

Pre-reading: Principal Component Analysis (PCA)

Carnegie Mellon University
School of Computer Science

Contents

1	PCA Math Background	1
1.1	PCA Motivation: Dimensionality Reduction	1
1.2	Projections	1
1.2.1	Scalar Projection	1
1.2.2	Vector Projection	2
1.3	Rotating Data	3
1.3.1	Rotation	3
1.3.2	Projection Matrix	4
1.4	Covariance Matrix	5
2	Summary	6

1 PCA Math Background

1.1 PCA Motivation: Dimensionality Reduction

Dimensionality reduction is a key technique in machine learning and data analysis. It aims to map the original dataset to a smaller dimension, where each sample has fewer features. Dimensionality reduction can be thought of as reducing the dimensions of the data matrix X from $N \times M$ to $N \times K$ where $K < M$, hence the name dimensionality reduction. By reducing the number of features, we can

- Reduce computational cost and therefore increase the training speed of our models because there are fewer features to train on
- Improve model performance by excluding irrelevant or noisy features
- Represent high-dimensional data in two or three dimensions for better visualization

Principal component analysis (PCA) is just one dimensionality reduction strategy, and it is the one we will cover in lecture. This pre-reading covers the linear algebra that PCA is built on: projecting a vector onto a direction, rewriting data in terms of a new set of axes, and summarizing the spread of a dataset with its covariance matrix.

Throughout, a dataset has N samples and M features, each sample is a column vector $\mathbf{x}^{(i)} \in \mathbb{R}^M$, and the data matrix $X \in \mathbb{R}^{N \times M}$ has one sample per row.

1.2 Projections

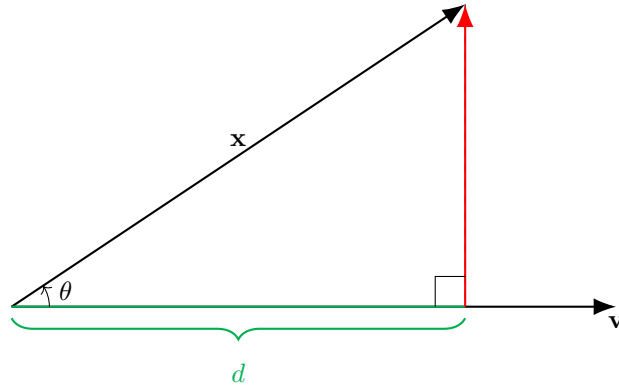
1.2.1 Scalar Projection

Given two vectors in $\mathbb{R}^M$ denoted $\mathbf{x}$ and $\mathbf{v}$, the **scalar projection** of $\mathbf{x}$ onto $\mathbf{v}$ is defined as:

$$d = \frac{\mathbf{v}^\top \mathbf{x}}{\|\mathbf{v}\|_2}$$

Note that if we assume that $\mathbf{v}$ is a unit vector, i.e., $\|\mathbf{v}\|_2 = 1$, the projection formula is *much* simpler:

$$d = \mathbf{v}^\top \mathbf{x}$$



The scalar projection, d , is the length of the vector $\mathbf{x}$ projected onto $\mathbf{v}$. We can prove this geometrically (recall that $\mathbf{v}^\top \mathbf{x} = \mathbf{v} \cdot \mathbf{x} = \|\mathbf{v}\|_2 \|\mathbf{x}\|_2 \cos(\theta)$):

$$\begin{aligned} \cos(\theta) &\triangleq \frac{d}{\|\mathbf{x}\|_2} \\ d &= \|\mathbf{x}\|_2 \cos(\theta) \\ d &= \frac{\mathbf{v}^\top \mathbf{x}}{\|\mathbf{v}\|_2} \end{aligned}$$

