CSC 592

Algorithms for Big Data

Prof. Noah M. Daniels

noah_daniels@uri.edu
http://www.cs.uri.edu/~ndaniels/
Tyler 250

Changes to course policies or schedule may occur in response to unforeseen circumstances. I will notify the class of any changes immediately.

Course Description: In this project-oriented course, we will explore algorithms for large data sets, including data sources that are growing faster than Moore's Law. We will cover mathematically rigorous models for developing efficient algorithms for large data sets. Topics will include sketching and streaming algorithms, randomized polynomial-time approximation schemes and their error bounds, compressive acceleration, and other sublinear-time and constant-memory algorithms. We will read at least one research paper every week, and every member of the class will be expected to present one of these research papers to the rest of the class. There will be a significant final project involving a real "big data" opportunity, which may involve an application to astronomy, oceanography, bioinformatics, security, or another field of your choosing. Alternatively, this project could involve the development of a novel algorithm, or a useful implementation of an algorithm that does not yet have such an implementation publicly available. **Prerequisite(s):** CSC 440 or equivalent experience, or permission of the instructor.

Texts:

All required readings will be provided as PDFs by the instructor. We will use perusall.com as a "social reading" platform, at no cost (PDFs of reading assignments will be posted there). Your class participation grade will depend, in part, on discussion, questions, and comments on the perusall platform.

Course Objectives:

At the completion of this course, students will be able to:

1. Analyze algorithms with sublinear or otherwise unusual (e.g. entropy-based) running times
2. Apply the laws of probability to determine error bounds for probabilistic algorithms
3. Read primary sources (research papers) and distill the contributions of a paper.
4. Use and implement some of the algorithms discussed in class
5. Identify a promising research problem and choose appropriate algorithms for analysis

Grading Rubric:

Class Participation	20%
Final Project (Proposal)	10%
Final Project (Paper)	60%
Final Project (Presentation)	10%

Letter Grade Distribution:

>= 93.00	A	80.00 - 82.99	B-	67.00 - 69.99	D+
90.00 - 92.99	A-	77.00 - 79.99	C+	63.00 - 66.99	D
87.00 - 89.99	B+	73.00 - 76.99	C	60.00 - 62.99	D-
83.00 - 86.99	B	70.00 - 72.99	C-	<= 59.99	F

Tentative Course Outline:
The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. The reading assignment listed for a given class period should be completed by that class period (hence, no reading due for the first lecture).
The final project has several components:

- a *proposal* which will be due by class time during week 6 of the course; this should be a 1-2 page (at most) description of what data sets, analysis, software development, or algorithm design you will pursue.
- a *draft paper* which will be due by class time during week 10 of the course. The draft will receive feedback in the style of peer reviews but will not count towards the grade. This draft should include thorough discussion of the project being pursued, but we recognize that some results and experimentation may be as yet incomplete.
- a *final presentation* during the last 2 weeks of class. Specifically, the last two class sessions will be devoted to in-class presentations, and students will be asked to sign up for time slots (approximately 20 minutes per project; remember that these can be group projects) in the prior week.
- a *final project paper* due at end of term in lieu of a final exam. This should generally not exceed 15 pages, and should follow the form of a research paper in computer science, aimed at a venue such as IEEE Big Data or Journal of Data Science. We encourage the paper to be written using LaTeX.

The final project can be done as a small group project (2-4) students or an individual project. Students will be expected to meet with the instructor outside of class time for feedback and coaching on their projects. The final project can take many forms: it could be the development of a novel algorithm, coupled with a formal proof of computational complexity or a proof-of-concept implementation; it could be the application of existing algorithms to a dataset of particular interest to the student; it could be a refined implementation of an existing algorithm that does not currently have good software support. In any case, some theoretical or implementation work must accompany the term paper, and it is encouraged to post software artifacts to online repositories such as GitHub.

Course Policies:

- **Attendance**
  - You are expected to attend class. I do not take attendance *per se*, but participation in class discussion will form a component of your grade.
- **Grades**
  - Grades in the **C** range represent performance that **meets expectations**; Grades in the **B** range represent performance that is **substantially better** than the expectations; Grades in the **A** range represent work that is **excellent**.
  - **Reading Assignments** will typically be peer-reviewed papers that have been published in the field. These will be made available to you at no cost. Each student in the class will be required to present one paper during the semester; this presentation will form the primary basis for the class participation grade.
  - **Term Projects** are expected to be ongoing work during the second half of the semester, and may be conducted individually or in small groups. A project proposal will be due partway through the semester; while this proposal will not receive a grade itself, feedback on this proposal will help you to successfully complete your term project. A draft of this proposal will be due towards the end of the semester, and feedback on this proposal will be conducted like academic peer review, encouraging you to improve your final paper. The final paper will be graded, and due on the date scheduled by the university for final exams. An in-class presentation on the final project will occur during the last two class sessions.
- **Cheating**
  - All work submitted must be your own work and that of your partners.
  - It is acceptable to use existing software libraries for the programming component of a project. However, such outside sources must be acknowledged in your paper as well as in any source code you provide.
  - **Any violation of these rules may result in a grade of 0 on the assignment. In addition, you may be reported to the Dean and the Office of Student Life.** See the University Manual for more information about the potential consequences of cheating `https://web.uri.edu/manual/chapter-8/chapter-8-2/`.
- **Assignments**
  - Reading assignments are required; you are expected to complete them on time and participate in class discussion (both online and in person). By all means, you are encouraged to ask questions when you do not understand something from a reading assignment.
- **Exams**
  - There will be no exams.

