Model Linear

Ali Akbar Septiandri

Universitas Al-Azhar Indonesia

aliakbars@live.com

April 1, 2020

Selayang Pandang

Bahan Bacaan

1. VanderPlas, J. (2016). Python Data Science Handbook. (In Depth: Linear Regression) `http://nbviewer.jupyter.org/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/05.06-Linear-Regression.ipynb`
2. Murray, I. (2016). MLPR class notes. (Linear Regression) `http://www.inf.ed.ac.uk/teaching/courses/mlpr/2016/notes/` (graduate level)
3. Septiandri, A.A. (2019). Artificial Intelligence Kuliah 2: Regresi Linear. Github.

Ulasan

Minggu lalu...

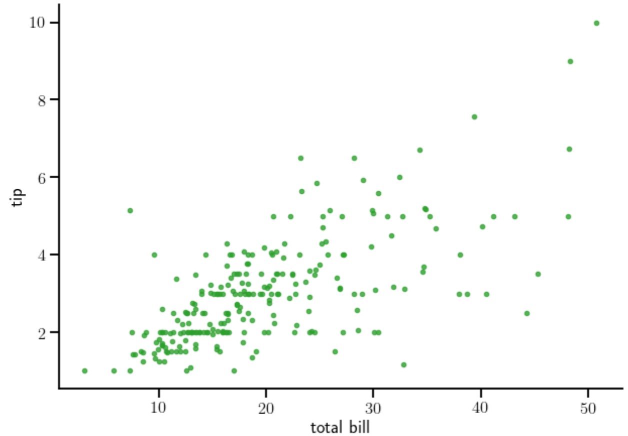
- Dimensionality Reduction
- Eigenvector & Eigenvalue
- Principal Component Analysis

Apa interpretasi dari determinan?
Apa hubungannya dengan nilai eigen?

Video dari Victor Lavrenko untuk PCA

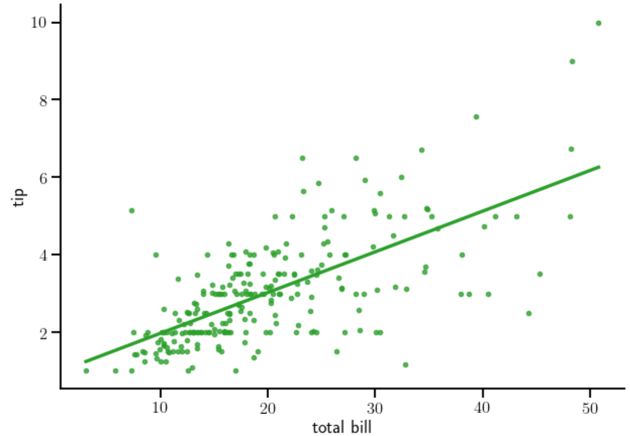
Regresi Linear

Prediksi hubungan antara dua variabel



Gambar: Data hubungan antara total harga pesanan dan tip yang diberikan

Prediksi hubungan antara dua variabel



Gambar: Data hubungan antara total harga pesanan dan tip yang diberikan

Simple Linear Regression

Fungsi linear

Kasus paling sederhana adalah mencocokkan garis lurus ke sekumpulan data

$$y = ax + b$$

dengan a adalah *slope*, sedangkan b dikenal dengan nama *intercept*.

Notasi lain

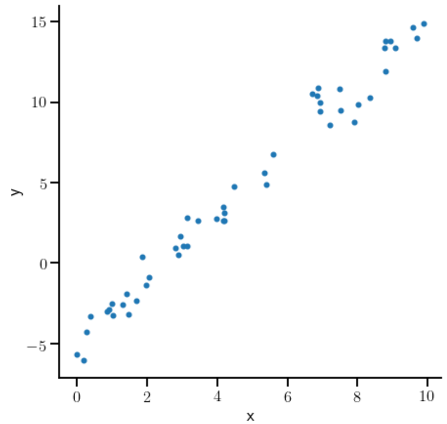
$$y = w_0 + w_1 x_1$$

dengan w adalah bobot atau koefisien.