1.2.2 Vector Projection

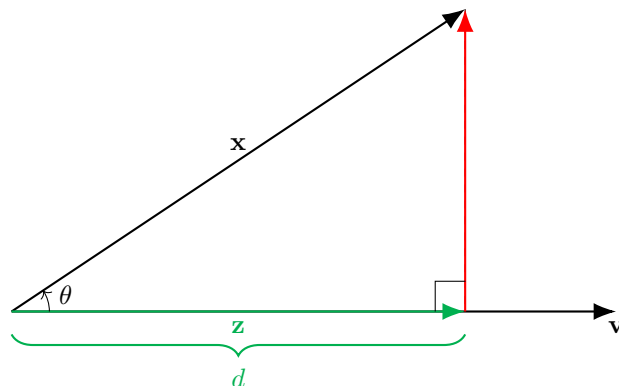
Recall from linear algebra the definition of vector projection. Given two vectors in $\mathbb{R}^M$ denoted $\mathbf{x}$ and $\mathbf{v}$, the **vector projection** of $\mathbf{x}$ onto $\mathbf{v}$, or $\text{proj}_{\mathbf{v}} \mathbf{x}$, is defined as:

$$\mathbf{z} = \frac{\mathbf{v}^\top \mathbf{x}}{\|\mathbf{v}\|_2} \frac{\mathbf{v}}{\|\mathbf{v}\|_2}$$

Note that if we assume that $\mathbf{v}$ is a unit vector, i.e., $\|\mathbf{v}\|_2 = 1$, the projection formula is *much* simpler:

$$\mathbf{z} = (\mathbf{v}^\top \mathbf{x}) \mathbf{v}$$

The projected vector $\mathbf{z}$ lies in the direction of $\mathbf{v}$ and represents the component of $\mathbf{x}$ in the direction of $\mathbf{v}$. We can think of the projection $\mathbf{z}$ as a restriction of $\mathbf{x}$ to the $\mathbf{v}$ -axis. The vector projection is visualized below:



Geometrically, the vector projection can be thought of as the vector $\mathbf{z}$ above, when both vectors are anchored at the origin. Its length is the scalar projection d , and its direction is the unit vector $\mathbf{v}$.

To develop a better understanding of how projections work and what they mean, feel free to use this [Desmos](#) link. Moving the points around in the Desmos link will change the projected vector.

1.3 Rotating Data

Now that we have established an intuition for projections, we can use this concept for rotating data. Recall that a basis for $\mathbb{R}^M$ is a set of vectors $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_M\}$ such that any vector $\mathbf{x} \in \mathbb{R}^M$ can be written as a linear combination of $\mathbf{v}_1, \dots, \mathbf{v}_M$. In these notes, our basis vectors will always be unit vectors that are perpendicular to each other (an **orthonormal basis**), so they can be thought of as the axes of our data. For example, the basis vectors $\mathbf{v}_1 = [1, 0]^\top$ and $\mathbf{v}_2 = [0, 1]^\top$ represent the x -axis and y -axis in the standard Cartesian coordinate system. You should observe that any vector $\mathbf{x} \in \mathbb{R}^2$ can be written as $z_1\mathbf{v}_1 + z_2\mathbf{v}_2$ where z_1 and z_2 are scalars.

1.3.1 Rotation

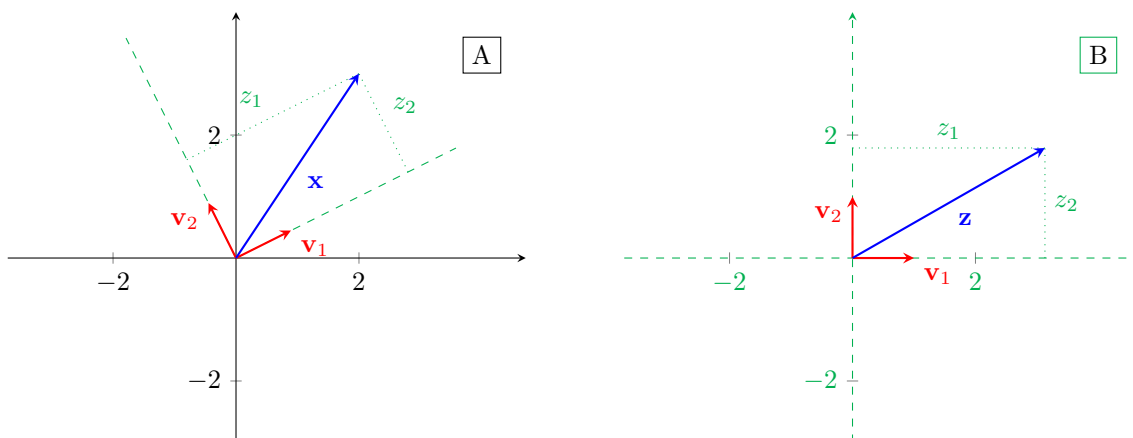
We can transform the data to be aligned to any set of axes by projecting onto the corresponding set of basis vectors. Consider the vector $\mathbf{x} = [2, 3]^\top$. Now we will project $\mathbf{x}$ onto the basis vectors

