Syllabus Last Revised on 5/7/2023 by Dr. James G. Shanahan.

Course Overview

This 3 Unit course covers the underlying principles required to develop scalable machine learning for structured and unstructured data at the petabyte scale. Content is delivered via asynchronous video lectures, readings from academic textbooks, synchronous session discussions, live code demonstrations, and independent homework assignments.

The course is designed around three goals. By the end of the term, students will:

1. … learn to recognize and apply key concepts in parallel computation and MapReduce design.
2. ... design stateless parallelizable implementations of core machine learning algorithms from scratch.
3. ... gain hands-on experience using Apache Hadoop and Apache Spark to analyze large datasets, develop, and deploy machine learning pipelines at scale in the cloud.

Course Prerequisites

- Solve problems end-to-end with machine learning
- Ability to read, understand (basic nuances), and run Python code
- Adapt Python code (reconfigure) so that machine-learning algorithms can be applied and tuned to solve problems (case studies) using machine learning
- Enough knowledge of calculus to be able to differentiate simple functions.
- Enough knowledge of linear algebra to understand simple equations involving vectors and matrices.
- Enough knowledge of probability theory to understand what constitutes a probability density.

Content Description

Until recently, “big data” was very much the purview of database management and summary statistics systems such as Hadoop (HDFS and MapReduce) and was primarily under-leveraged by machine learning. This course builds on and goes beyond this collect-and-analyze phase of big data by focusing on how machine learning algorithms can be rewritten and sometimes extended to scale to work on petabytes of data, both structured and unstructured, to generate sophisticated models that can be used for real-time predictions. Predictive modeling at this scale can lead to huge boosts in performance (typically in the order of 10–20%) over small-scale models running on stand-alone computers that require one to significantly down-sample and, necessarily, simplify big data. Concretely, this course focuses on how the map-reduce design paradigm from parallel computing can be extended and more faithfully leveraged to tackle the somewhat “embarrassingly parallel” task of machine learning (many machine learning algorithms fit this mold).

The Apache Spark project and its many related subprojects exemplify the continued relevance of map-reduce style algorithm design. Apache Spark is an open-source cluster-computing framework. It has emerged as the next-generation big data processing engine, overtaking Hadoop MapReduce, which helped ignite the big data revolution. Spark maintains MapReduce’s linear scalability and fault tolerance but extends it in a few critical ways: it is much faster (100 times faster for specific applications); much more straightforward to program in due to its rich APIs in Python, Java, and Scala (and R) and its core data abstraction, the distributed data frame; and it goes far beyond batch applications to support a variety of compute-intensive tasks, including interactive queries, streaming, machine learning, and graph processing.

This course will provide an accessible introduction to MapReduce frameworks such as Hadoop and Spark and their potential to revolutionize academic and commercial data science practices through scale. Conceptually, the course includes two simultaneous components. The first covers fundamental concepts of MapReduce parallel computing via Hadoop and Spark. The second focuses on hands-on algorithmic design and development in parallel computing environments such as Spark; developing algorithms from scratch, such as decision-tree learning; graph-processing algorithms such as PageRank and shortest path; gradient descent algorithms such as linear regression and classification, linear support vector machines; matrix factorization, and unsupervised machine learning algorithms. Industrial applications and deployments of MapReduce parallel compute frameworks from various fields, including advertising, finance, healthcare, and search engines, help tie these components together. Examples and exercises will be available in Python Jupyter notebooks that leverage the Hadoop streaming and PySpark frameworks.

Machine Learning at Scale: schedule

Module	Section	Async Lecture	Sync Lab + Review quiz	Assignments	Reading Materials
1	**Intro and review**	Machine Learning at Scale introduction and course overview	Command-line-based map-reduce, Google Cloud		Read: **ISL chapter 1 and sections 2.1 & 2.2** Skim: **Adam Drake Blog Post** Optional: **Fortmann-Roe Essay**, **Clever Machine Blog Post** , **Inside Big Data Blog Post** Live Lab 1 SLIDES: See Module TAB
2		Introduction to Hadoop Streaming	WordCount, secondary sorts,	**HW01 MapReduce via**	Read: **DITP Chapter 1 & Chapter 2**

