sciendo

JAISCR, 2019, Vol. 9, No. 4, pp. 293 – 302
10.2478/jaiscr-2019-0010

DECISION-MAKING ENHANCEMENT IN A BIG DATA ENVIRONMENT: APPLICATION OF THE K-MEANS ALGORITHM TO MIXED DATA

Oded Koren[1], Carina Antonia Hallin[2], Nir Perel[1,*], Dror Bendet[1]

[1]*School of Industrial Engineering and Management, Shenkar - Engineering. Design. Art, 12 Anne Frank st., Ramat Gan, Israel*

[2]*Department of International Economics, Government and Business, Copenhagen Business School, Solbjerg Pl. 3, 2000 Frederiksberg, Copenhagen, Demark*

[*]*E-mail: perelnir@gmail.com*

Submitted: 8th May 2019; Accepted: 25th July 2019

Abstract

Big data research has become an important discipline in information systems research. However, the flood of data being generated on the Internet is increasingly unstructured and non-numeric in the form of images and texts. Thus, research indicates that there is an increasing need to develop more efficient algorithms for treating mixed data in big data for effective decision making. In this paper, we apply the classical K-means algorithm to both numeric and categorical attributes in big data platforms. We first present an algorithm that handles the problem of mixed data. We then use big data platforms to implement the algorithm, demonstrating its functionalities by applying the algorithm in a detailed case study. This provides us with a solid basis for performing more targeted profiling for decision making and research using big data. Consequently, the decision makers will be able to treat mixed data, numerical and categorical data, to explain and predict phenomena in the big data ecosystem. Our research includes a detailed end-to-end case study that presents an implementation of the suggested procedure. This demonstrates its capabilities and the advantages that allow it to improve the decision-making process by targeting organizations' business requirements to a specific cluster[s]/profiles[s] based on the enhancement outcomes.

Keywords: Big data, mixed data, Hadoop, K-means, decision making

1 Introduction

Every organization at some point experiences a data-driven revolution in management. Firms adopt big data tools to capture enormous amounts of fine-grained data derived from social media activity, web browsing patterns, mobile phone usage, video, audio, images, text message usage, and new formations of data generation such as mobile use, messages over the Internet, and Internet of Things (IoT) usages [19]. Analysis of big data promises to produce insights and predictions that will revolutionize managerial decision making [25]. Big data offers the ability to render into data many aspects of the world that have never been quantified before, a process also referred to as "datafication" [7].

Historically, the information science discipline has focused on how to design and implement systems to provide the relevant data in the appropriate time [2]. However, the wealth of big data poses

challenges for effective decision making in terms of rigor and a number of variables. When working with large data sets from unknown sources, researchers must carefully evaluate the potential biases before drawing conclusions. The sheer size and variety of variables in the big data ecosystem requires too many observations that are complex and difficult to deal with (for a review, see [2]). The way to overcome such challenges is to develop better and simpler algorithms, systems, and processes that can break down and make sense of all the heterogeneous and fragmented information on the web. A big data ecosystem includes a platform that is enabled to handle a huge amount of data (on several levels) via a variety of tools. Use of big data technologies such as Apache Hadoop!, MapReduce, Apache Pig!, Apache Hive and Apache HBase (see [31]) is associated with the emergence of new technical skills. The early adaptation of big data tools attracted media attention, such as when Sears started to experiment with Apache Hadoop!, which was central to the first wave of big data investments. Of course, Sears had to learn Apache Hadoop! the hard way, through trial and error, because it had only a few outside experts available to guide its work when it introduced the software in 2010 [16].

