Day 2: Linear Regression

Summer STEM: Machine Learning

Nikola Janjušević, Akshaj Kumar Veldanda, Jacky Yuan, Tejaishwarya Gagadam

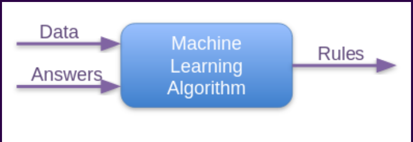
Department of Electrical and Computer Engineering
NYU Tandon School of Engineering
Brooklyn, New York

July 9, 2019

Outline

What is Machine Learning

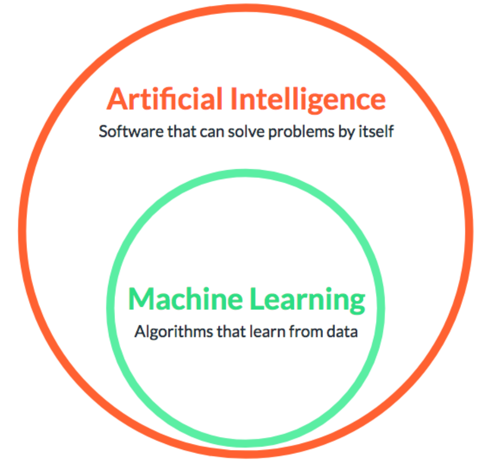
- Learn the algorithm from known data to generate the rules
- Make predictions on unknown data using these rules
- Very effective tool where human expertise is not available



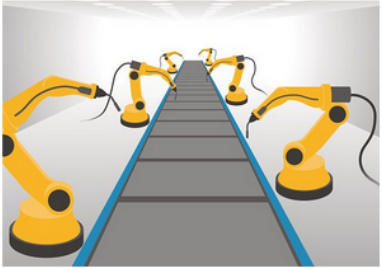
Artificial Intelligence

- Search
- Reasoning and Problem Solving
- Knowledge Representation
- Planning
- Learning
- Perception
- Natural Language Processing
- Motion and Manipulation
- Social and General Intelligence

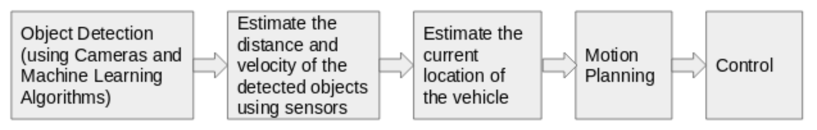
Machine Learning



Autonomous vs. Automated



Autonomous Example: Driver-less Cars



Why is Machine Learning so Prevalent?

- Database mining
- Medical records
- Computational biology
- Engineering
- Recommendation systems
- Understanding the human brain

Why Now?

- Big Data
  - Massive storage. Large data centers
  - Massive connectivity
  - Sources of data from internet and elsewhere
- Computational advances
  - Distributed machines, clusters
  - GPUs and hardware

Supervised Learning

- In supervised learning, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output.
- Supervised learning problems are categorized into "regression" and "classification" problems.
  - In a regression problem, we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function.
  - In a classification problem, we are instead trying to predict results in a discrete output. In other words, we are trying to map input variables into discrete categories.

Supervised Learning - Examples

- Regression - Given a picture of a person, we have to predict their age on the basis of the given picture
- Classification - Given a patient with a tumor, we have to predict whether the tumor is malignant or benign

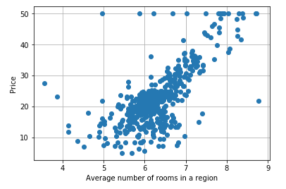
Unsupervised Learning

- Unsupervised learning allows us to approach problems with little or no idea what our results should look like. We can derive structure from data where we don't necessarily know the effect of the target variables.
- We can derive this structure by clustering the data based on relationships among the variables in the data.
- Example: Grouping individual voices from a mesh of sounds.

