Neurocomputing 163 (2015) 78–93

Contents lists available at ScienceDirect

Neurocomputing

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

Comparison of distributed evolutionary *k*-means clustering algorithms

M.C. Naldi [a,*], R.J.G.B. Campello [b]

[a] Federal University of Viçosa – UFV, Rodovia BR 354 – km 310, Caixa Postal: 22, CEP: 38.810-000, Rio Paranaíba, MG, Brazil
[b] Institute of Mathematics and Computer Sciences, University of São Paulo – USP, Av. Trabalhador São-Carlense, 400 Centro, Caixa Postal: 668, CEP: 13560-970, São Carlos, SP, Brazil

ARTICLE INFO

Article history:
Received 20 December 2013
Received in revised form
24 July 2014
Accepted 31 July 2014
Available online 20 April 2015

Keywords:
Distributed clustering
Evolutionary *k*-means
Privacy preservation
Low data transmission

ABSTRACT

Dealing with distributed data is one of the challenges for clustering, as most clustering techniques require the data to be centralized. One of them, *k*-means, has been elected as one of the most influential data mining algorithms for being simple, scalable, and easily modifiable to a variety of contexts and application domains. However, exact distributed versions of *k*-means are still sensitive to the selection of the initial cluster prototypes and require the number of clusters to be specified in advance. Additionally, preserving data privacy among repositories may be a complicating factor. In order to overcome *k*-means limitations, two different approaches were adopted in this paper: the first obtains a final model identical to the centralized version of the clustering algorithm and the second generates and selects clusters for each distributed data subset and combines them afterwards. It is also described how to apply the algorithms compared while preserving data privacy. The algorithms are compared experimentally from two perspectives: the theoretical one, through asymptotic complexity analyses, and the experimental one, through a comparative evaluation of results obtained from a collection of experiments and statistical tests. The results obtained indicate which algorithm is more suitable for each application scenario.

1. Introduction

Data clustering is a fundamental conceptual problem in data mining, in which one aims at determining a finite set of categories to describe a data set according to similarities among its objects [1]. This problem has broad applicability in areas that range from image and market segmentation to document categorization, bioinformatics, and distributed computing (e.g., see [2–4]), just to mention a few.

Many clustering algorithms have been proposed in the literature [2,3]. Among them, the *k*-means method has been investigated for more than half a century [5]. Recently, *k*-means has been elected one of the ten most influential data mining algorithms for being simple, scalable, and easy to adapt to different application domains [6]. However, *k*-means is sensitive to the selection of the initial cluster prototypes, as it may converge to suboptimal solutions if the initial prototypes are not properly chosen [2]. In addition, it requires the number of clusters, *k*, to be specified in advance. This can be quite restrictive in practice, since the number of clusters in a data set is generally unknown, especially in real-world applications involving high dimensional and/or distributed data.

A number of approximation algorithms have been investigated in the literature in an attempt to circumvent the above-mentioned *k*-means limitations. This includes the hybridization of *k*-means with some sort of general-purpose meta-heuristics adapted to the clustering problem [7]. Evolutionary algorithms are meta-heuristics widely believed to be able to provide satisfactory suboptimal solutions to NP-hard problems within an acceptable timeframe. From a combinatorial optimization perspective, clustering problems can be formally classified as NP-hard [8]. Probably for this reason, several evolutionary approaches for clustering problems have been proposed in the literature (e.g., see the monograph by [8] and the survey by [9] for extensive overviews). Of special interest here are those approaches based on the use the *k*-means as a local search operator to refine the global search performed by the evolutionary procedure. For instance, [10–17] adopted *k*-means for fine-tuning partitions produced by evolutionary operators designed to work with a fixed (user-defined) number of clusters *k*. Only a few papers in the literature have been devoted to evolutionary-guided *k*-means with a variable number of clusters [9]. In particular, the Evolutionary Algorithm for Clustering (EAC) proposed by [18] was mainly designed to evolve partitions with variable *k* by eliminating, splitting, and merging clusters that are systematically refined by the *k*-means algorithm. The use of guided

* Corresponding author.
E-mail addresses: murilocn@ufv.br (M.C. Naldi), campello@icmc.usp.br (R.J.G.B. Campello).

http://dx.doi.org/10.1016/j.neucom.2014.07.083
0925-2312/

mutation operators with self-adjusting application rates, among other features, has considerably improved the computational efficiency of the EAC [19]. Incorporating those features gave rise to the Fast Evolutionary Algorithm for Clustering (F-EAC) [19], which was shown (by means of extensive experiments and statistical tests) to be significantly more efficient than systematic approaches based on multiple executions of the *k*-means algorithm when the number of clusters in a data set is unknown [20,21]. Similar results were obtained when F-EAC was compared with other approximation algorithms [22]. Additionally, F-EAC variants were successfully developed for fuzzy clustering and relational data [20,23].

