Abstract: ED43C-1124

Equipping undergraduates in the physical sciences with critical machine learning skills

Jacob Bortnik[1]
[1]University of California at Los Angeles (UCLA)


Courtesy of Dr. Enrico Camporeale

Abstract

Machine learning (ML) has become an extremely popular topic in recent years, covering a broad range of applications from autonomous vehicles, to gaming, finance, logistical planning, advertising and even basic and applied science. Recent studies show that the number of workers in ML worldwide is currently about 300,000 but estimates shows that several million are needed, since roughly 70% of businesses will employ at least one form of ML in the coming decade.

On the other hand, university curricula have not kept pace with market demands. Most students in the division of physical sciences at UCLA still graduate without having any experience with ML, sometimes without ever having programmed a computer all, and thus need to seek out other resources to gain some ML skills in order to become marketable in the workplace. To address this felt need, we are introducing a new course at UCLA entitled "machine learning for the physical sciences", which will be aimed at upper-division undergraduates in the division of physical sciences, taught in Python with the aid of Jupyter notebooks, and focused in such a way as to make bright, mathematically-literate undergraduates with minimal programming experience maximally employable in both industry and academia.

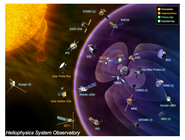
Introduction

ML has become extremely popular in recent years, causing in a huge interest among students. A few notable achievements:

- Developing scientific theory (Lobo & Levin, 2015)
- Passing 8th grade science test
- Autonomous vehicles, vision, NLP, and more!

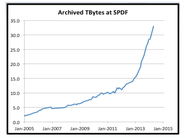
Industry (i.e., lots of jobs!):

- ML widely adopted in industry, Tencent estimates: ~300,000 AI professionals currently worldwide, but millions needed
- There is a "critical skills crisis"
- McKinsey Global Institute: by 2030, 70% of companies will adopt at least one form of AI, add $13 trillion economic output (1.2% global GDP)
- Ranked No. 1 job on Glassdoor!


Heliophysics System Observatory

Science (i.e., lots of data!):

- Science data is growing much fast than Moore's Law, e.g., genomics has 10,000x increase in 6 years [Kahn, Science, 331, 6018, 728-729]
- NASA: "Within 5 years, all SMD divisions are predicted to generate more than 100PB of data annually";
- E.g. SKA: 11 exabytes per day!



Who is This Class for?

Upper-division undergraduate students in the physical sciences:

- Departments: Math, Statistics, Chemistry, Physics, Earth & Planetary & Space Sciences, Atmos & Oceanic Sciences
- Have numeracy but little knowledge of machine learning
- Have some programming skills (preferably in Python)
- Graduating soon, thinking about jobs in industry or graduate school, and needing up-to-date set of tools

Beginning graduate students in the physical sciences:

- Have little or no background in machine learning and want to understand what it's about and how to apply it to their research
- To understand the research of other workers using ML
- As a first step to more advanced ML techniques (deep learning, convNets, RNN's etc.), as focus of their research

What Will Students get out of This Class?

- a good foundation in practical ML to understand others' work
- ability to immediately build their own machine learning models
- And be able to delve into more advanced deep learning models

Graduating students will be able to **tell potential employers** that:

- They have experience programming in Python, and used the popular ML library SciKitLearn with numPy and MatPlotLib
- Have experience loading, cleaning, and visualizing real data
- Have compiled several Jupyter notebooks where they have implemented 7 of the most popular ML algorithms
- Have worked on a complete, end-to-end ML project involving real data (of their own choosing), and will be able to show their source code as a Jupyter notebook, a complete high-quality final report describing the project, and be able to give the "elevator pitch", based on their presentation of the project.