Outline

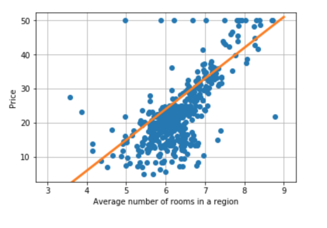
Linear Model

- Data Representation:
  - y = variable you are trying to predict. Also referred to as: Dependent variable, response variable, target variable etc.
  - x = what you are using to predict. Also referred to as: Independent variable, attribute, predictor etc.
  - Set of points, (x_i, y_i), $i = 1, ..., n$. Each data point is called a sample.
  - An efficient way to visualize the data is by plotting y vs x in a scatter plot.



Linear Model

- Assume a linear relationship $y = b + wx$
  - b = intercept
  - w = slope
- $\mathbf{w} = (b, w) = (w_0, w_1)$ are the parameters of the model



- Let's go to the lab to understand this further.

Outline

Is Your Model a Good Fit?

- How would you determine if your model is a good fit or not?
- Talk with your classmates next to you to see whose model fits the data the best
  - How will you determine this?
  - Is there a quantitative way?
  - Write python code if so.

Error Functions

- An **error function** quantifies the discrepancy between your model and the data.
  - They are non-negative, and go to zero as the model gets better.
- Common Error Functions:
  - Mean Squared Error: $MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$
  - Mean Absolute Error: $MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$
- In later units, we will refer to these as **cost functions** or **loss functions**.
- Compute MSE on your model
- How do we interpret MSE? MAE?
  - RMSE?

General Steps to Solve a Machine Learning Problem

- Load and visualize data

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, ..., n$

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, ..., n$
- Find an appropriate model to fit the data

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, ..., n$
- Find an appropriate model to fit the data
  - Eg: Linear model is $\hat{y} = wx + b$

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, ..., n$
- Find an appropriate model to fit the data
  - Eg: Linear model is $\hat{y} = wx + b$
- Choose an appropriate error function

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, ..., n$
- Find an appropriate model to fit the data
  - Eg: Linear model is $\hat{y} = wx + b$
- Choose an appropriate error function
  - $MSE = \sum_{i=1}^{N}(y_i - (b + wx_i))^2$

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, ..., n$
- Find an appropriate model to fit the data
  - Eg: Linear model is $\hat{y} = wx + b$
- Choose an appropriate error function
  - $MSE = \sum_{i=1}^{N}(y_i - (b + wx_i))^2$
- Find parameters that minimize the error function

General Steps to Solve a Machine Learning Problem

- Load and visualize data
  - (x_i, y_i), $i = 1, ..., n$
- Find an appropriate model to fit the data
  - Eg: Linear model is $\hat{y} = wx + b$
- Choose an appropriate error function
  - $MSE = \sum_{i=1}^{N}(y_i - (b + wx_i))^2$
- Find parameters that minimize the error function
  - Select b, w to minimize the error function

Least Squares Fit

- The **Least Squares Fit** is characterized by the minimization of the MSE error function:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Least Squares Fit

- The **Least Squares Fit** is characterized by the minimization of the MSE error function:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

  - Find the parameters, $\mathbf{w} = (b, w) = (w_0, w_1)$, that give the smallest MSE

Least Squares Fit

- The **Least Squares Fit** is characterized by the minimization of the MSE error function:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

  - Find the parameters, $\mathbf{w} = (b, w) = (w_0, w_1)$, that give the smallest MSE
- MSE is a useful metric because there exists an analytic solution to find the optimal parameters b and w

Outline

Mean, Variance, and Covariance, Correlation Coefficient

- Given feature-target data $(x_i, y_i),\ i = 1, 2, ..., N$

Mean, Variance, and Covariance, Correlation Coefficient

- Given feature-target data $(x_i, y_i),\ i = 1, 2, ..., N$
- Mean:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

Mean, Variance, and Covariance, Correlation Coefficient

- Given feature-target data $(x_i, y_i),\ i = 1, 2, ..., N$
- Mean:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

