Part 10 - Dimentionality Reduction dengan Principal Component Analysis

Data berdimensi tinggi sering kali memiliki struktur intrinsik berdimensi lebih rendah karena adanya korelasi dan redundansi antar dimensi. Reduksi dimensionalitas memanfaatkan struktur ini untuk merepresentasikan data secara lebih ringkas, idealnya tanpa kehilangan banyak informasi.

10.1 Pengaturan Masalah

Tujuan PCA adalah menemukan proyeksi $\tilde{x}_n$ dari titik data asli x_n yang semirip mungkin dengan aslinya, namun memiliki dimensionalitas intrinsik yang jauh lebih rendah. Kita mempertimbangkan sebuah set data $X = x_1, \ldots, x_N$, dengan $x_n \in \mathbb{R}^D$ dan rata-rata 0, yang memiliki matriks kovariansi data:

$$S = \frac{1}{N} \sum_{n=1}^{N} x_n x_n^\top$$

Kita mencari representasi terkompresi (kode) berdimensi rendah $z_n \in \mathbb{R}^M$ (dengan $M < D$) yang didefinisikan sebagai:

$$z_n = B^\top x_n$$

di mana $B := [b_1, \ldots, b_M] \in \mathbb{R}^{D \times M}$ adalah matriks proyeksi yang kolom-kolomnya b_i bersifat ortonormal. Vektor-vektor ini membentuk basis untuk subruang M-dimensi $U \subseteq \mathbb{R}^D$ tempat data yang diproyeksikan, $\tilde{x}_n$, berada. Dalam kerangka ini, $B^\top$ dapat dianggap sebagai *encoder* yang mengubah data x menjadi kode z, dan B sebagai *decoder* yang merekonstruksi data $\tilde{x} = Bz$ dari kode tersebut.

10.2 Perspektif Variansi Maksimum

Untuk komponen utama pertama, kita mencari sebuah vektor basis $b_1 \in \mathbb{R}^D$ yang memaksimalkan variansi dari data yang diproyeksikan ke arahnya. Variansi ini diberikan oleh:

$$V_1 = \frac{1}{N} \sum_{n=1}^{N} (b_1^\top x_n)^2 = b_1^\top \left(\frac{1}{N} \sum_{n=1}^{N} x_n x_n^\top \right) b_1 = b_1^\top S b_1$$

Untuk mencegah pembesaran magnitudo b_1 secara tak terbatas, kita menambahkan kendala bahwa b_1 harus merupakan vektor satuan, yaitu $|b_1|^2 = 1$. Masalah optimisasi terkendala ini:

$$\max_{b_1} b_1^\top S b_1 \quad \text{dengan kendala} \quad |b_1|^2 = 1$$

dapat diselesaikan menggunakan Pengali Lagrange, yang menghasilkan persamaan *eigenvalue*:

$$S b_1 = \lambda_1 b_1$$

Solusi yang memaksimalkan variansi adalah dengan memilih b_1 sebagai *eigenvector* dari matriks kovariansi data S yang berkorespondensi dengan *eigenvalue* terbesar, λ_1. Vektor ini disebut komponen utama pertama (*first principal component*). Secara umum, untuk menemukan subruang M-dimensi yang mempertahankan variansi paling besar, kita memilih M kolom dari matriks B sebagai M *eigenvector* dari S yang berasosiasi dengan M *eigenvalue* terbesar. Total variansi yang ditangkap oleh M komponen utama ini adalah jumlah dari *eigenvalue* terkait:

$$V_M = \sum_{m=1}^{M} \lambda_m$$

10.3 Perspektif Proyeksi

Pendekatan alternatif ini menurunkan PCA dengan cara meminimalkan galat rekonstruksi rata-rata antara data asli dan data yang diproyeksikan. Tujuannya adalah meminimalkan jarak Euklides kuadrat rata-rata:

$$J_M = \frac{1}{N} \sum_{n=1}^{N} |x_n - \tilde{x}_n|^2$$

Pertama, untuk basis ortonormal $b_1, \ldots, b_M$ yang diberikan, koordinat optimal z_n untuk proyeksi $\tilde{x}_n$ adalah hasil dari proyeksi ortogonal:

$$z_{in} = b_i^\top x_n$$

Dengan mensubstitusikan hasil ini, galat rekonstruksi dapat diturunkan menjadi:

$$J_M = \sum_{j=M+1}^{D} b_j^\top S b_j$$

Ini menunjukkan bahwa meminimalkan galat rekonstruksi setara dengan meminimalkan variansi data yang diproyeksikan ke komplemen ortogonal dari subruang utama. Untuk mencapai minimum, kita harus memilih basis $b_{M+1}, \ldots, b_D$ sebagai *eigenvector* dari S yang berasosiasi dengan *eigenvalue* terkecil. Akibatnya, basis untuk subruang utama $b_1, \ldots, b_M$ haruslah *eigenvector* yang berasosiasi dengan M *eigenvalue* terbesar, yang mengarah pada solusi yang sama dengan perspektif variansi maksimum. Galat rekonstruksi kuadrat rata-rata yang tersisa adalah jumlah dari *eigenvalue* yang tidak digunakan: $J_M = \sum_{j=M+1}^{D} \lambda_j$.

