z1', z2'... zk' by taking mean of the data points that belong to that particular cluster center [4]. Mathematically, this step can be formulated as

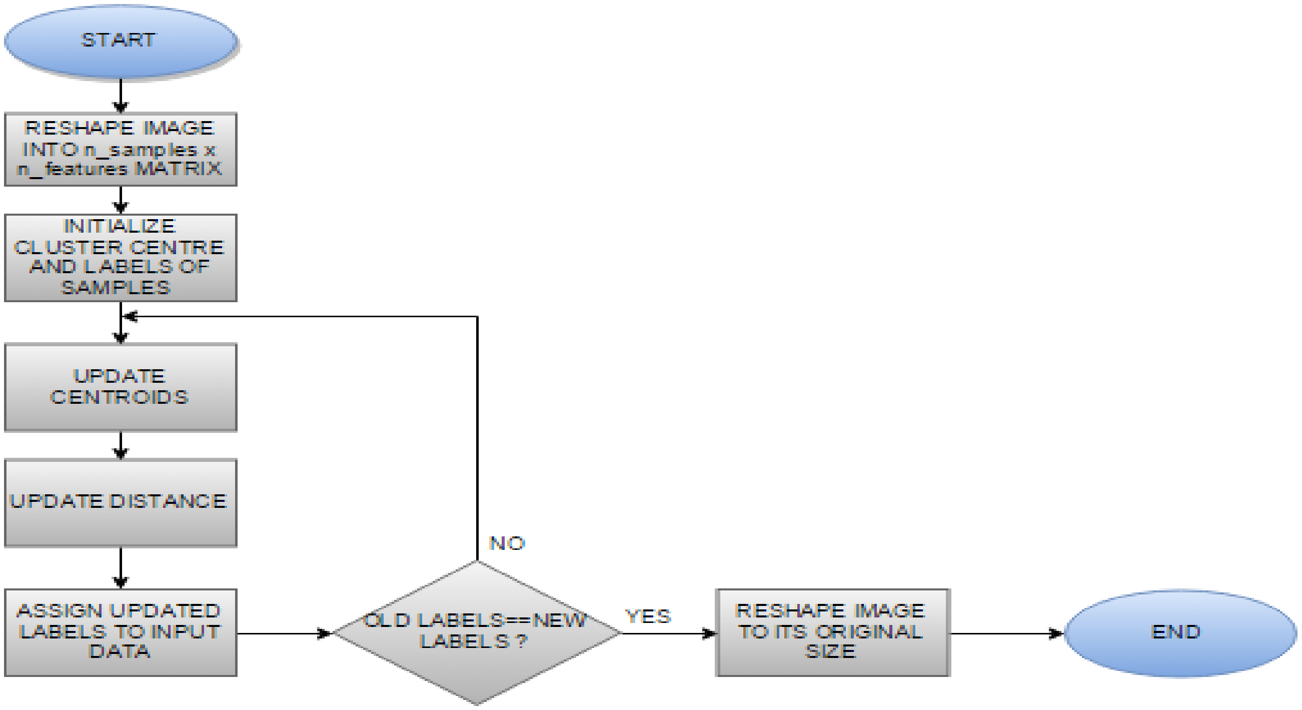


4. Termination

Repeat step 2 and step 3 until the algorithm converges i.e. the difference between newly assigned labels and older labels is less than a particular tolerance value or until the maximum number of iterations have been reached.


Initialization: randomly choose *K* centroids
the data points are assigned to the *K* sets
the centroid is updated for each set

Flow Chart:


START
RESHAPE IMAGE INTO n_samples x n_features MATRIX
INITIALIZE CLUSTER CENTRE AND LABELS OF SAMPLES
UPDATE CENTROIDS
UPDATE DISTANCE
ASSIGN UPDATED LABELS TO INPUT DATA
OLD LABELS==NEW LABELS ?
NO
YES
RESHAPE IMAGE TO ITS ORIGINAL SIZE
END