Processes for storing large amounts of data in the Hadoop Distributed File System (HDFS) can be executed via MapReduce [8]. Furthermore, there are other functionalities and tools for analysing information for various business purposes (such as machine learning algorithms). The ability to combine big data tools with different data analysis functionalities, such as Apache Hive and Apache Pig!, is growing, see e.g. [11, 22] and [26], as is the variety of other big data tools designed for handling data, such as ETL [32]. Big data is also being studied in relation to machine learning tools such as Apache Mahout [24]. The massive volumes of data in big data are treated using different capabilities and tools, see e.g. [9] and [32]. Apache Hadoop! is a platform that includes the ability to store, manage, read, write, and operate on massive amounts of data/files via HDFS, a system based on the Google File System (GFS) [13] that can analyse information for different purposes. Although these approaches have advanced the possibilities for dealing with massive data, they do not offer algorithms that can structure data effectively for analytical and decision-making purposes. For example, IBM's Watson may be on the cutting edge in natural language processing, but it has a long way to go in terms of the system's ability to absorb and interpret big data across the internet [2]. These observations reflect a need to develop new approaches for structuring and categorizing massive amounts of data in an emergent big data ecosystem.

K-means, a popular data clustering method, is a simple and elegant approach to partitioning a data set into K distinct clusters. This algorithm has been proposed by several scientists in different forms and under different assumptions. A review on the origins of the K-means algorithm can be found in [15] and [20]. First, a value of K is specified, and then the algorithm assigns each observation from the data set to exactly one of the K clusters. The Assignment is decided by minimizing the differences between observations that belong to the same cluster. These differences are commonly measured by squared Euclidean distance, but there are many other possible ways to define this concept. A recent example involving K-means utilizations can be found in [10], where the authors studied how different types of the community may affect the effectiveness of open-source software. In addition, in [12], the authors used the K-means method to investigate and identify different types of user role in innovation-contest communities. In [29] the authors applied the K-means algorithm to study time-varying effects on the allocation of marketing resources, and in [14] it was used to analyse doctors' profiles. Further studies on K-means can be found in [21] and in [17].

One challenge of using the K-means algorithm is that it works well with numeric data but is not directly applicable to non-numeric, categorical data (see [4]) because the Euclidean distance function is not meaningful when considering categorical values. This paper presents a novel approach that simplifies the challenges of mixed data for decision making in big data. We address the question of how the K-means algorithm can solve the problem of clustering mixed data in big data.

The performance of the K-means algorithm on categorical data has been studied in the information science literature, which describes how it converts multiple category attributes into binary attributes that it then treats as numeric, see [28]. However, this method may greatly increase the compu-

tational effort, especially when working with big data. Consequently, scholars have applied the K-modes algorithm and the K-prototypes algorithm [18]. The K-modes algorithm extends the K-means method of clustering categorical data by defining differences between clusters in terms of frequencies and by considering modes instead of means. The K-prototypes algorithm is a mixture of the K-means and the K-modes algorithms; that is, a defined cluster center (or representative) allows treating a clustering problem with categorical variables to be a traditional K-means problem [30]. The general method of choosing a cluster representative and measuring dissimilarities between clusters is performed by relative frequency-based methods [3] or by applying the K-means algorithm to mixed data, see [3] and [33]. However, the later studies were not performed in a big data environment. For example, the numerical studies presented in [3] considered data sets with at most 690 elements.

Our contribution is to adapt the K-means algorithm to mixed big data. That is, we use big data platforms (in terms of parallel computation techniques and storage capabilities) to explore how the K-means algorithm works on big data with both numeric and non-numeric variables. Since data size expands tremendously, analysing data on a single machine is inefficient. The most appropriate solution is to consider parallelism within a distributed computational framework. One of the most common programming frameworks for processing large-scale data sets using parallelism is MapReduce [8], which exploits the qualities of parallel computing, see [5] and [6].

In this paper, we address two fundamentals: (i) we provide a clustering algorithm that handles both numeric and categorical attributes in big data environments, based on the capabilities of big data tools and the K-means algorithm; and (ii) we explore how the results of the algorithm in a big data environment, based on the ability to support complex architectures, can extend the clustering, profiling, analysis, and predictions capabilities.

