2017 IEEE International Parallel and Distributed Processing Symposium Workshops

Teaching Future Big Data Analysts: Curriculum and Experience Report

Joshua Eckroth
Math/CS Department
Stetson University
DeLand, Florida
jeckroth@stetson.edu

***Abstract*—This paper documents the learning objectives, curriculum design, technology infrastructure, and classroom experience for a "big data mining and analytics" course at a small liberal arts college. The course serves as an elective for our Data Analytics minor as well as an elective for computer science and computer information systems majors. The course introduces students to data analysis, statistics, and plotting with Unix tools and the R language. It then transitions into big data projects making use of Apache Hadoop, HDFS, and Map-Reduce; Apache Spark; Apache Hive; and related tools. A primary learning objective is that students demonstrate the ability to identify which tools are most appropriate for specific datasets and data analysis tasks. We also expect students to be able to communicate their findings to a general audience. As potential future data analysts, we aim to give students the skills and sensibility to efficiently solve data analysis problems, big data or otherwise, in their future careers.**

***Keywords*-curriculum; big data; data mining; data analysis;**

I. Introduction

Stetson University, a small private liberal arts college, recently introduced an interdisciplinary Data Analytics minor that is designed in part to "prepare students for entry-level jobs in fields that apply Data Analytics and for graduate work in disciplines that utilize Data Analytics." In addition to a variety of core courses that cover programming, statistics, and databases, students may choose one of several data analysis courses. One such course, known as Big Data Mining and Analytics, focuses on parallel and distributed computing in order to perform data mining and analysis on very large datasets.

This paper documents the learning objectives, curriculum design, technology infrastructure, and classroom experience for Stetson's "big data" course, as it is known among the students. Computer science and computer information systems majors may also take this course as an elective, even if they are not working towards a Data Analytics minor. The prerequisites for the course do not include a statistics or database course, so a background in these subjects cannot be assumed. The course is designed to cover these topics in sufficient depth to ultimately work with big datasets and find meaningful insights from the data.

The desire for more data analysts in the workforce is clear [1]. "Big data," while overused and often considered a "buzz word," is a growing concern both inside and outside of academia. Truly big datasets can break traditional data analysis tools, and a wide variety of new tools have been developed to handle big data. Such tools include Apache Hadoop, Apache Spark, Apache Hive, BigML, Google BigQuery, et al. These tools require more computational resources and more complex workflows than traditional tools. Based on personal communications with industry leaders, we have learned that organizations want data analysts to know when to make use of big data tools, and which ones, and when to use simpler and less costly tools for smaller datasets. Thus, the big data course is designed not only to expose students to a variety of tools and techniques, but also to encourage development of a model or decision matrix for choosing the right tool for each job.

The remainder of this paper is organized as follows. Section II details our learning objectives for the course. Section III through Section VII summarize our curriculum for meeting these learning objectives and include sample projects. Section VIII demonstrates an example decision matrix for identifying the right tool for a data analysis job. Sections IX and X cover our course technology setup and experience in the classroom. Finally, Section XI offers a discussion and concluding remarks.

II. Learning objectives

Our learning objectives for this course focus on teaching students to make appropriate use of technology and effectively communicate their findings.

- Determine whether a data analysis task requires "big data" tools and techniques, or can be done with traditional methods, and identify appropriate tools to perform the analysis.
- Skillfully make use of tools to perform the data processing and analysis.
- Communicate the outcomes of data analysis with convincing arguments involving, as appropriate, text, tables, and plots.

These learning objectives are met by gradually increasing the complexity of data analysis projects over the term. In early projects, students are asked to perform simple tasks using tools like R and Unix command-line tools. In these early projects, they are not required to determine the

978-1-5386-3408-0/17 $31.00 © 2017 IEEE
DOI 10.1109/IPDPSW.2017.122

appropriate tool for the job. In later assignments, students are given more freedom to solve the data analysis problem in their own way. In all projects, students are required to communicate their findings by hiding the data analysis code and only presenting relevant text, tables, and plots in PDF form. Their work is graded 80% on completeness and 20% on clarity and the degree to which their report is convincing.

In the following sections, our curriculum is summarized in a topic breakdown that progresses from simple data analysis tasks to more complex, big data analysis tasks.

