Mining Biological Data on the Cloud – A MapReduce Approach

Zafeiria-Marina Ioannou, Nikolaos Nodarakis, Spyros Sioutas, Athanasios Tsakalidis, Giannis Tzimas

Mining Biological Data on the Cloud - A MapReduce Approach

Zafeiria-Marina Ioannou[1], Nikolaos Nodarakis[1], Spyros Sioutas[2], Athanasios Tsakalidis[1], and Giannis Tzimas[3]

[1] Computer Engineering and Informatics Department, University of Patras, 26500 Patras, Greece
{ioannouz,nodarakis,tsak}@ceid.upatras.gr
[2] Department of Informatics, Ionian University, 49100 Corfu, Greece
sioutas@ionio.gr
[3] Computer & Informatics Engineering Department, Technological Educational Institute of Western Greece, 26334 Patras, Greece
tzimas@cti.gr

Abstract. During last decades, bioinformatics has proven to be an emerging field of research leading to the development of a wide variety of applications. The primary goal of bioinformatics is to detect useful knowledge hidden under large volumes biological and biomedical data, gain a greater insight into their relationships and, therefore, enhance the discovery and the comprehension of biological processes. To achieve this, a great number of text mining techniques have been developed that efficiently manage and disclose meaningful patterns and correlations from biological and biomedical data repositories. However, as the volume of data grows rapidly these techniques cannot cope with the computational burden that is produced since they apply only in centralized environments. Consequently, a turn into distributed and parallel solutions is indispensable. In the context of this work, we propose an efficient and scalable solution, in the MapReduce framework, for mining and analyzing biological and biomedical data.

Keywords: Bioinformatics· Data mining· Text mining· Clustering· MapReduce· Hadoop

1 Introduction

The term *data mining* [9] refers to the process of data analysis to identify interesting and useful information and knowledge from huge collections of data. From a more general and more intuitive aspect, data mining is a research field in computer science that employs a wide diversity of well-established statistical and machine learning techniques, such as neural networks, to derive hidden correlations, patterns and trends from large datasets. Bioinformatics, is a prominent domain, among several others, where the existing data mining algorithms are applicable and enhance the process of knowledge discovery. Especially, as

the volume of biological and biomedical data accumulated in large repositories continues to expand at exponential rates, data mining techniques are of critical importance in extracting knowledge efficiently from such datasets. Adding the need to manage heterogeneous data, that in the vast majority are unstructured biomedical text documents (Biomedical Text Mining [1]), and automate the exploration procedure of them, it is easy to understand why data mining plays a fundamental role in bionformatics domain [22].

Despite how efficient a data mining technique might be, as the volume of data collections continues to expand at some point it will be impractical to use due to limits in resources posed by the centralized environment where the algorithm is executed. Typical Biomedical Text Mining tasks include automatic extraction of protein-protein interactions, named entity recognition, text classification and terminology extract. Consider PubMed[4], which is the most widely used biomedical bibliographic text base with millions of records and grows by a rate of 40,000 publications per month. To perform such tasks in an enormous corpus like PubMed is unthinkable. Most existing methods in literature [4, 5, 10, 11, 15, 16] apply to a few hundreds or thousands of records. As a result, high scalable implementations are required. Cloud computing technologies provide tools and infrastructure to create such solutions and manage the input data in a distributed way among multiple servers. The most popular and notably efficient tool is the *MapReduce* [6] programming model, developed by Google, for processing large scale data.

The method proposed in the context of this work, is under development and is an extension of the work in [13] which proposes an automatic and efficient clustering approach that performs well on multiple types of biological and biomedical data. In [13], two mining tools were developed, Bio Search Engine which is a text mining tool working with biomedical literature acquired from PubMed and Genome-Based Population Clustering tool which extracts knowledge from data acquired from FINDbase [8, 20, 21]. FINDbase[5] is an online resource documenting frequencies of pathogenic genetic variations leading to inherited disorders in various populations worldwide. In this paper, we take the data mining technique proposed in [13] one step further and adapt it to the needs of big data analysis. We propose a novel and effective data mining technique for extracting valuable knowledge from biological and biomedical data in the cloud. We focus only on Biomedical Text Mining, since the size of PubMed is adequately big and fit the needs of MapReduce model in contrary to the size of existing genetic and mutation databases (like FINDbase). Our approach uses *Hadoop* [18, 23], the open source MapReduce implementation, and *Mahout* [19] which is a scalable machine learning library built on top of Hadoop.

