CS-EJ3211 Machine Learning with Python

Session 2 - Linear Regression

Shamsi Abdurakhmanova

Aalto University
FITech

27.01.22

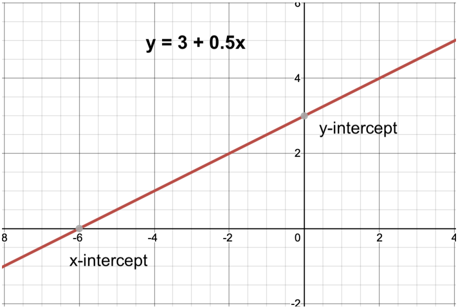
Linear Regression

$$y = b + wx$$

- b - bias, intercept, y-intercept
- w - slope, "rise over run", weight, coefficient
- x - feature, predictor, independent variable
- y - label, outcome, response variable, dependent variable

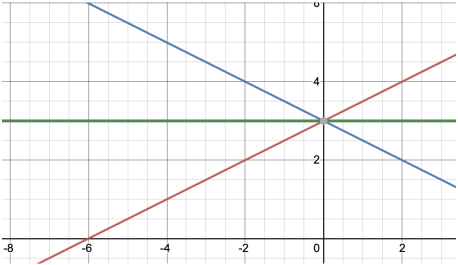
Linear Regression - example

$$slope = \frac{rise}{run} = \frac{y_1 - y_2}{x_1 - x_2} = \frac{0-3}{-6-0} = 0.5$$



Linear Regression - slope

Negative slope $w = -3$
Zero slope $w = 0$
Positive slope $w = 3$



Multiple Linear Regression

$$y = b + w_1 x_1 + w_2 x_2 + ... + w_n x_n$$

$$y = b + \mathbf{w}^T \mathbf{x}$$

- b - bias, intercept, y-intercept
- **w** - vector of coefficients $w_1, w_2, ..., w_n$
- **x** - vector of features $x_1, x_2, ..., x_n$

Multiple Linear Regression - Matrix form

For all (m) data points:

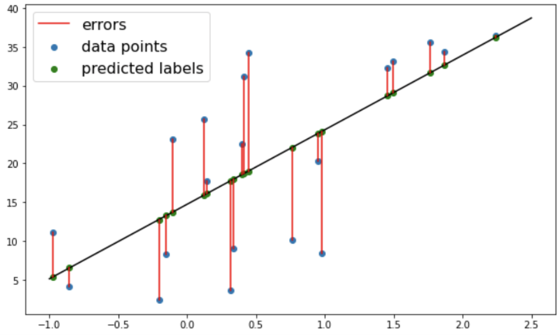
$$y = b + \mathbf{X}\mathbf{w}$$

- b - bias (scalar)
- $\mathbf{w}$ - ($n \times 1$) weight vector
- $\mathbf{X}$ - ($m \times n$) feature matrix
- $\mathbf{y}$ - ($m \times 1$) label vector

fitting the Linear Regression

Objective - find a hypothesis (model) of a form $h(x) = b + wx$, such that predictions $h(x) = \hat{y}$ are very close to "true" label $\hat{y} \approx y$.

$(y - \hat{y})$ is called prediction error or residual.

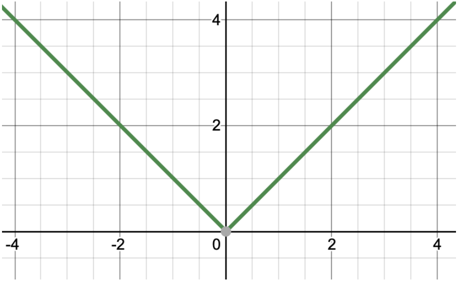


fitting the Linear Regression - MAE

Objective - find a hypothesis (model) of a form $h(x) = b + wx$, such that predictions $h(x) = \hat{y}$ are very close to "true" label $\hat{y} \approx y$.

