Chapter 38
A Fast Distribution-Based Clustering Algorithm for Massive Data

Xin Xu, Guilin Zhang, and Wei Wu

Abstract With the rapid development of data collection and storage technologies, the volume of data is getting so enormous for collection and analysis in a reasonable amount of time. Only a small fraction of the original data could be contained in the databases or data warehouses. Traditional clustering approaches are recognized as an indispensable solution to extract useful knowledge from data. However, existing conventional clustering methods all lack of robustness and computation efficiency when applied on massive data. In this work, we have made several efforts to better address the above problems with novel techniques of automatic window initialization, distribution density threshold, and window traversal based on distribution density.

Keywords Distribution-based clustering • Massive data • Data mining

38.1 Introduction

With the rapid development of data collection and storage technologies, it is not uncommon to have massive data in Gs or even Ps, either enterprise specific or private. In the past few years, there has been an exponential growth in the volumes of massive data. Since the data volume is so huge, the databases may contain just a fraction of the original data. For example, in sensor network, due to the transmission failures and variation in the information processing abilities of sensors, the data received at the server node may be incomplete. With no doubt, it is a big challenge to scramble and derive insights from the deluge of data.

It has been recognized that efficient clustering provides an indispensable solution to extract knowledge from such massive data. For this reason, it has already attracted considerable attention of researchers. Even though quite a large number of conventional clustering methods have been proposed, such as k-means [1, 2],

X. Xu (✉) • G. Zhang • W. Wu
Science and Technology on Information System Engineering Laboratory,
NRIEE, 210007 Nanjing, China
e-mail: flora.xin.xu@gmail.com

W.E. Wong (ed.), *Proceedings of the 4th International Conference on Computer Engineering and Networks*, Lecture Notes in Electrical Engineering 355,
DOI 10.1007/978-3-319-11104-9_38

k-window [3], mixture models [4], OPTICS [5], GDBSCAN [6], and hierarchical clustering [7], the traditional clustering methods all lack of either robustness or computation efficiency, thus difficult to be applied on massive data. Specifically, the conventional clustering algorithms usually fail to meet the following three requirements simultaneously: no prior knowledge of cluster number, ability to discover clusters of arbitrary shapes, computational efficiency, and ease of parallelization.

In this chapter, we have made the following major efforts to better address the above problems. The rest of the chapter is organized as follows. We briefly review related work in data clustering in Sect. 38.2. Our fast distribution-based clustering algorithm for massive data is formally proposed in Sect. 38.3. In Sect. 38.4, we present the experimental results. And we conclude in section Conclusion.

38.2 Related Work

We partition existing clustering methods related to our distribution-based clustering algorithm into three categories, k-mode clustering, variants of k-mode clustering, and the density-based clustering.

The main characteristic of the k-mode clustering methods is the demand of prior knowledge of cluster number. The representative k-mode clustering algorithms include k-means [1, 2] and general mixture model [4]. Another shortcoming of the class of k-mode clustering algorithms is that the shapes of discovered clusters are all convex, rendering it very difficult to capture clusters of arbitrary shapes. As a result, the limitation in flexibility and adaptability of k-mode clustering algorithms has impeded their application in wider domains.

Variants of the k-mode clustering methods have been proposed to address the problem of cluster number specification, such as k-windows [3] and robust multi-view k-means clustering algorithm [2]. These variants of the k-mode clustering algorithms manage to achieve a less time complexity and a better clustering result. Tasoulis proposed a generalization framework of k-windows clustering which explored the roles of different distance functions over the data sets of various structures [3]. In this way, the k-windows clustering algorithm would be scalable for data sets of unstructured nature, i.e., multimedia, time series, or genome sequence.

The final category refers to the density-based clustering, i.e., OPTICS [5] and GDBSCAN [6]. The density-based clustering method has been considered as the most robust one in terms of capturing clusters of arbitrary shapes. We ascribe the hierarchical clustering algorithms [7], as the class of density-based clustering as well. Single link, as one representative density-based clustering algorithm, iteratively merges the closest data pairs according to a certain distance function.

