PHƯƠNG PHÁP BÌNH PHƯƠNG TỐI THIỂU CHO THUẬT TOÁN MÁY HỌC HỒI QUY TUYẾN TÍNH

LEAST SQUARES METHOD FOR LINEAR REGRESSION ALGORITHM

***Nguyễn Văn Diêu*[1]**

[1]*Khoa CNTT, Trường ĐH Giao thông Vận tải Tp.HCM. dieu.nguyen@ut.edu.vn*

Tóm tắt: *Hồi quy tuyến tính là một trong những thuật toán cơ bản và có nhiều ứng dụng trong ngành máy học. Giải quyết thuật toán này có một số phương pháp như: Bình phương tối thiểu, gradient descent, maximum likelihooh, ... Phần này chúng tôi giới thiệu phương pháp bình phương tối thiểu cùng với minh họa bằng cách sử dụng thư viện chuẩn của Python và hệ sinh thái máy học của nó.*

Từ khóa: *Bình phương tối thiểu, Hồi quy tuyến tính, Ngôn ngữ Python, Máy học.*

Abstract: *Linear regression is a common task in machine learning with a variety of applications. To be solve this algorithm we have some methods: least squares, gradient descent, maximum likelihooh, ... In this section we provide a least squares method and illustrate by python language from scratch and its famous ecosystem for machine learning.*

Keywords: *Least Squares, Linear Regression, Python, Scikit-Learn, Machine Learning.*

1 Giới thiệu

Bài toán cơ bản và quan trọng được ứng dụng rộng rãi của ngành máy học là tìm cách "học" hay "suy diễn" hàm diễn tả mối quan hệ giữa một hoặc nhiều thuộc tính của dữ liệu và nhãn của chúng từ đó có thể dự đoán được nhãn của một dữ liệu bất kỳ.

$$f : Data\{att_1, att_2, ...\} \rightarrow \{label\}$$

att: Thuộc tính hay đặc trưng của data.
$label$: Nhãn hay target của data.

Nếu mối quan hệ của Data và {label} có dạng tuyến tính; nghĩa là mối quan hệ này biểu diễn được trên đường thẳng (trên $\mathbb{R}^2$), mặt phẳng (trên $\mathbb{R}^3$) hoặc siêu mặt phẳng (trên $\mathbb{R}^n$, n > 3) hay nói chính xác f là một hàm tuyến tính.

Khi đó ta có phương pháp học hồi quy tuyến tính [1].

2 Thuật toán hồi quy tuyến tính

Có nhiều thuật toán học hồi quy [2]: Linear Regression, Polynomial Regression, Logistic Regression, Ridge Regression, Support Vector Regression, ... Bài này đề cập đến Hồi quy tuyến tính đơn biến (Simple Linear Regression) và phương pháp bình phương tối thiểu (Least Squares Method) dùng trong thuật toán này.

Thuật toán hồi quy tuyến tính thuộc lớp bài toán học có giám sát (Supervised Learning) gồm các thành phần sau:

- Training set (tập huấn luyện):

$$\mathcal{D} = \{(x^{(1)}, y^{(1)}), ..., (x^{(N)}, y^{(N)})\}$$

  N: số lượng mẫu học.
  $x^{(i)} \in \mathbb{R}^d$
  $y^{(i)} \in \mathbb{R}$

- Hypothesis (giả thuyết):

$$\begin{aligned} h(\theta, x) &= \theta_0 + \theta_1 x_1 + ... + \theta_d x_d \\ &= \theta_0 + \sum_{i=1}^{d} \theta_i x_i \\ &= \Theta^T \mathbf{x} \end{aligned}$$

  Với:

$$\Theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$$

  Θ^T: Ma trận chuyển vị của Θ

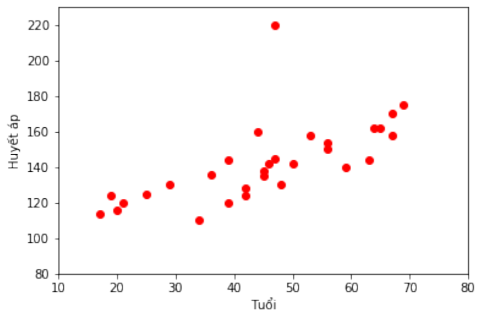
Mục đích của chúng ta là đi xác định $h(\theta, x)$ từ những giá trị x trong training set.

Đó là việc xác định các tham số θ sao cho khi kiểm tra một cặp (x, y) thì:

$$h(\theta, x) \sim y$$

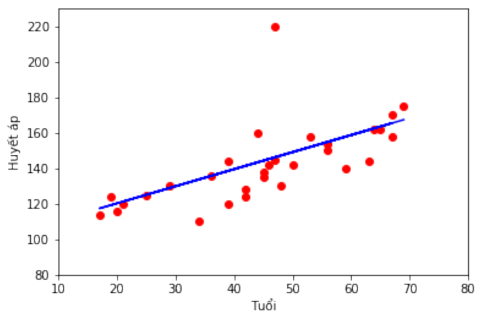
với xác suất xấp xỉ lớn nhất.

Ví dụ 1: Mối quan hệ giữa tuổi và huyết áp tâm thu [3] có đồ thị scatter (phân tán) [4] sau:



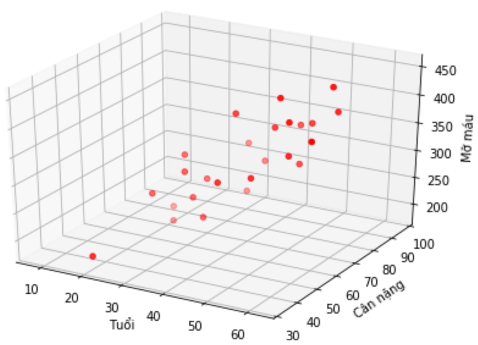
Hình 1. *Mối quan hệ giữa tuổi và huyết áp.*

Trường hợp này hypothesis cần được xây dựng là một phương trình đường thẳng trên $\mathbb{R}^2$ [4] có kết quả như sau:



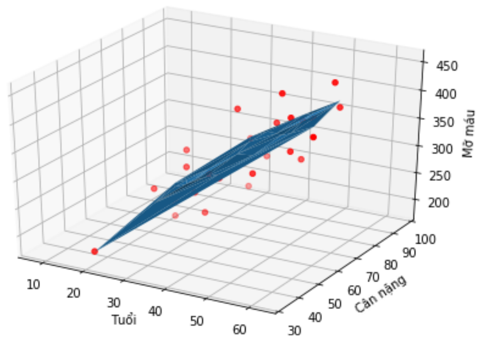
Hình 2. *Hypothesis của tuổi và huyết áp.*

Ví dụ 2: Hàm lượng mỡ trong máu liên quan đến hai đại lượng tuổi và cân nặng [3] có đồ thị scatter [4] như sau:



Hình 3. *Mỡ trong máu với tuổi và cân nặng.*

Hypothesis của nó là một phương trình mặt phẳng trên $\mathbb{R}^3$ [4] có kết quả như sau:



Hình 4. *Hypothesis tuyến tính.*

3 Hồi quy tuyến tính đơn biến

Xét $x \in \mathbb{R}$ thì ta có một hypothesis hồi quy tuyến tính đơn biến [5] (Simple Linear Regression):

$$\begin{aligned} h(\theta, x) &= \theta_0 + \theta_1 x \\ &= \Theta^T \mathbf{x} \end{aligned}$$

Với:

$\Theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} 1 \\ x \end{bmatrix}$

H.1 và H.2 của VD.1 diễn tả trường hợp hồi quy tuyến tính đơn biến.

