PROPOSAL ABSTRACT:

1 Introduction

As a consequence of the explosive growth of the WEB, the integration of search engines such as Google into personal computers and mobile devices, and the wide use of social networks, the task of clustering text data, i.e. Tweets or documents in a search engine, has each time an increasing importance because of the necessity of unraveling the underlying categories in large volumes of text data. Nowadays the generation of large amounts of documents surpasses the computational power of personal computers and even of high performance computers. As an example, it is estimated that the amount of WEB pages that popular search engines such as Yahoo! and Google index is higher than the tens of a billion. It is therefore of great interest to develop algorithmic techniques capable of automatically organize, classify and summarize document collections distributed in multiple machines of a network, and that also perform efficiently in modern hardware, particularly within parallel processing frameworks in multi-core architectures. In real problems such as *Collection Selection* for distributed document databases, in which for a given query the computer node containing the most suitable sub-collection must be selected to answer it [Crestani and Markov, 2013], the challenges related to the scalability and efficiency of *Knowledge Discovery* methods have become very important. Traditional algorithms often assume that the whole dataset is loaded into main memory (RAM) and thus every document can be accessed at any time with no access latency. In scenarios where the size of the collection is much bigger than the amount of available RAM either because of the number of documents or the length of each document, this ideal loading process is unfeasible due to real constraints in the storage or computational capabilities.

Often, text data is structured in digital collections of documents whose length (number of characters) is variable, e.g. WEB pages or the content generated by users in social networks such as Twitter or Facebook. In order to enable the processing of these collections, in a first stage of preprocessing a set of words occurring in the documents are extracted and sorted in lexicographical order; this word set is referred to as the Vocabulary. Then, the content of every document in the collection is represented as a vector in which each dimension denotes a specific word of the Vocabulary and its value in a document vector is given by a function of the number of occurrences of the word within the document and the number of documents in which it appears. As a natural consequence of the lexical richness of every language, the size of the Vocabulary, and in turn the dimensionality of the vector space onto which a document is represented, is far bigger than the data size that traditional clustering algorithms manage. Because of this, the task of automatic document clustering has high computational and storage (RAM and secondary memory) costs. Along with this, when the number of documents is large (tens or hundreds of thousand and even millions of items) then traditional techniques for processing and clustering documents, and current computational capabilities of a single machine and also high performance machines are insufficient. Even in the most favorable scenario the storage and computational power tackle the challenge but the response time are excessive.

There exist three approaches successfully applied to the construction of clustering algorithms capable of processing large volumes of data. The first one introduces constraints on the number of passes allowed on a document (related to the amount of time it is loaded into main memory). The second one exploits current multi-core architectures to perform parallel processing of the data, which does not solve the massive volume problem. The last one combines the computational power of a single machine together with the scalable storage capability of a distributed system by partitioning the dataset into several independent machines connected through a network.

The last above mentioned approach seems promising since it allows to exploit the local capabilities of single computers with multi-core architectures without sacrificing scalability to large data volumes because of its distributed design. Within this path there are two contexts regarding the data generation scenario. On the one hand, in some problems where the dataset is large but collected in a centralized fashion, the strategy employed consists in partitioning the collection into several machines or nodes of a network. This scheme leads to the transmission of a lot of data during the execution of the algorithm [Nagwani, 2015]. On the other hand, there are some problems where the data is generated in a distributed fashion and where besides it is not feasible to centralize the data because of high transmission costs or privacy issues [Jagannathan et al., 2005, Liu et al., 2012], e.g. search engines work with document collections originated and stored in different geographical locations.

