CS5487

Machine Learning: Principles and Practice

Sem B 2021/22

Dr. Antoni B. Chan

Instructors

- **Course Instructor**
  - Dr. Antoni Chan
    - Email: abchan@cityu.edu.hk
    - Office: Y6414; Zoom PMI: https://cityu.zoom.us/j/3057196306
    - Office Hours: Tue 6-7pm (Zoom link on canvas)
    - Phone: 3442 6509
- **Teaching Assistants (office hours via Zoom)**
  - HU Yao (TBD)
  - LIU Rui (TBD)
  - SHU Weibo (TBD)
  - WU Qiangqiang (TBD)
  - WU Wei (TBD)
  - *More info and Zoom links on Canvas.*

Lecture & Tutorial

- There is a four hour block for this course
  - The actual schedule is different from the official schedule.
  - *Actual class time will be 19:00-22:00*

	Official	Actual
Tue 18:00-19:00	Tutorial (T61)	**Office Hours**
Tue 19:00-20:00	Lecture	**Tutorial**
Tue 20:00-21:00	Lecture	**Lecture**
Tue 21:00-22:00	Tutorial (T62)	**Lecture**

  - Tutorial will cover material from previous lecture
  - Office hours will be held before tutorial

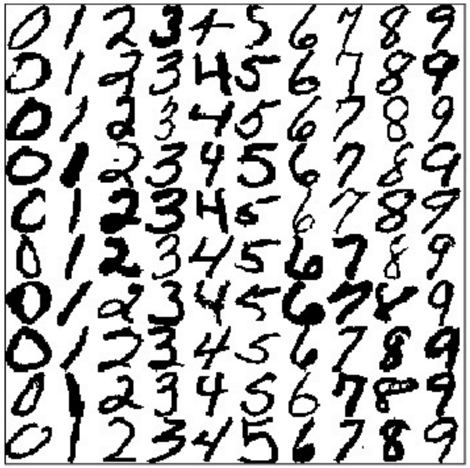
Lecture & Tutorial Format

- Lecture (Zoom)
  - LI 2614 is reserved for our class, so you can use that room during lecture time if you are on campus.
- Tutorial (Zoom)
  - LI 2614 is reserved for our class, so you can use that room during tutorial time if you are on campus.

- NOTE: Later may move to “mixed-mode” depending on the pandemic situation.

What is Machine Learning?

- Arthur Samuel, 1959
  - *Machine Learning*: field of study that gives computers the ability to learn without being explicitly programmed.
    - e.g. computer learns to play checkers by playing against itself.
- There are many applications that are difficult to program by hand.
  - Example:
    - handwritten digit recognition
    - 28x28 image → 784-dim vector
    - a lot of variations & permutations
    - difficult to identify rules & code by hand
  - ML solution:
    - gather some examples
    - train computer to discover differences automatically

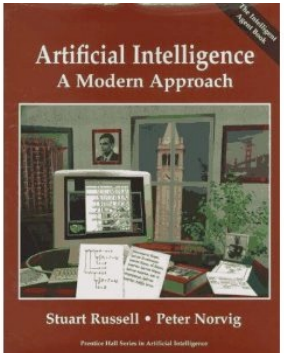


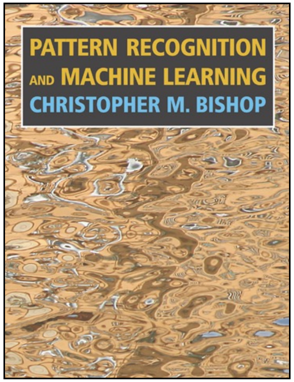
What is Machine Learning?

- Tom Mitchell, 1997
  - *Well-posed Learning Problem*: "A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*."
    - e.g., the computer gets better at recognizing digits as it sees more examples, as measured by the error rate.
- A closer look…
  - *"class of tasks T"*
    - learning is task-specific (recognition, grouping/clustering, …)
  - *"performance measure P"*
    - optimize a loss function (e.g., error rate), but also prevent overfitting (regularization). "generalization"
  - *"experience E"*
    - data-driven! more data is better!