3.1 Xây dựng Loss function (hàm tổn thất)

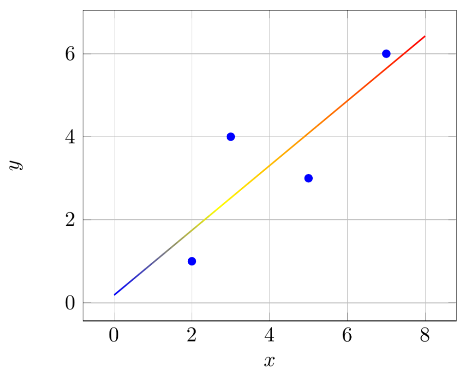
Ví dụ 3: Giả sử ta có tập dữ liệu với $N = 4$.

$$\begin{aligned} \mathcal{D} &= \{(x^{(1)}, y^{(1)}), ..., (x^{(N)}, y^{(N)})\} \\ &= \{(2, 1), (3, 4), (5, 3), (7, 6)\} \end{aligned}$$

Ta xác định:

$$h(\theta, x) = \theta_0 + \theta_1 x$$

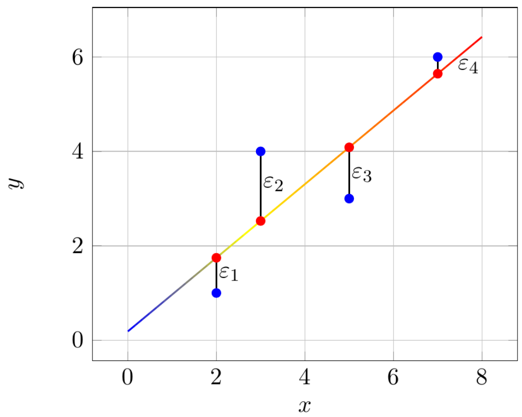
Giả sử ta tìm được một hypothesis:



Hình 5. *Scatter và hypothesis của VD.3.*

Bình phương sự sai biệt (tổn thất) ϵ_i cho từng cặp $(y^{(i)}, h(\theta, x^{(i)}))$ ta có loss function của cặp đó [6]:

$$\begin{aligned}\mathcal{L}_i(\theta_0,\theta_1) &= {\epsilon_i}^2\\ &= (y^{(i)} - h(\theta, x^{(i)}))^2\\ &= (y^{(i)} - (\theta_0 + \theta_1 x^{(i)}))^2\end{aligned}$$



Hình 6. *Tổn thất ϵ_i của $(y^{(i)}, h(\theta, x^{(i)}))$*

Trung bình của N tổn thất là:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i(\theta_0,\theta_1)$$

Ta tìm cách cực tiểu hóa loss function $\mathcal{L}$ theo hai tham số θ_0 và θ_1:

$$\underset{\theta_0,\theta_1}{argmin}\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i(\theta_0,\theta_1)$$

hay

$$\underset{\theta_0,\theta_1}{argmin}\frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - (\theta_0 + \theta_1 x^{(i)}))^2$$

Ta có:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - (\theta_0 + \theta_1 x^{(i)}))^2$$

Đạo hàm từng phần của $\mathcal{L}$ theo θ_0 và θ_1:

$$\frac{\partial\mathcal{L}}{\partial\theta_0} = \frac{2}{N}\sum_{i=1}^{N} -(y^{(i)} - (\theta_0 + \theta_1 x^{(i)}))$$

$$\frac{\partial^2\mathcal{L}}{\partial\theta_0^2} = \frac{2}{N}\sum_{i=1}^{N} 1 = 2$$

$$\frac{\partial\mathcal{L}}{\partial\theta_1} = \frac{2}{N}\sum_{i=1}^{N} -x^{(i)}(y^{(i)} - (\theta_0 + \theta_1 x^{(i)}))$$

$$\frac{\partial^2\mathcal{L}}{\partial\theta_1^2} = \frac{2}{N}\sum_{i=1}^{N} {x^{(i)}}^2$$

Đạo hàm cấp 2 của θ_0 và θ_1 đều dương, vậy ta có giá trị cực tiểu của loss function $\mathcal{L}$ tại các vị trí θ_0 và θ_1 có đạo hàm cấp 1 bằng không.