Simple Linear Regression

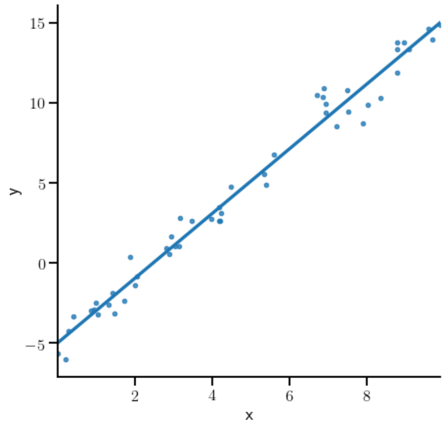
Example

```
rng = np.random.RandomState(1)
x = 10 * rng.rand(50)
y = 2 * x - 5 + rng.randn(50)
plt.scatter(x, y);
```



Gambar: Data yang dimunculkan secara acak [VanderPlas, 2016]

Mencocokkan Garis



Gambar: Hasil pencocokan garis [VanderPlas, 2016]

```
Model slope:  2.02720881036
Model intercept:  -4.99857708555
```

Bagaimana kalau ada lebih dari dua variabel yang ingin kita lihat hubungannya?

Multidimensional Linear Regression

Model

$$y = w_0 x_0 + w_1 x_1 + w_2 x_2 + ... + w_D x_D = \sum_{j=0}^{D} w_j x_j$$

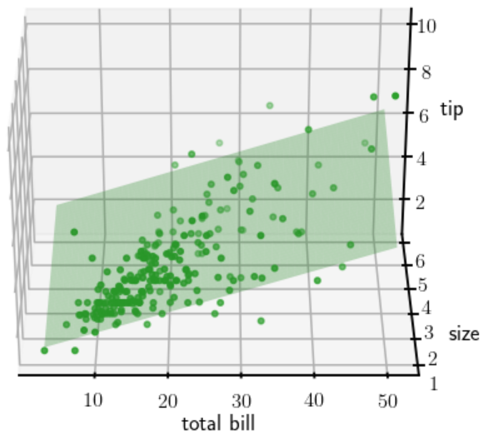
dengan $x_0 = 1$

Notasi matriks-vektor

$$y = \boldsymbol{\phi}\mathbf{w}$$

dengan $\boldsymbol{\phi} = (1, \mathbf{x}^T)$

Regresi linear untuk dua variabel



Gambar: Hubungan antara total bill dan jumlah tempat duduk (size) terhadap jumlah tip

Prediktor linear (contoh)

Vektor bobot $\mathbf{w} \in \mathbb{R}^D$
bias: 0.67
total bill: 0.09
size: 0.19

Vektor fitur $\phi(x) \in \mathbb{R}^D$
bias: 1
total bill: \$12.02
size: 2

$$\begin{aligned}\hat{y} &= \mathbf{w} \cdot \phi(x) \\ &= \sum_{j=1}^{D} w_j \phi_j(x) \\ &= 0.67(1) + 0.09(12.02) + 0.19(2) = 2.13\end{aligned}$$

Jadi, *diprediksi* bahwa untuk pelanggan dengan *total bill* $= \$12.02$ dan *size* $= 2$, pramusajinya akan mendapatkan $\approx \$2.13$.

Kita sudah tahu nilai y dan $\boldsymbol{\phi}$, tapi berapa nilai $\mathbf{w}$?

