Distributed K-Median Clustering with Application to Image Clustering

Aiyesha Ma and Ishwar K. Sethi

Department of Computer Science and Engineering
Oakland University
Rochester, Michigan

Abstract. Developing algorithms suitable for distributed environments is important as data becomes more distributed. This paper proposes a distributed K-Median clustering algorithm for use in a distributed environment with centralized server, such as the Napster model in a peer-to-peer environment. Several approximate methods for computing the median in a distributed environment are proposed and analyzed in the context of the iterative K-Median algorithm.
The proposed algorithm allows the clustering of multivariate data while ensuring that each cluster representative remains an item in the collection. This facilitates exploratory analysis where retaining a representative in the collection is important, such as imaging applications.

1 Introduction

The K-Means clustering algorithm is a well known and popular clustering technique, with many applications. This algorithm has the limitation that the mean vector is a new vector, and thus is only relevant in some applications. For example, suppose color is an attribute in the vector, then the mean of the colors red and yellow would not make any sense. Using centroids rather than means is one variation that forces the resulting cluster representative to be an instance in the collection, thus avoiding non-sensical intermediate steps and results.

The centroid is also the L_1 multivariate median, sometimes referred to as the spatial median. The centroid is defined as the object for which the cost function, or sum of the distances to all other objects in the cluster, is minimized. While the idea of K-median, or centroid based clustering, is not novel in a non-distributed environment, this paper presents this concept in the distributed environment. Unlike the mean, the median cannot be computed in a distributed environment without extensive communication cost. Thus, this paper explores and analyzes several approximate median computations that can be used to perform K-median clustering in a distributed environment. The proposed clustering algorithm is applied to an image collection to further demonstrate the results.

A brief background on distributed clustering is presented in Section 2. The proposed distributed K-median clustering algorithm is presented in Section 3 along with approximate median computation schemes and analysis. Section 4 describes and presents the application to image clustering. Following, in Section 5, is the conclusion.

Ma A. and K. Sethi I. (2007).
Distributed K-Median Clustering with Application to Image Clustering.
In *Proceedings of the 7th International Workshop on Pattern Recognition in Information Systems*, pages 215-220
DOI: 10.5220/0002425402150220

2 Background

While some effort has been made to parallelize a couple popular clustering methods (SOM [1], K-Means [2]), some parallelization methods are not applicable to distributed environments with high communication costs. Furthermore, other clustering methods may be more relevant to a particular application.

The k-means algorithm was first parallelized for SPMD architecture using a message passing interface by Dhillon and Modha in [2]. The algorithm is a two part iterative process that continues until either a steady state is reached or the maximum number of iterations has occurred. The first part of the algorithm computes the local centers or cluster means at each processor, in parallel. The second part of the algorithm consists of a round of communication in which the local means are communicated and global cluster centers are computed. There are variations on this basic algorithm for loosely coupled systems or systems with uneven distribution across nodes, where the round of communication after each local computation may not be efficient [3].

Unlike the target environment of this paper, research on content-based retrieval in P2P systems seems to focus on the decentralized arena. These systems use vector based indexing systems for the retrieval, but in addition they organize the network topology so that peers containing semantically similar documents are located near each other. Their aim is thus to improve query efficiency by first sending queries to those peers that are likely to contain that type of semantic content. [4–8]. Though much research has gone into information retrieval in P2P environments, little research has gone into developing methods for interactive browsing in P2P environments. In [9], the authors propose a browsing method for documents.

3 Distributed K-median Clustering

The distributed k-median algorithm presented here follows the general distributed k-means algorithm first developed by Dhillon and Modha in [2]. Instead of computing the mean vector as the cluster center, however, the cluster center is computed as an approximate global median.

