

Applied Machine Learning - Welcome

Contents

- [Introduction](#)
- [What's Inside](#)
- [Prerequisites](#)
- [Instructors](#)
- [Table of Contents](#)

Introduction

This course provides an overview of key algorithms and concepts in machine learning, with a focus on applications. Introduces supervised and unsupervised learning, including logistic regression, support vector machines, neural networks, Gaussian mixture models, as well as other methods for classification, regression, clustering, and dimensionality reduction. Covers foundational concepts such as overfitting, regularization, maximum likelihood estimation, generative models, latent variables, and non-parametric methods. Applications include data analysis on images, text, time series, and other types of data using modern software tools such as numpy, scikit-learn, and pytorch.

What's Inside

Machine Learning Algorithms	Mathematical Foundations	Algorithm Implementations
A broad overview of algorithms across ML: generative models, SVMs, tree-based algorithms, neural networks, gradient boosting, etc.	Rigorous definitions of key concepts including: overfitting, regularization, maximum likelihood estimation, latent variable models.	Most algorithms are implemented from scratch in Python using standard libraries such as numpy, scipy, or sklearn.

Prerequisites

This masters-level course requires a background in mathematics and programming at the level of introductory college courses. Experience in Python is recommended, but not required. A certain degree of ease with mathematics will be helpful.

- Programming experience (ideally Python; Cornell CS 1110 or equivalent)
- Linear algebra. (Cornell MATH 2210, MATH 4310 or equivalent)
- Statistics and probability. (Cornell STSCI 2100 or equivalent)

Instructors

These lecture notes accompany CS5785 Applied Machine Learning at Cornell University and Cornell Tech, as well as the open online [version](#) of that course. They are based on materials developed at Cornell by:

- [Volodymyr Kuleshov](#), Assistant Professor, Computer Science, Cornell Tech
- [Nathan Kallus](#), Associate Professor, Operations Research, Cornell Tech
- [Serge Belongie](#), Professor, Computer Science, University of Copenhagen

The open version of CS5785 and the accompanying online lectures have been produced by [Hongjun Wu](#). We are also grateful to over a dozen teaching assistants that have helped with drafts of these lecture notes.

Table of Contents

- [Lecture 1: Introduction to Machine Learning](#)
 - [1.1. What is Machine Learning?](#)
 - [1.2. Three Approaches to Machine Learning](#)
 - [1.3. Logistics and Course Information.](#)
- [Lecture 2: Supervised Machine Learning](#)
 - [2.1. Elements of A Supervised Machine Learning Problem](#)
 - [2.2. Anatomy of a Supervised Learning Problem: The Dataset](#)
 - [2.3. Anatomy of a Supervised Learning Problem: The Learning Algorithm](#)
- [Lecture 3: Linear Regression](#)
 - [3.1. Calculus Review](#)
 - [3.2. Gradient Descent in Linear Models](#)
 - [3.3. Ordinary Least Squares](#)
 - [3.4. Non-Linear Least Squares](#)
- [Lecture 4: Classification and Logistic Regression](#)
 - [4.1. Classification](#)
 - [4.2. Logistic Regression](#)
 - [4.3. Maximum Likelihood](#)
 - [4.4. Learning a Logistic Regression Model](#)
 - [4.5. Softmax Regression for Multi-Class Classification](#)
 - [4.6. Maximum Likelihood: Advanced Topics](#)
- [Lecture 5: Regularization](#)
 - [5.1. Two Failure Cases of Supervised Learning](#)
 - [5.2. Evaluating Supervised Learning Models](#)
 - [5.3. A Framework for Applying Supervised Learning](#)
 - [5.4. L2 Regularization](#)
 - [5.5. L1 Regularization and Sparsity](#)
 - [5.6. Why Does Supervised Learning Work?](#)
- [Lecture 6: Generative Models and Naive Bayes](#)
 - [6.1. Text Classification](#)
 - [6.2. Generative Models](#)
 - [6.3. Naive Bayes](#)
 - [6.4. Learning a Naive Bayes Model](#)
- [Lecture 7: Gaussian Discriminant Analysis](#)
 - [7.1. Revisiting Generative Models](#)
 - [7.2. Gaussian Mixture Models](#)
 - [7.3. Gaussian Discriminant Analysis](#)
 - [7.4. Discriminative vs. Generative Algorithms](#)

- **[Lecture 8: Unsupervised Learning](#)**
- [8.1. Introduction to Unsupervised Learning](#)
- [8.2. The Language of Unsupervised Learning](#)
- [8.3. Unsupervised Learning in Practice](#)
- **[Lecture 9: Density Estimation](#)**
- [9.1. Outlier Detection Using Probabilistic Models](#)
- [9.2. Kernel Density Estimation](#)
- [9.3. Nearest Neighbors](#)
- **[Lecture 10: Clustering](#)**
- [10.1. Gaussian Mixture Models for Clustering](#)
- [10.2. Expectation Maximization](#)
- [10.3. Expectation Maximization in Gaussian Mixture Models](#)
- [10.4. Generalization in Probabilistic Models](#)
- **[Lecture 12: Support Vector Machines](#)**
- [12.1. Classification Margins](#)
- [12.2. The Max-Margin Classifier](#)
- [12.2.2. Algorithm: Linear Support Vector Machine Classification](#)
- [12.3. Soft Margins and the Hinge Loss](#)
- [12.4. Optimization for SVMs](#)
- **[Lecture 13: Dual Formulation of Support Vector Machines](#)**
- [13.1. Lagrange Duality](#)
- [13.2. Dual Formulation of SVMs](#)
- [13.3. Practical Considerations for SVM Duals](#)
- **[Lecture 14: Kernels](#)**
- [14.1. The Kernel Trick in SVMs](#)
- [14.2. Kernelized Ridge Regression](#)
- [14.3. More on Kernels](#)
- **[Lecture 15: Tree-Based Algorithms](#)**
- [15.1. Decision Trees](#)
- [15.2. Learning Decision Trees](#)
- [15.3. Bagging](#)
- [15.4. Random Forests](#)
- **[Lecture 16: Boosting](#)**
- [16.1. Defining Boosting](#)
- [16.2. Structure of a Boosting Algorithm](#)
- [16.3. Adaboost](#)
- [16.4. Ensembling](#)
- [16.5. Additive Models](#)
- [16.6. Gradient Boosting](#)