III. Distinguishing big data from small data

Perhaps the most important skill a data analyst may possess is the skill to determine the right tool for a job. Due to the current popularity and hype surrounding data analysis, and big data analysis in particular, many high-powered tools are available and new tools continue to be developed. However, these tools are often more complex and require greater computational resources than traditional software like Unix command-line tools, Microsoft Excel, and statistics packages like R, SPSS, and others. Only when the data is so large that traditional tools cannot be used should a data analyst consider making use of big data tools.

To this end, we begin our course by defining "big data," adapting a definition from Rossum [2]:

> A data analysis task may be described as "big data" if the data to be processed have such high volume or velocity that *more than one* commodity machine is required to store and/or process the data.

We define a "commodity" machine as one that may be found in a typical small business data center. By today's standards, such a machine may contain a 2-8 core CPU, 8-16 GB memory, 1-4 TB hard disk, and gigabit ethernet. These numbers are vague approximations but sufficient to make the point: big data is at least several TB of data volume or millions of records per second data velocity.

Our definition of big data does not imply that smallish data, e.g., 10-100's of GB of data, can be analyzed by every traditional tool. For example, Microsoft Excel would surely crash when attempting to load such a dataset. In our experience, R also suffers as all the data must be kept in memory (RAM), though third-party libraries like Feather [3] are available to fix this limitation.

The first project in our course asks students to find two projects that purport to require "big data analysis" and argue whether or not the dataset truly qualifies as big data. In this first project, students also practice with basic R statistics functions and data manipulations in order to prepare for future projects.

IV. Data munging and cleanup

Data munging [4] and data cleanup describe the process of obtaining datasets, removing invalid or unnecessary values, resolving inconsistencies, and other transformations in order to arrange the data in a form easily processed. Often, the resulting form is a two-dimensional table (e.g., CSV files) or hierarchical form (e.g., HDF5 files). In our experience, a significant portion of a data analyst's time is spent in data munging tasks. Without sufficient data munging and cleanup, further analysis is impossible.

Students practice this process with a project whose statistical analysis component is minimal but whose data munging component is significant. They are asked to answer the question, "Do students from low-income areas of the US more often apply for and obtain student loans?" To do so, they download public FAFSA data from the US Student Aid website [5] and merge these data with IPEDS data [6] regarding the number of students, per institution, who apply for student loans. The data are available in numerous inconsistently named Excel files formatted for human consumption. The files contain additional columns and headers, colors, and other extraneous information that must be cleaned before further analysis is possible. Students may choose to use Unix tools like grep and awk, Perl code, and/or R code to transform the files. As our course is a junior/senior-level course, we require that students write code to automate their process rather than perform the operations manually.

V. Small data analysis and visualization

The small data analysis and visualization section of our course exposes students to normal data analysis techniques like statistical summarization, correlation tests, regression, and typical visualization techniques such as scatter plots, box plots, and histograms. Our course makes use of the R toolset and ggplot plotting library, but many other tools should prove equally capable in this section of the course.

At this time, we tell students that big data analysis often ends with small data analysis of the usual sort. We encourage them to transform big data into small data as early as possible to ease future analysis.

The project related to this section asks students to analyze a 3 GB dataset made available by Backblaze [7]. Backblaze describe the dataset as follows.

> Each day in the Backblaze data center, we take a snapshot of each operational hard drive. This snapshot includes basic drive information along with the S.M.A.R.T. statistics reported by that drive [7].

The students are asked to respond to the following prompts about the data:

- "Some manufacturer produced particularly bad drives of a certain size (in TB). Find this manufacturer by graphing annual failure rate (AFR) for all manufacturers and drive sizes."
- "Drives of a certain TB size-range (e.g., 3-4TB) have a higher proportion of failures than other sizes."

- "The hard disk S.M.A.R.T. stat #187, which measures read errors, is strongly positively correlated with disk failure."

Students are required to perform all of the operations in R exclusively. This can be a painful process as the dataset is just large enough to cause R to experience significant slowdown. Note that students normally access R by remotely accessing an RStudio Server instance provided by the department, so they are not required to possess a personal machine capable of keeping 3 GB of data in memory.

Students are asked to submit PDF reports of their findings rather than raw R code. Their work is evaluated on their use of appropriate statistical tests and visualizations and convincing arguments.