$$\mathbf{v}_1 = \left[\frac{2}{\sqrt{5}}, \frac{1}{\sqrt{5}} \right]^\top \text{ and } \mathbf{v}_2 = \left[\frac{-1}{\sqrt{5}}, \frac{2}{\sqrt{5}} \right]^\top$$

Note that $\mathbf{v}_1$ and $\mathbf{v}_2$ are once again unit vectors, and they are perpendicular to each other since $\mathbf{v}_1^\top \mathbf{v}_2 = 0$. Performing the projection calculation, we have that

$$z_1 = \mathbf{v}_1^\top \mathbf{x} = \frac{7}{\sqrt{5}} \approx 3.13$$

$$z_2 = \mathbf{v}_2^\top \mathbf{x} = \frac{4}{\sqrt{5}} \approx 1.79$$



Plot A shows $\mathbf{x}$ and the new axes $\mathbf{v}_1$ and $\mathbf{v}_2$ (dashed green) in the standard coordinate system. Plot B shows the same point written in terms of the new axes: $\mathbf{z} = [z_1, z_2]^\top$ is just $\mathbf{x}$ with its coordinates read off along $\mathbf{v}_1$ and $\mathbf{v}_2$ instead of along the standard axes.

1.3.2 Projection Matrix

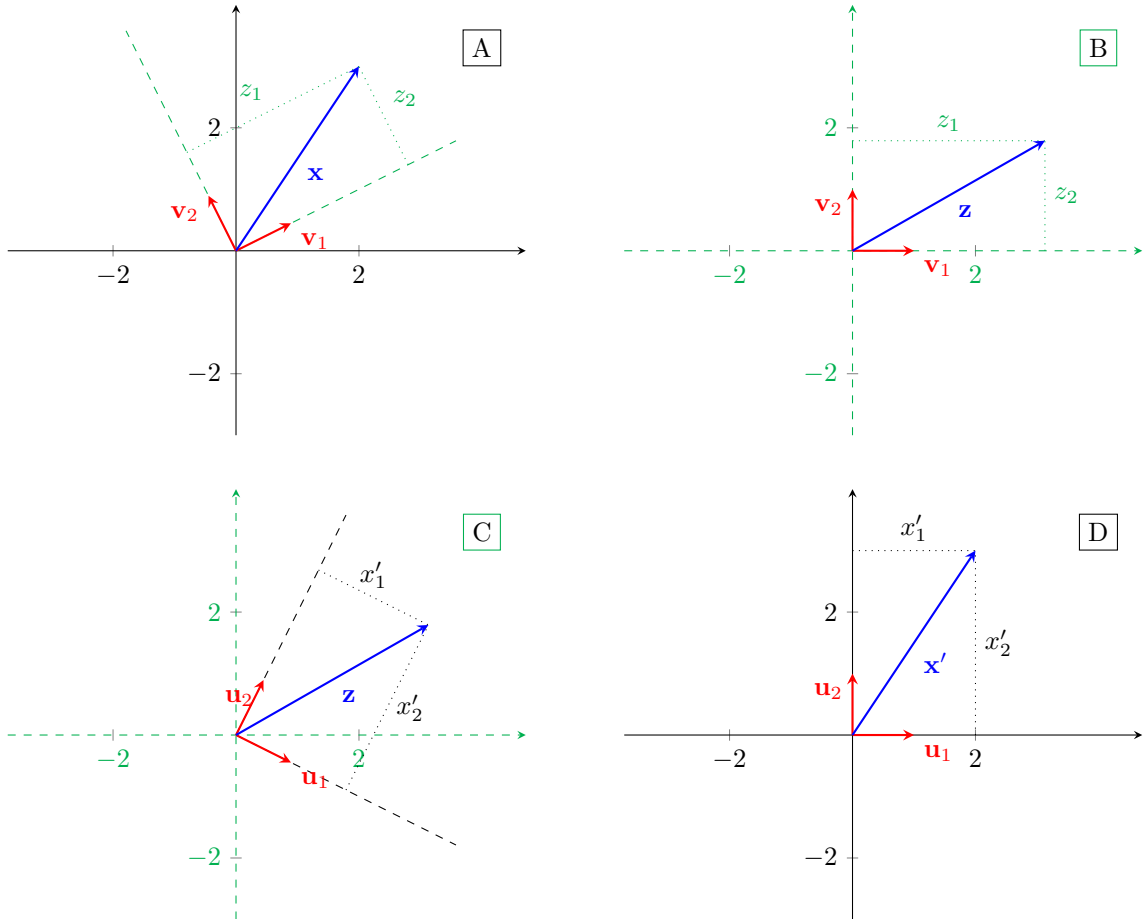
You may have noticed that when the $\mathbf{v}_i$ are unit vectors, z_i is just the dot product of $\mathbf{v}_i$ and $\mathbf{x}$. Let us now define V as the matrix whose rows are the basis vectors:

$$V = \begin{bmatrix} -\mathbf{v}_1^\top & - \\ \vdots & \\ -\mathbf{v}_M^\top & - \end{bmatrix}$$

Then it follows that $\mathbf{z} = V\mathbf{x}$, where the matrix V is called the **projection matrix**. Sometimes we want to reconstruct the original $\mathbf{x}$ from the projection $\mathbf{z}$. Let $\mathbf{x}' = V^\top \mathbf{z} = V^\top V \mathbf{x}$. When the projection matrix V is a square matrix, we are projecting $\mathbf{x}$ into a space with the same number of dimensions. Then $V^\top V = I \implies \mathbf{x}' = \mathbf{x}$, so we can reconstruct $\mathbf{x}$ perfectly. When we project into a lower-dimensional space, V is not a square matrix and thus, in general, we cannot reconstruct $\mathbf{x}$ perfectly.

Now, we can consider reversing that rotation through a different rotation matrix. Define $U = V^\top$. We can see that by applying the rotation U onto $\mathbf{z}$, we get $\mathbf{x}' = U\mathbf{z}$. Note that this is simply another rotation; however, because we have specifically chosen $U = V^{-1} = V^\top$, we get that $\mathbf{x}' = \mathbf{x}$.

Intuitively, if $\mathbf{x}$ represents the data in terms of the standard basis $[1, 0]^\top$ and $[0, 1]^\top$, then $\mathbf{z}$ represents the data in terms of $\mathbf{v}_1 = \begin{bmatrix} 2 \\ \sqrt{5} \end{bmatrix}$ and $\mathbf{v}_2 = \begin{bmatrix} -1 \\ \sqrt{5} \end{bmatrix}$. We can visualize what is happening below. Moving from plot A to plot B represents applying the rotation matrix V onto $\mathbf{x}$ to get $\mathbf{z}$. Then, by moving from plot C to plot D, we can see the reverse step by applying the rotation matrix U onto $\mathbf{z}$ to get $\mathbf{x}'$. In plot C, the rows $\mathbf{u}_1$ and $\mathbf{u}_2$ of U are drawn in the $\mathbf{z}$ coordinate system, and the components of $\mathbf{x}'$ are the projections of $\mathbf{z}$ onto them.



