EECS 375/475 and Data_Science 423: Machine Learning: Foundations, Applications, and Algorithms

Spring 2020: MWF 10-11a
Prof. A. K. Katsaggelos; aggk@eecs.northwestern.edu; M468
Virtual office hours: TBA
TA: Hao Zhou and Israel Ridgley
Virtual office hours: TBA

Prerequisites

A basic understanding of Linear Algebra and Vector Calculus (e.g., students should be able to easily compute gradients/Hessians of a multivariate function), as well as basic understanding of the Python or MATLAB/OCTAVE programming environments.

- **Course is crosslisted with Data_Sci 423**

REQUIRED TEXT: J. Watt, R. Borhani, and A. K. Katsaggelos, *Machine Learning Refined: Foundations, Algorithms, and Applications*, Cambridge University Press, 2nd edition, 2020.

COURSE OUTLINE:

1. Introduction
    1. What kinds of things can you build with machine learning tools?
    2. How does machine learning work? The 5 minute elevator pitch edition
    3. Predictive models – our basic building blocks
    4. Feature design and learning – what makes things distinct?
    5. Numerical optimization – the workhorse of machine learning
2. Fundamentals of numerical optimization
    1. Calculus defined optimality
    2. Using calculus to build useful algorithms
    3. Gradient descent
    4. Newton's method
3. Regression
    1. Linear regression - applications in climate science, feature selection, compression, neuroscience, and marketing
    2. Knowledge-driven feature design for regression
    3. Nonlinear regression
    4. The L-2 regularizer
4. Classification
    1. The perceptron
    2. Logistic regression/Support Vector Machines
    3. Multiclass classification
    4. Knowledge driven feature design for classification– examples from computer vision (object/face detection and recognition), text mining, and speech recognition

5. Probabilistic Formulation
    1. Regression
        1. Bayesian linear regression
        2. Non-linear regression
        3. Sparse linear regression
    2. Classification
        1. Bayesian logistic regression
        2. Non-linear logistic regression
        3. Boosting
6. Feature learning
    1. Function approximation and bases of features
    2. Feed-forward neural network bases, deep learning, and kernels
    3. Cross-validation
7. Special topics
    1. Step length determination for gradient methods
    2. Advanced gradient descent schemes: stochastic gradient descent and momentum
    3. Dimension reduction: K-means clustering and Principal Component Analysis

PROBLEM SETS:

Weekly pencil-and-paper and computer problems will be assigned and graded. A few small computer projects will also be assigned. Homeworks will be typically assigned on Fridays and will be due on Fridays a week later at midnight. **No late homeworks will be accepted.**

PROJECT:

A literature survey or computer project is required. Please submit a short description (a few paragraphs, less than a page) of the course project you would like to work on by **Monday April 4**, midnight. You are encouraged to work with other students in the class. The maximum size of a group is 4 people.

Project Scope:

Your choice of project is up to you (and your team). All we ask is that it be related to the course material, and that you pick something that interest you. If you are having trouble deciding on a project idea talk with one of us (TA and me).

For example, you could pursue:

• a machine learning curiosity (e.g., how do I put together a robust face detection pipeline?), and prototyping a simple version of the curiosity (e.g., putting together a simple face detection pipeline)

• pet projects / apps with a substantial machine learning component

• reading a selection of papers to better familiarize yourself with a specific topic in machine learning and presenting a comprehensive summary of the findings, with description of open problems and possible solution directions

Deliverables:

Your final deliverables for the course project will be

1) A short Python Jupyter notebook (less than 5 pages long) explaining and illustrating your project topic, including any fundamental Python code.

2) A short Youtube video (3 - 5 minutes in length maximum) summarizing what you / your team pursued, any trials and setbacks along the way, and conclusions. This can be a very simple / rough screencast where you walk through your work.

Both of these items (a URL to the Youtube video) are to be uploaded to canvas.

Due date: **Wed 6/10/20 at midnight.**

COURSE GRADE:

Final grades for the course will be based on the homework assignment grades (65%) and the project (35%).