A Quick View on Current Techniques and Machine Learning Algorithms for Big Data Analytics

Josep Lluis Berral-García
Barcelona Supercomputing Center, Jordi Girona 29-31, 08034, Barcelona, Spain
Tel: 0034 93 4137716, Fax: 0034 93 4137721, e-mail: josep.berral@bsc.es

ABSTRACT
Big-data is an excellent source of knowledge and information from our systems and clients, but dealing with such amount of data requires automation, and this brings us to data mining and machine learning techniques. In the ICT sector, as in many other sectors of research and industry, platforms and tools are being served and developed in order to help professionals to treat their data and learn from it automatically; most of those platforms coming from big companies like Google or Microsoft, or from incubators at the Apache Foundation. This brief review explains the basics of machine learning with some ICT examples, and enumerates some (but not all) of the most used tools for analyzing and modelling big-data.
Keywords: Big Data, Analytics, Machine Learning, Knowledge Discovery, Frameworks

1. INTRODUCTION

Dealing with big-data usually involves finding on it the relevant information, modelling the elements composing it, and transforming it into useful information and knowledge. For such goals most of professionals prefer to use Machine Learning techniques for modelling and prediction, data aggregation and clustering, and knowledge discovery. Machine Learning, as part of Data Mining, provides methods to treat and extract information from data automatically, where human operators and experts are not able to deal with because of the level of complexity or the volume to be treated per time unit [1].

For some time, machine learning has been a science applied in very specialized environments like medicine, earth sciences or marketing, but due to the apparition of big-data everybody has the need to treat it and also learn from it automatically. As a response to this, lots of platforms, tools, languages and applications have appeared to treat that data while providing machine learning algorithms; some for constant data processing, some for discovering unusual events in data, some for distributed environments, but all bridging the until now existing gap between artificial intelligence and research and industry. Such tools also cover important aspects for applied machine learning processes, aside of accuracy and complexity of algorithms: aspects like the capacity of parallelizing the learning and prediction stages, the kind of programming language to use for modeling our problem and retrieve our data, the available already implemented libraries assuring high performance, or the way the results are going to be collected and displayed.

In this quick view I am going to explain briefly the concepts of machine learning and data mining, for those professionals who are not familiar with these topics, and then enumerate some tools and platforms used by data scientists to deal with big-data and perform machine learning processes on it. Notice that the tools explained here are not the only ones existing (neither the best ones for every situation), and I am enumerating them because of our experience on using them at the Barcelona Supercomputing Center and our experience on introducing students into data mining through those tools. I encourage the reader to play with the explained tools and also explore new and more complex tools and languages, on their professional projects and research.

2. DATA MINING AND MACHINE LEARNING

Data Mining is the science dedicated to obtain information from data and knowledge from information. This field provides theory and methodologies on how and how much we can obtain useful information from data sets, and how to obtain knowledge from the obtained information. Due to the apparition of Big-Data, data mining techniques have become more and more relevant, providing ways to scrap some usefulness to immense datasets, or just deal with bigger amounts of data driving systems. Some data mining techniques allow separating useful from sterile data, detecting noisy data, modelling the system generating such data, sampling data while preserving its properties, and more. While data mining is heavily used with commercial and marketing purposes, by modelling users and customers or creating recommendation systems, ICT and most engineering fields can benefit from those same used techniques.

Machine Learning is a field of science included in Data Mining, focusing on the inference of models from a-priori known data in an automatic way. Let's think in a situation where we have a system and we want to create a model describing this system, in order to understand it better, obtain predictions for custom inputs as an oracle for heuristics, or recreate its behaviour as a simulation. For many situations the complexity of our system escapes our capabilities of recreating it manually, we do not have enough detailed knowledge to do so, or the system changes so fast that we do not have time to create a new model each time that it happens. Machine Learning provides a set of algorithms and techniques, mostly of them based in statistics and probability, inferring an approximate model when provided enough observations of the target system.

