2014 IEEE 26th International Conference on Tools with Artificial Intelligence

Strategies for Big Data Clustering

Olga Kurasova, Virginijus Marcinkevičius, Viktor Medvedev, Aurimas Rapečka, and Pavel Stefanovič
Vilnius University, Institute of Mathematics and Informatics
Akademijos str. 4, LT-08663 Vilnius, Lithuania
{*olga.kurasova, virginijus.marcinkevicius, viktor.medvedev, aurimas.rapecka, pavel.stefanovic*}*@mii.vu.lt*

***Abstract*—In the paper, an overview of methods and technologies used for big data clustering is presented. The clustering is one of the important data mining issue especially for big data analysis, where large volume data should be grouped. Here some clustering methods are described, great attention is paid to the k-means method and its modifications, because it still remains one of the popular methods and is implemented in innovative technologies for big data analysis. Neural network-based self-organizing maps and their extensions for big data clustering are reviewed, too. Some strategies for big data clustering are also presented and discussed. It is shown the data of which volume can be clustered in the well known data mining systems WEKA and KNIME and when new sophisticated technologies are needed.**

***Keywords*-big data; clustering methods; data mining; Hadoop**

I. Introduction

Not so long ago, data sets consisted of some hundreds of items. Nowadays technologies are able to store and process data ever a larger and larger amount of data. The data of this kind are called big data. Big data is the term for a collection of data sets so large and complex that it becomes difficult to process using traditional data processing tools. Big data can be characterized by three V's: volume (large amounts of data), variety (includes different types of data), and velocity (constantly accumulating new data) [1]. Data become big when their volume, velocity, or variety exceed the abilities of IT systems to store, analyse, and process them. Recently, widely understanding are being more popular by adding two additional V. Big data can be summarized by five V's – Volume, Velocity, Variety, Veracity, Value [2]. Big data are not just about lots of data, they are actually a new concept providing an opportunity to find a new insight into the existing data.

There are many applications of big data: business, technology, telecommunication, medicine, health care, and services, bioinformatics (genetics), science, e-commerce, finance, the Internet (information search, social networks), etc. Some sources of big data are actually new. Big data can be collected not only from computers, but also from billions of mobile phones, social media posts, different sensors installed in cars, utility meters, shipping and many other sources. In many cases, data are just being generated faster than they can be precessed and analysed.

Big data can include both unstructured and structured data. Unstructured data are the data that either do not have a pre-defined data model or are not organized in a pre-defined manner. Structured data are relatively simple and easy to analyse, because usually the data reside in databases in the form of columns and rows. The challenge for scientists is to develop tools to transform unstructured data to structured ones.

Often a structured data set X consists of data items $X_1, X_2, \ldots, X_m$ described by the features $x_1, x_2, \ldots, x_n$, where m is the number of items, n is the number of features. So, $X = \{X_1, X_2, \ldots, X_m\} = \{x_{ij},\ i = 1, \ldots, m,\ j = 1, \ldots, n\}$, where x_{ij} is the jth feature value of the ith object. In the case of big data, m and n are large enough. If the number of features n is high, the data is called the high dimensional data. The clustering of high dimensional data is useful solving dimensionality reduction as well as visualization problems [3], [4].

Big data bring new challenges to data mining because large volumes and different varieties must be taken into account. The common methods and tools for data processing and analysis are unable to manage such amounts of data, even if powerful computer clusters are used. To analyse big data, many new data mining and machine learning algorithms as well as technologies have been developed. So, big data do not only yield new data types and storage mechanisms, but also new methods of analysis.

When dealing with big data, a data clustering problem is one of the most important issues. Often data sets, especially big data sets, consist of some groups (clusters) and it is necessary to find the groups. Clustering methods have been applied to many important problems [5], for example, to discover healthcare trends in patient records, to eliminate duplicate entries in address lists, to identify new classes of stars in astronomical data, to divide data into groups that are meaningful, useful, to cluster millions of documents or web pages. To address these applications and many others a variety of clustering algorithms has been developed. There exist some limitations in the existing clustering methods, most algorithms require scanning the data set for several times, thus they are unsuitable for big data clustering. There is a lot of applications in which extremely large or big data sets need to be explored, but which are much too large to be processed by traditional clustering methods. The goal of

1082-3409/14 $31.00 © 2014 IEEE
DOI 10.1109/ICTAI.2014.115

this paper is to overview the methods and technologies used in order to cluster big data and to describe strategies for big data analysis.