ML vs AI

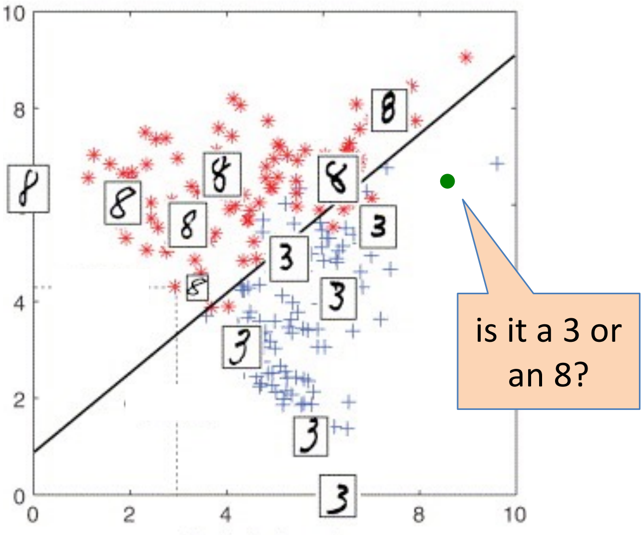
- Machine learning grew out of early work in AI
  - and other fields: statistics, physics, neuroscience, …
  - fueled by more powerful computers and more data.
- "Traditional" Artificial Intelligence (Russell-Norvig)
  - Turing test (is it a computer or a human?)
  - solving by searching (A*, α-β pruning, game playing)
  - knowledge-based (representation, reasoning, logic)
  - planning, scheduling
  - natural language processing
- Machine Learning (Bishop)
  - probability, statistics, Bayesian formulation
  - regression, classification
  - neural networks, kernel functions, kernel machines
  - graphical models, approximate inference
  - statistical learning theory
  - deep learning (deep neural networks)

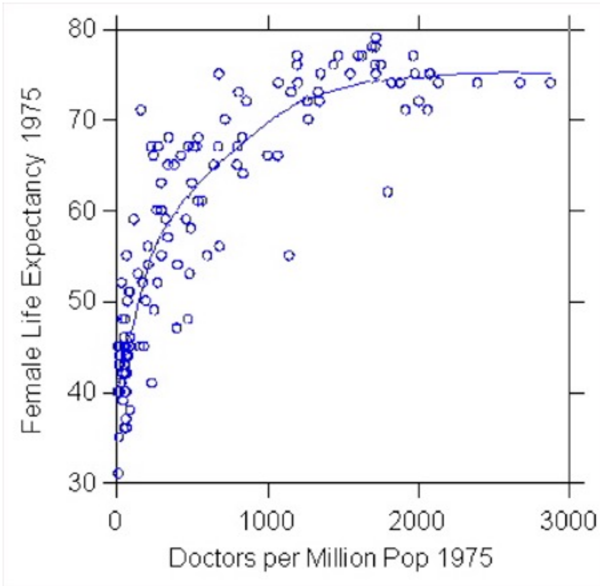




Topics in Machine Learning

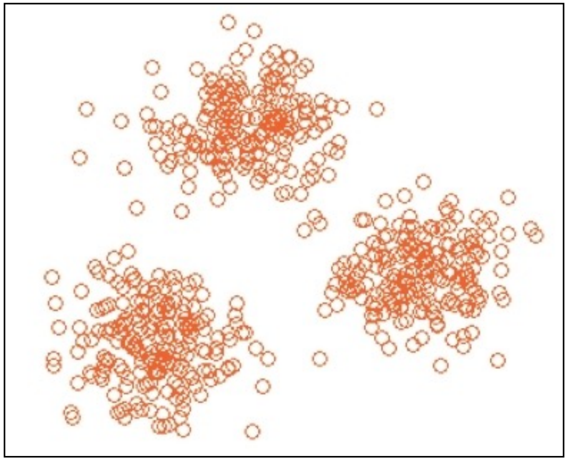
- *Supervised Learning*
  - training data has inputs and outputs
    - e.g., digit recognition (input=image, output=digit)
  - learn a function mapping inputs to outputs
    - classification & regression