2 Problem Definition and State of the Art

Let $X = \{X_1, X_2, \ldots, X_p\}$ be a collection of documents where $\bigcup_{i=1}^{p} X_i = X$ and $\forall i \neq j \in [1, \ldots, p], X_i \cap X_j = \emptyset$. This collection is partitioned into p different machines of a network, and also each subset X_i consists of n_i vectors in $\mathbb{R}^d$. Additionally, in a distributed environment, data sites may be homogeneous or heterogeneous, depending if different sites contain data for exactly the same set of features or for different set of features respectively. The distributed data clustering task consists in obtaining k data partitions $C_1, C_2 \ldots, C_k$ where $\bigcup_{i=1}^{k} C_i = X$ and also $\forall i \neq j \in [1, \ldots, k], C_i \cap C_j = \emptyset$, such that each one of them represents a topical category of the overall collection and in turn all the categories are covered by the partitions or clusters C_i. Expressed in a different manner, this means that every cluster contains more similar documents according to their content in comparison to the other ones contained in different clusters. The similarity between document vectors is usually measured by means of a function $S : \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$, e.g. the cosine similarity function:

$$S(x_a, y_b) = \frac{< x_a, x_b >}{\|x_a| |x_b|},$$

where $< u, v >$ with $u, v \in \mathbb{R}^d$ denotes the inner product between two vectors and $|u|$ corresponds to the Euclidean norm of a vector. In a nutshell, this task consists in finding a structure of compact and homogeneous groups with respect to a similarity value between their members that accounts for their content.

An overall operation mode of the distributed data clustering techniques consists of four stages: Initially, the dataset is partitioned into several nodes. Next, each node generates a clustering model over its corresponding local subset. These local models are then transmitted to a central node (e.g. A model can consists of representative points identified from a local clustering), which integrates the local models in a global solution. Eventually, this global model can be re-transmitted to the other nodes in order to refine their local models and then, by performing the same previous steps, a new depurated global model is built. This scheme is depicted in Figure 1.

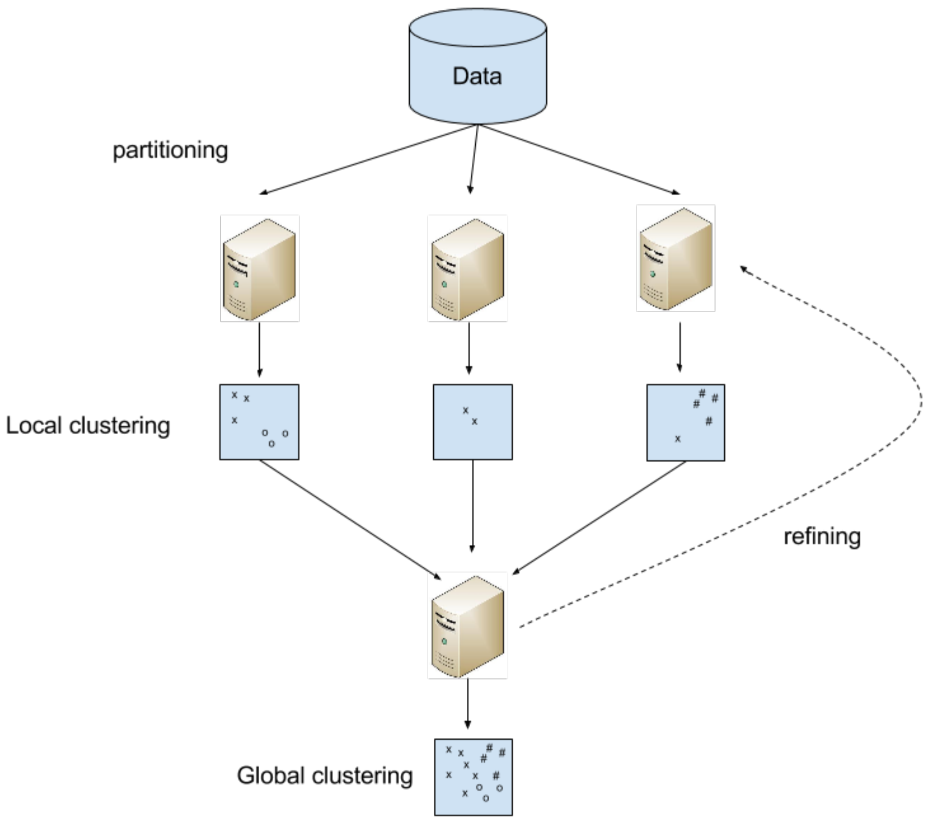

Figure 1: Overall scheme of operation of the distributed data clustering techniques.