II. Clustering Methods for Big Data

The definition of a clustering problem is as follows: given a data set $X = \{X_1, X_2, \ldots, X_m\}$ and an integer value k, the clustering problem is to define a mapping $f : X \mapsto \{1, \ldots, k\}$, where each item X_l, $l \in \{1, \ldots, m\}$ is assigned to one cluster $K_j, j = 1, \ldots, k$. A cluster K_j contains the items mapped to it: $K_j = \{X_l | f(X_l) = K_j, 1, \ldots, m,$ and $X_l \in X\}$. An item within a cluster is more similar to items within that cluster than it is similar to items outside it [6]. Usually one of Minkowski distances (eg. Euclidean distances) is used as a similarity measure of clusters. Various clustering methods have been developed: k-means [7], self-organizing maps [8], etc.

A. K-means Method and its Modifications

One of the most popular clustering methods is k-means. At first, the number k of desired clusters is selected and initial values of cluster centres are assigned. Then each data item is assigned to the cluster with the closest centres and new centres for each cluster are computed. The steps are repeated iteratively until stop or convergence criterion is satisfied. The convergence criterion can be based on the squared error (mean difference between the cluster centres and the items assigned to the clusters). The stop criterion can be a high number of iteration steps.

Over the past years, various extensions of the classical k-means algorithm have been developed, for example, kernel k-means [9], spherical k-means [10], Minkowski metric weighted k-means [11], fuzzy c-means [12] etc. The majority of them is modified to speed up calculations or for specific tasks. Due to its low computational cost and easily parallelized process, the classical k-means algorithm is well known for its efficiency in clustering large data sets, but some modifications of k-means are introduced as very specific tools for big data analysis.

In [13], the X-means method has been proposed to extend k-means with efficient estimation of the number of clusters. Here, the number of clusters is optimized using the Bayesian information criterion.

The classical k-means clustering was designed for solving single-view data clustering problem. In [14], a new robust multi-view k-means clustering method was proposed to integrate heterogeneous features for clustering.

Some k-means modifications for stream data are introduced [15]. The streaming k-means algorithm for well-clusterable data is published in [16]. The main k-means problem is where the data are too large to be stored in the main memory and must be accessed sequentially. In [17], several improved algorithms of Euclidean k-means are designed for stream data. Mainly, there are some simplifications of algorithm [16] (eg., an improved new manner by which the algorithm determines a better facility cost as the stream is processed, removing some unnecessary checks, etc.) and these simplifications determine that the new algorithm is more suitable for the analysis of large data sets as the previous one.

B. Self-organizing Maps and their Extensions

The self-organizing map (SOM) is a class of neural networks that are trained in an unsupervised way, using a competitive learning [8]. SOM is used for both clustering and visualization of data [18]. It is a set of nodes (grid), connected to each other via a rectangular or hexagonal topology. Sometimes the nodes are called neurons. The connections between the inputs and the nodes have weights, so a set of weights corresponds to each node. The set of weights forms a reference vector $M_{ij}, i = 1, \ldots, r, j = 1, \ldots, s$. So, the rectangular SOM is a two-dimensional array of neurons $M = \{M_{ij}, i = 1, \ldots, r, j = 1, \ldots, s\}$. Here r is the number of rows, and s is the number of columns. At each SOM learning step, an input vector (data item) $X_l \in \{X_1, X_2, \ldots, X_m\}$ is presented to the map. The Euclidean distances between X_l and each reference vector M_{ij} are calculated and the neuron, whose reference vector is closest to X_l, is designated as a winning neuron. The components of reference vectors M_{ij} are changed according to a learning rule: $M_{ij}(t+1) = M_{ij}(t) + h^c_{ij}(t)(X_l - M_{ij}(t))$, where t is the number of iteration, h^c_{ij} is the so-called neighbourhood function. The learning steps are repeated until the maximum number of iterations is attained. When the learning is completed, the winning neurons are defined for all data items $X_l, l = 1, \ldots, m$. The data items are distributed on the map forming some clusters [18].

Over the past decade many modifications and extensions of SOM have been created: merge self-organizing map (MSOM) [19], recursive self-organizing map (RecSOM) [20], WEBSOM [21], etc. Mostly all of them are created to speed-up the learning algorithm or to perform a specific tasks. For example, WEBSOM is the first SOM extension created for the textual document analysis.

Now a lot of researchers are still using SOM for different problem solutions. One of the newest SOM modifications is the batch-learning self-organizing map (BLSOM) which is used in the bioinformatics area [22]. In this method, SOM has been modified for genome informatics to make the learning process and resulting map independent of the data input. BLSOM is a powerful tool for big data analysis. It allows us to visualize and classify big sequences, obtained from genomes (millions of metagenomic sequences).