Our algorithm enables the application of the K-means algorithm to both numerical and non-numerical data. The empirical evidence is broadly supportive of the two issues we seek to address. We first create a procedure that "flattens" all the data from categorical and numerical data to pure numerical data. We then filter all the categorical classes into distinct groups, based on categorical combinations, which allows us to analyse each group separately (because we are dealing with big data, the grouping process and the K-means process are performed via big data platforms). That is, we perform the K-means algorithm only on the remaining numeric variables. Last, we collect all the groups' analysis outcomes. These outcomes can serve as the basis for further analysis and support the organization requirements and business needs.

This paper is an extended version of our conference paper [23]. We extend the conference paper in the following directions: We created a new data set and a new experiment based on the presented algorithm. The new dataset is much more complex (from the categorical variables point of view), in comparison with the one studied in the publication of the proceedings. Specifically, it demonstrates a process with 1600 different files, or, in our context, 1600 different characteristics, (with a total size of $\sim$1.05GB), while 36 files were considered in the publication of the proceedings; We present a full business use-case analysis, in which we present the aggregation outcomes (clusters) of 5,000 clusters, and show how the results can be used in a targeted profiling task; We present how the suggested procedure may improve a decision-making process.

Our study presents a method for treating mixed data in big data that was not previously possible. The approach advances the capabilities of dealing with massive data, such as in decision making, because profiling, forecasting, and other analyses can be performed in a more targeted manner.

Recent studies have discussed the relationship between big data and theory. For example, it is suggested that big data and theory can work synergistically to explore phenomena or solve problems by using big data platforms and tools to generate theoretical insights rather than starting with a preconceived theory [27]. Furthermore, in [1, p. xxii], the authors have indicated that "big data has potentially important implications for theory". On the one hand, theory can be replaced by patterns derived from data. On the other hand, data without theory lacks order, sense, and meaning. We have adopted the concept presented in these studies; that is, we present a method for analysing data in big data environments that can be applied to any rele-

vant theoretical issue.

The rest of the paper is organized as follows. In Section 2, we present our new alternative procedure for performing the K-means algorithm with mixed data in a big data environment. In Section 3 we present an implementation example of the proposed procedure. In Section 4, we present a detailed case study of the procedure, demonstrating how an organization can use the proposed new process to expand and improve its decision-making process. Section 5 concludes the paper.

2 Model Development

We argue that applying the K-means algorithm to mixed data in the big data ecosystem can enhance decision making and allow decision-makers to treat massive amounts of data. The current study thus analyses the impact of the K-means algorithm when applied to both numerical and categorical (non-numerical) data in big data platforms. The model assumes a data set that includes m categorical variables and n quantitative variables, and that categorical variable j may have $a_j \geq 2$ different states.

The K-Means Algorithm Procedure

Claim 1: Non-numeric data in big data can be assigned values.
Proof: We first perform the K-means algorithm on our data set by adopting the following steps:

1. Create $\prod_{j=1}^{m} a_j$ different types of group that differ by their categorical variable's values. Each record is assigned to its group, according to its categorical values.

2. Each group generated in step 1 is a file (or other storage format) in the big data platform (this will enable parallel computing in the next steps).

3. Perform a parallel K-means algorithm on all groups according to their numeric variables.

4. Aggregate all the clusters (K clusters from each group) from step 3 to one outcome for further analysis, as described in Section 4.

3 Implementation Example of the Algorithm

The following Section presents an end-to-end implementation example.

1. Upload the data set and categorical files to the HDFS (in Apache Hadoop!). Pre-set: each of the $\prod_{j=1}^{m} a_j$ possible combinations of the values of each categorical variable is in a separated file. Each file contains the records with the corresponding categorical values. This is a mandatory step because there is a need to create all combinations of the available states based on the definition/business requirements. Note that there might be empty files (groups) if there are no records with the corresponding categorical values.

