SELAYANG PANDANG

Model Linear

Ali Akbar Septiandri

Universitas Al-Azhar Indonesia aliak bars@live.com

March 30, 2020

1 Ulasan

2 Regresi Linear

Simple Linear Regression Basis Function Regression Regularisation



BAHAN BACAAN

- 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

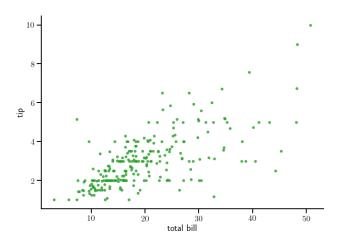
Apa interpretasi dari determinan? Apa hubungannya dengan nilai eigen?

◆□▶ ◆□▶ ◆■▶ ■ り९@

Video dari Victor Lavrenko untuk PCA

REGRESI LINEAR

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

$$y = ax + b$$

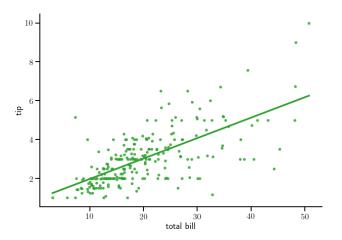
dengan a adalah slope, sedangkan b dikenal dengan nama intercept.

Notasi lain

$$y = w_0 + w_1 x_1$$

dengan w adalah bobot atau koefisien.

Prediksi hubungan antara dua Variabel



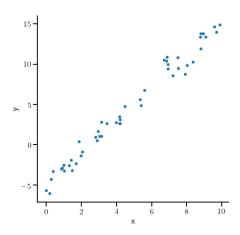
GAMBAR: Data hubungan antara total harga pesanan dan tip yang diberikan



SIMPLE LINEAR REGRESSION

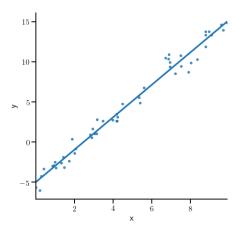
EXAMPLE

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



MULTIDIMENSIONAL LINEAR REGRESSION

Model

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

dengan $x_0 = 1$

Notasi matriks-vektor

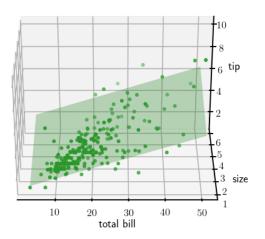
$$y = \phi \mathbf{w}$$

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

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



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

$$\hat{y} = \mathbf{w} \cdot \phi(x)$$

$$= \sum_{j=1}^{D} w_j \phi_j(x)$$

$$= 0.67(1) + 0.09(12.02) + 0.19(2) = 2.13$$

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

4□ > 4□ > 4□ > 4□ > 4□ > 4□ > 9

Kita sudah tahu nilai y dan ϕ , tapi berapa nilai \mathbf{w} ?



Nyatanya, kita tidak bisa mencari nila
i ϕ^{-1}

 ϕ bukan matriks bujur sangkar dan datanya mengandung noise

Asumsi Gaussian Noise

- Asumsikan $y = \mathbf{w}^T \boldsymbol{\phi} + \epsilon \text{ dengan } \epsilon \sim \mathcal{N}(0, \sigma_{\epsilon}^2)$
- 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

$$L(\mathbf{w}) = \sum_{i=1}^{N} \log p(y|\boldsymbol{\phi}, \mathbf{w})$$
$$= -\frac{N}{2} \log(2\pi\sigma_{\epsilon}^{2}) - \frac{1}{2\sigma_{\epsilon}^{2}} \sum_{i=1}^{N} (y_{i} - \mathbf{w}^{T} \boldsymbol{\phi}_{i})^{2}$$



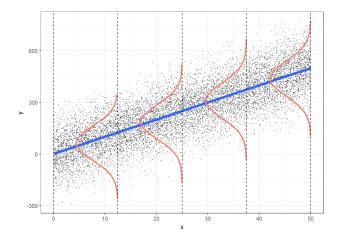
MEMINIMALKAN ERROR

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

dengan $C_1 > 0$ dan C_2 tidak terpengaruh oleh **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

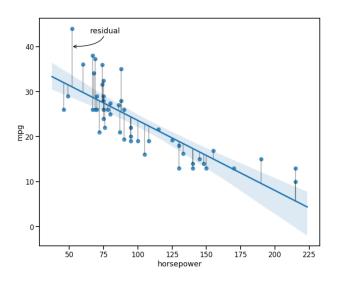
Fungsi Linear dengan Gaussian Noise



GAMBAR: Fungsi linear dengan Gaussian noise dalam asumsi ordinary least squares



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



Gambar: Meminimalkan sum of squared errors



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:

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

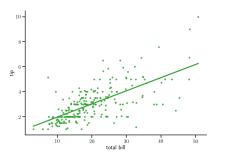
- Bagian $(\pmb{\phi}^T \pmb{\phi})^{-1} \pmb{\phi}^T$ dikenal sebagai pseudo-inverse

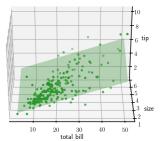
- Harus menggunakan loss function $E(\mathbf{w})$ yang dapat diminimalkan
- Pilihan umum: squared error

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



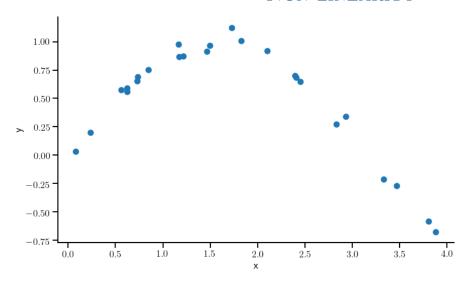
PERHATIKAN KEMBALI





Apa kekurangan dari regresi linear sederhana seperti ini?

