LearnI!

MACHINE LEARNING SYLLABUS

COURSE OUTLINE:

This course provides a concise introduction to the fundamental concepts in machine learning and popular machine learning algorithms. We will cover the standard and most popular supervised learning algorithms including linear regression, logistic regression, decision trees, k-nearest neighbour, an introduction to Bayesian learning and the naïve Bayes algorithm, support vector machines and kernels and neural networks with an introduction to Deep Learning. We will also cover the basic clustering algorithms. Feature reduction methods will also be discussed. We will introduce the basics of computational learning theory. In the course we will discuss various issues related to the application of machine learning algorithms. We will discuss hypothesis space, overfitting, bias and variance, trade-offs between representational power and learnability, evaluation strategies and cross-validation. The course will be accompanied by hands-on problem solving with programming in Python and some tutorial sessions.

COURSE PLAN:

WEEK-1:

Introduction:

Welcome to Machine Learning! In this module, we introduce the core idea of teaching a computer to learn concepts using data—without being explicitly programmed. The Course Wiki is under construction. Please visit the resources tab for the most complete and up-to-date information.

Linear Regression with One Variable:

Linear regression predicts a real-valued output based on an input value. We discuss the application of linear regression to housing price prediction, present the notion of a cost function, and introduce the gradient descent method for learning.

Linear Algebra Review:

This optional module provides a refresher on linear algebra concepts. Basic understanding of linear algebra is necessary for the rest of the course, especially as we begin to cover models with multiple variables.

WEEK-2:

Linear Regression with Multiple Variables:

What if your input has more than one value? In this module, we show how linear regression can be extended to accommodate multiple input features. We also discuss best practices for implementing linear regression.

Octave/MATLAB Tutorial:

This course includes programming assignments designed to help you understand how to implement the learning algorithms in practice. To complete the programming assignments, you will need to use Octave or MATLAB. This module introduces Octave/MATLAB and shows you how to submit an assignment.

WEEK-3:

Logistic Regression:

Logistic regression is a method for classifying data into discrete outcomes. For example, we might use logistic regression to classify an email as spam or not spam. In this module, we introduce the notion of classification, the cost function for logistic regression, and the application of logistic regression to multi-class classification.

Regularization:

Machine learning models need to generalize well to new examples that the model has not seen in practice. In this module, we introduce regularization, which helps prevent models from overfitting the training data.

WEEK-4:

Neural Networks: Representation:

Neural networks are a model inspired by how the brain works. It is widely used today in many applications: when your phone interprets and understand your voice commands, it is likely that a neural network is helping to understand your speech; when you cash a check, the machines that automatically read the digits also use neural networks.

WEEK-5:

Neural Networks: Learning

In this module, we introduce the backpropagation algorithm that is used to help learn parameters for a neural network. At the end of this module, you will be implementing your own neural network for digit recognition.

WEEK-6:

Advice for Applying Machine Learning:

Applying machine learning in practice is not always straightforward. In this module, we share best practices for applying machine learning in practice, and discuss the best ways to evaluate performance of the learned models.

Machine Learning System Design:

To optimize a machine learning algorithm, you'll need to first understand where the biggest improvements can be made. In this module, we discuss how to understand the performance of a machine learning system with multiple parts, and also how to deal with skewed data.

WEEK-7:

Support Vector Machines:

Support vector machines, or SVMs, is a machine learning algorithm for classification. We introduce the idea and intuitions behind SVMs and discuss how to use it in practice.

WEEK-8:

Unsupervised Learning:

We use unsupervised learning to build models that help us understand our data better. We discuss the k-Means algorithm for clustering that enable us to learn groupings of unlabelled data points.

Dimensionality Reduction:

In this module, we introduce Principal Components Analysis, and show how it can be used for data compression to speed up learning algorithms as well as for visualizations of complex datasets.

WEEK-9:

Anomaly Detection:

Given a large number of data points, we may sometimes want to figure out which ones vary significantly from the average. For example, in manufacturing, we may want to detect defects or anomalies. We show how a dataset can be modelled using a Gaussian distribution, and how the model can be used for anomaly detection.

Recommender Systems:

When you buy a product online, most websites automatically recommend other products that you may like. Recommender systems look at patterns of activities between different users and different products to produce these recommendations. In this module, we introduce recommender algorithms such as the collaborative filtering algorithm and low-rank matrix factorization.

WEEK-10:

Large Scale Machine Learning:

Machine learning works best when there is an abundance of data to leverage for training. In this module, we discuss how to apply the machine learning algorithms with large datasets.

WEEK-11:

Application Example: Photo OCR

Identifying and recognizing objects, words, and digits in an image is a challenging task. We discuss how a pipeline can be built to tackle this problem and how to analyse and improve the performance of such a system.