Machine Learning Theory (CS 6783)

Tu-Th 1:00 to 2:15 PM
Uris Library, 2B02
Online: Zoom link on course website
https://www.cs.cornell.edu/courses/cs6783/2021fa

Instructor : Karthik Sridharan

About the course

- No exams !
- 4 assignments that count towards your grades (40%)
- One term project (60%)

Assignments

- You are allowed a total of 7 days of late submission (across the 4 assignments)
- Assignments submissions via cms in pdf format

Term Project

- One research style project
- Literature survey sue mid semester (will announce date later)
- Project report due at the end
- A short presentation for fun towards end of semester
- Projects can be done in group sizes of at most 2

Pre-requisites

- This is a theory course and will be heavy on math!
- Basic probability theory
- Basics of algorithms and analysis
- Introductory level machine learning course
- *Mathematical maturity, comfortable reading/writing formal mathematical proofs.*

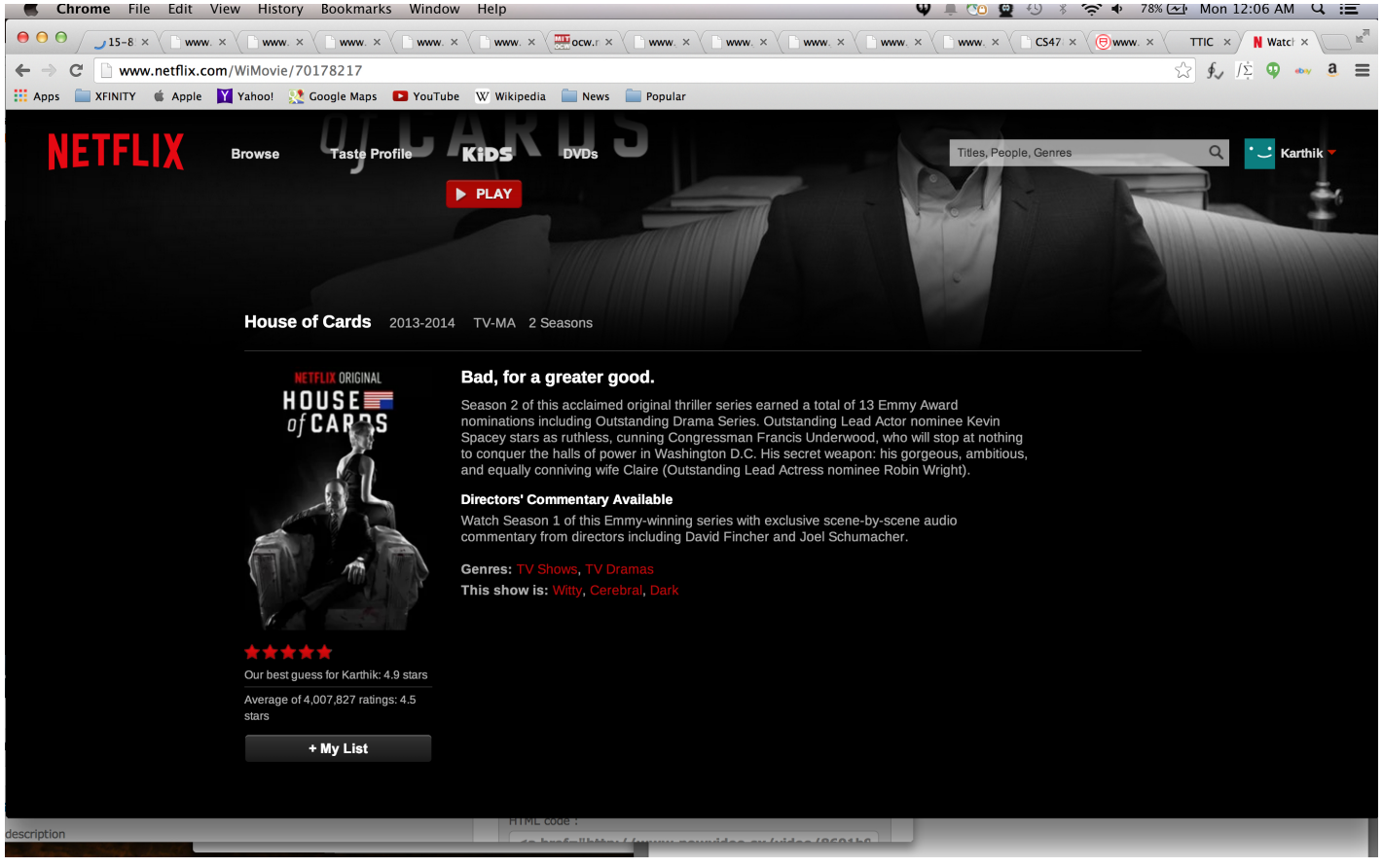
Lets get started . . .