38.3 Method

Suppose there are n number of samples from a \$$m$\$-dimensional data set and the maximum and minimum data values of samples in each dimension are $\mathrm{min_i}$ and $\mathrm{max_i}$, $1 \le i \le m$ respectively. Our fast distribution-based clustering algorithm proceeds in three major steps to identify the clusters of the original massive data, distribution density threshold specification, random window initialization, and distribution-based window traversal. A coverage threshold k is applied to ensure each sample is covered by at least k random windows. And, a granularity threshold g is specified in order to control the sizes of random windows. The windows iteratively merge into clusters until a distribution density threshold δ is reached. The final clusters would then be identified and output.

38.3.1 Distribution Density Threshold Specification

The traditional density-based clustering algorithms adopt a spatial density threshold calculated as the number of samples in one area unit. However, the traditional density-based clustering algorithms take in no account of the variation in the sample size. In a case that the underlying unknown data distribution of the original massive data is fixed, while the number of samples from the original data set varies significantly, i.e., from hundreds to millions, it is almost impossible to predetermine a single one density threshold to accommodate the different sample sizes. As can be seen, the density threshold for the sample set with millions of samples probably could have been 10,000 times as much as that for the sample set with hundreds of samples.

For this reason, a distribution density threshold δ is adopted instead of the traditional spatial density threshold as a more robust correlation measurement for data sets with unknown distributions. Mathematically, the distribution density threshold δ is computed as the proportion of samples covered by the existing random windows. In other words, the distribution density threshold δ is calculated as the proportion of samples that belong to the traversed random windows in our fast distribution-based clustering algorithm.

$$\delta = \frac{\mathrm{num.\,of\,samples\,in\,random\,windows}}{\mathrm{sample\,size}} \tag{38.1}$$

In this way, the underlying distribution of original data could be identified with a single 1 threshold even when the sample size varies. As for the above example, a distribution density threshold δ of 80 % could produce similar clustering results for sample sets whose sizes vary from hundreds to millions.

38.3.2 Random Window Initialization

At the second step, a large number of m-dimensional windows are generated randomly, each with width w_i for each dimension i, as calculated below:

$$w_i = (\max_i - \min_i) \times g \tag{38.2}$$

A sample s is assumed to be covered by window c if and only if the distance between sample s and the center of window c is within $w_i/2$ in each dimension i.

$$\text{Relation}(s, c) = \begin{cases} \text{covered}, & \text{if } \forall i |s_i - c_i| \leq w_i/2 \\ \text{uncovered}, & \text{otherwise} \end{cases} \tag{38.3}$$

Random windows would be generated continuously until each sample has been covered k times. During this procedure, we would record and update the coverage count of each sample correspondingly. The generated random windows can be either disjointed or overlapping with each other.

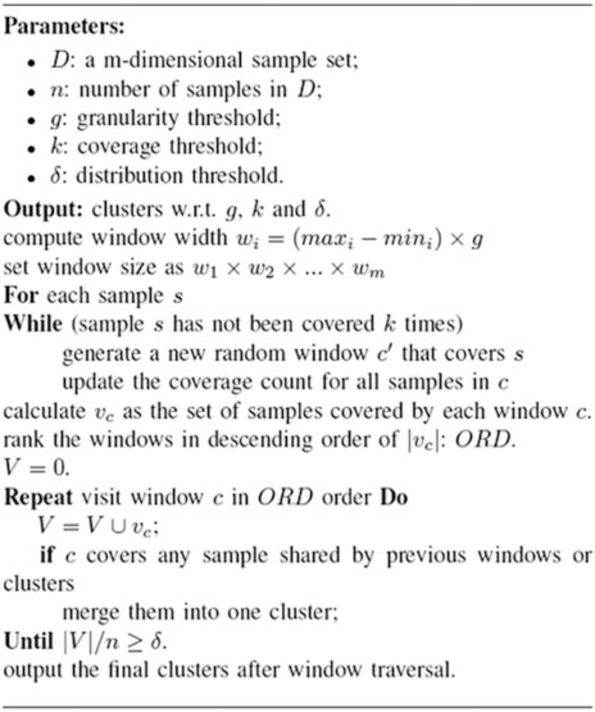
The experimental results indicate that the choice of the coverage threshold k would not influence the final clustering result that much. Usually, a coverage threshold k of 2 or 3 is applied in our fast distribution-based clustering algorithm.

