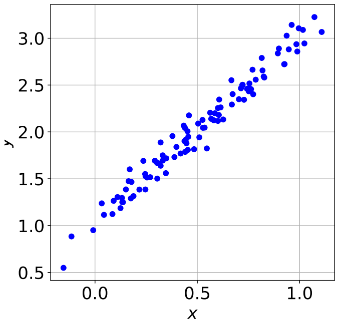
Linear Regression

September 9, 2017

Linear regression

Linear regression is the one of the simpliest approaches to supervised learning. It assumes the linear dependence of y on $x_1, x_2, \dots, x_p$.

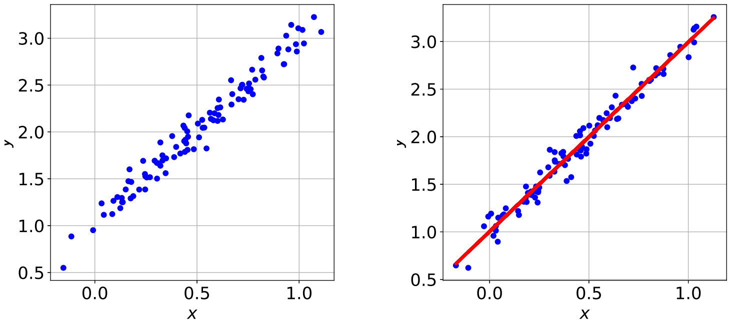
$$y = k\mathbf{x} + b, \quad \mathbf{x} = [x_1, x_2, \dots, x_p]. \tag{1}$$



Linear regression

Linear regression is the one of the simpliest approaches to supervised learning. It assumes the linear dependence of y on $x_1, x_2, \dots, x_p$.

$$y = k\mathbf{x} + b, \quad \mathbf{x} = [x_1, x_2, \dots, x_p]. \tag{1}$$



Model

We assume the linear behaviour for the parameters:

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \varepsilon, \tag{2}$$

where β_0 is the intercept and $\beta_1, \ldots, \beta_p$ are the coefficients.

Model

We assume the linear behaviour for the parameters:

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \varepsilon, \tag{2}$$

where β_0 is the intercept and $\beta_1, \ldots, \beta_p$ are the coefficients.

Given some estimates $\hat{\beta}_0, \ldots, \hat{\beta}_p$ for the model coefficients, we predict the future values for y as following:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_p x_p \tag{3}$$

Model

We assume the linear behaviour for the parameters:

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \varepsilon, \tag{2}$$

where β_0 is the intercept and $\beta_1, \ldots, \beta_p$ are the coefficients.

Given some estimates $\hat{\beta}_0, \ldots, \hat{\beta}_p$ for the model coefficients, we predict the future values for y as following:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_p x_p \tag{3}$$

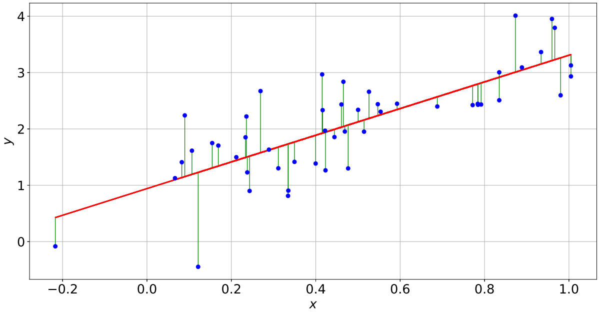
Residual sum of squares (RSS)

$$e_i = y_i - \hat{y}_i, \quad \sigma^2 = e_1^2 + \ldots e_p^2 \tag{4}$$

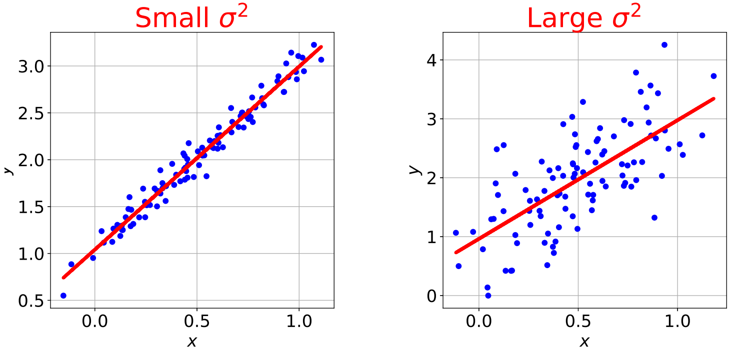
Distances

The main idea of the Linear Regression is to find the coefficients β_i in order to minimize σ^2:

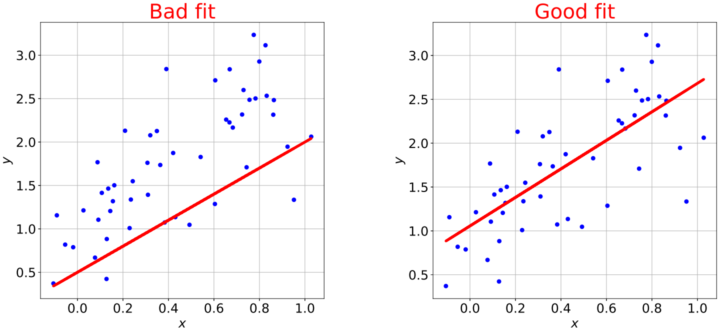
$$\min \sigma^2 = \sum_{i=1}^{p} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{p} e_i^2 \tag{5}$$



