Machine Learning: First Steps

John Donaghy

Outline

- Overview of ML
- Linear Regression
  - Linear Algebra
  - Statistics
- Stochastic Gradient Descent
- Scikit-Learn Implementation/workflow

Machine Learning Overview

- Supervised Learning
  - Regression, Classification
    - Linear Regression, Logistic Regression, Random Forests, multilayer perceptron, CNN, etc …
- Unsupervised Learning
  - Clustering, outlier detection, Generative models
    - Autoencoders, GAN, K-means, etc …
- Self Supervised Learning
  - “In self-supervised learning, the system learns to predict part of its input from other parts of its input” -LeCun

Linear Regression

$$f(\mathbf{x}) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + ...\epsilon \tag{1}$$

$$f(\mathbf{x}) = \theta^T \mathbf{x}$$

$$\mathcal{L} = \frac{1}{N} \sum_{n}^{N} (\theta^T \mathbf{x}_i - y_i)^2$$

y_i:data label(regression target), $\mathbf{x}_i$: feature inputs (data px1) , θ_i: model parameters (px1 absorbs ϵ)

Linear Algebra Approach

$$\nabla_\theta \mathcal{L} = \mathbf{0} \tag{2}$$

$$\frac{1}{N}\sum_{n}^{N} 2\left(\theta^T \mathbf{x}_i - y_i\right)\mathbf{x}_i = 0$$

$$\sum_{n}^{N}\left(\theta^T \mathbf{x}_i\right)\mathbf{x}_i = \sum_{n}^{N} y_i \mathbf{x}_i$$

$$\left(\sum_{n}^{N} \mathbf{x}_i \mathbf{x}_i^T\right)\theta = \sum_{n}^{N} y_i \mathbf{x}_i$$

$$\theta = \left(\sum_{n}^{N} \mathbf{x}_i \mathbf{x}_i^T\right)^{-1} \sum_{n}^{N} y_i \mathbf{x}_i$$

tensor inputs $\rightarrow$ $$\theta = \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y}$$

Statistics Approach

$$\mathbf{y} = \theta^T \mathbf{x} + \epsilon \tag{3}$$

define residual as $r \equiv \mathbf{y} - \theta^T \mathbf{x} = \epsilon$, as $n \to \infty$ allows us to use the central limit theorem

$$\epsilon \sim \mathcal{N}(0, 1)$$

$$y_i \sim \mathcal{N}(\theta^T \mathbf{x}_i, \sigma^2)$$

Maximum Likelihood: assume that your samples were drawn from the most probable distribution

$$p(\mathbf{y}|\mathbf{x}) = \prod_i p(y_i|\mathbf{x}_i)$$

$$p(y_i|\mathbf{x}_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(y_i - \theta^T \mathbf{x}_i)^2}{2\sigma^2}}$$

Statistics Approach

Using the log likelihood (easier to manipulate)

$$p(\mathbf{y}|\mathbf{x}) = \sum_i \left[\log \frac{1}{\sqrt{2\pi\sigma^2}} + \log e^{\frac{(y_i - \theta^T \mathbf{x}_i)^2}{2\sigma^2}} \right]$$

$$\nabla_\theta p(\mathbf{y}|\mathbf{x}) = 0$$

$$\theta = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y}$$

Linear Regression maximizes the probability of data with normal residuals !!!

SGD

- Optimal coefficients: $O(n \times p^3)$
- SGD: $O(k \times n \times p)$ with k=number of iterations
  - Sample subset of data points (stochastic)
  - Update
  - Repeat until convergence

$$\theta_t = \theta_{t-1} - \eta \nabla_\theta \mathcal{L}(\theta_{t-1}) \tag{4}$$

η: learning rate

$$\theta_t = \theta_{t-1} - 2\eta(\theta^T \mathbf{x}_i - y_i)\mathbf{x}_i$$