10.4 Komputasi Eigenvector dan Aproksimasi Pangkat Rendah

Pada sub-bab sebelumnya, kita telah memperoleh bahwa basis dari subruang utama (*principal subspace*) adalah *eigenvector* yang berasosiasi dengan *eigenvalue* terbesar dari matriks kovariansi data:

$$S = \frac{1}{N} \sum_{n=1}^{N} x_n x_n^\top = \frac{1}{N} X X^\top$$

di mana $X = [x_1, \ldots, x_N] \in \mathbb{R}^{D \times N}$ adalah matriks data. Untuk mendapatkan *eigenvalue* dan *eigenvector* yang bersesuaian dari S, kita dapat menggunakan dua pendekatan:

1. Melakukan dekomposisi *eigen* (lihat Sub-bab 4.2) dan menghitung *eigenvalue* serta *eigenvector* dari S secara langsung.
2. Menggunakan *Singular Value Decomposition* (SVD, lihat Sub-bab 4.5). Karena S simetris dan dapat difaktorkan menjadi $XX^\top$, *eigenvalue* dari S adalah kuadrat dari nilai singular (*singular values*) dari X.
   Lebih spesifik, SVD dari X diberikan oleh:

   $$X_{D \times N} = U_{D \times D} \Sigma_{D \times N} V^\top_{N \times N}$$

   di mana U dan V adalah matriks ortogonal dan Σ adalah matriks yang satu-satunya entri tak-nol adalah nilai singular $\sigma_{ii} \geq 0$. Dengan demikian, matriks kovariansi data menjadi:

   $$S = \frac{1}{N} X X^\top = \frac{1}{N} U \Sigma V^\top V \Sigma^\top U^\top = \frac{1}{N} U \Sigma \Sigma^\top U^\top$$

   Dari sini, dapat disimpulkan bahwa kolom-kolom dari U adalah *eigenvector* dari S.

Selanjutnya, *eigenvalue* λ_d dari S berhubungan dengan nilai singular σ_d dari X melalui:

$$\lambda_d = \frac{\sigma_d^2}{N}$$

Hubungan ini menyediakan koneksi antara sudut pandang variansi maksimum (Sub-bab 10.2) dan SVD.

10.4.1 PCA Menggunakan Aproksimasi Matriks Pangkat Rendah

PCA memilih kolom-kolom dari U sebagai *eigenvector* yang berasosiasi dengan M *eigenvalue* terbesar dari matriks kovariansi data S. Teorema Eckart-Young (Teorema 4.25) menawarkan cara langsung untuk mengestimasi representasi berdimensi rendah. Teorema ini menyatakan bahwa aproksimasi pangkat-M terbaik dari X:

$$\tilde{X}_M := \underset{\mathrm{rk}(A)\leq M}{\operatorname{argmin}} |X - A|_2 \in \mathbb{R}^{D\times N}$$

diperoleh dengan memotong SVD pada M nilai singular teratas. Dengan kata lain, kita memperoleh:

$$\tilde{X}_M = U_M \Sigma_M V_M^\top \in \mathbb{R}^{D\times N}$$

dengan $U_M = [u_1, \ldots, u_M]$, $V_M = [v_1, \ldots, v_M]$, dan Σ_M adalah matriks diagonal yang entri-entrinya adalah M nilai singular terbesar dari X.

10.4.2 Aspek Praktis

Secara teoretis, kita dapat menemukan *eigenvalue* sebagai akar dari polinomial karakteristik, namun untuk matriks yang lebih besar dari 4×4, hal ini tidak mungkin dilakukan karena Teorema Abel-Ruffini menyatakan bahwa tidak ada solusi aljabar untuk polinomial berderajat 5 atau lebih. Oleh karena itu, dalam praktiknya, kita menggunakan metode iteratif untuk menghitung *eigenvalue* atau nilai singular, yang sudah diimplementasikan dalam semua paket aljabar linear modern (misal `np.linalg.eigh` atau `np.linalg.svd`). Dalam banyak aplikasi seperti PCA, kita hanya memerlukan beberapa *eigenvector* teratas. Akan tidak efisien jika kita menghitung dekomposisi penuh lalu membuang sebagian besar hasilnya. Proses iteratif yang secara langsung mengoptimalkan *eigenvector* ini lebih efisien secara komputasi. Sebagai contoh ekstrem, jika hanya dibutuhkan *eigenvector* pertama, metode sederhana yang disebut *power iteration* sangat

efisien. Metode ini memilih sebuah vektor acak x_0 dan mengikuti iterasi:

$$x_{k+1} = \frac{Sx_k}{|Sx_k|}, \quad k = 0, 1, \dots$$

Barisan vektor ini akan konvergen ke *eigenvector* yang berasosiasi dengan *eigenvalue* terbesar dari S. Algoritma PageRank asli dari Google menggunakan algoritma semacam ini untuk memeringkat halaman web.