2.1 Machine Learning Basics

We can classify most of the machine learning techniques in two big groups, depending on our purposes for the modelling: supervised or unsupervised. On supervised techniques our goal is to create a model from observed examples a-posteriori, to classify or estimate new examples a-priori. Our *observations* (a.k.a. *instances*) are composed by *features* (a.k.a. *variables* or *inputs*) and *labels* (a.k.a. *response variables* or *outputs*), and the resulting *model* becomes a function like $f(x_{1\ldots n}) = \hat{y}_{1\ldots m}$ where x_i are the features and $\hat{y}_j$ are the produced labels. A model is satisfactory when the difference between the produced labels $\hat{y}_j$ and the real corresponding labels y_j in the example set is small. We can iterate the modelling process with different modelling algorithm parameters (a.k.a. *hyperparameters*) until we are satisfied with the resulting model, but not without finally testing the selected model with examples not seen during the iterative training. E.g. consider data collected in a data-centre with VMs serving web-services, and we want to be able to predict how much load a VM will generate by just observing the amount of clients querying the service. As the infrastructure provider may not have permission to inspect her customers VMs, she collects examples of the amount of sessions and packets towards a customer VM, and also the VM required load. The provider then uses a regression tree algorithm to obtain a model that, providing a number of sessions and a number of packets towards the VM she can estimate the load that the VM will require if that scenario happens, and provision the VM without violating the agreed SLA or overprovisioning resources.

On unsupervised learning our goal is to create a model from observed examples that tells us how the data relates; this is, we do not have labels for our data, and we want the model to tell us which labels would describe our data properly. The procedure is similar to supervised but without example *outputs*, as the estimated *outputs* provided by our model are the new knowledge we are looking for. E.g. consider data collected from a network traffic log, and the ISP wants to know if its users respond to a limited set of behaviours or patterns. The ISP collects examples of traffic and then applies a clustering algorithm that provides a model discriminating users by traffic behaviour, grouping them in *k* groups, each group with users with similar behaviour. Then the ISP operator can check which characteristics each group has, and if she is satisfied with the obtained knowledge, the model can be used to label new users by using the most similar group generated label, and/or apply different policies on each group according their characteristics.

2.2 Algorithms and Approaches

There are different algorithms for modelling, prediction and clustering, each one with its strengths and weaknesses and each best for certain kinds of problems.

Some example of algorithms can be the *Tree* algorithms like CART, Recursive Partition Trees or M5 [2], using metrics like the gain of information for each feature to find those that discriminate better the examples, then building a decision tree based on such feature values, and placing a target class at each leaf. Such trees can also be used for regression if at each leaf an average value for all examples ending on it is placed, or a regression for them is placed, having then a regression for a piece-wise linear function (M5P). Another set of algorithms like the *Lazy* ones consist on memorizing examples and comparing new observations to the ones in memory, like k-Nearest Neighbours [3], where the training examples are kept in the model, and when a new observation arrives the nearest k examples are used to vote its class or to average its expected value. Another set of algorithms like the *Bayesian* ones (e.g. the Naïve-Bayes or the Bayesian Networks) [4], use the theorem of Bayes to compute the probability of a new example belonging to each class conditioned to its features, using the examples to compute the probabilities of each having a specific value on each feature according to each class. Here it will depend on the consideration of independence or dependence of the features to choose between the Naïve and the Network approach. More algorithms like the *Support Vector Machines* also create classification models by representing data in a hyper-space and looking for the hyper-planes that better divide the different classes [5]. There are also the *Artificial Neural Networks* [6] that attempting to imitate the structure of connected brain neurons can learn to classify, predict or even label data by connecting perceptrons (artificial neurons with usually an aggregation and activation functions) in layers, and connecting layers into networks. Finally, among even more algorithms not described here for lack of space and abundance of them, we have *clustering* algorithms like the k-means or the DBSCAN, trying to group data by similarities and highlight differences and similarities between the groups found.