38.3.3 Distribution-Based Window Traversal

The third step is distribution-based window traversal. The criterion is quite straightforward. The windows that cover more samples are considered more important and thus would be traversed before the ones covering fewer samples. In the window traversal order, the adjacent traversed windows sharing any sample would merge with each other into a cluster. The proportion of covered samples of each cluster would be updated continuously. The window traversal procedure proceeds iteratively until the proportion of the sum of covered samples in all the traversed windows has reached a predefined distribution threshold δ. The remaining clusters would be output as the final ones.

Figure 38.1 illustrates the outline of our fast distribution-based clustering algorithm for massive data.

As can be seen, our fast distribution-based clustering algorithm scales much better on massive data than the previous k-mode clustering algorithms, the variants, and the density-based clustering algorithms.



```
Parameters:
  • $D$: a m-dimensional sample set;
  • $n$: number of samples in $D$;
  • $g$: granularity threshold;
  • $k$: coverage threshold;
  • $\delta$: distribution threshold.
Output: clusters w.r.t. $g$, $k$ and $\delta$.
compute window width $w_i = (max_i - min_i) \times g$
set window size as $w_1 \times w_2 \times ... \times w_m$
For each sample $s$
While (sample $s$ has not been covered $k$ times)
        generate a new random window $c'$ that covers $s$
        update the coverage count for all samples in $c$
calculate $v_c$ as the set of samples covered by each window $c$.
rank the windows in descending order of $|v_c|$: $ORD$.
$V = 0$.
Repeat visit window $c$ in $ORD$ order Do
    $V = V \cup v_c$;
    if $c$ covers any sample shared by previous windows or clusters
        merge them into one cluster;
Until $|V|/n \geq \delta$.
output the final clusters after window traversal.
```

Fig. 38.1 A fast distribution-based clustering algorithm for massive data

38.4 Results

In experiments, we evaluate our distribution-based clustering algorithm for massive data on multivariate normal distributed simulation data. Experimental results suggest that our distribution-based clustering algorithm is both effective and scalable for massive data.

38.4.1 Evaluation of Arbitrary-Shaped Cluster Identification

We compared k-means, k-windows, OPTICS, GDBSCAN, and single linkage algorithms against our method with a simulated data set in Fig. 38.2. The 2D data set is composed of a circle-shaped cluster and two bar-shaped clusters.

As can be seen, k-means and k-windows rely heavily on the prespecified cluster number k which is actually unlikely to be known in advance. Furthermore, these two algorithms recognize convex clusters only and may probably fail to detect clusters of arbitrary shapes. The density-based algorithms OPTICS, GDBSCAN, and hierarchical clustering are much more robust in dealing with arbitrary clusters. However, their computation cost is significantly higher than our algorithm.

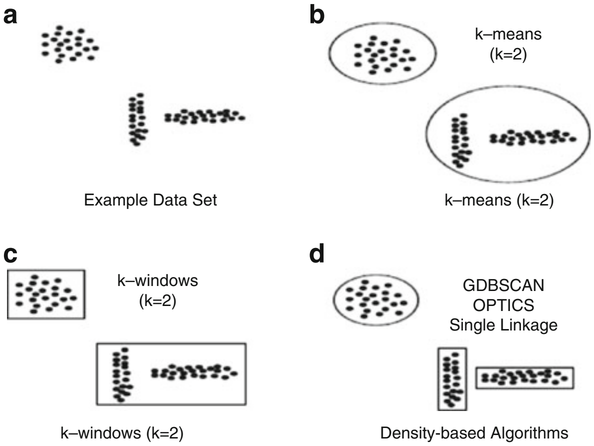


Fig. 38.2 Comparison against existing methods

38.4.2 Evaluation of Robustness

The experimental results indicate that the clustering results would not be affected much when the input parameters vary significantly. For instance, similar clustering results could be obtained when the distribution threshold δ varies between 70 and 95 %, the granularity threshold g fluctuates within range [1/8, 1/30], and the coverage threshold k changes between 1 and 4. It turns out that our distribution-based clustering algorithm is more robust for clustering massive data than the k-mode and its variants. When the initial cluster seeds or k-windows are not selected reasonably across the whole data space, the clustering results of the k-mode and its variants may have been degraded.

