IE531: Algorithms for Data Analytics
Spring, 2019

Instructor: Prof. R.S. Sreenivas
201E Transportation Building (Primary Office)
155 Coordinated Science Laboratory (Secondary Office)
e-mail: rsree@illinois.edu
Tue-Thu, 3:30-4:50PM, 101 Transportation Building Teaching
Assistant: Arun Raman (raman12@illinois.edu)
Office hours: Tuesdays 2:00 - 3:20 PM, CSL 157

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.

We will switch (effortlessly) between Python, MATLAB/SIMULINK and C++ when we discuss the various algorithmic aspects in this course. One of the objectives of the course is get you comfortable with programming. I will provide sample code to get you started on various topics. It is best if you supplemented these with your own as you progress through the course. If you are a novice at programming, you should be more diligent (than others). We will support two C/C++ compilers – *Microsoft Visual Studio* (Windows users) and *XCode* (Mac users). As a UIUC-student you can download a free-version of *Visual Studio* from by clicking this link. If you are a Mac-user, you can get *XCode* for free from the App-Store.

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 Review (1 week)**
   Reading List: Lesson 1 of my notes.

   (a) Review of Linear Algebra - *Rank*, *Determinant*, *Eigenvalues*, etc; (C++ code for) Solving systems of Linear Equations (i.e. $\mathbf{Ax} = \mathbf{b}$); (C++ Code for) Mixed-Integer Linear Programs; (C++ code for)

Pseudo-Random Number Generators and RV generation; *Randomized Algorithms* case-study – (C++ code for) the recursive $O(n)$ *Median-of-Medians* Algorithm.

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

   (a) *Singular Value Decomposition* (SVD); Lower-Rank Approximations; (C++ code for) SVD-Algorithms and their applications – Centering Data; *Principal Component Analysis* (PCA); Spherical Gaussian Clustering via SVD; Ranking Documents and Web Pages; Discrete Optimization via SVD.

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

   (a) *Markov Chains*; (C++ code for) Finding the *Stationary Distribution* of a Markov Chain; *Repeated-Squaring Method*; (C++ code) for *Markov-Chain Monte Carlo* (MCMC) methods – the *Metropolis-Hasting Algorithm* and *Gibbs Sampling*; The Web as a Markov Chain;

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

   (a) The *Perceptron Algorithm*; Kernel Functions; *Support-Vector Machines* (SVM); *Vapnik-Chervonenkis Dimension* (VC-Dimension); Boosting; Stochastic Gradient Descent Algorithm (and its details);

5. **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*;

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

   (a) Geometry of High-Dimensional Data; Generating Gaussians in High-Dimensions; *Johnson-Lindenstrauss Lemma* and it applications (C++ code); (C++ code for) Fitting a Spherical Gaussian to Data;

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.

8. **Topic Models, Hidden Markov Models (1 week)**
   Reading List: Chapter 9, FDS.

(a) *Dirichlet Process Models*; Illustration - Topic Analysis via *Linear Dirichlet Models*; *Hidden Markov Models*; *Belief Networks*; *Markov Random Fields.*

2 Grade Composition

- ≈ 5 Programming Assignments (30%).
- ≈ 4 Homework Assignments (30%).
- Mid-Term Exam (20%).
- End-Term Exam (20%).

3 General Instructions

I plan to have ≈ 5 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 turn-in electronic versions of your code to be tested/verified by the TA or me.

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 ≈ 4 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 mid-term exam will be held during class-hours at a date that is to be announced. I expect it to be during the first week of March. There is usually a prescribed time for the final exam schedule that can be found at (this link). I will update this document as soon as this information is available. 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 ±-grading system.

Everything about this course can be found on the University of Illinois' Compass 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. I will post my office hours after we have had a chance to converse about it. The TA will do the same, as well. This would be done after the first-week of class.

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