Another SOM modification for a large data set is an environment self-organizing map (EnvSOM) [23]. The EnvSOM algorithm consists of two phases. In the first phase, a SOM is trained using all the data features, but only environment features of the data are used to find a

neuron winner. In the second phase, a new SOM is created appropriately with information from the reference vectors of the first phase SOM. In this phase, SOM uses all the data set features for neuron winner computation. Thus, in this method, self-organizing map is influenced by environment conditions.

Some researchers combine self-organizing maps with the modified k-means algorithms to solve high dimensional data problems [24]. The main steps of the method are as follows: (1) SOM is used to reduce the dimensionality of the data and to determine the number of clusters; (2) the genetic algorithm is applied to the reduced dimensionality data in order to obtain the initial centres of the clusters; (3) the k-means algorithm is used to get the resultant clusters. Thus, in this method, SOM is used for dimensionality reduction and visualization.

Also, we can find SOM extensions with unusual visualization tools suitable for unstructured data. This visualization method [25] helps us to analyse several features at once, so it is much more suitable for big data visual analysis. As a result, we get SOM as a spider graph, where we can find a large number of analysed features in each graph.

One more modification for large data set clustering is the growing hierarchical self-organizing map (GHSOM). It can be used for high dimensional data analysis. The GHSOM method [26] can be used for different kinds of information clustering: textual, numerical, web pages, etc.

The self-organizing maps are mostly used for high dimensional data analysis. A lot of researchers use SOM as a visualization tool [27].

C. Other Clustering Methods

Cluster analysis was originated in the first half of the 20th century. Thus, various clustering methods have been developed. In this subsection, some popular methods are discussed.

Hierarchical clustering is a method which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types: (1) agglomerative – at first each data item corresponds to a cluster, then pairs of clusters are merged; (2) divisive – at first all data items are assigned to one cluster, then it splits recursively. To decide which clusters should be merged in the agglomerative case and which cluster should be split in the divisive case, a measure of dissimilarity between data items is required. The results of hierarchical clustering are usually presented in a dendrogram.

Balanced iterative reducing and clustering using hierarchies (BIRCH) have been proposed in [28]. The algorithm creates a so-called clustering feature tree that captures needed information in order to perform clustering. The existing agglomerative hierarchical clustering algorithm is used to cluster all leaf items of the tree. A set of clusters is obtained that captures major distribution patterns in the data.

In [29], the DBSCAN (density-based spatial clustering of application with noise) algorithm has been proposed. The key idea of the DBSCAN algorithm is that, for each data item of a cluster, the neighbourhood of a given radius has to contain at least a minimum number of data items, i.e., the density in the neighbourhood has to exceed some predefined threshold. This algorithm needs two input parameters: the minimum number of data items in any cluster and the threshold value of distance that delimits the neighbourhood area of a data item. The number of clusters, k, is determined by the algorithm itself. DBSCAN as well as k-means methods are implemented in the popular data mining systems, eg. WEKA, RapidMiner.

The OPTICS (ordering points to identify the clustering structure) algorithm has been developed in order to avoid one disadvantage of DBSCAN. The algorithm allows us to detect meaningful clusters in data of varying density [30].

Clustering using representatives (CURE) uses a constant number of representative items to represent a cluster [31]. At each step in the algorithm, clusters with the closest pair of representative items are chosen to be merged. Distance is measured as the closest pair of representative items that belong to different clusters. Then the representative items are shrunk toward the cluster centre. Clusters of unusual shapes can be better represented with multiple representative items than with only one, eg., center of the cluster.

Expectation minimization (EM) can be used for clustering data, too [32]. It is an iterative algorithm that is used in problems where data are incomplete or considered incomplete. Unlike the distance-based methods (such as k-means), EM is known to be an appropriate optimization algorithm for constructing proper statistical models of the data. EM is widely used in applications such as computer vision, speech processing, and pattern recognition. The algorithm consists of two steps – expectation (E) and maximization (M) – which are performed iteratively until some form of convergence is reached. The probability of each data item belonging to each cluster is estimated in the E-step. The M-step re-estimates the parameter vector of the probability distribution of each cluster. The algorithm terminates when the distribution parameters converge or reach the maximum number of iterations [33].

The idea of using canopies (overlapping cluster subsets defined through computationally cheap approximate distance measures) to reduce computational costs and maintain the accuracy in clustering large data sets has been presented in [5]. The canopy-based approach uses smart data division and subsequent aggregation to achieve clustering efficiencies.

In [34], a clustering-based support vector machine (CB-SVM) method has been presented, which is designed for handling very large data sets. CB-SVM applies a

hierarchical micro-clustering algorithm that scans the entire data set only once to provide an SVM with high quality items that carry the statistical summaries of the data such that the summaries maximize the benefit of learning the SVM.