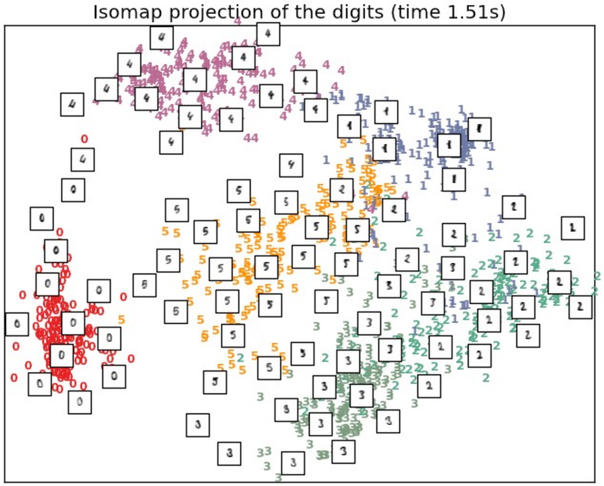




Topics in Machine Learning

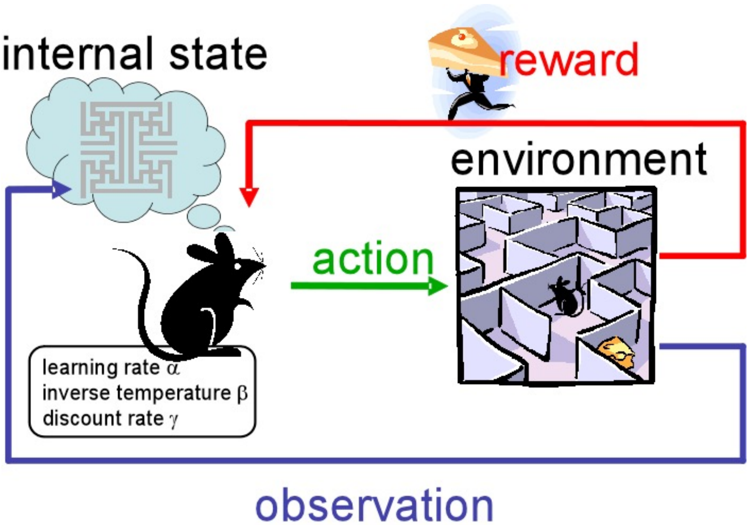
- *Unsupervised Learning*
  - training data only has inputs (no outputs)
    - e.g., collection of web documents
  - density estimation - learn a distribution over input space.
  - clustering - discover groups of similar examples.
  - visualization - project high-dim data to 2 or 3-dimensions.





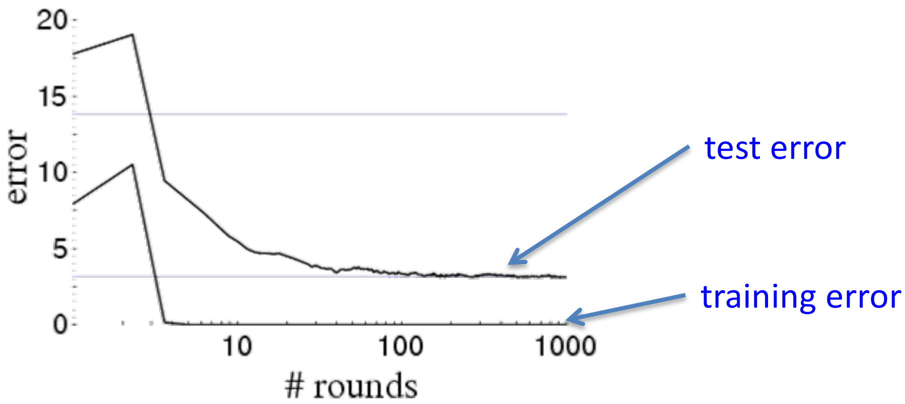
Topics in Machine Learning

- *Reinforcement Learning*
  – make a sequence of actions, given current states
    - e.g. a robot interacting with its environment
  – maximize the reward
    - at some point, receive a reward or a punishment.
    - actions may also affect future reward.



