Big Data Analytics CS594 (007) / CS690 (001), Fall 2018

University of Tennessee, Knoxville

Meeting Time and Place: Monday, 8AM – 11AM ET, Min Kao Engineering 419

Course Credit Hours: 3 hours

Instructor: Dr. Michela Taufer

Assistant Instructors: Danny Rorabaugh, Dylan Chapp, and Michael Wyatt

Teaching Assistant: Joe Teague

Office Hours: Monday, 2:15PM – 4PM, Min Kao Engineering 620

Course description

This course provides a practical introduction to big data analytics, blending theory (e.g., of clustering algorithms and techniques for dealing with noisy data) and practice (e.g., using Apache Spark, Jupyter Notebooks, and Github). Over the course of the semester, students will become familiar with modern data science methods, gain comfort with the tools of the trade, explore real-world data sets, and leverage the power of HPC resources to extract insights from data. Upon completing the course, students will have: used Jupyter notebooks to create reproducible, explanatory data science workflows; learned a modern MapReduce implementation, Apache Spark; implemented parallel clustering methods in Spark; studied strategies for overcoming the common imperfections in real-world datasets; and applied their new skills to extract insights from a high-dimensional medical dataset.

Course learning objectives

This course studies the high impact aspects of big data analytics and high performance computing from a practical perspective. Specifically, during the semester, students will learn how to use distributed programming models such as MapReduce and how to implement clustering and classification algorithms in MapReduce to enable scalable analysis of datasets across domains (e.g., medical, biological, and social sciences) on high-end clusters and supercomputers.

Targeted audience

The targeted audience of this course are engineering, computer science, and computational sciences students with interest in data analytics taking their first steps into the field. The course topics are relevant to these students because there is abundant interest in big data analytics on HPC systems. The main challenge is a lack of training courses that provide an easy-to-use interface to HPC systems and a software suite of scalable methods for data analytics. This course combines both aspects into a comprehensive semester-length course.

Student prerequisites and course requirements

Attendance is mandatory. Students must bring their own laptop to the lecture. Students must be proficient in Python. No book is required; reading material on Internet and in ACM or IEEE digital libraries may be used.

Class environment

This section presents methods of instruction and role of the students. At the beginning of the semester, each student will be given access to the tutorial GitHub repository with the tutorial materials (i.e., slides, assignments, and reading material). Each student will also be assigned a private repository for his/her own course material (i.e., solutions to problems discussed in class and project material). He/she will be provided with an account on a cluster running Apache Spark.

The course is structured in several modules. Each module starts with a short lecture in which the instructor presents a big data topic or algorithm and assigns a suite of simple, practical, hands-on exercises that can be tackled and solved with the presented methodology. Students will review a proposed strategy for a solution, extend with their own solutions, implement the solutions in the Jupyter Notebook (the entire hands-on exercise is annotated in the notebook and dedicated coding cells are provided for the solutions), and briefly discuss the findings with the instructor or course assistants. This format requires active participation and critical thinking skills as well as good programming skills in Python. During each module, students can work alone or in teams on the targeted, hands-on exercises; each student will be able to submit solutions to the assigned private repository before leaving the class and continue independently with any work that was not completed during the lecture during the rest of the week.

The initial module introduces the Jupyter notebook as an integrated development environment that is optimized to facilitate exploratory data analysis. Additionally, the version control system Git and its corresponding online community GitHub are introduced as integrated elements of the analytic workflow. In the subsequent modules, we introduce the MapReduce programming model through Apache Spark, a parallelization framework that implements in-memory, fault-tolerant MapReduce abstractions. We then cover fundamental clustering algorithms and strategies for coping with missing or malformed data. Finally, we integrate all of these elements in the final module that enables participants to perform exploratory data analyses of a nutritional/medical data set, via Jupyter, parallelized with Spark, on a cluster or the Cloud. The tight feedback loop between implementation and visualization afforded by the Jupyter notebook facilitates rapid comprehension of module contents. The Jupyter environment enables seamless visualization of results, thus building the students' intuition for how underlying algorithms and their parameters interact.

The final part of the semester is structured as a mini hackathon distributed across multiple lectures. During the mini hackathon, students will select a problem on big data and define a strategy to solve the problem supported by the tools learned during the first part of the semester. Example of problems will be provided. Students are encouraged to bring their own problems.