2. Multiply all the files (from step 1) to create multiple lines. Each line describes a unique combination. All lines are stored in a file in HDFS (in Apache Hadoop!) for parallel analysis (in a big data platform).

3. Filter the data set for each unique file (from step 2) and send the relevant quantitative variables to the relevant file.

4. Run (via bash script) the K-means algorithm (Apache Mahout) on each file that is located in a separated directory (from step 3) with the following parameters:

   – a configurable parameter, x, for the number of iterations (in this case we used 5 iterations for all K-means runs);
   – a number of clusters, K, which is influenced by the number of records per each unique file (from step 3). The number of clusters K increases when the number of records per file grows.

5. Gather all the clusters to one defined structure for additional analysis (compare between clusters, order, analysis, etc.).

Note that steps 1 to 3 were implemented and tested in a single-node environment. By using Apache Pig!, the following actions are performed (see the next Section for a detailed description):

1. loading the full data set.
2. creating all the categorical variables combinations.
3. filtering the relevant categorical variables and creating the groups/files (per combination) with the relevant filter quantitative variables.

4 End-to-End Case Study of the Procedure Implementation

In this Section, we present a detailed end-to-end case study of the implementation procedure, its use and functionalities. We also suggest how it may expand and improve the decision-making process.

4.1 Data set

The generated data set consists of 14 variables, 8 quantitative and 6 categorical. It contains 11M records, with a total of ∼1.05GB. The list of all variables (categorical and numerical) is as follows:

Quantitative variables: Age, Work years, Salary, Education years, Number of houses, Number of children, Travels per year, Number of vehicles.

The categorical variables and their possible values are given in Table 1. Note that the combinations of the six categorical variables can create at most 1600 distinct characteristics.

Based on the predefinition of the number of records per cluster, 5,000 clusters were created.

4.2 Procedure flow

Figure 1 describes the end-to-end technical implementation of the end-to-end use case. The main elements are:

– HDFS-this is the Apache Hadoop Distributed File System. The full data set (see step 1 in Figure 1) is uploaded to the HDFS. Then, all the different combinations are created, and the relevant fields are filtered from the full data. All the filters are stored in the HDFS (see step 2 in Figure 1) for the analysis process.

– MapReduce-this is the main component, which handles and manages all the parallel processing done when analysing the data (in the implementation example, we used Apache Mahout for the K-means algorithm).

– BI-this component represents the business intelligence requirements and demands. Obviously, the requirements change from organization to organizations, but the need to target specific profiles is likely to be the same across different businesses (which, in the querying purposes, may be distinguished by, for example, the querying values themselves, the data, the expected outcomes, etc.).

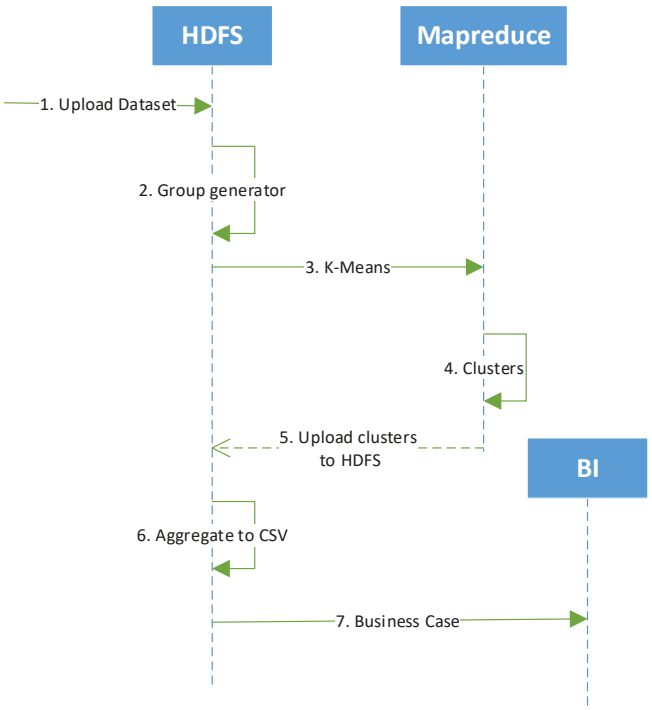

Figure 1. Procedure flow

Table 2 describes in detail the steps shown in Figure 1.