2.1 Parallel and distributed approaches for massive and high dimensional data clustering

As far as we know from the literature, most of the existing efforts for the construction of clustering techniques capable of operating in scenarios where the data is distributed have been focused on low dimensional data (less than 100 attributes) in contrast with document datasets in which a document vector for a small collection may have about 10^4 attributes. Nevertheless, the main advances in distributed data clustering are detailed below, specially highlighting those contributions focused on methods capable of dealing with high dimensional data.

1. Main contributions on parallel algorithms [Xu et al., 2002] and [Dhillon and Modha, 1999] propose parallel extensions for the algorithms DBSCAN and K-Means. Another scalable approach based on secondary memory consists in designing algorithms capable of working within the Mapreduce framework[1]. In this context it is possible to highlight the contributions made by [Das et al., 2007] in which they propose an implementation of the EM algorithm and also by [Ene et al., 2011] in which they tackle the K-Median problem by using Mapreduce. Another parallel approach for K-Means is presented by [Bahmani et al., 2012] and it is called K-Means++. Additionally, the EM-Tree proposed by [De Vries et al., 2015] is very interesting since it allows to process very large datasets, specifically the authors show that it can handle hundred of millions of WEB pages.

2. Approaches capable of dealing with high dimensional data [Kargupta et al., 2001] propose a method to obtain the Principal Components (PCA) over heterogeneous and distributed data. Based on this contribution on dimensionality reduction they also propose a clustering method that works over high dimensional data. Once the global principal components are obtained by using the distributed method and transmitted to each node, local data are projected onto the components and then a traditional clustering technique is applied. Finally, a central node integrates the local clusters in order to obtain a global clustering model. Several years later, [Liang et al., 2013] present another algorithm for principal components extraction over distributed data. To this end, each node computes PCA over its local data and transmit a fraction of them to a central node. This node uses the received components to estimate the global principal components, which are later transmitted to each node. After this, in every node, local data are projected onto the global components and the projected data are used for computing a coreset by means of a distributed algorithm. The global coreset built from the local projected data will be finally used to obtain a global clustering model.

   [Li et al., 2003] propose an algorithm called D-CoFD for high dimensional and distributed data that also is capable of dealing with homogeneous and heterogeneous environments.

3. Density based approaches [Januzaj et al., 2003] propose a distributed data clustering technique in which local nodes build models and transmit a set of representatives of each cluster to a central node. In this node a centralized clustering method is applied over the representatives and then the resulting model is re-transmitted to the local nodes to update their models. [Klusch et al., 2003] propose a clustering technique based on density estimates. [Januzaj et al., 2004] present a scalable version of DBSCAN that is also capable of operating over distributed collections. First, the best local representatives a selected depending on the number of points that each one represents and then those chosen points are sent to a central node. This central node clusters the received local representatives into a single new model, which is re-transmitted to the other nodes so they can improve their local group structure.

4. Approaches based on parametric models [Merugu and Ghosh, 2003] propose a clustering method that combines local parametric models, each one built on a single node, into a general one. The main contribution of this work consists in a method capable of dealing with distributed data that also considers the privacy of the local data in each node, since it transmits a summary of each local data and not raw points. [Kriegel et al., 2005] present a technique that fits a Gaussian mixture model in each node using the EM algorithm. Finally, all these Gaussian mixture models are integrated into a general parametric model.

5. Approaches based on representative points [Forman and Zhang, 2000] extend centroid based techniques to identify groups over distributed data. Specifically, they extend K-Means, K-Harmonic-Means and EM. [Qi et al., 2008] propose an approximate K-Median clustering technique that works over stresaming data, i.e. data is continuously collected. [Balcan et al., 2013] address the distributed clustering problem by using centroid based techniques that use a novel coreset construction method that works for distributed data. [Naldi and Campello, 2014] tackle the problem of the parameter selection of the K-Means clustering algorithm by using evolutionary algorithms. Additionally, they propose two strategies inspired in evolutionary algorithms for clustering distributed data using centroids.

