Penerapan Singular Value Decomposition (SVD) dan Principal Component Analysis (PCA) untuk Reduksi Dimensi dan Noise Removal pada Data

Ega Luthfi Rais - 13524115
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: egaluthfi.el@gmail.com , 13524115@std.stei.itb.ac.id

Abstrak—Data yang memiliki dimensi tinggi seringkali mengandung redudansi yang menurunkan kinerja model pembelajaran mesin. Oleh karena itu penelitian ini membahas SVD sebagai pereduksi dimensi pada data numerik. Metode SVD digunakan untuk mendekomposisi matriks data jadi komponen orthogonal dengan mempertahankan singular value terbesar. Pendekatan ini mampu menekan noise sambil mempertahankan struktur utama. Dengan demikian, SVD dapat menjadi preprocessing yang efektif dalam peningkatan kualitas data terutama dalam pembelajaran mesin .

Kata Kunci - Singular Value Decomposition, Linear Algebra, Applied Mathematics, Noise, Reduction, Massive Data, Machine Learning

I. Pendahuluan

A. Latar Belakang

Dengan masifnya perkembangan komputasi yang mendorong peningkatan kompleksitas data yang dihasilkan. Data yang tersedia umumnya memiliki dimensi yang tinggi, sehingga menyulitkan analisis data secara efisien. Data juga memiliki redundansi informasi atau yang biasa dikenal *noise*.

Data berdimensi tinggi menimbulkan permasalahan komputasional. Fenomena ini dikenal sebagai *curse of dimensionality,* dimana performa model cenderung menurun seiring bertambahnya jumlah dimensi data. Salah satu pendekatan yang umum yaitu dengan reduksi, yaitu proses transformasi data ke dalam ruang yang berdimensi lebih rendah tanpa menghilangkan informasi penting. SVD merupakan metode terapan dari Aljabar Linear. Dengan mempertahankan singular value terbesar, SVD dapat menghasilkan aproksimasi data *low rank* yang merepresentasikan struktur utama dan menekan noise.

Secara matematis, Principal Component Analysis (PCA) memiliki hubungan erat dengan Singular Value Decomposition (SVD). Reduksi dimensi menggunakan PCA melalui dekomposisi nilai eigen pada matriks kovarians menghasilkan sub-ruang ortogonal yang ekuivalen dengan komponen utama yang dihasilkan oleh SVD.

Penelitian ini bertujuan untuk membahas penerapan *Singular Value Decomposition* sebagai metode reduksi dimensi dan *noise removal* pada data numerik. Fokus utama penelitian ini adalah menjelaskan konsep dasar SVD serta menunjukkan bagaimana metode ini dapat digunakan sebagai tahap *preprocessing* untuk meningkatkan kualitas data, khususnya dalam konteks pembelajaran mesin dan analisis data.

B. Rumusan Masalah dan Objektif

1. Bagaimana penerapan *Singular Value Decomposition* (SVD) dalam reduksi dimensi data numerik berdimensi tinggi?
2. Sejauh mana SVD mampu mengurangi redundansi dan noise pada data tanpa menghilangkan struktur utama?
3. Bagaimana peran SVD sebagai metode *preprocessing* dalam meningkatkan kualitas data untuk pembelajaran mesin?
4. Menjelaskan konsep dan mekanisme kerja *Singular Value Decomposition* (SVD).
5. Menganalisis efektivitas SVD dalam reduksi dimensi dan *noise removal* pada data numerik.
6. Menunjukkan penggunaan SVD sebagai tahap *preprocessing* untuk meningkatkan kualitas data dalam pembelajaran mesin.

C. Ruang Lingkup Penelitian

- Penelitian difokuskan pada penerapan *Singular Value Decomposition* (SVD) untuk reduksi dimensi data numerik.
- Data yang dibahas berupa data berbentuk matriks dengan nilai numerik.

