COMPSCI 689
Lecture 1: Course Overview

Brendan O'Connor

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).

Introduction

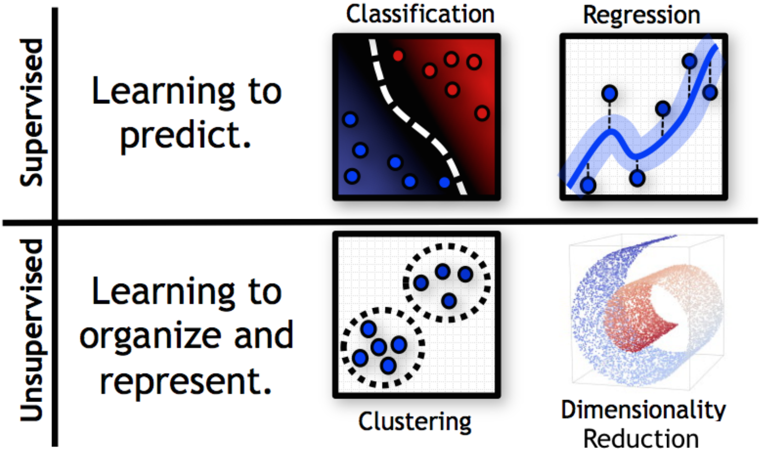
What is Machine Learning?

A definition of machine learning

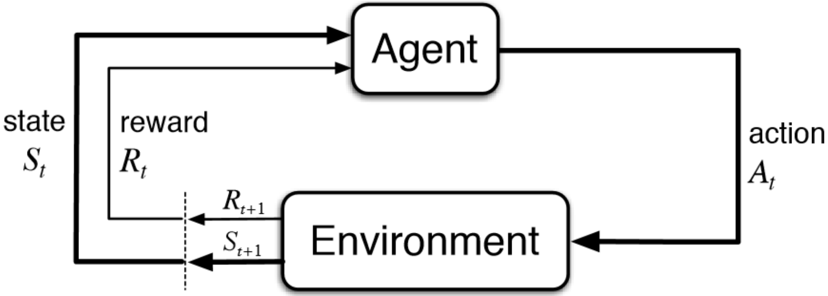
Mitchell (1997): "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Substitute "training data D" for "experience E."

Machine Learning Tasks



Machine Learning Tasks



Machine Learning Applications

Course Goals

The primary goals of this course are to equip students with the mathematical foundations and technical skills needed to participate in machine learning research activities.

Learning Outcomes

1. To be able to correctly define core supervised and unsupervised machine learning tasks.
2. To gain familiarity with optimization-based frameworks for developing machine learning algorithms including exact and iterative non-linear numerical optimization methods.
3. To gain familiarity with select classical machine learning models and algorithms including what task they solve, how they are derived, and what their computational scalability properties are.
4. To be able to develop customized machine learning models and prediction and/or inference methods for solving a specified task within multiple generalized modeling frameworks including neural networks and generalized probabilistic models

Learning Outcomes

5. To be able to explain concepts including generalization, capacity control, overfitting, training error, validation error, test error and to understand when to apply different metrics for evaluating the performance of machine learning methods.
6. To be able to design valid machine learning experiments for assessing model performance including while optimizing model hyper-parameters.
7. To be able to clearly communicate the results of machine learning experiments using appropriately selected and correctly formatted tables and plots and to be able to correctly interpret such output.
8. To gain practical experience developing efficient and numerically stable implementations of learning and inference methods for both classical and customized models.

Learning Outcomes

9. To gain practical experience using current machine learning software and tools based on Python including Numpy, SciPy, and PyTorch and to understand key facilities of these tools such as automatic differentiation.
10. To understand the limitations of optimization-based machine learning.

Logistics

- Course Website: We'll use the Piazza page to update with materials and links. From `https://people.cs.umass.edu/~brenocon/cs689_2023/`
- Lecture Videos: Lecture videos will be recorded on Zoom, and links posted to Piazza.
- Piazza Forum: Online, text-based discussion forum. Ask questions and discuss course material.
- Contacting course staff: Piazza private message (see syllabus).
- Assignment Submission: Gradescope.
- Office hours: Starting next week, planned for Wed 3-4pm.

Background

The course will build on the background material listed below. An undergraduate-level background in these topics should be sufficient. If you need to pick up topics or substantially revise material, it is possible to do so, but expect to need to invest more time in the course.