What is Machine Learning

Use past observations to automatically learn to make better predictions/decisions in the future.

WHERE IS IT USED ?

Recommendation Systems



Where is it used ?

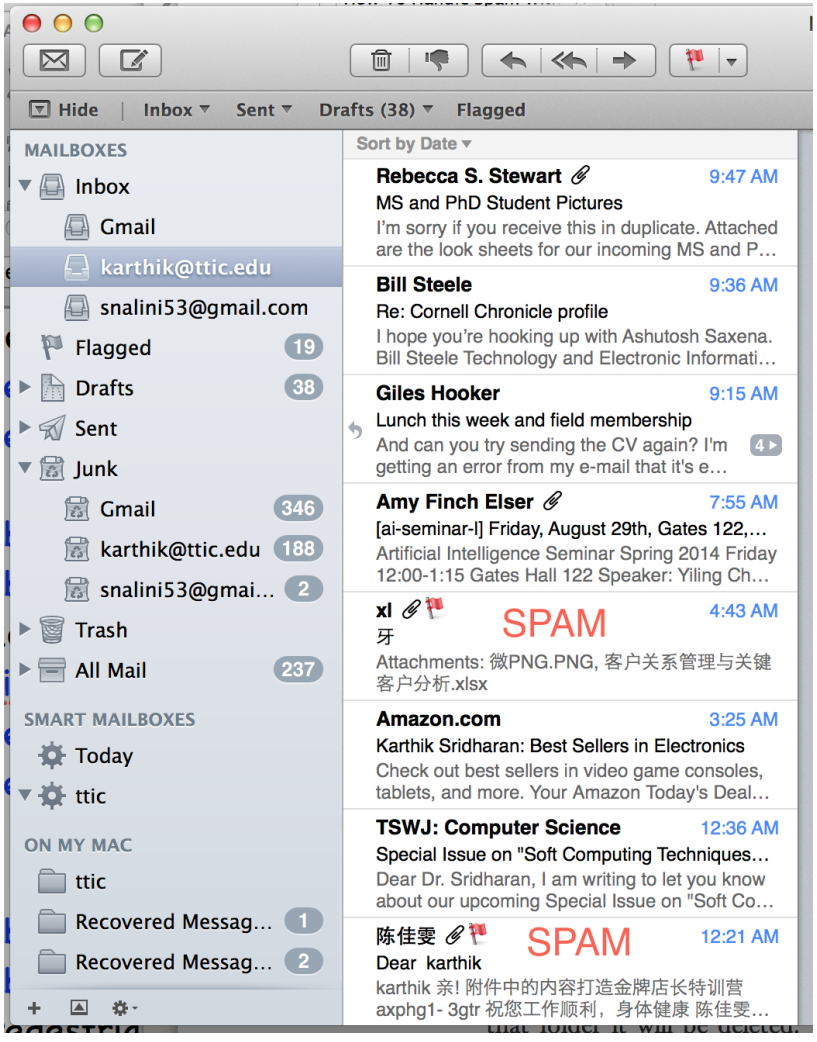
Pedestrian Detection

Where is it used ?

Market Predictions

WHERE IS IT USED ?

Spam Classification



Where is it used ?

- Online advertising (improving click through rates)
- Climate/weather prediction
- Text categorization
- Unsupervised clustering (of articles . . .)
- . . .

