Introduction to Machine Learning

Jianhua MEI

RSE Machine Learning and Econometrics Reading Group

October 23, 2023

Outline

Introduction

What is Machine Learning?
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.
Let's take a simple example:
Tasks T: Predict the house price in Canberra based on various property features.
Experience E: Dataset containing historical house sales in Canberra with features such as: Number of rooms, House size (in square meters), Proximity to city center (in kilometers)
Performance Measure P: Mean Squared Error (MSE) between the predicted house prices and the actual sale prices.
Goal: Minimize Mean Squared Error (MSE). As more data (experience) is collected and the computer program is further trained, we expect the MSE to decrease, indicating that the model's predictions are becoming more accurate.

Introduction

What is the different between Applied Econometrics and Machine Learning?

Applied Econometrics: Econometrics is primarily concerned with the interpretability. For example, "Adding one room can increase the value of a house by $20,000."
Machine Learning (ML): ML is typically more concerned with prediction accuracy rather than interpretability. The goal is to find a model that minimizes predictive error, even if that model might be hard to interpret or understand.

Overfitting and Regularization

The primary objective is often the enhancement of predictive accuracy and the minimize predictive error.Is the optimal strategy simply to increase the complexity of the model in pursuit of minimizing these errors?

Polynomial Regression:

$$\Phi(x) = [1, x, x^2, x^3, ..., x^D]$$

then polynomial regression has the form:

$$f(x, \theta) = \theta^T \Phi(x)$$

Overfitting and Regularization

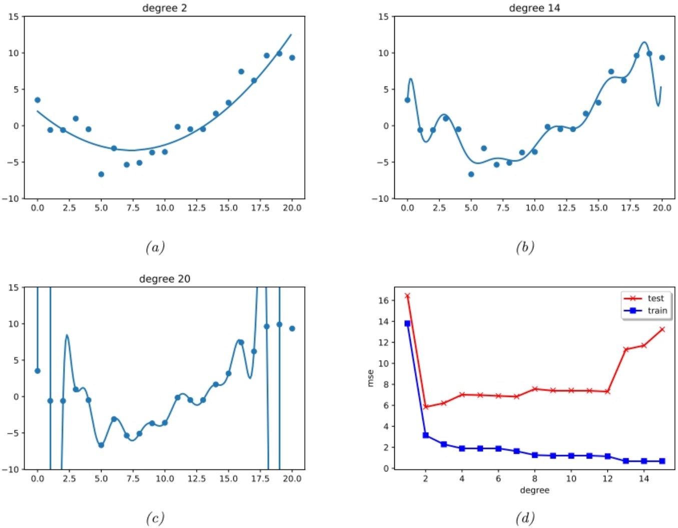


Figure 1.7: (a-c) Polynomials of degrees 2, 14 and 20 fit to 21 datapoints (the same data as in Figure 1.5). (d) MSE vs degree. Generated by linreg_poly_vs_degree.ipynb.

Overfitting and Regularization -Ridge

Ridge Regression (L2 Regularization)

Mathematical Representation:

$$J(\theta) = \text{MSE}(\theta) + \alpha \sum_{k=1}^{K} \theta_i^2$$

where:

- $J(\theta)$ is the cost function.
- $\text{MSE}(\theta)$ stands for Mean Squared Error.
- α is the regularization parameter.
- θ_i are the model parameters.

Characteristics:

1. For $\alpha = 0$, Ridge regression becomes linear regression.
2. As α increases, the regularization effect becomes stronger and the model weights will tend closer to zero, but not exactly zero.

Overfitting and Regularization -Lasso

Lasso Regression (L1 Regularization)

Mathematical Representation:

$$J(\theta) = \text{MSE}(\theta) + \alpha \sum_{k=1}^{K} |\theta_i|$$

where:

- $J(\theta)$ is the cost function.
- $\text{MSE}(\theta)$ stands for Mean Squared Error.
- α is the regularization parameter.
- θ_i are the model parameters.

Characteristics:

1. Lasso can produce a sparse model (many feature weights are zero) when some features are not significant.
2. For $\alpha = 0$, Lasso regression becomes linear regression.
3. As α increases, more and more feature weights will become zero.

Supervised Learning

The most common form of ML is supervised learning. In this problem, the task T is to learn a mapping f from inputs $x \in X$ to outputs $y \in Y$. The inputs x are also called the features.