The approximate global median for each cluster is computed as the weighted median of the local representatives for that cluster. Let, $X_P(C)$ be the set of pairs, consisting of representatives and their weights, for peer P for cluster C,

$$\begin{aligned} X_P(C) = \{(x_i, w_i) |\ & x_i \text{ is a representative in the collection,} \\ & w_i \text{ is the number of items in the collection that } x_i \text{ represents}\} \end{aligned}$$

Then, the approximate global median of all elements in cluster C,

$$Median(C_G) = Median(\{X_P(C) | \forall P\})$$

is computed by replicating each x_i, w_i times, and computing the median. For example, if $X_{P=1}(C = 1) = \{(P1x_1, 3), (P1x_2, 1)\}$ and $X_{P=2}(C = 1) = \{(P2x_1, 2)\}$, then the global median would be $Median(P1x_1, P1x_1, P1x_1, P1x_2, P2x_1, P2x_1)$.

The distributed clustering algorithm is shown in Algorithm 1. Here, $D(x_i, C)$ is the distance between x_i and cluster center C.

Algorithm 1 Distributed k-Median Clustering Algorithm.

```
Select initial cluster centers, $C \forall k$
repeat
    In Parallel do:
    for all $x_i \in P$ do
        for all $C$ do: Compute $D(x_i, C)$. end for
        Assign $x_i$ to cluster $C$, where $D(x_i, C)$ is minimized $\forall C$
    end for
    for all $C$ do: Compute $X_P(C)$. end for
    Communicate $X_P(C) \forall C$
    End In Parallel
    for all $C$ do: Compute $Median(C_G)$. end for
until centers are stabilized
```

3.1 Selecting Local Representatives

Presented here are four methods for selecting the local representatives $X_P(C)$ for cluster C. In the following, C_P refers to all items in cluster C at peer P, and $Size$ is the number of items in the set.

Local Median The representative of the cluster is selected as the local median. While this approach works for accurately computing the mean value in a distributed environment, it is only an approximate method when computing the median.

$$X_P(C) = \{(Median(C_P), Size(C_P))\}$$

Random Sampling: n representatives are randomly selected from each cluster.

$$X_P(C) = \{(R_i, Size(C_P)/n) | R_i = Random(C_P), i \in [1, n]\}$$

Semi-Structured Sampling: The local median and randomly chosen points are selected as representatives in such a way that all items in the cluster are within an associated hypersphere. Letting neighbors be defined as,

$$Neighbors(x_i) = \{x_j | \forall x_j \in C_P, i \neq j, D(x_i, x_j) < RepDist\ \}$$

where $RepDist$ is a parameter setting, specifying the maximum radius of the spherical volume. Then the representatives are selected as,

$$\begin{aligned} X_P(C) = \{\ & (Median(C_P), Size(Neighbors(Median(C_P))) + 1), \\ & (R_i, Size(Neighbors(R_i)) + 1)\} \end{aligned}$$

$$\begin{aligned} R_i = Random \Big(& C_P \cap \overline{Median(C_P) \cup Neighbors(Median(C_P))} \\ & \cap \overline{R_1 \cup Neighbors(R_1)} \cap \overline{R_2 \cup Neighbors(R_2)} \\ & \cap \ldots \cap \overline{R_{i-1} \cup Neighbors(R_{i-1})} \Big) \end{aligned}$$

The additional representatives, R_i's, are chosen until all items in the local cluster are represented. Thus $i \in [0, Size(C_P)]$, meaning the maximum number of representatives is the size of the local cluster, and the minimum number of representatives is 1 (just the local median).

Last Median: This approach uses two representatives: the local item nearest the last calculated global median, and the local median.

$$\begin{aligned} X_P(C) = \{\ &(Nearest(Median(C_G)), Closer(Nearest(Median(C_G)))), \\ &(Median(C_P), Closer(Median(C_P)))\} \end{aligned}$$

where $Nearest(Median(C_G)) = x_i$ such that $\forall x_i, x_j \in C_P, i \neq j$, $D(x_i, Median(C_G)) < D(x_j, Median(C_G))$. $Closer$ indicates all items in the local cluster that are closer to one representative than the other. Thus, this is defined for $Nearest(Median(C_G))$ as the set $\{x_i | \forall x_i \in C_P, D(x_i, Nearest(Median(C_G))) < D(x_i, Median(C_P))\}$, and for $Median(C_P)$ this is defined as the set $\{x_i | \forall x_i \in C_P, D(x_i, Median(C_P)) < D(x_i, Nearest(Median(C_G)))\}$.

