Elements of Machine Learning

https://www.cs.duke.edu/courses/fall19/compsci371d/

Introduction and Logistics
Machine Learning Applications

- **Data Security**: Is this file malware?
- **Fraud Detection**: Is this transaction money laundering?
- **Personal Security**: What’s in your bag? Is that you?
- **Photo Collections**: Here are all photos of Jenny playing tennis
- **Financial Trading**: Is this trade likely to profit me?
- **Healthcare**: Does this scan have a tumor? Do these symptoms suggest diabetes?
- **Marketing Personalization**: What can I sell *you*? What movies do you like?
- **Online Search**: Why did/didn’t you like this search result?
- **Speech Processing**: What did you say? Let me transfer your call
- **Natural Language Processing**: Here is the information you need
- **Chatbots**: I can help you with your order. Tell me more about your symptoms
- **Smart Cars**: Are you comfortable? Are you alert? Stay in lane! Let me drive…
Machine Learning in One Slide

- Identify a function $y = f(x)$:
  
  $$x = \text{email, } y = \text{SPAM/NO SPAM}$$

- Give lots of examples (a training set):
  
  $$T = \{(x_1, y_1), \ldots, (x_N, y_N)\}$$

- A learner is another function $\lambda$:
  
  It takes $T$ as input and outputs an approximation to $f$:

  $$h = \lambda(T)$$

- Hopefully, $f$ and $h$ behave about the same even for previously unseen data:

  $$h(x) \approx f(x)$$

- That’s the big problem!

- ML is not (just) data fitting
Logistics
Emergency Procedures

- Know your **exits**
- **Stop me** if you see something dangerous I don’t see
- **Run** if you can (leave stuff behind, hands visible)
- **Hide** otherwise (lock yourself in if possible)
- **Fight** if you must, and if your life is in immediate danger
- [More details linked from the class web page announcements]
Academic Integrity

- **Short version: Cheating will be prosecuted**
- Cheating: Using someone else’s material in your work without giving credit [Lone exception: class materials need not be cited]
- Ditto for making materials available to others
- Giver/receiver are treated the same
- Format for using/making available is immaterial
- Only communication allowed during homework is with your group peers, if any, and with the teaching staff
371D

- **371D**: Remember the D when you look for web sites, etc.

- **371D**: “3” means “advanced undergraduate course”
  
  - Possibly more prerequisites

  - No drama with programming, installing, …

  - Includes picking up Python and Jupyter notebooks (but we’ll help)
Teaching Staff

- **Graduate TAs**: Tianyu Wang, Shuai Yuan
- **Undergraduate TAs**: Alex Rubin, Brandon Guo, Christopher Wolff, Diane Hu, Lucas Liu, Stephanie Mei, Suchir Bhatt, Venkat Subramaniam, Xiaolan You

- If you like this course, please volunteer to TA next year!
- Each of us will have two office hours per week
  > 20 office hours per week. Times and venues TBA

- **Check the online calendar before you go to office hours**
- Four recitation sections per week
- We’ll keep listening to Piazza (at reasonable hours)

- **Talk to us!** We are here to help you learn
Recitation Sections

• An opportunity to dig deeper and clarify

• **Ask questions**

• Recitations are mandatory

• Stick with your recitation section. Ask TAs involved *well ahead of time* if you need to change for a serious reason

• **Recitations start after Labor Day**

• All recitations are 50 minutes, regardless of DukeHub listing
Lecture Notes

• There are many good books on machine learning (see class Resources page)

• There aren’t many books that support an undergraduate machine learning class well

• This is good but terse and theoretical: S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning*, Cambridge, 2014. Optional reading

• I’ll refresh notes on the class Syllabus web page as we go along. They are required reading, and your main source of information

• All appendices are optional reading

• A status bar on the Syllabus page will show where we are

• Feel free to integrate with other sources. See Resources web page
Programming

• All programming will be in **Python 3** (not 2!)
• If you know how to program, picking up Python takes a few hours and Google while you program
• If you don’t know how to program, this class may not be for you
• You will write **Jupyter Notebooks** for homework. They are easy to get used to, and let you intersperse text, math, figures, and code
• A first homework assignment will help you ease into these tools
• The **Anaconda** distribution for everything you need is very strongly recommended
• See the **Resources** web page for tutorials on Python 3, Jupyter, Anaconda
Homework

• About one per week
• Some math, some text, some programming
• OK to work in groups of one, two, three [but no division of labor!]
• One submission per group, remember to list all names!
• No late homework allowed
• Lowest assignment grade (including a 0) dropped
• Jupyter notebooks → HTML → PDF
• Keep Jupyter cells small
• Two submissions on Gradescope: PDF, Notebook
Grades

• Midterm 25%, Final 25%, Homework 45%, Participation 5%

• Participation includes attendance, engagement, and filling out a teaching evaluation form

• **No late homework**

• **No excuses for absences** (includes sports, illness, etc)

• However
  
  • Lowest homework score is dropped
  
  • No penalty for 0-5 absences total (across lectures and recitations)