Efficient Distributed Density Peaks for Clustering Large Data Sets in MapReduce (Extended Abstract)

Yanfeng Zhang*, Shimin Chen†, Ge Yu*
*Northeastern University
†SKL Computer Architecture, Institute of Computing Technology, CAS

***Abstract*—Density Peaks (DP) is a recently proposed clustering algorithm that has distinctive advantages over existing clustering algorithms. It has already been used in a wide range of applications. However, DP requires computing the distance between every pair of input points, therefore incurring quadratic computation overhead, which is prohibitive for large data sets. In this paper, we propose an efficient distributed algorithm LSH-DDP, which is an approximate algorithm that exploits Locality Sensitive Hashing. We present formal analysis of LSH-DDP, and show that the approximation quality and the runtime can be controlled by tuning the parameters of LSH-DDP. Experimental results on both a local cluster and EC2 show that LSH-DDP achieves a factor of 1.7–70x speedup over the naïve distributed DP implementation and 2x speedup over the state-of-the-art EDDPC approach, while returning comparable cluster results.**

I. Introduction

Clustering is a common technique widely used in many fields, including data mining, machine learning, information retrieval, image processing, and bioinformatics. Density Peaks (DP) Cluster [1] is a novel clustering algorithm recently proposed by Rodriguez and Laio. The algorithm is based on two observations: (i) cluster centers are often surrounded by neighbors with lower local densities, and (ii) they are at a relatively large distance from any points with higher local densities. Correspondingly, DP computes two metrics for every data point: (i) its *local density* ρ and (ii) its distance δ from other points with higher density. DP uses the two metrics to locate density peaks, which are the cluster centers.

The local density ρ_i of data point i is computed as

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \tag{1}$$

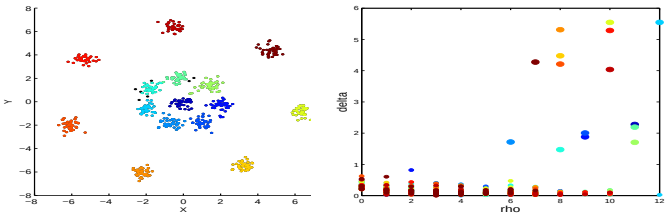
where $\chi(x) = 1$ if $x < 0$ and $\chi(x) = 0$ otherwise, and d_c is called the cutoff distance.

The δ_i distance of data point i is computed as

$$\delta_i = \min_{j|\rho_j > \rho_i} (d_{ij}) \tag{2}$$

It is the minimum distance from point i to any other point whose local density is higher than point i. Suppose $j = argmin_{j|\rho_j>\rho_i}(d_{ij})$. We say that point i is *assigned* to point j, and point j is referred to as the *upslope point* of point i.

Fig. 1a shows the distribution of a set of data points. Each point is depicted on a *decision graph* as shown in Fig. 1b by using (ρ_i, δ_i) as its x-y coordinate. Then the *density peaks* can be identified as outliers in the top right. Given the selected density peaks (cluster centers), it is straightforward to follow the assignment chain (generated when computing δ) and to assign each point to a specific density peak or cluster.



(a) Data distribution (b) Decision graph

Fig. 1. Illustrative figures for DP algorithm.

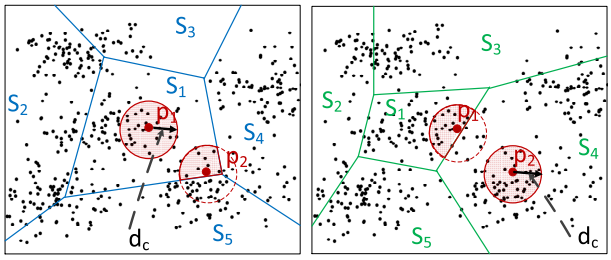
Due to its effectiveness and novelty, DP algorithm is originally published in Science Magazine [1] in June, 2014. In the past two years since its publication, DP has already been employed in a wide range of applications. While DP is attractive for its effectiveness and its simplicity, the application of DP is limited by its computational cost. In order to obtain ρ and δ, DP computes the distance between every pair of points. That is, given N points in the input data set, DP's computational cost is O(N^2). As a result, it can be very time consuming to perform DP for large data sets. Therrefore, we propose an efficient MapReduce algorithm for DP so that this promising clustering algorithm can be more broadly used.

II. Ideas and Techniques

We consider approximate algorithms in order to reduce the computation and communication cost of distributed DP. We observe that DP takes advantage of the local characteristics (such as local density) of the data points for clustering. Therefore, it is natural to employ Locality-Sensitive Hashing (LSH) [2] to partition the input data so that closer points are more likely to be assigned to the same partition.

To reduce the number of false negatives, we employ a combination of M hash groups, $(G_1, G_2, \ldots, G_M)$. That is, the point set is partitioned in M different ways, and we will have M LSH partition layouts $(\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_M)$ of the set S. For example, Fig. 2 illustrates two possible LSH partitions of point set S. The use of multiple hash groups can mitigate the problem. By using the first partition layout as shown in Fig. 2a, ρ_1 is correctly approximated but ρ_2 is wrongly approximated. While by using the second partition layout as shown in Fig. 2b, ρ_2 is correctly approximated but ρ_1 is wrongly approximated. By an aggregation phase that compares and chooses the larger one, we can obtain the correct results for both ρ_1 and ρ_2. Similarly, we can approximate δ values.

