JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Computer Science EN 601.475/675
Machine Learning
Spring 2020

Course Syllabus version 4

Course Information

Course Location:	Remsen Hall 101
Lecture Times:	Monday & Wednesday, 4:30 – 5:45 PM
Recitation Time:	Friday, 4:30 – 5:45 PM
Course Instructor:	Dr. Philip Graff (pgraff2@cs.jhu.edu, pgraff2@jhu.edu)
Office Hours:	Monday & Wednesday, 6 – 7 PM, Remsen 101
Teaching Assistant:	Molly O'Brien (molly@jhu.edu)
Office Hours:	Tuesday, 3 – 4 PM, Malone 216
Course Assistants:	Matthew Figdore (mfigdor1@jhu.edu, co-Head CA) Darius Irani (dirani2@jhu.edu, co-Head CA) Morgan Hobson (mhobson5@jhu.edu), Spencer Loggia (sloggia1@jhu.edu), Jiatong Shi (jshi34@jhu.edu), Siqi Tang (stang46@jhu.edu), Jason Wong (jwong62@jhu.edu), Darren Edmonds (dedmon13@jhu.edu), Fei Wu (fwu24@jhu.edu)

Course Description

This course takes an application-driven approach to current topics in machine learning. The course covers supervised learning, unsupervised learning, semi-supervised learning, and several other learning settings. We will cover popular algorithms and will focus on how statistical learning algorithms are applied to real world applications. Students will implement several learning algorithms throughout the semester. The goal of this course is to provide students with the basic tools they need to approach various applications, such as:

- Biology/Bioinformatics
- Information Retrieval
- Natural Language Processing
- Speech Processing
- Computer Vision

We will focus on fundamental methods applicable to all applications. Application specific techniques, such as feature extraction, will be covered only to the benefit of understanding the basic methods.

Course Goals

This course has three main goals:

1. Students will learn the fundamentals of machine learning
2. Students will learn to implement machine learning algorithms
3. Students will learn how to apply machine learning to different settings and how to evaluate the results and models they obtain

Course Requirements

This course expects students to have strong programming skills in Python 3. All assignments and in-class examples will be done in Python 3. Students are also expected to have experience and comfort with the relevant mathematical topics, namely linear algebra, multi-variate calculus, and probability.

Required Textbooks

The official textbook for this course is *Machine Learning: A Probabilistic Perspective* by Kevin Murphy. There is only one edition of the book, but multiple printings. Later printings will contain fixes for errors found in earlier versions of the book. Page numberings may be different, but section numbers (which we will be using), are the same. Having the single, official, book provides a consistent presentation of the material across all topics.

A secondary book that will also be drawn from heavily is *Pattern Recognition and Machine Learning* by Chris Bishop. Sections from this book are also provided in the readings list. Additional readings may also be provided for particular topics. These may offer a different – or better – representation of the material. These will be available freely online.

Murphy is more up to date than Bishop and covers more topics, along with a notation more common in the machine learning community. However, Bishop does present some topics more clearly than Murphy. A lot of the course structure is originally based on Bishop and so readings in Murphy may jump around.

Assignments and Grading

Grading for this class will be done through Gradescope. All students must enroll in the course on Gradscope.com with their JHED email address so that their grades can be properly linked back to them. The course “Entry Code” is **MK8J8N**. All assignments must be submitted on Gradescope and all grades and re-grade requests will be handled through that website.

The course will have a total of 1000 points, with a student's final grade being determined by the number of points they earn. There will be five (5) homework assignments (a.k.a. "Projects") throughout the semester. These will have a total value of 500 points, although they may not be distributed evenly. Each project will consist of a written portion and a programming portion. Written portions will ask questions that cover fundamental concepts covered in the class (lectures and readings). There will also be a midterm exam and a final exam, each worth 250 points.

Class discussions about projects, exams, and any other relevant topics will be held on Piazza. Students can sign up for the class through Piazza.com at the following link:
https://piazza.com/jhu/spring2020/601475
The main course page is available at:
https://piazza.com/jhu/spring2020/601475/home

Late Policy

Late homework and project assignments will be accepted up to 72 hours past the due date. Exceptions will only be given in extreme cases. However, every student is permitted to hand-in project assignments late without penalty using a 72-hour grace period for the entire semester. This means that you can choose to hand-in the first homework 70 hours late and the second homework 2 hours late, but then every other homework and project assignment must be on time for the rest of the semester. You may divide these 72 hours as you see fit, but once you have used up all of the time, you will be given no more. I will round-up to the hour (minutes don't count). Each late hour over the maximum 72 will result in the loss of a point out of the total for the course.

