universität freiburg

Assignment 02

Machine Learning WS23/24

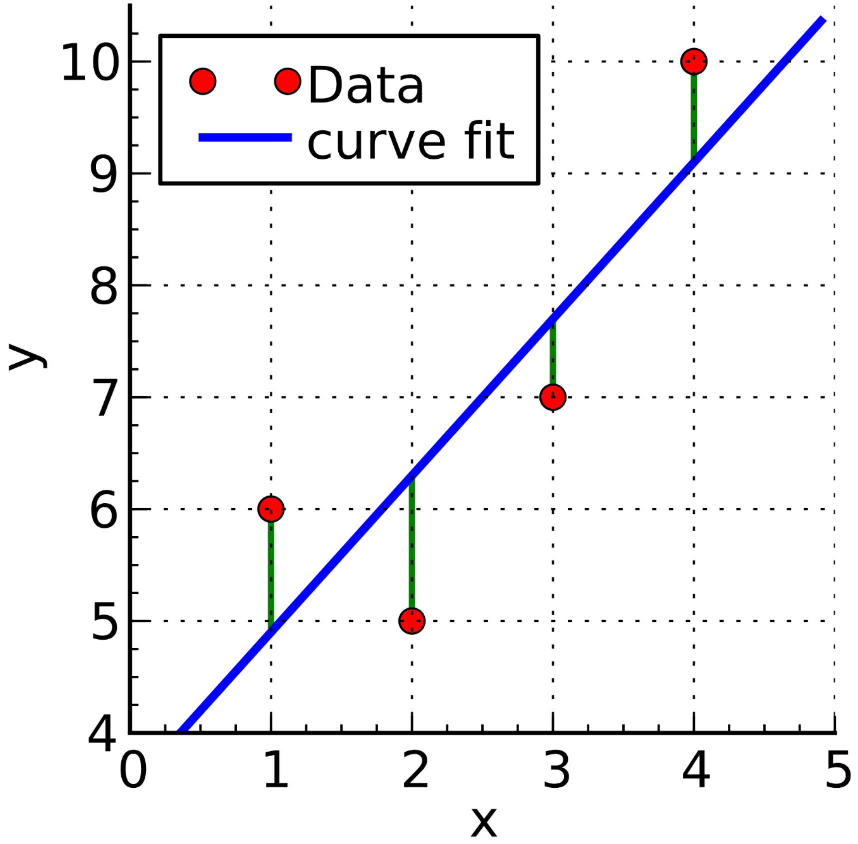
Assignment 02
Solution

1. Linear Regression
2. Logistic Regression

Linear Regression

Linear Regression

Linear regression models the relationship between a dependent variable $\boldsymbol{Y}$ and one or more independent variables $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}\}$ by fitting a linear equation of the form $\boldsymbol{Y} = \boldsymbol{w_0} + \boldsymbol{w_1 x_1} + \boldsymbol{w_2 x_2} + \dots + \boldsymbol{w_N x_N} + \boldsymbol{\epsilon}$. Here, $\boldsymbol{w_0}$ is the y-intercept, $\boldsymbol{w_1}, \boldsymbol{w_2}, \dots, \boldsymbol{w_N}$, are the coefficients of the predictors $\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_N}$, and $\boldsymbol{\epsilon}$ represents the error term. The goal of linear regression is to find the best-fit line that minimizes the sum of squared residuals (the differences between observed and predicted values) across all observations in the dataset.



Linear Regression

Use Linear Regression to solve for the parameters.
Do not include the bias term. Use the closed-form solution.
Further calculate the changes for a person with Age = 40, BMI = 32.5.

Given w are the parameters (no intercept),
X are the input features (design matrix)
and y are the labels, the closed form solution is:

$$w = (X^T X)^{-1} X^T y$$

Predictions are done like this:

$$\hat{y} = Xw$$

```python
import numpy as np

w = np.linalg.inv(X.T@X)@X.T@y
y_test_pred = X_test@w
```

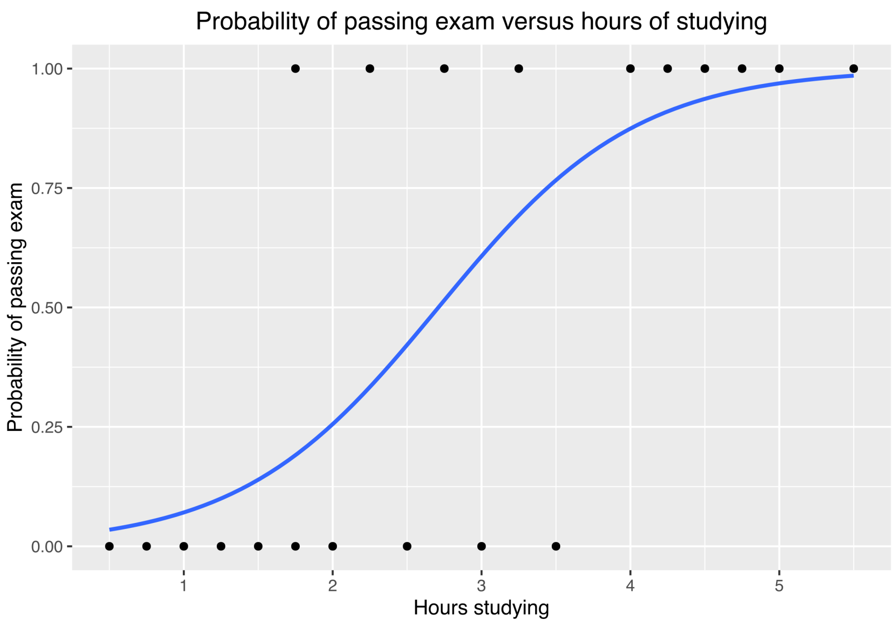
```python
from sklearn import linear_model

reg = linear_model.LinearRegression(
    fit_intercept=False
)
reg.fit(X, y)
y_test_pred = reg.predict(X_test)
```

Result: `7328.64906056`

Logistic Regression

Logistic Regression

Logistic regression models the probability that a binary dependent variable $\boldsymbol{Y}$ takes on the value 1 given one or more independent variables $X = \{x_1, x_2, \ldots, x_n\}$ with the form $P(\,Y = 1\,|X; w) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_N x_N)}}$. The function maps any input on the real line to a value between 0 and 1, suitable for a probability measure. The goal of logistic regression is to find the best parameters that maximize the likelihood of observing the given set of outcomes.