A scalable method to cluster data sets too large to fit in memory is presented in [35]. The clustering algorithm is piecemeal principal direction divisive partitioning (PMPDDP), in which the original data are broken up into sections which will fit into memory and be clustered. The cluster centres are used to create approximations to the original data items, and each original data item is represented by a linear combination of these centres.

DESCRY is a method to identify clusters in a large high dimensional data set having a different size and shape [36]. DESCRY discovers clusters having a different shape, size, and density and when data contain noise by finding and clustering a small set of points, called meta-points, that well depict the shape of clusters, present in the data set. Final clusters are obtained by assigning each point to one of the partial clusters.

III. Technologies for Big Data Clustering

If the data sets to be clustered are not so big, the well known data mining systems (analytics) can be used. In the WEKA system (http://www.cs.waikato.ac.nz/ml/weka), the following clustering methods are implemented: simple k-means, X-means, DBSCAN, OPTICS, EM, hierarchical clustering, and some other. K-means, fuzzy c-means, hierarchical clustering, self organizing tree algorithm (SOTA) are implemented in the KNIME system (http://knime.com). In the RapidMiner system, the k-means method and its two modifications, X-means, k-medoids, DBSCAN, EM, SOM, and some other methods are implemented (http://rapidminer.com). K-means, hierarchical clustering, and SOM are developed in the Orange system (http://orange.biolab.si).

Big data concept does impact on the current data analytics. In response to the demand for platforms suited to big data analytics, vendors have released a slew of new product types including analytic databases, data warehouse appliances, columnar databases, no-SQL database, distributed file systems, and so on [37]. Nowadays, the most popular analytics are still working with data warehouses (DW) or enterprise data warehouses (EDW). In DW, the data are stored in a structured form. Therefore, relational database management systems (RDBMS) can manage the storage of such data.

When we are dealing with big data, we are dealing with known big data challenges such as data velocity, data volume and data variety. EDW and Hadoop technologies can helpful to manage these challenges. Apache Hadoop is an open source release of a technology that preceded just almost every data storage and analytics tool that has since been labelled 'big data' (http://hadoop.apache.org).

Table I
Comparison of RDBMS and Hadoop

	RDBMS	Hadoop/MapReduce
Data Size	Gigabytes (Terabytes)	Petabytes (Hexabytes)
Access	Interactive and Batch	Batch
Updates	Read/Write many times	Write once/read many times
Structure	Static Schema	Dynamic Schema
Integrity	Hight	Low
Query Response Time	Can be nearly immediate	Has latency (due to batch processing)

With Hadoop it is possible to build easily and cost effectively very large scale distributed data storage and data processing solutions using low cost servers and low cost networking hardware. The Hadoop File System (HDFS) allows you to send data into Hadoop and then works as if your data are simultaneously on all the disks and all the servers in the cluster. In the cluster we have multiple computers, so Hadoop provides a new approach to distributed computing by implementing an idea called MapReduce. MapReduce is essentially a programming model for processing massive data sets with a parallel distributed algorithm that allows for splitting, processing and aggregation of large data sets. Comparing to traditional relational database management systems (RDBMS), Hadoop has problems with a query response time and integrity with other products like data analytics (see Table I).

Companies that have been developing traditional RDBMS, EWS or Business intelligence analytics for these databases have another approach in the form of distributed query processing. The computational approach to distributed query processing is called Massively Parallel Processing (MPP). In MPP, processing of data is distributed across a bank of compute nodes, these separate nodes process their data in parallel and the node-level output sets are assembled together to produce a final result. MPP is employed in high-end data warehousing appliances. Almost all of these products started out as offerings from pure-play companies and later most of them have been assimilated into the mega-vendor world (eg. Netezza was acquired by IBM, Vertica by HP, Greenplum by EMC and Microsofts acquisition of DATAllegro resulted in an MPP version of SQL Server, called Parallel Data Warehouse Edition (SQL PDW, Microsoft Polybase)). In Apache Hadoop, simillar MPP solutions are Apache Hive, Apache Pig, Cloudera Impala.

By looking at business intelligence aspects of analytics over big data, the state-of-the art research result is represented by Hive [38], a business intelligence system/tool for querying and managing structured data, built on the top

of Hadoop HDFS. Hive allows us to obtain the final analytics components (in the form of diagrams, plots, dashboards, and so on) from the big data processed materialized, and stored via Hadoop. Also, Hive introduces a SQL-like query language, called HiveQL [38], which runs MapReduce jobs immersed into SQL statements.

