COMPSCI 689
Lecture 1: Course Overview

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

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

Introduction

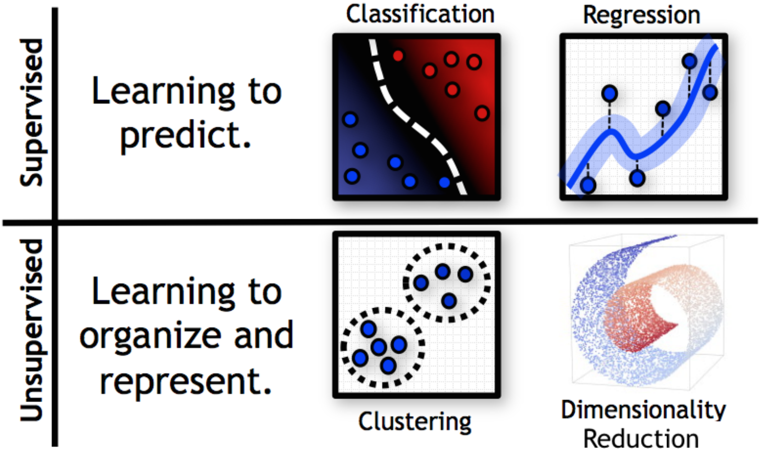
What is Machine Learning?

Views on Machine Learning

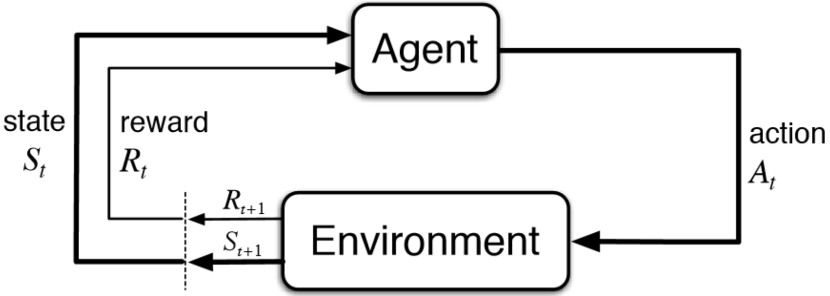
Mitchell (1997): "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

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

Machine Learning Tasks



Machine Learning Tasks



Machine Learning Applications

Machine Learning in Industry

https://nips.cc/Conferences/2016/Sponsors

Relationship to Other Fields

- Machine Learning and Artificial Intelligence
- Machine Learning and Probability/Statistics
- Machine Learning and Numerical Optimization
- Machine Learning and Function Approximation
- Machine Learning and Cognitive Science
- Machine Learning and Neuroscience
- Machine Learning and Data Mining
- Machine Learning and Data Science
- Machine Learning and Big Data

Course Goals

The aim of this course is to provide the foundations necessary to conduct machine learning research including developing novel machine learning models and algorithms and conducting machine learning experiments. The course will cover:

- Mathematical and statistical foundations of learning.
- Select supervised and unsupervised learning models (linear regression, generalized linear models, neural networks, support vector machines, mixture models, PCA).
- Derivation and implementation of optimization-based and Bayesian learning algorithms.

This course may touch briefly on deep learning, reinforcement learning, graphical models, and learning theory, which all have dedicated courses.

Prerequisites

The course will build on the material listed below. All students are expected to be familiar with this material or have the ability to make up any gaps in their backgrounds on their own.

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

See Moodle 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 3.6.
- The Anaconda Python 3.6 distribution is recommended (`https://www.anaconda.com/download/`).

Logistics

Logistics and course details:

- Lectures: Mo/We 2:30-3:45pm
- Instructor Office Hours: Tu 9:30-10:30.
- Course Website: `https://moodle.umass.edu/`
- Discussion Board: `https://piazza.com/umass/fall2019/compsci689/`
- Course e-Mail: Piazza private message
- Course Policies: Refer to the syllabus on Moodle site.

Text Books

The course will use a primary text and other freely available texts as needed:

- 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)

Readings are intended to be completed before class.

Course Evaluation

The course has two evaluation tracks this year: a project-based track and an exam-based track. Students can choose which track they want to complete.