- Linear Algebra
- Multivariate differential and integral calculus
- Probability and Statistics
- Algorithms and Data Structures
- Programming Languages

See course website for a detailed list of math background topics.

Programming and Computing

- Students need access to computing to complete regular assignments (any moderately recent laptop/desktop should do).
- Programming assignments will use Python 3.8+ and other Python packages (e.g., SciKit-Learn, PyTorch).
- The Anaconda Python 3 distribution is recommended (`https://www.anaconda.com/download/`).

Text Books

The course will use a primary text and other freely available texts as needed. Links are available on Moodle.

- MLPP: *Machine Learning: A probabilistic Perspective*. Murphy. (Primary)
- NO: *Numerical Optimization*. Nocedal and Wright. (Supplemental)
- DL: *Deep Learning*. Goodfellow, Bengio and Courville. (Supplemental)
- CO: Convex Optimization (Supplemental)

Evaluation

- Homework (4): 50%
- Midterm Exam: 25%
- Final Exam: 25%

Homework Assignments

- Completed as individual work.
- Each assignment will consist of a mix of derivations, written problems, implementations in Python and experimentation.
- One week to complete any required derivations and preliminary coding (part 1), followed by a second week to finish implementations and experiments (part 2).
- Deliverables are a written report for parts 1 and 2, and code for part 2 only.

Exams

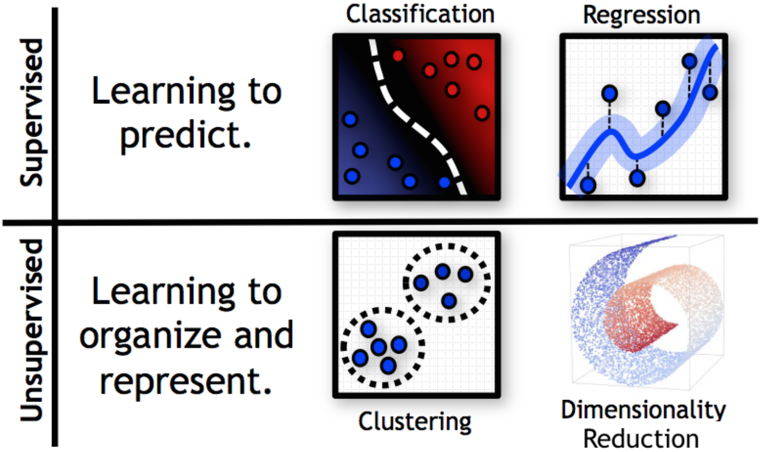
- Evening midterm exam 7-9pm at roughly the mid-semester point covering material to that point.
- Final exam covering material in second half of the course.
- Both exams use one “cheat sheet” page of notes. No electronic devices permitted.

Course Policies

The course syllabus describes course policies in detail including:

- Late homework
- Collaboration
- Academic honesty
- Community code of conduct

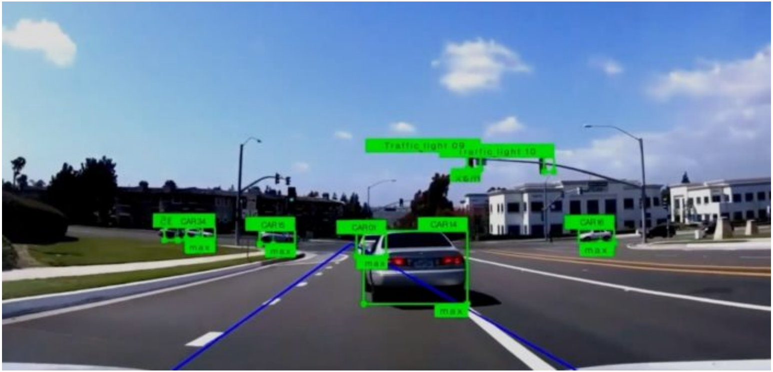
Machine Learning Tasks



Supervised Learning

- In the supervised setting, the data take the form of tuples $(\mathbf{x}, \mathbf{y})$.
- In this setting, the $\mathbf{x}$'s are readily observable, but for some reason, the $\mathbf{y}$'s are not.
- Thus, given the value of $\mathbf{x}$, we would like to *predict* the corresponding value of $\mathbf{y}$.
- To predict the value of $\mathbf{y}$ that corresponds to a given $\mathbf{x}$, we use a *prediction function* $f(\mathbf{x})$.
- The two main types of supervised problems are regression where $\mathbf{y}$ is real-valued, and classification where $\mathbf{y}$ is categorical.