To process the big data analysis, new analytics are created. Apache Mahout, MADlib, SQL-MapReduce, Apache Drill, etc. Apache Mahout is a library with extended learning system capabilities and data mining algorithms such as: clustering, classification, collaborative filtering and frequent pattern mining. The core of clustering, classification, collaborative filtering algorithm realizations is based on the MapReduce paradigm. Apache Mahout currently has implementation of k-means, canopy, fuzzy k-means, EM, hierarchical and some other clustering algorithms (http://mahout.apache.org).

MADlib is an open source library for scalable in-database analytics (http://madlib.net). It provides data-parallel implementations of mathematical, statistical and machine learning methods for structured and unstructured data. Teradata provides a pre-built library of patented SQL-MapReduce functions, accessible via SQL, for data acquisition, data preparation, analysis and visualization (http://www.teradata.com). It provides a broad range of analytical techniques such as SQL, MapReduce, statistical, text analytics, graph, etc. in a single platform. Apache Drill is a distributed system for an interactive analysis of large-scale datasets, based on Google's Dremel [39]. Its goal is to efficiently process a nested data on the scale of 10000 servers or more and to be able to process petabytes of data and trillions of items in seconds.

Radoop is powered by RapidMiner, the popular open source data mining tool, and it provides a simple and flexible data flow interface for defining big data analytics processes (http://www.radoop.eu). It integrates with the existing Hadoop clusters and works seamlessly with many different Hadoop distributions: Apache Hadoop, Cloudera Distribution including Apache Hadoop (CDH) and the Hortonworks Data Platform (HDP). Radoop has an easy-to-use data flow interface for analytics, clustering, and visualization of big data.

One of the technologies for big data analysis have been proposed by a Pervasive DataRush. It is possible to use DataRush on open source software KNIME and Hadoop. With Pervasive DataRush for KNIME complex workflows can process much larger data sets and the process of data become 2-10 times faster on the same hardware (http://bigdata.pervasive.com/Products/RushAccelerator-for-KNIME.aspx). Users have access to a library of scalable, high-throughput DataRush nodes to tackle big data analysis.

Technology for mining big data streams is SAMOA (Scalable Advanced Massive Online Analysis) framework [40]. It features a pluggable architecture that allows it to run on several distributed stream processing engines. SAMOA includes distributed algorithms for the most common machine learning tasks such as classification and clustering. It includes distributed versions of classical streaming algorithms such as Hoeffding decision trees and k-means-based clustering.

IV. Data Clustering using Different Strategies

Although the Internet is full of information on big data, however, there is a lack of systematized information about which methods and techniques to use for a big data analysis. In this section, strategies of big data clustering is presented. We also show in which cases the usual data mining systems are enough, and where more sophisticated technologies should be used. In Fig. 1, the schema of strategies for big data clustering is presented. Suppose we have a data set that would be clustered. If the number of the data items does not exceed m', we can use the well known data mining systems. Usually, the number m' varies from a few thousands to several millions depending on computational resources of a personal computer. In this case, we deal with large data. If the data set is bigger, systems and technologies based on parallel and distributed computing should be used. There are two cases: if the number of data items does not exceed m'' ($m'' \approx 100$ millions), the data can be clustered using the data mining systems which support possibilities to perform computations on Grids and computer clusters; otherwise (when m'' is very huge and the data exceeds a terabyte), we deal with big data and Hadoop based technologies and libraries should be used for data clustering. It should be noted that these strategies can be applied not only for data clustering, but also for solving other data mining problems.

Two well known data mining systems WEKA and KNIME are used in order to show the data of which volume can be clustered using limited computational resources. Some data sets of large volumes are generated by WEKA. A personal computer (CPU: Intel Core i7-2600, RAM: 8 GB) is used for computations. The data are clustered by some clustering methods and the computational time spent for clustering is presented in Tables II and III. K-means, X-means, EM, DBSCAN, and OPTICS are used in WEKA. K-means and fuzzy c-means are used in KNIME. The time of k-means with RushAccelerator for KNIME is computed (Table III).