Missing a deadline can be stressful, and it is not always within your control. Issues arise both academic and personal that cause you to fall behind. Students often blame themselves and believe that if they work harder, they can catch up, only to fall further behind. We understand that difficult situations arise, and we want to help you manage them to ensure you can stay on track with the course. The key is to email the instructor as soon as possible when you think you may miss a deadline. Note that you don't have to email if you plan to use late hours, only if you believe the late hours will be insufficient, or if you have an emergency. If you contact the instructor ahead of time, we may be able to work together to ensure that you are not penalized for late submissions. However, if we find out after the deadline has passed, we are very limited in our ability to assist.

Cheating Policy

We take cheating very seriously. We expect every student to have read the Department of Computer Science Academic Integrity Code and will hold students accountable to it. So that course policies are clear, here is review of relevant rules (in addition to the integrity code).

- Every exam, project, homework and any other work completed during this course must be entirely your own. Copying any material from other students or the web is expressly prohibited.
- All exams are closed book unless otherwise stated. This means that students may not reference any material during an exam that is not provided as part of the exam.
- Any collaboration between students during an exam will be considered cheating.

- If a student copies your work, even without your knowledge, you are cheating. It is your responsibility to ensure that no one has access to your work.
- There is no statute of limitations on punishing cheating. Even if we find on the last day of the semester that you had cheated on projects, you will be punished.
- Talking with other students to understand homework and course material is strongly encouraged. However, discussing an assignment and cheating are very different things. If you copy someone else's work, you are cheating. If you let someone copy your work, you are cheating. If someone tells you the answer you are cheating. Everything you hand in must be in your own words based on your understanding of the solution.

For homework assignments, here is a list of helpful guidelines. When in doubt, please ask!

Cheating

- Copying any part of a homework from someone else.
- Verbally telling someone the answer to a homework question.
- Looking at someone else's code or solution.
- Obtaining any part of your solution or code from any online resource or software library.

Not-Cheating

- Explaining the homework question to someone else.
- Discuss at a high level the homework.
- Helping someone think through a problem.
- Directing someone to a section of the textbook, reading, or online resource that helps explain a concept.

I am aware that many of the programming assignments will ask students to implement algorithms already available online. I will try to avoid direct duplication when possible. However, you are not permitted to copy any part of your code from other libraries.

What happens when you cheat?

We will be carefully examining projects and exams for signs of cheating. If you cheat, at a minimum you will be given a 0 for the assignment or exam. More likely, you will have the total value of the homework or exam subtracted from your grade, i.e. if you cheat on an exam worth 15% of your grade, you will get a 0 on the exam and have an additional 15% of your grade deducted. In some cases, cheating will be reported to the appropriate university board, which can result in failing the class, suspension of expulsion.

Remember:

- **DO** help each other understand the lectures, readings and projects.
- **DO NOT** complete each other's homework.

Personal Wellbeing

- If you are sick, particularly with an illness that may be contagious, notify me by email but do not come to class. Rather, visit the Health and Wellness: 1 East 31 Street, 410-516-8270.

See also http://studentaffairs.jhu.edu/student-life/support-and-assistance/absences-from-class/illness-note-policy/

- All students with disabilities who require accommodations for this course should contact me at their earliest convenience to discuss their specific needs. If you have a documented disability, you must be registered with the JHU Office for Student Disability Services (385 Garland Hall; 410-516-4720; http://web.jhu.edu/disabilities/) to receive accommodations.
- If you are struggling with anxiety, stress, depression or other mental health related concerns, please consider visiting the JHU Counseling Center. If you are concerned about a friend, please encourage that person to seek out our services. The Counseling Center is located at 3003 North Charles Street in Suite S-200 and can be reached at 410-516-8278 and online at http://studentaffairs.jhu.edu/counselingcenter/

Classroom Climate

I am committed to creating a classroom environment that values the diversity of experiences and perspectives that all students bring. Everyone here has the right to be treated with dignity and respect. I believe fostering an inclusive climate is important because research and my experience show that students who interact with peers who are different from themselves learn new things and experience tangible educational outcomes. Please join me in creating a welcoming and vibrant classroom climate. Note that you should expect to be challenged intellectually by me, the TAs, and your peers, and at times this may feel uncomfortable. Indeed, it can be helpful to be pushed sometimes in order to learn and grow. But at no time in this learning process should someone be singled out or treated unequally on the basis of any seen or unseen part of their identity.