Example: Object Detection



Supervised Learning Notation

- Input Space: $\mathcal{X}$
- Output Space: $\mathcal{Y}$
- Input: $\mathbf{x} \in \mathcal{X}$
- Output: $\mathbf{y} \in \mathcal{Y}$
- Prediction Function: $f : \mathcal{X} \to \mathcal{Y}$

In general, the input and output spaces can be arbitrary sets. In the most common case $\mathcal{X} = \mathbb{R}^D$ and either $\mathcal{Y} = \mathbb{R}^K$ or $\mathcal{Y}$ is a finite set.

Example: Binary Classification

In binary classification, the output y can be one of two possible values. The output y is referred to as the “class" or “label" of $\mathbf{x}$. The inputs $\mathbf{x}$ are also referred to as “features." The goal is to predict the class given the features.

- Input Space: $\mathcal{X}$
- Output Space: $\mathcal{Y} = \{0, 1\}$ or $\{-1, 1\}$
- Input: $\mathbf{x} \in \mathcal{X}$
- Output: $\mathbf{y} \in \mathcal{Y}$
- Prediction Function: $f : \mathcal{X} \to \mathcal{Y}$

Example: Multi-Class Classification

In multi-class classification, the output y can be one of C possible values. The output y is referred to as the “class" or “label" of $\mathbf{x}$. The inputs $\mathbf{x}$ are also referred to as “features." The goal is to predict the class given the features.

- Input Space: $\mathcal{X}$
- Output Space: $\mathcal{Y} = \{1, ..., C\}$
- Input: $\mathbf{x} \in \mathcal{X}$
- Output: $\mathbf{y} \in \mathcal{Y}$
- Prediction Function: $f : \mathcal{X} \to \mathcal{Y}$

Example: Regression

In regression, the output y is a real-valued number also called the "target," "dependent variable," "outcome variable" or "response variable." The $\mathbf{x}$ inputs are also called "features," "independent variables," "predictors" or "covariates."

- Input Space: $\mathcal{X}$
- Output Space: $\mathcal{Y} = \mathbb{R}$
- Input: $\mathbf{x} \in \mathcal{X}$
- Output: $\mathbf{y} \in \mathcal{Y}$
- Prediction Function: $f : \mathcal{X} \to \mathcal{Y}$

The Supervised Learning Problem

The Supervised Learning Problem

Given a *data set* consisting of a collection of input-output tuples $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n) | \mathbf{x}_n \in \mathcal{X}, \mathbf{y}_n \in \mathcal{Y}, 1 \leq n \leq N\}$, select the best prediction function $f : \mathcal{X} \rightarrow \mathcal{Y}$.

Note: A *data set* is not a mathematical set. It is a collection of elements that allows repetition.

Prediction Loss Functions

Prediction Loss Function: A prediction loss function $L\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a real-valued function that is bounded below (typically at 0), and that satisfies $L(\mathbf{y}, \mathbf{y}) \leq L(\mathbf{y}, \mathbf{y}')$ for all $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$.

Examples:

- Squared Loss: $L_{sqr}(\mathbf{y}, \mathbf{y}') = \|\mathbf{y} - \mathbf{y}'\|_2^2 = \sum_{k=1}^{K} (\mathbf{y}_k - \mathbf{y}'_k)^2$
- Absolute Loss: $L_{abs}(\mathbf{y}, \mathbf{y}') = \|\mathbf{y} - \mathbf{y}'\|_1 = \sum_{k=1}^{K} |\mathbf{y}_k - \mathbf{y}'_k|$
- 0/1 Loss: $L_{01}(\mathbf{y}, \mathbf{y}') = [\mathbf{y} \neq \mathbf{y}']$

Given a loss function L, an instance $(\mathbf{x}, \mathbf{y})$, and a prediction function f, we compute the loss of f on $(\mathbf{x}, \mathbf{y})$ as $L(\mathbf{y}, f(\mathbf{x}))$.

Do we now have enough information to select the optimal f given a data set $\mathcal{D}$?