Principal Component Analysis (PCA)

Salman Hanif - 13523056

Cara Kerja Algoritma

Algoritma Principal Component Analysis (PCA) bekerja dengan tujuan untuk melakukan reduksi pada dimensi data (secara kolom/feature). Reduksi feature pada PCA ini didasarkan pada tingkat important / pentingnya sebuah kolom. Tingkat important dari featur ini diukur dengan nilai skalar eigen yang didapatkan.

$$(1) \quad \left| \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = 0$$

$$(2) \quad \begin{vmatrix} 3-\lambda & 1 \\ 0 & 2-\lambda \end{vmatrix} = 0$$

$$(3) \quad (3-\lambda)(2-\lambda) = 0 \quad \lambda_1 = 3, \lambda_2 = 2$$

Alur kerja fungsi fit dan transform :

Karena tujuan dari PCA untuk mereduksi sebuah dataset, sebenarnya untuk dataset tunggal yang akan direduksi, praktik fit dan transform bisa digabung menjadi 1 fungsi tunggal saja seperti fit_transform.

Fit bertugas untuk menghitung semua parameter yang diperlukan oleh data :

1. Menghitung rata-rata (mean) dari setiap fitur dalam data pelatihan dan menggunakannya untuk memusatkan data.
2. Menghitung matriks kovarians untuk memahami bagaimana setiap fitur bervariasi satu sama lain.
3. Mencari vektor eigen dan nilai eigen dari matriks kovarians. Vektor eigen menunjukkan arah-arah utama variasi data, sedangkan nilai eigen menunjukkan seberapa besar variasi tersebut.

4. Mengurutkan vektor eigen berdasarkan nilai eigen dari yang terbesar, lalu memilih sejumlah vektor teratas yang akan menjadi komponen utama.

Transform untuk penerapan parameter untuk mengubah data :

1. Menggunakan rata-rata yang sama dari langkah fit untuk memusatkan data baru yang akan diproyeksikan.
2. Mengalikan data yang telah dipusatkan dengan matriks komponen utama yang ditemukan pada langkah fit. Ini akan mengubah data dari ruang dimensi aslinya menjadi ruang dimensi yang lebih rendah.

Hasil Evaluasi model dari hasil scratch dan dari library. Jelaskan perbedaan

Dalam evaluasi, model saya dapat bekerja dengan baik mereduksi sebuah dataset dengan pilihan jumlah n_feature yang ingin dipertahankan bisa ditulis langsung, atau menggunakan n_multiplier.

Perbedaan dengan PCA dari library, Sklearn menggunakan SVD dalam mengubah dataset ke vektor eigen dan nilai eigen. Sedangkan saya menggunakan fungsi np.linalg.eg, metode penghitungan dengan matriks kovarian.

Improvement yang bisa dilakukan

Penerapan SVD untuk mendapatkan nilai dan vektor eigen bisa dicoba dan dibandingkan kapabilitasnya dengan metode kovarians. Kemudian untuk data yang sangat besar, bisa dikembangkan incremental PCA (pemrosesan data secara batch) seperti yang dimiliki sklearn agar lebih efisien dan bisa digunakan komputer dengan kapasitas ram lebih kecil.