ELEG/FSAN 817 Large Scale Machine Learning

Credits: 3
Fall 2021
Meeting Times: TBD
Location: TBD

Instructor: Austin J. Brockmeier, Ph.D. (he/him/his)
Assistant Professor
Department of Electrical and Computer Engineering
Department of Computer and Information Sciences
Resident Faculty, Data Science Institute

Email: ajbrock@udel.edu
Office hours: TBD

Description

Large-scale machine learning is an introduction to the analysis and processing of massive high-dimensional data. Massive data sets generally involve growth not only in the number of individuals represented but also in the number of descriptive parameters of the individuals, leading to exponential growth in the number of hypotheses considered. New approaches to address these problems exploit sparsity prior concepts from optimization theory, signal processing, statistics, and machine learning.

Prerequisites

Previous machine learning or pattern recognition course such as FSAN/ELEG 815 or equivalent (CISC684); Previous experience with a programming language suitable for data science. This course is meant to build on previous experience in machine learning and data science methodology and theory.

Topics

"I learned very early the difference between knowing the name of something and knowing something." — Richard Feynman

- **Understanding high-dimensional data**
  - Feature selection and regularization: shrinkage, sparsity, LASSO
  - Distances between data points for different norms in high-dimensional spaces
  - Independent versus correlated features: information theory for redundancy reduction
  - Representations and algorithms: Johnson-Lindenstrauss lemma, hashing trick, bags of features
- **Experiment design, evaluation metrics, cross-validation, and model selection**
  - Randomization, bag of little bootstraps, consistency, stability
- **Combinatorial hypothesis and optimization, uncertainty in high dimensional space**
  - Convex optimization, polytopes, Frank-Wolfe algorithm, gradient boosting, random forests
  - Support vector machines and the kernel trick, Nyström approximation, Gaussian processes

    - Hyper-parameter optimization, Bayesian optimization
- **Generalized machine learning paradigms and structured data**
    - Multilabel, multiclass, multiview, multitask, and multi-instance learning
    - Group LASSO, matrix-variate and tensor-variate features and responses
    - Semi-supervised learning, active learning
- **Uncertainty in high dimensional space**
    - Gaussian processes, hyperparameter exploration, Bayesian optimization, reinforcement learning
- **Approximating and completing large matrices**
    - Matrix sketching, low-rank decompositions, CUR decomposition, matrix completion, tensors
- **Neural networks**
    - Auto-encoders, generative adversarial networks

Learning Outcomes

- At the completion of this course an engaged student will be able to
    a. mathematically formulate data science and machine learning tasks (problem framing), with clearly stated objective, assumptions, constraints, and the mathematical characteristics of input and output.
    b. analyze the global and local curvature of objective functions and constraint sets.
    c. choose appropriately any regularization, a model selection criterion, and a valid experimental design (including hyperparameter selection) to ensure generalizability and reproducibility.
    d. summarize and critique descriptions of machine learning methods, experimental design, result discussions (including statistical tests) in a peer review setting with constructive feedback.
    e. select and justify appropriate algorithms, data structures, and relaxations for large-scale problems such that computation can be successfully executed with an understanding of the trade-off between approximation and complexity.
    f. list the challenges, errors, and uncertainties inherent with large-scale data.
    g. understand techniques for processing or representing large-scale data including sparse matrices, low-rank matrices, tensor formulations, tree structures, and neural networks with weight sharing, convolutional, and recurrent architectures.
    h. explain and justify randomization and sampling techniques useful to large-scale data.
    i. compare and contrast different optimization techniques and specific algorithms, such as constraint relaxation, greedy algorithms, and distributed optimization.

Course Elements and Assessment

“For the things we have to learn before we can do them, we learn by doing them.” —Aristotle

- Weekly assigned readings of journal articles and book chapters with summaries (10%).
- Homework assignments, including algorithms implementation and simulation (20%).
- Quizzes (5% optional)
- Midterm examination (in class: 15% or 10% with quiz scores; take-home 10%).
- Project (35%).
- Peer feedback of project reports (10%).

- The project will be broken into a series of assessment due in sequence (Formulate, analyze, design/choose, analyze, implement, experiment, analyze, discuss)
  - Initial abstract (2%)
  - Revised abstract (3%)
  - Rebuttal of peer review, clarified methodology, experimental design, and plan for analysis and predictions of results (5%)
  - Mathematical formulation, experimental design, and initial results and plan for remainder (10%)
  - Final report (15%)
- Paper summaries for assigned readings (Comprehend, summarize, critique, extract insight, catalog resources) consisting of
  - a short description 4–8 sentences in your own words,
  - a discussion of insights from reading the paper (2–5 sentences)
  - a list of any resources (algorithms, data sets, experimental designs, proof techniques, statistical tests, theorems, visualizations) you would find useful for your own or future research,
  - and a perspective on how it can be applied to your own research or project.
- Each student will provide peer feedback on three other students projects in the style of conference peer review (10%)
  - Peer feedback project abstract (2%)
  - Peer feedback on formulation, experimental design, and results plan (3%)
  - Peer feedback on final report (5%)

Important Dates

- 10/13 Midterm examination
- 10/25 Project abstract due
- 11/2 Revised project abstract due
- 11/8 Peer feedback on project abstract due
- 11/17 Project formulation, experimental design, and design of presentation of results due
- 11/29 Peer feedback on above due
- 12/6 Project final description due
- 12/13 Peer feedback on final project due

Learning Resources

Canvas

Weekly readings, homework, project details will be posted on Canvas

Software

Projects and homework will require access to a computer programming environment suitable for data science. Suggested languages: R, python, MATLAB/ Octave, Julia