			Naive Bayes	**CMDLine due Midnight Sunday (at the end of week 2)**	Read: **IIR CH.13** Optional: **Michael Noll Hadoop MR Tutorial**
3		Map-Reduce Algorithm Design patterns and map-shuffle-reduce	Relative frequencies, custom partitioning, order inversion, inverted index		Read: **DITP sections 2.4 - 2.7 and 3.1-3.4** Read: **IIR sections 13.1 and 13.2** Read: HDG - Part II Chapter 7 - How MapReduce Works Skim: **Total Order Sort Guide**, **EECS Map Reduce Notes** OPTIONAL: **http://blog.ditullio.fr/category/hadoop-basics/** Slides: Lab Notebook is self-contained
4	**Spark**	Intro to Spark/Map-Reduce with RDDs (part 1)	Transformations, actions, RDDs, dataframes, datasets, broadcasting, closures	**HW02 (Hadoop NB) due Midnight Sunday.**	Read: **HP Spark chapter 2** Read: **Spark RDD Programming Guide** Skim: **Learning Spark ch 3 & 4** Skim: **DISCO Paper** Skim: **DocSim Paper** Additional Spark resources for weeks 4, and 5 • Holden Karau - Spark summit 2017 **https://www.youtube.com/watch?v=4xsBQYdHgn8&feature=youtu.be** [40 minutes] • Debugging Spark Holden Karau -Dec 2017 **https://www.youtube.com/watch?v=s5p15QT0Zj8&list=WL&index=7&t=0s** [45 minutes]
5		Intro to Spark/Map-Reduce with RDDs (part 2)	KMeans Clustering, pairs and stripes		• Watch: **Debugging Spark - Holden Karau - Dec 2017** [45 minutes] • Skim Chapter 16 in **IIR Book**
6	**Distributed Machine learning**	Distributed Supervised ML (part 1): Linear Regression	Mean squared error, gradient descent for LR, LR pipelines, house price prediction	**HW03 (Spark synonym) due midnight Sunday**	• Read: **ISL sections 3.1, 3.2 and 6.1, 6.2** • Skim: **UCI cs273a Loss Functions Lecture**
7		Distributed Supervised ML (part 2): Linear Classification, Maximum likelihood, logistic/softmax regression	LASSO, Ridge, L1 and L2 regularization, missing data, categorical data, feature selection.		• Read:**DDS chapter 5** • Read: **ISL chapter 4 & section 5.1** • Skim: **MMS chapter 11**

8		Big Data Systems and Pipelines	File formats, User-defined functions, joins, checkpoint	**HW04 (Gradient descent) due midnight Sunday**	• Skim: **HP Spark ch 3-4** • Read: **HP Spark ch 5-6** • Read: **Format Wars Post** • Optional: **SparkSession article** , **Sparkour recipe**
9	**Graphs**	Graph Algorithms at Scale (part 1): shortest path etc.	Final Project (FP) begins! Select your team members. For more details, please see the **Final Project module**. Adjacency lists, graph traversal, SSSP, Dijsktras, A*	**FP: Finalize Team members for the final project**	• Read: **DITP chapter 5** • Skim: **Cornell CS 312 Dijkstra's Lecture**
10		Graph Algorithms at Scale (part 2): pagerank and variants	Stochastic matrices, Eigenvectors, Markov Chains, teleportation, matrix multiplication at scale	FP Phase 1 is due Sunday midnight at the end of this week	
11	**Trees**	Non-gradient-ML: decision trees, random forests, ensembles	Trees and forests, GBDT.	FP Phase 2 is due Sunday midnight at the end of this week **HW5 due on Sunday**	• Skim TOC in **DATA MINING WITH DECISION TREES, Theory and Applications** • A most excellent introduction to Gradient Boosting: **How to explain gradient boosting**
12		Mid-Project review and presentations	EDA on the full dataset, Join on the full dataset, Baseline pipeline on the full dataset, cross-fold validation, Evaluation metrics	FP Phase 3 due	No async videos this week. Mid-Project reviews and presentations
13	**Recommenders**	Recommender systems, ALS, and Spark ML	Alternating Least Squares (Closed-form solution, GD solution), SVD, PCA, Field aware	FP Phase 4 is due Sunday midnight at the end of week 13 (this is a strict deadline)	• Read: **MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS** • Read: **DDS Chapter 8** • Skim: **Learning Spark ch 11**

			factorization machine		
14		Final Project Presentations	Final Project Presentations and course wrapup	FP Phase 5: Team presentation during week 14's live sessions.	No async videos this week. Final Project reports and presentations.