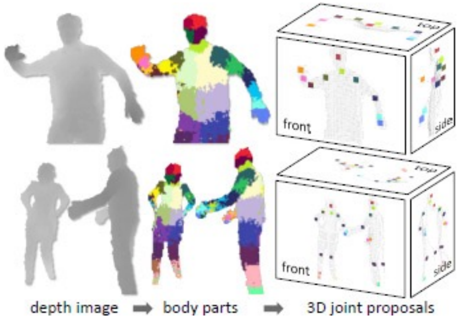
Topics in Machine Learning

- *Learning Theory*
  - why does machine learning work?
    - performance guarantees – bounds on the expected test error.
  - what types of functions can be represent by an algorithm, and how much data do we need?
    - empirical risk minimization, VC dimension, PAC learning



ML in the Real World

- Google
  - spam email classifier, speech recognition, machine translation, quick links, AlphaGo
- Face Detection/Recognition
  - digital cameras, Google street view, Facebook
- Business
  - credit card fraud detection
  - stock trading (portfolio optimization)
- Recommendation systems
  - Netflix, Amazon
- Analysis of Documents
  - clustering, discovery, search
- Digit recognition
  - USPS Zipcodes, check amounts
- Human pose recognition (Kinect)
- Controllers (reinforcement learning)
- Self-driving cars



Building Blocks for ML

- What tools do we need?
  - *linear algebra*
    - matrices, inverse, eigenvector, SVD, …
  - *calculus*
    - integrals, derivatives, …
  - *probability & statistics*
    - random variables, expectation, Bayes' theorem, …
  - *information theory*
    - entropy, KL divergence, mutual information
  - *optimization theory*
    - Lagrange multipliers, duality, KKT condition

(don't worry, we will review these concepts when necessary…)

(also, more resources are on the course website)

https://xkcd.com/1838/

Course Intended Learning Outcomes (CILOs)

1. Identify and explain common machine learning algorithms;
2. Implement machine learning algorithms and apply them to solve real-world problems;
3. Analyze and evaluate the effectiveness of different machine learning algorithms, and assess their relative merits;
4. Design and create new machine learning algorithms to address algorithmic shortcomings and solve particular problems.

Teaching and Learning Activities (TLAs)

- *Lecture* (2 hrs)
  - lecture on the whiteboard (iPad)…there are **no** ppt slides!
  - *Why?*
    - slows down the pace; see the material 3 times.
    - think critically (e.g., teach *how* to derive)
    - active vs. passive learning
  - *What if I make a mistake copying down my notes?*
    - part of being a postgrad student is being able to catch your own mistakes (being self-critical).
    - check reference textbooks, or discuss with classmates.
- *Course Packet*
  - problems on analyzing and deriving ML algorithms.
  - (also a lot of useful properties/models that are not covered in lecture)
- *Tutorial* (1 hr)
  - work on selected problems from the course packet.

Teaching and Learning Activities (TLAs)

- *Assignments*
  - theory: selected problems from the course packet.
  - programming: implementing ML algorithms and try on small datasets.
- *Course Project*
  - use machine learning to solve a real-world problem.
    - e.g., something related to your research.
    - if you don't have a research topic, there will be a recommended course project (e.g., digit recognition).
  - course report and poster presentation (or spotlight videos).
  - **groups of 2**.

ICAP Theory of Cognitive Engagement

- *Which type of engagement is best for learning?*
  - **Interactive > Constructive > Active > Passive**

	Interactive	Constructive	Active	Passive
Behavior	Collaborate	Generate	Manipulate	Attend
Description	collaborating with a peer, building knowledge co-constructively	producing additional small pieces of knowledge beyond what was presented in instructional material	manipulating instruction material (taking notes, highlighting), but not adding new information	paying attention
CS5487 activity	Course project	Course packet (assignments)	Lecture, tutorial	

  - **Paper:** M. Chi, et al. "Translating the ICAP Theory of Cognitive Engagement Into Practice", *Cognitive Science*, 2018.
  - **Talk:** https://youtu.be/JLpZJCBj2qY

Assessment

- Assignments (30%)
  - *theory* (10%) – 1-2 problems per week
  - *programming* (20%) – 2 programming assignments
- Midterm (10%)
- Course Project (30%)
- Final Exam (30%)
  - 2-hour final examination
  - **exam problems are selected from the course packet**.
  - you get an A4 cheat sheet (both sides, *handwritten by pen or pencil*)
- Requirement:
  - To pass course, need at least 30% on the exam and 30% on the project.
  - To get an A…

