Information Sciences 181 (2011) 1722–1734

Contents lists available at ScienceDirect

Information Sciences

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

An agglomerative clustering algorithm using a dynamic *k*-nearest-neighbor list

Jim Z.C. Lai [a], Tsung-Jen Huang [a,b,*]

[a] Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202, Taiwan, ROC
[b] Department of Information and Communications Research Laboratories, Industrial Technology Research Institute, Hsinchu 310, Taiwan, ROC

ARTICLE INFO

Article history:
Received 24 February 2010
Received in revised form 28 December 2010
Accepted 2 January 2011
Available online 9 January 2011

Keywords:
Nearest neighbor
Agglomerative clustering
Vector quantization

ABSTRACT

In this paper, a new algorithm is developed to reduce the computational complexity of Ward's method. The proposed approach uses a dynamic *k*-nearest-neighbor list to avoid the determination of a cluster's nearest neighbor at some steps of the cluster merge. Double linked algorithm (DLA) can significantly reduce the computing time of the fast pairwise nearest neighbor (FPNN) algorithm by obtaining an approximate solution of hierarchical agglomerative clustering. In this paper, we propose a method to resolve the problem of a non-optimal solution for DLA while keeping the corresponding advantage of low computational complexity. The computational complexity of the proposed method DKNNA + FS (dynamic *k*-nearest-neighbor algorithm with a fast search) in terms of the number of distance calculations is $O(N^2)$, where N is the number of data points. Compared to FPNN with a fast search (FPNN + FS), the proposed method using the same fast search algorithm (DKNNA + FS) can reduce the computing time by a factor of 1.90–2.18 for the data set from a real image. In comparison with FPNN + FS, DKNNA + FS can reduce the computing time by a factor of 1.92–2.02 using the data set generated from three images. Compared to DLA with a fast search (DLA + FS), DKNNA + FS can decrease the average mean square error by 1.26% for the same data set.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

Data clustering is frequently used in a number of applications, such as vector quantization (VQ) [9,16,17,20], document collection [1,3], pattern recognition [26], knowledge discovery [6], speaker recognition [22], fault detection [10], and web/data mining [4]. Among the clustering formulations which minimize a cost function, *k*-means clustering is perhaps the most widely-used and studied [11]. The *k*-means clustering algorithm, which is also called the generalized Lloyd algorithm (GLA), is a special case of the generalized hard clustering scheme, when point representatives are adopted and the squared Euclidean distance is used to measure the distortion (dissimilarity) between a data point and its cluster representative.

The main drawback of the GLA is that it gets stuck to the local optimal solution, and to resolve this problem, simulated annealing was proposed [9]. However, simulated annealing requires a large amount of computing time and only gains a little improvement [15]. Another approach of obtaining a codebook is the hierarchical agglomerative clustering [25], which is also called the Ward's method [28]. The double linked algorithm (DLA) uses the approximate *k*-nearest-neighbor graph for agglomerative clustering [8], and can dramatically reduce the computing time of Ward's method by obtaining the

* Corresponding author at: Depatment of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202, Taiwan, ROC.
E-mail address: ph0427@gmail.com (T.-J. Huang).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.01.011

approximate solution. The hierarchical agglomerative clustering can usually obtain a better clustering result than the GLA [8]. For a data set of N data objects, the computational complexity of GLA in terms of the number of distance calculations is $O(NMt)$, where M is the number of clusters, and t is the number of iterations. It is noted that $M << N$ and $t << N$[13] in general. The computational complexity of Ward's method is $O(N^3)$ [25,28].

To reduce the computational complexity of Ward's method, Kurita proposed a method of storing all pairwise cluster distances into a heap structure [14]. However, this method requires $O(N^2)$ memory to store distances, and it is impractical for large data sets. The computational complexity of Kurita's method is $O(N^2 \log_2 N)$. Fränti et al. [7] also proposed a fast pairwise nearest neighbor (FPNN) algorithm to reduce the computing time of Ward's method. The computational complexity of the FPNN algorithm is $O(\tau N^2)$, where τ is the average number of clusters to be updated at each stage of the cluster merge. Kaukoranta et al. [12] presented the lazy pairwise nearest neighbor method (Lazy) to postpone some distance calculations in the process of the cluster merge. Virmajoki et al. [27] proposed an algorithm, which combines partial distortion search (PDS) [2], mean-distance-ordered partial search (MPS) [24], and Lazy [12] to speed up the FPNN. This algorithm is referred to as FPNN + PDS + MPS + Lazy in this paper. Recently, several interesting algorithms have been developed to improve the clustering performance. Lee et al. [19] developed a graph-based method to improve clustering accuracy. In reference [23], a tabu search based approach is proposed to improve the clustering result.

