Revised: 10/16/17

UC Davis
MSBA 422: Big Data
Winter 2018

Instructor Information:
Andy Barkett
Email: abarkett@gmail.com
Phone: 510.517.1581
Office hours by appointment in person before or after class, or by audio/video conference

## Course Overview

Students will become familiar with several common big data problem formulations such as indexing large sets of data, logging and parsing extremely high frequency events, making semantic sense of unstructured data, mining large data sets to create predictive models, and creating solvers for NP-complete problems. For each of these five problem types, the student will be able to evaluate and contrast relevant algorithmic approaches and system architectures. The student will also gain basic familiarity with a technology stack and toolchain appropriate to each algorithmic approach.

Topics and tools to be covered include, but are not limited to: Hadoop, various database systems, distributed computing, MapReduce, Mahout, Pig, Hive, and others.

Students will also be introduced to analyzing important tradeoffs between speed and reliability, performance and scalability, etc.

## Course Objectives

Students will…

1. Learn what 'big data' is and isn't. Understand what distributed computing means.
2. Become familiar with several common 'big data' computing challenges and situations.
3. Be able to identify a reference architecture appropriate to each computing challenge.
4. Be familiar with the basic toolchain required for each computing challenge.
5. Be able to assess pros and cons of different system designs and tool sets.
6. Be introduced to reference materials, additional courses, open source projects, and online tutorials that may help them solve big data challenges outside the classroom.

## Course Modules:

### Module 1 (Weekend 1-2) – Indexing, MapReduce, and Hadoop

Reading: Selections from *MapReduce Design Patterns* by Miner and Shook.

In-class assessment: None

Theory Lecture: Intro to indexing, inverse indices, origins of MapReduce, relationship between indexing and searching

Practical Lecture: Distributed MapReduce, Intro to Hadoop, HDFS, Hadoop operations and performance, Hadoop vs Spark, Solr

Group Assignment 1: Create an indexing Hadoop job appropriate for loading into solr.  Perform solr searches.

### Module 2 (Weekend 3-4) - Processing Logs at Scale

Reading: Overview of HDFS, Intro to Apache Flume, GCP Cloud Dataflow Docs (https://cloud.google.com/solutions/processing-logs-at-scale-using-dataflow)

In-class assessment: Quiz 1 on MapReduce/Hadoop

Theory Lecture: Simple log tailing, Detecting security event / intrusion pattern in logs, Issues with log rotation vs streaming

Practical lecture: Intro to logs, HDFS, logrotate, RegEx approach to log parsing, Why not use Splunk?, Cloud logging stacks.  Elasticsearch

Group Assignment 2: Re-format log data, create detectors for arbitrary security events / anomaly detection.

### Module 3 (Weekend 5-6) - Text Analytics

Reading: Selections from *Sentiment Analysis in Social Networks* by Pozzi, Fersini, Messina, and Liu, Selections from *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* by Geron.

In-class assessment: Quiz 2 on log parsing

Theory Lecture: Simple Classification vs NLP vs Text Mining/Analytics, NLP and Machine Learning

Practical Lecture: Scikit-Learn problem formulation, NLTK, POS tagging, Frequency distributions, Classifiers and Scikit-learn.  Comparison with Mahout.
Group Assignment 3: Sentiment analysis on corpus of data.  Assessment of accuracy.

### Module 4 (Weekend 7-8) - NP-complete problems and Genetic Algorithms

Reading: Wired article (https://www.wired.com/2013/01/traveling-salesman-problem/), Selections from Applegate and Bixby: *The Traveling Salesman Problem: A Computational Study*

In-class Assessment: Quiz 3 on Text Analysis, especially using Scikit-learn.

Theory Lecture: Examples of TSP in the wild. Naïve vs Dynamic solutions. Speed vs completeness tradeoffs, Genetic Algorithm Basics

Practical Lecture: Genetic algorithm single-machine approach. Distributing a genetic algorithm. General intro to serialization libraries/frameworks

Group Assignment 4: Create simple genetic algorithm framework for TSP. Choose distributed computing 'engine.' Apply Protobuf/Thrift/Avro. Call from different language.

## Module 5 (Weekend 9-10) - Creating predictive models from large datasets

Reading: Selections from Lewis – *Moneyball,* Selections from Miller – *Modeling Techniques in Predictive Analytics with Python and R: A Guide to Data Science*

In-class Assessment: Quiz 4 on speed vs completeness tradeoffs

Theory Lecture: Baseball statistics, Prediction vs Description vs Classification vs Clustering, Pig vs Hive. Measuring Predictive Model Success

Practical Lecture: Develop a simple predictive model. Explore Pig/Hive implementation. Finding a model vs enabling analysts to find models.

Group Assignment 5: Use Pig to implement two predictive models. Implement Hive to allow report creation by non-programmer.

## Groups

Groups will be chosen by students, of size 4-5, and will be chosen before or during the first class session. Groups are expected to distribute work amongst the group.

## Final Exam

There will be no final exam.

## Assignments and Grading Policy

This is a letter-graded, 3-unit course. Attendance is mandatory. There will be no make-up assignments or quizzes.

All assignments are due before the start of the next weekend's classes. All assignments are group assignments. All quizzes are individual quizzes, closed-notes, closed-computer.

Grades will be on a curve, and the components of the grade will be as follows:

Total grade – 100%

- Group Assignments (5) – 40%, 8% each
- Individual Quizzes (4) – 40%, 10% each
- Class Participation  - 10%
- Bonus Project (optional) – 10% (extra credit)