10.5 PCA dalam Dimensi Tinggi

Ketika dimensionalitas data D jauh lebih besar daripada jumlah titik data N (yaitu, $N \ll D$), menghitung dekomposisi *eigen* dari matriks kovariansi $S \in \mathbb{R}^{D \times D}$ menjadi sangat mahal secara komputasi. Untuk mengatasi ini, kita dapat memanfaatkan fakta bahwa *eigenvalue* tak-nol dari matriks $XX^\top$ sama dengan *eigenvalue* tak-nol dari $X^\top X$. Alih-alih menyelesaikan persamaan *eigenvalue* untuk matriks $S = \frac{1}{N} XX^\top$ yang berukuran $D \times D$, kita menyelesaikan masalah *eigenvalue* untuk matriks yang jauh lebih kecil, yaitu $\frac{1}{N} X^\top X \in \mathbb{R}^{N \times N}$. Jika c_m adalah *eigenvector* dari $\frac{1}{N} X^\top X$, maka *eigenvector* b_m dari S yang kita cari dapat dipulihkan melalui transformasi Xc_m (dan normalisasi). Ini secara dramatis mengurangi beban komputasi ketika $N \ll D$.

10.6 Langkah-Langkah Kunci dalam PCA

1. **Pengurangan Rata-rata**: Langkah pertama adalah memusatkan data dengan menghitung rata-rata μ dari set data, kemudian mengurangkannya dari setiap titik data. Proses ini memastikan bahwa set data yang dihasilkan memiliki rata-rata 0.
2. **Standardisasi**: Setelah data terpusat, bagi setiap titik data dengan deviasi standar σ_d pada setiap dimensi $d = 1, \dots, D$. Langkah ini bertujuan untuk membuat data menjadi bebas unit dan memiliki variansi 1 di sepanjang setiap sumbu.
3. **Dekomposisi Eigen dari Matriks Kovariansi**: Hitung matriks kovariansi data dari data yang telah distandardisasi. Selanjutnya, lakukan dekomposisi *eigen* pada matriks ini untuk menemukan *eigenvalue* dan *eigenvector* yang bersesuaian. Karena matriks kovariansi bersifat simetris, Teorema Spektral menjamin bahwa kita dapat menemukan basis ortonormal (*orthonormal basis*) dari *eigenvector*. *Eigenvector* dengan *eigenvalue* terbesar akan menjadi basis untuk subruang utama (*principal subspace*).
4. **Proyeksi**: Untuk memproyeksikan titik data baru $x^* \in \mathbb{R}^D$ ke subruang utama, pertama-tama titik data tersebut harus distandardisasi menggunakan rata-rata μ_d dan deviasi

standar σ_d dari data pelatihan:

$$x_*^{(d)} \leftarrow \frac{x_*^{(d)} - \mu_d}{\sigma_d}, \quad d = 1, \ldots, D$$

Proyeksi $\tilde{x}_*$ kemudian didapatkan melalui $\tilde{x}_* = BB^\top x_*$, dengan B adalah matriks yang kolom-kolomnya merupakan *yang berasosiasi dengan* terbesar. Koordinat dari proyeksi ini dalam basis subruang utama adalah $z_* = B^\top x_*$. Untuk mengembalikan hasil proyeksi ke ruang data asli (sebelum standardisasi), prosesnya harus dibalik:

$$\tilde{x}_*^{(d)} \leftarrow \tilde{x}_*^{(d)} \sigma_d + \mu_d, \quad d = 1, \ldots, D$$

10.7 Perspektif Variabel Laten

PCA dapat diformulasikan sebagai model probabilistik yang disebut *Probabilistic PCA* (PPCA). PPCA mengasumsikan sebuah proses generatif di mana data berdimensi tinggi $x \in \mathbb{R}^D$ dihasilkan dari variabel laten berdimensi rendah $z \in \mathbb{R}^M$. Model ini didefinisikan oleh:

1. Sebuah prior Gaussian standar pada variabel laten: $p(z) = \mathcal{N}(z|0, I)$.
2. Sebuah hubungan linear antara variabel laten dan data yang diamati, dengan derau (noise) Gaussian: $x = Bz + \mu + \epsilon$, di mana $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. Ini mendefinisikan distribusi kondisional $p(x|z) = \mathcal{N}(x|Bz + \mu, \sigma^2 I)$. Dengan mengintegrasikan (marginalisasi) variabel laten z, *likelihood* data diberikan oleh sebuah distribusi Gaussian:

   $$p(x) = \mathcal{N}(x|\mu, BB^\top + \sigma^2 I)$$

   PCA standar yang telah dibahas sebelumnya merupakan kasus khusus dari PPCA dalam batas tanpa derau, yaitu ketika $\sigma^2 \to 0$. Formulasi probabilistik ini menawarkan keuntungan seperti penanganan data yang hilang dan kemampuan untuk memperluas model, misalnya menjadi campuran model PCA (*mixture of PCA models*).