VI. Big data storage and processing

At this point, about half-way through the course, we introduce big data tools such as Apache Hadoop, HDFS, and Hive. In particular, we first examine HDFS as a distributed large file store, and investigate Hive as a data store built on top of HDFS that provides a SQL-like interface. Next, we introduce the Map-Reduce processing paradigm and Spark's more general directed acyclic graph processing paradigm. As mentioned previously, we encourage students to treat big data tools as a means to efficiently extract small data which is then used for analysis.

The project related to this section of the course involves the full dump of Stack Exchange questions and answers [8]. At the time we obtained the dataset, it measured 116 GB of uncompressed XML files, but it has surely grown since that time. First, students are asked to write Map-Reduce jobs in Java or Python to summarize the dataset and produce a report in order to answer the following questions:

- "Do high-reputation users answer questions faster?"
- "Is it better to answer a question very quickly (or even first) to earn high reputation, or does time-to-answer not matter much at all?"
- "Does the first answer usually get the most votes?"
- "Does the first answer usually get accepted?"

After completing the Map-Reduce jobs, we then ask students to import the dataset into Hive using custom regular expressions to extract the few relevant fields out of the XML files upon import. Then, they are asked to attempt to arrive at the same conclusions by querying the dataset using Hive's query language. Inevitably, students find that Hive is much easier to use than Map-Reduce for these kinds of straightforward queries.

In the current iteration of this course in Spring 2017, we plan to introduce a second big data processing project that does not benefit from Hive. In this project, students are provided with 10,000 6000x6000 pixel images of the night sky from the Pan-STARRS1 data archive [9], constituting 331 GB. Students are asked simply to count the stars in each image. Then, using a script provided to them, their star counts are overlaid as a two-dimensional density plot atop a grid of the stitched star images. In order to count the stars, they must make use of a technique like Hough circle detection from OpenCV [10] as well as a distributed processing pipeline to parallelize the work.

VII. Big machine learning

The final section of the course addresses machine learning on big data. Machine learning includes unsupervised techniques like clustering and supervised techniques like learning spam detectors. In big data sets, a tool like Apache Mahout running on Apache Spark may be appropriate. Mahout supports stochastic singular value decomposition, naïve Bayesian classification, and recommendation engine modeling.

We also introduce Weka [11] as a workbench for experimenting with many different machine learning algorithms. Weka does not handle large datasets, so data must be transformed into small data ahead of time.

Many projects may be appropriate for the machine learning section of the course. We have had success with two projects in a past offering of this course. First, we asked students to train a model to classify email messages as spam/ham. We used the TREC 2007 corpus [12]. The corpus is relatively small at just 700 MB spread across 75k email messages, but serves as a good practice dataset for Mahout and Weka workflows.

In the second project, students were tasked with an unsupervised learning challenge. Given 10,000 images of cats, with the cat's face coordinates marked in every image, determine whether or not distinct breeds of cats are represented in the dataset. Most students chose to cluster the images according to the top one or two colors of each cat's fur. Although fur color does not uniquely identify a cat's breed, the resulting clusters, plotted using multidimensional scaling in R and using the original images in thumbnail sizes, clearly showed some distinct groups. Unfortunately, this dataset is no longer publicly available.

As before, students were required to write a report of their findings that emphasized their supporting arguments rather than the specific code or procedures used to reach those findings.

VIII. Decision matrix

Our first learning objective states that students should be able to determine the appropriate tools to use for a data analysis task. In the current iteration of this course, students are asked to build a personal decision matrix that relates features of a data analysis task with tools introduced in the course or tools that they find in their own research. A sample decision matrix is shown in Table I, although each student's matrix may differ. Throughout the course, we plan for students to share their decision matrices and learn from each other to fill out the various features and tools.

Table I
A SAMPLE DECISION MATRIX THAT RELATES FEATURES OF A DATA ANALYSIS TASK WITH VARIOUS TOOLS.

