

UCI-Math10

This is the repository for Math 10 Intro to Programming for Data Science

Math 10 is the first dedicated programming class in the Data Science specialization designed mainly for Math majors at University of California Irvine. Some of current de facto algorithms will be featured, and some theorems in Mathematics behind in data science/machine learning are to be verified using Python, and the format can be adapted to other popular languages like R and Julia.

Prerequisites:

MATH 2D Multivariate Calculus

MATH 3A Linear Algebra(can be taken concurrently)

MATH 9 Introduction to Programming for Numerical Analysis

Recommended:

MATH 130A Probability I

ICS 31 Introduction to Programming

Lecture notes (Jupyter notebooks) are available in the Lectures folder.

Lecture	Contents
Lecture 1	Intro to Jupyter notebooks, expressions, operations, variables
Lecture 2	Defining your own functions, types (float, bool, int), Lists, IF-ELSE

Lecture	Contents
Lecture 3	Numpy arrays I, tuples, slicing
Lecture 4	Numpy arrays II, WHILE and FOR loops vs vectorization
Lecture 5	Numpy arrays III, advanced slicing; Matplotlib I, pyplot
Lecture 6	Numpy arrays IV, Linear algebra routines
Lecture 7	Matplotlib II, histograms
Lecture 8	Randomness I; Matplotlib III, scatter plot
Lecture 9	Randomness II, descriptive statistics, sampling data
Lecture 10	Randomness III, random walks, Law of large numbers
Lecture 11	Introduction to class and methods, object-oriented programming
Lecture 12	Optimization I: Optimizing functions, gradient descent
Lecture 13	Fitting data I: Linear model, regression, least-square
Lecture 14	Optimization II: Solving linear regression by gradient descent
Lecture 15	Fitting data II: Overfitting, interpolation, multivariate linear regression
Lecture 16	Classification I: Bayesian classification, supervised learning models
Lecture 17	Classification II: Logistic regression, binary classifier
Lecture 18	Classification III: Softmax regression, multiclass classifier

Lecture	Contents
Lecture 19	Optimization III: Stochastic gradient descent
Lecture 20	Classification IV: K-nearest neighbor
Lecture 21	Dimension reduction: Singular Value Decomposition (SVD), Principal Component Analysis
Lecture 22	Feedforward Neural Networks I: models, activation functions, regularizations
Lecture 23	Feedforward Neural Networks II: backpropagation
Lecture 24	KFold, PyTorch, Autograd, and other tools to look at

Labs and Homeworks

There are two Labs per week. One is a Lab exercise, aiming to review and sharpen your programming skills. The other is a graded Lab assignment, which is like a collaborative programming quiz. Homework is assigned on a weekly basis, the later ones may look a mini project. Lab assignments' and Homeworks' solutions are available on [Canvas](#).

Textbook

No official textbook but we will use the following as references: [Scientific Computation: Python Hacking for Math Junkies. Version3, With iPython](#) (Math 9 reference book)

[Python Data Science Handbook. Online version](#)

Software

Python 3 and Jupyter notebook (iPython). Please install [Anaconda](#). To start Jupyter notebook, you can either use the Anaconda Navigator GUI, or start Terminal on Mac OS/Linux, Anaconda prompt on Windows: in the directory of .ipynbfile, run the command `jupyter notebook` to start a notebook in your browser (Chrome recommended). If Jupyter complains that a specific package is missing when you run your notebook, then

return to the command line, execute `conda install <name of package>`, and re-run the notebook cell.

Final Project

There is one final project using Kaggle in-class competition. A standard classification problem similar to the Kaggle famous starter competition [Digit Recognizer](#) based on MNIST dataset will be featured. You will use the techniques learned in class and not in class (e.g., random forest, gradient boosting, etc) to classify objects.

- [Winter 2019 final project: Learn the handwritten characters in ancient Japanese](#)
- [Spring 2019 final project: Is your algorithm fashionable enough to classify sneakers?](#)

Acknowledgements

A major portion of the first half of the course is adapted from [Umut Isik's Math 9 in Winter 2017](#) with much more emphases on vectorization, and instead the materials are presented using classic toy examples in data science (Iris, wine quality, Boston housing prices, MNIST, etc). Part of the second half of this course (regressions, classifications, multi-layer neural net, PCA) is adapted from [Stanford Deep Learning Tutorial](#)'s MATLAB codes to vectorized implementations in `numpy` from scratch, together with their `scikit-learn`'s counterparts.