The remainder of the paper is organized as follows: Section 2 discusses related work, Section 3 provides a full analysis of the algorithm and proceeds into a detail examination of all of its steps and finally Section 4 concludes the paper and presents future steps.

[4] http://www.ncbi.nlm.nih.gov/pubmed/
[5] http://www.findbase.org

2 Related Work

2.1 Biomedical Text Mining

Biomedical Text Mining or BioNLP is the field of research that deals with the automatic processing, retrieval and analysis of scientific texts and more generally literature from the biomedical domain by applying text mining techniques aimed at uncovering previously unknown knowledge. Currently, there has been significant research advances in the area of biomedical text mining, including named entity recognition, text classification, terminology extraction, relationship extraction and hypothesis generation [4].

Text Document clustering, unlike text document classification, is an unsupervised learning process that does not depend on prior knowledge or domain expertise. In particular, document clustering is the task where similar documents are grouped into clusters. Existing biomedical text mining systems that cluster results into topics [15] include GOPubMed[6], ClusterMed[7] and XplorMed[8]. GOPubMed uses both Medical Subject Headings (MeSH) terms and Gene Ontology (GO) in order to organize the search results and, thus, to enhance the user navigation and the search possibilities. It is also capable of sorting results into four categories: "what", "who", "where" and "when". Another prominent example is ClusterMed that also employs clustering in six different ways: i) Title, Abstract and MeSH terms, ii) Title and Abstract, iii) MeSH terms only, iv) Author names, v) Affiliations, vi) Date of publication. XplorMed organizes results by MeSH categories, extracts topic keywords and their co-occurrences and furthermore it provides an interactive navigation through abstracts. For a comprehensive survey of such biomedical text mining systems along with their various characteristics and features, one can consult [4, 5, 15, 16].

It should be noted that besides clustering, there are a handful of other fruitful techniques that have been applied to mine biological text data, that either deviate apart from the bag-of-words model and the tf-idf representation or employ other learning techniques different from clustering. From these approaches, worth to be mentioned are approaches based on the incorporation of semantic information and ontologies in order to correctly disambiguate the meaning of various terms. We could also refer to the employment of second order n-gram Markov models, probabilistic suffix analysis, and named entity recognition. Furthermore special mentioning deserves automatic term recognition techniques, that recognize domain concepts and using automatic term restructuring technique permit the text content organization into knowledge structures (terminologies). More details for these techniques can be found in [1].

2.2 Hadoop and MapReduce Framework

In the context of this work, we focus on the use of *Hadoop* which is an open source framework for managing large-scale data and running computationally heavy

[6] http://www.gopubmed.org/web/gopubmed/
[7] http://clustermed.info/
[8] http://www.ogic.ca/projects/xplormed/

tasks in a parallel and distributed manner. Hadoop follows *MapReduce* principles and is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Here we describe in more details how it works.

A task performing a specific computation in Hadoop is called MapReduce *job*. The input data for each job is split and distributed among the nodes consisting the cluster and then is processed by a number of tasks executed on these nodes. The categories in which these tasks belong are defined as *Map* and *Reduce*. Each Map task receives input data and processes them by calling a user-defined *Map* function which outputs a set of intermediate key-value pairs. After that, a Shuffle process groups all intermediate values associated with the same key I and each group is assigned to the corresponding Reduce task. Each Reduce task also calls a user-defined *Reduce* function which accepts an intermediate key I and a set of values for that key and outputs the final set of key-value pairs. The intermediate values are supplied to the Reduce function via an iterator in order for the system to handle lists of values that are too large to fit in memory. Using the MapReduce programming model, the user does not have to worry about how the program is executed over the cluster or designing fault tolerance protocols in case a node fails. The system handles these issues itself thus allowing the user to focus on his own problem exclusively.

Mahout is a scalable machine learning library that consists of a set of data mining algorithms (e.g. k-means clustering, naive Bayes classifier etc.) implemented in the Hadoop MapReduce framework.