Given the approximated ρ_i, δ_i, we draw the decision graph in a central manner. The density peaks are identified as outliers



(a) LSH partition layout 1 (b) LSH partition layout 2

Fig. 2. ρ computation in two partition layouts (in plane view).

in the top right of decision graph. However, drawing a visible figure with millions of points is not feasible. To address this problem, we combine a set of close points with similar (ρ_i, δ_i) as a supernode but leaving the points with large (ρ_i, δ_i) drawn separately since only the points with large (ρ_i, δ_i) could be considered as density peaks. Note that it is possible to design certain criteria for choosing the peaks automatically. However, we believe it is better to retain this user-algorithm interaction, since the visualized reference (i.e., decision graph) provides users with an opportunity to better understand the data and choose the preferred clustering result. This is a key feature that distinguishes DP from other clustering algorithms (e.g., Kmeans and DBSCAN), which require users to face the challenge of specifying key algorithm parameters in advance. Given the chosen density peaks (i.e., cluster centers), we follow the upslope point for each point to assign it to a cluster. We then aggregates the local results from all partitions to obtain the final approximate results.

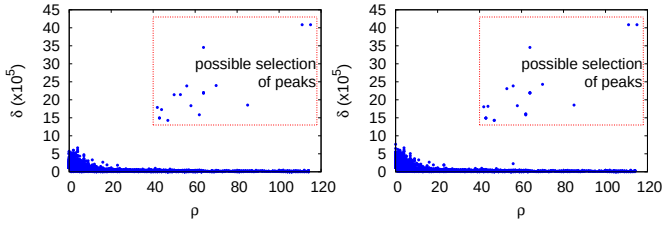
However, there are two challenges in employing LSH for DP. The first challenge is the accurate approximation of δ. While the local density ρ is a local property, δ is the the minimum distance to other points with higher ρ. Given a point p, it is possible that other points with higher ρ are far away from p and thus do not reside in p's local partition. Furthermore, since the density peaks are with large δ_i, they are distant from each other and are unlikely to be hashed to the same bucket under a locality-preserving hash function. Therefore, LSH-DDP may wrongly recognize these density peaks as the absolute density peak. Although these points are very likely to be the local density peaks and also probably be chosen as density peaks in the density peak selection step, a few wrong selections of density peaks will change the cluster result and result in more fine-grained clusters.

The second challenge is to provide guarantees for approximation accuracy of δ and ρ. It would be nice if LSH parameters such as the number of hash functions and the number of local partitions can be derived from the approximation accuracy target specified by the user. Finally, the LSH parameters may also impact the runtime of the solution. Therefore, it is important to study the tradeoff between approximation quality and efficiency.

We propose δ rectification approach with theoretical guarantees to solve the first challenge and propose an offline parameter tuning scheme for the second challenge. By applying these technics, LSH-DDP becomes more effective and efficient.

III. Experimental Results

Effectiveness. In order to visualize the cluster result, we run the naive implementation Basic-DDP and LSH-DDP on a small sized 2D data set, S2. Fig. 3(a) and (b) show the decision graphs for Basic-DDP and LSH-DDP, respectively. The decision graph of Basic-DDP is generated using the computed exact (ρ, δ) values, while the decision graph of LSH-DDP is drawn using the approximate ($\hat{\rho}$, $\hat{\delta}$) values. We show a possible selection of peaks on the two decision graphs (i.e. all points that satisfy $\rho > 40$ and $\delta > 14$).



(a) Basic-DDP (b) LSH-DDP

Fig. 3. Decision graphs (S2).

We see that the decision graphs of Basic-DDP and LSH-DDP are roughly the same. The only difference is that one more peaks is chosen in LSH-DDP decision graph. This is reflected in the cluster result. One more group of points is clustered in LSH-DDP. However, the cluster results of Basic-DDP and LSH-DDP are almost the same. Differences exist only at boundary points and/or for deciding whether a set of points should be clustered at a finer granularity.

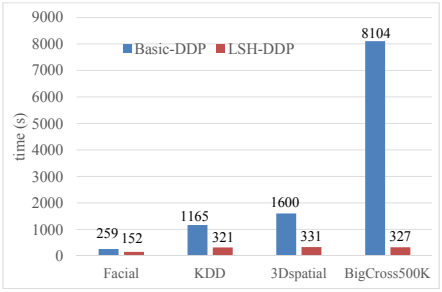


Fig. 4. Basic-DDP vs. LSH-DDP on runtime for different data sets.

Efficiency. We run the naive MapReduce implementation i.e., Basic-DDP and LSH-DDP on the Facial, KDD, 3Dspatial, and BigCross500K datasets on the local cluster of machines. As shown in Fig. 4, LSH-DDP is dramatically better than Basic-DDP, achieving 1.7–24x speedups. Moreover, the larger the data set size, the more benefit LSH-DDP brings. Further, LSH-DDP achieves 2x speedup over the state-of-the-art EDDPC approach [3], while returning comparable cluster results.

Acknowledgment

This work was partially supported by National Natural Science Foundation of China (61672141, 61528203), State Key Laboratory of Computer Architecture CAS (CARCH201610).

References

[1] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, June 2014.

[2] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *SCG '04*, 2004, pp. 253–262.

[3] S. Gong and Y. Zhang, "Eddpc: An efficient distributed density peaks clustering algorithm," in *NDBC '15*, 2015.