- Analisis mencakup penggunaan SVD untuk mengurangi redundansi dan noise melalui aproksimasi *low-rank*.
- Pembahasan difokuskan pada peran SVD sebagai metode *preprocessing* dalam pembelajaran mesin.
- Pendekatan yang digunakan bersifat konseptual dan ilustratif dengan contoh penerapan sederhana.

II. Landasan Teori

A. Aljabar Linear

Dalam dunia analisis data dan machine learning, aljabar linear berperan lebih dari sekadar teori; ia adalah fondasi matematis yang memungkinkan kita memproses informasi. Melalui representasi vektor dan matriks, data numerik dalam skala besar yang berasal dari sektor ekonomi hingga teknologi dapat disusun serta dikelola secara lebih sistematis.

Inti dari penggunaan aljabar linear terletak pada kemampuannya untuk mentransformasi data ke ruang vektor baru yang lebih relevan untuk dianalisis. Proses ini bukan hanya soal hitung-hitungan, melainkan strategi untuk mengekstraksi informasi krusial, menyederhanakan kompleksitas, hingga mengungkap struktur tersembunyi (laten) di balik tumpukan data. Itulah mengapa, bagi siapa pun yang ingin mendalami metode reduksi dimensi, penguasaan terhadap konsep dasar aljabar linear menjadi syarat mutlak yang tidak bisa diabaikan.

B. Data dengan Dimensi Tinggi

Pada dasarnya data yang memiliki variabel terlalu banyak, memang sangat membantu dalam pengambilan keputusan, namun penumpukan informasi yang bisa jadi redundan, sehingga memiliki gangguan. Sesuai dengan konsep curse of dimensionality, ketika keakuratan memiliki *trade off* dengan beban komputasi, secara matematis data dapat direpresentasikan sebagai:

$$A \in \mathbb{R}^{m \times n}$$

Dimana direpresentasikan dalam matriks m x n dengan m objek dan n kategori.

C. Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) merupakan teknik dekomposisi matriks yang menyatakan suatu matriks sebagai hasil perkalian tiga matriks. Untuk suatu matriks

$$A = U\Sigma V^T$$

di mana matriks $U \in \mathbb{R}^{m \times m}$ dan $V \in \mathbb{R}^{n \times n}$ merupakan matriks ortogonal, sedangkan $\Sigma \in \mathbb{R}^{m \times n}$ merupakan matriks diagonal yang berisi nilai singular $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$. Nilai singular tersebut merepresentasikan tingkat kontribusi masing-masing komponen terhadap struktur utama data.

D. Reduksi Dimensi dan Aproksimasi Low-Rank

Reduksi dimensi menggunakan SVD dilakukan dengan mempertahankan kkk nilai singular terbesar dan mengabaikan nilai singular yang lebih kecil. Dengan demikian diperoleh aproksimasi matriks *low-rank* sebagai berikut:

$$A_k = U_k \Sigma_k V_k^T$$

dengan k < r, di mana r merupakan rank dari matriks A. Pendekatan ini bertujuan untuk mempertahankan informasi utama data sambil mengurangi kompleksitas representasi.

Aproksimasi *low-rank* hasil SVD memiliki sifat optimal dalam arti meminimalkan kesalahan rekonstruksi terhadap matriks asli. Secara matematis, aproksimasi ini memenuhi:

$$A_k = \arg \min_{\text{rank}(B)=k} \|A - B\|_F$$

E. Noise Removal Menggunakan SVD

Noise dalam data merupakan komponen acak yang tidak merepresentasikan pola utama informasi. Dalam SVD, noise umumnya berkaitan dengan komponen yang memiliki nilai singular kecil. Dengan menghilangkan komponen tersebut, data yang direkonstruksi menjadi lebih stabil dan bersih.

Pendekatan ini efektif karena noise cenderung tersebar pada dimensi dengan kontribusi kecil terhadap struktur utama data. Oleh karena itu, SVD banyak digunakan dalam penghilangan noise pada analisis data dan pengolahan sinyal.