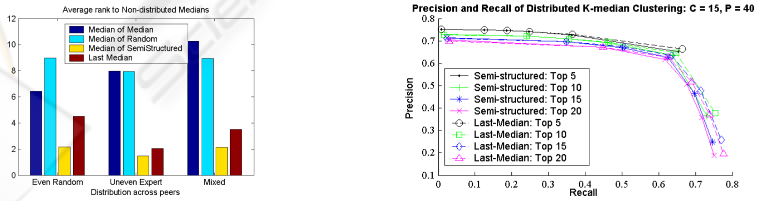
3.2 Analysis of Approximate Global Medians

A test set of 80 vectors, V, of length 2 were generated, with 4 groups of 20 points. Each group was normally distributed ($\sigma = 1$) within a quadrant in the x-y plane. The vectors were assigned to 2 peers in three possible ways: Even Random ($Size(V_P) = Size(V)/P$), Uneven Expert ($Size(V_P) \sim NORMAL(\mu = Size(V)/P, \sigma = 5)$, and peers were experts), and Mixed (uneven number of vectors, with 3/4 expert and 1/4 random). Each K-median algorithm was run with the same initial centers for a maximum of 10 iterations; convergence was usually around 3 or 4 iterations.

To compare the performance, the vectors were ranked by distance to the non-distributed medians, and the performance was the average rank of the distributed medians.

The average performance over 100 test runs is shown in Figure 1(a). The Semi-structured and Last-Median approaches tend to choose as medians points that are on average only two or three points away from the median obtained by non-distributed K-median clustering.

The semi-structured sampling approach tends to increase in the number of representatives as the number of dimensions or the complexity of the data increases. Due to this phenomena the Last-Median approach may be a better trade-off between communication (and overhead costs) and performance.



(a) Test Points: Comparison of Approximate Median Approaches for Various Data Assignments

(b) Image Clustering: Comparison of distributed to non-distributed k-median clustering, with uneven expert distribution, K = 15, P = 40

Fig. 1. Performance of proposed K-Median Clustering.

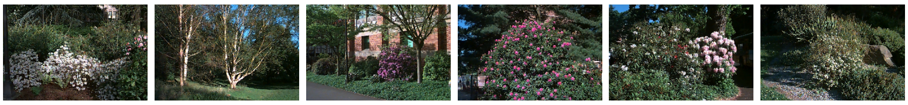


Fig. 2. Leftmost is cluster center, Spring-Flowers upper row, Cambridge lower row.

4 Application to Image Clustering

The experimental data set consisted of 7100 color images from: CD photo collection, Benchathlon [http://www.benchathlon.net], and University of Washington [http://www.cs.washington.edu/research/imagedatabase/groundtruth/]. Each feature vector consisted of global histogram with 256 bins in the HSV (Hue, Saturation, Value) color space, with 16 bins in hue, 4 bins in saturation, and 4 bins in value. This feature vector was chosen to comply with the MPEG 7 specification of a color descriptor [10, 11], and to provide a base estimate of the performance.

Images were assigned to 40 peers by Uneven Random (not shown for brevity) and Uneven Expert. The latter is more likely to be the case in practice, where a peer will have a set of favorite topics. Clustering was performed using the semi-structured and last median approaches, with $k \in \{9, 15, 40\}$. Clustering generally took between 4 and 7 iterations for all approaches.

The non-distributed clustering results were used as a baseline to analyze the distributed approaches. Letting an image I be an image in the collection, suppose I is in cluster C_N of the non-distributed approach. Then we consider $\{Near(I) \in C_N\}$ to be the relevant images to retrieve, where $Near(I)$ are the closest n images. If I is in cluster C_D of the distributed approach, then $\{Near(I) \in C_D\}$ are the closest m images retrieved. Thus, we define,

$$Precision = Size(\{Near(I) \in C_N\} \cap \{Near(I) \in C_D\})/n$$

$$Recall = Size(\{Near(I) \in C_N\} \cap \{Near(I) \in C_D\})/m$$

Tests were conducted with $n \in [5, 10, 15, 20]$), and $m \in [5, 10, 15, 20, 30, ClusterSize]$. Each image in the collection was used as a query image, and the results were averaged over the entire collection for each test. Results for 15 clusters are shown in Figure 1(b).

