Data Science and Distributed Intelligence: Recent Developments and Future Insights

Alfredo Cuzzocrea, Mohamed Medhat Gaber

ICAR-CNR and University of Calabria
I-87036 Cosenza, Italy
School of Computing, University of Portsmouth,
Portsmouth PO1 3HE, Hampshire, UK

Abstract. *Big Data*, *Data Science* and *MapReduce* are three keywords that have flooded our research papers and technical articles during the last two years. Also, due to the inherent distributed nature of computational infrastructures supporting Data Science (like *Clouds* and *Grids*), it is natural to view *Distributed Intelligence* as the most natural underlying paradigm for novel Data Science challenges. Following this major trend, in this paper we provide a background of these new terms, followed by a discussion of recent developments in the data mining and data warehousing areas in the light of aforementioned keywords. Finally, we provide our insights of the next stages in research and developments in this area.

1 Introduction

The two terms *Big Data* [28] and *MapReduce* [12] have dominated the scene in the intelligent data analysis field during the last two years. They are in fact the cause and effect of the rapid growth in data observed in the digital world. The phenomenon of very large databases and very high rate streaming data has been coined recently as Big Data. The largest two databases for Amazon account for 42 terabytes of data in total, and YouTube receives at least 65,000 new videos per day. Such figures increase every day and we are literally drowning in high waves of data. Making sense out of this data has become more important than ever in the knowledge era. With the birth of *learning from data streams*, Mutukrishnan in his later published book [24] has defined data streams as "*data arriving in a high rate that challenges our computation and communication capabilities*". In fact, this definition is now more true than back then. In spite of the continuous advances in our computation and communication capabilities, the data growth has been much faster, and the problem has become even more challenging. As a natural reaction to this worsening, a number of advanced techniques for data streams have been proposed, ranging from *compression paradigms* (e.g., [11, 10]) to intelligent approaches that successfully exploit the nature of such data sources, like their *multidimensionality*, to gain in effectiveness and efficiency during the processing phases (e.g., [8]), and recent initiatives that are capable of dealing with complex characteristics of such data sources, like their *uncertainty* and *imprecision*, as dictated by modern stream applicative settings (e.g., *social networks*, *Sensor Web*, *Clouds* – [9]).

Addressing such challenges has kept Data Mining and Machine Learning practitioners and researchers busy with exploring the possible solutions. MapReduce has come as

a potentially effective solution when dealing with large datasets, by enabling the breakdown of the main process into smaller tasks. Each of these tasks could be performed either in a *parallel* or *distributed* processing mode of operation. This allows the speed-up of performing complex data processing tasks, in an attempt to catch up with high speed large volume of data generated by scientific applications, such as the promising contexts of *analytics over large-scale multidimensional data* and *large-scale sensor network data processing*. With Big Data and MapReduce at the front of the scene, a new term describing the process of dealing with very large dataset has been coined, *Data Science*.

In line with this, when these kind of dataset are processed on top of a service-oriented infrastructure like the novel *Cloud Computing* one [2], the terms "*Database as a Service*" (DaaS) [19] and "*Infrastructure as a Service*" (IaaS) arise, and it is become critical to understand how Data Science can be coupled with distributed, service-oriented infrastructures, with novel and promising computational metaphors. Hence, due to the inherent distributed nature of computational infrastructures like Clouds (but also *Grids* [15]), it is natural to view *Distributed Intelligence* as the most natural underlying paradigm for novel Data Science challenges.

Following this major trend, in this paper we highlight the development in the Data Science area by first providing the necessary background of the Big Data and MapReduce in Section 2. Recent developments in the Data Mining field with the emergence of data science are provided in Section 3, followed by the recent developments on OLAP and Data Warehousing in Section 4. In Section 5, we present the foreseen future development in this area. Finally, in Section 6, we provide a summary and conclusions of our research.

2 The Emergence of Data Science

In his famous article "*What is Data Science?*" [23], Loukides has enumerated differences between Data Science and traditional statistical analysis. Mainly, Data Science deals with the whole process of gathering data, pre-processing them and finally making sense out of them, producing what he termed as *data products*. This definition may be confused with any definition given to Data Mining and Data Warehousing processes. What really makes Data Science different, however, is the holistic approach when looking at producing a data product. This is especially true with the large volumes of noisy and unstructured data generated in our daily lives, from social media to search terms on Google. Traditional Data Mining and Warehousing strategies become no longer valid when dealing with such large and dynamic data sources.