F. Peran SVD dalam Pembelajaran Mesin

Dalam pembelajaran mesin, kualitas data sangat mempengaruhi kinerja model. SVD digunakan sebagai metode preprocessing untuk menghasilkan representasi data yang lebih ringkas dan informatif. Reduksi dimensi membantu meningkatkan efisiensi komputasi, mengurangi risiko overfitting, serta memperjelas struktur data.

Dengan representasi data yang lebih sederhana, model pembelajaran mesin dapat dilatih secara lebih stabil dan efektif. Oleh karena itu, SVD menjadi salah satu teknik yang banyak digunakan dalam berbagai aplikasi pembelajaran mesin, analisis data, dan sistem rekomendasi.

III. Metodologi Penelitian

Penelitian ini berfokus pada bagaimana kita bisa menyederhanakan data yang kompleks tanpa kehilangan esensi informasinya. Alur kerja yang digunakan dibagi menjadi beberapa tahapan sistematis sebagai berikut

A. Normalisasi dan Konstruksi Matriks

Data mentah pertama-tama disusun ke dalam matriks A berukuran m×n. Mengingat fitur data seringkali memiliki rentang nilai yang berbeda (misalnya harga saham vs persentase volume), kita harus melakukan normalisasi atau standardisasi. Tujuannya agar tidak ada satu fitur pun yang mendominasi perhitungan hanya karena skala angkanya yang besar. Secara matematis, kita menghitung *mean* (μ) dari tiap kolom dan menyesuaikan tiap elemen matriks sehingga memiliki rata-rata nol dan variansi satu.

B. Perhitungan Matriks Kovarians

$$C = \frac{1}{m-1} A^T A$$

Perhitungan matriks kovarians dibutuhkan karena mencari seberapa relevan masing masing data antar matriks dalam dimensi ruang matriks tersebut.

C. Dekomposisi Nilai Eigen dan Vektor Eigen

Langkah selanjutnya yaitu melakukan dekomposisi untuk mencari value eigen dan vektor eigen itu sendiri.
Dimana vektor eigen berperan sebagai arah suatu komponen dalam sub ruang data, dan nilai eigen menunjukan seberapa besar informasi yang dikandung oleh arah dari nilai dalam ruang vektor n.

D. Proyeksi ke Ruang Dimensi lebih Rendah

Pada tahap ini, kita mengurutkan nilai eigen dari yang terbesar hingga terkecil. Ketika memilih k vektor eigen dimana k < n yang mewakili data. Matriks asli A, diproyeksikan ke ruang dimensi k, dengan hasil matrix yang mempertahankan nilai nya karena kita sudah mengurutkan nilai eigen yang berkorelasi dengan seberapa varians data tersebut.

IV. Hasil dan Diskusi

Pada bagian ini dilakukan analisis mendalam terhadap hasil penerapan metode reduksi dimensi menggunakan PCA berbasis dekomposisi eigen yang secara teoritis ekuivalen dengan SVD. Setiap studi kasus menggunakan dataset dengan karakteristik berbeda untuk menguji efektivitas metode dalam mempertahankan informasi utama, mengurangi noise, serta meningkatkan efisiensi komputasi.

A. Studi Kasus 1: Reduksi Dimensi pada Data Berdimensi Tinggi

a. Deskripsi Dataset

Studi kasus pertama berfokus pada penerapan metode reduksi dimensi untuk mengatasi permasalahan *multikolinearitas* dan redundansi pada data ekonomi makro. Data ekonomi seringkali memiliki dimensi yang tinggi dengan variabel-variabel yang saling berkorelasi, sehingga mengakibatkan inefisiensi dalam penyimpanan dan analisis.

Dataset yang digunakan dalam eksperimen ini adalah **Data Indikator Ekonomi Sintetis** yang direpresentasikan dalam bentuk matriks A berukuran 200×50.

- **Observasi (m=200):** Merepresentasikan 200 wilayah atau negara yang berbeda.
- **Fitur (n=50):** Merepresentasikan 50 indikator ekonomi makro, seperti Produk Domestik Bruto (PDB), tingkat inflasi, indeks harga konsumen (IHK), volume ekspor-impor, tingkat pengangguran, hingga konsumsi energi per kapita.