Nyatanya, kita tidak bisa mencari nilai $\boldsymbol{\phi}^{-1}$

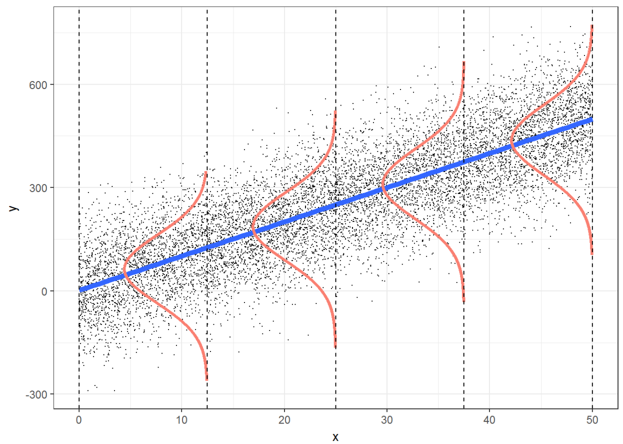
$\boldsymbol{\phi}$ bukan matriks bujur sangkar dan datanya mengandung *noise*

Asumsi Gaussian Noise

- Asumsikan $y = \mathbf{w}^T\boldsymbol{\phi} + \epsilon$ dengan $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$
- Berdasarkan asumsi distribusi Gaussian, implikasinya $p(y|\boldsymbol{\phi}, \mathbf{w}) = \mathcal{N}(y; \mathbf{w}^T\boldsymbol{\phi}, \sigma_\epsilon^2)$
- Dengan asumsi i.i.d., nilai log likelihood menjadi

$$\begin{aligned} L(\mathbf{w}) &= \sum_{i=1}^{N} \log p(y|\boldsymbol{\phi}, \mathbf{w}) \\ &= -\frac{N}{2}\log(2\pi\sigma_\epsilon^2) - \frac{1}{2\sigma_\epsilon^2}\sum_{i=1}^{N}(y_i - \mathbf{w}^T\boldsymbol{\phi}_i)^2 \end{aligned}$$

Fungsi Linear dengan Gaussian Noise



Gambar: Fungsi linear dengan Gaussian *noise* dalam asumsi *ordinary least squares*

Meminimalkan Error

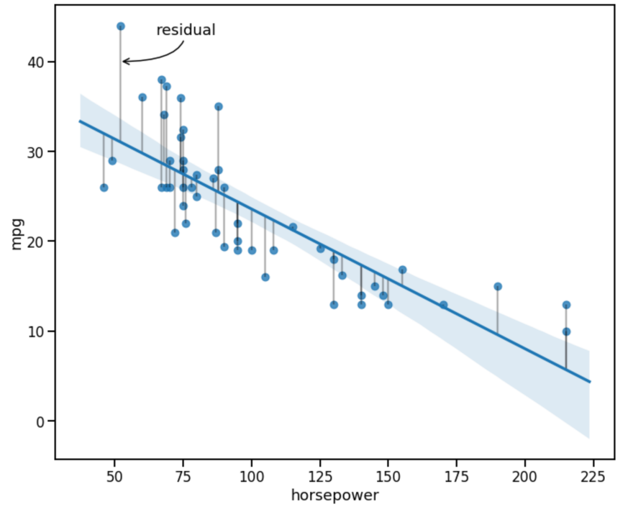
$$\begin{aligned} L(\mathbf{w}) &= -\frac{N}{2}\log(2\pi\sigma_\epsilon^2) - \frac{1}{2\sigma_\epsilon^2}\sum_{i=1}^{N}(y_i - \mathbf{w}^T\boldsymbol{\phi}_i)^2 \\ &= -C_2 - C_1\sum_{i=1}^{N}(y_i - \mathbf{w}^T\boldsymbol{\phi}_i)^2 \end{aligned}$$

dengan $C_1 > 0$ dan C_2 tidak terpengaruh oleh $\mathbf{w}$. Beberapa hal yang perlu diketahui:

- Mengalikan dengan konstanta positif tidak akan mengubah titik maksimum
- Menambahkan konstanta tidak mengubah titik maksimum
- $\sum_{i=1}^{N}(y_i - \mathbf{w}^T\boldsymbol{\phi}_i)^2$ adalah *sum of squared errors*

Jadi, memaksimalkan *likelihood* akan sama dengan meminimalkan *sum of squared error*.

