Lecture 1:
A basic introduction to ML
What is Machine learning

Task $T$ → Learning Algorithm → Performance $P$

Experience $E$ (Data)

Machine learning studies algorithms that
- Improve *performance* $P$
- at some *task* $T$
- based on *experience* $E$
Machine learning in Computer Science

• Machine learning is already the preferred approach to
  – Speech recognition, Natural language processing
  – Computer vision
  – Medical outcomes analysis
  – Robot control
  – ...

• This trend is growing
  – Improved machine learning algorithms
  – Increase data capture, and new sensors
  – Increasing demand for self-customization to user and environment
Topics

Machine Learning

- Supervised Learning
- Semi-supervised Learning
- Unsupervised Learning
- Reinforcement Learning
Supervised Learning

- Learn to predict output from input.
- Output can be
  - continuous: regression problems

Example: Predicting the price of a house based on its square footage
Supervised Learning

• Learn to predict output from input.

• Output can be
  – continuous: regression problems
  – Discrete: classification problems

Example: classify a loan applicant as either high risk or low risk based on income and saving amount.
Unsupervised Learning

• Given a collection of examples (objects), discover self-similar groups within the data – clustering

Example: clustering artwork
Unsupervised Learning

• Given a collection of examples (objects), discover self-similar groups within the data – clustering

![Image Segmentation](image.png)
Unsupervised Learning

• Given a collection of examples (objects), discover self-similar groups within the data – **clustering**

• Learn the underlying distribution that generates the data we observe – **density estimation**

• Represent high dimensional data using a low-dimensional representation for compression or visualization – **dimension reduction**
Reinforcement Learning

• Learn to act
• An agent
  – Observes the environment
  – Takes action
  – With each action, receives rewards/punishments
  – Goal: learn a policy that optimizes rewards
• No examples of optimal outputs are given
• Not covered in this class. Take 533 (spring) if you want to learn about this.
When do we need computer to learn?

Do we need learning to do tax return?
Appropriate Applications for Supervised Learning

• Situations where there is no human expert
  – x: bond graph of a new molecule, f(x): predicted binding strength to AIDS protease molecule
  – x: nano modification structure to a Fuel cell, f(x): predicted power output strength by the fuel cell

• Situations where humans can perform the task but can’t describe how they do it
  – x: picture of a hand-written character, f(x): ascii code of the character
  – x: recording of a bird song, f(x): species of the bird

• Situations where the desired function is changing frequently
  – x: description of stock prices and trades for last 10 days, f(x): recommended stock transactions

• Situations where each user needs a customized function $f$
  – x: incoming email message, f(x): importance score for presenting to the user (or deleting without presenting)
Supervised learning (basic setup)

• Given: a set of training examples
  \[(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\]
  – \(x_i\): the input of the \(i\)-th example (\(x_i \in \mathbb{R}^d\), i.e., a vector)
  – \(y_i\) is its corresponding output (continuous or discrete)
  – We assume there is some underlying function \(f\) that maps from \(x\) to \(y\) – our target function

• Goal: find a good approximation of \(f\) so that accurate prediction can be made for previously unseen \(x\)
Example: regression

The underline function:

\[ t = \sin(2\pi x) + \varepsilon \]

where \( \varepsilon \) is Gaussian noise

Given training examples shown as blue circles

Examples are generated based on the green line (the true underlying function)

Learning goal: make accurate predictions of the \( t \) values for some new values of \( x \) (values that are not included in training)
Polynomial curve fitting

- There are infinite functions that will fit the training data perfectly.
- In order to learn, we have to focus on a limited set of possible functions
  - We call this our *hypothesis space*
  - E.g., all M-th order polynomial functions
    \[ y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M \]
  - \( \mathbf{w} = (w_0, w_1, \ldots, w_M) \) represents the unknown parameters that we wish to learn from the training data
- Learning here means to find a good set of parameters \( \mathbf{w} \) to minimize some loss function

  \[ E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} (y(x_n, \mathbf{w}) - t_n)^2 \]

This optimization problem can be solved easily.
We will not focus on solving this at this point, will revisit this later.
Important Issue: Model Selection

- The red line shows the function learned with different M values
- Which M should we choose – this is a model selection problem
- Can we use $E(w)$ that we define in previous slides as a criterion to choose M?

$$E(w) = \frac{1}{2} \sum_{n=1}^{N} (y(x_n, w) - t_n)^2$$

Sum-of-squares error
Over-fitting

• As $M$ increases, loss on the training data decreases monotonically
• However, the loss on test data starts to increase after a while
  • Why? Is this a fluke or generally true?

It turns out this is generally the case – caused by over-fitting
Over-fitting

• Over-fitting refers to the phenomenon when the learner adjusts to some random signals in the training data that is not relevant to the target function

• Real example:
  – In Bug ID project, x: image of a robotically maneuvered bug; f(x): the species of the bug
  – Initial attempt yields close-to-perfect accuracy
  – Reason: different species were imaged in different batches, one species when imaging, had a peculiar air bubble in the image
Overfitting

• Over-fitting happens when
  – There is too little data (or some systematic bias in the data)
  – There are too many parameters
Key Issues in Machine Learning

• What are good hypothesis spaces?
  – Linear functions? Polynomials?
  – which spaces have been useful in practical applications?

• How to select among different hypothesis spaces?
  – The Model selection problem
  – Trade-off between over-fitting and under-fitting

• How can we optimize accuracy on future data points?
  – This is often called the Generalization Error – error on unseen data pts
  – Related to the issue of “overfitting”, i.e., the model fitting to the peculiarities rather than the generalities of the data

• What level of confidence should we have in the results? (A statistical question)
  – How much training data is required to find an accurate hypotheses with high probability? This is the topic of learning theory

• Are some learning problems computationally intractable? (A computational question)
  – Some learning problems are provably hard
  – Heuristic / greedy approaches are often used when this is the case

• How can we formulate application problems as machine learning problems? (the engineering question)