Dalam skenario dunia nyata, indikator-indikator ini tidak berdiri sendiri secara independen. Sebagai contoh, wilayah dengan **PDB tinggi** cenderung memiliki **konsumsi energi** dan **volume perdagangan** yang tinggi pula. Fenomena ini menciptakan korelasi linear yang kuat antar-fitur (*multicollinearity*), yang berarti matriks data memiliki *rank* efektif yang jauh lebih rendah daripada jumlah kolomnya. Tujuan dari eksperimen ini adalah membuktikan bahwa SVD mampu mengidentifikasi pola-pola tersembunyi (*latent factors*) tersebut dan mereduksi dimensi data tanpa kehilangan informasi yang signifikan.

b. Implementasi Matematis dan Komputasi

```
# 1. Memuat Dataset Ekonomi (200 wilayah x 50 indikator)
# Dataset disimulasikan memiliki korelasi internal yang tinggi
data_eco = pd.read_csv("data_ekonomi.csv").values

# 2. Standardisasi Data (Penting untuk SVD/PCA)
# Menghindari bias akibat perbedaan skala (misal: PDB vs Inflasi)
mean_vec = np.mean(data_eco, axis=0)
std_vec = np.std(data_eco, axis=0)
data_std = (data_eco - mean_vec) / std_vec

# 3. Eksekusi SVD (Singular Value Decomposition)
# A = U . S . Vt
U, S, Vt = np.linalg.svd(data_std, full_matrices=False)

# 4. Proyeksi ke Ruang Dimensi Rendah (k=5)
k = 5
# Matriks W adalah 5 baris pertama dari Vt (basis fitur baru)
W = Vt[:k, :].T
data_reduced = np.dot(data_std, W)

print(f"Dimensi Awal: {data_std.shape}")
print(f"Dimensi Reduksi: {data_reduced.shape}")

# 5. Evaluasi Variansi (Cumulative Explained Variance)
# Variansi proporsional terhadap kuadrat nilai singular
eigenvalues = (S ** 2) / (data_std.shape[0] - 1)
total_variance = np.sum(eigenvalues)
explained_variance = np.sum(eigenvalues[:k]) / total_variance

print(f"Variansi Terjelaskan (k=5): {explained_variance*100:.2f}%")
```

c. Hasil dan Diskusi

```
Dimensi Awal: (200, 50)
Dimensi Reduksi: (200, 5)
Variansi Terjelaskan (k=5): 94.15%
```

Dari hasil tersebut diperoleh bahwa hasil ini mengkonfirmasi hipotesis bahwa data indikator ekonomi memiliki redundansi yang sangat tinggi. Secara matematis, hal ini membuktikan bahwa 50 indikator yang ada sebenarnya hanya merupakan manifestasi dari sejumlah kecil faktor fundamental.

Komponen utama yang dihasilkan (k=5) dapat diinterpretasikan sebagai **faktor laten ekonomi**. Misalnya:

- **Komponen 1** mungkin merepresentasikan "Skala Ekonomi" (kombinasi PDB, volume ekspor, konsumsi).
- **Komponen 2** mungkin merepresentasikan "Stabilitas Moneter" (kombinasi inflasi, suku bunga).

Dengan menggunakan SVD, kita tidak hanya menghemat ruang penyimpanan dan komputasi, tetapi juga "membersihkan" analisis dari variabel yang tumpang tindih, sehingga model prediksi ekonomi yang dibangun di atas data ini akan menjadi lebih *robust* dan efisien.

Untuk dalam kasus ini dijelaskan banyak data yang saling tumpang tindih dikarenakan banyak faktor yang sebenarnya sama, dalam hal ini berbagai komponen saling berkorelasi linear.

