Considerations about Data Processing, Machine Learning, HPC, Apache Spark and GPU

Giang Nguyen, Ján Astaloš, Ladislav Hluchý

Department of Parallel and Distributed Information Processing
Institute of Informatics, Slovak Academy of Science
Dúbravská cesta 9, 845 07 Bratislava, Slovakia

{giang, astalos, hluchy}.ui@savba.sk

Abstract. Recently, the terms Internet of Things (IoT), Big Data and Machine Learning become very hot topics in both research and commercial spheres. IoT refers to the world of devices connected to the Internet, which is the way the massive amount of data is continuously collected, concentrated and managed. Raw data can also come from other processes such as information retrieval, web monitoring, database systems and so on. Mining in such data means of analysis in order to obtain usable results and/or knowledge. This paper presents several considerations about large-scale data, data processing and data mining using machine learning techniques with technological backgrounds towards high performance computing (HPC), Apache Spark and GPU that enable and accelerate the whole process.

Contribution type: Work-in-progress paper

Keywords: data processing, data mining, machine learning, HPC, Spark, GPU

1 Introduction

It is clear that machine learning (ML) algorithms learn from data and data is de facto the heart of many solutions. The availability of high performance infrastructures, technologies and available machine learning libraries in combinations with computational and/or data intensive strategies open nearly unlimited possibilities for data mining (DM). However, one important point is the flexibility of a solution design, which must be done around, at least, the 3Vs (*Volume, Velocity and Variety*) of data towards efficiency criterions such as resources, performance, cost efficiency, etc. A universal solution for the *"Big Data"* challenges does still not exist, however the coupling of strategies and technologies upon mathematical backgrounds and data-centric approach based on real requirements is a good starting point. In practical scenarios with big and large-scale data contexts, the use of incremental algorithms is visibly increased [4][6] with satisfied reported results of models' performance in comparisons with traditional in-memory algorithms.

2 Data mining using machine learning techniques

Nowadays, the global data production is continually increased by worldwide distributed ubiquitous sensors for long-term monitoring. Mining in such data means of analysis in order to obtain usable results and/or knowledge. Currently, ML techniques in general and supervised learning approaches in particular, play the central role in many practical/commercial cases. In general, ML approaches can be divided [1] into:

- Traditional in-memory learning (offline learning) where whole data for training can be loaded into machine memory. The main advantage of this approach is in many existing algorithms, number of available libraries, each with numerous methods and implementation improvements to achieve precise results. The disadvantage is the memory limitations that imply only use of small data sets.
- Incremental learning (online learning) does not require the whole data to be loaded into the machine memory at once. Instead, it loads the data in batches. These algorithms use limited memory and limited processing time per item, therefore, the input data set can be large-scale without memory limitation. On the other hand, the number of available algorithms are limited in comparison to in-memory approach.
- Distributed learning: which is typically coupled with infrastructure i.e. DAS (Data Analytics Supercomputer e.g. Apache Spark [2]). It is usually applied on very large data sets, which do not fit into memory of one machine. DAS is usually utilized also as a whole ecosystem with data processing, data integration and data management.

If a set of ready for use machine learning methods is extensive, their implementations are also rich and available in many languages with many versions and improvements. The most well-known ML libraries (or collections) are (Tab 1.):

Tab 1. The most well-known ML libraries

Library (impl. language)	Strong points	Weak points
Weka3 (Java)	general purpose, GUI, popular	small datasets, GUI, popular
MOA (Weka related)	data stream mining, concept drift, recommender systems	
R, Python (and libraries)	statistics, ML, very popular	R vs. Python
RapidMiner	general purpose, DB connection, popular	
Scikit-Learn (Python)	general purpose, popular	small datasets
NLTK (Python) Clojure	general purpose, natural language toolkit and text mining	small datasets
PyBrain (Python)	neural network, reinforcement learning, evolution, easy use	good for study and experiments
MLLib (Scala, Java)	Spark distributed scalable ML framework, growing community	coupled with infrastructure
Mahout (Java)	Hadoop ML framework	come with Hadoop overhead

H2O.ai	massively scalable Big Data analysis, distributed processing (Hadoop, Spark)	
Shogun (C++)	general purpose, designed for large scale learning, kernel methods, SVM, HMM	
LIBSVM (C++) LIBLINEAR (C++)	integrated software, large-scale data	narrowed approach
Vowpal Wabbit (C++)	fast out-of-core ML system, on-line learning	limited number of algorithms
XGBoost	parallelized general purpose gradient boosting library	narrowed approach
MatLab, GNU Octave	scientific libraries	math oriented

One of the most used *data mining concept and methodology* [8] is CRISP-DM (Cross-Industry Process for Data Mining), which consists of six steps: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment. The Data Preparation step consists of sub-steps: Data Transformation, Exploratory Data Analysis (EDA) and Feature Engineering. The group of the first five steps are also called the development phase. The deployment step is also called the production phase.