For some years neural networks have experienced a period of "delusion" and such techniques were left just to classify images, considering the hard effort of preparing or tuning the algorithms for new and changing data different from images not worth it. But recently research on neural networks, focused on new ways to build or to train them, has brought techniques like Deep Learning, Constraint Restricted Boltzmann Machines, or Recurrent Neural Networks [7], in front of the nowadays common Feed-Forward Neural Networks. The fact that companies like Google and Facebook invested money and computational resources on them, finding direct applications on web-content classification and user modelling for marketing recommendations, helped these techniques to regain interest from the scientific community and the industry.

3. EXECUTION FRAMEWORKS

In Big-Data scenarios, operators, managers and data scientists want to obtain information and knowledge from huge data-sets or from wide streams of data. In order to make this process easy, letting data scientists to focus on automating this process and focus on the results, several frameworks have appeared to provide such service. Here a few of them, but not the only ones, are summarized.

3.1 Map-Reduce frameworks: Apache Hadoop and Spark

Most machine learning processes can be parallelized by realizing a computation process for each input and then aggregate the process outputs into the solution. Such processes can be solved using a Map-Reduce solution: data is split in parts, each part is computed in parallel and the results are aggregated into the solution. Apache Hadoop [8] is a widely used open-source framework in Java for such purposes. Users can deploy on their own clusters or data-centres, or can get a solution from companies offering Hadoop as Platform as a Service and focusing only on deploying their applications.

Based on Hadoop, Apache Spark [9] is a solution attempting to improve performance by focusing in specific functionalities like graph analysis, machine learning and data-streaming, programmed in Scala, and programmable in Scala or Python. While Hadoop has been on the market for long time and currently has more functions covering more aspects of the industry, Spark grows by focusing and improving specific problems, most of them related with machine learning and big data treatment.

3.2 Google's TensorFlow

Recently Google made public one of their frameworks for machine learning processes, TensorFlow [10], based on tensor operations and focused on conventional hardware but also of GPU devices. This technology is released after its extensive use inside Google for their services like Gmail, image search, voice recognition, et cetera. Released under License Apache 2.0, TensorFlow can be downloaded by users, installed on their deployments (clusters or servers with CPUs and GPUs), and used through Python scripting and its provided libraries, indicating which and how many CPUs and GPUs can be used.

3.3 Microsoft's Azure-ML

Microsoft Azure-ML is a platform based upon the Infrastructure as a Service platform Azure, oriented to deploy data-flow pipeline processes [11]. Users design the data-flow processes by indicating the sources of their data, the processes to be executed over their data, and the desired outputs. Processes can be already programmed ones, including aggregation functions, machine learning algorithms or data treatment functions. Also they can be deployed on Azure services for storage and computation. Further, Microsoft recently incorporated native support for R programming into their systems, allowing the user to program her own algorithms into the pipeline.

4. IMPLEMENTED LIBRARIES AND PLATFORMS

Most of the previously named algorithms have implementations publicly available, in different tools, platforms and libraries. Here some examples of the most popular ones among the immense existing amount of them are commented.

4.1 Languages: R-cran and Python Sci-Kit

R is a programming language oriented towards statistics and data analysis, with an open-source platform implementation and a wide community supporting it with libraries and plug-ins [12]. The base package already includes the fundamental statistic functions, linear regression algorithms and clustering methods, plus packages like *kknn*, *e1701*, *nnet* or *rpart* among others provide algorithms for supervised and unsupervised learning. Users can download the R interpreter and libraries, also IDEs for a more comfortable usage like RStudio. R language is also present on Hadoop (R-Hadoop) or Azure-ML (native interpreter), where R applications can be deployed and run on cluster systems as a Map-Reduce application or as a data-pipeline in Azure systems.

For Python users wanting to deploy application using machine learning, there is a library called SciKit-Learn [13] providing a wide sort of algorithms for classification, regression and clustering. The library, written in Python and Cython, is open-source and has extended documentation.