Examples of Supervised Learning algorithms include:

- **Linear Regression**
- **Logistic Regression**: Binary classification problems.
- **Lasso and Ridge Regression**: Linear regression techniques with regularization.
- **Decision Trees**: Splits a dataset into subsets based on the value of input features. Useful for both regression and classification.
- **Random Forest**: An ensemble method that creates multiple decision trees and merges them to produce a more robust prediction.
- **K-Nearest Neighbors (KNN)**: Classifies a data point based on how its neighbors are classified.
- **Neutral Networks**: A type of neural network used for tasks like image recognition and sequence prediction.

Supervised Learning - Decision Trees

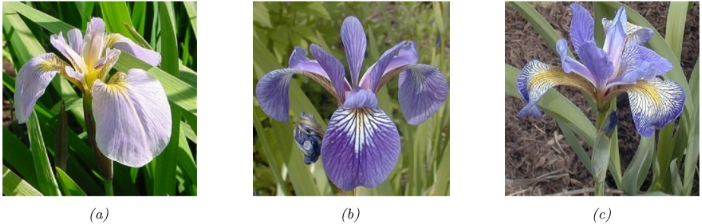
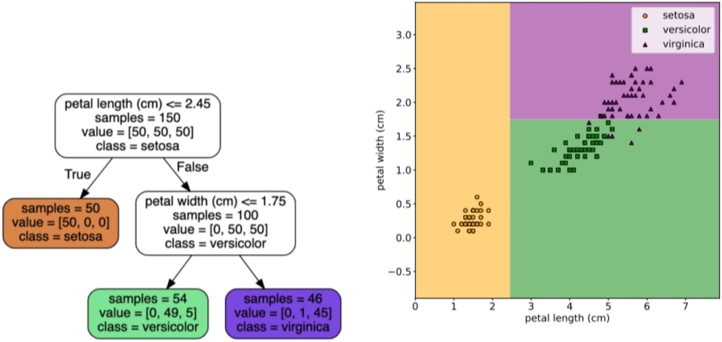


Figure 1.1: Three types of Iris flowers: Setosa, Versicolor and Virginica. Used with kind permission of Dennis Kramb and SIGNA.

index	sl	sw	pl	pw	label
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
	...				
50	7.0	3.2	4.7	1.4	Versicolor
	...				
149	5.9	3.0	5.1	1.8	Virginica

Table 1.1: A subset of the Iris design matrix. The features are: sepal length, sepal width, petal length, petal width. There are 50 examples of each class.

Supervised Learning - Decision Trees



(a) *(b)*

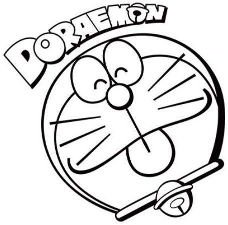
Figure 1.4: Example of a decision tree of depth 2 applied to the Iris data, using just the petal length and petal width features. Leaf nodes are color coded according to the predicted class. The number of training samples that pass from the root to a node is shown inside each box; we show how many values of each class fall into this node. This vector of counts can be normalized to get a distribution over class labels for each node. We can then pick the majority class. Adapted from Figures 6.1 and 6.2 of [Gér19]. Generated by iris_dtree.ipynb.

Supervised Learning - Neural Networks

If you want the computer to directly read pictures of iris flowers and train a mapping between the images and their labels to serve as a classifier, how should you proceed?

The input space X is the set of images, which is a very high-dimensional space:

- For a black and white image D1 $\times$ D2 pixels, represent each intensity with an interger from range $\{0, 1, ..., 255\}$.
- For a color image with C $=$ 3 channels (e.g., RGB) and D1 $\times$ D2 pixels D $=$ C $\times$ D1 $\times$ D2.