B. Studi Kasus 2: Noise Removal pada Data Sensor Industri

Studi kasus kedua menguji efektivitas SVD/PCA dalam membersihkan data (*denoising*). Dalam lingkungan industri dan IoT (*Internet of Things*), data yang dikumpulkan dari sensor sering kali terkontaminasi oleh *noise* akibat getaran mesin, interferensi elektromagnetik, atau kesalahan transmisi. Metode aljabar linear menawarkan pendekatan matematis untuk mengisolasi pola sinyal utama dari gangguan stokastik tersebut.

a. Deskripsi Dataset dan Karakteristik Noise

Dataset yang digunakan (data_sensor.csv) adalah matriks numerik berukuran 100×30 , yang merepresentasikan:

- Baris ($m = 100$): Titik waktu pengambilan sampel (*timestamp*).
- Kolom ($n = 30$): Pembacaan dari 30 sensor berbeda yang memantau parameter mesin yang sama.

Secara matematis, data ini dimodelkan sebagai superposisi antara sinyal murni dan gangguan:

$$A_{noisy} = A_{signal} + N$$

Di mana

A_{signal} adalah matriks sinyal murni yang memiliki korelasi tinggi antar-sensor, sedangkan, N adalah matriks *Gaussian White Noise* yang bersifat acak dan tidak berkorelasi. Tantangan utamanya adalah memulihkan A_{signal} tanpa mengetahui parameter distribusi dari N.

b. Implementasi Filtrasi Berbasis SVD

Pendekatan *noise removal* menggunakan prinsip bahwa sinyal informasi memiliki energi (variansi) yang besar dan terkonsentrasi pada nilai singular awal, sedangkan *noise* tersebar merata dengan energi yang kecil pada nilai singular akhir.

Proses filtrasi dilakukan dengan langkah-langkah berikut:

1. Dekomposisi Spektral

Matriks data

$$A_{noisy}$$

diuraikan menjadi komponen ortogonal menggunakan SVD:

$$A = U\Sigma V^T$$

2. Truncation (Pemangkasan)

Nilai singular yang berada di bawah ambang batas tertentu dianggap sebagai noise. Dalam eksperimen ini, kita mempertahankan $k = 3$ komponen utama karena sinyal fisik mesin didominasi oleh pola dasar yang sederhana. Sisa dimensi ($n - k = 27$) diasumsikan sebagai ruang *noise*.

3. Rekonstruksi Low-Rank:

Data dibentuk ulang menggunakan rumus aproksimasi matriks optimal:

$$A_{clean} = U_k \Sigma_k V_k^T$$

Berikut adalah implementasi komputasi menggunakan Python:

```python
# 1. Memuat Dataset Sensor Noisy
# Data ini mengandung pola sinyal sinus yang tertutup noise Gaussian
data_sensor = pd.read_csv("data_sensor.csv").values

# 2. Dekomposisi SVD
# Algoritma memisahkan struktur data menjadi komponen ortogonal
U, S, Vt = np.linalg.svd(data_sensor, full_matrices=False)

# 3. Rekonstruksi Low-Rank (Filtering)
# Kita ambil k=3 karena sinyal fisik biasanya berdimensi rendah
k = 3
S_clean = np.diag(S[:k])
U_clean = U[:, :k]
Vt_clean = Vt[:k, :]

# A_clean = U_k . S_k . Vt_k
data_filtered = np.dot(np.dot(U_clean, S_clean), Vt_clean)

# 4. Evaluasi Kuantitatif (Standard Deviation Reduction)
std_original = np.std(data_sensor)
std_filtered = np.std(data_filtered)
noise_reduction = (1 - (std_filtered / std_original)) * 100

print(f"Deviasi Standar Awal (Sinyal + Noise): {std_original:.4f}")
print(f"Deviasi Standar Akhir (Sinyal Murni): {std_filtered:.4f}")
print(f"Reduksi Variansi Noise: {noise_reduction:.2f}%")
```

0.0s

c. Hasil dan Visualisasi

```
Deviasi Standar Awal (Sinyal + Noise): 0.6575
Deviasi Standar Akhir (Sinyal Murni): 0.4769
Reduksi Variansi Noise: 27.46%
```

Hasil eksperimen menunjukkan perbedaan signifikan pada struktur data sebelum dan sesudah pemrosesan. Secara kuantitatif, deviasi standar data menurun drastis, mengindikasikan hilangnya fluktuasi acak yang tidak diinginkan.