Exam Track

- Assignments (5) 60%
- Quizzes (weekly) 12%
- Final Exam 28%

Project Track

- Assignments (4) 48%
- Quizzes (weekly) 12%
- Project Proposal 5%
- Project Report 35%

Course Projects

- Course projects may be completed individually or in groups of 2 or 3 (max).
- Course projects will focus on deriving, implementing and testing machine learning algorithms.
- **Note: Straight application projects are not allowed.**
- One-page project proposals will be due by the mid-semester date.
- Final project reports will be due on the last day of classes.
- Students must submit a project proposal to be eligible for the project track. Teams can switch from the project track to the exam track at any time before assignment 5 is due.
- Details are available now on Moodle.

Collaboration

Collaboration Policy: Homework assignments are considered individual work. You may discuss the problems with other students; however, to avoid issues with the course's academic honesty policy, you should not take any materials out of such discussions (writing, whiteboard photos, etc.). You should never share your completed or in-progress code or write-up with another student in any form, or request to see another student's code or write-up. Your derivations, code, and write-up must be your own work. Quizzes are strictly individual work. No collaboration of any kind is permitted. Collaboration within your project team is allowed and encouraged for final course projects.

Academic Honesty Policy

Academic Honesty Policy: You are required to list the names of anyone you discuss problems with on the first page of your solution report. Copying any solution materials (derivations, code, method descriptions) from external sources (books, web pages, etc.) or from other students is considered cheating. Sharing your code or solutions with other students is also considered cheating. This includes sharing material with students in future years, or requesting material from students who took the course in previous years. Collaboration indistinguishable from copying will be treated as copying. All instances of suspected cheating will be dealt with through official UMass Amherst Academic Honesty Procedures. Students are expected to be familiar with the relevant policies and procedures: www.umass.edu/honesty.

Supervised Learning Definitions

In supervised learning, our goal is to predict the output $\mathbf{y}$ that corresponds to an input $\mathbf{x}$ using a prediction function $f(\mathbf{x})$. We assume samples $(\mathbf{x}, \mathbf{y})$ are drawn from $P_*(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y})$.

Basic Definitions:

- Input: $\mathbf{x} \in \mathcal{X}$
- Output: $\mathbf{y} \in \mathcal{Y}$
- True Joint Distribution: $P_*(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}) = P_*(\mathbf{x}, \mathbf{y})$
- Prediction Function: $f \colon \mathcal{X} \to \mathcal{Y}$

What are some examples of binary classification problems?

Example: Binary Classification

In binary classification, our goal is to predict the class y that corresponds to an input $\mathbf{x}$ using a prediction function $f(\mathbf{x})$. We assume samples $(\mathbf{x}, y)$ are drawn from $P_*(\mathbf{X} = \mathbf{x}, Y = y)$.

Basic Definitions:

- Input: $\mathbf{x} \in \mathcal{X}$
- Output: $y \in \{0, 1\}$
- True Joint Distribution: $P_*(\mathbf{X} = \mathbf{x}, Y = y) = P_*(\mathbf{x}, y)$
- Prediction Function: $f : \mathcal{X} \to \{0, 1\}$

What are some examples of binary classification problems?

Example: Multi-Class Classification

In multi-Class classification, our goal is to predict the class y that corresponds to an input $\mathbf{x}$ using a prediction function $f(\mathbf{x})$. We assume samples $(\mathbf{x}, y)$ are drawn from $P_*(\mathbf{X} = \mathbf{x}, Y = y)$.

Basic Definitions:

- Input: $\mathbf{x} \in \mathcal{X}$
- Output: $y \in \{1, ..., C\}$
- True Joint Distribution: $P_*(\mathbf{X} = \mathbf{x}, Y = y) = P_*(\mathbf{x}, y)$
- Prediction Function: $f : \mathcal{X} \to \{1, ..., C\}$

What are some examples of multi-class classification problems?

Example: Regression

In regression, our goal is to predict the target value y that corresponds to an input $\mathbf{x}$ using a prediction function $f(\mathbf{x})$. We assume samples $(\mathbf{x}, y)$ are drawn from $P_*(\mathbf{X} = \mathbf{x}, Y = y)$.

Basic Definitions:

- Input: $\mathbf{x} \in \mathcal{X}$
- Output: $y \in \mathbb{R}$
- True Joint Distribution: $P_*(\mathbf{X} = \mathbf{x}, Y = y) = P_*(\mathbf{x}, y)$
- Prediction Function: $f : \mathcal{X} \to \mathbb{R}$

What are some examples of regression problems?

Distributional Supervised Learning Problem

Question

Given the joint distribution $P_*(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y})$, what is the best choice of prediction function f?

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

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

Which of these losses are better suited for classification problems?

Next class: How to identify the optimal f for a given loss function.