Although the main interest upon DM/ML is broadly paid to the Modeling step and algorithms, one important point remains the fact that ML algorithms learn from data. Therefore, in practice, Data Understanding and Data Preparation can consume up to 80% of the entire time of every DM using ML techniques project. Data Preparation is also slangy labeled Data Munging or Data Wrangling, which refer to strenuous work. Certain problem-solving techniques e.g. Forward Selection, Backward Eliminations in the Feature Engineering sub-step or grid-search in the Modeling step can lead to computational intensive tasks especially when ML input data is large-scale or big. HPC (high-performance computing) cluster can be utilized for concurrent training of models in order to shorten the development time.

In the following parts, some practical notes around data processing and DM process using ML techniques for commercial and research applications with IISAS participation in recent years are presented.

Malicious behavior detection in mobile devices log domain. When everyone owns and uses mobile devices such as smartphones and/or tablets, the demand of *cybersecurity* and *situational awareness* is pushing towards. This involved work was a part of the six-month pilot research done for IBM Slovakia. The interest was if it is possible to detect malicious behaviors of mobile devices based on collected logs of mobile devices. Raw data - logs from mobile devices belongs to human-generated data class, which are not so *"Big"* as machine-generated data. Data mining using ML techniques in this domain involved through following obstacles:

- Collected raw logs are extremely noisy for the specific detection purpose. The logs contain a lot of information about continuous monitoring processes such us timing (clocks, alarms, calendars), positions, accelerators, display setting and adapting, network and power monitoring, scanning processes, etc.

- Low occurrences of malicious behaviors - malware related activities, which caused imbalanced classes of data used for supervised ML;
- Feature extraction for data with evolving characteristics i.e. number of applications on mobile devices is changed based on users' demands without any limitations;
- Privacy preserving data mining of personal sensitive information.
- DM process required thorough Data Understanding in collaboration with domain experts, Data Preparation (especially EDA) and Feature Engineering. ML technique applied in this case was simple supervised binary classification with incremental learning. The obtained results were highly satisfied to distinguish malicious behavior from the normal one.

Click-through-rate advertising: raw and ML data are really big in both development and production phases. Applied analyzing techniques are e.g. reservoir (sub)sampling, biases monitoring, smoothing, sliding windows with settable size, forgetting mechanism, etc. came with *adaptive online learning* (retraining in combination with incremental adaptation). ML data is highly imbalanced as usually in many commercial cases that implies boosting one class against the second by reducing number of negative examples. Feature selections and feature combinations are also utilized to improve models' performance. The production infrastructure is high-performance Hadoop cluster of the Magnetic Media Online, Inc. technology company (USA).

Power utility for functional awareness of monitoring stations: raw input data in this case is quite interesting, it is not *"Big"* in any one of 3Vs, but contains pure numerical and structured data collected from monitoring stations during several years. Such data can be called large-scale, which causes computational intensive tasks with memory consumption in the development phase. The question was if it is possible to realize the production on single machine with limited memory due to cost and energy efficiency. The solution can be any of traditional in-memory approach in a machine with larger memory, incremental learning or distributed learning with Spark installation in single machine for the production. However, the use of the incremental learning to overcome machine memory limitation can be the less painful way on both phases.

3 Machine learning and many-core accelerators

In the recent years the accelerators have been successfully used (not only) in machine learning and deep learning applications [4]. Manufacturers often offer the possibility to enhance hardware configuration with many-core accelerators to improve machine/cluster performance. If we look at the list of top 500 most powerful supercomputers, we can see the increasing trend in both number of systems that employ the accelerators and their performance share. Most popular models of accelerators are based on *MIC (Many Integrated Cores)* and *GPU (Graphics Processing Unit)* architectures. The accelerators are able to offer significant performance increase for many application domains e.g. the work [5] realized in collaboration between TUKE (Technical University of Košice) and IISAS (Institute of Informatics, Slovak Academy of Science). The main feature of the many-core accelerators is massively parallel architecture (e.g. new NVIDIA P100 accelerator contains 3840 CUDA cores), allowing them to speed up computations that

involve matrix-based operations, which is a heart of many ML implementations. Many popular ML frameworks and libraries already offer the possibility to use GPU accelerators to speed up learning process with supported interfaces in various languages e.g.:

Tab 2. Popular ML frameworks and libraries

Library (impl. language)	Main purposes
Theano (Python)	math expression compiler
Tensorflow (C++, Python)	numerical computation library by data flow graphs
Keras (Python)	minimalist, highly modular neural networks library capable of running on top of TensorFlow or Theano
Caffe (C/C++, Python, MatLab, CLI)	deep learning framework for image processing
CNTK (C++, CLI)	unified deep-learning toolkit that implements CNN and RNN training for speech, image and text data
DL4J (Java, Scala)	distributed deep-learning library written for Java and Scala, integrated with Hadoop and Spark
Neon (Python)	Nervana's Python-based deep learning library
Torch (C/LuaJIT)	NN and optimization libraries that puts GPUs first
MatConvNet	Convolutional Neural Networks (CNNs) for MatLab

Some of them also allow to use optimized CUDA Deep Neural Network (cuDNN) library to improve the performance even further. Similar to the ML libraries mentioned in Section 2, ML libraries with GPU support are also diverted in various implementation levels for various specific purposes such as image, voice and text processing.