Residual



Gambar: Meminimalkan *sum of squared errors*

Loss Function

- Harus menggunakan *loss function* $E(\mathbf{w})$ yang dapat diminimalkan
- Pilihan umum: *squared error*

$$E(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 = (\mathbf{y} - \boldsymbol{\phi}\mathbf{w})^T (\mathbf{y} - \boldsymbol{\phi}\mathbf{w})$$

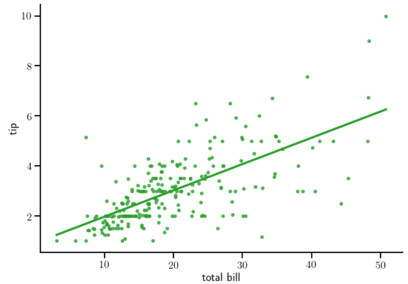
Solusi

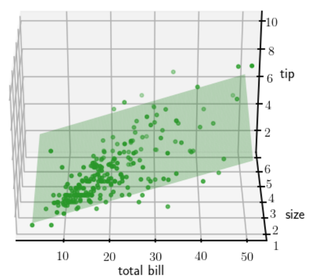
- Jawaban: Minimalkan $E(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$ dengan mencari turunan parsial yang diatur sama dengan 0
- Solusi analitis:

$$\hat{\mathbf{w}} = (\boldsymbol{\phi}^T \boldsymbol{\phi})^{-1} \boldsymbol{\phi}^T \mathbf{y}$$

- Bagian $(\boldsymbol{\phi}^T \boldsymbol{\phi})^{-1} \boldsymbol{\phi}^T$ dikenal sebagai *pseudo-inverse*

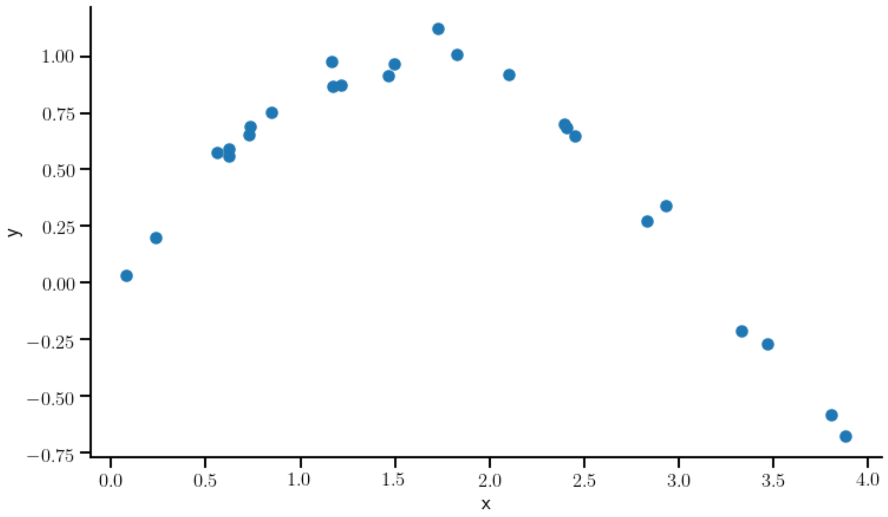
Perhatikan kembali





Apa kekurangan dari regresi linear sederhana seperti ini?