Least Squares



Good fit



Sklearn and pandas

```
#import libraries
from sklearn.linear_model import LinearRegression
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#load dataset
df = pd.read_csv('num.csv',index_col=0)
df.head()
```

	x0	x1	y
0	0.112464	0.186432	0.443470
1	0.331576	0.161625	0.612255

Fitting linear model

```
#create the model
lin = LinearRegression()

#fit the model
lin.fit(df[['x0','x1']],df['y'])

#coefficients and intercept
print('coef: %s, int: %s' % (lin.coef_, lin.intercept_))
```

coef: [1.01105486, 2.02920003], int: -0.00699671485255

Feature Selection

When all features are on the same scale, the most important once have the highest coefficients $(\beta_1, \ldots, \beta_p)$ in the model.

Feature Selection

When all features are on the same scale, the most important once have the highest coefficients $(\beta_1, \ldots, \beta_p)$ in the model.

$$y = 100x_0 + 50x_1 + 0.1x_2$$

Feature Selection

When all features are on the same scale, the most important once have the highest coefficients $(\beta_1, \ldots, \beta_p)$ in the model.

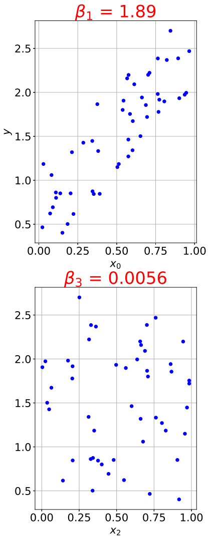
$$y = 100x_0 + 50x_1 + \cancel{0.1x_2}$$

Given a dataset:

	x_0	x_1	x_2	y
0	0.538975	0.222050	0.813623	1.438320
1	0.960462	0.173696	0.548318	2.042426
2	0.923627	0.556668	0.945707	2.503608
3	0.115383	0.791808	0.746768	1.165680
4	0.457036	0.605950	0.671633	1.716914

Given a dataset:

	x_0	x_1	x_2	y
0	0.538975	0.222050	0.813623	1.438320
1	0.960462	0.173696	0.548318	2.042426
2	0.923627	0.556668	0.945707	2.503608
3	0.115383	0.791808	0.746768	1.165680
4	0.457036	0.605950	0.671633	1.716914



$\beta_1 = 1.89$
$\beta_2 = 1.0$
$\beta_3 = 0.0056$
y
0.5
1.0
1.5
2.0
2.5
0.00
0.25
0.50
0.75
1.00
x_0
x_1
x_2

Possible problems

True model: $y = x_1 + x_2$,
We observe: $\hat{y} = x_1 + x_2 + \varepsilon$,
Variables $x_1 \approx x_2$.

Possible problems

True model: $y = x_1 + x_2$,
We observe: $\hat{y} = x_1 + x_2 + \varepsilon$,
Variables $x_1 \approx x_2$.

Depending on the noise ε we might be getting:

$$y = 2x_1, \tag{6a}$$
$$y = -x_1 + 3x_2. \tag{6b}$$

Possible problems

True model: $y = x_1 + x_2$,
We observe: $\hat{y} = x_1 + x_2 + \varepsilon$,
Variables $x_1 \approx x_2$.

Depending on the noise ε we might be getting:

$$y = 2x_1, \tag{6a}$$
$$y = -x_1 + 3x_2. \tag{6b}$$

The solution is to used **Regularized models**:

- L1 regularization/Lasso;
- L2 regularization/Ridge.

Regularized models

L1 regularization/Lasso

L1 regularization adds a penalty $\alpha \sum_{i=1}^{p} |\beta_i|$. It forces weak features to have zero coefficients:

$$\min \sigma^2 = \sum_{\mathbf{i=1}}^{\mathbf{p}} \mathbf{e_i} + \alpha \sum_{\mathbf{i=1}}^{\mathbf{p}} |\beta_\mathbf{i}|$$

Regularized models

L1 regularization/Lasso

L1 regularization adds a penalty $\alpha \sum_{i=1}^{p} |\beta_i|$. It forces weak features to have zero coefficients:

$$\min \sigma^2 = \sum_{i=1}^{p} e_i + \alpha \sum_{i=1}^{p} |\beta_i|$$

L2 regularization/Ridge

L2 regularization adds the **L2 norm** penalty to the loss function: $\alpha \sum_{i=1}^{p} \beta_i^2$. In this model the correlated features tend to get similar coefficients:

$$\min \sigma^2 = \sum_{i=1}^{p} e_i + \alpha \sum_{i=1}^{p} \beta_i^2$$

Summary

Lasso is more useful for selecting a strong subset of features for improving model performance. **Ridge** on the other hand can be used for data interpretation due to its stability.

In this presentation we covered:

- Introduction to the linear regression;
- The idea of the least squares method;
- What is the good fit;
- Feature selections with linear regression;
- Improving linear regression stability with Lasso and Ridge models.