3 The Proposed Mining Algorithm

In the following subsections, we present in detail the steps of the proposed data mining technique. The solution consists of a combination of MapReduce jobs that run either on the Hadoop or Mahout framework. The basic steps of the algorithm are: 1) Preprocessing of the data collection in order to represent them in a more processable structure, 2) Latent Semantic Indexing and 3) Coarse Clustering in an attempt to generate an initial partition of data, 4) Agglomerative Hierarchical Clustering [17] in order to reduce the number of the initial clusters (in case that the number is relatively large) and 5) spherical k-means algorithm [7, 14] in order to enhance the quality of the proposed clustering. Each of the aforementioned steps is a MapReduce job (or a series of MapReduce jobs) and the output of each MapReduce job feeds the input of the next job in the sequence. Note that we combine hierarchical and non-hierarchical clustering techniques to improve the efficiency and accuracy of clustering [3, 24]. An overview of the algorithmic process is depicted in Figure 1.

3.1 Preprocessing the Data Collection

Data preprocessing is an essential step in the data mining process that includes feature selection and representation of data. The data elements have to be transformed into a representation suitable for computational use. For this purpose,

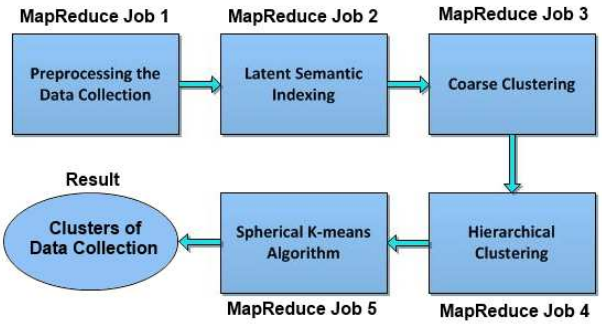

Fig. 1. Overview of MapReduce Clustering Algorithm

we display a method for preprocessing the corpus of PubMed which constitutes the input dataset of our mining method.

The bag-of-words model is the most widely used document representation adopted in the text clustering algorithms. According to the principles of this model, each document is represented as a sequence of terms/words. Initially, we parse each document and distinguish the lexical units (tokens) that constitute it. After this, we proceed two kinds of feature reduction. At first, we remove the stopwords. A stopword is defined as a term which is not thought to convey any meaning as a dimension of vector space. Typically, a compilation of stopwords consists of very common words such as articles, preposition etc. By removing them, we achieve a reduction in the dimensionality of the index by 20-30%. The second step involves Part-Of-Speech Tagging (POS Tagging) and lemmatization. POS Tagging is the process of assigning a particular part of speech (e.g. noun, verb, adjective, etc.) to each term of a document while lemmatization is the process of reducing the words to their basic form (lemma). Consequently, we end up with the unique lemmas of noun words of all documents. To put through the POS Tagging process we use the GENIA Tagger[9] which is specifically tuned for biomedical texts.

Subsequently, according to the vector space model [2], each document is represented as a vector of m dimensions, where m is the number of unique lemmas. For each document, we measure the significance of each lemma in its content using the TF-IDF (Term Frequency - Inverse Document Frequency) scheme. The term frequency is simply the number of times a lemma appears in a document, whereas the inverse document frequency is a measure obtained by dividing the total number of documents by the number of the documents containing the term. So, more formally, we define $TF-IDF(t,d,D) = TF(t,d) \cdot IDF(t,D)$, where $IDF(t,D) = \log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right)$. Consequently, lemmas that appear frequently in a document but have low document frequency in the whole data collection are given a high value in the TF-IDF scheme.

[9] http://www.nactem.ac.uk/tsujii/GENIA/tagger/

In order to preprocess the data in the way we described, we developed a MapReduce Job. The Map function takes as input the data collection of PubMed, proceeds to POS Tagging of the terms and extracts the lemmas of noun words. For each lemma, it outputs a key-value pair in the form <lemma, d, 1>, where lemma is the key and the value composes of number 1 and the name of document that contains the lemma. The Reduce function retrieves the unique lemmas and calculates the TF-IDF value for each document they belong. Next, it outputs key-value pairs in the format <lemma, tfidf, d>, where lemma is the key and the value composes of the TF-IDF weight and the name of the document that is binded with the lemma and the TF-IDF value. The overall outcome of this job is a term-document matrix A of dimension $m \times n$, where m is the number of unique lemmas and n is the number of documents.

