Future Generation Computer Systems 34 (2014) 26–46

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Shortest-linkage-based parallel hierarchical clustering on main-belt moving objects of the solar system

Cheng-Hsien Tang [a], Meng-Feng Tsai [a], Shan-Hao Chuang [a], Jen-Jung Cheng [b], Wei-Jen Wang [a,*]

[a] Department of Computer Science and Information Engineering, National Central University, Jhongli City, Taoyuan County 32001, Taiwan
[b] Department of Computer Science and Information Engineering, National Taiwan University, Taiwan

HIGHLIGHTS

- We parallelize traditional hierarchical clustering based on shortest linkage.
- We use two real datasets, 550,000 and 300,000 objects, for performance evaluations.
- The results show that our pruning strategy reduces execution time and storage usage.
- Our fast update algorithm adds an object into a tree of 550,000 objects in seconds.

ARTICLE INFO

Article history:
Received 18 November 2012
Received in revised form
11 December 2013
Accepted 13 December 2013
Available online 21 December 2013

Keywords:
Hierarchical clustering
Incremental update
Parallel computing

ABSTRACT

Data clustering is an important data preparation process in many scientific analysis researches. In astronomy, although the distributed environments and modern observation techniques enable users to collect and access huge amounts of data, the corresponding clustering process may become very costly. One of the challenges is that the sequential clustering algorithms, that can be applied to cluster hundreds of thousand main-belt asteroids to reason about the origins of the main-belt asteroids, may not be used in the distributed environment directly. Therefore, this study focuses on the problem of parallelizing the traditional hierarchical agglomerative clustering algorithm using shortest-linkage. We propose a new parallel hierarchical agglomerative clustering algorithm based on the master–worker model. The master process divides the whole computation into several small tasks, and distributes the tasks to the worker processes for parallel processing. Then, the master process merges the results from the worker processes to form a hierarchical data structure. The proposed algorithm uses a pruning threshold to reduce the execution time and the storage requirement during the computation. It also supports fast incremental update that merges new data items into a constructed hierarchical tree in seconds, given a tree of about 550,000 data items. To evaluate the performance of our algorithm, this study has conducted several experiments using the MPCORB dataset and a dataset from the DVO database. The results confirm the efficiency of our proposed methodology. Compared with prior similar studies, the proposed algorithm is more flexible and practical in the problem of distributed hierarchical agglomerative clustering.

1. Introduction

Data clustering is essentially the process of grouping similar data items into classes. Similarity or distance measurements are usually determined based on the types of data values as well as the application requirements, without having to consult domain experts. The clustering process automatically discovers hidden relationships and supports consequent complicated analysis. Among various clustering methods, hierarchical clustering is a popular category for many applications because the outcomes of clustering create a hierarchical structure. The hierarchical structure reveals clusters at different levels of granularity, and thus provides great flexibility of examining data classes. With the versatile hierarchical structure, users are also free from considerations about deciding the number of appropriate clusters. Hierarchical agglomerative clustering (HAC), a sub-group of hierarchical clustering, links similar data items into micro clusters, and then continuously merges them to form macro clusters, level by level. The traditional HAC algorithm is an unsupervised learning algorithm, which does not involve labeling data or assigning the number of clusters in advance. It typically requires calculating the distances of all pairs of data items and micro clusters to construct a hierarchical tree.

* Corresponding author. Tel.: +886 3 4227151x35205; fax: +886 3 422 2681.
E-mail address: wjwang@csie.ncu.edu.tw (W.-J. Wang).

0167-739X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.future.2013.12.029

This study was motivated by the issue of applying HAC to the study of the main-belt asteroids of the solar system. The main-belt asteroids are small objects located roughly between the orbits of the planets Mars and Jupiter. The existing main-belt asteroids are a result of numerous collisions among ancient asteroids. After each collision, a parent asteroid may produce new asteroids of similar orbits. Therefore, the asteroids can be clustered into several asteroid families, each of which is a group of asteroids of similar orbits and may present a group of a common origin [1]. There are two major reasons why HAC is a better solution to this particular problem. First, it is not appropriate to use any directly divisive, multi-dimensional clustering approach, which uses pre-partitioning to create small hyperspace cells, corresponding to several small tasks of clustering. The similarity function of two orbits involves the computation of circular functions such as the trigonometric functions, and thus clustering of orbits may not directly use the hyperspace partitioning technique. This restriction limits the choices of using existing clustering algorithms. Second, the similarity (or distance) functions have some tunable coefficients [2] making it hard to guess a right clustering threshold even for an expert. As a result, using a hierarchical tree can reduce the time for obtaining multiple clustering results under different clustering thresholds.