3.1.1 Xác định θ_0

Tính $\frac{\partial\mathcal{L}}{\partial\theta_0} = 0$ để xác định θ_0

$$\frac{2}{N}\sum_{i=1}^{N} -(y^{(i)} - (\theta_0 + \theta_1 x^{(i)})) = 0$$

$$\Leftrightarrow \frac{1}{N}\sum_{i=1}^{N}(-y^{(i)} + \theta_0 + \theta_1 x^{(i)}) = 0$$

$$\Leftrightarrow -\frac{1}{N}\sum_{i=1}^{N} y^{(i)} + \frac{1}{N}\sum_{i=1}^{N}\theta_0$$

$$+\,\theta_1\frac{1}{N}\sum_{i=1}^{N} x^{(i)} = 0$$

$$\Leftrightarrow -\overline{y} + \theta_0 + \theta_1\overline{x} = 0$$

Với

$$\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y^{(i)}$$

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x^{(i)}$$

Ta được

$$\boxed{\theta_0 = \overline{y} - \theta_1\overline{x}} \qquad (1)$$

3.1.2 Xác định θ_1

Tính $\frac{\partial\mathcal{L}}{\partial\theta_1} = 0$ để xác định θ_1

$$\frac{2}{N}\sum_{i=1}^{N} -x^{(i)}(y^{(i)} - (\theta_0 + \theta_1 x^{(i)})) = 0$$

$$\Leftrightarrow \frac{1}{N}\sum_{i=1}^{N}(x^{(i)}y^{(i)} - \theta_0 x^{(i)} - \theta_1 {x^{(i)}}^2) = 0$$

Thay θ_0 từ phương trình (1) ta được

$$\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}y^{(i)} - x^{(i)}\overline{y} + \theta_1 x^{(i)}\overline{x} - \theta_1 {x^{(i)}}^2) = 0$$

$$\Leftrightarrow \frac{1}{N}\sum_{i=1}^{N}(x^{(i)}y^{(i)}) - \frac{1}{N}\sum_{i=1}^{N}(x^{(i)}\overline{y}) - \overline{x}\,\overline{y} + \overline{x}\,\overline{y}$$

$$-\theta_1\frac{1}{N}\sum_{i=1}^{N}x^{(i)2}+\theta_1\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}\overline{y})=0$$

$$\begin{aligned}\Leftrightarrow\ &\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}y^{(i)})-\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}\overline{y})\\&-\frac{1}{N}\sum_{i=1}^{N}(\overline{x}y^{(i)})+\frac{1}{N}\sum_{i=1}^{N}(\overline{x}\,\overline{y})\\&-\theta_1\frac{1}{N}\sum_{i=1}^{N}x^{(i)2}+\theta_1\overline{x}\,\overline{x}=0\end{aligned}$$

$$\begin{aligned}\Leftrightarrow\ &\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}y^{(i)}-x^{(i)}\overline{y}-\overline{x}y^{(i)}+\overline{x}\,\overline{y})\\&-\theta_1\frac{1}{N}\sum_{i=1}^{N}x^{(i)2}+2\theta_1\overline{x}\,\overline{x}-\theta_1\overline{x}\,\overline{x}=0\end{aligned}$$

$$\begin{aligned}\Leftrightarrow\ &\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}-\overline{x})(y^{(i)}-\overline{y})-\theta_1\frac{1}{N}\sum_{i=1}^{N}x^{(i)2}\\&+\theta_1\frac{1}{N}\sum_{i=1}^{N}2x^{(i)}\overline{x}-\theta_1\frac{1}{N}\sum_{i=1}^{N}\overline{x}^2=0\end{aligned}$$

$$\begin{aligned}\Leftrightarrow\ &\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}-\overline{x})(y^{(i)}-\overline{y})\\&-\theta_1\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}-\overline{x})^2=0\end{aligned}$$

Suy ra

$$\theta_1=\frac{\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}-\overline{x})(y^{(i)}-\overline{y})}{\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}-\overline{x})^2}$$

hay

$$\boxed{\theta_1=\frac{Cov(x,y)}{Var(x)}} \tag{2}$$

$Cov(x,y)$: Hiệp phương sai của hai biến ngẫu nhiên x và y.
$Var(x)$: Phương sai của biến ngẫu nhiên x.

