IE531 (AD, B): Algorithms for Data Analytics
218 Ceramics Building
Spring, 2022

Instructor: Prof. R.S. Sreenivas
201E Transportation Building (Primary Office)
e-mail: rsree@illinois.edu
Tue-Thu, 3:30PM-4:50PM,
Teaching Assistants: Ping (Evelyn) Ma (pingm@illinois.edu) and
Shreekant Sudheer Gokhale (gokhale6@illinois.edu)
TA Office hours: TBA.

Course Description: We will cover the foundations of Data Science in this course. The text requires familiarity with several diverse algorithmic- and mathematical-concepts. I will go over the foundational-material (Linear Algebra, Computing/Programming, Probability and Statistics, Optimization) using my notes. Following this, we will use the text for Low-Rank approximations, Markov Models, Machine Learning, Streaming Data models, Clustering Algorithms, and Hidden Markov Models.

All of my lessons will be in the form of Jupyter Notebooks for this course. I am working under the impression that this being a 500-level course, you do not need any hand-holding when it comes to Python Programming. You can figure most of the material out from my Notebooks. I will use C++ in lectures when I need speed (to illustrate a concept in the classroom environment), but you do not need to be familiar with it.

Primary Text: A. Blum, J. Hopcroft and R. Kannan, *Foundations of Data Science* (FDS), January 2018. You can download the latest-version of this text by clicking this link.

1 Tentative Syllabus

"You must fill your heads with wisdom before you can break boards with it."
–– Karate instructor on "The Simpsons".

Lectures (Tentative; exact-coverage depends on the background of the class!)

1. **Linear Algebra, Optimization & Computation, Probability Review (1 week)**
   Reading List: Lesson 1 of my notes.

   (a) Master-Theorem of Recursion, (Python code for) the *Repeated Squaring* Algorithm, (Python code for) the recursive $O(n)$ *Median-of-Medians* Algorithm, (Python Code for) the *Randomized* Median-of-Medians Algorithm; Review of Linear Algebra; (Python code for) Solving systems of Linear Equations (i.e. $\mathbf{Ax} = \mathbf{b}$); (Python Code for)

Mixed-Integer Linear Programs, LP-Duality, (Python code for) *Semi-Definite Programming*, (Python code for) *Quadratic Programming*; (Python Code for) Pseudo-Random Number Generators and RV generation, Review of Markov-Chains, Spectral Properties of Markov-Chains.

2. **Best-Fit Procedures (2 weeks)**
   Reading List: Chapter 3, FDS.

   (a) *Singular Value Decomposition* (SVD); Lower-Rank Approximations; (Python code for) SVD-Algorithms and their applications – Centering Data; *Principal Component Analysis* (PCA); *Pseudo-Inverse* of a Matrix; Relationship between SVD and SDP; Other "Best-Fit Procedures" – Fourier Transforms, (Python code for) 1-D and 2-D Fast Fourier Transforms; etc.

3. **Geometry of High-Dimensional Space (2 weeks)**
   Reading List: Chapter 2, FDS.

   (a) Tail-Bounds (*Markov-*, *Chebyshev*, and *Chernoff* Bounds); Geometry of High-Dimensional Data; Generating Gaussians in High-Dimensions; *Johnson-Lindenstrauss Lemma* and it applications (Python code); Achlioptas' Distribution (Python Code);

4. **Markov Chains (2 weeks)**
   Reading List: Chapter 4, FDS.

   (a) *Markov Chains*; (Python code for) Finding the *Stationary Distribution* of a Markov Chain; *Repeated-Squaring Method*; (Python code) for *Markov-Chain Monte Carlo* (MCMC) methods – the *Metropolis-Hasting Algorithm* and *Gibbs Sampling*; *Finite Markov Decision Processes* (MDPs); *Dynamic Programming* (DP);

5. **Machine Learning (2 weeks)**
   Reading List: Chapter 5, FDS.