4.3 Run criteria

After running the procedure, the clusters are aggregated to a single file that contains all results and time-stamps. Note that there are no numerical values for records that are not the owners of a house and are experts in finance. Overall, through the filtering process, we identified 40 group types with empty numerical values. We performed the K-means algorithm in groups of 100 directories, where the predefined number of clusters,

Table 1. Categorical variables

Variable	Values
Gender	woman, man (2 options)
Living zone	most-expensive, high-expensive, medium, below-medium (4 options)
House own	no, yes (2 options)
Health	healthy, good, limited, extremely limited, dying (5 options)
Marital status	single, married, divorced, widowed (4 options)
Financial knowledge	clueless, below-average, average, above-average, expert (5 options)

Table 2. Procedure steps

in**Step**	**Name**	**Description**
in1	Upload data set	Upload the data set to the HDFS. This is the complete full data set that includes the values of all 14 variables. The data set includes the raw values of both the quantitative and the categorical variables.
in2	Group generator	Generate the groups by the total combinations of the categorical variables. In this case, we created 1,600 different groups, based on the combinations of six different categorical values. In each group, we filtered the relevant quantitative values (due to this, we ended up with 40 groups out of 1,600 that did not contain any quantitative values). Each unique group (generated based on its special combination of the categorical values) contains only the relevant filtered quantitative values and is stored as a separated subset data set (e.g. the implementation can be formatted as a file, a table, etc.) in the HDFS (spread as blocks in the data nodes).
in3	K-means	Perform the K-means algorithm (via Apache Mahout) on each separated group/file/subset of the data set, generated in the previous step (step 2).
in4-5	Upload clusters to HDFS	Gather and aggregate all the clusters (step 4) and upload the clusters created from the previous step to the HDFS (step 5).
in6	Aggregate to CSV	Aggregate all clusters to CSV. The resulting aggregation (based on a well-defined format) provides the platform for additional analysis and querying for various business purposes.
in7	Business case	The business querying process emphasizes and demonstrates a business targeting example based on the market and/or other business requirements and demands.

K, was selected based on the number of reorders per group/file/subset data set, according to Table 3. The number of iterations in each K-means procedure was five.

Table 3. Predefined number of clusters according to number of records

K	Number of records in the file/group
1	Between 0 and 1,000
2	Between 1,000 and 2,000
3	Between 2,000 and 5,000
5	Between 5,000 and 10,000
10	Above 10,000

Based on Table 3, the K-means algorithm created a total of 5,940 clusters (1 to 10 clusters per group for each of the 1,560 groups with values). Note that identifying the 40 profiles that did not contain any quantitative information/values is extremely important, because there may be different business demands and needs that require the identification of the profiles without any values or without any records/observations.

4.4 BI use case

The main question is: how can an organization use the massive number of clusters aggregated into one massive data set for decision making? The aggregated cluster data sets, which contain unique profiles and groups, form a valuable information pattern that can enable targeting to a specific population.

This Section demonstrates an example of a use case for a specific decision-making need. We assume that we have gathered and created the aggregated clustered data set, and that the company would like to examine whether it is profitable to invest in a specific profile segmentation, according to the company's targets and needs.

Table 4. Target query values

Variable	Values
Education years	> 14
Number of children	> 2
Work years	> 10
Travels per year	> 5

As an illustration, we start with 5,940 clusters, generated from 1,560 groups. The 5,940 clusters contain a total of 11M specific resources (where each resource is a specific user observation). Table 4 presents the values that were specified (in this example) for the requirements that define the target population.

