Lecture 5, Jan 26, 2024

Principal Component Analysis (PCA)

Definition

Dimensionality Reduction: Given a dataset $\mathcal{D} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ where $\boldsymbol{x}^{(i)} \in \mathbb{R}^D$, find a mapping $f : \mathbb{R}^D \mapsto \mathbb{R}^d$ where $d < D$ is a lower dimensional space.

- Dimensionality reduction is a type of unsupervised learning
  - PCA is a dimensionality reduction technique
  - Other techniques can include autoencoders, etc
- Dimensionality reduction can be used for a number of purposes:
  - Saving computational time/memory (helps with the curse of dimensionality)
  - Reduces overfitting
  - Visualize high-dimensional datasets
- We're essentially trying to create a summary of the data
- PCA is one of the only dimensionality reduction techniques with a closed-form solution
- PCA uses a linear model with the form $\boldsymbol{z} = \boldsymbol{U}^T(\boldsymbol{x} - \boldsymbol{b})$ where $\boldsymbol{U} \in \mathbb{R}^{D \times d}$ is an orthonormal matrix and $\boldsymbol{b} \in \mathbb{R}^D$
  - These orthonormal columns form a basis for a subspace $\mathcal{S}$
  - The projection of $\boldsymbol{x}$ onto $\mathcal{S}$ is the point $\tilde{\boldsymbol{x}} \in \mathcal{S}$ closes to $\boldsymbol{x}$ (this is known as the *reproduction* of $\boldsymbol{x}$
  - $\boldsymbol{z}$ is the *representation* or *code* of $\boldsymbol{x}$
- Choose $\boldsymbol{b} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \boldsymbol{x}^{(i)}$
- Finding a general matrix $\boldsymbol{U}$ is challenging, so we will start with a single column vector $\boldsymbol{u}$
  - We aim to minimize the reconstruction error: $\mathcal{L}(\boldsymbol{u}, \boldsymbol{b}) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \|\boldsymbol{x}^{(i)} - (\boldsymbol{u}\boldsymbol{u}^T(\boldsymbol{x}^{(i)} - \boldsymbol{b}) + \boldsymbol{b})\|_2^2$
    - $\hat{\boldsymbol{x}}^{(i)} = \boldsymbol{u}\boldsymbol{z} + \boldsymbol{b} = \boldsymbol{u}\boldsymbol{u}^T(\boldsymbol{x} - \boldsymbol{b}) + \boldsymbol{b}$
  - If the data is centered then $\boldsymbol{b} = \boldsymbol{0}$, so $\mathcal{L}(\boldsymbol{u}) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \|\boldsymbol{x}^{(i)} - \boldsymbol{u}\boldsymbol{u}^T\boldsymbol{x}^{(i)}\|_2^2$
- Expanding the reconstruction error:
  - $$\begin{aligned}\mathcal{L}(\boldsymbol{u}) &= \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}^{(i)} - \boldsymbol{u}\boldsymbol{u}^T\boldsymbol{x}^{(i)})^T(\boldsymbol{x}^{(i)} - \boldsymbol{u}\boldsymbol{u}^T\boldsymbol{x}^{(i)}) \\ &= \frac{1}{N}\sum_{i=1}^{n} -2\boldsymbol{x}^{(i)^T}\boldsymbol{u}\boldsymbol{u}^T\boldsymbol{x}^{(i)} + \boldsymbol{x}^{(i)^T}\boldsymbol{u}\boldsymbol{u}^T\boldsymbol{u}\boldsymbol{u}^T\boldsymbol{x}^{(i)} + \text{const} \\ &= \frac{1}{N}\sum_{i=1}^{N} -\boldsymbol{x}^{(i)^T}\boldsymbol{u}\boldsymbol{u}^T\boldsymbol{x}^{(i)} + \text{const}\end{aligned}$$
  - So we can formate the problem as minimizing $-\dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \boldsymbol{x}^{(i)^T}\boldsymbol{u}\boldsymbol{u}^T\boldsymbol{x}^{(i)}$ subject to $\boldsymbol{u}^T\boldsymbol{u} = 1$
  - Equivalently, maximize $\dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \boldsymbol{x}^{(i)^T}\boldsymbol{u}\boldsymbol{u}^T\boldsymbol{x}^{(i)} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \|\boldsymbol{z}^{(i)}\|_2^2$ subject to $\boldsymbol{u}^T\boldsymbol{u} = 1$
- Note the mean of $\boldsymbol{z}$ is zero since we centered $\boldsymbol{x}$ so the objective function is equivalent to $\dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \|\boldsymbol{z}^{(i)} - \bar{\boldsymbol{z}}\|_2^2$
  - Minimizing the reconstruction error is equivalent to maximizing the variance of the code vectors