If you ever have concerns in this course about harassment, discrimination, or any unequal treatment, or if you seek accommodations or resources, I invite you to share directly with me or the TAs. I promise that we will take your communication seriously and to seek mutually acceptable resolutions and accommodations. Reporting will never impact your course grade. You may also share concerns with the Computer Science Department Chair (Prof. Randal Burns, randal@cs.jhu.edu), the Director of Undergraduate Studies (Prof. Joanne Selinsky, joanne@cs.jhu.edu), the Assistant Dean for Diversity and Inclusion (Darlene Saporu, dsaporu@jhu.edu), or the Office of Institutional Equity (oie@jhu.edu). In handling reports, people will protect your privacy as much as possible, but faculty and staff are required to officially report information for some cases (e.g. sexual harassment).

Recitation Sections

Recitation sections are optional class meetings that take place on Friday. These are led by the TAs. Topics will be posted to the course syllabus as they are decided. Topics include covering additional background material, reviewing course material, exploring additional topics, and reviewing homework solutions. In rare instances, a class lecture may be moved to the recitation section time slot on Friday. These will be announced in advance.

Relevant Material

Other Textbooks

Since the explosion of machine learning, there are now many textbooks that can be referred to at varying levels of specificity. Here are just a couple that are good references for theoretical and practical application of machine learning.

- *The Elements of Statistical Learning* by Trevor Hastie, Robert Tibshirani, and Jerome Friedman (2nd Ed., 2016)
- *Information Theory, Inference, and Learning Algorithms* by David MacKay (1st Ed., 2003) (Free online: http://www.inference.org.uk/mackay/itprnn/book.html)
- *Python Data Science Handbook* by Jake VanderPlas (1st Ed., 2016)
- *Python for Data Analysis* by Wes McKinney (2nd Ed., 2017)
- *Hands-On Machine Learning with Scikit-Learn and TensorFlow* by Aurelien Geron (1st Ed., 2017)
- *The Hundred-Page Machine Learning Book* by Andriy Burkov, free online from http://themlbook.com/ as well as sold in e-book and physical copies

Other Courses

Many machine learning courses now have online lectures and notes, a selection is given below. These may be reviewed for alternative presentations of the material.

- Cornell Machine Learning (CS 4780): http://www.cs.cornell.edu/courses/cs4780/2018fa/syllabus/index.html
- Stanford Machine Learning (CS 229): http://cs229.stanford.edu/
- UPenn Machine Learning (CIS520): https://alliance.seas.upenn.edu/~cis520/wiki/
- NYU Machine Learning and Pattern Recognition (G22-2565-001, Fall 2005): https://cs.nyu.edu/~yann/2005f-G22-2565-001/

Software

Students working in Python may familiarize themselves with the libraries below for data analysis and machine learning:

- **Numpy** – efficient library for tabular data operations
- **Scipy** – library of common scientific functions
- **Pandas** – front-end for tabular data analysis with optimized routines and useful interfaces, backed by numpy and used by many developers
- **Scikit-Learn** – library of machine learning algorithms
- **Tensorflow** – Google’s open-sourced deep learning framework
- **PyTorch** – machine learning library with particularly good support for deep learning
- **XGBoost** – boosted decision trees with GPU support

- **NLTK** – Python implementation of many popular NLP algorithms

For data visualization, **matplotlib**, **seaborn**, and **bokeh** are extremely useful.

Other useful software for machine learning includes, but is not limited to:

- **Weka** – a general machine learning tool that has a long pedigree
- **KNIME** – for building data analysis and machine learning pipelines with a GUI
- **libSVM** – standard library for SVMs in C and Java
- **H20.ai** – an open-source machine learning platform with additional commercial products

Johns Hopkins Library Links

- Engineering Research Guide
- Recent machine learning publications and books

Related JH Courses

There are many relevant courses at Johns Hopkins for machine learning:
https://ml.jhu.edu/courses/

Course Schedule

Readings with the prefix “M” indicate *Machine Learning: A Probabilistic Perspective* by Kevin Murphy. For example, “M: 1” means Murphy chapter 1. Unless otherwise noted, skip sections that have a * in the title, these are optional. Alternate readings with the prefix “B” indicate *Pattern Recognition and Machine Learning* by Chris Bishop. Optional readings will be noted in *italics*.