3.2 Latent Semantic Indexing

Latent Semantic Indexing (LSI) [2] is a prominent indexing and retrieval method for extracting hidden relationships between terms and concepts contained in large, unstructured collections of documents. The main idea of the method is the projection of document vectors in a new low-dimensional space through the Singular Value Decomposition (SVD) of the term-document matrix A. This can significantly reduce the number of computations needed in the next steps of our data mining algorithm.

The Singular Value Decomposition of an $m \times n$ matrix A of rank r expresses A as a product of three simpler matrices, $A = USV^T$ where $S = diag(\sigma_1, ..., \sigma_r)$ is a diagonal matrix containing the set of singular values, $U = (u_1, ..., u_r)$ is an $m \times r$ matrix whose columns are orthonormal and $V = (v_1, ..., v_r)$ is an $n \times r$ matrix which is also column-orthonormal. LSI omits all but the k largest singular values in the above decomposition, for some appropriate k which will be the dimension of the low-dimensional space referred to in the description above. It should be small enough to enable fast retrieval and large enough to adequately capture the structure of the corpus. Consequently, after SVD and dimension reduction, a matrix $A_k = U_k S_k {V^T}_k$ is created, where $S_k = diag(\sigma_1, ..., \sigma_k)$, $U_k = (u_1, ..., u_k)$ and $V_k = (v_1, ..., v_k)$.

To this end, we utilize the Singular Value Decomposition algorithm implemented in Mahout. More specifically, Mahout implements a version of Lanczos algorithm[10] as a series of MapReduce jobs. We provide as input to the algorithm the matrix A, the number of row and columns and the rank of the output matrix and it produces the V_k and S_k matrices, needed for the next step of our solution.

3.3 Coarse (initial) Clustering

Fuzzy Clustering [12] has been widely used in the area of information retrieval and data mining. In traditional Fuzzy Clustering Methods, data elements can belong to more than one clusters, and each data element is associated to every

[10] http://en.wikipedia.org/wiki/Lanczos_algorithm

cluster based on a membership function, that differentiates on the specific cluster. In our approach, we interpret the results of the LSI transformation of the initial matrix A as a form of fuzzy clustering and then we transform the fuzzy clustering to a hard-crisp clustering by assigning each document to exactly one cluster, the cluster where the document has the highest degree of participation. In this way, we produce an initial rough clustering of documents. Later these initial clusters are going to be reduced (Agglomerative Hierarchical Clustering) and furthermore re-organized (spherical k-means algorithm) in order to enhance their quality.

In particular, we rely on the $n \times k$ matrix $V_k S_k$, produced by SVD in the previous step, and consider its k columns as a set of k clusters and the n rows as documents. Each value in position (i, j) of $V_k S_k$ defines the document's i degree of participation to the cluster j. This fuzzy clustering is transformed to a crisp clustering by assigning each document to exactly one cluster, where the document has the highest degree of participation according to the values of the $V_k S_k$.

To perform the fuzzy clustering we run the respective algorithm implemented in the Mahout framework. The algorithm works as follows:

- Each document is assigned to the cluster where it has the highest degree of participation according to the membership values of $V_k S_k$.
- Each initial cluster is represented by a central vector c [17], called centroid, where $c = \frac{1}{|S|} \sum_{x \in S} x$.

More specifically, the Map function reads the probability membership values of documents and assigns them to the corresponding cluster. The records produced by the Map function are key-value pairs where the key is a cluster identifier and the value is a vector that represents a row of the matrix A. The Reduce function receives all the key-value pairs from the map task and produces a centroid for each cluster. The key of the output record is a cluster identifier and the value is the centroid of the cluster.

3.4 Hierarchical clustering

Hierarchical Clustering [17] is a widely used data analysis method for identifying relatively homogenous clusters of experimental data items based on selected measured characteristics. Commonly, hierarchical clustering techniques generate a set of nested clusters, with a single all-inclusive cluster at the top and single point clusters at the bottom. The result of a hierarchical clustering algorithm can be graphically displayed as tree, called a dendrogram.

