>> Principal Component Analysis (PCA)

PCA چیست؟ (?)

تکنیک آماری ← برای کاهش ابعاد + فشرده‌سازی داده + استخراج ویژگی‌ها
Dimensionality Reduction

کاربرد ها:
- Data Visualization
- Noise Reduction
- Feature Extraction
- Preprocessing
  - Data Compression
  - Improving Algorithm → الگوریتم‌های clustering، k-nearest neighbors و ... بهتر هستند.
  - Avoiding curse of Dimensionality ← یعنی داده‌ها در ابعاد فضای بالاتر پراکنده‌تر

$x = [x_1, x_2, \dots, x_d]$

$xW = Z$

$Z = [z_1, z_2, \dots, z_k]$

مراحل #

1. standardize d-dim dataset $\quad \frac{x - \mu}{\sigma} = X_{std}$

2. make covariance matrix $\quad cov(X_{std}) = \frac{1}{n-1} X_{std}^T X_{std}$

3. Calculate EigenValues & EigenVectors

$$cov(X_{std}) \cdot V = \lambda V$$

4. Sort eigenVals desc & related eigenVect

$$EigenVal\_EigenVect\_Pairs = \{(\lambda_1, V_1), (\lambda_2, V_2), \dots, (\lambda_m, V_m)\}$$

5. Form the projection matrix

$$W = [V_1, V_2, \dots, V_k]$$

6. Transform the data

$$X_{reduced} = X_{std} \cdot W$$

ویژگی مهم PCA ← Unsupervised

>> Dimensionality Reduction in Machine Learning

روش‌ها:
1. Principal Component Analysis (PCA)
2. Linear Discriminant Analysis (LDA)
3. t-Distributed Stochastic Neighbor Embedding (t-SNE)
4. Auto encoders
5. Factor Analysis
6. Kernel PCA

⊳ Linear Discriminant Analysis (LDA)

هدف: Project the data onto a lower-dimentional space with good class separability → to { -avoid overfitting, -reduce computational costs

نکته مهم: Supervised است

گام‌ها

0- استاندارد سازی داده‌ها

1- compute mean vectors (برای هر کلاس)

$$\mu_i = \frac{1}{N_i} \sum_{x \in X_i} x$$

2- compute within-class scatter matrix (S_w):

$$S_w = \sum_{i=1}^{K} \sum_{x \in X_i} (x - \mu_i)(x - \mu_i)^T$$

3- Between-class scatter matrix (S_b)

$$S_b = \sum_{i=1}^{K} N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

4- Compute eigenval & eigenVects $\Longrightarrow S_w^{-1} S_B v = \lambda v$

5- Select the top K eigenvectors $\rightarrow W = [v_1, v_2, \dots, v_K]$

6- Transform the Data $\longrightarrow Y = X W$

>> t- distributed stochastic neighbor embedding (t-SNE)

کاربردها: { dimensionality reduction
data Visualization

← از نوع غیرخطی کاهش ابعاد

چگونه کار می‌کند؟

1. Pairwise Similarities

- در فضای ابعاد بالا، شباهت های زوجی نقطه داده های ورودی را اندازه گیری می‌کند
- استفاده از توزیع گاوسی برای محاسبه شباهت ها (احتمالات شرطی) بین نقاط، جایی که نقاط شبیه دارای احتمالات بالا هستند

2. Low-Dimensional Mapping

- در فضای ابعاد کم، الگوریتم مذکور شباهت ها را مدل می‌کند (با استفاده از Student's t-distribution with one degree of freedom)
- باعث می‌شود که "Crowding Problem" رفع شود (برخورد نقاط در ابعاد پایین تر)

3. Cost Function

t-SNE سعی می‌کند تفاوت بین ۲ توزیع احتمال (بعد بالا و بعد کوچک) را با تابع هزینه ای بنام واگرایی Kullback-Leibler، کم کند.