What is Learning Theory

- "Develops a rigorous and quantitative understanding of machine learning problems"
  - Formalize the machine learning problem setup
  - Understand fundamental limitations: how well can we hope to learn, how many samples do we need, how do we collect them, how computationally efficient can we be
  - Develop algorithm design tools that under the said formalism provides "provable guarantees" on performance
- Understand and analyze existing approaches mathematically, when do they succeed and when do they fail, when should you use which algorithm

What is Machine Learning Theory

- How do we formalize machine learning problems
- Right framework for right problems (Eg. online , statistical)
- How do we pick the right model to use and what are the tradeoffs between various models
- How many instances do we need to see to learn to given accuracy
- How do we design learning algorithms with provable guarantees on performance
- *Computational learning theory : which problems are efficiently learnable*

Formalizing Learning Problems

- How is data generated ?
- How do we measure performance or success ?
- Where do we place our prior assumption or model assumptions ?

Formalizing Learning Problems

- How is data generated ?
- How do we measure performance or success ?
- Where do we place our prior assumption or model assumptions ?
- *What we observe ?*

Outline of Topics

- Learning problem and frameworks, settings, minimax rates
- Statistical learning theory
  - Probably Approximately Correct (PAC) and Agnostic PAC frameworks
  - Empirical Risk Minimization, Uniform convergence, Empirical process theory
  - Bound on learning rates: MDL bounds, PAC Bayes theorem, Rademacher complexity, VC dimension, covering numbers, fat-shattering dimension
  - Supervised learning : necessary and sufficient conditions for learnability
- Online learning theory
  - Sequential minimax and value of online learning game
  - Regret bounds: Sequential Rademacher complexity, Littlestone dimension, sequential covering numbers, sequential fat-shattering dimension
  - Online supervised learning : necessary & sufficient conditions for learnability
- Algorithms for online convex optimization: Exponential weights algorithm, strong convexity, exp-concavity and rates, Online mirror descent
- Deriving generic learning algorithms : relaxations, random play-outs
- If time permits, uses of learning theory results in optimization, approximation algorithms, perhaps a bit of bandits, ...

Three Scenarios

1. Recognizing which species of animal is it from images (or gambling at a roulette table)
2. Investing a penny a day based on stock market expert advice
3. Investing a pot of gold (a fixed amount) for n days based on stock market expert advice

How do we formalize each one?

Scenario I: Statistical Learning

Scenario II: Online Learning

Scenario III: Online Learning with States (Policy regret)

Learning Problem : Basic Notation

- Input space/ feature space : $\mathcal{X}$

  (Eg. bag-of-words, n-grams, vector of grey-scale values, user-movie pair to rate)

- Output space/ label space $\mathcal{Y}$

  (Eg. $\{\pm 1\}$, $[K]$, $\mathbb{R}$-valued output, structured output)

- Loss function : $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$

  (Eg. $0-1$ loss $\ell(y', y) = \mathbf{1}\{y' \neq y\}$, sq-loss $\ell(y', y) = (y - y')^2$), absolute loss $\ell(y', y) = |y - y'|$

  Measures performance/cost per instance (inaccuracy of prediction/ cost of decision).

- Model class/Hypothesis class $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$

  (Eg. $\mathcal{F} = \{x \mapsto f^\top x : \|f\|_2 \leq 1\}$, $\mathcal{F} = \{x \mapsto \text{sign}(f^\top x)\}$)

A First Example: Online Binary Classification

$\mathcal{Y} = \{\pm 1\}$, $\ell(y', y) = \mathbf{1}\{y' \neq y\}$, $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ is a finite set of models

Scenario One: IID Samples

- At round t learner receives sample $x_t \sim \mathbf{D}_X$ (drawn iid from fixed distribution)
- Learner makes prediction $\hat{y}_t \in \{\pm 1\}$
- $y_t = f^*(x_t)$ is revealed where $f^* \in \mathcal{F}$ is unknown to learner

What is our strategy? How well does it do, how many mistakes?

Scenario One: IID Samples

Scenario Two: Binary Classification Arbitrary Sequence

- At round t learner receives instance $x_t \in \mathcal{X}$ chosen arbitrarily
- Learner makes prediction $\hat{y}_t \in \{\pm 1\}$
- $y_t = f^*(x_t)$ is revealed where $f^* \in \mathcal{F}$ is unknown to learner

What is our strategy? How well does it do, how many mistakes?

Scenario Two: Binary Classification Arbitrary

Sneek Peek

- Different learning frameworks (more formally)
- No Free Lunch Theorems
- Minimax rates for various setting/problems
- Comparing the various settings