CPE 490 590: Machine Learning for Engineering Applications

10 Dimensionality Reduction

Rahul Bhadani

Electrical & Computer Engineering, The University of Alabama in Huntsville

Outline

Announcement

Homework 3

- Due: March 7, 2025
- Advanced Regression, Logistics Regression, and Neural Network

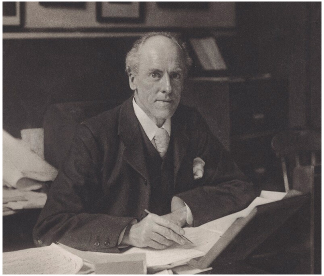
Motivation

Motivating Dimensionality Reduction

- There is a curse of dimensionality! The complexity of a model increases with the dimensionality. Sometimes exponentially!
- So, why should we perform dimensionality reduction?
  - reduces the time complexity: less computation
  - reduces the space complexity: fewer parameters
  - saves costs: some features/variables cost money
  - makes interpreting complex high-dimensional data
  - Can you visualize data with more than 3-dimensions?

Principal Component Analysis (PCA)

PCA: Proposed more than 100 years ago!



Karl Pearson (invented in 1901)

Harol Hotelling (invented in 1933)

Why?

- Helpful for reducing number of features
- For 2D/3D visualization
- Helpful in unsupervised algorithms
- Data storage cost can be lowered

Intuition Behind PCA

x_1: length of the car, x_2: width of the car

Class Work

Intuition Behind PCA

x_1: length of the car, x_3: diameter of the wheel

Class Work

Intuition Behind PCA

x_1: length of the car, x_4: height of the car

Class Work

Problem Setting

We are interested in finding projections $\tilde{\mathbf{x}}_n$ of data points $\mathbf{x}_n$ that are are similar to the original data points as possible, but which have significant lower intrinsic dimensionality.

Question

Given a dataset $X = \{\mathbf{x}_1, \cdots \mathbf{x}_n\}$, $\mathbf{x}_i \in \mathbb{R}^D$, is there a subspace $\mathbb{R}^M$, $M << D$ in which X approximately lies?

How does PCA work

Class Work

How does PCA work

Class Work

How does PCA work

Class Work

How does PCA work

Class Work

How does PCA work

Class Work

How does PCA work

Class Work

% Mathematics behind PCA

Feature Preprocessing

- Normalize the dataset so that $\mu = 0$, $\sigma = 1$.

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$$

$$\mathbf{x}_i = \mathbf{x}_i - \boldsymbol{\mu}$$

$$\sigma_j^2 = \frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \mu_j)^2$$

$$x_{ij} = \frac{x_{ij}}{\sigma_j}$$

Rescaling helps in finding duplicate features such as km/h, mph

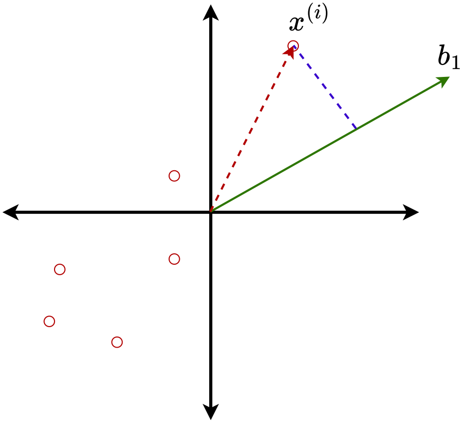
Feature Processing

More concretely, we transform dataset, so that we have iid dataset $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ with 0 mean and convariance matrix Σ:

$$\Sigma = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top$$

Here $\mathbf{x}_i$ is a D-dimensional vector.

Find the Direction of Maximum Variance



To project $\mathbf{x}_i$ onto a new axis $\mathbf{B}$:

$$\mathbf{z}_i = \mathrm{proj}_{\mathbf{B}}(\mathbf{x}_i) = \mathbf{B}^\top \mathbf{x}_i$$

We define the projection matrix as $\mathbf{B} := [\mathbf{b}_1, \cdots, \mathbf{b}_M] \in \mathbb{R}^{D \times M}$.
We assume that the columns of $\mathbf{B}$ are orthonormal such that

$$||b_i|| = 1$$

and $\mathbf{b}_i^\top \mathbf{b}_j = 0$.

Find the Direction of Maximal Variance

We maximize the variance of the low-dimensional coe using a sequential approach. Start with a single vector $_{\mathbf{1}} \in \mathbb{R}^{\mathbf{D}}$ that maximizes the variance of the projected data.

How do we choose unit vector $\mathbf{b}_1$ so that we maximize

$$V = \frac{1}{n}\sum(\mathbf{b}_1^\top \mathbf{x}_i)^2$$

Here, expressing with sum of squares accounts for negative projections in other quadrants.