K-means, X-means, EM, and fuzzy c-means are able to cluster rather large data (about 1 million items) in the acceptable time (less than 3 hours). k-means takes the least time (less than 7 minutes). Usage of RushAccelerator for KNIME allows us to increase computations for several times. DBSCAN and OPTICS are not suitable for big or even large data clustering when a personal computer without additional technologies is used. If we deal with large data using WEKA it is necessary to apply additional tools, eg. employ a command-line interface to interact with

Table II
COMPUTATIONAL TIME OF CLUSTERING (IN SEC.) IN WEKA

Number of items m	k-means	X-means	EM	DBSCAN	OPTICS
260 538	52	59	931	5847	N/A
521 082	183	229	2458	N/A	N/A
781 623	456	368	4732	N/A	N/A
1042171	438	620	8653	N/A	N/A

N/A means that computational time exceeds
3 hours or memory problem arises

Table III
COMPUTATIONAL TIME OF CLUSTERING (IN SEC.) IN KNIME

Number of items m	Number of features n	k-means	k-means (RushAccelerator)	fuzzy c-means
260 538	2	62	4	2413
521 082	2	191	5	1186
781 623	2	282	14	3557
1 042 171	2	393	25	5038

WEKA, write code directly in Java or a Java-based scripting language such as Groovy or Jython, or using MOA data stream software. Using other data mining systems, the results would be similar. If we deal with big data, some innovative technologies should be used (Fig. 1).

V. GENERALIZATION AND CONCLUSIONS

Challenges and problems of big data clustering are analysed in the paper. The clustering methods and technologies are discussed. The well known data mining systems usually use a power and resources of only one personal computer. Nowadays, new devices, social media, and other sources generate the data of huge volumes. More innovative technologies which would be need for big data analysis. In this paper, the strategies for big data clustering have been presented.

The selection of the strategy depends on the volume of data analysed. When we deal with a large data set, the well known data mining systems usually are used. The complex problems of data analysis require usage of parallel and distributed computing-based systems and technologies. Big data initiate development of new technologies. Hadoop-based technologies and libraries are the most popular solutions for big data analysis and clustering.

ACKNOWLEDGMENT

This work has been supported by the project 'Theoretical and engineering aspects of e-service technology development and application in high-performance computing platforms' (No. VP1-3.1-ŠMM-08-K-01-010) funded by the European Social Fund.

REFERENCES

[1] S. Schmidt, *Data is exploding: the 3 V's of big data*. Business Computing World, 2012.

[2] Y. Zhai, Y.-S. Ong, and IW. Tsang, The Emerging "Big Dimensionality". In *Proceedings of the 22nd International Conference on World Wide Web Companion*, Computational Intelligence Magazine, IEEE, vol. 9, no. 3, pp. 14–26, 2014.

[3] V. Medvedev, G. Dzemyda, O. Kurasova, and V. Marcinkevičius, "Efficient data projection for visual analysis of large data sets using neural networks", *Informatica*, vol. 22, no. 4, pp. 507–520, 2011.

[4] G. Dzemyda, O. Kurasova, and V. Medvedev, "Dimension reduction and data visualization using neural networks", in *Maglogiannis, I., Karpouzis, K., Wallace, M., Soldatos, J., eds.: Emerging Artificial Intelligence Applications in Computer Engineering. Volume 160 of Frontiers in Artificial Intelligence and Applications*, IOS Press, 2007, pp. 25–49.

[5] A. McCallum, K. Nigam,and L. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching", in *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 169–178, 2000.

[6] M.H. Dunham, *Data Mining: Introductory and Advanced Topics*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2002.

[7] J. MacQueen, "Some methods for classification and analysis of multivariate observations", in *Le Cam, L.M., Neyman, J., eds.: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley and Los Angeles, CA, USA, University of California Press, vol. 1, pp. 281–297, 1967.

[8] T. Kohonen, Overture. *Self-Organizing neural networks: recent advances and applications*, Springer-Verlag, New York, NY, USA, 2002, pp. 1–12.

[9] I. Dhillon, Y. Guan, B. Kulis, "Kernel k-means: spectral clustering and normalized cuts", in *Proceeding KDD '04 Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 551–556, 2004.

[10] I. Dhillon and D. Modha, "Concept decompositions for large sparse text data using clustering", *Machine Learning*, vol. 42, no. 1–2, pp. 143–175, 2001.

[11] R.C. de Amorim and B. Mirkin, "Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering", *Pattern Recognition*, vol. 45, no. 3, pp. 1061–1075, 2012.

[12] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, 1981.

[13] D. Pelleg and A.W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters", in *Proceedings of the 17th International Conference on Machine Learning*, Morgan Kaufmann, pp. 727–734, 2000.

[14] X. Cai, F. Nie, and H. Huang, "Multi-view k-means clustering on big data", in *Rossi, F., ed.: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013, IJCAI/AAAI (2013)*.

[15] N. Ailon, R. Jaiswal, and C. Monteleoni, "Streaming k-means approximation", in *Proceedings of 23rd Annual Conference on Neural Information Processing Systems, NIPS 2009*, pp. 10–18, 2009.