Disability, Access, and Inclusion Services for Students Statement:

Your access in this course is important. Please send me your Disability, Access, and Inclusion (DAI) accommodation letter early in the semester so that we have adequate time to discuss

and arrange your approved academic accommodations. If you have not yet established services through DAI, please contact them to engage in a confidential conversation about the process for requesting reasonable accommodations in the classroom. DAI can be reached by calling: 401-874-2098, visiting: web.uri.edu/disability, or emailing: dai@etal.uri.edu. We are available to meet with students enrolled in Kingston as well as Providence courses.

Anti-Bias Syllabus Statement:

We respect the rights and dignity of each individual and group. We reject prejudice and intolerance, and we work to understand differences. We believe that equity and inclusion are critical components for campus community members to thrive. If you are a target or a witness of a bias incident, you are encouraged to submit a report to the URI Bias Response Team at www.uri.edu/brt. There you will also find people and resources to help.

Academic Honesty Policy:

All submitted work must be your own. If you consult other sources (class readings, articles or books from the library, articles available through internet databases, or websites) these MUST be properly documented, or you will be charged with plagiarism and will receive an F for the paper. In some cases, this may result in a failure of the course as well. In addition, the charge of academic dishonesty will go on your record in the Office of Student Life. If you have any doubt about what constitutes plagiarism, visit the following websites: the URI Student Handbook, and Sections 8.27.10 – 8.27.21 of the University Manual (web.uri.edu/manual/).

Attendance

Students are expected to attend class and classroom activities. Occasionally, students may miss class activities due to illness, severe weather, or sanctioned University events. If ill, students should not attend class and should seek medical attention especially if they have a communicable disease such as influenza (flu). Students should not attend class when the University announces classes are cancelled due to severe weather. Also, it is the policy of the University of Rhode Island to accord students, on an individual basis, the opportunity to observe their traditional religious holidays. Students desiring to observe a holiday of special importance must inform each instructor and discuss options for missed classes or examinations. See Sections 8.51.11 – 8.51.14 of the University Manual for policy regarding make-up of missed class or examinations.

Table 1: Weekly Course Schedule (all deliverables are due at the beginning of class in the week indicated)

Week	**Content**
1 Reading	Moore's Law & Sources of Big Data: motivation and logistics. Probability refresher. None
2 Reading	Sketching & Streaming: very small-space data structures. Bloom filters, vEB trees. Donoho, "High Dimensional Data Analysis: The Curses and Blessings of Dimensionality"
3 Reading	Dimensionality Reduction: techniques for reducing data dimension while preserving geometric structure Yu & Weber, "HyperMinHash: MinHash in LogLog Space"; Yiu, "Understanding PCA (Principal Components Analysis)"
4 Reading	Manifold Learning: more dimension reduction Priebe & Cowen, "Approximate Distance Clustering"; Cannon, et al., "Approximate Distance Classification"; Cowen & Priebe, "Randomized Nonlinear Projections Uncover High-Dimensional Structure"
5 Reading	Dimension Reduction for Visualization (t-SNE and UMAP) van der Maaten & Hinton, "Visualizing Data Using t-SNE"; McInnes, et al., "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction"
6 Reading	Locality-Sensitive Hashing: even more dimension reduction. **Term Project Proposals Due by start of class** Gionis, et al., "Similarity Search in High Dimensions via Hashing"; Luo, et al., "Low-density locality-sensitive hashing boosts metagenomic binning"
7 Reading	Compressive Acceleration: compression for speed, algorithms that scale with entropy Daniels, et al., "Compressive Genomics for Protein Databases"; Yu, Daniels, et al. "Entropy-Scaling Search of Massive Biological Data"; Ishaq, et al., "Clustered Hierarchical Entropy-Scaling Search of Astronomical and Biological Data"
8 Reading	More on Compression and Information Theory Berger, et al., "Levenshtein Distance, Sequence Comparison, and Biological Database Search"; Burrows & Wheeler, "A Block-sorting Lossless Data Compression Algorithm"
9 Reading	Succinct data structures and graph algorithms Karger, et al., "Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web"; Barabasi & Albert, "Emergence and Scaling in Random Networks"; Cao, et al., "Going the Distance for Protein Function Prediction: A New Distance Metric for Protein Interaction Networks"
10 Reading	Computational Topology Term Project Drafts Due Moitra, et al., "Cluster-based Data Reduction for Persistent Homology"; Carlsson, "Topology and Data"
11 Reading	Anomaly Detection in Big Data Settings. Ishaq, et al., "Clustered Hierarchical Anomaly and Outlier Detection Algorithms"
12	**Project presentations**
13	**Project presentations**
Finals	**Final Papers Due on date & time specified for final exams**