4.2 Prepared packages: Weka and MOA

For more than a decade a Java package from Waikato University named Weka [14] has been around machine learning users, beginners and experts, providing an extensive set of algorithms for classification, regression, clustering and data treatment. This package can be used as a standalone tool, where the user can execute its GUI and perform learning experiments, visualize data and save models, or the user can use its libraries and functions directly from her code. MOA [15] is a stream learning package from the same researchers, including packages for data stream mining. Both Weka and MOA are open-source software and fully documented.

4.3 Displaying Data: ElasticSearch + Logstash + Kibana

Examining and then displaying big-data, before or after machine learning processes, is often as much important as obtaining models or making predictions. If the most relevant found data or computed predictions cannot be transmitted to the corresponding professional or final user as a support for the obtained conclusions or recommendations, it has no meaning at all. One way to collect and index big-data towards browsing and communication is to use ElasticSearch [16], a search engine based on Apache Lucene [17], able to store and index data efficiently. The user can download ES and deploy on a single machine or an entire cluster, to store *documents* (instances of data) and browse them, also install associated packages like LogStash for processing *log-files* and importing them periodically into ES, and Kibana for displaying the updated data through a web-GUI. Even though ES does not perform machine learning it is able to aggregate data and allow users to discover knowledge and relations from their data.

5. CONCLUSIONS

Here in this paper I briefly explained the basic notions of machine learning and data mining, and named some of the most used tools and platforms for dealing with big-data using machine learning methods, focused on introducing those professionals unfamiliar with those topics into the data mining world, so they can incorporate new ideas and methods into their day-by-day industrial or research activity. Not entering into detail for each concept, the intention of this tutorial has been to show new ways of treating data, show new possibilities for problems with large data-sets, and point to some names for readers to start discovering new techniques for improving their solutions on big-data needs.

ACKNOWLEDGEMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 639595).

REFERENCES

[1] Hastie, T.; Tibshirani, R. & Friedman, J. (2001), The Elements of Statistical Learning, Springer New York Inc., New York, NY, USA.

[2] Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984). Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software.

[3] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 46 (3): 175–185.

[4] Russell, S.; Norvig, P. (2003). Artificial Intelligence: A Modern Approach (2nd edition). Prentice Hall.

[5] Cortes, C.; Vapnik, V. (1995). Support-vector networks. Machine Learning 20 (3): 273.

[6] Bishop, C.M. (1995) Neural Networks for Pattern Recognition, Oxford: Oxford University Press.

[7] LeCun, Y.; Bengio, Y. & Hinton G. (2015). Deep learning, Nature 521, no. 7553: 436-444.

[8] White, T. (2009). Hadoop: The Definitive Guide (1st edition). O'Reilly Media, Inc. Software available from https://hadoop.apache.org

[9] Zaharia, M.; Chowdhury, M.; Franklin, M.J.; Shenker, S. & Stoica, I. (2010). Spark: cluster computing with working sets. In Proceedings of the 2nd USENIX conference on Hot topics in cloud computing. USENIX Association, Berkeley, CA, USA. Software available from https://spark.apache.org

[10] Abadi, M. Et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from https://tensorflow.org

[11] Microsoft Azure Machine Learning. At https://studio.azureml.net , accessed in March 2016.

[12] R Development Core Team (2008). R: A language and environment for statistical computing. R foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org

[13] Pedregosa, F. et al. (2011) Scikit-learn: Machine Learning in Python, Journal on Machine Learning Research 12, pp. 2825-2830. Software available from https://scikit-learn.org

[14] Hall, M; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P. & Witten, I.H. (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1. Software available from http://www.cs.waikato.ac.nz/ml/weka/

[15] Bifet, A.; Holmes, G.; Kirkby, R. & Pfahringer, B. (2010). MOA: Massive Online Analysis. Journal on Machine Learning Research 11, pp. 1601-1604. Software available from http://moa.cms.waikato.ac.nz

[16] ElasticSearch. Software available from https://www.elastic.co/products/elasticsearch

[17] McCandless, M.; Hatcher, E. & Gospodnetic, O. (2010). Lucene in Action, Second Edition: Covers Apache Lucene 3.0. Manning Publications Co., Greenwich, CT, USA. Software available from https://lucene.apache.org