Logistic Regression

Starting from negative loglikelihood (binary cross entropy loss),

derive the update rule for w for gradient descent.

Ignore the bias term for the moment.

$$J = -\sum_{n=1}^{N} y_n log\, p_n + (1 - y_n) log(1 - p_n)$$

where $p_n = h_w(x_n) = P(\, y = 1 \mid X = x_n; w) = \sigma(x_n)$

$$\sigma(x_n) = \frac{e^{x_n w}}{1 + e^{x_n w}} = \frac{1}{1 + e^{-x_n w}}$$

Update rule:

$$w^{(t+1)} = w^{(t)} - \alpha * \frac{\partial J}{\partial w}$$

Logistic Regression

$$J = -\sum_{n=1}^{N} y_n \log p_n + (1 - y_n) \log(1 - p_n)$$

$$= -\sum_{n=1}^{N} y_n \log \frac{e^{x_n w}}{1 + e^{x_n w}} + (1 - y_n) \log(1 - \frac{e^{x_n w}}{1 + e^{x_n w}})$$

$$= -\sum_{n=1}^{N} y_n (x_n w - \log(1 + e^{x_n w})) + (1 - y_n) \log \frac{1 + e^{x_n w} - e^{x_n w}}{1 + e^{x_n w}}$$

$$= -\sum_{n=1}^{N} y_n (x_n w - \log(1 + e^{x_n w})) + (1 - y_n) \log \frac{1}{1 + e^{x_n w}}$$

Logistic Regression

$$J = -\sum_{n=1}^{N} y_n(x_n w - \log(1 + e^{x_n w})) + (1 - y_n) \log \frac{1}{1 + e^{x_n w}}$$

$$= -\sum_{n=1}^{N} y_n(x_n w - \log(1 + e^{x_n w})) + (1 - y_n) (-\log(1 + e^{x_n w}))$$

$$= -\sum_{n=1}^{N} y_n * x_n w - \log(1 + e^{x_n w})$$

Logistic Regression

$$\frac{\partial J}{\partial w} = \frac{\partial}{\partial w}\left(-\sum_{n=1}^{N} y_n * x_n\, w \;-\log(1+e^{x_n w})\right)$$

$$= -\sum_{n=1}^{N} y_n x_n - \frac{1}{1+\;e^{x_n w}} e^{x_n w} x_n$$

$$= -\sum_{n=1}^{N} x_n(y_n - p(y=1\mid x = x_n; w))$$

$$= -X^T(y \;- p)$$

where $p = \begin{pmatrix} p(y=1\mid X = x_1; w) \\ p(y=1\mid X = x_2; w) \\ \vdots \\ p(y=1\mid X = x_N; w) \end{pmatrix}$

Logistic Regression

Update rule:

$$w^{(t+1)} = w^{(t)} + \alpha * X^T(y - p)$$

where $p = \begin{pmatrix} p(y = 1 \mid X = x_1; w) \\ p(y = 1 \mid X = x_2; w) \\ \vdots \\ p(y = 1 \mid X = x_N; w) \end{pmatrix}$

Logistic Regression

Perform one gradient descent update step using the initial parameters as $w^{(0)} = (0 \quad 0 \quad 0)^T$ and learning rate $\alpha = 0.5$ using the following data:

Note: Think about a good way to include the bias in the data instead of deriving its update rule on its own.

x_0	x_1	x_2	y
1	2	4	1
1	3	4	1
1	-4	-2	0
1	-2	-6	0

Logistic Regression

Perform one gradient descent update step using the initial parameters as $w^{(0)} = (0 \quad 0 \quad 0)^T$ and learning rate $\alpha = 0.5$:

$$p_1 = \frac{1}{1 + \ e^{-(1*0+2*0+4*0)}}$$

$$= \frac{1}{1+1} = 0.5$$

similarly for all x_n. The outcome is 50% probability, which makes sense as we have not learned anything yet.

Logistic Regression

Perform one gradient descent update step using the initial parameters as $w^{(0)} = (0 \quad 0 \quad 0)^T$ and learning rate $\alpha = 0.5$:

$$w^{(t+1)} = w^{(t)} + \alpha * X^T(y - p)$$

$$X^T(y - p) = X^T\left(\begin{bmatrix}1\\1\\0\\0\end{bmatrix} - \begin{bmatrix}0.5\\0.5\\-0.5\\-0.5\end{bmatrix}\right) = \begin{bmatrix}0\\5.5\\7.5\end{bmatrix}$$

$$w^{(1)} = \begin{bmatrix}0\\0\\0\end{bmatrix} + 0.5 * \begin{bmatrix}0\\5.5\\7.5\end{bmatrix} = \begin{bmatrix}0\\2.75\\3.75\end{bmatrix}$$

Logistic Regression

Predict $P(y = 1 \mid X = [-1\ 1])$:

$$P(y = 1 \mid X = [-1\ 1]) = \frac{1}{1 + e^{-(1*0+(-1)*2.75+1*3.75)}}$$

$$= 0.731$$