# Fall 2019 Computer Science W4771
## MACHINE LEARNING

<table>
<thead>
<tr>
<th>Section</th>
<th>Call Number: 35934 Points: 3</th>
<th>View in Vergil</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td><strong>Day/Time:</strong> MW 2:40pm-3:55pm</td>
<td><strong>Location:</strong> 501 Northwest Corner Building</td>
</tr>
<tr>
<td></td>
<td><strong>Enrollment:</strong> 157 students (164 max) as of September 3, 2019</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Instructor:</strong> Daniel Hsu</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Call Number: 17928 Points: 3</td>
<td>View in Vergil</td>
</tr>
<tr>
<td>H02</td>
<td><strong>Enrollment:</strong> 56 students (100 max) as of September 3, 2019</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Instructor:</strong> Kriste Krstovski</td>
<td></td>
</tr>
</tbody>
</table>
Overview

- What is machine learning?
- Basic topics/challenges in machine learning
Applications I

- Image classification: Predict bird species depicted in image

![Image of a blue bird](image-url)
Recommender systems: Predict how user would rate a movie (Koren, Bell, and Volinsky, 2009)
Applications III

- Machine translation: Predict French translation of English sentence
  (Google translate)

Show us the documents

Montrez-nous les documents
Applications IV

- Chess: Predict win probability of a move in given game state (AlphaZero)
Work in ML

- **Applied ML**
  - Collect/prepare data, build/train models, analyze errors
- **ML developer**
  - Implement ML algorithms and infrastructure
- **ML research**
  - Design/analyze models and algorithms
Mathematical and computational prerequisites

- **Math**
  - Linear algebra, probability, multivariable calculus, reading and writing proofs
- **Software/programming**
  - Much ML work is implemented in python with libraries such as numpy and pytorch

Q: Who has learned about the Singular Value Decomposition?
Basic setting: supervised learning

- Training data: dataset comprised of \textit{labeled examples}
  - \textit{Labeled example}: a pair (input, label)
- Goal: learn function to predict label from input for new examples

Figure 1: Schematic for supervised learning
Examples of functions I

- Decision tree

1: if age $\geq$ 40 then
2:   if genre = western then
3:     return 4.3
4:   else if release date $> 1998$ then
5:     return 2.5
6:   else
7:     :
8:   end if
9: else if $\cdots$ then
10: :
11: end if
Examples of functions II

- Linear classifier

1: if $0.335 \cdot x_1 + 2.5 \cdot x_2 + \cdots + 6.35 \cdot x_{10^6} > 4.3$ then
2: return spam
3: else
4: return not spam
5: end if
Examples of functions III

- Neural network

input  hidden units  output
Types of prediction problems

- **Binary classification**
  - Given an email, is it spam or not?
  - (What’s the probability that it is spam?)

- **Multi-class classification**
  - Given an image, what animal is depicted?
  - (Or which animals are depicted?)

- **Regression**
  - Given clinical measurements, what is level of tumor antigens?
  - (In absolute level? Log-scale?)

- **Structured output prediction**
  - Given a sentence, what is its grammatical parse tree?
  - (Or dependency tree?)

- ...
Beyond supervised learning

- Unsupervised learning / probabilistic modeling
- Online learning
- Reinforcement learning
Challenges in supervised learning

- Might not have the right data
- Might pick a bad model
- Might not fit training data well (under-fitting)
- Might fit the training data too well (over-fitting)
- Training data could be noisy / corrupted (robustness)
- ...
Example: over-fitting

- Which polynomial degree to use?
- Truth: $y = 0 \cdot x + \text{noise}$
- Red points: training data
- Green points: unseen data

\[ \hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_2 = 0.00690718, R_2 = 0.00870077 \]
Example: over-fitting

- Which polynomial degree to use?
- Truth: $y = 0 \cdot x + \text{noise}$
- Red points: training data
- Green points: unseen data
Example: over-fitting

- Which polynomial degree to use?
- Truth: $y = 0 \cdot x + \text{noise}$
- Red points: training data
- Green points: unseen data