Sitting in …

- *Can I just attend the lectures without registering?*
  - The answer is **no**.
- Here are the reasons:
  1. The assignments are also teaching activities that give you hands-on experience (both in theory and programming).
     - If you are going to do these, you might as well do it for credit.
  2. Free to do whatever you want for the course project – apply ML to your research problem / ideas.
     - A good course project could lead to a publication.
  3. Machine Learning is an important topic in CS and increasingly important in other fields.
     - It looks good on your transcript/CV.
- *I can offer exceptions to undergraduate students on a case-by-case basis.*

CS5487 Schedule (2022B)

Wk (M/DD)	Lecture	Topics	Reference	Assessment
1. (1/11)	1. Introduction & Review	Probability & Statistics, Linear Algebra	PRML 1.2, 2, Appendix C	
2. (1/18)	2. Parameter Estimation	MLE, bias/variance, LS regression	PRML 2.1-2.4, 3.1; DHS 3.2	
3. (1/25)	3. Bayesian Estimation	MAP, Bayes estimation, Gaussians	PRML 2.3, 3.1, 3.3; DHS 3.3-3.5	PA1 out
4. (2/8)	4. Parametric Clustering	K-means, GMM & EM	PRML 9; DHS 10.1-10.4	
5. (2/15)	5. Non-parametric Clustering	KDE & mean-shift	PRML 2.5; DHS 4.1-4.4; m-s paper	
6. (2/22)	6. Bayesian classifiers	Bayesian Decision Theory, Gaussian classifier, Naïve Bayes	PRML 1.5; DHS 2.1-2.6	PA1 due, PA2 out
7. (3/1)	*Midterm Quiz*			
8. (3/8)	7. Dimensionality & Linear Dimensionality Reduction	PCA, LSA, pPCA, FA, CCA	PRML 12-12.2, 12.4; DHS 3.7-3.8	
9. (3/15)	8. Discriminative Learning – Linear Classifers (I)	least-squares classification, perceptron, logistic regression, empirical risk minimization	PRML 4.1-4.3	PA2 due, Project out
10. (3/22)	9. Discriminative Learning – Linear Classifers (II)	linear SVM, regularized risk	PRML 7.1; DHS 5.11	Project proposal due
11. (3/29)	10. Kernels	kernel functions, kernel SVM	PRML 6.1-6.2, 6.4, 7.1	
12. (4/5)	*No class*	*Ching Ming Festival*		
13. (4/12)	11. Non-linear Dim. Reduction	kernel PCA, pre-image problem	PRML 12.3	
14. (4/22)		Course Project Presentations		Project report due

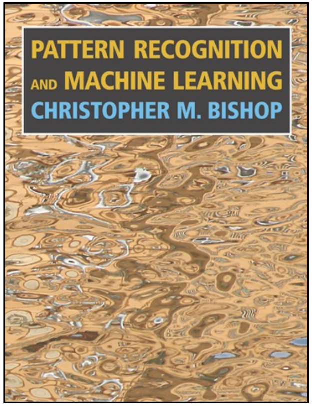
PRML = Bishop, "Pattern Recognition and Machine Learning.
DHS = Duda, Hart, & Stork, "Pattern Classification".

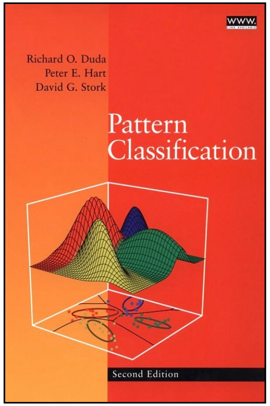
About Due Dates

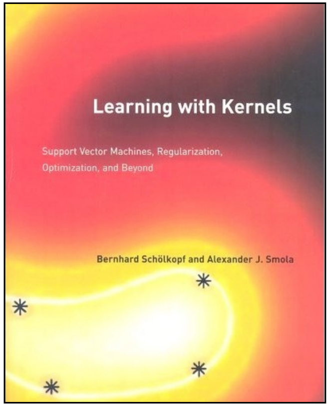
- Note: all due dates are "soft", in the sense that I won't take points off if you are late during the semester.
- On the other hand, you should try to do the assigned HW problems week-by-week, as this will help you understand the concepts after you have seen them in class.
- The programming assignment should also be done in a reasonable time frame, since they will give you some hands-on experience using the algorithms, and may give you some ideas for your course project later.
- The **hard** deadline for **all** course work (problem sets, programming assignments, course project) is ***Week 14, Friday, Apr 22 @ 5pm.***
  - I will handout solutions to the problem sets *after* this deadline.
  - ***Any work submitted after this deadline will not be graded.***