In this paper, we will present a fast agglomerative clustering algorithm to resolve the problem that DLA can obtain only an approximate solution of agglomerative clustering while maintaining the low computational complexity of DLA. The proposed method uses the dynamic *KNN* (*k*-nearest-neighbor) list to store k nearest neighbors for each cluster. Using the proposed approach, the *KNN* list for each data point should be determined in the initialization process, which is also required for DLA. Therefore, after each merging process, the proposed approach first determines a set of clusters the nearest neighbors of which should be updated. In the merging and updating process of each iteration, we update the *KNN* lists of clusters which are affected by the merging process. If the *KNN* lists are empty for some of the clusters being updated, their nearest neighbors are determined by searching all the clusters. That is, the proposed approach can guarantee the exactness of a cluster's nearest neighbors and can obtain as good a clustering result as that of the FPNN and Ward's method. The FPNN, which uses 1*NN* list, can be considered to be a special case of the proposed method. A key point of the proposed approach is that the updating process is avoided until the *KNN* list of the cluster being updated is empty. This is because of the monotony principle, which states that any optimal merge cannot decrease the merging cost of any existing cluster pair [12]. The merging cost is defined as being the cluster distance between a cluster and its nearest neighbor. To study the effect of search algorithm on the proposed method, we use full search, MPS + PDS + Lazy [24,27], and fast search [18,21] algorithms to determine k nearest neighbors of a cluster.

This paper is organized as follows. Section 2 describes the double linked algorithm. Section 3 presents the algorithm developed in this paper. Some experimental results are given in Section 4, and concluding remarks are presented in Section 5.

2. Double linked algorithm

Ward [28] proposed a general hierarchical clustering method to obtain the minimum increase of information loss, which is defined in terms of an error sum-of-squares criterion. Initially, each single node forms a cluster by itself, and then the two clusters with minimal distance are merged repeatedly until the desired number of clusters is obtained. Ward's method partitions a set of N data points into M clusters through a sequence of merging operations. The distortion between two data points $\mathbf{X} = [x_1, x_2, \ldots, x_d]^t$ and $\mathbf{Y} = [y_1, y_2, \ldots, y_d]^t$ is defined as being the squared Euclidean distance between these two data points. That is, $d(\mathbf{X}, \mathbf{Y})$ is defined by the following equation:

$$d(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{d} |x_i - y_i|^2. \tag{1}$$

The increase in the distortion of the two merging clusters R_a and R_b into one cluster R_{ab} can be calculated by the following equation [5]:

$$D_{a,b} = \frac{n_a n_b}{n_a + n_b} d(\mathbf{C}_a, \mathbf{C}_b), \tag{2}$$

where $n_a = |R_a|$ is the number of data points in R_a; $n_b = |R_b|$ is the number of data points in R_b; $\mathbf{C}_a$ is the center of R_a; and $\mathbf{C}_b$ is the center of R_b. $D_{a,b}$ is called the cluster distance between R_a and R_b. The cluster center $\mathbf{C}_{ab}$ and cardinality n_{ab} (the number of data points) of R_{ab} are updated as follows:

$$\mathbf{C}_{ab} = (n_a \mathbf{C}_a + n_b \mathbf{C}_b)/(n_a + n_b), \tag{3}$$

$$n_{ab} = n_a + n_b. \tag{4}$$

In this paper, clusters R_a and R_b will be abbreviated to clusters a and b, respectively.

It was noted that Ward's method had expensive computations. Therefore, Fränti et al. [8] used the *KNN* list and *IKNN* list for each cluster to reduce the corresponding computational complexity. The *KNN* list records k nearest neighbors of a cluster, while the *IKNN* list stores clusters having a particular cluster as one of their k nearest neighbors. That is, *KNN*[*l*][*j*] records the

دریافت فوری ← متن کامل مقاله

ISIArticles
مرجع مقالات تخصصی ایران

✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
✓ امکان دانلود نسخه ترجمه شده مقالات
✓ پذیرش سفارش ترجمه تخصصی
✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
✓ دانلود فوری مقاله پس از پرداخت آنلاین
✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات