Princeton University Computer Science
Fall 2025

COS 324: Introduction to Machine Learning
Principal Component Analysis (PCA)

Alex Tao, Edin Hasanovic, Victoria Dean, Ruth Fong

Last updated: Thursday 18th September, 2025, 12:31pm.

1. Linear Algebra Refresher

Below is a refresher of relevant linear algebra topics referenced in the PCA lecture. Students should be able to identify eigenvectors and eigenvalues from a set of options, but are not expected to directly compute them.

Definition 1.1 (Orthonormality). A set of k vectors $\{\vec{\mathbf{u}}_i \in \mathbb{R}^d\}_{i=1}^k$ is **orthonormal** if they are:

1. **Orthogonal** to one another (i.e. all pairs of inner products are 0) i.e. $\langle \vec{\mathbf{u}}_i, \vec{\mathbf{u}}_j \rangle = \vec{\mathbf{u}}_i \cdot \vec{\mathbf{u}}_j = 0$ for $i \neq j$, where $\vec{\mathbf{u}} \cdot \vec{\mathbf{v}} = \sum_{i=1}^d u_i \cdot v_i, \vec{\mathbf{u}}, \vec{\mathbf{v}} \in \mathbb{R}^d$ and

2. Each of unit length, i.e., they are **unit vectors**, which satisfy $\|\vec{\mathbf{u}}_i\|_2^2 = 1$ (Euclidean length $= 1$).

Such a set of vectors forms a **basis** if they are **linearly independent**.

Definition 1.2 (Linear Independence). A set of vectors is **linearly independent** if it cannot be written as a linear combination of one another. $\vec{\mathbf{u}}_1, ..., \vec{\mathbf{u}}_k \in \mathbb{R}^d$ are linearly independent if

$$\alpha_1 \vec{\mathbf{u}}_1 + \alpha_2 \vec{\mathbf{u}}_2 + \cdots + \alpha_k \vec{\mathbf{u}}_k = \vec{\mathbf{0}}.$$

Fact: If the columns of matrix $\mathbf{U} \in \mathbb{R}^{d \times k}$ are the vectors of the orthonormal basis $\{\vec{\mathbf{u}}_i \in \mathbb{R}^d\}_{i=1}^k$, then $\mathbf{U}$ is a rank-k matrix.

Definition 1.3 (Span). Suppose $\vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2, \ldots, \vec{\mathbf{u}}_k \in \mathbb{R}^d$. Then $\text{span}(\vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2, \ldots, \vec{\mathbf{u}}_k)$ is the set of all vectors $\vec{\mathbf{v}} \in \mathbb{R}^d$ that can be expressed as a linear combination of $\vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2, \ldots, \vec{\mathbf{u}}_k$, i.e., $\vec{\mathbf{v}}$ can be written as follows:

$$\vec{\mathbf{v}} = \alpha_1 \vec{\mathbf{u}}_1 + \alpha_2 \vec{\mathbf{u}}_2 + \ldots + \alpha_k \vec{\mathbf{u}}_k,$$

for $\alpha_1, \alpha_2, \ldots, \alpha_k \in \mathbb{R}$.

Below we outline some of the key linear algebra definitions and facts to understanding the result for the ***Best Orthonormal Basis*** claim.

Definition 1.4 (Eigenvector and Eigenvalue). Given a square $n \times n$ matrix $\mathbf{M}$, a vector $\vec{\mathbf{v}} \in \mathbb{R}^n - \{\vec{\mathbf{0}}\}$ is an **eigenvector** with an **eigenvalue** λ if $\mathbf{M}\vec{\mathbf{v}} = \lambda \vec{\mathbf{v}}$ (i.e. a scaled version of itself).

Note: For most vectors $\vec{\mathbf{x}} \in \mathbb{R}^n$, $\vec{\mathbf{z}} = \mathbf{M}\vec{\mathbf{x}}$ is a very different vector from $\vec{\mathbf{x}}$ (i.e. most vectors $\vec{\mathbf{x}} \in \mathbb{R}^n$ are not eigenvectors for a given matrix $\mathbf{M}$).

Example: Let $\mathbf{M} = \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$ and consider the vector $\vec{\mathbf{v}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. Then:

$$\mathbf{M}\vec{\mathbf{v}} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} = 2\vec{\mathbf{v}}$$

So, $\vec{\mathbf{v}}$ is an eigenvector of $\mathbf{M}$ with eigenvalue $\lambda = 2$.

Armed with these definitions, we are now ready to give proof for the best low-rank approximation that minimizes the reconstruction error.

2. Proof for best projection

Goal: Given a centered dataset $\{\vec{\mathbf{v}}_i \in \mathbb{R}^d\}_{i=1}^N$ (i.e. mean of dataset is $\vec{\mathbf{0}}$), an orthonormal basis of k vectors $\mathcal{U} = \{\vec{\mathbf{u}}_j \in \mathbb{R}^d\}_{j=1}^k$, and a new datapoint $\vec{\mathbf{v}} \in \mathbb{R}^d$, find its best low-rank approximation (i.e. best projection) $\widehat{\vec{\mathbf{v}}} \in \text{span}(\mathcal{U})$ that minimizes its reconstruction error:

$$\min \frac{1}{N} \|\vec{\mathbf{v}} - \widehat{\vec{\mathbf{v}}}\|_2^2 \tag{1}$$

where

$$\widehat{\vec{\mathbf{v}}} = \alpha_1 \vec{\mathbf{u}}_1 + \alpha_2 \vec{\mathbf{u}}_2 + \ldots + \alpha_k \vec{\mathbf{u}}_k = \sum j = 1^k \alpha_j \vec{\mathbf{u}}_j, \text{ using } \widehat{\vec{\mathbf{v}}} \in \text{span}(\mathcal{U}) \tag{2}$$

This amounts to finding the best solution for the α_i terms.