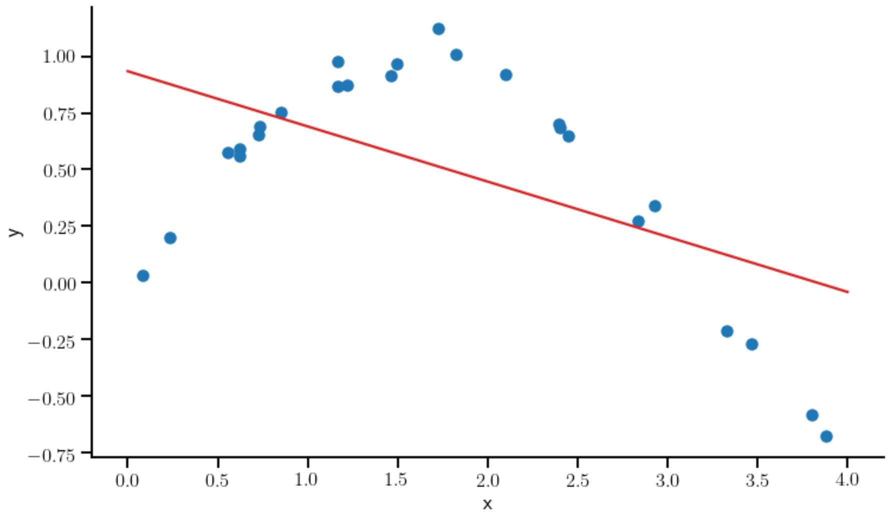
Non-linearity



Gambar: Data yang dihasilkan dari fungsi sin dengan *noise*

Bagaimana kalau datanya seperti ini?

Underfitting



Gambar: Hasil *fitting* regresi linear sederhana

Jika model yang dihasilkan lebih sederhana dibandingkan data yang seharusnya dicocokkan, maka model tersebut disebut mengalami underfitting.

Polynomial Basis Functions

Regresi linear dengan fungsi basis polinomial

Jika kita mengubah $x_p = f_p(x)$, dengan $f_p()$ adalah fungsi transformasi, maka untuk $f_p() = x^p$ dan x adalah input berdimensi satu, modelnya menjadi

$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \dots$$

Polynomial Basis Functions

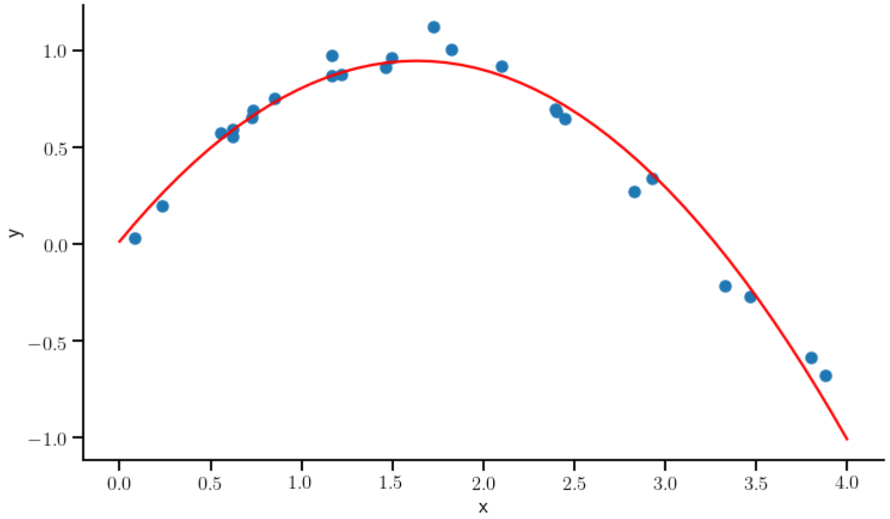
In

```
from sklearn.preprocessing import PolynomialFeatures
x = np.array([2, 3, 4])
poly = PolynomialFeatures(3, include_bias=False)
poly.fit_transform(x[:, None])
```

Out

```
array([[  2.,   4.,   8.],
       [  3.,   9.,  27.],
       [  4.,  16.,  64.]])
```