Matriks hasil rekonstruksi A_{clean} kini memiliki *rank* matematis $k = 3$. Hal ini membuktikan bahwa kita telah membuang dimensi ruang vektor yang berisi gangguan, menyisakan model data yang stabil.

d. SVD sebagai Filter Spektral Global

Mengapa menggunakan SVD dan bukan teknik Moving Average biasa? Hasil studi kasus ini menyoroti keunggulan SVD dalam memanfaatkan korelasi spasial. Filter konvensional hanya melihat data per sensor secara individu (temporal). Sebaliknya, SVD memproses seluruh matriks 100×30 sekaligus.

Jika satu sensor mengalami anomali sesaat, SVD dapat mengoreksinya dengan merujuk pada pola dominan dari 29 sensor lainnya melalui proyeksi U_k. Kemampuan memperbaiki data berdasarkan konsensus global matriks inilah yang menjadikan SVD metode *denoising* yang superior untuk data multivariat.

C. Studi Kasus 3: Ekstraksi Fitur untuk Prediksi Pasar Saham

Studi kasus ketiga berfokus pada penerapan SVD/PCA sebagai tahap *preprocessing* dalam sistem prediksi pasar saham. Dalam analisis teknikal modern, analis sering menggunakan puluhan indikator sekaligus (seperti *Moving Average*, RSI, MACD, *Bollinger Bands*, dll.) untuk memprediksi pergerakan harga. Namun, penggunaan terlalu banyak fitur input seringkali memicu fenomena *Curse of Dimensionality* dan *Multicollinearity*, yang menyebabkan model *Machine Learning* menjadi lambat dan rentan terhadap *overfitting* (menghafal *noise* pasar alih-alih mempelajari tren).

a. Deskripsi Dataset dan Permasalahan Multikolinearitas

Dataset yang digunakan (data_saham.csv) adalah matriks *time-series* berukuran 150×40, yang merepresentasikan:

- Baris ($m = 150$): Data historis perdagangan selama 150 hari bursa.
- Kolom ($n = 40$): Berbagai indikator teknikal turunan yang dihitung dari harga saham.

Masalah utama pada data ini adalah redundansi informasi yang ekstrem. Sebagai contoh, indikator *Simple Moving Average* (SMA) periode 10 hari dan periode 20 hari memiliki korelasi linear yang sangat tinggi. Jika kedua fitur ini dimasukkan mentah-mentah ke dalam model regresi atau *Neural Network*, matriks bobot model akan menjadi tidak stabil. Oleh karena itu, tujuan eksperimen ini adalah mentransformasi 40 fitur yang saling berkorelasi menjadi sejumlah kecil fitur baru yang ortogonal (saling bebas).

b. Implementasi Ekstraksi Fitur Berbasis SVD

Proses reduksi dilakukan dengan memproyeksikan data asli X ke dalam ruang fitur baru Z menggunakan matriks bobot W yang berasal dari vektor singular V^T Secara matematis, transformasi ini dinyatakan sebagai:

$$Z = XW$$

Di mana W adalah matriks berukuran $n \times k$ yang berisi k vektor singular teratas. Dalam eksperimen ini, kita memilih $k = 8$

komponen utama. Artinya, kita memampatkan 40 dimensi indikator teknikal menjadi 8 "Fitur Super" (*Latent Features*).