The Direction with Maximal Variance

Expanding

$$V = \frac{1}{n}\sum(\mathbf{b}_1^\top \mathbf{x}_i)^2 = \frac{1}{n}\sum_{i=1}^{n} \mathbf{b}_1^\top \mathbf{x}_i \mathbf{x}_i \mathbf{b}_1$$

$$= \mathbf{b}_1^\top \left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top\right) \mathbf{b}_1 = \mathbf{b}_1^\top \Sigma \mathbf{b}_1$$

The dot product is symmetric, i.e. $\mathbf{b}_1^\top \mathbf{x}_i = \mathbf{x}_i^\top \mathbf{b}_1$.

Lagrange Multiplier to Solve the Optimization Problem

Our goal is to maximize $\mathbf{b}_1^\top \Sigma \mathbf{b}_1$ such that

$$\mathbf{b}_1^\top \mathbf{b}_1 = 1 \Rightarrow \mathbf{b}_1^\top \mathbf{b}_1 - 1 = 0$$

We use the method of Lagrange Multiplier

$$\mathcal{L}(\mathbf{b}_1, \lambda) = \mathbf{b}_1^\top \Sigma \mathbf{b}_1 - \lambda(\mathbf{b}_1^\top \mathbf{b}_1 - 1)$$

Lagrange Multiplier to Solve the Optimization Problem

$$\mathcal{L}(\mathbf{b}_1, \lambda) = \mathbf{b}_1^\top \Sigma \mathbf{b}_1 - \lambda(\mathbf{b}_1^\top \mathbf{b}_1 - 1)$$

Take the gradient with respect to $\mathbf{b}_1$

$$\nabla_{\mathbf{b}_1} \mathcal{L}(\mathbf{b}_1, \lambda) = \Sigma \mathbf{b}_1 - \lambda \mathbf{b}_1 = 0$$

$$\Rightarrow \Sigma \mathbf{b}_1 = \lambda \mathbf{b}_1$$

We see that $\mathbf{b}_1$ is an eignevector of the covariance matrix Σ and the lagrange multiplier λ is the corresponding eigenvalue.

Eigenvalue Problem

From the eigenvector property:

$$V = \mathbf{b}_1^\top \Sigma \mathbf{b}_1 = \lambda \mathbf{b}_1^\top \mathbf{b}_1 = \lambda$$

Hence, the variance of the data projected onto a 1-D subspace equal the eigenvalue associated the basis vector $\mathbf{b}_1$ that spans the subspace.

To maximimze the variance of the low-dimensional code, we choose the basis vector associated with the largest eigenvalue of the convariance matrix. This eigenvector is called the **first principal component**.

Prinicpal Components

If we consider all basis vectors, there will be D solution, i.e. D eigenvalues.
If you want to project into, say, $\mathbb{R}^2$ from $\mathbb{R}^6$, you select the first basis b_1 and b_2, sorted by the magnitude of respective eigenvalues.

Note

$b_1, b_2, \cdots$ are new basis vectors for the reduced data.

Reduced Dimensions

The reduced dimension, z_i can be written as

$$\mathbf{z}_i = \begin{bmatrix} \mathbf{b}_1^\top \mathbf{x}_i \\ \mathbf{b}_2^\top \mathbf{x}_i \\ \mathbf{b}_3^\top \mathbf{x}_i \\ \vdots \end{bmatrix} \quad \begin{matrix} \leftarrow \text{projection onto } \mathbf{b}_1 \\ \leftarrow \text{projection onto } \mathbf{b}_2 \\ \leftarrow \text{projection onto } \mathbf{b}_3 \\ \vdots \end{matrix}$$

Here, $\mathbf{z}_i \in \mathbb{R}^M$.

Reconstruction

We can reconstruct $\mathbf{x}_i$ as folllows, although with loss of information as :

$$\mathbf{x}_i^{\text{recon}} = \mathbf{B}\mathbf{z}_i = \sum_{j=1}^{M} z_{ij}\mathbf{b}_j$$

Note

The quality of the reconstruction depends on the number of principal components M used. If $M = D$, the reconstruction will be perfect, but if $M < D$, some information will be lost.

Reconstruction Error

$$\text{Error} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{x}_i^{\text{recon}}\|^2$$

This error measures how well the reduced-dimensional data approximates the original data. We should be looking at PCA minimizing this error while reducing the dimensionality of the data.

Note

The reconstruction error is directly related to the eigenvalues of the covariance matrix. The smaller the eigenvalues of the discarded components, the smaller the reconstruction error.

Choosing the Number of Principal Components

To choose the number of principal components M, we can use the following criteria:

- **Variance Explained:** Select enough components to explain a desired percentage of the total variance (e.g., 95%).
- **Scree Plot:** Plot the eigenvalues in descending order and look for an "elbow" point where the eigenvalues start to level off.
- **Cumulative Variance:** Calculate the cumulative sum of the eigenvalues and stop when the cumulative sum reaches a certain threshold.

> The number of principal components M is a trade-off between dimensionality reduction and the amount of information retained.

The End