Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Two-level k-means clustering algorithm for k–τ relationship establishment and linear-time classification

Radha Chitta [a,]*, M. Narasimha Murty [b]

[a] Department of Electrical Engineering, Indian Institute of Science, Bangalore 560012, India
[b] Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560012, India

ARTICLE INFO

Article history:
Received 12 January 2009
Received in revised form
25 August 2009
Accepted 22 September 2009

Keywords:
Clustering
k-Means
Classification
Linear-time complexity
Support vector machines
k-Nearest neighbor classifier

ABSTRACT

Partitional clustering algorithms, which partition the dataset into a pre-defined number of clusters, can be broadly classified into two types: algorithms which explicitly take the number of clusters as input and algorithms that take the expected size of a cluster as input. In this paper, we propose a variant of the k-means algorithm and prove that it is more efficient than standard k-means algorithms. An important contribution of this paper is the establishment of a relation between the number of clusters and the size of the clusters in a dataset through the analysis of our algorithm. We also demonstrate that the integration of this algorithm as a pre-processing step in classification algorithms reduces their running-time complexity.

1. Introduction

Clustering, the process of grouping similar patterns together, finds application in pattern analysis, decision making and various machine learning problems. Clustering algorithms [1] can be broadly classified into hierarchical and partitional clustering algorithms. Hierarchical algorithms, as the name suggests, produce a hierarchy of clusters. Agglomerative hierarchical clustering algorithms initially treat each pattern as a separate cluster and merge clusters recursively based on their distance from each other until a single cluster containing all the patterns is obtained. Divisive hierarchical algorithms start with a single cluster containing all the patterns and recursively divide the cluster till each cluster comprises a single pattern. Partitional clustering algorithms partition the dataset into a pre-defined number of clusters. They can further be classified into two categories based on their input parameters: algorithms which explicitly take the number of clusters k as input and algorithms that take as input, a threshold τ on the radius of the clusters which indirectly determines the number of clusters. k-means and k-medoids are examples of algorithms in the first category. Algorithms like Leader, DB-Scan, etc., fall in the second category.

In this paper, we propose the *two-level k-means clustering algorithm*, a variant of the MacQueens k-means algorithm. It takes τ as input and initially partitions the dataset into a predefined k' $(k' < k)$ number of clusters. It then refines these clusters using the threshold τ to generate k clusters. We show both analytically and empirically that our algorithm requires lesser number of distance computations than standard k-means algorithms and hence gives better time performance. We establish a relationship between τ and k through the analysis of our algorithm which forms a bridge between the two types of partitional clustering algorithms. We also establish a bound on the clustering error of our algorithm.

We finally present a generic scheme for the integration of the two-level k-means algorithm into classification algorithms. We prove that this integration renders the running time complexity of the classification algorithm linear. We integrate our algorithm in Support Vector Machines and k-NN classifier and thus demonstrate that better time performance is achieved.

2. Background

In this section, we present a brief introduction to the k-means problem and associated k-means clustering algorithms that the proposed algorithm is based on.

* Corresponding author. Present address: Yahoo! Software Development India, Torrey Pines, Embassy Golf Links Business Park, Off Indiranagar - Koramangala Intermediate Ring Road, Bangalore 560071, India. Tel.: +91 80 30774396; fax: +91 80 30774455.

E-mail addresses: cradha@ymail.com (R. Chitta), mnm@csa.iisc.ernet.in (M. Narasimha Murty).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.09.019

2.1. *k*-Means algorithm

The *k*-means problem is to determine *k* points called *centers* so as to minimize the *clustering error*, defined as the sum of the distances of all data points to their respective cluster centers. The most commonly used algorithm for solving this problem is the *Lloyd's k-means algorithm* [2,3] which iteratively assigns the patterns to clusters and computes the cluster centers.

MacQueens *k*-means algorithm [4] is a two-pass variant of the Lloyd's *k*-means algorithm:

1. Choose the first *k* patterns as the initial *k* centers. Assign each of the remaining $N-k$ patterns to the cluster whose center is closest. Calculate the new centers of the clusters obtained.
2. Assign each of the *N* patterns to one of the *k* clusters obtained in step 1 based on its distance from the cluster centers and recompute the centers.

Analysis: Distance computation is the only time-consuming operation in this algorithm. So, we focus on the number of distance computations performed.

In step 1, the number of distance computations needed is given by $k(N-k)$. The number of distance computations in step 2 equals Nk. This implies that the total number of distance computations equals $k(N-k)+Nk=2Nk-k^2$ and the complexity is $O(Nk)$.