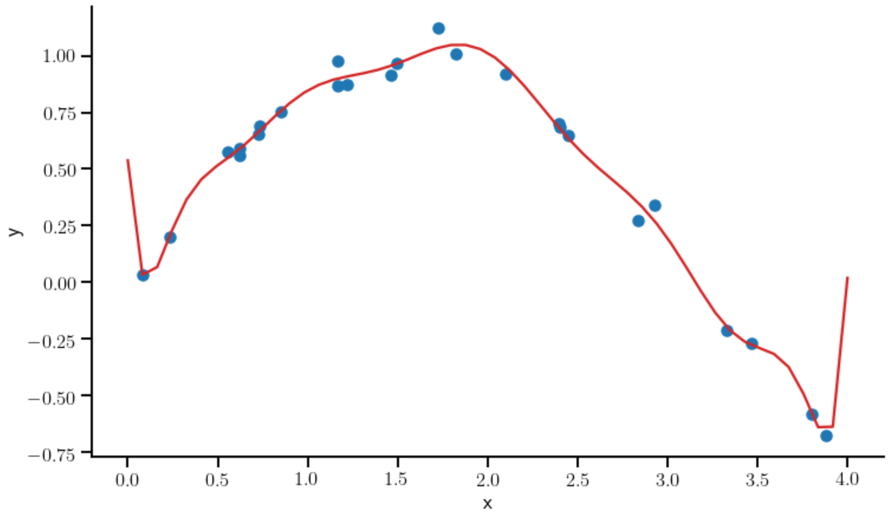
Best-fit



Gambar: Hasil *fitting* fungsi basis polinomial $p = 2$

Apa yang terjadi jika p dibuat lebih besar?

Overfitting



Gambar: Hasil *fitting* fungsi basis polinomial $p = 12$

Jika model yang dihasilkan lebih kompleks ($\sim$ parameternya banyak) dibandingkan data yang seharusnya dicocokkan, maka model tersebut disebut mengalami overfitting.

Kita dapat menggunkan fungsi basis Gaussian sebagai alternatif (*non-examinable*)

Bagaimana cara menghindari *overfitting*?

Ridge Regression

- Digunakan untuk menghindari *overfitting*
- Dikenal juga sebagai L_2 *regularisation* atau *Tikhonov regularisation*
- Pemberian penalti untuk koefisien model

$$P = \alpha \sum_{j=1}^{p} w_j^2 = \alpha \|\mathbf{w}\|_2^2$$

Loss Function pada Ridge Regression

- *Loss function* yang harus diminimalkan menjadi

$$E(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \alpha \|\mathbf{w}\|_2^2$$

dengan $\|\mathbf{w}\|_d = (\sum_{j=1}^{p} |w_j|^d)^{\frac{1}{d}}$

Loss Function pada Ridge Regression

- *Loss function* yang harus diminimalkan menjadi

$$E(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \alpha \|\mathbf{w}\|_2^2$$

dengan $\|\mathbf{w}\|_d = (\sum_{j=1}^{p} |w_j|^d)^{\frac{1}{d}}$

- Parameter α (terkadang juga ditulis sebagai λ) bernilai bebas (ditentukan oleh pengguna)

Loss Function pada Ridge Regression

- *Loss function* yang harus diminimalkan menjadi

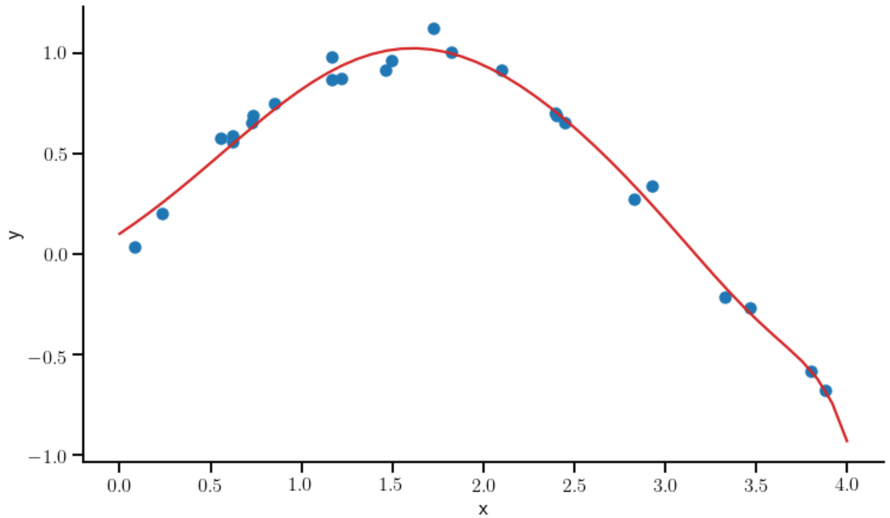
$$E(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \alpha \|\mathbf{w}\|_2^2$$

dengan $\|\mathbf{w}\|_d = (\sum_{j=1}^{p} |w_j|^d)^{\frac{1}{d}}$

- Parameter α (terkadang juga ditulis sebagai λ) bernilai bebas (ditentukan oleh pengguna)
- Solusi analitis:

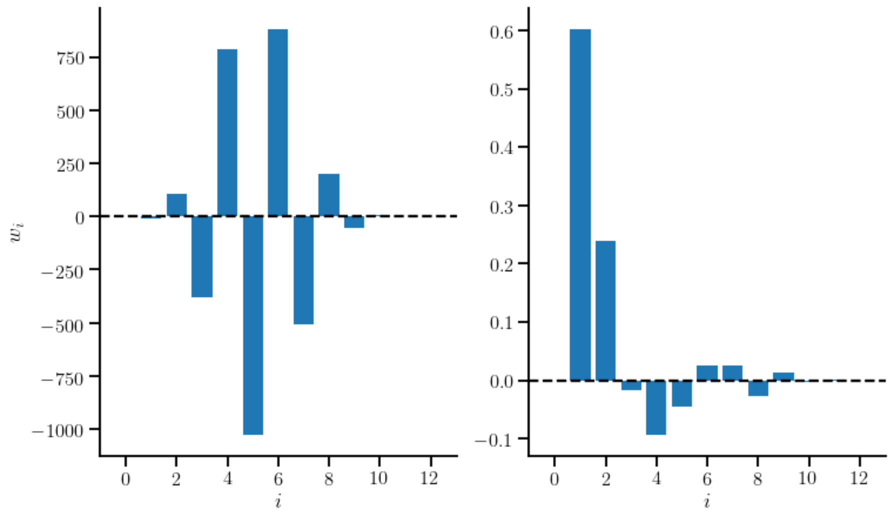
$$\hat{\mathbf{w}} = (\boldsymbol{\phi}^T \boldsymbol{\phi} + \alpha I_p)^{-1} \boldsymbol{\phi}^T \mathbf{y}$$

Ridge Regression



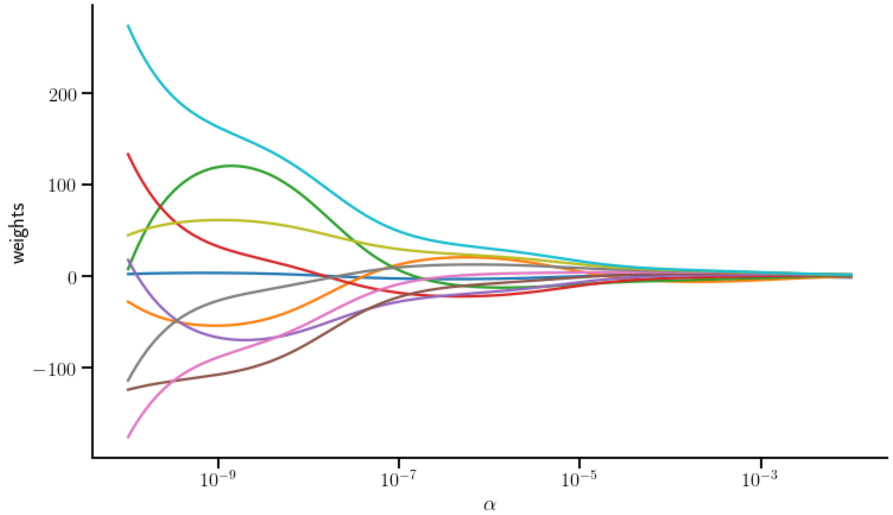
Gambar: *Fitting* fungsi basis polinomial $p = 12$ dengan Ridge $\alpha = 0.1$

Perubahan Koefisien



Gambar: Dengan ridge regression, koefisien diubah menjadi kecil

Perubahan w terhadap α



Gambar: Semakin besar α, w semakin kecil (sumber: scikit-learn)