- Variance:

$$\sigma_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2, \quad \sigma_y^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2$$

Mean, Variance, and Covariance, Correlation Coefficient

- Given feature-target data $(x_i, y_i),\ i = 1, 2, ..., N$
- Mean:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

- Variance:

$$\sigma_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2, \quad \sigma_y^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2$$

- **Covariance**:

$$\sigma_{xy} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})$$

Mean, Variance, and Covariance, Correlation Coefficient

- Given feature-target data $(x_i, y_i),\ i = 1, 2, ..., N$
- Mean:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

- Variance:

$$\sigma_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2, \quad \sigma_y^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2$$

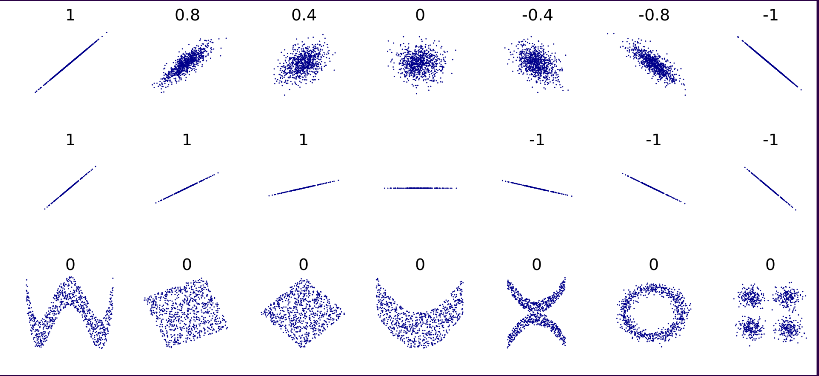
- **Covariance**:

$$\sigma_{xy} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})$$

- **Correlation Coefficient**:

$$\rho = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

Lab: Gaining Intuition



Outline

LS Fit Solution

- Model:

$$\hat{y} = b + wx$$

LS Fit Solution

- Model:

$$\hat{y} = b + wx$$

- Optimization:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

LS Fit Solution

- Model:

$$\hat{y} = b + wx$$

- Optimization:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- Solution:

$$\hat{y} = \bar{y} + \rho\frac{\sigma_y}{\sigma_x}(x - \bar{x})$$

LS Fit Solution

- Model:

$$\hat{y} = b + wx$$

- Optimization:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- Solution:

$$\hat{y} = \bar{y} + \rho\frac{\sigma_y}{\sigma_x}(x - \bar{x})$$

$$w = \rho\frac{\sigma_y}{\sigma_x}, \quad b = \bar{y} - w\bar{x}$$

LS Fit Solution

- Model:

$$\hat{y} = b + wx$$

- Optimization:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- Solution:

$$\hat{y} = \bar{y} + \rho\frac{\sigma_y}{\sigma_x}(x - \bar{x})$$

$$w = \rho\frac{\sigma_y}{\sigma_x}, \quad b = \bar{y} - w\bar{x}$$

- Prediction:

$$y_{new} = b + wx_{new}$$

LS Fit Solution

- Model:

$$\hat{y} = b + wx$$

- Optimization:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- Solution:

$$\hat{y} = \bar{y} + \rho\frac{\sigma_y}{\sigma_x}(x - \bar{x})$$

$$w = \rho\frac{\sigma_y}{\sigma_x}, \quad b = \bar{y} - w\bar{x}$$

- Prediction:

$$y_{new} = b + wx_{new}$$

- Compute the LS fit model

Lab: Find/Build and fit your own data

1. Find your a data set
   - Google: "[subject you're interested in] dataset"
   - `https://archive.ics.uci.edu/ml/datasets.php`
   - `https://toolbox.google.com/datasetsearch`