Learning Resources

- Course website on Canvas – CS5487
  - announcements, course packet, discussion board
- Textbooks
  - **Bishop, *Pattern Recognition and Machine Learning*.**
  - Duda, Hart, & Stork, *Pattern Classification*.
  - Schölkopf and Smola, *Learning with kernels: support vector machines, regularization, optimization, and Beyond*.







Important Publication Venues

- *Conferences*
  - Neural Information Processing Systems (NeurIPS)
  - International Conf. on Machine Learning (ICML)
  - International Conf. on Learning Representations (ICLR)
- *Journals*
  - IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)
  - Journal of Machine Learning Research (JMLR)
  - IEEE Trans. Neural Networks and Learning Systems
    - (formally, IEEE Trans. Neural Networks)

Computer Resources

- Programming Languages for Machine Learning
  - **Python (numpy, scipy, sklearn; installed in CS lab)**
    - CS Jupyter hub server; use EID to login.
      - https://ltjh.cs.cityu.edu.hk/
  - **MATLAB (university site license)**
  - Octave (open-source clone of MATLAB)
  - C++ (OpenCV library)
- Tutorials
  - Python:
    - https://docs.scipy.org/doc/numpy/user/quickstart.html
    - http://scikit-learn.org/stable/tutorial/index.html
  - Matlab:
    - http://www.mathworks.com/academia/student_center/tutorials/launchpad.html
    - http://www.stanford.edu/~wfsharpe/mia/mat/mia_mat3.htm
    - http://www.math.mtu.edu/~msgocken/intro/intro.html
  - Octave: http://www.gnu.org/software/octave/

Academic Honesty

- CityU has *Rules of Academic Honesty* and has required all students to complete an online tutorial on subject and declare your understanding
- Plagiarism…
  - It is serious fraud to plagiarize others' work.
  - Punishment ranges from warning to course failure.
- How to prevent plagiarism...
  - Finish the assignments by yourself! You have to write the program/solution yourself.
    - okay to talk about *how* to do the problem with your classmates.
    - **Protect your code**; don't give it away as a "reference" copy.
  - In plagiarism cases, we treat the giver and the copier as both guilty.
  - You hurt your own grades by not reporting cheating.
- As instructors...
  - We have responsibility to report academic dishonesty cases so as not to compromise the quality of education
  - We take suspected plagiarism cases very seriously.