[16] V. Braverman, A. Meyerson, R. Ostrovsky, A. Roytman, M. Shindler, and B. Tagiku, "Streaming k-means on well-clusterable data", in *Randall, D., ed.: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, SIAM*, pp. 26–40, 2011.

[17] M. Shindler, A. Wong, and A. Meyerson, "Fast and accurate k-means for large datasets", in *Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F.C.N., Weinberger, K.Q., eds.: Proceedings of 25th Annual Conference on Neural Information Processing Systems NIPS*, pp. 2375–2383, 2011.

[18] G. Dzemyda, O. Kurasova, and J. Žilinskas, *Multidimensional Data Visualization: Methods and Applications*, Springer Optimization and Its Applications, Springer, 2013.

[19] B. Hammer, A. Micheli, A. Sperduti, and M. Strickert, "A general framework for unsupervised processing of structured data", *Neurocomputing*, vol. 57, pp. 3–35, 2004.

[20] T. Voegtlin, "Recursive self-organizing maps", *Neural Networks*, vol. 15, no. 8–9, pp. 979–991, 2002.

[21] K. Lagus, S. Kaski, and T. Kohonen, "Mining massive document collections by the WEBSOM method", *Information Sciences*, vol. 163, no. 1–3, pp. 135–156, 2004.

[22] Y. Iwasaki, T. Abe, Y. Wada, K. Wada, and T. Ikemura, "Novel bioinformatics strategies for prediction of directional sequence changes in influenza virus genomes and for surveillance of potentially hazardous strains", *BMC Infectious Diseases*, vol. 13, no. 386, 2013.

[23] S. Alonso, M. Sulkava, M.A. Prada, M. Domínguez-González, and J. Hollmén, "EnvSOM: a SOM algorithm conditioned on the environment for clustering and visualization", in *WSOM 2011*, vol. 6731 of Lecture Notes in Computer Science, Springer, pp. 61–70, 2011.

[24] M. Mishra and H. Behera, "Kohonen self organizing map with modified k-means clustering for high dimensional data set", *International Journal of Applied Information Systems*, vol. 2, no. 3, pp. 34–39, 2012.

[25] A. Prakash, "Reconstructing self organizing maps as spider graphs for better visual interpretation of large unstructured datasets", *Infosys Lab Briefings*, vol. 11, no. 1, 2012.

[26] M. Dittenbach, D. Merkl, and A. Rauber, "The growing hierarchical self-organizing map", *IEEE - INNS - ENNS International Joint Conference on Neural Networks*, vol. 6, 2000.

[27] P. Stefanovič and O. Kurasova, "Visual analysis of self-organizing maps", *Nonlinear analysis: Modelling and Control*, vol. 16, no. 4, pp. 488–504, 2011.

[28] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases", in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data. SIGMOD '96, New York, NY, USA, ACM*, pp. 103–114, 1996.

[29] M. Ester, H.P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", in *Proceedings of 2nd International Conference on Knowledge Discovery and and Data Mining*, AAAI Press, pp. 226–231, 1996.

[30] M. Ankerst, M.M. Breunig, H.P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure", in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD'99, ACM, pp. 49–60, 1999.

[31] S. Guha, R. Rastogi, and K. Shim, "CURE: an efficient clustering algorithm for large databases", *Information Systems*, vol. 26, no. 1, pp. 35–58, 2001.

[32] G. McLachlan and T. Krishnan, *The EM algorithm and extensions. 2nd edn*, Wiley series in probability and statistics, 2008.

[33] A. Adigun Abimbola, O. Omidiora Elijah, and O. Olabiyisi Stephen, "An exploratory study of k-means and expectation maximization algorithms", *British Journal of Mathematics and Computer Science*, vol .2, no. 2, pp. 62–71, 2012.

[34] H. Yu, J. Yang, and J. Han, "Classifying large data sets using svms with hierarchical clusters", in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, ACM*, pp. 306–315, 2003.

[35] L. David and B. Daniel, "Clustering very large datasets using a low memory matrix factored representation", *Computational Intelligence*, vol. 25, pp. 114–135, 2009.

[36] A. Fabrizio, P. Clara, and R. Massimo, "DESCRY: a density based clustering algorithm for very large data sets", *Intelligent Data Engineering and Automated Learning, IDEAL 2004, Lecture Notes in Computer Science*, vol. 3177, pp. 203–210, 2004.

[37] P. Russom, *Big data analytics. TDWI best practices report, 4th quarter*, Technical report, 2011.

[38] A. Cuzzocrea, I.Y. Song, and K.C. Davis, "Analytics over large-scale multidimensional data: the big data revolution!", in *Song, I.Y., Cuzzocrea, A., Davis, K.C., eds.: Proceedings of the 14th International Workshop on Data Warehousing and OLAP, DOLAP 2011, ACM*, pp. 101–104, 2011.

