Math 490: Mathematics of Machine Learning (3 credits)

Course Description

Machine learning is a growing field at the intersection of probability, statistics, optimization, and computer science, which aims to develop algorithms for making predictions based on data. This course will cover foundational models and mathematics for machine learning, including statistical learning theory and neural networks with a project component.

Prerequisites: Math 461 or Stat 410 and one of CS 101 or 125 or equivalent.

Course Objectives

- Record and document mathematics behind the machine learning algorithms in terms of strengths and weaknesses of the underlying problems
- Describe how to avoid overfitting issues that arises while dealing with large datasets using some stable machine learning algorithms
- Be able to apply mathematical modeling in improving the learnability of the machine learning algorithms
- Be able to implement different machine learning algorithms in python for real world problems such as research articles review or building a business model
- Justify the use of a particular machine learning algorithm by doing the comparative analysis of the method with other similar algorithms in terms of their mathematical vs. computational complexity or stability
- Demonstrate the knowledge about theoretical concepts of machine learning such as probably approximately correct (PAC), convex learning problems, regularized loss minimization

Course Content

1. Probability background, machine learning set-up, defining error/loss/risk, empirical risk minimization (ERM) algorithm

2. ERM with inductive bias, defining learnability, probably approximately correct (PAC), learnable, agnostic PAC-learnable, releasing the realizability assumption, generalized loss functions

3. Finite Classes are agnostic PAC learnable, bias-complexity tradeoff, no-free-lunch theorem

4. Error decomposition, approximation error, estimation error, example of learnable infinite-size classes, Vapnik–Chervonenkis dimension

5. Threshold functions, intervals, axis-aligned rectangles, fundamental theorem of machine learning, proof of Sauer’s lemma and the growth function

6. Uniform convergence for classes of small effective size theorem, linear predictors, halfspaces, linear programming for the class of halfspaces, perceptron for halfspaces, Vapnik–Chervonenkis dimension of halfspaces, linear regression algorithm

7. Least squares, linear regression for polynomials, regression tasks, logistic regression, adaptive boosting

8. Weak learner, efficient Implementation of ERM for decision stumps, AdaBoost algorithm, linear combinations of hypotheses, Vapnik–Chervonenkis dimension of $L(B,T)$, application of boosting algorithm in computer vision for facial recognition

9. Convex learning problems, Lipschitz functions, smoothness, learnability of convex learning problems, surrogate loss functions, overview about regularization and stability

10. Regularized loss minimization, stable algorithms do not overfit, Tikhonov regularization as a stabilizer, controlling the fitting/stability tradeoff, stochastic gradient descent (SGD), gradient descent

11. Subgradients for non-differentiable functions, SGD algorithm for loss minimization, SGD for risk minimization, SGD for regularized loss minimization, support vector machines (SVM), hard margin SVM

12. Soft margin SVM, SVM optimality conditions, implementing soft margin SVM with SGD

13. Kernel method, kernel trick, representer theorem, implementing soft margin SVM with kernels, decision trees algorithm, nearest neighbor (NN) algorithmsk, NN algorithm for binary classification

14. Artificial neural networks (ANNs), learning neural networks, expressive power of NNs, sample complexity of NNs

Format

- This is an online course featuring video lectures from the UIUC Fall 2019 course taught by Dr. Kay Kirkpatrick

- Text: Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge: Cambridge University Press.

- Students must be able to print out assignments, write out solutions, then scan their written work and upload them to Moodle

- Students will use Python both within the lessons and to complete two midterm projects and will need to download and use Anaconda software to their computer.

- This course requires two paper-based midterm exams. Students will schedule their exams with NetMath and access their exams online. There is also a Python-based final project.