Machine Learning by Stanford University

Study Sheet by Reah Miyara ✉ml@reah.me

Intro to Machine Learning

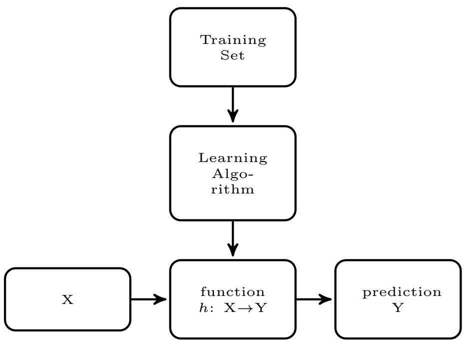
- **ML** – a computer program with increased performance P at some class of tasks T with experience E.
- **Supervised** – given a ['ground truth'] data set, predict output given the input. Types of prediction:

  1. **Regression** – continuous, numerical
  2. **Classification** – discrete, categorical

- **Unsupervised** – derive structure from data based on relationships among variables (with no prior knowledge as to what the results should look like)

Linear Regression with One Variable

- **Learning Goal** – given a training set, learn a function h: X→Y so $h(x)$ is a good y predictor



- **Hypothesis** – $h_\theta(x) = \theta_0 + \theta_1 x$
- **Cost Function** – takes an average difference of all results of the hypothesis with inputs from the x values and the actual y values. Goal: minimize θ_0, θ_1

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2 \qquad (1)$$

(1) Squared Error function or Mean Squared Error function

- **Gradient Descent Algorithm** repeat until convergence

$$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta_0, \theta_1) \qquad (2)$$

Multivariate Linear Regression

$$h_\theta(x) = \begin{bmatrix}\theta_0 & \theta_1 & \dots & \theta_n\end{bmatrix}\begin{bmatrix}x_0 \\ x_1 \\ \vdots \\ x_n\end{bmatrix} = \theta^T x$$

- **Gradient Descent Algorithm** repeat until convergence

$$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)} - y^{(i)})\cdot x_j^{(i)} \text{ ; j:= 0...n} \qquad (3)$$

- **Feature Scaling** – divide the input values by the range (max – min). Input values in roughly the same range speed up the convergence of gradient descent.
- **Mean Normalization** – subtract the mean for an input variable from the values for that input variable.

$$x_i := \frac{x_i - \mu_i}{s_i} \qquad (4)$$

(4) μ_i is the mean & s_i is the range, $(max - min)$, of all values for feature i

- **Learning Rate** – α too small $\implies$ slow convergence; α too large $\implies$ may not converge.

Normal Equation

- **Normal Equation** – non-iterative algorithm for minimizing $J(\theta)$; *note* : O(n^3) to calculate X^TX

$$\theta = (X^TX)^{-1}X^Ty \qquad (5)$$

m examples $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$; n features

$$x^{(i)} = \begin{bmatrix}x_0^{(i)} \\ x_1^{(i)} \\ \vdots \\ x_n^{(i)}\end{bmatrix} \in \mathbb{R}^{n+1} \;\mid\; \text{X}= \begin{bmatrix}(x^{(1)})^T \\ (x^{(2)})^T \\ \vdots \\ (x^{(n)})^T\end{bmatrix} \;\mid\; \text{y} = \begin{bmatrix}y^1 \\ y^2 \\ \vdots \\ y^m\end{bmatrix}$$

$x^{(i)}$ = training vector i (containing values from all features); $X \to m$x(n+1)

- **If X^{T}X is noninvertible**, common causes include:

  1. Redundant features, where two features are very closely related (i.e. they are linearly dependent)
  2. Too many features (e.g. m $\leq$ n). In this case, delete some features or use 'regularization'.