EE-UY/CS-UY 4563: Introduction to Machine Learning

Overview

This course provides a hands on approach to machine learning and statistical pattern recognition. The course describes fundamental algorithms for linear regression, classification, model selection, support vector machines, neural networks, dimensionality reduction and clustering. The course includes computer exercises on real and synthetic data using current software tools. A number of applications are demonstrated on audio and image processing, text classification, and more. Students should have competency in computer programming

- Prof: Sundeep Rangan, srangan@nyu.edu
- TA: Ish Jain , ishjain@nyu.edu
- Lectures TuTh 4:30-5:50 Room RH315
- Optional weekly recitations. Time to be scheduled
  - Help with Python labs and homework.
- Texts:
  - James, Witten, Hastie and Tibshirani, "An Introduction to Statistical Learning", https://web.stanford.edu/~hastie/local.ftp/Springer/ISLR_print1.pdf
    Note: While this text uses R, the class will be in Python.
  - Raschka, "Python Machine Learning", 2015. http://file.allitebooks.com/20151017/Python%20Machine%20Learning.pdf
- Supplementary texts and resources
  - Bishop, "Pattern Recognition and Machine Learning"
  - Installing python (need to do this before first recitation): http://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/index.html
  - Python tutorial: https://docs.python.org/3/tutorial/
- Grading:
  - Midterm 1: 25%, Midterm 2: 25%, Final project: 25%, Labs, homework & quizzes 25%,
  - Labs will involve approximately six python-based exercises.
  - Final project is done in groups of two.
  - Midterm exams are closed book with cheat sheets. Students will need to be able to write simple python in the exams.
- Pre-requisites:
  - One of: MA-UY 2224 (Data Analysis); MA-UY 2222 (Data Analysis 2) or EE-UY 2223 (Probability) or equivalent.
  - Programming experience is essential, including some exposure or willingness to learn object-oriented programming.
  - No experience in python is required as python will be taught as part of the class.

Tentative Outline

- Introduction
  - Provide examples of machine learning problems used today
  - Formulate machine learning problems (identify task, data, objectives)
  - Classify ML problems as supervised vs. unsupervised, regression vs. classification
  - For supervised learning, identify the predictors and target variables
  - Determine the role of expert knowledge in the task vs. data-driven learning
  - Install python and jupyter notebook and run simple programs
- Single Variable Linear regression
  - Read and parse data from a text file in python
  - Visualize data via a scatter plot
  - Formulate a linear model for data. Identify dependent, independent variables.
  - Compute and draw linear fit of data
  - Assess model accuracy via R^2 and RSE
  - Perform vector manipulations in python (slicing, vectorized operations, etc.)
  - Demo: Analyze automobile mileage data
  - Lab: Analyze Boston housing data
- Multiple variable regression
  - Describe a multiple variable linear model for data
  - Represent data in matrix form and perform simple linear algebra calculations on data
  - Derive the LS solution for multiple variable regression
  - Extend multiple regression models with nonlinear transforms
  - Manipulate multi-dimensional arrays in python
  - Compute LS solutions for data using python packages
  - Represent time-series via multiple linear regression
  - Demo: Understanding factors for blood glucose levels
  - Lab: Characterize robot dynamics
- Model validation and cross validation
  - Identify the order in a multiple linear regression model
  - Visually identify overfitting and underfitting
  - Perform simple cross validation tests (K-fold CV, LOOCV)
  - Demo: Polynomial model selection
  - Lab: Continue previous lab
- Midterm 1
- Linear classification and Logistic Regression
  - Formulate a machine learning problem as a classification problem
  - Visualize linear classification data using a scatter plot.
  - Describe a linear classifier as an equation and on a plot.
  - Determine visually if data is perfectly linearly separable.
  - Formulate a classification problem using logistic regression (Binary and multi-class)
  - Describe the logistic and soft-max function
  - Derive the loss function for ML estimation of the weights in logistic regression
  - Use sklearn packages to fit logistic regression models
  - Measure the accuracy of classification

  - Adjust threshold of classifiers for trading off types of classification errors. Draw a ROC curve.
  - Demo: Diagnosis breast cancer from characteristics of samples
  - Lab: Identify genes for Down's syndrome in mice from gene expression data
- Gradient descent optimization
  - Describe machine learning problems as optimization problem
  - Compute the gradient of a multiple variable function
  - Compute gradients under linear transforms
  - Describe and program a simple gradient descent algorithm
  - Describe the role of step size in gradient descent
  - Demo: Nonlinear least squares
- Support vector machines (SVMs)
  - Describe image classification as a classification problem
  - Interpret weights in linear classification of images
  - Define the margin in linear classification
  - Describe the SVM classification problem.
  - Write equations for solutions of constrained optimization using the Lagrangian.
  - Describe a kernel SVM problem
  - Select SVM parameters from cross-validation
  - Use python packages for diplaying images and computing SVMs
  - Demo: MNIST digit classification using SVM
  - Lab: Extend MNIST with alpha-numeric characters
- Neural networks and Tensorflow
  - Describe a simple neural network model with a single hidden layer
  - Describe piecewise functions via neural network
  - Describe a NN with a computation graph, and derive the backpropagation equations for gradient descent using the computation graph
  - Install Tensorflow, define a computation graph for a simple model and train the model
  - Describe the stochastic gradient descent algorithm with mini-batches and implement SGD in Tensorflow.
  - Write a python program with multiple files using import.
  - Demo: Synthetic data for a nonlinear 2D classification
  - Lab: Music instrument classification from mel features
- Midterm 2
- Convolutional and deep networks
  - Write equations for 1D and 2D convolution and describe the convolutional match filter for a feature
  - Describe a multi-layer NN with convolutional, max pooling and fully connected layers
  - Implement and train a convolutional NN in Tensorflow
  - Build an image reader in Tensorflow for large datasets
  - Deploy a python program in the cloud [Note: Google Cloud or AWS is required]
  - Demo: CIFAR10 image classification in Tensorflow [Note GPU is required]
  - Lab: None. But, Assistance will be provided if students have interest for their projects.
- PCA
  - Describe dimensionality reduction and its role

  - Derive the equations for the PCA of data
  - Compute the PCA using an SVD
  - Derive the proportion of variance from the SVD.
  - Implement PCA and reconstructions in python using SVD
  - Demo: Face approximations using the Labelled Faces in the Wild dataset
  - Lab: Use PCA for dimensionality reduction in neural electrode recordings
- Clustering and K-Means
  - Describe a cluster of data and distance function
  - Describe the K-means algorithm
  - Describe Bag of Words model in text, stopwords, TF-IDF scores
  - Perform K-means in python using built-in functions and the sklearn TfIDfVectorizer.
  - Demo: Document classification for 20 newsgroup data
  - Lab: None. Will assist students with these techniques if needed for their project.
- Final project