The demand for even more powerful hardware for deep learning applications caused that main manufacturer of GPU accelerators NVIDIA made considerable investments to the development of the new architecture called Pascal and special purpose system DGX-1 optimized for many-layered DNN. Among the new features most notable are the *„half-precision"*, which allows to reach 21.2 Teraflops and 160 GB/s bidirectional interconnect that significantly improves the scalability in multi-GPU systems.

The matrix-based operations on Apache Spark can be computationally accelerated under same logic like GPU/CUDA acceleration. Here is a *similar logic* between Apache Spark vs. GPU processing (not only) from ML viewpoint:

- If data fits into memory of one machine, GPU is faster, otherwise Spark;
- Spark logic is similar to CUDA host logic in the mean of SIMD processing;
- Spark network overhead vs. PCI-express transfer overhead;
- MapPartitions is like kernel launch, partitions are like CUDA blocks;
- Model parallelism vs. data parallelism: Data parallelism presents single instruction to multiple data items, ideal workload for a SIMD computer architecture; Model parallelism gives every processor the same data but applies a different model to it; Hybrid approach presents combination of data and model parallelism.

Potential benefits[1, 2] of using GPUs to further accelerate Spark performance is also done with positive results.

4 Conclusions

This paper presents a few considerations about working and mining in large-scale data using ML techniques in our department in recent years. We hope that such notes are useful for readers with nearby research interests and would like to thank to colleagues and reviewers for consultations and advices on the paper preparation.

Acknowledgements: This work is supported by projects VEGA 2/0167/16 and EGI-Engage EU H2020-654142. Simulations and technical realization are realized on the hardware equipment obtained within the project SIVVP ERDF ITMS 26230120002.

Remarks: In addition to standard HPC computational power, SIVVP[3] (Slovak Infrastructure for High Performance Computing) HPC clusters are also enhanced by GPU accelerators NVIDIA M2050/M2070 (448 CUDA cores) and K20 (2496 CUDA cores) to allow researchers from Slovakia to use GPU accelerated systems for research purposes. The installed GPU capacity is as follows: Institute of Informatics SAS, Bratislava: 16x K20 + 2x M2070; Matej Bel University, Banská Bystrica: 6x K20 + 2x M2070; Technical University of Košice: 2x K20 + 2x M2070; Institute of Experimental Physics, Košice: 10x K20 + 32x M2070; University of Žilina: 2x M2070; Slovak University of Technology in Bratislava: 8x M2050.

References

1. Gama J, Žliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A.: A survey on concept drift adaptation. ACM Computing Surveys (CSUR). 2014 Apr 1;46(4):44 pages.
2. Karau H., Konwinski A., Wendell P., Zaharia M.: Learning Spark. Published by O'Reilly Media, Inc. © 2015 Databricks, 242 pages, ISBN: 978-1-449-35862-4.
3. Lacey G., Taylor G. W., & Areibi, S. (2016). Deep Learning on FPGAs: Past, Present, and Future. arXiv preprint arXiv:1602.04283.
4. Lopes N., Ribeiro B.: Machine Learning for Adaptive Many-Core Machines - A Practical Approach. Studies in Big Data, Volume 7, Springer International Publishing Switzerland, 2015, 251 pages, ISBN 978-3-319-06937-1, ISSN 2197-6503.
5. Naščák D., Koštial I., Mikula J., Olijar A., Astaloš J.: Acceleration of simulation models for raw materials thermal treatment. 12th International Carpathian Control Conference: ICCC´2011, pp. 207-212, ISBN 978-161284359-9.
6. Rozinajová V. et al: Otvorené smery výskumu v oblasti dátovej analytiky. WIKT 2015 proceedings, pp.4-7, ISBN 978-80-553-2271-1.

[1] http://www.slideshare.net/continuumio/gpu-computing-with-apache-spark-and-python

[2] http://www.nextplatform.com/2016/02/24/hadoop-spark-deep-learning-mesh-on-single-gpu-cluster/

[3] SIVVP - Slovak Infrastructure for High Performance Computing (http://www.sivvp.sk/)

7. Sumeet Dua and Xian Du: Data Mining and Machine Learning in Cybersecurity. CRC Press, Taylor & Francis Group, 248 pages, 2011, ISBN-13 978-1-4398-3943-0.
8. Vadovský, M., Michalik, P., Zolotová, I. and Paralič, J.: Better IT services by means of data mining. IEEE Int. Symposium on Applied Machine Intelligence and Informatics SAMI 2016, pp. 187-192, 2016, ISBN 978-146738740-8.