Thus, the phenomenon of Big Data has dictated the emergence of a new field that encompasses a number of well-established areas, including at the front line, Data Mining and Warehousing. This is the Data Science field, a term that we will encounter very often for some years to come. Scaling up the data analysis techniques to cope with Big Data has spotted the light on old functional programming functions, `map` and `reduce`, giving raise to the MapReduce computational paradigm. In the following subsections, a discussion of the Big Data phenomenon and how the two functions `map` and `reduce` have helped scaling up Big Data problems within the MapReduce paradigm is provided.

2.1 The Big Data Phenomenon

Soulellis [27] has enumerated a number of examples of Big Data. These include: (*i*) approximately, one zettabyte (i.e., 1,000,000,000,000 bytes) of data have been produced in 2010; (*ii*) it is estimated that 8 zettabytes of data will be produced in 2015; (*iii*) more than 30,000 tweets are sent every minute actually. All these examples well-describe the Big Data phenomenon that characterizes actual information systems. More interestingly and in addition to these examples, 90% of our data was the result of only the last two years of data production.

As a consequence, we are facing a big challenge with such huge data, and adequate analysis of these data can help advancing our knowledge greatly. There is no doubt that there is a great business advantage when enterprises are able to use such data to guide their decision making. It is a well-known news story that *GAP* store chain management have reverted their decision to change the company's logo when sentiment extracted from social media revealed that the customers did not like the new logo [4]. Another example is the controversial news story that *TARGET* department store have been able to predict that a teenage girl is pregnant using her new pattern of purchasing [21].

Not only business enterprises can benefit from such large data repositories, scientific discoveries can be also drawn from big data collected using advanced instruments generating data at very high rates. *Galaxy Zoo* [22] is one example of large data repository that uses the emergence of citizen science. Citizen science uses crowd annotation and data collection for the use in scientific research. In *Galaxy Zoo*, a very large collection of images representing galaxies are provided for users to annotate.

2.2 The MapReduce Computational Paradigm

MapReduce is a programming model that uses a *divide and conquer* method to speed-up processing large datasets [12]. It has been used in 2003 for the implementation of inverted index within the *Google Search Engine* in order to efficiently handle the search process. Also, it has been successfully exploited to handle large scale Machine Learning and Text Analytics tasks within *Google Analytics*. *Hadoop* [3] is the widely-known open source implementation of MapReduce.

The model of *MapReduce* has two main functions (`map` and `reduce`). The `map` function processes a key/value pair to produce a number of intermediate key/value pairs, which are then processed using the `reduce` function to merge all the intermediate pairs, and obtain the final result. A simple example by Dean and Ghemawat [12] has a `map` function that takes a string, and outputs the value 1 for each occurrence of a word in that string. The `reduce` function in turn processes this output to sum up the total occurrences for each word in the given string. MapReduce has attracted a great deal of attention over the past five years. More than 100,000 jobs uses MapReduce run on Google clusters every day [12]. Similar examples could be found in other giant software firms. MapReduce is a natural way for distributed and parallel processing of large datasets. Thus, Big Data has found a feasible way to be consumed by processing applications. However, the pace of data generation is still a big issue. This is especially true when we deal with Data Mining and Warehousing applications.

MapReduce is strictly related to the DaaS and IaaS metaphors mentioned in Section 1. Here, we discuss DaaS and IaaS in greater detail. DaaS defines a set of tools that provide final users with seamless mechanisms for creating, storing, accessing and managing their proper databases on remote (data) servers. Due to the nave features of Big Data, DaaS is the most appropriate computational data framework to implement Big Data repositories [2]. MapReduce is a relevant realization of the DaaS initiative. IaaS is a provision model according to which organizations outsource infrastructures (i.e., hardware, software, network) used to support ICT operations. The IaaS provider is responsible for housing, running and maintaining these services, by ensuring important capabilities like elasticity, pay-per-use, transfer of risk and low time to market. Due to specific application requirements of applications running over Big Data repositories, IaaS is the most appropriate computational service framework to implement Big Data applications.

3 Recent Data Mining Developments

A number of Data Mining techniques have been developed utilizing the MapReduce framework to scale up to Big Data. Papadimitriou *et al* [26] have classified the applicability of speedingup Data Mining algorithms using MapReduce into the following three categories: *one-iteration techniques* that are perfect for MapReduce, *multiple-iterations techniques* that are applicable taking into consideration that only small amount of shared information needs to be synchronized among iterations, and *not- applicable techniques* that typically require large shared information to be synchronized. Examples of the first category are *Canopy* for clustering and *Naive Bayes* and *K-Nearest Neighbours* (K-NN) for classification. *K-means* for clustering and *Gaussian Mixture* for classification represent typical examples of the second category. Finally, *Support Vector Machine* (SVM) cannot easily utilize MapReduce for speeding-up the execution of the method, hence it is a well representative of the third category. However, it is still possible to design an implement techniques falling in the third category in a way that they can utilize the benefits of using the MapReduce framework.

