Advanced Science and Technology Letters
Vol.100 (Architecture and Civil Engineering 2015), pp.159-164
http://dx.doi.org/10.14257/astl.2015.100.35

Improving Scalability of the Nested Partition-Based Clustering

Jaekyung Yang[1], Jumi Kim[2], Wooyeon Yu[3]

[1] Dept. of Industrial and Information Systems Engineering, Chonbuk National University
567 Baekje-Daero, Deokjin-Gu, Jeonju 561-756, KOREA
jkyang@jbnu.ac.kr
[2]Korea Small Business Institute (KOSBI),
16-2 Yoido-dong, Yeongdeungpo-ku, Seoul, 150-742, KOREA
jmkim@kosbi.re.kr
[3]Dept. of Industrial and Management Engineering, Myongji University
116 Myongji-Ro, Cheoin-Gu, Yongin, Gyeonggi-Do, 449-728, KOREA
Corresponding Author: wyyuie@mju.ac.kr

Abstract. Many researchers have endeavored to improve scalability of clustering algorithm in the data mining field, since the induction of data mining models generally takes longer as data size increases even though computer systems become capable of calculating much faster. Thus, the scalability is naturally the critical issue that the data mining community faces. One of methods to handle this problem is to use a part of all data. Another scalable approach is to use an efficient search technique for finding a promising region that may have a good solution. In this paper we investigate how to improve scalability of the nested partition (NP) based clustering algorithm. With respect to scalability of the NP based algorithm, it is important to reduce the number of backtrackings which arise to modify incorrect partitioning moves. In the NP framework, we take solution samples from each partitioning region for evaluating the performance of solutions. If the variance of solution performances is large, it indicates the partition may be wrong so that the algorithm go back to the previous parent region for correcting the false partitioning move. Thus we employ the sampling scheme that can reduce the variance of solution samples. Then we show that the NP based clustering algorithm can be scalable by the solution sampling scheme that solves the noisy performance problems.

Keywords: Clustering, NP (Nested Partition), Scalability, Data Mining

1 Introduction

Recently, the databases have been exponentially getting larger as such kind data of mobile sensors and SNS become more popularly used. For the various uses of the enormous data, many efforts can be made to extract meaningful information from such large databases. Thus the scalability is one of the critical issues that the data mining community faces [1]. This paper treats a specific technique, data clustering with re-

ISSN: 2287-1233 ASTL

spect to the enhancement of scalability. Basically the goal of data clustering is to distinguish a cluster of records that is a native group or a key point of data. It can be applied for various fields such as marketing, wireless sensor networks (WSNs), etc. It can be used for recognizing customer tendencies that confirm the kind of their common features for the purpose of marketing. As the potential use of WSNs has been focused on more recently, some clustering schemes have been developed for grouping a large number of sensor nodes in WSNs into non-overlapping clusters.

There is extensive research on scalable data mining. The research includes parallel mining algorithm [4] and data reduction methods that reduce the database dimension by eliminating the unnecessary or abundant attributes have been investigated for the enhancement of the scalability [8]. Another research on scalability dealt with the usage of selected instances from datasets, which shows that it is more scalable than whole data instances [7]. It is usually known that the simplest instance selection method as a scalable manner is random sampling. Many authors employ random sampling for various data mining methods such as clustering [5], association rules [11] and decision tree [2]. It is difficult to determine the size of the samples when a sampling strategy is used. This sample size may affect the performance of an algorithm. A very small size of samples may have the result biased or spoiled. One way to solve this problem is to use an optimal sampling method [3], or statistical selection such as Rinott‟s two stage ranking and selection procedure [9]. This paper presents the scalable capability of NPCLUSTER (Nested Partitions based Clustering) [6] by employing random sampling of instances and adjusting the number of solution samples from each partitioning region in the NP framework.

It is generally expected that the performance of clustering using a small size of samples is not much better than that using whole data sets, while the computation time will be much reduced. However the NP framework probabilistically ensures an acceptable quality of performances by reducing variances of sample performances in each region. When the variance of performances in each region is getting larger, the number of backtracking in the NP procedure increases for ensuring a correct move. We present that when the NPCLUSTER uses a fairly small number of instances and a sufficient number of solution samples to minimize a sample variance, the computation time of NPCLUSTER can be scalable with respect to the data size.

2 Nested Partition Based Clustering (NPCLUSTER)

The partitional clustering problem can be formulated as an optimization problem. Thus it is solved within the NP framework which was originally developed by [10]. In particular, the NPCLUSTER (nested partitions based clustering method) is one of NP based clustering algorithms for nominal data and incorporates k-means into the same framework. In this approach the NPCLUSTER partitions a given data set into m clusters and that each cluster is defined by its center (each instance is assigned to the closest center). Thus, the decision variables are the coordinates of the j-th cluster,

where Therefore, this clustering problem reduces the locating of the centers in order to optimize certain performance. The

NPCLUSTER partitions to find out the coordinates of one attribute of a cluster at a time. Then, the center coordinate values are limited within the value range of one attribute at each iteration of a partitioning tree. In a general NP method, we take random samples from each sub-region and then incorporate k-means algorithm for fast convergence into the NP. Thus the result from the improved centers of clusters is used to select the next most promising region. The selected most promising region is partitioned. Then, what remains from the feasible region is aggregated into the surrounding region.