– or –

2. Build your own data set
   - Only need 10+ samples

Outline

Extending the Model

- Model: $\hat{y} = b \quad + w_1 x_1 + w_2 x_2 + ... + w_D x_D$

Extending the Model

- Model: $\hat{y} = b \quad + w_1 x_1 + w_2 x_2 + ... + w_D x_D$
- Suppose we have multiple features corresponding to a single target output:

Extending the Model

- Model: $\hat{y} = b \quad + w_1x_1 + w_2x_2 + ... + w_Dx_D$
- Suppose we have multiple features corresponding to a single target output:
  - y, single target variable

Extending the Model

- Model: $\hat{y} = b \quad + w_1 x_1 + w_2 x_2 + ... + w_D x_D$
- Suppose we have multiple features corresponding to a single target output:
  - y, single target variable
  - multiple features: $x_1, x_2, ..., x_D$

Extending the Model

- Model: $\hat{y} = b \quad + w_1 x_1 + w_2 x_2 + ... + w_D x_D$
- Suppose we have multiple features corresponding to a single target output:
  - y, single target variable
  - multiple features: $x_1, x_2, ..., x_D$
  - Target variable is a linear combination of all feature variables

Extending the Model

- Model: $\hat{y} = b \quad + w_1 x_1 + w_2 x_2 + ... + w_D x_D$
- Suppose we have multiple features corresponding to a single target output:
  - y, single target variable
  - multiple features: $x_1, x_2, ..., x_D$
  - Target variable is a linear combination of all feature variables
  - **Linear Combination**: sum of scaled features

Extending the Model

- Model: $\hat{y} = b \quad + w_1 x_1 + w_2 x_2 + ... + w_D x_D$
- Suppose we have multiple features corresponding to a single target output:
  - y, single target variable
  - multiple features: $x_1, x_2, ..., x_D$
  - Target variable is a linear combination of all feature variables
  - **Linear Combination**: sum of scaled features
- Vector Notation:

Extending the Model

- Model: $\hat{y} = \quad w_0 + w_1 x_1 + w_2 x_2 + ... + w_D x_D$
- Suppose we have multiple features corresponding to a single target output:
  - y, single target variable
  - multiple features: $x_1, x_2, ..., x_D$
  - Target variable is a linear combination of all feature variables
  - **Linear Combination**: sum of scaled features
- Vector Notation:

  Let $\mathbf{x} = [1, x_1, x_2, ..., x_D]^T$

Extending the Model

- Model: $\hat{y} = \; w_0 + w_1 x_1 + w_2 x_2 + ... + w_D x_D$
- Suppose we have multiple features corresponding to a single target output:
  - y, single target variable
  - multiple features: $x_1, x_2, ..., x_D$
  - Target variable is a linear combination of all feature variables
  - **Linear Combination**: sum of scaled features
- Vector Notation:

  Let $\mathbf{x} = [1, x_1, x_2, ..., x_D]^T$
  and $\mathbf{w} = [w_0, w_1, ..., w_D]^T$

Extending the Model

- Model: $\hat{y} = \quad w_0 + w_1 x_1 + w_2 x_2 + ... + w_D x_D$
- Suppose we have multiple features corresponding to a single target output:
  - y, single target variable
  - multiple features: $x_1, x_2, ..., x_D$
  - Target variable is a linear combination of all feature variables
  - **Linear Combination**: sum of scaled features
- Vector Notation:

  Let $\mathbf{x} = [1, x_1, x_2, ..., x_D]^T$
  and $\mathbf{w} = [w_0, w_1, ..., w_D]^T$
  So, $\hat{y} = \mathbf{x}^T \mathbf{w}$

Matrix Formulation

- We want to minimize the squared error over all the samples: $\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{N}(y_i - \mathbf{x}_i^T\mathbf{w})^2$

Matrix Formulation

- We want to minimize the squared error over all the samples: $\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{N}(y_i - \mathbf{x}_i^T\mathbf{w})^2$
- Design Matrix: Let, $X = \begin{bmatrix} 1 & x_{1_1} & \cdots & x_{1_D} \\ 1 & x_{2_1} & \cdots & x_{2_D} \\ \vdots & & \ddots & \\ 1 & x_{N_1} & \cdots & x_{N_D} \end{bmatrix}$