Feature	Unix tools	Excel	R	MySQL	Hive	Spark	Mahout	Weka
Exploratory analysis	•	•	•					•
Plotting		•	•					
Batch download and web scraping	•		•					
SQL-like queries (small-medium data)				•				
SQL-like queries (big data)					•			
Arbitrary processing (small-medium data)	•		•					
Arbitrary processing (big data)						•		
Image processing (small-medium data)	•		•					
Image processing (big data)						•		
Machine learning (small-medium data)			•					•
Machine learning (big data)							•	

IX. COURSE SETUP

Big data mining and analysis require substantial computational resources. For most small universities and colleges, cloud computing services such as Amazon AWS, Google Compute Engine, and Microsoft Azure are the only feasible way to give students hands-on experience with big datasets. Small grants are often available to offset the costs of these services, e.g., GitHub's Student Developer Pack for Amazon credits, Google Cloud Platform Education Grants, and Azure in Education grants. Alternatively, the costs can be covered by students in a "lab fee." Rabkin et al. [13] documented their experiences with Amazon's cloud computing resources made available through a grant. Whether the resources are paid via grants or lab fees, the cost is fixed and predetermines the amount of computational resources available to the student. Thus, students must plan carefully how they will use the resources without running out. Such planning can be challenging since students are not experts in big data analysis tools and faulty coding or misconfigured tools can lead to significant budget overruns.

An alternative or supplementary approach is to use in-house computing resources. The Math/CS department at Stetson University possesses a moderately-sized server capable of hosting many virtual machines. We configured this server to run Apache Hadoop in a virtual cluster environment, documented previously [14]. Excluding electrical power and hardware maintenance, the virtual cluster has a fixed cost and can be configured for larger or smaller virtual clusters to illustrate the benefits of parallel workloads as well as the overhead of distributing jobs and data to different machines. Before the course began, we downloaded various large datasets and loaded them into the virtual cluster's HDFS storage facility.

X. EXPERIENCE IN THE CLASSROOM

The Big Data Mining and Analytics course was taught in Spring 2015 and is currently in-session in Spring 2017. We aim for yearly repetitions in Spring 2018 and beyond due to the popularity and relevance of the course.

Student feedback about the course was consistently high in measures of usefulness and relevance, quality of the lecture material and projects, and appropriateness of the workload. More interestingly, students reported strong growth during the course. Table II shows average anonymous ratings by 17 students for questions relating to their personal growth.

Several students added anonymous comments that praise various aspects of the course. They generally found the projects to be well designed, challenging, and relevant, and they enjoyed the hands-on nature of the course. Responses to the query, "Which aspects of this course helped you learn the most?" include:

- "[...] mostly the projects and how they were structured, they allowed us to take the content from the class and apply it."
- "The hands on labs and extremely well put together projects."
- "The practice with the various tools and software used in the class."
- "Hands on experience with current trends in the industry."
- "The projects that involved finding our own solution to a data analytics problem were the most challenging and I learned a lot from them. It was also nice to learn new techniques from the student presentations."

On the other hand, two negative comments stand out as warnings for teachers of a similar course. In response to the query, "Do you have suggestions for improving any aspects of the course? If so, how might the course be improved?" two students wrote,

- "Add additional servers for student use."
- "Elaborate on key concepts more and explain things that will be needed for particular projects in more detail instead of just vaguely going over everything and leaving us to figure out the projects ourselves."

Regarding the first comment, as documented previously [14], the virtual cluster is not as performant as a comparable physical cluster in workloads with heavy disk I/O. We encourage teachers to consider making use of cloud

Table II
STUDENT RATINGS OF THEIR PERSONAL GROWTH DURING THE COURSE. EACH PROMPT HAS A RANGE OF 0 TO 5 REPRESENTING CHOICES "NO APPARENT PROGRESS" (0) TO "EXCEPTIONAL PROGRESS" (5). $n = 17$

Prompt	Mean	Std dev.
Gaining factual knowledge (terminology, classifications, methods, trends)	4.29	0.89
Learning fundamental principles, generalizations, or theories	4.18	0.86
Learning to apply course material (to improve thinking, problem solving, and decisions)	4.18	0.78
Developing specific skills, competencies, and points of view needed by professionals in the field most closely related to this course	4.19	0.96
Learning how to find and use resources for answering questions or solving problems	4.24	0.81
Acquiring an interest in learning more by asking my own questions and seeking answers	4.18	0.92

computing resources to alleviate this bottleneck. The second comment reflects on our choice to gradually introduce more open-ended and underspecified projects in order to encourage students to identify appropriate tools and techniques on their own. However, a careful balance must be maintained between providing a complete set of procedures and tools for students to follow and leaving students entirely on their own.