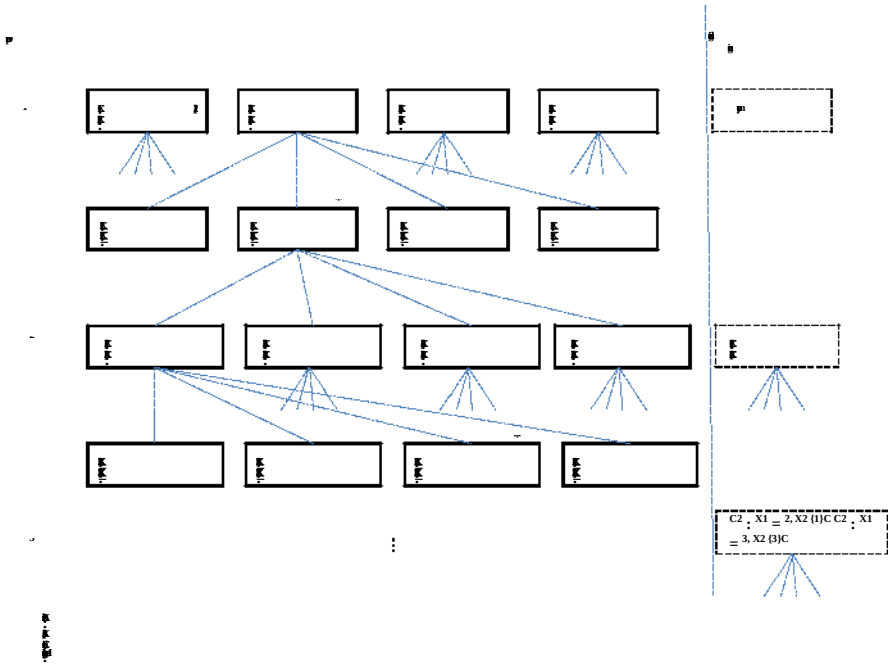

Fig. 1. Nested partitioning tree for clustering

The NPCLUSTER proceeds to partition sub-regions that are generated with values which can have an attribute. For example, the attribute named outlook has a range of values, {sunny, rainy, overcast} the promising region can be a sub-region such as {sunny, rainy} in this case. Figure 1 shows partitioning in case that all the attributes have integer values, {1, 2, 3, ..., 6}. If we look into the second node in depth 1 then,

C1: $x_1 \in \{2, 3\}$ represents the center of the first cluster whose first attribute has a value, 2 or 3, whereas the center the second cluster C_2 has a value, 1 or 3. This node derives offspring nodes by enumerative works. At this stage, the center value, x_1 of the offspring nodes is determined. From the first center value of two clusters, C_1: $x_1 = 2$;

C2: $x_1 = 1$, we can have samples with the center of a first cluster by randomly selecting instances whose first attribute value is 2 and then similarly do samples with the center of a second cluster by selecting randomly instances whose first attribute value is 1. The sample size is denoted by n. The centers of clusters are used as initial positions of clusters of k-means algorithm, and we calculate the averaged similarity of clusters

over the solution samples. The node with the best averaged similarity value is chosen as the most promising region (*) as shown in Figure 1. The remaining region except the most promising region is set as the surrounding region. If there is a best similarity performance in the surrounding region then, we backtrack to the previous iteration.

3 Improving Scalability

The original NPCLUSTER uses the fixed number of solution samples, *n* from each partitioning region. The NP based schemes select the most promising region that has the best performance of solution samples. If the surrounding region has the best performance solution sample, we backtrack to the previous parent region. The frequent backtrackings can make the algorithm computationally burden. Thus it is very important to reduce the number of backtrackings by increasing possibility including good quality samples in the partitioning regions. We may increase the number of solution samples for increasing the possibility including good quality samples. However there is a tradeoff between the sufficient number of samples and the computation time. Hence, we need to determine the optimal sample size from each region in the NP framework. In order to find the optimal sample size, we conducted experiments on various solution sample sizes over several fractional instances.

For the experiment, we used two data sets from the UCL repository of machine learning databases (http://www.ics.uci.edu/~mlearn/MLRepository.html, accessed on 10/20/2011). We used a part of the original data sets, in case they have excessive number of instances and we conducted data preprocessing. Table 1 shows a number of both the instances and attributes of data sets that were used for the experiment after preprocessing. We conducted five replications for each data set by using each algorithm and computed averaged computation time, similarity and number of backtrackings as a result.

Table 1. Characteristics of the test datasets

Data set	Number of instances	Number of attributes
breast-cancer-wisconsin	699	10
tic-tac-toe	958	9

There is one note where the similarity in the numerical results can be calculated as a Euclidean Distance. Thus, the smaller the value of similarity, better are the performances. The numerical results are reported in the following figures. In order to compare the numerical results over the solution sample size, we used various instance sample rate, R = {0.1, 0.2, 0.4, 0.8, 1.0} and the number of clusters, 2 in the experiment.