As detailed in Table 5, only five profiles (clusters) out of 5,940 fulfilled the requirements of the target profiles. As a result of this profiling process, the company can focus only on users within these profiles.

Table 6 presents the profiles' quantitative K-means information (per profile), as well as the number of observations per profile (denoted by *N*).

That is, five profiles, which contain 2,085 observations (out of 11M), satisfy the business needs. This might be very valuable information for a business, because it can help and support the decision-making process. For example, the business may wish to concentrate on the largest group-in our case the group with 749 cases (profile 2). Furthermore, the business might be interested in which profile has the highest average salary (profile 5), etc.

5 Discussion and Conclusions

In this paper, we presented a new approach that overcomes the difficulty of working with mixed data for decision making in a big data environment. The power of clustering and narrowing down the profiles to targeted groups, based on the business needs, improves the decision-making process. In our testing and implementation of the K-means algorithm, we found that the algorithm worked well in the runs. However, the complexity of analysis of our suggested procedure must be tested in future studies. We argue that the complexity of our process is more efficient when compared to the complexity of a regular K-means algorithm that runs on a full data set, due to its ability to reduce the size of the data set. The algorithm runs on subsets that possess fewer records per group. This influences the number of K-means iterations per group. Furthermore, note that, in a big data environment, all the K-means calculations can be done in parallel in different data nodes. Therefore, we believe that the complexity will be influenced mostly by the size of

Table 5. Business query target profiles

Variable	Profile 1	Profile 2	Profile 3	Profile 4	Profile 5
Gender	Man	Man	Man	Man	Man
Living zone	Below-medium	Below-medium	High-expensive	Most-expensive	Most-expensive
House own	No	Yes	No	Yes	Yes
Health	Good	Healthy	Healthy	Healthy	Healthy
Marital status	Married	Divorced	Married	Divorced	Married
Financial knowledge	Below-average	Average	Average	Average	Clueless

Table 6. Designated 5 cluster values

Categories	Profile 1	Profile 2	Profile 3	Profile 4	Profile 5
N	143	749	296	306	591
c:Age	45.256	45.544	46.418	46.282	45.927
c:Work years	10.355	10.779	11.226	11.071	10.748
c:Salary	30,396.14	35,441.57	30,590.66	37,623.20	39,196.51
c:Education years	15.07	14.955	15.24	15.363	15.435
c:number of homes	0	1.777	0	1.856	2.076
c:number of children	3.441	2.877	3.294	2.899	3.098
c:Travels per year	5.545	5.298	5.142	5.297	5.078
c:Vehicles	1.35	1.344	1.659	1.304	1.547
r:Age	5.945	6.136	6.125	6.234	6.383
r:Work years	6.138	6.261	6.352	6.695	6.624
r:Salary	3,212.126	2,496.518	2,846.027	2,379.237	2,488.128
r:Education years	2.42	2.936	2.581	2.9	3.228
r:Number of homes	0	0.72	0	0.709	0.884
r:Number of children	0.686	1.072	0.752	0.996	0.986
r:Travels per year	0.611	1.088	0.915	1.042	1.218
r:Vehicles	0.75	0.932	1.091	0.958	0.967

the largest group that will be generated.

The method can be applied to any relevant theoretical question in a big data environment for decision making. An example is exploring what can explain fluctuations in the economy over time based on a set of selected variables. Such variables may be generated from target surveying of public crowds on what are important behavioral criteria that can explain changes in the economy. Data sets from big data environments should analyse any decision-making issues by defining x number of possible explanatory variables that should predict a dependent variable. This allows for deriving possible patterns by testing these variables. Thus, our approach allows clustering of data in a more straightforward way to develop new theories.