6. Hierarchical clustering approaches A hierarchical algorithm that works on distributed and heterogeneous data is presented by [Johnson and Kargupta, 2000]. This method assumes that data is vertically partitioned, thus all nodes contain the same set of points, but characterized by different features. At the beginning, a dendrogram is generated in each node, and then it is transmitted to a central node. Finally, this node combines all the dendrograms into a global model. [Jin et al., 2015] propose a hierarchical clustering algorithm for distributed data that builds the clusters by incrementally processing the data points. The authors re-state the hierarchical clustering problem as a Minimum Spanning Tree construction problem over a graph. In order to integrate several local models they also propose a technique for combining multiple minimum spanning trees, assuming that these trees were obtained from disjoint subgraphs of the complete original graph. This mixture procedure iterates until a single tree is obtained, which in turn denotes the hierarchical clustering originally pursued.

[1]Hadoop MapReduce is a software solution that enables the construction of applications capable of processing large amounts of data (e.g. Terabytes) in a parallel fashion over big computer clusters.

3 Hypothesis

A distributed master/slave approach for the design of single-pass and parallel clustering algorithms will enable the construction of methods capable of attaining results comparable to a centralized scheme, in terms of standard clustering performance measures such as Rand-score, Mutual Information and V-Measure, for high dimensional, massive and distributed document databases.

4 Goals

The general aim of this work consists in developing new parallel clustering techniques capable of dealing with large document databases that also can be stored in several nodes of a computer network. Additionally, this work comprises the following specific objectives:

- To develop parallel clustering scheme for document collections that also performs a single-pass over each document, i.e. access to secondary memory during the algorithm execution is restricted.
- To extend the parallel scheme proposed to environments where the document collection is distributed into several machines.
- To validate the proposed models with benchmark and real text data extracted from specialized sites such as UCI and NIST.
- To disseminate the obtained results and methods in this research in scientific journals registered in the ISI catalog.

5 Methodology

In order to acomplish the goals of this project the following general and specific activities are distinguished:

1. Study and discussion of the state of the art literature. Specifically, a constant and complete revision of the literature related to:
   - Parallel and distributed Clustering methods.
   - Theoretical and experimental aspects involved in the single-pass and parallel clustering methods for distributed data.
   - Hashing strategies for the efficient estimation of neighborhood over centralized and distributed data collections.
   - Clustering methods for massive data collections.
   - Distributed and parallel Machine Learning algorithms.
2. Colaboration and dissemination activities:
   - Seminar with topics about Clustering and Non-Supervised methods and their applications over problems involving distributed and massive data collections. Researchers, practitioners and students (undergraduate and postgraduate) will be included.
   - Participation in national and international conferences related to Machine Learning techniques, Intelligent Systems for distributed data and Knowledge extraction from large databases (at least one per year).
   - Establishing contact for cooperation with related research groups either in the national as in the international sphere.
3. Design and construction of the proposed methods. In addition and considering the previously posed hypothesis and its validation, the following steps are established:
   - Modeling and analysis of results under environments presenting distributed and high dimensional data.
   - Construction of the proposed models. Development of distributed processing scheme for single-pass Clustering algorithm capable of operating under scenarios in which the vectors, each one representing a document, are distributed into several computers (workers).
   - An empirical study of techniques for summarizing massive text volumes (e.g. Coresets, medoids) with the aim to generate more succint snapshots of the current state of the stream. This is of particular interest in a distributed scenario in which a message passing based synchronization is performed between workers and the communication cost is restrictive.
   - Analysis of the algorithmic complexity in terms of space and time of the proposed models.

4. During the development of the algorithm implementations we identified the following steps: (a) Construction of a modular framework adequately general to serve as a basis for the posterior construction of the proposal and its variants (b) The implementation and testing of several distributed scenarios (e.g. Balanced and Imbalance load in workers, arrival of new instances and its posterior labeling and the incorporation of concept-drift by introducing new classes to the algorithm). (c) Implementation and evaluation of baseline algorithms of the state of the art (e.g. Spark-Kmeans, Spark-DBSCAN and RACHET [[Samatova et al., 2002]]).

