Hands-on Machine Learning

Tutorial-1

Contents

- Working with real data
- Univariate linear regression
  - Get the data
  - Discover & visualize the data
  - Train a model
- Multivariate Linear Regression
- Linear Regression using Normal Equation
- Linear Regression using Gradient Descent

Working with real data

- Task : Build a model of housing price in California to **predict** the **median housing price** in any district
- Data: California census data

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0

median_income	median_house_value	ocean_proximity
8.3252	452600.0	NEAR BAY
8.3014	358500.0	NEAR BAY
7.2574	352100.0	NEAR BAY
5.6431	341300.0	NEAR BAY
3.8462	342200.0	NEAR BAY

Overall Pipeline

- Get the data
- Discover and visualize the data to gain insights
- Prepare the data for machine learning algorithms
- Select and train a model
- Fine tune your model

Look at the big picture

- Frame the problem
  - Supervised or unsupervised learning
  - Classification or regression
- Select a perfomance measure: how much error the system makes in its prediction
- Check the assumptions :whether to get the actual price or just categories(cheap, medium, expensive)

$$\text{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(h(\mathbf{x}^{(i)}) - y^{(i)})^2}$$

$$\text{MAE}(\mathbf{X}, h) = \frac{1}{m}\sum_{i=1}^{m}\left| h(\mathbf{x}^{(i)}) - y^{(i)} \right|$$

Univariate Linear Regression

- Load the data with a single feature variable and a predictor variable
- Create a test set
- Look at the data structure
- Visualize the data
- Look for correlations
- Data cleaning
- Train a regression model

Multivariate Linear Regression

- Load the data with a multiple feature variable and a predictor variable
- Create a test set
- Look at the data structure
- Visualize the data
- Look for correlations
- Data cleaning
- Train a regression model

Linear Regression using Normal Equation

$$\text{MSE}(\mathbf{X}, h_\theta) = \frac{1}{m}\sum_{i=1}^{m}(\theta^T \cdot \mathbf{x}^{(i)} - y^{(i)})^2$$

$$\text{Normal Equation: } \theta = (X^T X)^{-1} X^T \vec{y}$$

Linear regression using Gradient Descent

$$\mathrm{MSE}(\boldsymbol{X}, h_\theta) = \frac{1}{m}\sum_{i=1}^{m}(\theta^T \cdot \boldsymbol{x}^{(i)} - y^{(i)})^2$$

$$\nabla_\theta \mathrm{MSE}(\theta) = \begin{pmatrix} \frac{\partial}{\partial\theta_0}\mathrm{MSE}(\theta) \\ \frac{\partial}{\partial\theta_1}\mathrm{MSE}(\theta) \\ \vdots \\ \frac{\partial}{\partial\theta_n}\mathrm{MSE}(\theta) \end{pmatrix} = \frac{2}{m}\boldsymbol{X}^T \cdot (\boldsymbol{X}\cdot\theta - \boldsymbol{y})$$

$$\theta^{(\text{next step})} = \theta - \eta\nabla_\theta \mathrm{MSE}(\theta)$$

Installation

- Python, jupyter , matplotlib, numpy, pandas, scipy, sklearn
- Prefer Anaconda virtual environment and jupyter notebook to work, you can use other suitable installation as per your comfort.

Reference

- Aurélien Géron, “Hands-On Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems”
- https://github.com/ageron/handson-ml
- https://github.com/harihari1989/LinearRegression
- http://www.ozzieliu.com/tutorials/Linear-Regression-Gradient-Descent.html