The two approaches performed similarly, however, the Last-Median approach favored larger k and faster peer computation times, while the semistructured approach favored smaller k and faster centralized server computation times. The maximum distance parameter for the semi-structured approach significantly affects the performance results, calculation times, and communication overhead. Thus the Last-Median approach may perform better when less information is available.

As mentioned earlier, the target of this application is to facilitate indexing and browsing of the image collection over a distributed network. Figure 2 depicts a cluster center (left) and nearest five images within the cluster for one cluster when the image set is unevenly distributed across 40 peers, and clustered with the Last-Median approach and $k = 40$. Other results will be available at http://iielab-secs.secs.oakland.edu.

5 Conclusion

This paper presents a k-median clustering approach for use in a distributed environment, such as a peer-to-peer system. While the presented approach uses the Napster model of a centralized coordinator and index, the clustering method could be extended to decentralized models by deciding on a communication scheme.

This paper compared several methods for computing an approximate median using only summary data for each peer and the approaches were analyzed within the context of the k-median clustering algorithm. It was noted that variations in data distribution (such as random versus expert) affected the performance of the proposed methods. Overall, two approaches performed well regardless, but had other trade-offs to consider.

The results of image clustering showed that enough similarities exist between the clusters produced with the non-distributed clustering and those produced with distributed clustering to ensure that browsing and indexing methods using the approximate approaches in the distributed environment are possible. Furthermore the clustering algorithm worked well given the limitations of the feature vector used.

References

1. Lawrence, R.D., Almasi, G.S., Rushmeier, H.E.: A scalable parallel algorithm for self-organizing maps with applications to sparse data mining problems. Data Mining and Knowledge Discovery **3** (1999) 171–195
2. Dhillon, I.S., Modha, D.S.: A data clustering algorithm on distributed memory multiprocessors. Large-Scale Parallel Data Mining, Lecture Notes in Artificial Intelligence **1759** (2000) 245–260
3. Jin, R., Goswami, A., Agrawal, G.: Fast and exact out-of-core and distributed k-means clustering. Knowledge and Information System Journal (2005) Online first.
4. Müller, W., Henrich, A.: Fast retrieval of high-dimensional feature vectors in P2P networks using compact peer data summaries. In: MIR '03: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval, New York, NY, USA, ACM Press (2003) 79–86
5. Müller, W., Eisenhardt, M., Henrich, A.: Scalable summary based retrieval in P2P networks. In: CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management, New York, NY, USA, ACM Press (2005) 586–593
6. Blanquer, I., Hernndez, V., Mas, F.: A P2P platform for sharing radiological images and diagnoses. In: Proc. Medical Image Computing and Computer Assisted Intervention (MICCAI). (2004)
7. King, I., Ng, C.H., Sia, K.C.: Distributed content-based visual information retrieval system on peer-to-peer networks. ACM Trans. Inf. Syst. **22** (2004) 477–501
8. Yang, Z.: Interactive content-based image retrieval in the peer-to-peer network using self-organizing maps. In: HUT T-110.551 Seminar on Internetworking. (2005)
9. Fischer, G., Nurzenski, A.: Towards scatter/gather browsing in a hierarchical peer-to-peer network. In: P2PIR'05: Proceedings of the 2005 ACM workshop on Information retrieval in peer-to-peer networks, New York, NY, USA, ACM Press (2005) 25–32
10. Manjunath, B.S., Ohm, J.R., Vasudevan, V.V., Yamada, A.: Color and texture descriptors. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY **11** (2001)
11. Sikora, T.: The mpeg-7 visual standard for content descriptionan overview. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY **11** (2001)