In the literature, applications using traditional HAC algorithms usually focus on a small dataset [3–9], which may be analyzed by a shared-memory system. However, those shared-memory-based solutions are not appropriate to some of today's big data applications because the applications may take an extremely large size of data as their inputs, such as the astronomy applications. For example, the Pan-STARRS project[1] collected about 25,000 science quality images until early May 2010. Each image requires about 2 GB of storage, resulting in a storage requirement of 50 Tb. The data can be reduced to several smaller datasets, ranging from 10^5 to 10^{12} data items. However, the space complexity of a HAC algorithm, which is $\mathcal{O}(n^2)$ where n is the number of data items, still creates a requirement of terabyte-scale storage space for clustering the smaller datasets in practice. In addition, the complexity of a HAC algorithm is polynomial time, ranging from $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$. For a vast dataset, those algorithms usually demand long processing time. Furthermore, the HAC requires comparing all intermediately aggregated results, making it difficult to allot computations into independent tasks and to split them into different machines. Adapting HAC methods to a distributed environment therefore cannot have such a straightforward parallel approach as the Map-Reduced model. Therefore, the major challenge of this research is to improve the execution time and to minimize the individual storage space in a distributed computing environment. Since a HAC algorithm uses the similarity value of each pair of data items to cluster data, it can be treated as an instance of the all-pairs problem [10]. In a distributed environment, the all-pairs problem becomes critical when the system tries to synchronize large amounts of data, to balance the workload, and to process computation efficiently. Moretti et al. [10] shows that when a non-expert user attempts to solve all-pairs problems on a distributed environment, the performance may be even lower than using a single machine.

In addition to the efficiency issue, maintaining the up-to-date clustering result is also an important issue since we do not want to re-construct a HAC of thousands of data items every time when a new data item is obtained. This problem should be solved by using a fast incremental update procedure. The challenge is that, most scientists prefer a single and unique result since it accommodates the creation of further analytical tools and methods that outputs re-producible results. Therefore, approximate hierarchical clustering algorithms such as [11–13] are not appropriate solutions based on this requirement. A new definite, efficient incremental algorithm that efficiently updates a hierarchical agglomerative tree is preferable in this situation.

This research aims to parallelize the shortest-linkage-based HAC algorithm on a distributed environment. We choose shortest-linkage over complete-linkage and average-linkage because of its simplicity. Our idea to parallelize the traditional HAC algorithm is to construct multiple hierarchical sub-trees in a distributed environment concurrently, where each sub-tree corresponds to one group of data. Based on available correspondence, the adapted HAC algorithm achieves higher applicability and efficiency.

The proposed algorithm uses a divide-and-conquer strategy to parallelize hierarchical agglomerative clustering. The key technique is to use a distance threshold to prune the pairs of large distance values and then clusters the data items using the residue pairs hierarchically and concurrently. The pruning does not affect the outcome of clustering since the pruned pairs of large distance values in the early procedures are considered in the latter procedures. Moreover, the proposed algorithm supports incremental update that takes advantages of the distributed HAC data structure. Thus, it can merge new data items into a constructed hierarchical tree in a short time. We use a real dataset, the orbits of the main-belt asteroids of the solar system, to evaluate the performance of the proposed algorithm. The dataset contains about 550,000 data items. The experimental result shows that the proposed method can solve the problem of space limitation and reduce the required execution time. We also use another astronomical dataset of 300,000 data items, of which the similarity is measured in Euclid distance, to further evaluate the performance of the proposed method.

The major contribution of this research is that we successfully port the sequential shortest-linkage-based HAC algorithm to a distributed environment. In addition, we provide a fast update procedure that can add new data items to an existing shortest-linkage-based hierarchical agglomerative tree, and thus the update function avoids re-building of the whole tree. Although the algorithm is originally designed to support HAC on the main-belt asteroids, we believe it can apply to other applications of which have similar problem properties to main-belt asteroids clustering. Compared to the prior similar studies of parallel HAC, the proposed algorithm does not assume shared-memory, which has been widely adopted in the prior studies [11,14–18]. It supports incremental update for new data items, while existing HAC algorithms do not support [11–19]. Moreover, the proposed algorithm does not adapt the resulting tree to an approximate result such as [11,13]. Therefore, the proposed algorithm is more flexible and practical in the problem of distributed HAC.

The rest of the paper is organized as follows: Section 2 provides the background and the related work of this study. Section 3 presents the methodology of this study. Section 4 provides an example that demonstrates how the proposed algorithm works. Section 5 shows the experimental results, given a main-belt asteroid dataset and a sky object (DVO) dataset. We have discussed related issues of this study in Section 6, and then described the conclusions and the future work of this study in Section 7.

2. Background and related work

Many researchers have proposed several kinds of clustering methods for different problems in the literature. In this section, we will introduce the background knowledge as well as the related work of this study.

[1] Pan-STARRS: The Panoramic Survey Telescope and Rapid Response System, http://pan-STARRS.ifa.hawaii.edu.