- $\frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{z}^{(i)}\|_2^2 = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{u}^T(\boldsymbol{x}^{(i)}-\boldsymbol{\mu})(\boldsymbol{x}^{(i)}-\boldsymbol{\mu})^T\boldsymbol{u}$
$$= \boldsymbol{u}^T\left[\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}^{(i)}-\boldsymbol{\mu})(\boldsymbol{x}^{(i)}-\boldsymbol{\mu})^T\right]\boldsymbol{u}$$
$$= \boldsymbol{u}^T\boldsymbol{\Sigma}\boldsymbol{u}$$
$$= \boldsymbol{u}^T\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T\boldsymbol{u}$$
$$= \boldsymbol{a}^T\boldsymbol{\Lambda}\boldsymbol{a}$$
$$= \sum_{j=1}^{D}\lambda_j\boldsymbol{a}_j^2$$
  - We can decompose $\boldsymbol{\Sigma}$ since it is symmetric positive definite, as it is the empirical covariance matrix
  - $\boldsymbol{a} = \boldsymbol{Q}^T\boldsymbol{u}$ is a change of basis to the eigenbasis of $\boldsymbol{\Sigma}$
- Assuming all λ_i are sorted and distinct, we can choose $a_1 = \pm 1$ and $a_j = 0$ (since the first eigenvalue is the largest eigenvalue) in order to maximize the objective
  - Therefore $\boldsymbol{u} = \boldsymbol{Q}\boldsymbol{a} = \boldsymbol{q}_1$ which is just the top eigenvector
  - More generally, we can show that the kth principal component is given by the kth eigenvector of $\boldsymbol{\Sigma}$ (Courant-Fischer Theorem)
- Alternative derivation: we want to maximize a
  - The Lagrangian is $\mathcal{L}(\boldsymbol{u},\gamma) = \boldsymbol{u}^T\boldsymbol{\Sigma}\boldsymbol{u} + \gamma(1-\boldsymbol{u}^T\boldsymbol{u})$
  - $\vec{\nabla}_{\boldsymbol{u}}\mathcal{L} = (\boldsymbol{\Sigma}+\boldsymbol{\Sigma}^T)\boldsymbol{u} - 2\gamma\mathbf{1}\boldsymbol{u} = 0 \implies 2\boldsymbol{\Sigma}\boldsymbol{u} = 2\gamma\boldsymbol{u} \implies \boldsymbol{\Sigma}\boldsymbol{u} = \gamma\boldsymbol{u}$
- We can also perform PCA with SVD:
  - If $\boldsymbol{X}$ is a data matrix written in centered form, then the covariance matrix is $\boldsymbol{\Sigma} = \frac{1}{N}\boldsymbol{X}^T\boldsymbol{X}$
  - Using an SVD, we can write $\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{S}_1\boldsymbol{U}_1^T\boldsymbol{U}_1\boldsymbol{S}\boldsymbol{V}^T = \frac{1}{N}\boldsymbol{V}\boldsymbol{S}_1^2\boldsymbol{V}^T$
  - Since this is equal to $\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$ and spectral decompositions are unique, we must have that the columns of $\boldsymbol{V}$ are the principal components and $\frac{\boldsymbol{S}_1^2}{N} = \boldsymbol{\Lambda}$
  - So to construct the PCA we can just take the first d columns
  - Using SVD is faster and more stable
- Note the code vectors given by PCA are de-correlated (i.e. their covariance matrix is diagonal)