5. Design of the experiments and validation of the proposed algorithms. In order to validate the proposed techniques, we plan to use benchmark datasets coming from several sources; namely, the UCI machine learning repository[2], the Tipster text collection[3] and the GOV2 text collection[4]. In some cases the document set is labeled or at least some labels can be inferred from relevance judges (e.g. the Tipster collection), and in some other no label information is available (e.g. GOV2 collection). There exist several measures for the quantification of clustering quality. When labeling information is available external measures such as V-measure [[Rosenberg and Hirschberg, 2007]] and Adjusted Rand Index [[Hubert and Arabie, 1985]] are applied. In the absence of this information, internal measures such as the Silhouette [[Rousseeuw, 1987]] coefficient and CVNN [[Liu et al., 2013]] are utilized. As the partitioning of the data is performed at random, the validation will be conducted by executing 20 runs over each dataset. Then, the average and standard deviation of each attained quality measure will be reported.

6 Work Plan

The stages were explained in the Methodology section and are the following:

- *Stage 1*: Study and Discussion of the relevant literature for the proposal.
- *Stage 2*: Design of the proposed models and algorithms:
  1. Design of a processing strategy for distributed data clustering.
  2. Implementation of the proposed scheme.
- *Stage 3*: Design of Experiments and validation of the proposed algorithms:
  1. Synthetic data generation.
  2. Processing real text data. For the evaluation of clustering algorithms a general practice consists in using datasets employed for classification since these have labeling information, hence external evaluation measures for Clustering can be used. Additionally, some datasets not labeled but large in size, such as GOV2, will be considered.
  3. Comparative analysis against state of the art techniques. External and internal measures will be used and the achieved values will be compared against the ones attained by techniques such as DBSCAN and KMeans implemented within the MapReduce framework for distributed computing.
- *Stage 4*: Dissemination of the attained results.
- *Stage 5*: Study of algorithmic complexity in terms of time and space.
  – Exploration of potential optimizations of the proposed algorithms.
  – Discussion about the scalability of the proposed algorithms.
- *Stage 6*: Integration of the proposals with GPU computation at the Master and Workers levels. In this stage, GPU will be specially used to improve array products for distance computation.
- *Stage 7*: Final dissemination of the attained results and contributions of this project.

In summary, during year 2018, we will concentrate on the development of a distributed framework that addresses the high dimensional data clustering task from a core-representatives approach. That is, the overall data clustering is obtained from representatives chosen independently in each local worker (S1). Therefore the effort will be put on techniques based on Coresets and distances derived from Nearest-Neighbors shared between data points (S2). In order to assess the performance of the proposal, real high dimensional datasets will be used and a thorough experimental validation will be made (measuring clustering quality through several executions of the algorithm and under different parameter settings) (S3). At the end of 2018, the built algorithms together with their attained results will be reported in at least one indexed journal (S4).

During the year 2019, we will focus firstly, on the study of the algorithmic complexity of the overall distributed framework along with the complexity of the tasks performed by each worker (S5). Lastly, the effort will be put on the usage of GPU

[2] `http://archive.ics.uci.edu/ml/index.php`
[3] `https://catalog.ldc.upenn.edu/LDC93T3A`
[4] `http://ir.dcs.gla.ac.uk/test_collections/gov2-summary.htm`

computation to enhance performance times specially on the computation of distances and neighborhoods (these are the two most expensive operations) (S6). The improvements and findings obtained during this year will be published in at least one indexed journal (S7).

7 Work in progress

The Shared-Nearest-Neighbor clustering algorithm presented by [Ertöz et al., 2003] proposes the integration of the DBSCAN clustering algorithm along with a shared-neighbors based similarity measure. This scheme tackles the variable density, size among clusters and also the curse of dimensionality in a centralized setting. In spite of its valuable contribution this method assumes that pairwise distance/similarity computation and storage is always feasible. Due to the quadratic time complexity involved in the computation of the distances and the similar space complexity to store these data, centralized methods that rely on this calculations do not scale in massive data scenarios.