Loss - Minimize (average) absolute prediction error $|(y - \hat{y})|$ or mean absolute error (MAE):

$$MAE = \frac{1}{m}\sum_{i=1}^{m} |(y^i - \hat{y}^i)|$$

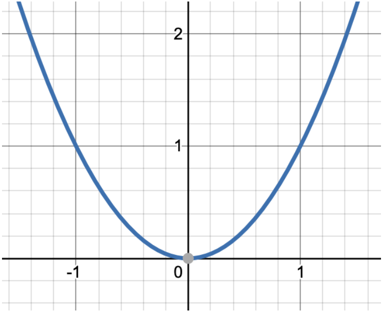


fitting the Linear Regression - MSE

Objective - find a hypothesis (model) of a form $h(x) = b + wx$, such that predictions $h(x) = \hat{y}$ are very close to "true" label $\hat{y} \approx y$.

Loss - Minimize (average) squared prediction error $(y - \hat{y})^2$ or mean squared error (MSE):

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



fitting Regression - MAE vs MSE

MAE

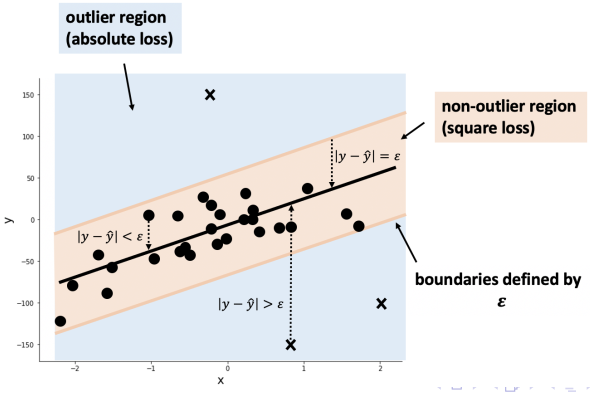
- robust to outliers
- non smooth (non-differentiable at $x = 0$) - difficult to optimize
- no analytical solution

MSE

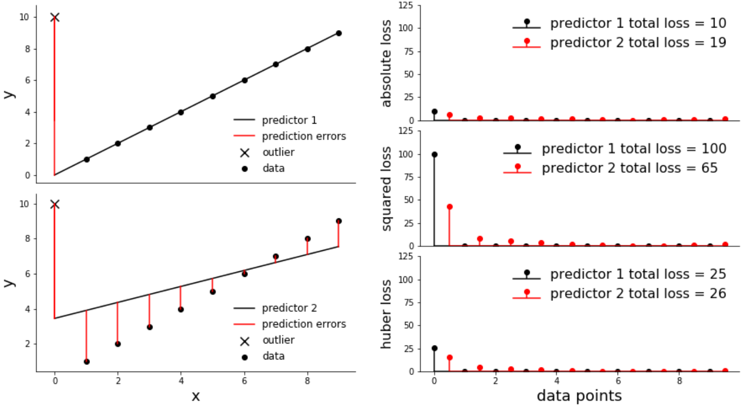
- not robust to outliers
- smooth - easy to optimize
- analytical solution exists (OLS)

fitting the Linear Regression - Huber Loss

$$\text{Huber Loss} = \begin{cases} (1/2)(y-\hat{y})^2 & \text{for } |y-\hat{y}| \leq \varepsilon \\ \varepsilon(|y-\hat{y}| - \varepsilon/2) & \text{else.} \end{cases}$$



fitting the Linear Regression - Huber Loss



Prediction and Inference

Prediction - find hypothesis with best prediction "accuracy".
Inference - understand the relationship between independent x and dependent y variables.

Assumptions for linear regression

$$y = b + wx + \epsilon$$

- y is real number
- x and y have approximately linear relationship
- samples/ observations are independent
- residual/ error is $\epsilon \sim \mathcal{N}(0, \sigma^2)$

Student tasks - NumPy array slicing

```
# Choose first (r+1) cols
# 1st col, r=0
X[:,:1]

# 1st,2nd cols, r=1
X[:,:2]

# 1st,2nd,3rd cols, r=2
X[:,:3]
```

```
# Choose rows
# 1st row
X[:1]

# 1st,2nd rows
X[:2]

# 1st,2nd,3rd rows
X[:3]
```

Student tasks - For loop

```python
for r in range(5):
    # use `r` to slice array
    print(r)
```

```
0
1
2
3
4
```