Lecture Notes: Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)

QTM 347: Machine Learning I
Emory University

September 4, 2025

1 Motivation: Why Reduce Dimensions?

Modern datasets often measure many features at once. For example:

- X_1: population of a city,
- X_2: advertising spending in that city,
- X_3: average income,
- X_4: number of schools,
- ...

Challenges of high dimension:

1. **Visualization:** humans can only see in 2D or 3D.
2. **Computation:** algorithms become slow when p (the number of features) is large.
3. **Overfitting:** models may memorize noise rather than structure.
4. **Correlation:** many features contain overlapping information.

The Goal of PCA

PCA finds new features (called *principal components*) that capture the directions of maximum variation in the data. Often, only a few principal components are needed to explain most of the structure.

—

2 Linear Algebra Foundations

We build PCA using linear algebra. Let us recall key tools.

2.1 Scalars, Vectors, and Matrices

- A **scalar** is a single number, e.g. 5.
- A **vector** is a column of numbers, e.g.

$$\mathbf{x} = \begin{bmatrix} 2 \\ 7 \\ -1 \end{bmatrix}.$$

- A **matrix** is a rectangular array of numbers. Example:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}.$$

  Here A has 3 rows and 2 columns.

2.2 Matrix Multiplication and the Transpose

If A is $m \times n$ and B is $n \times p$, then AB is $m \times p$. Each entry:

$$(AB)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}.$$

Transpose: For any matrix A, the transpose A^T is defined by swapping rows and columns:

$$(A^T)_{ij} = A_{ji}.$$

Why is this important? Because when we compute variances, we need to measure "spread along directions." This requires multiplying a column vector by its transpose. For example:

$$\mathbf{x}^T \mathbf{x} = \sum_{i=1}^{n} x_i^2,$$

which is the squared length of $\mathbf{x}$. This is why transposes appear throughout PCA: they allow us to turn column vectors into row vectors so multiplication is defined.

—

2.3 Determinants and Eigenvalues

For a 2×2 matrix:

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc.$$

Eigenvalues/eigenvectors: For square M, λ and $\mathbf{v} \neq 0$ satisfy

$$M\mathbf{v} = \lambda \mathbf{v}.$$

To find λ, solve

$$\det(M - \lambda I) = 0.$$

—

3 Step 1: Centering the Data

Suppose $X \in \mathbb{R}^{n\times p}$ with n observations, p features.

We subtract the mean of each column:

$$X_c = X - \mathbf{1}\bar{X},$$

where $\mathbf{1}$ is an $n \times 1$ vector of ones and $\bar{X}$ is the row of column means.

This ensures that each feature has mean zero. This is crucial: variance calculations are simpler, and new components z_{i1} also have mean zero.

—

4 Step 2: Building a New Feature z_{i1}

PCA constructs *new features* as weighted combinations of the old ones.

For each observation i:

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \cdots + \phi_{p1}x_{ip}.$$

Here:

- x_{ij} = value of feature j for observation i.
- ϕ_{j1} = loading (weight) applied to feature j.
- $\boldsymbol{\phi}_1 = (\phi_{11}, \phi_{21}, \dots, \phi_{p1})^T$ = loading vector.

In matrix form:

$$z_{i1} = \mathbf{x}_{c,i}^T \boldsymbol{\phi}_1,$$

where $\mathbf{x}_{c,i}$ is the centered row vector for observation i.

Normalization Rule

We require

$$\|\boldsymbol{\phi}_1\|^2 = \sum_{j=1}^{p} \phi_{j1}^2 = 1.$$

This constraint fixes the "length" of the vector $\boldsymbol{\phi}_1$, preventing us from inflating variance by scaling weights arbitrarily.

—

5 Step 3: Variance of the New Feature

Variance of the set $\{z_{i1}\}_{i=1}^{n}$ is

$$\mathrm{Var}(z_1) = \frac{1}{n}\sum_{i=1}^{n}(z_{i1})^2.$$

Why square? Because variance must treat negative and positive deviations symmetrically. If we did not square, large positive and large negative values could cancel, incorrectly suggesting low variation. Squaring ensures all deviations contribute positively.

Why divide by n? Because variance is an average, not a total. We want a scale-free measure of spread.

—

6 Step 4: Optimization Problem

Our goal:

$$\max_{\boldsymbol{\phi}_1} \frac{1}{n}\sum_{i=1}^{n}(z_{i1})^2.$$

But recall

$$z_{i1} = \mathbf{x}_{c,i}^T \boldsymbol{\phi}_1.$$

So

$$(z_{i1})^2 = (\mathbf{x}_{c,i}^T \boldsymbol{\phi}_1)^2.$$

Using matrix rules, this becomes:

$$\frac{1}{n}\sum_{i=1}^{n}(z_{i1})^2 = \frac{1}{n}\boldsymbol{\phi}_1^T X_c^T X_c \boldsymbol{\phi}_1.$$

Define the covariance matrix:

$$S = \frac{1}{n}X_c^T X_c.$$

Then

$$\mathrm{Var}(z_1) = \boldsymbol{\phi}_1^T S \boldsymbol{\phi}_1.$$

Final Optimization Problem

$$\max_{\boldsymbol{\phi}_1} \ \boldsymbol{\phi}_1^T S \boldsymbol{\phi}_1 \quad \text{subject to } \|\boldsymbol{\phi}_1\|^2 = 1.$$

This is exactly the **Rayleigh quotient**, which is maximized when $\boldsymbol{\phi}_1$ is the eigenvector of S corresponding to the largest eigenvalue.

—

7 Step 5: Why the Transpose Appears

Note the crucial role of the transpose. Our data X_c is $n \times p$: n rows (observations), p columns (features).