Note that this paper does not include a complexity analysis comparing the presented K-means method for mixed data in a big data environment with a straightforward K-means algorithm for mixed data. Nevertheless, we claim that the complexity of the presented method is better because of:

1. A reduction of the data set size: each group's analysis is conducted on fewer observations (due to the filtering of the relevant data; see Sections 2 and 3);

2. Parallelization of the analysis process: the big data architecture enables us to perform the analysis process in parallel (per group/file that is allocated on the HDFS platform based on the MapReduce job). Based on the assumption that all the data nodes possess the same capacity and performance capabilities (during the procedure run time), we can also assume that the largest group (the subset data set) will have the highest complexity and will therefore influence the overall complexity in the greatest manner. However, this theoretical assumption still needs to be validated.

We demonstrated the strength of the enhancement outcomes compared to basic K-means outcomes that should benefit predictions and consequential decision making. This procedure enables an organization to perform a more accurate analysis

of data and may create better business understanding and insights from the business data for a variety of services and needs. This implementation may also improve business decision-making processes due to business data comprehension. The ability to combine different types of data and to simplify the outcomes and focus on selected profiles or groups with reduced observations improves the ability of a business to enhance understanding of the outcomes and to provide improved analysis, prediction accuracy, growth, and new horizons.

References

[1] Ahmed Abbasi, Suprateek Sarker, and Roger HL Chiang. Big data research in information systems: Toward an inclusive research agenda. Journal of the Association for Information Systems, 17(2):I, 2016.

[2] Ritu Agarwal and Vasant Dhar. Big data, data science, and analytics: The opportunity and challenge for is research. Information Systems Research, 25(3):443–448, 2014.

[3] Amir Ahmad and Lipika Dey. A k-mean clustering algorithm for mixed numeric and categorical data. Data & Knowledge Engineering, 63(2):503–527, 2007.

[4] Pavel Berkhin. A survey of clustering data mining techniques. In Grouping multidimensional data, pages 25–71. Springer, 2006.

[5] Xiao Cai, Feiping Nie, and Heng Huang. Multi-view k-means clustering on big data. In Twenty-Third International Joint conference on artificial intelligence, 2013.

[6] Xiaoli Cui, Pingfei Zhu, Xin Yang, Keqiu Li, and Changqing Ji. Optimized big data k-means clustering using mapreduce. The Journal of Supercomputing, 70(3):1249–1259, 2014.

[7] Kenneth Cukier and Viktor Mayer-Schoenberger. The rise of big data: How it's changing the way we think about the world. Foreign Aff., 92:28, 2013.

[8] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. Communications of the ACM, 51(1):107–113, 2008.

[9] Yuri Demchenko, Canh Ngo, and Peter Membrey. Architecture framework and components for the big data ecosystem. Journal of System and Network Engineering, pages 1–31, 2013.

[10] Dany Di Tullio and D Sandy Staples. The governance and control of open source software projects. Journal of Management Information Systems, 30(3):49–80, 2013.

[11] Gal Engelberg, Oded Koren, and Nir Perel. Big data performance evaluation analysis using apache pig. International Journal of Software Engineering and Its Applications, 10(11):429–440, 2016.

[12] Johann Füller, Katja Hutter, Julia Hautz, and Kurt Matzler. User roles and contributions in innovation-contest communities. Journal of Management Information Systems, 31(1):273–308, 2014.

[13] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. In Proceedings of the 19th ACM Symposium on Operating Systems Principles, pages 20–43, Bolton Landing, NY, 2003.

[14] Shanshan Guo, Xitong Guo, Yulin Fang, and Doug Vogel. How doctors gain social and economic returns in online health-care communities: a professional capital perspective. Journal of Management Information Systems, 34(2):487–519, 2017.

[15] Bock Hans-Hermann. Origins and extensions of the k-means algorithm in cluster analysis. Journal Electronique dHistoire des Probabilités et de la Statistique Electronic Journal for History of Probability and Statistics, 4:48–49, 2008.