In addition, when the sample size varies between 2,000 and 10,000, the final clustering results remain similar under the same parameter setting. On the contrary, the final clustering results of the k-mode clustering, its variants, and the density-based clustering algorithms would be quite different.

38.4.3 Evaluation of Computational Efficiency

We compared the efficiency of our distribution-based clustering algorithm against hierarchical clustering and density-based clustering algorithms with varying sample sizes. Since the runtime of hierarchical clustering and density-based clustering are similar, we only report the comparison with hierarchical clustering here. We used the above initial parameter setting for our distribution-based clustering algorithm and vary the data set sizes between 200,000 and 600,000. Experiments on simulation data sets of various sizes show that our method is significantly more efficient than the hierarchical clustering algorithm. As can be seen from Fig. 38.3, our distribution-based clustering is orders of magnitude faster than the hierarchical

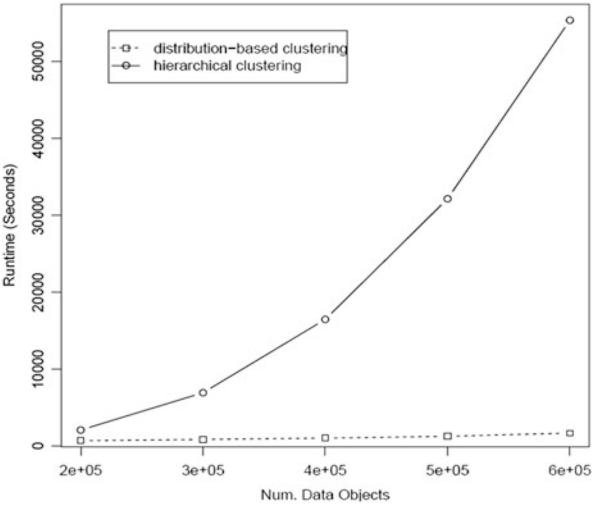


Fig. 38.3 Efficiency comparison

clustering algorithm. In addition, our distribution-based clustering algorithm is easy to parallelize for distributed data and scale well on massive data of enormous size.

Conclusion

In this work, we have proposed a novel distribution-based clustering method for massive data. With this method, we no longer need to specify the number of clusters, as required by the k-mode clustering algorithms. Instead, with our proposed coverage constraint, sufficient number of random windows will be generated automatically. And, by adopting a distribution threshold, we are able to avoid specifying different spatial density threshold due to the variation in sample sizes. Experimental results also indicate that our algorithm is both robust and efficient compared with the k-mode and density-based clustering algorithms.

References

1. Oyana TJ. A new-fangled FES-k-means clustering algorithm for disease discovery and visual analytics. EURASIP J Bioinformatics Syst Biol. 2010;2010:981–7.
2. Cai X, Nie F, Huang H. Multi-view K-means clustering on big data. In: Proceedings of the 23rd international joint conference on artificial intelligence. Beijing, China: AAAI press; 2013. pp. 2598–604
3. Tasoulis DK, Vrahatis MN. Generalizing the k-Windows clustering algorithm in metric spaces. Math Comput Model. 2007;46:268–77.
4. Kannan R, Salmasian H, Vempala S. The spectral method for general mixture models. SIAM J Comput. 2008;38(3):1141–56.

5. Ankerst M, Breunig MM, Kriegel H-P, Sander J. Optics: ordering points to identify the clustering structure. In: 1999 ACM SIGMOD international conference on management of data. Philadelphia, PA: ACM Press; 1999. pp. 49–60.
6. Sander J, Ester M, Kriegel H-P, Xu X. Density-based clustering in spatial databases: the algorithm gdbscan and its applications. Data Mining Knowl Discov. 1998;2(2):169–94.
7. Krishnamurthy A, Balakrishnan S, Xu M, Singh A. Efficient active algorithms for hierarchical clustering. In: Proceedings of the 29th international conference on machine learning. icml.cc/. Edinburgh, Scotland: Omnipress; 2012. pp. 887–94.