In clustering, Ene *et al* [14] have developed two clustering algorithms, namely, *MapReduce-kCenter* and *MapReduce-kMedian*, targeted to extend classical Data Mining methods to MapReduce framework, for efficiency purposes. The two algorithms run in a constant number of MapReduce rounds achieving a constant factor approximation. Experimentally, significant speed up of the proposed techniques have been reported. Another clustering technique has been proposed by Corderio *et al* in [7]. This technique aims at minimizing the I/O and the network cost, proposing the so-called "*Best of both Worlds*" *BoW* technique, which supports subspace clustering on very-large high-dimensional datasets on top of MapReduce. Papadimitriou and Sun [25] have used the MapReduce framework to develop a distributed co-clustering algorithm, that has been coined *DisCo*. Experimentally the technique was able to scale up to several hundreds to gigabytes of data. Utilization of MapReduce for K-means has been reported in [5], proving that a speed up of an average of 1.937 can be achieved on a dual core processor. A more generic contribution has been developed by Ghoting *et al* [17], which propose a generic toolkit for the development of Data Mining algorithms using MapRe-

duce, termed as *NIMBLE*. In classification, Chu *et al* [5] have also utilized MapReduce to speed-up Naive Bayes, *Neural Networks*, *Logistic Regression* and *Linear SVM*. It has been experimentally proven that running these techniques on a dual core processor speeds-up target techniques approximately by 2 times (in average: 1.950 for Naive Bayes, 1.905 for Neural Networks, 1.930 for Logistic Regression, and 1.819 for Linear SVM).

4 Recent Data Warehousing and OLAP Developments

Among the recent advances on Data Warehousing and OLAP over Big Data, *analytics over Big Data* play a relevant role in this respect. Let us focus on this research challenge in a greater detail. Analytics can be intended as complex procedures running over large-scale, enormous-in-size data repositories (like Big Data repositories) whose main goal is that of extracting useful knowledge kept in such repositories. Two main problems arise, in this respect. The first one is represented by the issue of conveying Big Data stored in heterogeneous and different-in-nature data sources (e.g., legacy systems, Web, scientific data repositories, sensor and stream databases, social networks) into a structured, hence well-interpretable, format. The second one is represented by the issue of managing, processing and transforming so-extracted structured data repositories in order to derive *Business Intelligence* (BI) components like diagrams, plots, dashboards, and so forth, for decision making purposes. Actually, both these aspects are of emerging interest for a wide spectrum of research communities, and more properly for the Data Warehousing and OLAP research community. As a consequence, this has generated a rich literature. At the industrial research side, Hadoop [3] and *Hive* [29] are two fortunate implementations of the ETL (*Extraction-Transformation-Loading*) layer and the BI layer of Big Data applications, respectively.

Although analytics over large-scale data repositories have been deeply investigated recently, the problem of extending actual models and algorithms proposed in this respect to the specific *Big Multidimensional Data* context plays a leading role, as multidimensional data naturally marry with analytics. Analytics over Big Data repositories has recently received a great deal of attention from the research communities. One of the most significant application scenarios where Big Data arise is, without doubt, scientific computing. Here, scientists and researchers produce huge amounts of data per-day via experiments (e.g., think of disciplines like high-energy physics, astronomy, biology, bio-medicine, and so forth) but extracting useful knowledge for decision making purposes from these massive, large-scale data repositories is almost impossible for actual DBMS-inspired analysis tools.

In response to this "*computational emergency*", the Hadoop system has been proposed, as above-mentioned. Hadoop runs MapReduce tasks over Big Data, and also it makes available the *Hadoop Distributed File System* (HDFS) [3] for supporting file-oriented, distributed data management operations efficiently. It has been highlighted that Hadoop is a kind of *MAD* system [6] meaning that (*i*) it is capable of attracting all data sources (*M* standing for *Magnetism*), (*ii*) it is capable of adapting its engines to evolutions that may occur in big data sources (*A* standing for *Agility*), (*iii*) it is capable of supporting depth analytics over big data sources much more beyond the possibili-

ties of traditional SQL-based analysis tools (*D* standing for *Depth*). In a sense, Hadoop can be reasonably considered as the evolution of next-generation Data Warehousing systems, with particular regards to the ETL phase of such systems.

Several studies, like [13], have provided recommendations for further improving the computational capabilities of Hadoop, whereas [1] proposes *HadoopDB*, a novel hybrid architecture that combines MapReduce and traditional DBMS technologies for supporting advanced analytics over large-scale data repositories. Furthermore, *Starfish* [20] is a recent self-tuning system for supporting big data analytics that is still based on Hadoop but it incorporates special features trying to achieve higher performance by means of *adaptive metaphors*. By looking at BI aspects of analytics over big data, the state-of-the-art research result is represented by Hive [29], a BI system/tool for querying and managing structured data built on top of the Hadoop's HDFS. Hive which allows us to obtain the final analytics components (in the form of diagrams, plots, dashboards, and so forth) from the big data processed, materialized and stored via Hadoop. Also, Hive introduces a SQL-like query language, called *HiveQL* [29], which runs MapReduce jobs immersed into SQL statements.

5 What is Next?!

Web 2.0 applications will continue generating Big Data for few years to come, along with the ever increasing volumes of scientific data that is generated continuously and in a very high data rate. Smartphones as an emerging source of Big Data have started to provide us with rich source of fine-grained sensory data. The data gravity principle has never been as true as today and it will become even more important in the near future. This principle as stated by the Data Mining guru *Gregory Piatetsky-Shapiro* is that *"the bigger the data, the harder it is to move it, so logic need to come to big data"*. Some new directions that are likely to continue in the Data Mining research area related to Big Data include the following topics. (*i*) *Mobile Data Mining* A focus of performing Data Mining locally on handheld devices has attracted a great deal of attention recently. Addressing the issues of limited resources and changing context of mobile users has been addressed in a large number of proposals. Examples include the work by Gaber [16] and Gomes *et al* [18]. (*ii*) *Embedded Data Mining in Wireless Sensor Networks* It has been proved experimentally that in-network processing of wireless sensor networks is the most feasible mode of operation for such networks. Accordingly, a number of techniques have been developed to mine data on board wireless sensor nodes. Accordingly, a number of techniques for mining data on board wireless sensor nodes have been developed (e.g., [30]).

As regards to Data Warehousing and OLAP research area related to Big Data, there still are a number of open research problems, some of which can be summarized by the following questions. (*i*) *How To Directly Integrate Multidimensional Data Sources Into The Hadoop Lifecycle?* Hadoop populates the underlying structured Big Data repositories from heterogeneous and different-in-nature data sources, such as legacy systems, Web, scientific data sets, sensor and stream databases, social networks, and so forth. Despite this, no research efforts have been devoted to the yet-relevant issue of *directly integrating multidimensional data sources* into the Hadoop lifecycle, which is an exciting

research challenge for next-generation Data Warehousing and OLAP research. (*ii*) *How To Model and Design Multidimensional Extensions of HiveQL?* In order to achieve an effective integration of multidimensional data models with analytics over Big Data, the query language HiveQL must be enriched with multidimensional extensions as well. These extensions should take into consideration language syntax aspects as well as query optimization and evaluation aspects, perhaps by inheriting lessons learned in the context of actual *MDX-like languages* for multidimensional data. (*iii*) *How to Design Complex Analytics over Hadoop-Integrated Multidimensional Data?* Multidimensional data provide add-on value to Big Data analytics. In this respect, design complex analytics over Hadoop-integrated multidimensional data plays a critical role. Actual analytics, although quite well-developed, still do not go beyond classical BI components, like diagrams, plots, dashboards, and so forth, but complex BI processes of very large organizations demand for *more advanced BI-oriented decision support tools*, perhaps by integrating principles and results of different-in-nature disciplines like statistics.

6 Summary and Conclusions

In this paper, we have discussed the emergence of Data Science and its consequent developments in the areas of Data Mining and Data Warehousing. We have also put in emphasis the natural marriage between Data Science and Distributed Intelligence paradigms, due to the inherent distributed nature of computational infrastructures supporting Data Science (like Clouds and Grids). As active researchers in this field, we have also highlighted possible future directions for further developments of both Data Mining and Data Warehousing areas related to Big Data. We observe how these directions will result from the Big Data phenomenon with extreme high gravity distribution. New models of data processing will be required in the near future, opening the door for new key players to take leading roles in the market.

