Course Logistics and Overview

David S. Rosenberg

New York University

January 23, 2018
Logistics
Logistics

- **Class webpage**: [https://davidrosenberg.github.io/ml2018](https://davidrosenberg.github.io/ml2018)
  - Syllabus on the website
- **Piazza**: [https://piazza.com/nyu/spring2018/dsga1003](https://piazza.com/nyu/spring2018/dsga1003)
  - **All class announcements via Piazza**
  - Ask all questions on Piazza
- **Class Times**
  - Tuesdays “Lecture”: 5:20 - 7pm (GSACL C95)
  - Wednesdays “Lab”: 6:45 - 7:35pm (Meyer 121)
  - (Both are required.)
Course Staff

TA:
- Ben Jakubowski (CDS MS Data Science, 2017)

Graders:
- Lisa Ren (Head Grader)
- Utku Evci
- Mi Fang
- Sanyam Kapoor
- Nan Wu
- Zemin Yu

Project Advisers:
- Kurt Miller, Brian d’Alessandro, Bonnie Ray, Daniel Chen, Elliot Ash, Vitaly Kuznetsov, David Frohardt-Lane
Evaluation

- About 7 or 8 homeworks (40%)
- Two tests (40%)
  - Midterm Exam (20%) in Week 7 (March 6th)
  - Final Exam (20%) - Final Exam Period (tentatively May 15th)
- Project (20%)
  - Project proposal (Week 8) and project report (Week 15)
- These scores determine “class rank”.
- Typical grade distribution: A (40%), A- (20%), B+ (20%), B (10%), B- (5%), <B- (5%)
There will be a significant number of **optional homework problems**

**Grade-wise**

- Optional problems **do not** contribute to your homework grade.
- They are a separate grade category
- Primarily used to boost a borderline grade at the end of the term
- **At most, increases final grade by half a letter (e.g. B+ to A-)**
  - In 2017, about 10% of people has letter grade increases from optional credit.
  - (To a lesser extent, Piazza and class participation can also help bump up a borderline grade.)

- It’s primarily for highly motivated individuals (who have the time) to
  - Learn more concepts and practice more techniques

- High performance on optional homework is something I can mention in recommendation letters.
Lab Sessions

- Some led by TA Ben Jakubowski, some by me
- Most will be lecture format
- Meetings with project advisors
- Tomorrow: Guest lecture from Brett Bernstein (last year’s TA)
Homework (40%)

- First assignment out now – due week from Thursday 10pm
- Submit with Gradescope (details on website)
- Homeworks should be submitted as a PDF document.
- Late homework: Accepted up to 48 hours late with 20% penalty
- Collaboration is fine, but
  - Write up solutions and code on your own
  - List names of who you talked to about each problem
- When graders identify copying, we’re obliged to tell the administration, which gets uncomfortable for everybody.
Some notes on website, and will be posting more information on Piazza.

Logistics:
- 3 students per group (exceptions possible)
- First meeting with advisers (Wed, March 7)
- Project proposal due after Spring Break (Thurs, March 22)

Some project advisers supply code and project ideas
- Law and economics (Daniel Chen and Elliott Ash)
- Sports Betting (David Frohardt-Lane)
Prerequisites

- DS-GA 1001: Introduction to Data Science
- DS-GA 1002: Statistical and Mathematical Methods
- Math
  - Multivariate Calculus
  - Linear Algebra
  - Probability Theory
  - Statistics
  - [Preferred] Proof-based linear algebra or real analysis
- Python programming (numpy)
Course Overview and Goals
12 weeks of instruction + 1 week midterm exam + 1 week final exam review

- 4-5 weeks: **Linear** methods for **binary classification** and **regression** (also **kernel methods**)

- 2 Weeks: Conditional **probability models**, **Bayesian** methods

- 1 Week: **Multiclass** and introduction to **structured prediction**

- 3-4 weeks: **Nonlinear** methods (**trees**, **ensemble** methods, and **neural networks**)

- 2 Weeks: **Unsupervised** learning: **clustering** and **matrix factorization**
High Level Goals of the Class

- Learn fundamental building blocks of machine learning

- Goal is to start seeing
  - **fancy new method A** “is just” familiar thing B + familiar thing C + tweak D
  - SVM “is just” ERM with hinge loss with $\ell_2$ regularization
  - Pegasos “is just” SVM with SGD with a particular step size rule
  - Random forest “is just” bagging with trees, with an interesting tweak on choosing splitting variables
We will learn how to build all ML algorithms **from scratch** – no ML libraries, just numpy.

Once we have built it from scratch once, we can use the sklearn version.

For projects, you should NOT code ML algorithms yourself, except in exceptional circumstances
- use existing frameworks (sklearn, xgboost, tensorflow, etc)