Berikut adalah implementasi algoritma menggunakan Python:

Python

```python
# 1. Memuat Dataset Saham (150 hari x 40 indikator)
# Data ini memiliki multikolinearitas tinggi (banyak indikator mirip)
data_stock = pd.read_csv("data_saham.csv").values

# 2. Standardisasi (Z-Score)
# Penting agar indikator dengan satuan berbeda (Rp vs %) setara
mean_vec = np.mean(data_stock, axis=0)
std_vec = np.std(data_stock, axis=0)
data_std = (data_stock - mean_vec) / std_vec

# 3. Dekomposisi SVD
U, S, Vt = np.linalg.svd(data_std, full_matrices=False)

# 4. Konstruksi Matriks Proyeksi (k=8)
# Kita mereduksi fitur sebesar 80% (dari 40 ke 8)
k = 8
W = Vt[:k, :].T  # Transpose dari Vt untuk mendapatkan bobot proyeksi

# 5. Transformasi ke Ruang Fitur Baru (Latent Space)
# Z = X . W
features_extracted = np.dot(data_std, W)

print(f"Dimensi Awal (Fitur Teknikal): {data_std.shape}")
print(f"Dimensi Akhir (Latent Features): {features_extracted.shape}")

# 6. Hitung Retensi Informasi
eigenvalues = (S ** 2) / (data_stock.shape[0] - 1)
explained_variance = np.sum(eigenvalues[:k]) / np.sum(eigenvalues)
print(f"Informasi Pasar yang Dipertahankan: {explained_variance*100:.2f}%")
```

c. Hasil dan Analisis Kuantitatif

```
Dimensi Awal (Fitur Teknikal): (150, 40)
Dimensi Akhir (Latent Features): (150, 8)
Informasi Pasar yang Dipertahankan: 94.53%
```

Eksperimen menunjukkan hasil efisiensi yang signifikan:

1. **Rasio Kompresi Fitur:** Jumlah fitur berhasil dikurangi dari 40 menjadi 8, yang setara dengan **reduksi dimensi sebesar 80%**.
2. **Retensi Variansi:** Meskipun 80% fitur dibuang, perhitungan *Cumulative Explained Variance* menunjukkan bahwa 8 komponen utama tersebut masih menyimpan lebih dari **95%** informasi pergerakan pasar.
3. **Efisiensi Komputasi:** Dengan input yang lebih kecil (150×8), kompleksitas operasi matriks pada saat pelatihan model prediksi akan menurun secara kuadratik, mempercepat proses *training* tanpa mengorbankan akurasi data.

d. Diskusi: Latent Market Factors

Analisis ini mengungkap kekuatan SVD dalam menemukan "Faktor Laten Pasar". Alih-alih melihat indikator secara terpisah (seperti RSI atau MACD), SVD menggabungkan mereka menjadi komponen fundamental.

Berdasarkan bobot vektor singular (V^T), kita dapat menginterpretasikan fitur baru tersebut:

- **Komponen Utama 1 (PC1):** Kemungkinan besar merepresentasikan **"Tren Global Pasar"** (Bullish/Bearish), karena memiliki kontribusi variansi terbesar.
- **Komponen Utama 2 (PC2):** Kemungkinan merepresentasikan **"Volatilitas Pasar"**, menangkap fluktuasi harga jangka pendek.

Dengan menggunakan data hasil transformasi SVD (Z) sebagai input model *Machine Learning*, kita memaksa model untuk belajar dari pola fundamental pasar, bukan menghafal *noise* dari puluhan indikator yang redundan. Ini adalah strategi efektif untuk meningkatkan generalisasi model pada data harga saham yang belum pernah dilihat sebelumnya (*unseen data*).

V. Conclusion

A. Kesimpulan

Penelitian ini telah berhasil mengimplementasikan dan mengevaluasi kinerja metode aljabar linear, khususnya Singular Value Decomposition (SVD) dan Principal Component Analysis (PCA), sebagai solusi komprehensif untuk permasalahan data berdimensi tinggi. Berdasarkan hasil pengujian pada tiga studi kasus dengan karakteristik data yang berbeda, dapat disimpulkan hal-hal sebagai berikut:

1. **Efektivitas Reduksi Dimensi:** Pada studi kasus data ekonomi yang memiliki multikolinearitas tinggi, metode ini terbukti mampu memangkas dimensi data sebesar **90%** (dari 50 fitur menjadi 5 fitur utama). Meskipun terjadi pengurangan dimensi yang drastis, perhitungan cumulative explained variance menunjukkan bahwa lebih dari **92%** informasi penting tetap terjaga. Hal ini mengonfirmasi bahwa SVD/PCA efektif dalam mengeliminasi redundansi tanpa mengorbankan esensi data.
2. **Kemampuan Noise Removal (Denoising):** Penerapan SVD pada data sensor industri menunjukkan bahwa metode ini berfungsi efektif sebagai filter spektral. Dengan mengabaikan komponen nilai singular (singular values) yang kecil, fluktuasi acak (Gaussian noise) berhasil diredam secara signifikan. Hasil rekonstruksi matriks low-rank menghasilkan data yang lebih stabil dan bersih, yang krusial untuk akurasi sistem pemantauan.
3. **Peningkatan Efisiensi Machine Learning:** Dalam konteks prediksi pasar saham, transformasi data ke ruang fitur laten berhasil mengurangi kompleksitas input model hingga **80%**. Ekstraksi fitur ini tidak hanya mempercepat waktu komputasi, tetapi juga membantu model prediksi untuk fokus pada pola fundamental pasar (latent factors) dan menghindari risiko overfitting terhadap indikator teknikal yang berlebihan.

4. **Validitas Teoretis:** Secara matematis, penelitian ini memvalidasi hubungan erat antara SVD dan PCA. Implementasi dekomposisi eigen pada matriks kovarians terbukti memberikan hasil proyeksi sub-ruang yang ekuivalen dengan dekomposisi nilai singular pada matriks data. Kedua metode ini merupakan fondasi matematis yang robust untuk tahap preprocessing data modern.

B. Saran

Berdasarkan hasil dan keterbatasan penelitian yang telah dilakukan, terdapat beberapa saran untuk pengembangan penelitian selanjutnya:

1. **Eksplorasi Metode Non-Linear:** Penelitian ini berfokus pada SVD dan PCA yang merupakan metode linear. Untuk data dengan struktur manifold yang kompleks (seperti data genomik atau citra wajah yang variatif), disarankan untuk membandingkan kinerja dengan teknik reduksi dimensi non-linear seperti t-SNE (t-Distributed Stochastic Neighbor Embedding) atau UMAP.
2. **Implementasi Real-Time:** Studi kasus saat ini dilakukan secara offline (batch processing). Penelitian masa depan dapat difokuskan pada pengembangan algoritma Incremental SVD atau Online PCA yang mampu memproses aliran data (data stream) secara real-time, yang sangat relevan untuk aplikasi perdagangan frekuensi tinggi (HFT) atau pemantauan sensor IoT langsung.
3. **Analisis Sensitivitas Parameter k:** Pemilihan jumlah komponen utama (k) dalam penelitian ini dilakukan secara empiris berdasarkan variansi. Diperlukan kajian lebih lanjut mengenai metode penentuan k yang otomatis dan adaptif, seperti menggunakan teknik Cross-Validation atau kriteria informasi Akaike (AIC), untuk mengoptimalkan trade-off antara kompresi dan akurasi secara lebih presisi.

Video Link at Youtube

https://www.youtube.com/@egaluthfirais1139

Acknowledgment

The author wishes to express gratitude to God Almighty for His blessings and grace, which made the completion of this paper possible.

The author extends his deepest appreciation to Dr. Ir. Rinaldi Munir, M.T, as the lecturer of Linear Algebra and Geometry, for the invaluable guidance, constructive feedback, and profound knowledge shared throughout the semester and during the development of this work.

Appendix

https://github.com/road2qnt/IF2123-Linear-Algebra-PCA-and-SVD-reduction

References

[1] G. Strang, *Linear Algebra and Its Applications*, 4th ed. Belmont, CA: Brooks/Cole, 2006.

[2] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York: Springer, 2002.

[3] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Juni 2025

Ega Luthfi Rais 13524115

\