The amount of data produced has grown substantially over the years. Collections of documents, images, bioinformatics, and other types of data are created and increased by new technologies. In this scenario, there is a trend and growing need to distribute large data sets across separate repositories known as *data sites*. In many cases, the data are naturally distributed or generated and stored at different data sites. On these grounds, clustering algorithms must be able to extract relevant information from distributed data with good computational performance and scalability [24,4]. However, most clustering techniques, including the ones previously mentioned, consider that the data are centralized. The centralization of a large distributed data set implies high transmission and storage costs, which greatly increases the overall time of the mining process. In most cases, this option is not feasible due to computational limitations related to the working memory capacity or time availability for centralized (rather than distributed) processing. An additional complicating factor resides on preserving data privacy, which is a legal obligation in some European countries and the United States, among other countries [25]. In some scenarios, the data may be analyzed inside the repository it belongs to, but cannot be shared with any other repositories. One such example is a collaboration among different companies to obtain an improved data analysis that preserves data confidentiality.

Many Distributed Data Mining (DDM) clustering techniques have been proposed in the literature [26–28]. In our previous study [29], the distributed version of the F-EAC was proposed, called Distributed Fast Evolutionary Algorithm for Clustering (DF-EAC). The algorithm obtains, for distributed data sets, the exact result of the F-EAC for centralized data sets. However, the use of a silhouette by the DF-EAC requires multiple rounds of communication among data repositories and data privacy is not preserved by the algorithm. In other studies [30,31], the Combinations of Distributed Clustering (CDC) were proposed, which are a category of algorithms based on the generation and selection of evolutionary *k*-means clustering at each data site and, after that, the combination of the clusters obtained into a single clustering solution that represents the whole data set. CDC algorithms evaluate partitions with a relative cluster validity index, which affects their performance and result quality. Comparisons among algorithms based on different validity indices have never been made in previous studies.

Based on a set of experiments and analyses, the present paper indicates that evolutionary algorithms for clustering can be successfully applied to distributed data, specially for scenarios where the number of clusters is unknown. In particular, a novel comparison among DF-EAC and CDC algorithms using different validity indices was conducted based on two perspectives: the theoretical one, through asymptotic complexity analyses, and the experimental one, through a comparative evaluation of results obtained from a collection of experiments and statistical tests. Additionally, DF-EAC was revisited and two modifications were investigated: the first preserves data privacy among repositories and the second uses an alternative relative validation index to evaluate the resulting clusters. The modified DF-EAC variants are also compared in this study.

The remainder of this paper is organized as follows. In Section 2, a brief description of the area within which this study falls is provided. Then, in Section 3, the DF-EAC is presented, followed by a description of how it is distributed and of its complexity analysis. The CDC algorithms are described in Section 4. In Section 5, the DF-EAC and CDC algorithms are experimentally compared in order to determine which algorithms are most appropriate for each application scenario. Finally, the conclusions are addressed in Section 6.

2. Distributed clustering and privacy preservation

According to [32], DDM techniques involve discovering patterns or generating models from distributed data for which centralization is neither feasible nor desirable. In order to solve this problem, different algorithms or different parts of one algorithm are usually applied to distributed subsets of the data and, later, the results are combined into a final solution [33]. An overview of a typical DDM application is illustrated in Fig. 1.

The DDM algorithms can be categorized into exact or approximate [34]. On the one hand, exact algorithms produce a final model identical to a hypothetical model generated by a centralized algorithm with access to the full data set. On the other hand, approximate algorithms produce a model that approximates a centralized model, usually with less data transmission or computational savings.

A review of DDM techniques can be found in [32] and an extensive DDM bibliography can be consulted in [35]. In order to meet the increasing need for distributed computational techniques with good performance and scalability, distributed versions of classical clustering algorithms have been proposed. One of the most cited distributed versions of the *k*-means algorithm was proposed by [36], later improved by [37] and adapted to peer-to-peer networks by [38,39]. Ref. [40] proposed a technique to parallelize algorithms based on centroids, which includes not only the *k*-means algorithm, but others like the Expectation Maximization [27] and BIRCH [26] algorithms. Other papers proposed the distribution of hierarchical clustering algorithms with the main objective of dividing the calculation of data dissimilarity among different processing units [41,42]. Ref. [28] proposed a parallel version of the BIRCH algorithm that balances the computational load among processors in a cyclic manner. Like the *k*-means, hierarchical algorithms were also adapted to peer-to-peer networks [34]. Ref. [43] developed a partitioning-distributed clustering

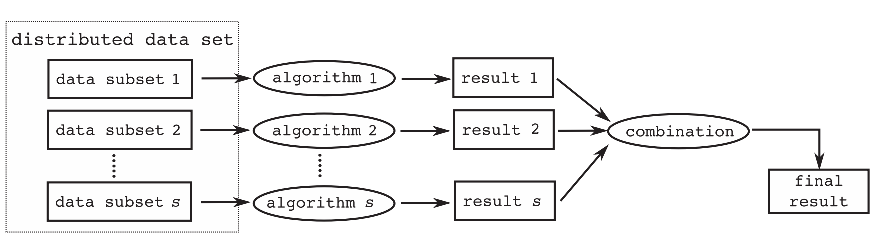

Fig. 1. Overview of a typical DDM application.