4. Optimization

استفاده از گرادیان کاهشی برای پیدا کردن بهترین نمایش low-dim به طوری که ساختار high-dim تا جایی که امکان دارد حفظ شود

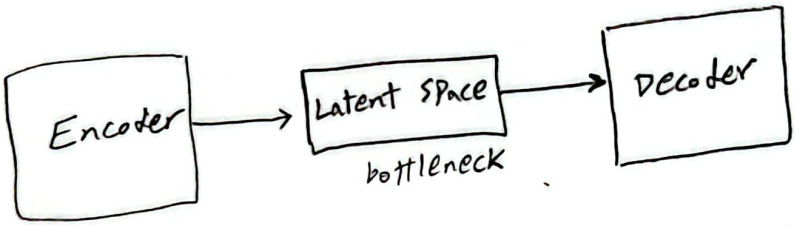
معایب:
- محاسبات بالا (هر pair باید محاسبه شود)
- حساسیت به hyperparameter ها
- توجه بیشتر به Local Structure به جای Global Structure

طبق تجربه ← perplexity خوب است که بین 5 تا 50 باشد

>> Auto Encoder

→ unsupervised learning

کاربردها:
- dimensionality reduction
- feature learning
- data compression



هدف: کمینه کردن اختلاف بین ورودی و خروجی (reconstruction error)

معمولا استفاده از MSE

انواع Autoencoder:
- Vanilla: یک encoder و یک decoder
- Denoising
- Sparse
- Variational → introduce probabilistic elements to learn the distribution of the input data
- Convolutional
- Undercomplete

کاربردها:
- کاهش ابعاد ← مانند PCA ولی دارای قابلیت گرفتن روابط غیرخطی
- حذف نویز
- تشخیص ناهنجاری
- ایجاد data sample های جدید

>> Factor Analysis

- روشی آماری برای یافتن روابط بین متغیرها

اهداف:
- شناسایی ساختار درونی داده های مشاهده شده
- کاهش ابعاد داده

انواع:
- Exploratory Factor Analysis (EFA) → مناسب برای زمانی که underlying structure را نمی شناسیم
- Confirmatory Factor Analysis (CFA) → زمانی که محقق فرضیاتی در مورد مسئله مورد پیش در نظر دارد

Latent variables → Factors

گام ها

1- آماده سازی داده ها

- استاندارد سازی داده ها

2- انتخاب تعداد فاکتورها

روش های کمک (استفاده از):
- معیار کایزر (eigVal > 1)
- scree plot
- آنالیز موازی

3- استخراج فاکتورها

- Principal Component Analysis (PCA)
- Common Factor Analysis
  - Principal Axis Factoring (PAF)
  - Maximum Likelihood

4- چرخش

اعمال برای آنکه output بیشتر تفسیر پذیر باشد ← با رسیدن به ساختار فاکتور ساده تر با حفظ تعداد فاکتورها
- orthogonal Rotation (ex: Varimax) → uncorrelated تصور می شود فاکتورها
- Oblique Rotation (ex: Promax) → Allows factor to be correlated

5 - تفسیر

شناسایی و نام گذاری فاکتورها براساس متغیرهای مربوط به آنها

6 - اعتبار سنجی

بررسی پایداری و اعتماد ساختار فاکتور $\xleftarrow[\text{از}]{\text{با استفاده}}$ split-sample validation / cross-validation

⇐ یکی از پیش فرض‌های مهم ← خطی بودن رابطه بین متغیرها و فاکتورها

>> kernel PCA

توسعه یافته PCA برای non-linear mapping of data

Project s high-dimensional data into lower-dimensional space

PCA شامل موارد زیر است:

1. Centering the data
2. computing the covariance matrix
3. Eigen decomposition (of cov matrix)
4. Projection

K-PCA ،، داریم:

⇒ map the data into a higher-dimensional feature space

1. choosing a kernel $(K(x,y))$ ← موردها:
   - Linear kernel: $K(x,y) = x^T y$
   - Polynomial kernel: $K(x,y) = (x^T y + c)^d$
   - Ganssian (RBF) kernel: $K(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$
     $= \exp(-\gamma\|x-y\|^2)$
2. Computing the kernel matrix (K)

$K_{ij} = K(x_i, y_i)$

3. Centering the Kernel Matrix

$$K_c = K - 1_n K - K 1_n + 1_n K 1_n$$

$1_n \rightarrow$ $n \times n$ matrix with all elements equal to $1/n$

4. Eigen decomposition (on K_c)

5. Projection (using eigenvect from step 4)