Readings

This course will use a combination of textbook chapters and some online readings. The books highlighted in blue are some of your instructor's favorites. Books marked with an asterisk are not available for free, and you will need to purchase them. The rest of the readings are available for free online.

Recommended Textbooks:

For access to O'Reilly Books please, please use the following link:

- **https://go.oreilly.com/university-of-california-berkeley**

Textbooks (the ones highlighted in yellow are core):

  - Lin, Jimmy, & Dyer, Chris. (2010). *Data-intensive text processing with MapReduce*. San Rafael, CA: Morgan & Claypool Publishers. (Free online)
  - **Jules S. Damji, Brooke Wenig, Tathagata Das, and Denny Lee (2020)** ***Learning Spark: Lightning-fast data analytics,*** **O'Reilly Publishers.**

- Karau, Warren. (2017). *High Performance Spark: Best Practices for Scaling and Optimizing Apache Spark.* Sebastopol, CA: O'Reilly Publishers.

- Hastie, Trevor, Tibshirani, Robert, & Friedman, Jerome. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Stanford, CA: Springer Science+Business Media. (Free online)
- Hastie, Trevor, Tibshirani, Robert, Witten, Daniela, & James, Gareth. (2014). *An Introduction to Statistical Learning: with Applications in R*. Stanford, CA: Springer Publishing Company. (Free online)
  - **https://lagunita.stanford.edu/courses/course-v1:ComputerScience+MMDS+SelfPaced/course/**

- *Ryza, Sandy, Laserson, Uri, Owen, Sean, & Wills, Josh. (2015). *Advanced analytics with Spark: Patterns for learning from data at scale.* Sebastopol, CA: O'Reilly Publishers.
- Leskovec Jure, Rajaraman Anand, Ullman Jeff, (2014). *Mining of Massive Datasets*, Cambridge University Press. Book available online at **http://www.mmds.org/**
  - ***https://lagunita.stanford.edu/courses/course-v1:ComputerScience+MMDS+SelfPaced/course/***
- *Doing Data Science* by O'Neil & Shutt, ISBN-13: 978-1449358655, ISBN-10: 1449358659
- *Hadoop: The Definitive Guide* by Tom White, ISBN: 978-1-491-90163-2; O'Reilly 2015
- Spark: The Definitive Guide: Big Data Processing Made Simple By Bill Chambers, Matei Zaharia, ISBN-13: 978-1491912218, ISBN-10: 1491912219, O'Reilly 2018

Weekly Materials:

Reading Assignment Abbreviations:

HDG = Hadoop: Definitive Guide (4th Edition) by Tom White

SDG = Spark: The Definitive Guide: Big Data Processing Made Simple By Bill Chambers, Matei Zaharia

DITP = Data Intensive Text Processing With Map Reduce by Lin & Dyer

IIR = Introduction to Information Retrieval by Manning, Raghavan, & Shutze

ISL = Introduction to Statistical Learning by Witten, James, Hastie, & Tibshirani

MMS = Modern Multivariate Statistical Techniques by Izenman

Learning Spark* = Learning Spark: High Performance Big Data Analysis by Karau, Konwinski, Wendell, and Zaharia

HP Spark* = High Performance Spark by Karau and Warren

DDS* = Doing Data Science by O'Neil & Shutt

*Starred books are ones you will need to purchase or borrow from the UCB library. All other reading materials are open source & linked here for your convenience.

To access the library, go to **http://oskicat.berkeley.edu/** and use your Berkeley login.

Things you need to know prior to starting 261

The assumption is that having completed the prerequisites for 261, you have competence in the below tooling. If you are not comfortable with the below tooling, **expect to at least double the amount of time that 261 will take you to complete**.