To capture variance across features, we need a $p \times p$ matrix. If we computed $X_c X_c^T$, we'd get an $n \times n$ matrix, which measures similarity across *observations*.

Instead, PCA is about correlations among features. That's why we use

$$S = \tfrac{1}{n} X_c^T X_c,$$

which is $p \times p$. Here, the transpose X_c^T flips the shape so multiplication is valid and produces a square, symmetric matrix suitable for eigenvalue analysis.

—

8 Step 6: Singular Value Decomposition (SVD)

8.1 Definition

For any matrix $X_c \in \mathbb{R}^{n \times p}$,

$$X_c = U \Sigma V^T,$$

with:

- $U \in \mathbb{R}^{n \times n}$ orthogonal ($U^T U = I$),
- $V \in \mathbb{R}^{p \times p}$ orthogonal,
- $\Sigma \in \mathbb{R}^{n \times p}$ diagonal with nonnegative singular values $\sigma_1 \geq \sigma_2 \geq \cdots$.

8.2 Connection to PCA

$$X_c^T X_c = V \Sigma^2 V^T.$$

Thus:

- The eigenvectors of S are the columns of V.
- The eigenvalues of S are σ_i^2 / n.

So computing PCA is equivalent to computing the SVD of X_c.

—

9 Worked Example: Three Cities, Two Features

9.1 Dataset

$$X = \begin{bmatrix} 20 & 10 \\ 30 & 15 \\ 40 & 20 \end{bmatrix}.$$

9.2 Centering

Column means: $(30, 15)$. Subtract:

$$X_c = \begin{bmatrix} -10 & -5 \\ 0 & 0 \\ 10 & 5 \end{bmatrix}.$$

9.3 Covariance

$$X_c^T X_c = \begin{bmatrix} 200 & 100 \\ 100 & 50 \end{bmatrix}, \quad S = \tfrac{1}{3} \begin{bmatrix} 200 & 100 \\ 100 & 50 \end{bmatrix} = \begin{bmatrix} 66.67 & 33.33 \\ 33.33 & 16.67 \end{bmatrix}.$$

9.4 Eigenvalues

Solve

$$\det \begin{bmatrix} 66.67 - \lambda & 33.33 \\ 33.33 & 16.67 - \lambda \end{bmatrix} = 0.$$

Expand:

$$(66.67 - \lambda)(16.67 - \lambda) - 33.33^2 = 0.$$

Simplify:

$$\lambda^2 - 83.34\lambda + 0.22 = 0.$$

Roots:

$$\lambda_1 \approx 83.34, \quad \lambda_2 \approx 0.$$

9.5 Eigenvector for $\lambda = 83.34$

Solve

$$\begin{bmatrix} -16.67 & 33.33 \\ 33.33 & -66.67 \end{bmatrix} \mathbf{v} = 0.$$

Equation: $-16.67v_1 + 33.33v_2 = 0 \implies v_2 = 0.5v_1$.
Normalize:

$$\mathbf{w} = \frac{1}{\sqrt{1^2 + 0.5^2}} \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.894 \\ 0.447 \end{bmatrix}.$$

9.6 Projection

$$Z = X_c\mathbf{w} = \begin{bmatrix} -11.18 \\ 0 \\ 11.18 \end{bmatrix}.$$

Variance:

$$\frac{1}{3}\big((-11.18)^2 + 0^2 + (11.18)^2\big) = 83.3.$$

—

10 Step 7: Full SVD of the Example

We compute the SVD of

$$X_c = \begin{bmatrix} -10 & -5 \\ 0 & 0 \\ 10 & 5 \end{bmatrix}.$$

1. Compute $X_c^T X_c$ (done above):

$$\begin{bmatrix} 200 & 100 \\ 100 & 50 \end{bmatrix}.$$

2. Eigenvalues: $250, 0$. Singular values are $\sigma_1 = \sqrt{250} \approx 15.81$, $\sigma_2 = 0$.
3. Eigenvectors of $X_c^T X_c$ (right singular vectors V):

$$v_1 = \begin{bmatrix} 0.894 \\ 0.447 \end{bmatrix}, \quad v_2 = \begin{bmatrix} -0.447 \\ 0.894 \end{bmatrix}.$$

4. Left singular vectors: $u_i = \frac{1}{\sigma_i} X_c v_i$.
Compute u_1:

$$X_c v_1 = \begin{bmatrix} -10 & -5 \\ 0 & 0 \\ 10 & 5 \end{bmatrix} \begin{bmatrix} 0.894 \\ 0.447 \end{bmatrix} = \begin{bmatrix} -11.18 \\ 0 \\ 11.18 \end{bmatrix}.$$

Divide by $\sigma_1 = 15.81$:

$$u_1 = \begin{bmatrix} -0.707 \\ 0 \\ 0.707 \end{bmatrix}.$$

Thus:

$$U = \begin{bmatrix} -0.707 & * \\ 0 & * \\ 0.707 & * \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 15.81 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} 0.894 & -0.447 \\ 0.447 & 0.894 \end{bmatrix}.$$

So indeed

$$X_c = U \Sigma V^T.$$

—

11 Summary

Key Points

- PCA constructs new features z_{i1} as weighted sums of original features.
- The goal is to maximize variance of these new features.
- The problem reduces to the eigenvalue problem for the covariance matrix $S = \frac{1}{n}X_c^T X_c$.
- Transposes appear to ensure dimensions match and to form a symmetric $p \times p$ matrix.
- Eigenvectors of S give directions of maximum variance (principal components).
- SVD provides a computational method: $X_c = U\Sigma V^T$, with V giving the loadings.
- Worked example (3 cities) shows all steps: centering, covariance, eigenvalues, eigenvectors, projection, variance, full SVD.