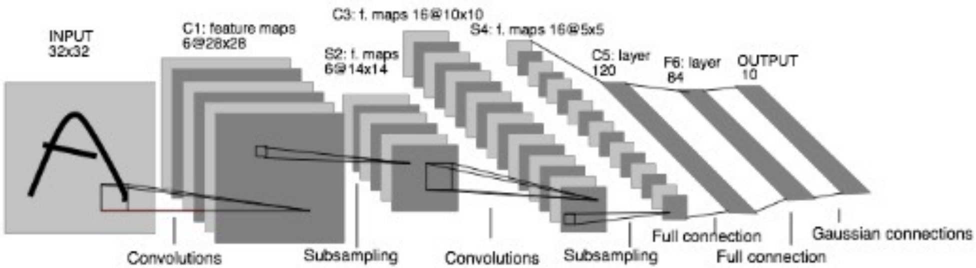
Programming

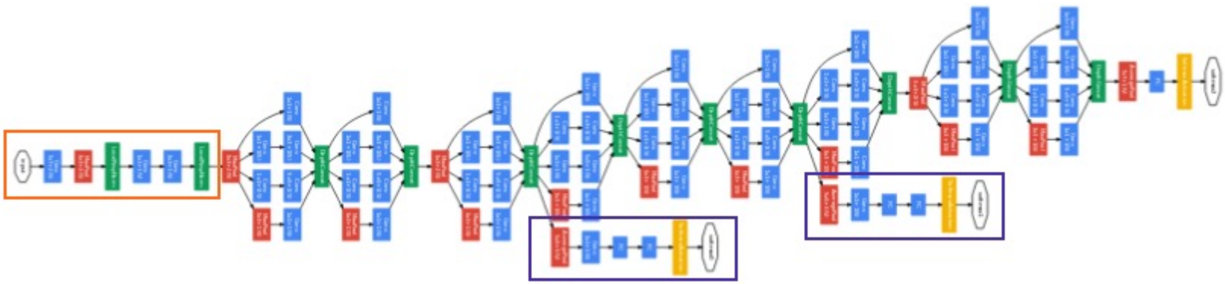
- For the programming assignments
  - *You should implement the algorithms using your own code.*
    - Do **NOT** use someone else's implementation (including libraries)
    - You can use common library components and functions (e.g., matrix operations, optimization toolboxes, etc).
    - If you use anything special, make sure to mention it in your report.
    - Part of this course is about how to implement ML algorithms.
  - If you **DO** use someone else's implementation, you should mention it in your report.
    - 10-20% of the marks for the programming assignment are allocated for implementation. If you use a 3rd party implementation, you will not get these marks.

- For the course projects
  - You may use 3rd party libraries. You must acknowledge the libraries that you use in your project report.

Neural Networks & Deep Learning

- *Deep learning* is supervised/unsupervised learning using multi-layer neural networks.
  - Improved training algorithms to prevent overfitting
  - Faster parallelization (GPUs)
  - More data (millions of examples)