Improved k-means clustering algorithm

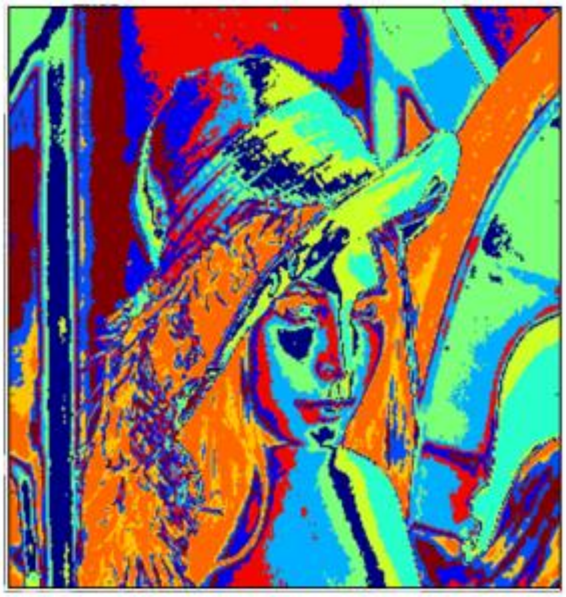
The standard k-means algorithm needs to calculate the distance from the each date object to all the centers of k clusters when it executes the iteration each time, which takes up a lot of execution time especially for large-capacity databases. For the shortcomings of the above k-means algorithm. The main idea of algorithm is to set two simple data structures to retain the labels of cluster and the distance of all the date objects to the nearest cluster during the each iteration, that can be used in next iteration, we calculate the distance between the current date object and the new cluster center, if the computed distance is smaller than or equal to the distance to the old center, the data object stays in it's cluster that was assigned to in previous iteration. Therefore, there is no need to calculate the distance from this data object to the other k- clustering centers, saving the calculative time to the k-cluster centers. Otherwise, we must calculate the distance from the current data object to all k cluster centers, and find the nearest cluster center and assign this point to the nearest cluster center and then we seperately record the label of nearest cluster center and the distance to it's center. Because in each interation some
data points still remain in the original cluster, it means that some parts of the data points will not be calculated, saving a total time of calculating the distance, thereby enhancing the efficiency of the algorithm.

The Filtering Algorithm

This algorithm begins by storing the data points in a kd-tree. Recall that, in each stage of Lloyd's algorithm, the nearest center to each data point is computed and each center is moved to the centroid of the associated neighbors. The idea is to maintain, for each node of the tree, a subset of candidate centers. The candidates for each node are pruned, or filtered, as they are propagated to the node's children. Since the kd-tree is computed for the data points rather than for the centers, there is no need to update this structure with each stage of Lloyd's algorithm. For each node of the kd-tree, we maintain a set of candidate centers. This is defined to be a subset of center points that might serve as the nearest neighbor for some point lying within the associated cell. The candidate centers for the root consist of all k centers. We then propagate candidates down the tree as follows: For each node u, let C denote its cell and let Z denote its candidate set. First, compute the candidate z* that belongs to Z that is closest to the midpoint of C. Then, for each of the remaining candidates z belongs to Z, if no part of C is closer to z than it is to z*, we can infer that z is not the nearest center to any data point associated with u and, hence, we can prune, or filter, z from the list of candidates. If u is associated with a single candidate (which must be z*) then z* is the nearest neighbor of all its data points. We can assign them to z* by adding the associated weighted centroid and counts to z*. Otherwise, if u is an internal node, we recurse on its children. If u is a leaf node, we compute the distances from its associated data point to all the candidates in Z and assign the data point to its nearest center.

Result:

Input Image



Segmented Image

CONCLUSION:

We have presented an efficient implementation of Lloyd's k-means clustering algorithm, called the filtering algorithm. The algorithm is easy to implement and only requires that a kd-tree be built once for the given data points. Efficiency is achieved because the data points do not vary throughout the computation and, hence, this data structure does not need to be recomputed at each stage. Since there are typically many more data points than ªqueryº points (i.e., centers), the relative advantage provided by preprocessing in the above manner is greater. This proposed method finding the better initial centroids and provides an efficient way of assigning the data points to the suitable clusters. This method ensures the total mechanism of clustering in O(nlogn) time without loss the correctness of clusters. This approach does not require any additional inputs like threshold values.

REFERENCES:

1. Ujjwal Maulik, Sanghamitra Bandyopadhyay, Genetic algorithm-based clustering technique, Pattern Recognition 33 (2000) 1455-1465.

2. Kumara Sastry, David Goldberg, Graham Kendall, “Genetic Algorithms,” in SEARCH METHODOLOGIES, Springer, 2005, ch. 4, pp. 99–125.

3. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, “Region-Based Convolutional Networks for Accurate Object Detection and Segmentation”, IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 1, January 2016

4. Tapas Kanungo, Nathan S. Netanyahu, Angela Y. Wu, David M. Mount, Christine D. Piatko, Ruth Silverman, “An Efficient k-Means Clustering Algorithm: Analysis and Implementation”, IEEE transactions on pattern analysis and machine intelligence, vol. 24, no. 7, July 2002

5. Xiang-Yang Wang, , Zhi-Fang Wu, Liang Chen, Hong-Liang Zheng, ,Hong-Ying Yang “Pixel classification based color image segmentation using quaternion exponent moments” in Neural Networks Volume 74, February 2016, Pages 1–13

6. Gonzalez,Rafael C., Richard E.Woods” Image Segmentation” in Digital Image Processing Prentice Hall , New Jersey 07458, pp567-643