References

1. A. Abouzeid, K. Bajda-Pawlikowski, D. J. Abadi, A. Rasin, and A. Silberschatz. Hadoopdb: An architectural hybrid of mapreduce and dbms technologies for analytical workloads. *PVLDB*, 2(1):922–933, 2009.
2. D. Agrawal, S. Das, and A. E. Abbadi. Big data and cloud computing: current state and future opportunities. In *EDBT*, pages 530–533, 2011.
3. Apache. Hadoop. http://wiki.apache.org/hadoop, July 2011.
4. BBC. Gap scraps new logo after online outcry. BBC Website: http://www.bbc.co.uk/news/business-11520930, October 2010.
5. C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. R. Bradski, A. Y. Ng, and K. Olukotun. Map-reduce for machine learning on multicore. In *NIPS*, pages 281–288, 2006.
6. J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstein, and C. Welton. Mad skills: New analysis practices for big data. *PVLDB*, 2(2):1481–1492, 2009.
7. R. L. F. Cordeiro, C. T. Jr., A. J. M. Traina, J. López, U. Kang, and C. Faloutsos. Clustering very large multi-dimensional datasets with mapreduce. In *KDD*, pages 690–698, 2011.
8. A. Cuzzocrea. Cams: Olaping multidimensional data streams efficiently. In *DaWaK*, pages 48–62, 2009.

9. A. Cuzzocrea. Retrieving accurate estimates to olap queries over uncertain and imprecise multidimensional data streams. In *SSDBM*, pages 575–576, 2011.
10. A. Cuzzocrea and S. Chakravarthy. Event-based lossy compression for effective and efficient olap over data streams. *Data Knowl. Eng.*, 69(7):678–708, 2010.
11. A. Cuzzocrea, F. Furfaro, G. M. Mazzeo, and D. Saccà. A grid framework for approximate aggregate query answering on summarized sensor network readings. In *OTM Workshops*, pages 144–153, 2004.
12. J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.
13. J. Dittrich, J.-A. Quiané-Ruiz, A. Jindal, Y. Kargin, V. Setty, and J. Schad. Hadoop++: Making a yellow elephant run like a cheetah (without it even noticing). *PVLDB*, 3(1):518–529, 2010.
14. A. Ene, S. Im, and B. Moseley. Fast clustering using mapreduce. In *KDD*, pages 681–689, 2011.
15. I. T. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *IJHPCA*, 15(3):200–222, 2001.
16. M. M. Gaber. Data stream mining using granularity-based approach. In *Foundations of Computational Intelligence (6)*, pages 47–66. Springer, 2009.
17. A. Ghoting, P. Kambadur, E. P. D. Pednault, and R. Kannan. Nimble: a toolkit for the implementation of parallel data mining and machine learning algorithms on mapreduce. In *KDD*, pages 334–342, 2011.
18. J. B. Gomes, M. M. Gaber, P. A. C. Sousa, and E. M. Ruiz. Context-aware collaborative data stream mining in ubiquitous devices. In *IDA*, pages 22–33, 2011.
19. H. Hacigümüs, S. Mehrotra, and B. R. Iyer. Providing database as a service. In *ICDE*, pages 29–38, 2002.
20. H. Herodotou, H. Lim, G. Luo, N. Borisov, L. Dong, F. B. Cetin, and S. Babu. Starfish: A self-tuning system for big data analytics. In *CIDR*, pages 261–272, 2011.
21. K. Hill. How target figured out a teen girl was pregnant before her father did. Forbes, February 2012.
22. C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M. J. Raddick, R. C. Nichol, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg. Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society*, 389(3):1179–1189, Sept. 2008.
23. M. Loukides. What is data science? the future belongs to the companies and people that turn data into products. An OReilly Radar Report, June 2010.
24. S. Muthukrishnan. *Data streams: algorithms and applications*. Foundations and trends in theoretical computer science. Now Publishers, 2005.
25. S. Papadimitriou and J. Sun. Disco: Distributed co-clustering with map-reduce: A case study towards petabyte-scale end-to-end mining. In *ICDM*, pages 512–521, 2008.
26. S. Papadimitriou, J. Sun, and R. Yan. Large-scale data mining: Mapreduce and beyond. Tutorial in KDD10, July 2010.
27. G. Soulellis. Emerging trends in big data and analytics. Big Data Innovation - London 2012, April 2012.
28. M. Stonebraker and J. Hong. Researchers' big data crisis; understanding design and functionality. *Commun. ACM*, 55(2):10–11, 2012.
29. A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Anthony, H. Liu, and R. Murthy. Hive - a petabyte scale data warehouse using hadoop. In *ICDE*, pages 996–1005, 2010.
30. J. Yin and M. M. Gaber. Clustering distributed time series in sensor networks. In *ICDM*, pages 678–687, 2008.