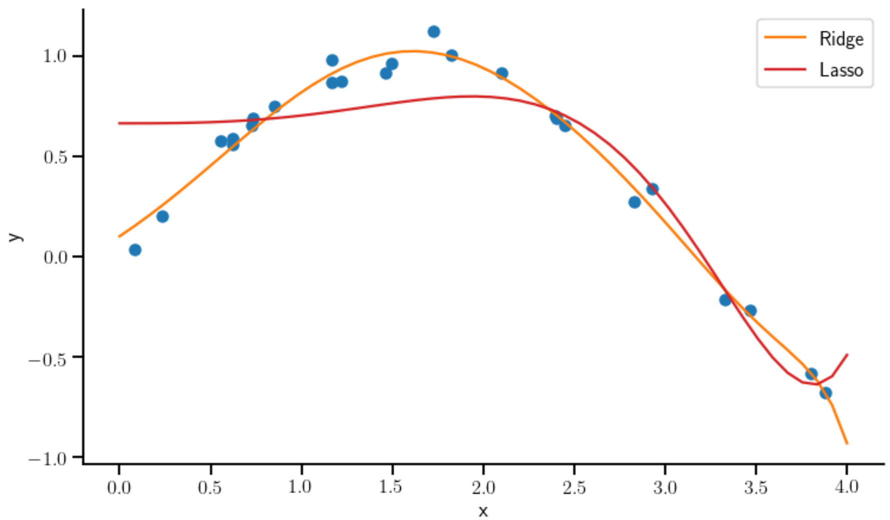
Lasso Regression

- Secara konsep mirip seperti *ridge regression*
- Penalti dengan jumlah nilai absolut dari koefisien (1-norms; L_1 *regularisation*)

$$P = \alpha \sum_{j=1}^{p} |w_j|$$

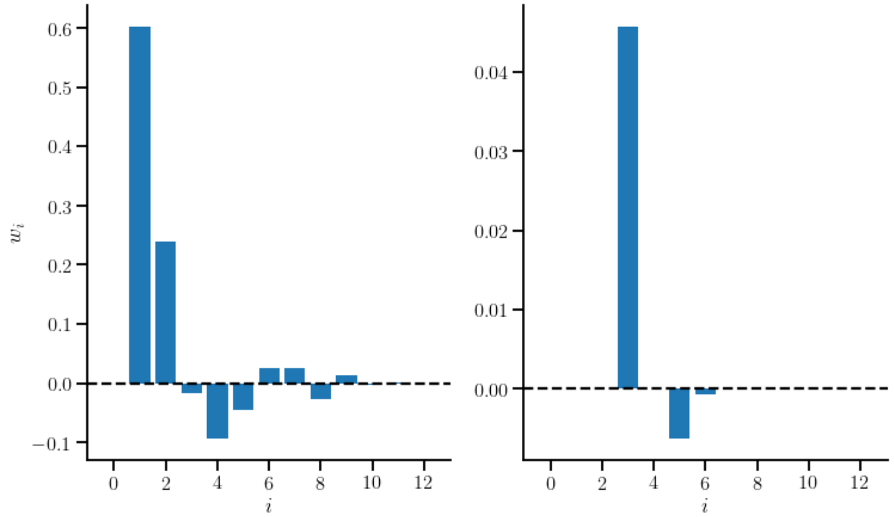
- Bekerja dengan membuat banyak koefisien bernilai nol

Ridge vs Lasso Regression



Gambar: Perbandingan Ridge dan Lasso dengan $\alpha = 0.1$

Perubahan Koefisien



Gambar: Perbandingan Ridge dan Lasso

Ikhtisar

- Regresi linear sebagai fungsi dengan Gaussian *noise*
- Asumsi Gaussian noise → *sum of squared error*
- *Ordinary least squares* (OLS) didapatkan dengan solusi analitis dari fungsi error
- Transformasi fitur dan regularisasi

Referensi

Jake VanderPlas (2016)
In Depth: Linear Regression
Python Data Science Handbook

Terima kasih