Non-Linearity



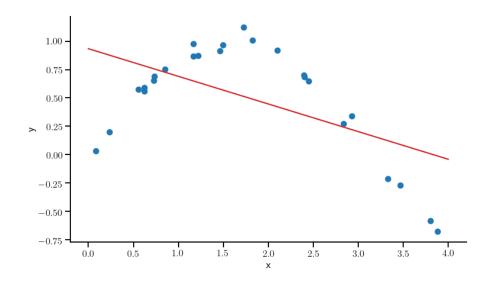
Gambar: Data yang dihasilkan dari fungsi sin dengan noise

Bagaimana kalau datanya seperti ini?



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

Underfitting



Gambar: Hasil fitting regresi linear sederhana



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

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

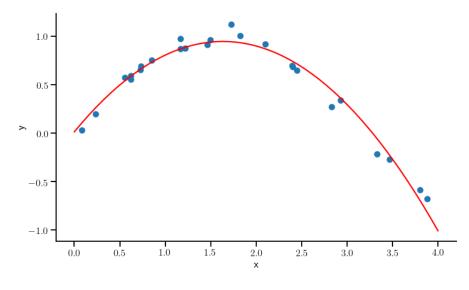
IN

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

OUT



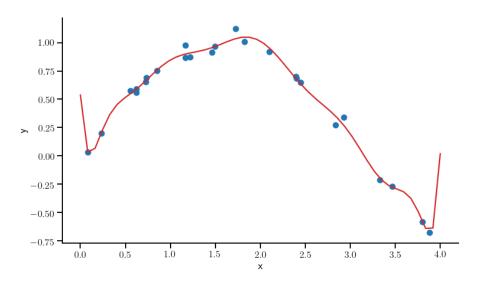
Apa yang terjadi jika p dibuat lebih besar?



Gambar: Hasil fitting fungsi basis polinomial p=2



OVERFITTING



Gambar: Hasil fitting fungsi basis polinomial p = 12

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

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





RIDGE REGRESSION

Bagaimana cara menghindari overfitting?

- 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)
- Solusi analitis:

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

Loss Function pada Ridge Regression

• Loss function yang harus diminimalkan menjadi

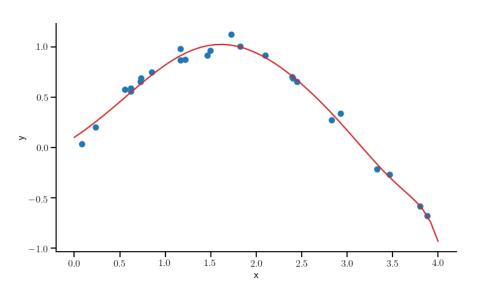
$$E(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 - \alpha \|\mathbf{w}\|_2^2$$

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)

◆ロト ◆卸 ▶ ◆ 差 ▶ ◆ 差 ● かへ

RIDGE REGRESSION

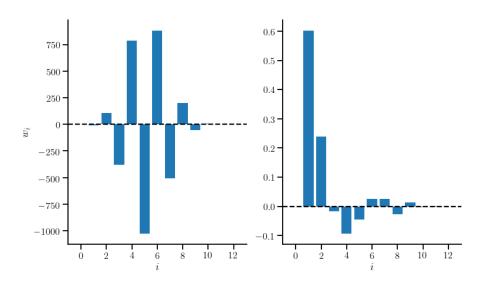


Gambar: Fitting fungsi basis polinomial p = 12 dengan Ridge $\alpha = 0.1$





PERUBAHAN KOEFISIEN



GAMBAR: Dengan ridge regression, koefisien diubah menjadi kecil



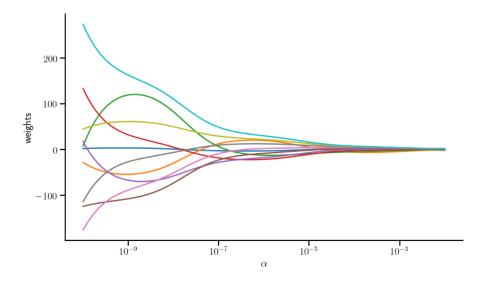
LASSO REGRESSION

- 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

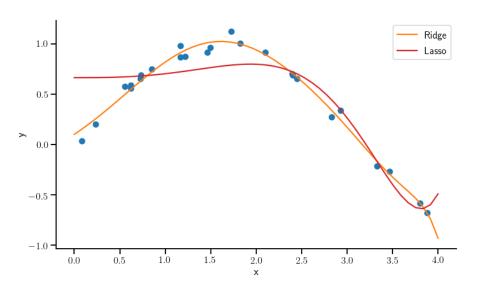
Perubahan w terhadap α



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



RIDGE VS LASSO REGRESSION

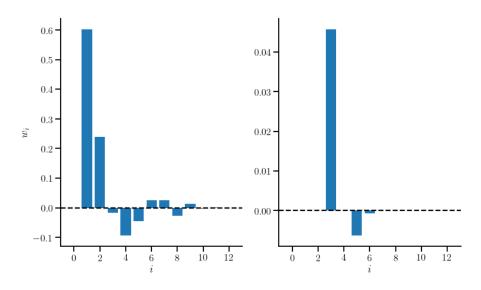


Gambar: Perbandingan Ridge dan Lasso dengan $\alpha=0.1$



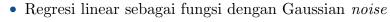
PERUBAHAN KOEFISIEN

IKHTISAR



GAMBAR: Perbandingan Ridge dan Lasso

Referensi



- Asumsi Gaussian noise \rightarrow sum of squared error
- Ordinary least squares (OLS) didapatkan dengan solusi analitis dari fungsi error
- Transformasi fitur dan regularisasi

Terima kasih

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