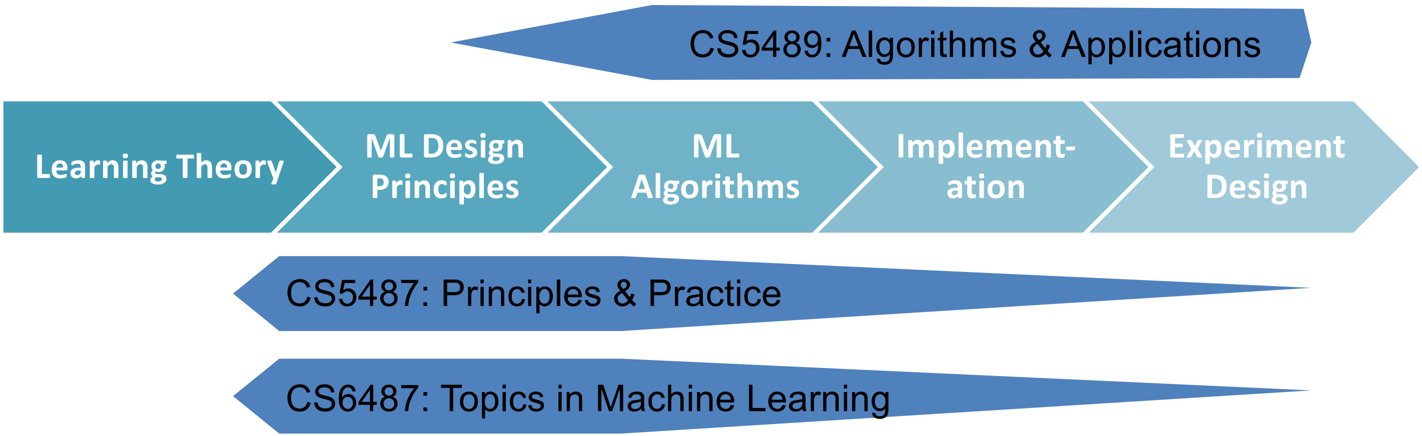


Related Courses

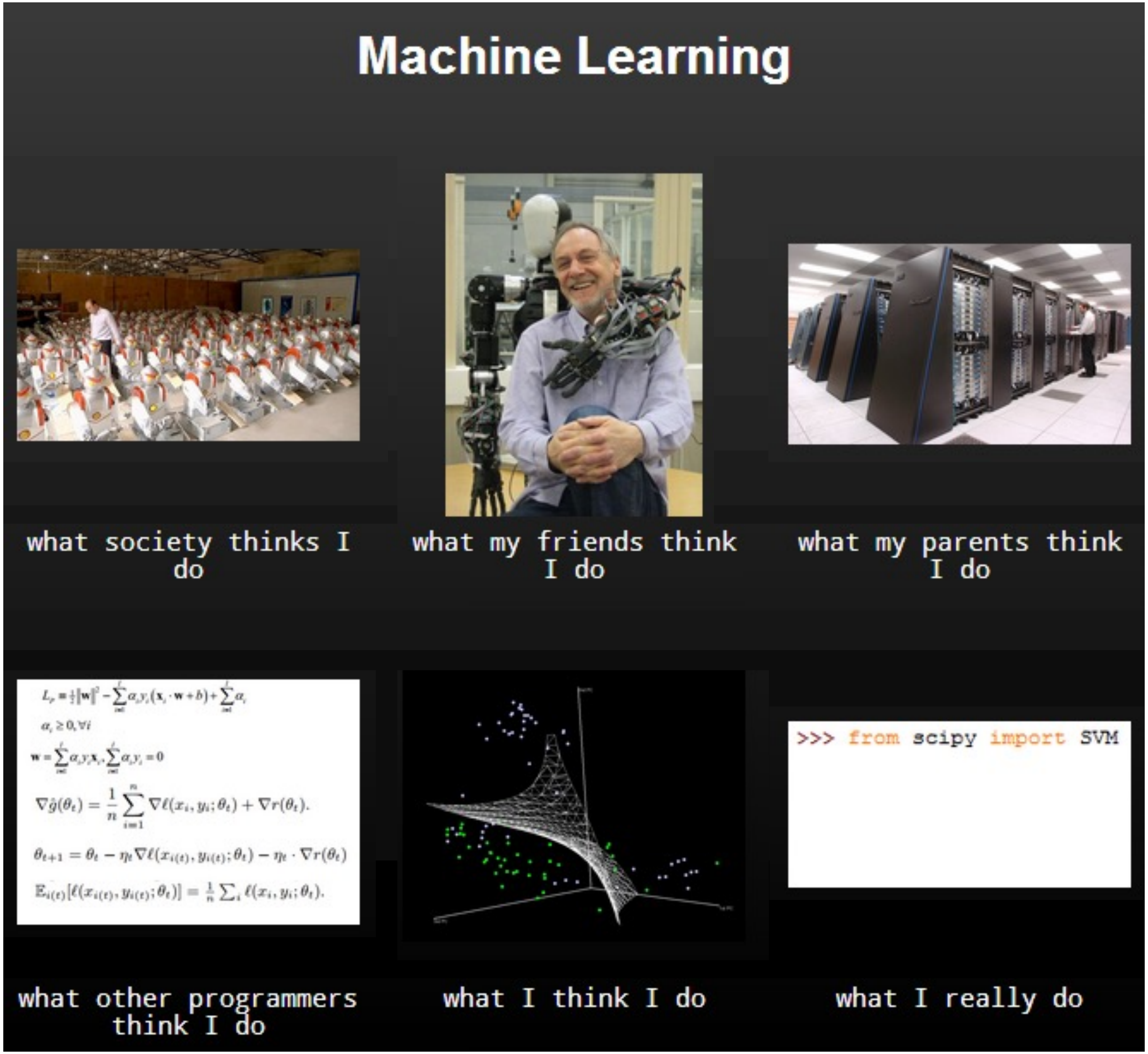
- CS5489 Machine Learning: Algorithms & Applications
  - Taught by me
  - Topics: machine learning applications, deep learning, experiment design.
- CS6487 Topics in Machine Learning (not offered)
  - Topics: varies by lecturer
- CS5486 Intelligent Systems
  - Topics: knowledge systems, fuzzy systems, rule-based systems, evolutionary computation.
- CS5491 Artificial Intelligence
  - Topics: Knowledge representation, uncertainty reasoning, reinforcement learning, AI-based search.

Relationship with other ML courses

- CS5489 mainly focuses on the intuition of how machine learning algorithms work, implementation of algorithms, applying machine learning to applications, and analyzing the results.



- CS5487/CS6487 mainly focus on ML design principles and derivation of algorithms.


Machine Learning
what society thinks I do
what my friends think I do
what my parents think I do
what other programmers think I do
what I think I do
>>> from scipy import SVM
what I really do