Language, Tools, and Libraries

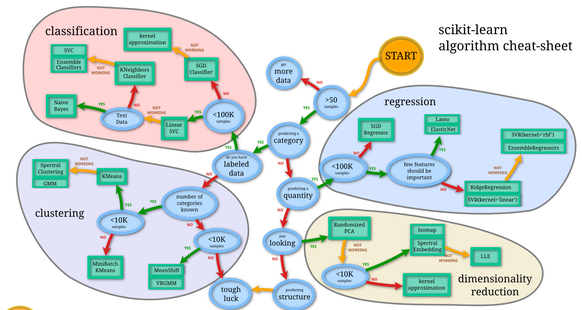
Python: *Top* programming language for AI/ML; free & powerful
Google colaboratory: is a free, online implementation of Jupyter notebooks, linked to the students google account (which all UCLA students get automatically).

- Students can write, edit, and run code entirely online
- No setup of Python+libraries is required on students' machines, so no compatibility or OS issues
- No need for powerful computer, google colab will run on any browser, even phone or tablet

NumPy: a very popular matlab-like numerical library in Python upon which most (Python) ML libraries are built
MatPlotLib: Popular and powerful matlab-like plotting library
SciKitLearn: Most popular ML library for basic algorithms
Chosen algorithms: Survey of ~20-30 websites of most popular ML algorithms. Same as Burkov's "100-page ML book".

Proposed syllabus

WEEK (in 2020)	TOPIC	IN CLASS CODING (using only NumPy)	EVALUATION (WITH SKL)
1. Mar 30-Apr 3	Introduction: what is machine learning?	Intro to Jupyter notebooks (google colaboratory), NumPy, plotting, SK-Learn	--
2. Apr 6-10	**(I) Linear regression** (supervised-regress)	Use linear regression on real data; implement matrix inversion	HW 1
3. Apr 13-17	**(II) Logistic regression** (supervised-classify)	Implement gradient descent algorithm Implement logistic regression	HW 2
4. Apr 20-24	**(III) Decision trees/forests** (supervised-classify, regress & ensemble methods)	Implement simple decision tree	HW 3
5. Apr 27- May 1	**(IV) Support Vector Machines** (supervised Regression & Classification)	Implement SVM	Project proposals due
6. May 4-8	**(V) K-means** (unsupervised-clustering)	Implement simple K-means algorithm	HW 4
7. May 11-15	**(IV) Principle Component Analysis** (unsupervised-dimension reduction)	Implement SVM using NumPy tools	--
8. May 18-22	**(VII) Neural networks** (supervised regression & classification)	Implement simple, feedforward, single-hidden layer neural net	HW 5
9. May 25-29	More on neural networks, backpropagation, finish.	Work on final projects	--
10. Jun 1-Jun 5	In-class final project presentations		Final projects due



Summary, Class Structure and Evaluation

10-week class: "**AOS111: Introduction to Machine Learning for the Physical Sciences**". Evaluation consists of 3 parts:
[20%] In-class attendance and participation (2%/week x 10 weeks)
[40%] HW sets, 8% each. 3 before the midterm week, and 2 after the midterm week. They will work with real (provided) datasets, and extend the algorithms we cover in class with more extensive parameter scans and visualization. Jupyter notebooks with annotation in MD and visualization to be submitted.
[40%] Final projects. Problems are defined by the students themselves, using datasets of their own choosing. To be submitted: Jupyter notebooks with all code+dataset [10%], a written report [20%], and oral presentation in last week [10%]

Contact

Prof. Jacob Bortnik
Department of Atmospheric and Oceanic Sciences, UCLA
Email: jbortnik@gmail.com
Website: https://dept.atmos.ucla.edu/jbortnik

Class References

1. Marsland, Stephen (2014), "*Machine Learning: An Algorithmic Perspective, Second Edition*", Published by Chapman & Hall/CRC Machine Learning & Pattern Recognition 2nd Ed., ISBN-10: 1466583282; ISBN-13: 978-1466583283.
2. Burkov, Andriy (2019), "*The Hundred-Page Machine Learning Book*", Published by Andriy Burkov, ISBN-10: 199957950X; ISBN-13: 978-1999579500.
3. Geron, Aurelien (2017), "Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 1st Edition", Published by O'Reilly Media, ISBN-10: 1491962291; ISBN-13: 978-1491962299.