$\hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_4 = 0.0062397, R_4 = 0.013063$
Example: over-fitting

- Which polynomial degree to use?
- Truth: $y = 0 \cdot x + \text{noise}$
- Red points: training data
- Green points: unseen data

\[ \hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_5 = 0.00582684, R_5 = 0.00975194 \]
Example: over-fitting

- Which polynomial degree to use?
- Truth: \( y = 0 \cdot x + \text{noise} \)
- Red points: training data
- Green points: unseen data

\[ \hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_6 = 0.00571136, R_6 = 0.0142185 \]
Example: over-fitting

- Which polynomial degree to use?
- Truth: \( y = 0 \cdot x + \text{noise} \)
- Red points: training data
- Green points: unseen data
Example: over-fitting

- Which polynomial degree to use?
- Truth: $y = 0 \cdot x + \text{noise}$
- Red points: training data
- Green points: unseen data

$\hat{R}_1 = 0.00692304, R_1 = 0.00897457, \hat{R}_8 = 0.00564127, R_8 = 0.0440347$
Example: over-fitting

- Which polynomial degree to use?
- Truth: $y = 0 \cdot x + \text{noise}$
- Red points: training data
- Green points: unseen data

$\hat{R}_1 = 0.00692304$, $R_1 = 0.00897457$, $\hat{R}_9 = 0.00541878$, $R_9 = 0.42463$
Example: the right data

- Given a college applicant, will they graduate if admitted?
- What is appropriate training data?
  - input = past applicant; label = admitted or not
  - input = past admit; label = graduated or not
  - input = past applicant; label = graduated or not
Overview of the rest of the course

- Non-parametric methods
  - Simple and flexible methods for prediction
- Prediction theory
  - Statistical model for studying prediction problems
- Regression
  - Models and methods for predicting real-valued outcomes
  - Inductive bias, features, kernels
- Classification
  - Models and methods for predicting discrete-valued outcomes
  - Surrogate losses, margins, cost-sensitive risk, fairness, ensemble methods
- Optimization
  - Convex optimization and neural network training
- Unsupervised learning
  - Methods for clustering and matrix approximation
What to expect in the class

1. **How to use sklearn, pytorch, tensorflow, ...**
   [We expect you will be able to learn these software tools on your own!]

2. **Algorithmic and statistical principles**
   - Well-weathered models + methods (e.g., logistic reg., gradient descent)
   - Not the latest nonsense that shows up on arXiv

3. **Programming and proofs**
   - No need to be a professional hacker or mathematician
   - But, need to exercise your critical thinking!
More about the course


2. **Course assistants (CAs):** Achille, Arjun, Dinko, Erica, Miles, …
   - Office hours start *next week*.

3. **Piazza, Courseworks, Gradescope**
   - Use Piazza to ask (and answer!) questions about the course
   - E-mail instructor for administrative issues (e.g., sign drop form)
   - Do not e-mail CAs. All regrade requests handled through Gradescope.

4. **Syllabus:** see website
   - Reading + homework assignments: 34% of grade
   - Exams on October 16 and December 18: 66% of grade

5. **Disability services:** [https://health.columbia.edu/](https://health.columbia.edu/)
   - If needed, please make arrangements with disability services by 9/18.
1. **Cheating**: Don’t do it.
   - See syllabus for policies & consequences
   - If you are unsure if something constitutes cheating, ask the instructor!

2. **Cheating out of desperation** is still cheating.
   - *Instead*: Get help early! Go to office hours, find a tutor, etc.
   - *We are here to help.*
Homework 0

• Due Monday 9/9 at 2:30 PM

• Instructions (also on website):
  • Create an account on Gradescope (if you don’t have one already)
  • Add this course using the Entry Code M3D5EX
  • Use your UNI as your “Student ID”
  • Solve simple problems about mathematical prerequisites

• REQUIRED. Cannot pass the class without completing it.

• Not used to determine enrollment / waiting list status.

• HWx for x>0 will generally not be “multiple choice”.

Questions?