Hierarchical clustering algorithms can be divided into two basic approaches: agglomerative (merging) and divisive (splitting). In the context of this work, we adopt the agglomerative approach in an attempt to reduce the number of initial clusters, and develop an iterative algorithm in the Hadoop framework. The agglomerative approach is a bottom-up clustering method that starts with all the data elements as individual clusters and at each step it merges the most similar or closest pair of clusters, based on a cluster similarity or distance measure. The

steps are repeated until the desired number of clusters is obtained according to a user defined parameter.

The algorithm consists of an iterative step with two phases. Phase 1 computes similarity between the clusters and phase 2 merges the two most similar clusters. The process repeats until the above prerequisite is fulfilled. The agglomerative approach attempts to shrink the number of initial clusters produced by LSI and fuzzy clustering since the produced number of clusters is equal to the LSI dimension which is generally high. At each step, the algorithm merges the most similar pair of clusters based on the UPGMA (Unweighted Pair Group Method with Arithmetic Mean) scheme:

$$similarity(cluster1, cluster2) = \frac{\sum_{x \in cluster1, y \in cluster2} \cos(x, y)}{size(cluster1) \cdot size(cluster2)}$$

3.5 Spherical k-means Algorithm

The k-means algorithm is an efficient and well known method for clustering large portions of data. Typically, the k-means algorithm determines the distance (similarity) between a data element in a cluster and its cluster centroid by using the squared Euclidean distance measure. The spherical k-means algorithm [14] is a variant of the original method that uses the cosine similarity as distance measure and achieves better results on high-dimensional data such as text documents. Note however that the above mentioned algorithm is significantly sensitive to the initialization procedure meaning that its clustering performance depends heavily on the initial selection of cluster centroids. As a consequence, this selection constitutes a task of crucial importance. In an attempt to overcome this issue and with the goal of enhancing the achieved performance, we initialize the spherical k-means algorithm with the cluster centroids obtained by the previous steps of the proposed technique. Furthermore, the motivation for applying the spherical k-means as the final step of our clustering approach, is to enhance the quality and the precision of the clustering produced by the previous steps of the proposed technique.

More specifically, we utilize a MapReduce version of a refinement algorithm [14] that uses local search in order to refine the clusters generated by spherical k-means. The algorithm alternates between two steps (a) first variation and (b) spherical k-means. The first step moves a single data element from one cluster to another increasing in this way the value of the objective function of the clustering. A sequence of first variation moves allows an escape from a local maximum, so that fresh iterations of spherical k-means (in the second step) can further increase the objective function value. The k-means algorithm is implemented in the Mahout framework and we provide the cosine similarity as input parameter, in order for the method to behave similar to the spherical k-means approach.

4 Conclusions and Future Work

In this paper, we present a novel data mining algorithm for clustering biological and biomedical data in the cloud. The method constitutes of a series of steps/algorithms and is currently under development. We utilize the Hadoop and Mahout frameworks to implement existing centralized data mining algorithms that fit the MapReduce programming model. In the near future, we intend to finish the implementation of our approach and run extensive experiments using the enormous data repository of PubMed. We want to compare the efficiency of our method with existing centralized methods and measure the gain we earn using parallel and distributed solutions instead of centralized ones. Moreover, we plan to explore more existing data mining techniques and create respective MapReduce versions of them to experiment with.

Acknowledgements. This research has been co-financed by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Thales. Investing in knowledge society through the European Social Fund.

References

1. Ananiadou, S., Mcnaught, J.: Text Mining for Biology and Biomedicine. Artech House (2006)
2. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval: The Concepts and Technology behind Search, 2nd edn. ACM Press (2011)
3. Chen, B., Harrison, R., Pan, Y., Tai, P.: Novel Hybrid Hierarchical-K-means Clustering Method (H-K-means) for Microarray Analysis. In: Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference - Workshops. IEEE Computer Society, Washington, DC, USA, pp. 105–108 (2005)
4. Cohen, A.M., Herch, W.R.: A Survey of Current Work in Biomedical Text Mining. Brief Bioinform 6, 57–71 (2005)
5. Dai, H.J., Lin, J.Y.W., Huang, C.H., Chou, P.H., Tsai, R.T.H., Hsu, W.L.: A Survey of State of the Art Biomedical Text Mining Techniques for Semantic Analysis. In: Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing, pp. 410-417 (2008)
6. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: Proceedings of the 6th Symposium on Operating Systems Design and Implementation, pp. 137–150. USENIX Association, Berkeley, CA, USA (2004)
7. Dhillon, I.S., Guan, Y., Kogan, J.: Iterative Clustering of High Dimensional Text Data Augmented by Local Search. In: Proceedings of the 2002 IEEE International Conference on Data Mining, pp. 131–138 (2002)
8. Georgitsi, M., Viennas, E., Gkantouna, V., Christodoulopoulou, E., Zagoriti, Z., Tafrali, C., Ntellos, F., Giannakopoulou, O., Boulakou, A., Vlahopoulou, P., Kyriacou, E., Tsaknakis, J., Tsakalidis, A., Poulas, K., Tzimas, G., Patrinos, G.: Population-Specific Documentation of Pharmacogenomic Markers and their Allelic Frequencies in FINDbase. Pharmacogenomics 12, 49–58 (2011)

9. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann Publishers, San Francisco (2006)
10. Ioannou, M., Makris, C., Tzimas, G., Viennas, E.: A Text Mining Approach for Biomedical Documents. In: Proceedings of the 6th Conference of the Hellenic Society for Computational Biology and Bioinformatics, Patras, Greece (2011)
11. Ioannou, M., Patrinos, G., Tzimas, G.: Genome-based Population Clustering: Nuggets of Truth Buried in a Pile of Numbers?. In: Proceedings of the 1st Workshop on Algorithms for Data and Text Mining in Bioinformatics organized in the 8th Artificial Intelligence Applications and Innovations Conference, Halkidiki, Greece (2012)
12. Inoue, K., Urahama, K.: Fuzzy Clustering Based on Cooccurence Matrix and Its Application to Data Retrieval. Electron. Comm. Jpn. Pt. 2 84, 10–19. (2001)
13. Ioannou, M., Makris, C., Patrinos, G., Tzimas, G.: A Set of Novel Mining Tools for Efficient Biological Knowledge Discovery. In: Artificial Intelligence Review, Springer (2013)
14. Kogan, J.: Introduction to Clustering Large and High-Dimensional Data. Cambridge University Press, New York, pp. 51–72 (2007)
15. Lu, Z.: Pubmed and Beyond: A Survey of Web Tools for Searching Biomedical Literature. Database, Oxford (2011)
16. Manconi, A., Vargiu, E., Armano, G., Milanesi, L.: Literature Retrieval and Mining in Bioinformatics: State of the Art and Challenges. In: Adv. Bioinformatics (2012)
17. Steinbach, M., Karypis, G., Kumar, V.: A Comparison of Document Clustering Techniques. In: Proceedings of the KDD Workshop on Text Mining, 6th ACM SIGKDD International Conference on Data Mining (2000)
18. The apache software foundation: Hadoop homepage, `http://hadoop.apache.org/`
19. The apache software foundation: Mahout homepage, `https://mahout.apache.org/`
20. Van Baal, S., Kaimakis, P., Phommarinh, M., Koumbi, D., Cuppens, H., Riccardino, F., Macek, M. Jr, Scriver, C.R., Patrinos. G.: FINDbase: A Relational Database Recording Frequencies of Genetic Defects Leading to Inherited Disorders Worldwide. Nucleic Acids Res 35 (2007)
21. Viennas, E., Gkantouna, V., Ioannou, M., Georgitsi, M., Rigou, M., Poulas, K., Patrinos, G., Tzimas, G.: Population-Ethnic Group Specific Genome Variation Allele Frequency Data: A Querying and Visualization Journey. Genomics 100, 93–101 (2012)
22. Wang, J.T.L., Zaki, M.J., Toivonen, H.T.T., Shasha, D.: Data Mining in Bioinformatics. In: Advanced Information and Knowledge Processing. Springer (2005)
23. White, T.: Hadoop: The Definitive Guide, 3rd Edition. O'Reilly Media / Yahoo Press (2012)
24. Zhang, C., Xia, S.: K-means Clustering Algorithm with Improved Initial Center, Knowledge Discovery and Data Mining, pp.790–792 (2009)
25. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an Efficient Data Clustering Method for Very Large Databases. In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, pp. 103–114 (1996)
26. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: a New Data Clustering Algorithm and its Applications. Journal of Data Mining and Knowledge Discovery 1, 141–182 (1997)