- Linear Algebra and Calculus
  - Derivatives
  - Dot Products
  - Matrix multiplication
  - Math notation
- Python Programming
  - General programming tasks such as string manipulation, list comprehensions, and control structures
  - Familiarity with Numpy and Pandas
- Bash skills
  - Control structures (if, while, etc.),
  - HEREDOCS
- Linux Skills
  - Determine running processes
  - Determine ports currently in use

Assignments and Materials

This is an upper-level graduate course. As such, we expect students to exhibit a high level of conscientiousness and initiative in their approach to preparing for class and completing assigned work. Each week you will have assigned video content as well as readings -- both should be completed before your live session*. In live sessions, we will typically review a short set of conceptual slides and/or break into groups to do a code demonstration. Sometimes, your live session instructor may ask you to review the first section of a demo notebook before you arrive in class. These code “demos” are designed to set you up for success on the homework. They offer a low-risk opportunity to build supporting skills-- the more actively you engage with them during class, the less time you will spend on homework.

Grading Policy

% of Final Grade Component	
50%	Homework Assignments (5 assignments, 10% each)
40%	Final Project
10%	Live session attendance and participation

The work of all students is reported in terms of the following grades:

Letter grades		Range	Interpretation
A+	<= 100%	to 98 %	Excellent
A	< 98%	to 92.5%	Excellent
A-	< 92.5%	to 90%	Very Good
B+	< 90%	to 82.5%	Good
B	< 82.5%	to 75%	Good
B-	< 75%	to 67.5%	Fair
C+	< 67.5%	to 57.5%	Below Expectations
C	< 57.5%	to 50%	Below Expectations
C-	< 50%	to 42.5%	Below Expectations
D+	< 42.5%	to 32.5%	Below Expectations
D	< 32.5%	to 25%	Below Expectations
F	< 25%	to 0%	Below Expectations

Letter grades	Range	Interpretation
I	Incomplete	Work incomplete due to circumstances beyond the student's control, but of passing quality; not included in grade-point computation after fall 1973.

Homework

Each independent homework assignment consists of a python notebook with coding and short response conceptual questions. We believe these assignments to be active learning experiences and hope you will approach them as an opportunity to develop your understanding and not just a source of a grade. Feedback from past students has consistently indicated that the challenge of these assignments makes the course valuable to them post-MIDS. For now, here's what you need to know:

- **Accessing and Submitting:** Homework assignments and submission details can be accessed via the Modules TAB. E.g., See Module 1 for HW1.
- **Time Commitment:** We expect a well-prepared student to spend 5-10 hours on each homework. Depending on your background, this time will vary. **If you do not have a background in software engineering or coding, mathematics, and/or statistics, be prepared to spend more time beyond the 10 hours.**
- **Late Policy:** As much as we are committed to serving a rigorous course, we also know you are hard-working professionals, parents, and partners. If you are running late on an assignment, please inform your instructor and the TAs before the deadline. If this happens too frequently, you will be penalized per the late submission guidelines associated with each assignment.
- **Partial Credit:** We grade each multi-part question holistically; always attempt as much as possible because we'd love to give you partial credit for partial understanding.

Grading Policy

Mastery-Based Grading - We seek evidence that you understand or misunderstand the learning objective/key concept. Every question will receive one of 4 scores: 100, 90, 50, or 0, as well as written feedback. A 90 or 100 for any answer that does demonstrate understanding of the Learning Target (including answers that have errors in code that are unrelated to the key concept). A 50 for any answer that fails to demonstrate an understanding of the Learning Target (including vague but plausible answers). 0 for a blank or nearly blank response.

For each question, the score you get is not an indication of a "percent of the sub-questions you got right" instead, you should think of it as a categorical indicator.

Each question had someone or two core concepts plus a lot of details. The goal is to avoid docking students multiple times for little details but still ensure that overall you get a clear message if you miss an important concept - so anything that shows confusion/error around the core learning target gets a 50. In contrast, any other error or combination of errors gets a 90.

This can seem harsh when you're on the 50% end of that equation, but we've found that over the course of a full assignment, it helps as much as it hurts. The goal of the grades on the homework is formative rather than summative.

Final Project

The final project is a group assignment that offers an opportunity to demonstrate mastery of the course materials and goals. We will assign groups and release a rubric in week 9. Your team will have four to five weeks to organize the workload, perform relevant EDA, implement the algorithms, and deliver a python notebook-based analysis report, including citations, experimental results, and a discussion of parallelization concepts that impacted the design choices you made. For more background and expectations on the final project, please see the **Final Project - Flight Delays**module. We offer a Project Leaderboard where you share your best pipeline results as part of your submissions for each project phase. We hope you enjoy this "coopetition" experience.

