ALJABAR LINIER UNTUK SAINS DATA
Pertemuan 15

Egi Safitri, S.Mat., M.Si
Program Studi Sains Data
Fakultas Ilmu Komputer
Institut Informatika dan Bisnis Darmajaya, Bandar Lampung

2024

1. Aljabar Linier dan Regresi Linier
2. Klasifikasi Menggunakan Metode Vektor Support (SVM)
3. Jaringan Saraf Tiruan (Neural Networks) dan Aljabar Linier

Pendahuluan

Regresi linear adalah metode statistik untuk memodelkan hubungan antara variabel dependen y dan satu atau lebih variabel independen x. Dalam bentuk yang paling sederhana, hubungan ini dapat dinyatakan sebagai:

$$y = \beta_0 + \beta_1 x + \epsilon$$

Model Regresi Linear

Model regresi linear dapat ditulis dalam bentuk matriks sebagai berikut:

$$\mathbf{y} = X\beta + \epsilon$$

di mana:

- $\mathbf{y}$ adalah vektor nilai dependen (ukuran $n \times 1$)
- X adalah matriks desain (ukuran $n \times p$)
- β adalah vektor koefisien (ukuran $p \times 1$)
- ϵ adalah vektor error (ukuran $n \times 1$)

Matriks Desain

Matriks desain X biasanya termasuk kolom konstanta (intersep) dan kolom variabel independen:

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}$$

Solusi Regresi Linear

Tujuan dari regresi linear adalah untuk menemukan estimasi terbaik dari β, yang meminimalkan jumlah kuadrat error:

$$\hat{\beta} = \arg\min_{\beta} \|\mathbf{y} - X\beta\|^2$$

Solusi dengan Aljabar Linear

Solusi dari persamaan di atas dapat ditemukan dengan menggunakan aljabar linear:

$$\hat{\beta} = (X^T X)^{-1} X^T \mathbf{y}$$

Contoh Soal

Diberikan data berikut:

$$X = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 6 \\ 5 \\ 7 \\ 10 \end{pmatrix}$$

Cari estimasi β menggunakan metode regresi linear.

Langkah 1: Hitung $X^T X$

$$X^T X = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{pmatrix} = \begin{pmatrix} 4 & 10 \\ 10 & 30 \end{pmatrix}$$

Langkah 2: Hitung $X^T\mathbf{y}$

$$X^T\mathbf{y} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{pmatrix} \begin{pmatrix} 6 \\ 5 \\ 7 \\ 10 \end{pmatrix} = \begin{pmatrix} 28 \\ 70 \end{pmatrix}$$

Langkah 3: Hitung $\hat{\beta}$

$$\hat{\beta} = (X^T X)^{-1} X^T \mathbf{y}$$

Hitung invers dari $X^T X$:

$$(X^T X)^{-1} = \begin{pmatrix} 4 & 10 \\ 10 & 30 \end{pmatrix}^{-1} = \frac{1}{(4)(30) - (10)(10)} \begin{pmatrix} 30 & -10 \\ -10 & 4 \end{pmatrix} = \left(\begin{matrix} 1.5 & -0 \\ -0.5 & 0 \end{matrix} \right.$$

$$\hat{\beta} = \begin{pmatrix} 1.5 & -0.5 \\ -0.5 & 0.2 \end{pmatrix} \begin{pmatrix} 28 \\ 70 \end{pmatrix} = \begin{pmatrix} 1.5 \cdot 28 + (-0.5) \cdot 70 \\ -0.5 \cdot 28 + 0.2 \cdot 70 \end{pmatrix} = \begin{pmatrix} -7 \\ 3 \end{pmatrix}$$

Kesimpulan

- Dengan menggunakan aljabar linear, kita dapat menyelesaikan masalah regresi linear.
- Estimasi koefisien β diperoleh dengan $\hat{\beta} = (X^T X)^{-1} X^T \mathbf{y}$.
- Dalam contoh ini, estimasi koefisien adalah $\hat{\beta} = \begin{pmatrix} -7 \\ 3 \end{pmatrix}$.

Pengenalan SVM

Support Vector Machine (SVM) adalah salah satu metode klasifikasi yang digunakan dalam machine learning untuk mencari hyperplane terbaik yang memisahkan data ke dalam kelas yang berbeda.

Prinsip Dasar SVM

SVM bekerja dengan mencari hyperplane yang memaksimalkan margin antara dua kelas data. Margin adalah jarak terdekat antara titik data dari kedua kelas dan hyperplane.

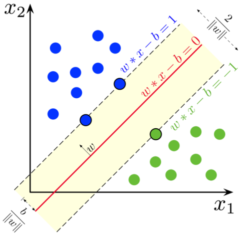

Figure: Hyperplane dan Margin pada SVM

Hyperplane

Hyperplane adalah suatu bidang dalam ruang fitur yang digunakan untuk memisahkan data ke dalam kelas yang berbeda.

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

di mana **w** adalah vektor bobot dan b adalah bias.

Fungsi Objektif SVM

Fungsi objektif SVM adalah untuk memaksimalkan margin dengan meminimalkan $\|\mathbf{w}\|$:

$$\min \frac{1}{2}\|\mathbf{w}\|^2$$

dengan kendala:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \forall i$$

di mana y_i adalah label kelas dari titik data $\mathbf{x}_i$.