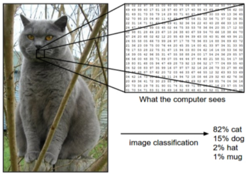
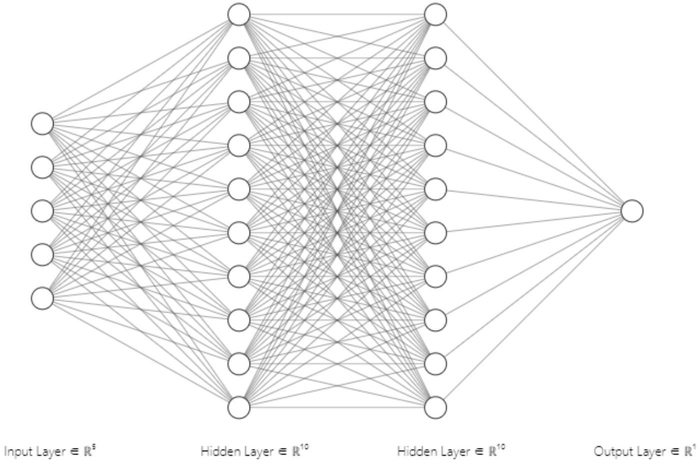


Figure 1.2: Illustration of the image classification problem. From https://cs231n.github.io/. Used with kind permission of Andrej Karpathy.

Supervised Learning - Neural Netwotks



Unsupervised Learning

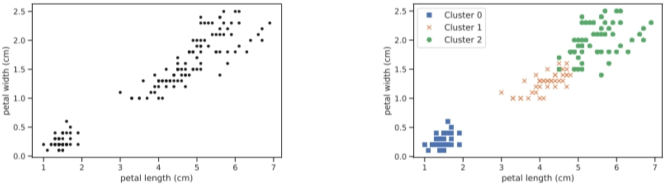
In supervised learning, we assume that each input example x in the training set has an associated set of output targets y, and our goal is to learn the input-output mapping. In unsupervised learning, we just get observed "inputs" X without any corresponding "outputs" Y. This is called unsupervised learning.

- **K-Means:** An algorithm that partitions a dataset into K distinct, non-overlapping subsets (or clusters). It assigns each data point to the cluster whose center (mean) is closest.
- **Principal Component Analysis (PCA):** A dimensionality reduction method that identifies the axes in the dataset that maximize variance and processes data to project it onto these axes.
- **Autoencoders:** A type of neural network used for dimensionality reduction or for feature learning. It learns to encode the input data and then decode it, trying to reconstruct the original input.

Unsupervised Learning- K-Means

index	sl	sw	pl	pw	label
0	5.1	3.5	1.4	0.2	[illegible]
1	4.9	3.0	1.4	0.2	[illegible]
	...				
50	7.0	3.2	4.7	1.4	[illegible]
	...				
149	5.9	3.0	5.1	1.8	[illegible]

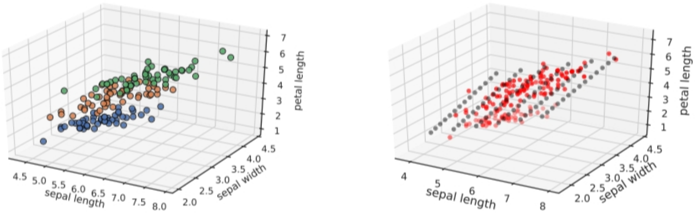
Table 1.1: A subset of the Iris design matrix. The features are: sepal length, sepal width, petal length, petal width. There are 50 examples of each class.



(a) (b)

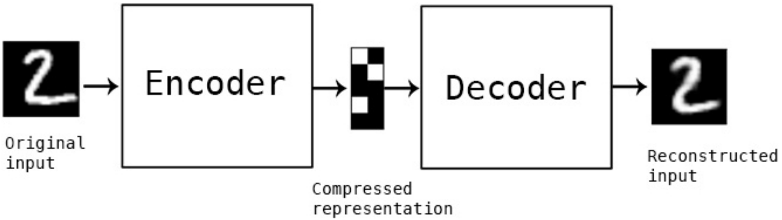
Figure 1.8: (a) A scatterplot of the petal features from the iris dataset. (b) The result of unsupervised clustering using $K = 3$. Generated by iris_kmeans.ipynb.

Unsupervised Learning- Pricinpal Component Analysis(PCA)



(a) (b)

Unsupervised Learning- Autoencoder



Reinforcement Learning

Reinforcement learning is a type of machine learning where an agent learns how to behave in an environment by performing actions and receiving rewards or penalties in return. The agent's objective is to maximize the cumulative reward over time by discovering the best strategy or policy through trial and error.

Reference

[1] Kevin P.Murphy: Probabilistic Machine Learning Ch1
[2] Chollet, François. "Building autoencoders in Keras." Keras Blog. https://blog.keras.io/building-autoencoders-in-keras.html.