Logistics and Communication

Canvas is the source of ground truth for assignments, deadlines, and policies in this course. We will use Canvas Announcements for any core course announcements. We will rely on Slack for course discussions and or any questions you may have regarding course content. However, if you need to contact your section instructor for a non-content-related reason, please use slack/email.

Slack Channels

Slack serves a few purposes for this course. In the first week of classes, you will be expected to join three channels:

- main: one for general discussion
- infrastructure: one for troubleshooting infrastructure issues related
- announcements: and one for announcements (although we are planning to use Canvas Announcements for this also).

#data-sci-261-<year>-<semester>-announcements are for faculty to make announcements to all students. You can reply in a thread to an announcement to request clarification, but please do not post questions or comments on this channel.

To keep these channels helpful for everyone, please follow a few norms:

- Use threaded replies to help keep different lines of conversation easy to find.
- Respect that everyone comes from a different background -- share your silly questions freely and answer others' queries with good cheer.
- When asking a question, describe what you've already tried to resolve your question and reference any course materials you may be looking at -- this helps us make better suggestions faster.
- Commiserating can be a form of support (especially in a very time-consuming class) but watch out for the line where commiserating turns into complaining -- don't bring or put others down and use other channels (e.g., surveys or email) to share constructive criticism intended for instructors' ears.

ACTION: please join the following course slack channels for the current semester (these will remain open for the first few days of classes so you can add yourself directly):

1. datasci-261-<year>-<semester>-announcements
2. datasci-261-<year>-<semester>-main
3. datasci-261-<year>-<semester>-infrastructure

Where: where semester is fall|spring|summer and year is the current year, so for the summer semester in 2023 the slack channels are:

1. datasci-261-2023-spring-announcements
2. datasci-261-2023-spring-main
3. datasci-261-2023-spring-infrastructure

Requirements for in-class participation:

We believe in the importance of the social aspects of learning: between students and between students and instructors, and we recognize that knowledge-building is not solely occurring on an individual level but that it is built by social activity involving people and by members engaged in the activity. Participation and communication are key aspects of this course that are vital to the learning experiences of you and your classmates.

Therefore, we like to remind all students of the following requirements for live class sessions:

- Students are required to join live class sessions from a study environment with video turned on and with a headset for clear audio, without background movement or background noise, and with an internet connection suitable for video streaming.
- You are expected to engage in class discussions, breakout room discussions, and exercises and to be present and attentive to your and other teams' in-class presentations.
- Keep your microphone on mute when not talking to avoid background noise. Do your best to minimize distractions in the background video, and ensure that your camera is on while you are engaged in discussions.

In exceptional circumstances, if you cannot meet in a space with no background movement or if your connection is poor, make arrangements with your instructor (beforehand, if possible) to explain your situation. Sometimes connections and circumstances make turning off video the best option. Suppose this is a recurring issue in your study environment. In that case, you are responsible for finding a different environment allowing you to participate in classes without distracting your classmates.

Office Hours

Office hours will be held every week. Refer to the DigitalCampus and course repo for dates/times that each instructor will be available.

Infrastructure

Although learning how to set up a distributed environment for parallel computation is *not* a learning objective for this course, many of the concepts that are central to understanding parallel computation *do* require students to learn a little bit about infrastructure. We hope to keep infrastructure frustrations to a minimum by providing two consistent environments for students so that we can help you navigate reproducible errors related to your environment configuration:

- **Jupyter Labs PaaS on Google Cloud DataProc**. See **Module 1** for more details. Please set this up during week 1.
- **Databricks**: In addition to the docker container, we will be using DataBricks to complete some Spark assignments/projects.
- **Students that need assistance should email help@ischool.berkeley.edu.** This will open a ticket in their issue-tracking system. The folks at 'help' will follow up with you initially via email. If they can't help you resolve the issue that way, they'll set up a Zoom conference with you, have you share their screen, and walk you through the process.

In the past, students with the skill set to do so have set up their environment (e.g., using Google Cloud, AWS, or Databricks). While you are welcome to do so, you should know that Hadoop and Spark may have different out-of-the-box behavior depending on your system. We cannot support you directly unless you use the provided course materials.