An initial approach that we took is to explore the construction of clustering algorithms capable of dealing with high dimensional and distributed text collections by adapting the centralized algorithm proposed by [Ertöz et al., 2003]. Consequently, the curse of dimensionality is addressed by treating the distance between two data points as a function of the nearest neighbors they share (nearest in terms of a traditional metric such as the euclidean distance). Additionally, the large number of observations and the potential noise or abnormal data points are addressed by a representatives selection mechanism performed on each worker. Is in this way that each worker reports only those points from its local subset that share a sufficient quantity of neighbors with their nearest neighbors. The rationale behind this scheme is that the presence of noise is denoted by more or less isolated data points. Then, the master node concentrates all these core points, builds a network based on their shared-nearest-neighbor similarity and labels them by using a network propagation scheme such as the one proposed by [Raghavan et al., 2007]. Finally, those core points along with their labels are transmitted back to the worker nodes, each worker labels its local subset by using the label of its nearest corepoint and then retransmits this information to the master node. Once the master node receives all the labels it reports the final clustering of the collection.

The ongoing work described above has been tested on synthetic and real text data but with reduced size. We expect to be able to test the algorithm with really large collections (more than several hundred of thousands of documents) and also to improve the local workers by exploiting the power of GPU computing.

8 Available Resources

The resources available for this project at the *Escuela de Ingeniera Informtica* of the *Pontificia Universidad Catlica de Valparaso* are:

- University library, free access to electronic libraries such as IEEE and Springer and the personal library of the sponsoring researcher at the previously mentioned campus.
- A room for conferences.
- An office with telephone and fast Internet connection.
- A Dell R730 server (20 cores and 128 GB RAM).

Open access to programming languages such as Julia, Python, Java, C++ and R, and to the Latex document preparation system.

References

[Bahmani et al., 2012] Bahmani, B., Moseley, B., Vattani, A., Kumar, R., and Vassilvitskii, S. (2012). Scalable K-Means ++. Proceedings of the VLDB Endowment (PVLDB), 5:622–633.

[Balcan et al., 2013] Balcan, M. F., Ehrlich, S., and Liang, Y. (2013). Distributed k -Means and k -Median Clustering on General Topologies. Advances in Neural Information Processing Systems 26 (NIPS 2013), pages 1–9.

[Crestani and Markov, 2013] Crestani, F., and Markov, I. (2013). Distributed Information Retrieval and Applications. 35th European Conference on IR Research, 865–868.

[Das et al., 2007] Das, A., Datar, M., Garg, A., and Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering. In Proceedings of the 16th international conference on World Wide Web, pages 271–280. ACM.

[De Vries et al., 2015] De Vries, C. M., De Vine, L., Geva, S., and Nayak, R. (2015). Parallel streaming signature em-tree: A clustering algorithm for web scale applications. In Proceedings of the 24th International Conference on World Wide Web, pages 216–226. ACM.

[Dhillon and Modha, 1999] Dhillon, I. S. and Modha, D. S. (1999). A data-clustering algorithm on distributed memory multiprocessors. LargeScale Parallel Data Mining, 1759(802):245–260.

[Ene et al., 2011] Ene, A., Im, S., and Moseley, B. (2011). Fast Clustering using MapReduce. Kdd, 681–689.

[Ertöz et al., 2003] Ertöz, L., Steinbach, M., and Kumar, V. (2003). Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. Proceedings of the SIAM International Conference on Data Mining, 47–58.

[Forman and Zhang, 2000] Forman, G. and Zhang, B. (2000). Distributed data clustering can be efficient and exact. ACM SIGKDD Explorations Newsletter, 2(2):34–38.

[Han et al., 2011] Han, J., Pei, J., and Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.

[Hubert and Arabie, 1985] Hubert, L. and Arabie, P. (1985). Comparing partitions. Journal of Classification, 2:1, 193–218, Springer.

[Jagannathan et al., 2005] Jagannathan, G., and Wright, R. N. (2005). Privacy-preserving Distributed K-means Clustering over Arbitrarily Partitioned Data. Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, 593–599.