SVM dengan Soft Margin

Untuk data yang tidak dapat dipisahkan secara sempurna, digunakan SVM dengan soft margin yang memperkenalkan variabel slack ξ_i:

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

dengan kendala:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i$$

Kernel SVM

Kernel SVM memungkinkan pemetaan data ke dimensi yang lebih tinggi untuk menangani kasus data yang tidak linier.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

Beberapa kernel yang umum digunakan:

- Kernel Linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$
- Kernel Polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$
- Kernel Gaussian (RBF): $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$

Kelebihan dan Kelemahan SVM

- Kelebihan:
    - Efektif dalam ruang berdimensi tinggi.
    - Masih efektif jika jumlah dimensi lebih besar dari jumlah sampel.
    - Menggunakan subset titik pelatihan (support vectors), sehingga hemat memori.
- Kelemahan:
    - Tidak berkinerja baik jika data memiliki banyak noise.
    - Pemilihan kernel yang tepat bisa sulit.
    - Kecepatan pelatihan lambat untuk dataset besar.

Contoh Implementasi SVM

- Langkah 1: Import library dan dataset
- Langkah 2: Pisahkan data menjadi pelatihan dan pengujian
- Langkah 3: Latih model SVM dengan kernel yang dipilih
- Langkah 4: Evaluasi kinerja model

Contoh Kode Implementasi SVM

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report

# Langkah 1: Import dataset
iris = datasets.load_iris()
X = iris.data
y = iris.target

# Langkah 2: Pisahkan data
X_train, X_test, y_train, y_test = train_test_split(X, y, test

# Langkah 3: Latih model
model = SVC(kernel='linear')
model.fit(X_train, y_train)
```

Kesimpulan

- SVM adalah metode klasifikasi yang efektif untuk berbagai aplikasi.
- SVM bekerja dengan mencari hyperplane yang memaksimalkan margin antara dua kelas data.
- Kernel SVM memungkinkan pemetaan data ke dimensi yang lebih tinggi untuk menangani data yang tidak linier.

Pengantar JST

Jaringan Saraf Tiruan (Neural Networks) adalah model komputasi yang terinspirasi oleh cara kerja otak manusia. Model ini sangat berguna dalam berbagai aplikasi, mulai dari pengenalan gambar hingga pemrosesan bahasa alami.

Struktur Dasar Jaringan Saraf Tiruan

- Neuron: Unit dasar yang memproses informasi.
- Lapisan (Layer): Kumpulan neuron yang bekerja bersama. Ada tiga jenis lapisan utama:
  - Lapisan Masukan (Input Layer)
  - Lapisan Tersembunyi (Hidden Layer)
  - Lapisan Keluaran (Output Layer)
- Koneksi dan Bobot: Koneksi antara neuron dengan bobot yang menentukan kekuatan sinyal.

Model Matematika Jaringan Saraf

Jaringan saraf tiruan dapat direpresentasikan dengan persamaan aljabar linier. Misalkan ada satu lapisan tersembunyi, kita dapat menuliskan:

$$\mathbf{z} = W\mathbf{x} + \mathbf{b}$$

$$\mathbf{a} = \sigma(\mathbf{z})$$

- $\mathbf{x}$: Vektor masukan.
- W: Matriks bobot.
- $\mathbf{b}$: Vektor bias.
- σ: Fungsi aktivasi.

Fungsi Aktivasi

Fungsi aktivasi adalah fungsi non-linier yang diterapkan pada setiap neuron. Beberapa fungsi aktivasi populer:

- Sigmoid: $\sigma(z) = \frac{1}{1+e^{-z}}$
- ReLU (Rectified Linear Unit): $\sigma(z) = \max(0, z)$
- Tanh: $\sigma(z) = \tanh(z)$

Proses Pelatihan Jaringan Saraf

Pelatihan jaringan saraf melibatkan dua langkah utama:

- Forward Propagation: Menghitung keluaran jaringan dengan input yang diberikan.
- Backward Propagation: Menghitung gradien kesalahan dan memperbarui bobot untuk meminimalkan kesalahan.

$$\Delta W = -\eta \frac{\partial L}{\partial W}$$

- L: Fungsi kerugian (loss function).
- η: Laju pembelajaran (learning rate).

Contoh Aplikasi Jaringan Saraf

- Pengenalan Gambar: Mengidentifikasi objek dalam gambar.
- Pemrosesan Bahasa Alami: Penerjemahan bahasa, analisis sentimen.
- Prediksi Keuangan: Memprediksi harga saham.

Aljabar Linier dalam Jaringan Saraf Tiruan

Aljabar linier memainkan peran penting dalam operasi jaringan saraf, termasuk:

- Matriks bobot dan operasi matriks.
- Transformasi linier pada data.
- Optimisasi melalui gradien.

Matriks Bobot dan Operasi Matriks

Setiap lapisan dalam jaringan saraf memiliki matriks bobot yang menentukan hubungan antara neuron. Operasi matriks digunakan untuk menghitung keluaran dari lapisan-lapisan ini.

$$\mathbf{Z} = W\mathbf{X} + \mathbf{b}$$

Transformasi Linier pada Data

Transformasi linier digunakan untuk mengubah data masukan ke dalam bentuk yang dapat diproses oleh jaringan saraf.

$$\mathbf{y} = A\mathbf{x}$$

di mana A adalah matriks transformasi, $\mathbf{x}$ adalah vektor masukan, dan $\mathbf{y}$ adalah vektor keluaran.

Optimisasi melalui Gradien

Proses backward propagation melibatkan perhitungan gradien dari fungsi kerugian terhadap bobot untuk memperbarui bobot jaringan.

$$\nabla_W L = \frac{\partial L}{\partial W}$$

Kesimpulan

- Jaringan saraf tiruan adalah model komputasi yang kuat untuk berbagai aplikasi.
- Aljabar linier adalah dasar matematis yang penting untuk operasi jaringan saraf.
- Pemahaman aljabar linier membantu dalam merancang dan melatih jaringan saraf secara efektif.