[16] Doug Henschen. Why sears is going all-in on hadoop. Information week. Retrieved July, 1:2014, 2012.

[17] Joshua Zhexue Huang, Michael K Ng, Hongqiang Rong, and Zichen Li. Automated variable weighting in k-means type clustering. IEEE Transactions on Pattern Analysis & Machine Intelligence, (5):657–668, 2005.

[18] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. Data mining and knowledge discovery, 2(3):283–304, 1998.

[19] Cisco Visual Networking Index. The zettabyte era–trends and analysis. Cisco white paper, 2013.

[20] Anil K Jain. Data clustering: 50 years beyond k-means. Pattern recognition letters, 31(8):651–666, 2010.

[21] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis & Machine Intelligence, (7):881–892, 2002.

[22] Daniel Kendal, Oded Koren, and Nir Perel. Pig vs. hive use case analysis. International Journal of Database Theory and Application, 9(12):267–276, 2016.

[23] Oded Koren, Carina Antonia Hallin, Nir Perel, and Dror Bendet. Enhancement of the k-means algorithm for mixed data in big data platforms. In Proceedings of SAI Intelligent Systems Conference, pages 1025–1040. Springer, 2018.

[24] Sara Landset, Taghi M Khoshgoftaar, Aaron N Richter, and Tawfiq Hasanin. A survey of open source tools for machine learning with big data in the hadoop ecosystem. Journal of Big Data, 2(1):24, 2015.

[25] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H Byers. Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute, 2011.

[26] R Angelin Preethi and J Elavarasi. Big data analytics using hadoop tools, pache hive vs apache pig. International Journal of Emerging Technology in Computer Science & Electronics, 24(3), 2017.

[27] Arun Rai. Editor's comments: Synergies between big data and theory. MIS quarterly, 40(2):iii–ix, 2016.

[28] Henri Ralambondrainy. A conceptual version of the k-means algorithm. Pattern Recognition Letters, 16(11):1147–1157, 1995.

[29] Alok R Saboo, V Kumar, and Insu Park. Using big data to model time-varying effects for marketing resource (re) allocation. MIS Quarterly, 40(4), 2016.

[30] Ohn Mar San, Van-Nam Huynh, and Yoshiteru Nakamori. An alternative extension of the k-means algorithm for clustering categorical data. International Journal of Applied Mathematics and Computer Science, 14:241–247, 2004.

[31] Prasanna Tambe. Big data investment, skills, and firm value. Management Science, 60(6):1452–1469, 2014.

[32] Tom White. Hadoop: The definitive guide. O'Reilly Media, Inc., 2012.

[33] Rui Xu and Donald C Wunsch. Survey of clustering algorithms. IEEE Transactions on Neural Networks, 16(3):645–678, 2005.

Oded Koren has a Ph.D. from Tel-Aviv University and he is a full faculty member at the Department of Industrial Engineering and Management in Shenkar - Engineering, Design, Art. Dr. Koren's research interests Dr. Koren's research interests are in big data and AI domains.

Carina Antonia Hallin is the founder and head of the Collective Intelligence Unit at the Department of International Economics, Government and Business, Copenhagen Business School. She holds a Ph.D. degree in strategy and knowledge management. Her research interests include collective intelligence, AI, human-machine interaction and decision support systems. She has published in the fields of decision science, computer science, strategy and management, and her work is cited by Forbes Magazine.

Nir Perel received a Ph.D. degree in operations research from Tel-Aviv University, Israel. He is a senior faculty member and the deputy dean of the school of Industrial Engineering and Management in Shenkar - Engineering, Design, Art. His research interests include operations research modeling and queueing theory, as well as statistical learning and big data.

Dror Bendet is an M.Sc student and a research assistant in the School of Industrial Engineering & Management at Shenkar - Engineering. Design. Art. and was a Research Associate at Collective Intelligence Unit, Copenhagen Business School. His interests are big data, machine learning and artificial intelligence.