3. Related work

Our two-level *k*-means algorithm belongs to the class of multi-level clustering algorithms. These techniques essentially involve the following generic steps:

- divide the given dataset into blocks,
- cluster each block separately into a predefined number of clusters, and
- cluster the cluster centers obtained from all the blocks into the required number of clusters.

Multi-level clustering enables easier handling of large datasets since only one block needs to be held in the memory at any time. The divide-and-conquer strategy employed allows parallelization which, in turn, increases the clustering speed. Also, different clustering algorithms can be adopted at different levels allowing the incorporation of domain knowledge.

Algorithm Small-Space [5] arbitrarily partitions the input dataset χ into *l* blocks. Each block is clustered into *k* clusters. A second level dataset χ' is formed from the $O(lk)$ cluster centers, each weighted by the number of patterns assigned to it. χ' is then clustered into *k* clusters. This can be extended to multiple levels. It has been proved that the clustering error of the solution obtained through *Algorithm Small-Space* is a constant-factor approximation of the clustering error of the optimal clustering algorithm.

Clustering algorithms have been employed to improve the efficiency of different classifiers. We apply our clustering-based classification technique to Support Vector Machines and the *k*-NNC classifier.

Clustering-based SVM (CB-SVM) [6] integrates BIRCH [7], a hierarchical micro-clustering algorithm into SVM. CB-SVM reduces the number of training patterns that are input to SVM by first clustering the dataset using BIRCH and identifying the patterns that will contribute to the learning process. Empirical results show that CB-SVM achieves a significant reduction in the training time and improvement in test accuracy.

Zhang and Srihari [8] construct a Cluster Tree for the given data. The tree is traversed top-down based on the similarity between the test pattern and the nodes at each level to determine the *k* -nearest neighbors of the test pattern. The class of the test pattern is then determined by majority voting.

4. Two-level *k*-means clustering algorithm

In this section, we present the *two-level k-means clustering algorithm* (see Fig. 1). We prove that it requires lesser number of distance computations than MacQueens *k*-means algorithm.

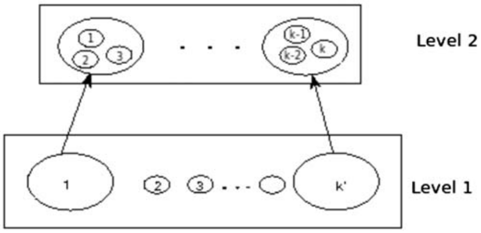

Fig. 1. Two-level *k*-means clustering algorithm.

Further analysis establishes the relationship between the number of clusters in the dataset and the expected size of each cluster. We then prove that its clustering error is less than twice the error of the optimal clustering algorithm.

Inputs: Dataset $\chi=\{x_i\}_{i=1}^{N}$, $x_i \in \Re^n$; Radius threshold τ
Algorithm:

1. In the first level, cluster the given dataset into an arbitrarily chosen k' number of clusters.
2. Calculate the radius r_i $(i=1,2,\ldots,k')$ of the *i*th cluster.
3. If the radius of the cluster is greater than the user-defined threshold τ, split it using *k*-means with the number of clusters set to $(r_i/\tau)^n$. This is the second level of clustering which yields a total of *k* clusters, each having a radius less than τ.

4.1. Analysis

4.1.1. Bound on number of distance computations

We are given a dataset $\chi=\{x_1,x_2,\ldots,x_N\}$ where $x_i \in \Re^n$ are independent samples, all drawn from the same distribution (see Table 1).

On using the MacQueens two-pass *k*-means algorithm for clustering the data into *k* clusters, the number of distance computations made is given by

$$ND_1 = 2Nk - k^2 \tag{4.1}$$

Suppose we use the same two-pass algorithm in both the levels of the two-level *k*-means algorithm described. Then the number of distance computations in level 1 is given by

$$ND_{L1} = 2Nk' - (k')^2$$

After the first level of clustering, we have k' clusters $\{\chi_1,\chi_2,\ldots,\chi_{k'}\}$ with centers $\{c_1,c_2,\ldots,c_{k'}\}$. Let us define the radius of a cluster χ_i as

$$r_i = \max_{x_j \in \chi_i} \quad d(x_j, c_i) \tag{4.2}$$

where $d(x,y)$ is the metric distance between vectors *x* and *y*. Cluster *i* is sub-clustered in the second level if $r_i > \tau$. For $1 \le i \le k'$