   (a) The *Perceptron Algorithm*; Kernel Functions; *Support-Vector Machines* (SVM); *Long-Short-Term-Memory* (LSTM) Machines; TensorFlow Illustration;

6. **Streaming, Sketching and Sampling (2 weeks)**
   Reading List: Chapter 6, FDS.

   (a) Data-Stream Management; Turnstile Models; Stream Sampling; *Bloom Filters*; Counting distinct- frequent- elements in a stream; *Flajolet-Martin Algorithm*; Moment-estimation; *Alon-Matias-Szegedy Algorithm* for 2nd moments; *Matrix Sketching*;

7. **Clustering (1 week)**
   Reading List: Chapter 7, FDS.

(a) *Spectral Clustering*; *K-means* Clustering; *K-Center* Clustering; Kernel Methods for Clustering; Social Networks and Spectral Clustering.

2 Grade Composition

- ≈ 7 Programming Assignments (30%).
- ≈ 5 Homework Assignments (30%).
- Mid-Term Exam (20%).
- End-Term Exam (20%).

3 General Instructions

I plan to have ≈ 7 programming assignments for this course. Since the pace of the course will be dictated by the class needs/skills, the exact number of these assignments might vary. The contribution to your final grade will be 30% independent of the number. You will submit your Jupyter Notebooks on Canvas to be tested/verified by the TA.

Please do not ask for extensions in the 11-*th* hour. You will get full-credit if your submitted code works when compiled and run on a data-set of our choice. We will revert back to you if there is an error/problem with your submission. You then have three days to turn-in a corrected version, at the loss of 20 points. This process is repeated at most two times (i.e. inclusive of your first attempt, you have three chances at getting the programming assignment right).

We will have ≈ 5 Homework Assignments, their contribution to the final grade will be 30% independent of the number. You will turn in a PDF-version of your homework by their due dates.

The written Mid-Term and End-Term exams will have short-answers that test your familiarity with the foundations/theory. The students who have registered for Section AD (i.e. the on-campus students) will have their mid-term exam tentatively on March 30, 2023. The students in Section ONL will work with the *Office of Online and Professional Engineering Programs*. There is usually a prescribed time for the final exam schedule that can be found at (this link). Based on my interpretation of the information at this link, the final exam for this course will be held

8:00-11:00 a.m., Wednesday, May 10

It would not hurt to double-check this site (and if things change, I will update this document accordingly). The exam-schedule for students in the online version of this class will be determined by the *Office of Online and Professional Engineering Programs* – I will update you via e-mail as this information is made available. I intend to use the $\pm$-grading system.

Homework 1	3 February 2023
Programming Assignment 1	10 February 2023
Homework 2	17 February 2023
Programming Assignment 2	24 February 2023
Homework 3	3 March 2023
Programming Assignment 3	10 March 2023
Spring Break	11-19 March 2023
Homework 4	24 March 2023
Mid-Term Exam	30 March 2023
Homework 5	7 April 2027
Programming Assignment 4	14 April 2023
Programming Assignment 5	21 April 2023
Programming Assignment 6	28 April 2023
Programming Assignment 7	5 May 2023
End-Term Exam	8:00AM-11:00AM, 8 May 2023

Table 1: Tentative Due Dates for Homework, Programming Assignments, and Exams.

Everything about this course can be found on the University of Illinois' Canvas Website. It is your responsibility to check the above URL regularly for updates/due-date-announcements as the course progresses.

I am happy to correspond via e-mail if you are having trouble with the algorithmic-aspects of the course material. For the nitty-gritty details of the programming assignments, I ask that you approach the TA first, if you need additional assistance, you can definitely contact me. We have on-campus and on-line students in this class. The TAs office hours are posted above. If you need to meet with me, send me an e-mail, and we will figure out a time that works for both of us. The tentative schedule for the various submissions of graded-work is shown in table 1.

Good luck and I am looking forward to seeing you do well in this course!