[Januzaj et al., 2003] Januzaj, E., Kriegel, H.-P., and Pfeifle, M. (2003). Towards Effective and Efficient Distributed Clustering. Workshop on Clustering Large Data Sets, pages 49–58.

[Januzaj et al., 2004] Januzaj, E., Kriegel, H.-P., and Pfeifle, M. (2004). Scalable Density-Based Distributed Clustering. pages 231–244.

[Jin et al., 2015] Jin, C., Chen, Z., Hendrix, W., Agrawal, A., and Choudhary, A. (2015). Incremental, Distributed Single-linkage Hierarchical Clustering Algorithm Using Mapreduce. Proceedings of the Symposium on High Performance Computing, pages 83–92.

[Johnson and Kargupta, 2000] Johnson, E. and Kargupta, H. (2000). Collective, hierarchical clustering from distributed, heterogeneous data. Lecture Notes in Computer Science, 1759:221–244.

[Kargupta et al., 2001] Kargupta, H., Huang, W., Sivakumar, K., and Johnson, E. (2001). Distributed clustering using collective principal component analysis. Knowledge and Information Systems, 3(4):422–448.

[Klusch et al., 2003] Klusch, M., Lodi, S., and Moro, G. (2003). Distributed clustering based on sampling local density estimates. IJCAI International Joint Conference on Artificial Intelligence, pages 485–490.

[Kriegel et al., 2005] Kriegel, H.-p., Kr, P., Pryakhin, A., and Schubert, M. (2005). Effective and Efficient Distributed Model-based Clustering.

[Li et al., 2003] Li, T., Zhu, S., and Ogihara, M. (2003). Algorithms for Clustering High Dimensional and Distributed Data. Intelligent Data Analysis Journal, 7(February):1–36.

[Liang et al., 2013] Liang, Y., Balcan, M.-f., and Kanchanapally, V. (2013). Distributed PCA and k-Means Clustering. The Big Learning Workshop in NIPS 2013, pages 1–8.

[Liu et al., 2012] Liu, J., Huang, J. Z., Luo, J., and Xiong, L. (2012). Privacy Preserving Distributed DBSCAN Clustering. Proceedings of the 2012 Joint EDBT/ICDT Workshops, 177–185.

[Liu et al., 2013] Liu, Y., and Li, Z.m and Xiong, H., and Gao, X., and Wu, J., and Wu, S. (2013). Understanding and Enhancement of Internal Clustering Validation Measures. IEEE Transactions on Cybernetics, 43:3, pp. 982–994, June 2013.

[Merugu and Ghosh, 2003] Merugu, S. and Ghosh, J. (2003). Privacy-preserving Distributed Clustering using Generative Models. Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM), pages 0–7.

[Nagwani, 2015] Nagwani, N. K. (2015). Summarizing large text collection using topic modeling and clustering based on MapReduce framework. Journal of Big Data, 2:1–18.

[Naldi and Campello, 2014] Naldi, M. C. and Campello, R. J. G. B. (2014). Evolutionary k-means for distributed data sets. Neurocomputing, 127:30–42.

[Qi et al., 2008] Qi, Z. and Jinze, L., and Wei, W. (2008). Approximate clustering on distributed data streams. Proceedings - International Conference on Data Engineering, 00:1131–1139.

[Raghavan et al., 2007] Raghavan, U., N. and Albert, R. and Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. Physical review, 76(3).

[Rosenberg and Hirschberg, 2007] Rosenberg, A. and Hirschberg, J. (2007). V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure.. Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 7:410–420.

[Rousseeuw, 1987] Rousseeuw Peter J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20:53–65.

[Samatova et al., 2002] Samatova, N. F. and Ostrouchov, G. and Geist, A. and Melechko, A. V. (2002). RACHET: An Efficient Cover-Based Merging of Clustering Hierarchies from Distributed Datasets. Journal of Distributed and Parallel Databases, 11:2, 157–180, Springer.

[Xu et al., 2002] Xu, X., Jäger, J., and Kriegel, H. (2002). A fast parallel clustering algorithm for large spatial databases. High Performance Data Mining, 290:263–290.