1.4 Covariance Matrix

PCA chooses its axes by looking at the variance of the dataset, so we need a way to measure that. In machine learning, when our dataset $\mathcal{D}$ has multiple features, each sample is represented as a column vector $\mathbf{x}^{(i)} \in \mathbb{R}^M$. Then we can represent the entire dataset as a matrix $X = [\mathbf{x}^{(1)} \dots \mathbf{x}^{(N)}]^\top \in \mathbb{R}^{N \times M}$, with one sample per row.

In this context, variance measures the variability of each feature in the dataset, as well as the relationships between different features. We will now introduce the covariance matrix.

The **covariance matrix** is an $M \times M$ square matrix that summarizes the covariance between each pair of features in the dataset. The size of the matrix is determined by the number of features, and the entry in row j and column k is the covariance between feature j and feature k .

- **Diagonal Elements:** The diagonal elements of the covariance matrix represent the variances of each feature. In other words, they represent how spread out the values are for a specific feature. Mathematically, the variance of feature m is $\frac{1}{N} \sum_{i=1}^N (x_m^{(i)} - \mu_m)^2$ where $x_m^{(i)}$ is the value of feature m for the i -th data point and $\mu_m = \frac{1}{N} \sum_{i=1}^N x_m^{(i)}$ is the mean of feature m .
- **Off-Diagonal Elements:** The off-diagonal elements of the covariance matrix represent the covariance between different features. Covariance is a measure of how two features vary together. If the covariance is positive, it means that when one feature increases, the other tends to increase as well. If it's negative, it means that when one feature increases, the other tends to decrease. Mathematically, the covariance between feature j and feature k is $\frac{1}{N} \sum_{i=1}^N (x_j^{(i)} - \mu_j)(x_k^{(i)} - \mu_k)$.

Generally, we assume that our data has been **centered**, i.e., the mean of each feature has been subtracted off, so that every feature has mean zero:

$$\mu_m = \frac{1}{N} \sum_{i=1}^N x_m^{(i)} = 0 \quad \text{for every feature } m.$$

Because each feature of our dataset has a mean of zero, the covariance between features j and k simplifies to $\frac{1}{N} \sum_{i=1}^N x_j^{(i)} x_k^{(i)}$, which is $\frac{1}{N}$ times the dot product of column j and column k of the data matrix X . Stacking all of these entries together, the whole covariance matrix is simply

$$\frac{1}{N} X^\top X$$

2 Summary

Throughout, $\mathbf{v}$ and the rows $\mathbf{v}_i$ of V are unit vectors, and the $\mathbf{v}_i$ are perpendicular to each other.

Expression	What it is
$d = \mathbf{v}^\top \mathbf{x}$	Scalar projection of $\mathbf{x}$ onto $\mathbf{v}$: the (signed) length of the shadow that $\mathbf{x}$ casts on the $\mathbf{v}$ -axis.
$\mathbf{z} = (\mathbf{v}^\top \mathbf{x}) \mathbf{v}$	Vector projection of $\mathbf{x}$ onto $\mathbf{v}$: the shadow itself, a vector of length $ d $ along $\mathbf{v}$.
$\mathbf{z} = V\mathbf{x}$	The coordinates of $\mathbf{x}$ in the new axes $\mathbf{v}_1, \dots, \mathbf{v}_M$; each $z_i = \mathbf{v}_i^\top \mathbf{x}$ is a scalar projection.
$\mathbf{x}' = V^\top \mathbf{z}$	Reconstruction of $\mathbf{x}$ from its new coordinates. Exact when V is square; only approximate when V keeps fewer than M axes.
$\frac{1}{N} X^\top X$	Covariance matrix of centered data. Diagonal entries are the variances of the features, off-diagonal entries are the covariances between pairs of features.

In lecture, we will put these pieces together: PCA picks the new axes $\mathbf{v}_1, \dots, \mathbf{v}_K$ so that the projected data $\mathbf{z} = V\mathbf{x}$ keeps as much of the variance of the dataset as possible, and we will see that those axes come from the covariance matrix $\frac{1}{N} X^\top X$.