XI. DISCUSSION AND CONCLUSION

The Big Data Mining and Analytics course is one of the more popular and anticipated courses in the Math/CS department at Stetson University. Students appear to be encouraged by successful job placement in data analyst positions from recently graduated seniors who took the course in 2015, and by the focus on practical and current tools like R, Hadoop, and Weka.

This course suffers from a few limitations. First, the course does not make use of truly big datasets on the order of TB or larger due to limited publicly available datasets of that size and limited resources including time required to complete each project. Nevertheless, we feel that students gain practical experience with big data tools, and the students' feedback indicates that they agree. Second, the course only covered a subset of available tools. For a discussion of more options, refer to Dobre and Xhafa [15], who review a wide variety of tools for both big data volume and velocity situations.

This course differs from similar courses at other institutions in its wide range of topics from small data analysis and visualization to big data processing. For a more narrowly focused course, review Silva's, et al.'s account [16] of a course focused on big data tools including Map-Reduce and Hive. They did not include a data analysis component in their course, which is a central feature of our course. Likewise, Ngo, et al. [17] developed a course focused on Map-Reduce.

In the current and future iterations of this course, we aim to make use of the decision matrix described in Section VIII and introduce new tools like Apache Kafka for data streams with high velocity. We also plan to continuously update the projects in order to keep up to date with publicly available datasets, new data mining and analysis tools, and maintain the overall "freshness" of the course.

XII. ACKNOWLEDGMENTS

This project was supported by the Brown Center for Faculty Innovation & Excellence at Stetson University.

REFERENCES

[1] T. H. Davenport and D. Patil, "Data scientist: The sexiest job of the 21st century," *Harvard Business Review*, October 2012.

[2] P. Russom, "Big data analytics," in *TDWI Best Practices Report, Fourth Quarter*, 2011.

[3] W. McKinney and H. Wickham, "Feather: fast, interoperable data frame storage," https://github.com/wesm/feather.

[4] J. Heer and S. Kandel, "Interactive analysis of big data," *XRDS: Crossroads, The ACM Magazine for Students*, vol. 19, no. 1, pp. 50–54, 2012.

[5] "Federal Student Aid," https://studentaid.ed.gov/sa/.

[6] National Center for Education Statistics, "The Integrated Postsecondary Education Data System," https://nces.ed.gov/ipeds/.

[7] "Backblaze hard drive data and stats," https://www.backblaze.com/b2/hard-drive-test-data.html.

[8] "Stack Exchange Data Dump," https://archive.org/details/stackexchange.

[9] H. Flewelling, E. Magnier, K. Chambers, J. Heasley, C. Holmberg, M. Huber, W. Sweeney, C. Waters, T. Chen, D. Farrow *et al.*, "The Pan-STARRS1 Database and Data Products," *arXiv preprint arXiv:1612.05243*, 2016.

[10] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[11] E. Frank, M. A. Hall, and I. H. Witten, *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*, 4th ed. Morgan Kaufmann, 2016.

[12] G. V. Cormack and T. R. Lynam, "Trec 2007 public corpus," http://plg.uwaterloo.ca/ gvcormac/spam/, 2005.

[13] A. S. Rabkin, C. Reiss, R. Katz, and D. Patterson, "Experiences teaching mapreduce in the cloud," in *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*. ACM, 2012, pp. 601–606.

[14] J. Eckroth, "Teaching big data with a virtual cluster," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. ACM, 2016, pp. 175–180.

[15] C. Dobre and F. Xhafa, "Parallel programming paradigms and frameworks in big data era," *International Journal of Parallel Programming*, vol. 42, no. 5, pp. 710–738, 2014.

[16] Y. N. Silva, S. W. Dietrich, J. M. Reed, and L. M. Tsosie, "Integrating big data into the computing curricula," in *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, 2014, pp. 139–144.

[17] L. B. Ngo, E. B. Duffy, and A. W. Apon, "Teaching HDFS/MapReduce systems concepts to undergraduates," in *IEEE International Parallel & Distributed Processing Symposium Workshops (IPDPSW)*, 2014, pp. 1114–1121.