Date	Topics	Readings
Mon, Jan 27	Course Overview Syllabus review, “What is machine learning?” Machine Learning Overview Supervised learning, unsupervised learning, classification, regression	M: 1 B: 1
Wed, Jan 29	Linear Regression Introduces regression problems, under/over-fitting, bias/variance trade-off, model evaluation, model comparison	M: 7, excluding 7.4 and 7.6 B: 3
Fri, Jan 31	*Recitation: Probability* Events, random variables, probabilities, pdf, pmf, cdf, mean, mode, median, variance, multivariate distributions, marginals, conditionals, Bayes theorem, independence	*Those without a probability background should read M: 2.1-2.4.1 and/or B: 2 and appendix B*
Mon, Feb 3	Linear Regression (continued) k-Nearest Neighbors and Naïve Bayes	M: 1.4 (stop at 1.4.3), 3.5 B: 2.5.2
Wed, Feb 5	Logistic Regression Introduces classification problems	M: 8 (stop at 8.3.7), 13.3 B: 4
Fri, Feb 7	*No Recitation*	
Mon, Feb 10	Perceptron Introduces online learning	M: 8.5
Wed, Feb 12	Support Vector Machines Max-margin classification and optimization	M: 14.5 B: 7.1
Fri, Feb 14	*Recitation: Math Review* Linear algebra, calculus, optimization	*B: appendix on Lagrange Multipliers*
Mon, Feb 17	Kernel Methods Dual optimization, kernel trick	M: 14.1-14.2 B: 6.1-6.2
Wed, Feb 19	Decision Trees Construction, pruning, over-fitting	M: 2.8, 16.2
Fri, Feb 21	*Recitation: Recap + TBD*	
Mon, Feb 24	Boosting Ensemble methods	M: 16.4, 16.6 B: 14.1-14.3
Wed, Feb 26	Deep Learning part 1	M: 16.5, 27.7, 28 B: 5.1-5.3, 5.5
Fri, Feb 28	*Recitation: PyTorch Introduction*	
Mon, Mar 2	Deep Learning part 2	
Wed, Mar 4	Deep Learning part 3	
Fri, Mar 6	*Recitation: Midterm Review*	
Mon, Mar 9	Midterm Review or TBD	

Wed, Mar 11	Midterm	
Fri, Mar 13	*No Recitation*	
Mon, Mar 16	*Spring Break*	
Wed, Mar 18	*Spring Break*	
Fri, Mar 20	*Spring Break*	
Mon, Mar 23	Clustering K-Means	M: 25.1, 11 (stop at 11.4) B: 9
Wed, Mar 25	Expectation Maximization part 1	M: 11.4 (stop at 11.4.8)
Fri, Mar 27	*Recitation: Midterm Review*	
Mon, Mar 30	Expectation Maximization part 2	
Wed, Apr 1	Dimensionality Reduction PCA	M: 12.2 B: 12.1-12.3
Fri, Apr 3	*Recitation: Deep Learning Examples*	
Mon, Apr 6	Graphical Models part 1 Bayesian networks and conditional independence	M: 10 B: 8.1-8.2
Wed, Apr 8	Graphical Models part 2 MRFs and exact inference	M: 19.1-19.3, 19.5 B: 8.3-8.4
Fri, Apr 10	*Recitation: TBD*	
Mon, Apr 13	Graphical Models part 3 Inference	M: 20 (stop at 20.3)
Wed, Apr 15	Graphical Models part 4 Max-sum and max-product	
Fri, Apr 17	*Recitation: Graphical Models*	
Mon, Apr 20	Structured Prediction part 1 Margin-based methods, HMMs, CRFs	M: 17 (stop at 17.6), 19.6, 19.7 B: 13.1-13.2
Wed, Apr 22	Structured Prediction part 2 Recurrent neural networks	
Fri, Apr 24	*Recitation: TBD*	
Mon, Apr 27	Practical Machine Learning Building a machine learning/data science project, model bias, ethics	
Wed, Apr 29	Final Review	
Fri, May 1	*No Recitation*	
Wed, May 6	**Final (2 - 5 PM)**	