3.2 Mã lệnh Python

Mã lệnh Python cho các tính toán Simple Linear Regression:

$Mean(x)=\frac{1}{N}\sum_{i=1}^{N}x^{(i)}$

```python
def mean(values):
    return sum(values) / float(len(values))
```

$Var(x)=\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}-\overline{x})^2$

```python
def variance(values, mean):
    return sum([(x-mean)**2 for x in values]) / float(len(values))
```

$Cov(x,y)=\frac{1}{N}\sum_{i=1}^{N}(x^{(i)}-\overline{x})(y^{(i)}-\overline{y})$

```python
def covariance(x, mean_x, y, mean_y):
    covar = 0.0
    for i in range(len(x)):
        covar += (x[i] - mean_x) * (y[i] - mean_y)
    return covar / float(len(x))
```

Tính các hệ số θ_0 và θ_1:

```python
def coefficients(x, y):
    x_mean, y_mean = mean(x), mean(y)
    theta1 = covariance(x, x_mean, y, y_mean) / variance(x, x_mean)
    theta0 = y_mean - theta1 * x_mean
    return [theta0, theta1]
```

Xác định kết quả hồi quy của tập test:

```python
def SLR_predict(theta0, theta1, test):
    predict = list()
    for x in test:
        h = theta0 + theta1 * x
    predict.append(h)
    return predict
```

Áp dụng cho VD.3:

```python
data = [[2, 1], [3, 4], [5, 3], [7, 6]]
```

```
x = [row[0] for row in data]
y = [row[1] for row in data]
theta0, theta1 = coefficients(x, y)
```

Các hệ số theta tính được:
$\theta_0 = 0.18644067796610164$
$\theta_1 = 0.7796610169491526$

$$h(\theta, x) = 0.186 + 0.780x$$

Chính là đường thẳng minh họa ở H.5 và H.6.

Giá trị hồi quy của $test = [4, 8, 9]$:

```
SLR_predict(theta0, theta1, test)
```

Kết quả:

```
[3.305084745762712, 6.423728813559322, 7.203389830508474]
```

3.3 Mã lệnh Python OOP[1]

Áp dụng cho class Simple Linear Regression (SLR):

```
class SLR:

    def __init__(self):
        pass

    def fix(self, x, y):
        self.theta0, self.theta1 = SLR.coef(self, x, y)

    def predict(self, test):
        predict = list()
        for x in test:
            h = self.theta0 + self.theta1 * x
        predict.append(h)
        return predict

    def mean(values):
        return sum(values) / float(len(values))

    def variance(values, mean):
        return sum([(x-mean)**2 for x in values]) / float(len(values))

    def covariance(x, mean_x, y, mean_y):
        covar = 0.0
        for i in range(len(x)):
            covar += (x[i] - mean_x) * (y[i] - mean_y)
        return covar / float(len(x))

    def coef(self, x, y):
        x_mean, y_mean = SLR.mean(x), SLR.mean(y)
        self.theta1 = SLR.covariance(x, x_mean, y, y_mean) / SLR.variance(x, x_mean)
        self.theta0 = y_mean - self.theta1 * x_mean
        return [self.theta0, self.theta1]
```

Áp dụng cho VD.3:

```
data = [[2, 1], [3, 4], [5, 3], [7, 6]]
x = [row[0] for row in data]
y = [row[1] for row in data]
test = [4, 8, 9]
```

```
vd3 = SLR()
vd3.fix(x, y)
vd3.predict(test)
```

Kết quả tập $\{y\}$ của test:

```
[3.305084745762712, 6.423728813559322, 7.203389830508474]
```

3.4 Sử dụng hệ sinh thái Machine Learning của Python

Python có nhiều thư viện đủ để chúng ta triển khai các thuật toán máy học. Chúng ta dùng tối thiểu các thư viện: Numpy, Pandas,