[39] S. Melnik, A. Gubarev, J.J. Long, G. Romer, S. Shivakumar, M. Tolton, and T. Vassilakis, "Dremel: Interactive analysis of web-scale datasets", in *Proceedings of the 36th International Conference on Very Large Data Bases*, pp. 330–339, 2010.

[40] M. Gianmarco De Francisci, "SAMOA: a platform for mining big data streams", in *Proceedings of the 22nd International Conference on World Wide Web Companion, WWW '13 Companion*, pp. 777–778, 2013.

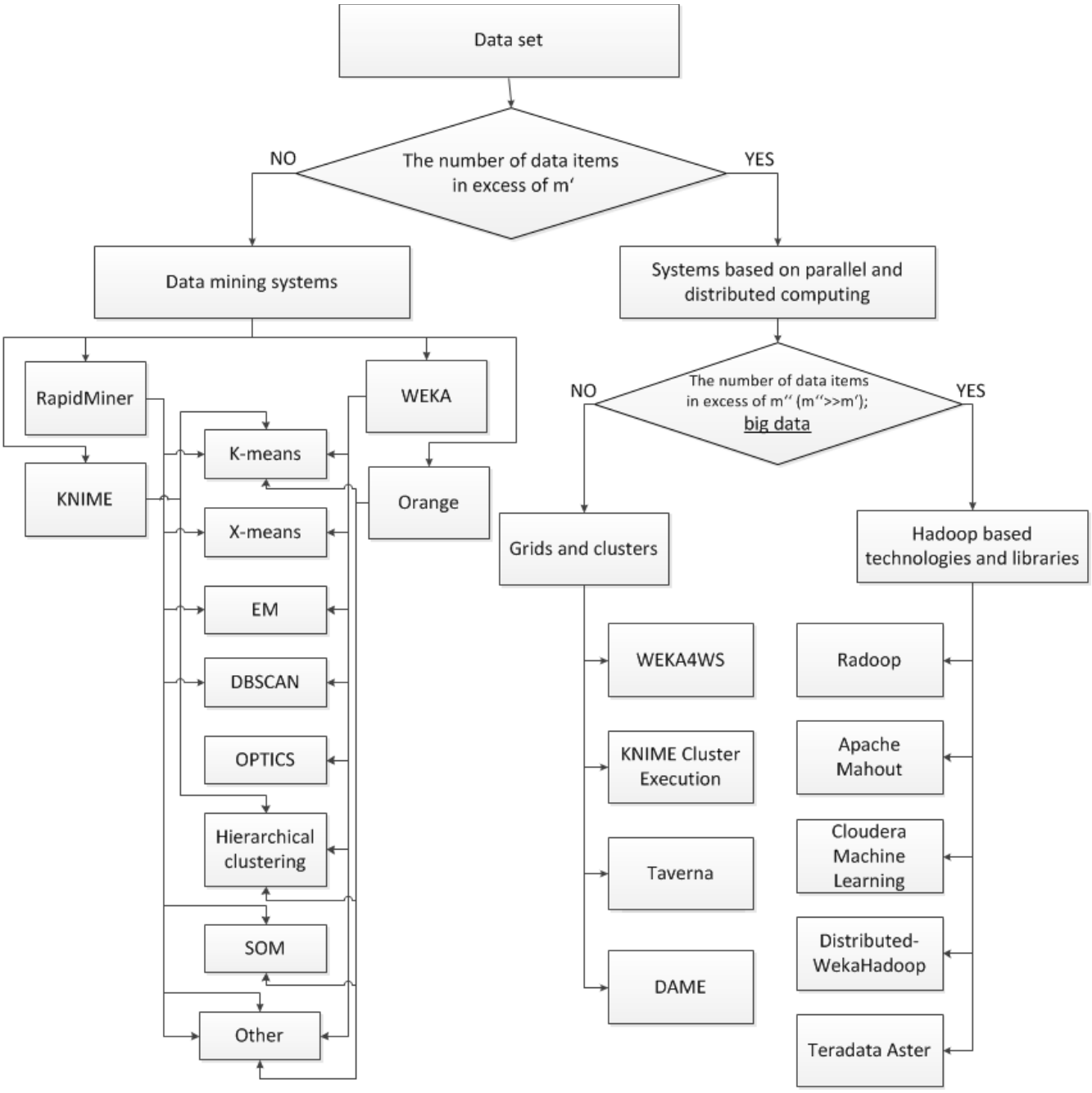

Figure 1. Schema of strategies for big data clustering