As shown in Figure 2, the similarity of NPCLUSTER does not show distinct differences between the number of solution samples and various instances sampling rates

except the case using solution sample size 5 in "tic-tac-toe" dataset. This result means that the NPCLUSTER has a capability for finding fairly good solutions using a partitioning move correction.

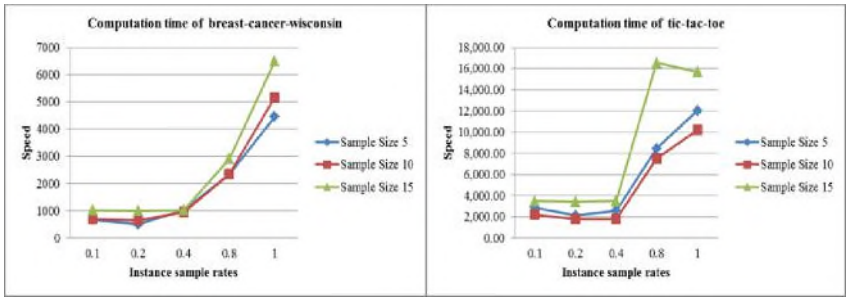

Fig. 2. Similarities of NPCLUSTER over solution sample sizes

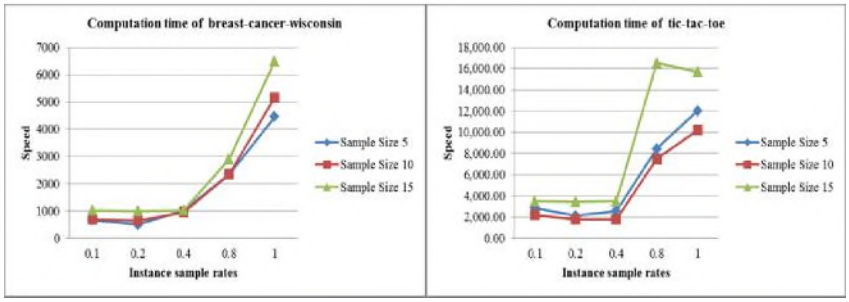

Fig. 3. Computation time of NPCLUSTER over solution sample sizes

Figure 3 reports that the computation time of NPCLUSTER with solution sample size 10 is fastest except two cases with sample size 5 and instance sample rate 20% and 100% in "breast-cancer" dataset. It implies that there is an optimal sample size for improving scalability by prohibiting backtracks without sacrificing a solution quality. In this case we can say that the optimal solution sample size is 10 experimentally at the instance sample rate 20% ~ 40%.

5 Conclusions

We showed that the NPCLUSTER is scalable with respect to the number of solution samples and instance samples with the numerical results. The NPCLUSTER also guarantees a high quality of solution in spite of using a part of the instances and insufficient solution samples. This implies that there is the optimal number of instances and solution samples. For the future research, we can investigate a sampling scheme for finding an optimal sample size within the NP framework for scalability.

Acknowledgments. This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0008995)

References

1. Bradley, PS., Gehrke, J., Ramakrishnan, R. and Srikant R.: "Scaling mining algorithms to large databases", Communications of the ACM, Vol. 45 Issue 8, pp.38-43 (2002)
2. Chauchat, JH. and Rakotomalala, R.: "Sampling strategies for building decision trees from very large databases comprising many continuous attributes", In: Liu and Motada (Eds), Instance Selection and Construction for Data Mining, Kluwer: London (2001)
3. Domingo, C., Gavalda, R. and Watanabe, R.: "Adaptive sampling methods for scaling up knowledge discovery algorithms", Data Mining and Knowledge Discovery, Vol. 6 No. 2, pp.131-152 (2002)
4. Forman, G. and Zhang, B.: "Scalability for clustering algorithms revisited", SIGKDD Explor,Vol. 2 No. 1, pp. 51-57 (2000)
5. Kaufman, L. and Rousseeuw, P.J..: Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, New York (1990)
6. Kim, J., Yang, J. and Ó lafsson, S.: "An optimization approach to partitional data clustering", Journal of the Operational Research Society, Vol. 60, pp.1069-1084 (2009)
7. Liu, H. and Motoda, H.: Instance Selection and Construction for Data Mining, Kluwer Academic Publishers, London (2001)
8. Ó lafsson, S.: "Improving scalability of e-commerce systems with knowledge discovery" in: Prabhu, V, kumar, S and Kamath, M (Eds). Scalable Enterprise system-An Introduction to Recent Advances, Kluwer, Massachusetts, pp.193-216 (2003)
9. Rinott, Y.: "On two-stage selection procedures and related probability-in-equalities", Communications in Statistics, A7, pp.799-811 (1978)
10. Shi, L and Ó lafsson S.: "Nested partitions method for global optimization", Operations Research, Vol. 48, No. 3, pp. 390-407 (2000)
11. Toivonen, H.: "Sampling large databases for association rules" in: Vijayaraman TM, Buchmann, AP, Mohan, C and Sarda, NL (Eds): Proceedings of VLDB 22nd Conference, Morgan Kaufmann, Athens, Greece, pp. 186-195 (1996)