[1] Object Oriented Programming

Matplotlib và Sikitikit-learn cho bài toán của VD.1 và VD.2:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

Đọc data huyết áp và tuổi từ file ".csv":

```python
dataset = pd.read_csv('AgeBloodPressure.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values
```

Biểu đồ scatter quan hệ giữa huyết áp và tuổi của VD.1, H.1:

```python
plt.scatter(X, y, color = 'red')
plt.xlabel('Tuoi')
plt.ylabel('Huyet ap')
plt.ylim(80,230)
plt.xlim(10,80)
plt.show()
```

Biểu đồ hypothesis của VD.1, H.2:

```python
Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, test_size = 1/3, random_state = 0)
slr = LinearRegression()
slr.fit(Xtrain, ytrain)
plt.plot(X1, slr.predict(X1), color = 'blue')
plt.scatter(X, y,  marker='o', c='r', edgecolor='b')
plt.xlabel('$X$')
plt.ylabel('$y$')
plt.tick_params(axis='x', colors='red')
plt.tick_params(axis='y', colors='blue')
plt.show()
```

Đọc data của VD.2 từ file ".csv" và vẽ biểu đồ scatter mỡ máu, tuổi và cân nặng, H.3:

```python
dataset = pd.read_csv('WeightAgeBloodFat.csv')
X = dataset.iloc[:,:-1]
y = dataset.iloc[:,2]

fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(dataset.iloc[:,0], dataset.iloc[:,1], dataset.iloc[:,2], c='r', marker='o')
ax.set_xlabel('Tuoi')
ax.set_ylabel('Can nang')
ax.set_zlabel('Mo mau')
ax.set_xlim3d(5,65)
ax.set_ylim3d(30,100)
plt.show()
```

Biểu đồ scatter và mặt phẳng hypothesis của VD.2, H.4:

```python
slr = LinearRegression()
slr.fit(X_train, y_train)

fig = plt.figure()
ax = Axes3D(fig)

ax.plot_trisurf(dataset.iloc[:,0], dataset.iloc[:,1], slr.predict(X) , linewidth=0.2, antialiased=True)

fig = plt.figure()
ax = Axes3D(fig)

ax.scatter(dataset.iloc[:,0], dataset.iloc[:,1], dataset.iloc[:,2], c='r', marker='o')
ax.set_xlabel('Tuoi')
ax.set_ylabel('Can nang')
ax.set_zlabel('Mo mau')
ax.set_xlim3d(5,65)
ax.set_ylim3d(30,100)
plt.show()
```

4 Kết luận

Như trên, dựa trên phương pháp bình phương tối thiểu, chúng tôi đã trình bày chi tiết từ ý tưởng đến việc xây dựng mô hình cho lớp bài toán máy học hồi quy tuyến tính đơn biến cũng như giới thiệu mã lệnh Python cho mô hình này.

Đây là mô hình phổ biến, cơ bản, dễ tiếp thu cho sinh viên và những ai mới bắt đầu quan tâm đến Máy học. Từ đây, cùng với các kỹ thuật tiền xử lý dữ liệu, chọn mẫu training, testing và đánh giá hypothesis ... chúng ta có thể áp dụng dễ dàng cho nhiều bài toán rất thú vị trong thực tế.

5 Tài liệu tham khảo

[1] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1999.

[2] ListenData, *15 types of regression you should know*. [Online]. Available: `https://www.r-bloggers.com/15-types-of-regression-you-should-know/`.

[3] A. Asuncion and D. Newman, *Machine learning repository*. [Online]. Available: `https://archive.ics.uci.edu/ml/index.php`.

[4] J. Hunter, D. Dale, E. Firing, and M. Droettboom, *Matplotlib*. [Online]. Available: `https://matplotlib.org/index.html`.

[5] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning from Theory to Algorithms*. Cambridge University, 2014.

[6] S. Rogers and M. Girolami, *A first course in Machine Learning*. CRC Press, 2017.