Detailed outline of the course

The course is structured in several modules:

Module I - Programming environment and infrastructure

Topics:

- Programming with Jupyter notebooks
- Version control with Git & GitHub

Practical hands-on exercise: Sequential text analysis and visualization using Jupyter notebooks

Module II - Introduction to MapReduce and Apache Spark

Topics:

- MapReduce as a programming model
- Map, Sort, Shuffle, Reduce workflow
- Partitioners and combiners
- Hadoop filesystem
- Overview of Apache Spark
- Resilient distributed datasets (RDDs)
- Parallel operations on RDDs
- In-memory computation in Spark

Practical hands-on exercise: Parallel text analysis (word, letter, and positional frequencies) using Spark

Module III - Clustering algorithms and their implementation in a MapReduce paradigm

Topics:

- Implementation of k-means from scratch in Spark
- Clustering with k-means and DBSCAN

Practical hands-on exercise: Application of parallel k-means from Apache Spark MLlib and parallel DBSCAN from tutorial-provided library on real-world datasets

Module IV - Cope with missing data and applicability of MapReduce clustering techniques on real datasets

Topics:

- Taxonomy of missing data scenarios
- Techniques for handling missing data
- Parameter tuning for clustering algorithms

Practical hands-on exercise: Handle missing data and tune DBSCAN parameters for high-quality clustering of real-world datasets

Module V - Optimizing Apache Spark for HPC and Cloud platforms

Topics:

- Launching Apache Spark on a batch system (e.g., SLURM) or on the Cloud (e.g., XSEDE Jetstream)
- Optimizing Spark's I/O for parallel file systems (e.g., Lustre)
- Managing RDD partitions

Practical hands-on exercise: Run previous exercises on HPC and Cloud resources with performance comparisons

Module VI – Working on real datasets

Topics:

- Working with medical/dietary/social-economic data from National Health and Nutrition Examination Survey (NHANES).
- Working with Medicaid and Medicare data from a US state
- Work with Soil Moisture data from the ESA-CCI Initiative
- Work on your own dataset – need approval from instructor

Practical hands-on exercise: Develop a set of key questions and use the tools learned in this course to answer the questions; present the outcome in a poster and a 2-page extended abstract

How a student can be successful in this course

Each content module is paired with hands-on exercises that reinforce the concepts introduced. Over the course of the semester, each student will complete multiple hands-on exercises that demonstrate the utility and expressivity of the MapReduce paradigm. Each student must complete and submit his/her own hands-on exercise in the assigned GitHub (private) repository.

To succeed, students shall:

1) Attend the lectures and actively participate in the class activities
2) Submit the solution(s) to the hands-on exercises in GitHub before the next lecture (next Monday before 8AM ET)
3) Work on a project (projects requires instructor approval) and implement original solution(s) to the project problem(s)
4) Submit a poster and a 2-page extended abstract describing the solution(s)
5) Present the posters in a poster session that will be scheduled the last week of the semester.

Failing to succeed in one or more of the 5 points above will result in failing the course. This course does not have a final exam.

Academic integrity

Students may discuss hands-on exercises and the project with peers. However, all the work students submit **must be their own** and all explanations must be in their own words. This means that students cannot write solutions of hands-on exercises in a group. Students cannot use the web to locate answers to any hands-on exercise. If students do not have time to complete an assignment, it is better they submit partial solutions than to get answers from someone else. **Cheating students will be prosecuted according to the university guidelines.** Students should get acquainted with their rights and responsibilities as explained in the Student Guide to University Policies (https://hilltopics.utk.edu/student-code-of-conduct/)

Emergency absences

If serious illnesses, family emergencies, or other crises occur during the term, one of the key things students must do is contact the dean of your college as soon as possible. This office can assist you in notifying faculty and in validating for your teachers what has happened. Such validation will be necessary for you to make up missed class work (https://dos.utk.edu/absence-notifications/)

Disability services

Any student who feels s/he may need an accommodation based on the impact of a disability should contact Student Disability Services in Dunford Hall, at 865-974-6087, or by video relay at 865-622-6566, to coordinate reasonable academic accommodations.