Matrix Formulation

- We want to minimize the squared error over all the samples: $\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{N}(y_i - \mathbf{x}_i^T\mathbf{w})^2$
- Design Matrix: Let, $X = \begin{bmatrix} 1 & x_{1_1} & \cdots & x_{1_D} \\ 1 & x_{2_1} & \cdots & x_{2_D} \\ \vdots & & \ddots & \\ 1 & x_{N_1} & \cdots & x_{N_D} \end{bmatrix}$
- Can express these conditions in matrix form: $\mathbf{y} = X\mathbf{w}$

NYU TANDON SCHOOL OF ENGINEERING

Matrix Formulation

- We want to minimize the squared error over all the samples: $\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{N}(y_i - \mathbf{x}_i^T\mathbf{w})^2$
- Design Matrix: Let, $X = \begin{bmatrix} 1 & x_{1_1} & \cdots & x_{1_D} \\ 1 & x_{2_1} & \cdots & x_{2_D} \\ \vdots & & \ddots & \\ 1 & x_{N_1} & \cdots & x_{N_D} \end{bmatrix}$
- Can express these conditions in matrix form: $\mathbf{y} = X\mathbf{w}$
- Solution: Pseudo-Inverse

Matrix Formulation

- We want to minimize the squared error over all the samples: $\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{N}(y_i - \mathbf{x}_i^T\mathbf{w})^2$
- Design Matrix: Let, $X = \begin{bmatrix} 1 & x_{1_1} & \cdots & x_{1_D} \\ 1 & x_{2_1} & \cdots & x_{2_D} \\ \vdots & & \ddots & \\ 1 & x_{N_1} & \cdots & x_{N_D} \end{bmatrix}$
- Can express these conditions in matrix form: $\mathbf{y} = X\mathbf{w}$
- Solution: Pseudo-Inverse
  - There isn't a true solution to this equation.

Matrix Formulation

- We want to minimize the squared error over all the samples: $\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{N}(y_i - \mathbf{x}_i^T \mathbf{w})^2$
- Design Matrix: Let, $X = \begin{bmatrix} 1 & x_{1_1} & \cdots & x_{1_D} \\ 1 & x_{2_1} & \cdots & x_{2_D} \\ \vdots & & \ddots & \\ 1 & x_{N_1} & \cdots & x_{N_D} \end{bmatrix}$
- Can express these conditions in matrix form: $\mathbf{y} = X\mathbf{w}$
- Solution: Pseudo-Inverse
    - There isn't a true solution to this equation.
- We say $\mathbf{w}^\star$ solves $\mathbf{y} = X\mathbf{w}$ in the least squares sense, where

$$\mathbf{w}^\star = X^\dagger \mathbf{y}$$

NYU TANDON SCHOOL OF ENGINEERING

Boston Housing Demo

- Using multiple features to predicting house prices
  - Crime rate per capita, number of rooms, student-teacher ratio at local schools, ...
- Lets use a linear model that takes into account all the collected data

M6: *Demo*: Multivariable Regression on Boston Housing Data

```
import pandas as pd
import numpy as np

df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/housing/housing.data',
                 header=None,delim_whitespace=True,names=names,na_values='?')

df.head(6)  # print the first six samples
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	PRICE
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2
5	0.02985	0.0	2.18	0	0.458	6.430	58.7	6.0622	3	222.0	18.7	394.12	5.21	28.7

NYU TANDON SCHOOL OF ENGINEERING

Outline

Robot Arm Calibration

- Let's train a model based on the given data.
- In this lab we're going to:
  - Predict the *current* drawn
  - Predictors, X: Robot arm's joint angles, velocity, acceleration, strain gauge readings (load measurement).

Thank You!

- Next Class: Generalization Error
- How do our models hold up against prediction new data?