Claim: The best projection $\widehat{\vec{\mathbf{v}}}$ for $\vec{\mathbf{v}}$ that minimizes the above reconstruction error is defined as follows:

$$\widehat{\vec{v}} = \sum_{j=1}^k (\vec{\mathbf{v}} \cdot \vec{\mathbf{u}}_j) \vec{\mathbf{u}}_j, \text{ where } \alpha_j = \vec{\mathbf{v}} \cdot \vec{\mathbf{u}}_j \tag{3}$$

Approach: We'll prove this by computing the partial derivatives of the reconstruction loss with respect to (w.r.t.) α_i, setting it to 0, and solving for α_i (Problem 7.1.3 in course notes).

2.1. Re-write reconstruction error.

We'll first re-write the reconstruction error L for datapoint $\vec{\mathbf{v}}$:

$$\begin{aligned}
L &= \|\vec{\mathbf{v}} - \widehat{\vec{\mathbf{v}}}\|_2^2 && (4) \\
&= \|\vec{\mathbf{v}} - \sum_{j=1}^k \alpha_j \vec{\mathbf{u}}_j\|_2^2, \text{ using Equation (2)} && (5) \\
&= (\vec{\mathbf{v}} - \sum_{j=1}^k \alpha_j \vec{\mathbf{u}}_j) \cdot (\vec{\mathbf{v}} - \sum_{j=1}^k \alpha_j \vec{\mathbf{u}}_j), \text{ using } \|\vec{\mathbf{v}}\|_2^2 = \vec{\mathbf{v}} \cdot \vec{\mathbf{v}} && (6) \\
&= \vec{\mathbf{v}} \cdot \vec{\mathbf{v}} - 2\vec{\mathbf{v}} \cdot (\sum_{j=1}^k \alpha_j \vec{\mathbf{u}}_j) + \sum_{i=1}^k \sum_{j=1}^k \alpha_i \alpha_j (\vec{\mathbf{u}}_i \cdot \vec{\mathbf{u}}_j) && (7) \\
&= \|\vec{\mathbf{v}}\|_2^2 - 2\vec{\mathbf{v}} \cdot (\sum_{j=1}^k \alpha_j \vec{\mathbf{u}}_j) + \sum_{j=1}^k \alpha_j^2 (\vec{\mathbf{u}}_j \cdot \vec{\mathbf{u}}_j), \text{ using } \vec{\mathbf{u}}_i \cdot \vec{\mathbf{u}}_j = 0 \text{ if } i \neq j \text{ (orthonormal prop.)} && (8)
\end{aligned}$$

$$= \|\vec{\mathbf{v}}\|_2^2 - 2\vec{\mathbf{v}} \cdot (\sum_{j=1}^{k} \alpha_j \vec{\mathbf{u}}_j) + \sum_{j=1}^{k} \alpha_j^2, \text{ using } \vec{\mathbf{u}}_j \cdot \vec{\mathbf{u}}_j = 1 \text{ (orthonormal property)} \tag{9}$$

2.2. **Set partial derivative to 0 and solve.** Now, let's compute the partial derivative $\dfrac{\partial L}{\partial \alpha_i}$, set it to 0, and solve for α_i:

$$\frac{\partial L}{\partial \alpha_i} = \frac{\partial}{\partial \alpha_i} \left(\|\vec{\mathbf{v}}\|_2^2 - 2\vec{\mathbf{v}} \cdot (\sum_{j=1}^{k} \alpha_j \vec{\mathbf{u}}_j) + \sum_{j=1}^{k} \alpha_j^2 \right) \tag{10}$$

$$= -2\vec{\mathbf{v}} \cdot \vec{\mathbf{u}}_i + 2\alpha_i = 0 \tag{11}$$

$$\rightarrow \alpha_i = \vec{\mathbf{v}} \cdot \vec{\mathbf{u}}_i \tag{12}$$

Finally, plugging $\alpha_i = \vec{\mathbf{v}} \cdot \vec{\mathbf{u}}_i$ (Equation (12)) back into $\widehat{\vec{\mathbf{v}}}$:

$$\widehat{\vec{\mathbf{v}}} = \sum_{j=1}^{k} \alpha_j \vec{\mathbf{u}}_j = \sum_{j=1}^{k} (\vec{\mathbf{v}} \cdot \vec{\mathbf{u}}_j) \vec{\mathbf{u}}_j \tag{13}$$

This is what we wanted to prove.

3. Best Orthonormal Basis

Goal: Find the best orthonormal basis vectors $\vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2, \ldots, \vec{\mathbf{u}}_k \in \mathbb{R}^d$ such that the following reconstruction error is minimized:

$$\frac{1}{N}\sum_{i=1}^{N} \|\vec{\mathbf{v}}_i - \widehat{\vec{\mathbf{v}}}_i\|_2^2, \text{ where } \widehat{\vec{\mathbf{v}}} = \Sigma_{j=1}^{k}(\vec{\mathbf{v}} \cdot \vec{\mathbf{u}}_j)\vec{\mathbf{u}}_j \text{ from the } \textbf{\textit{Best Projection}} \text{ claim}$$

Claim: The best orthonormal basis vectors are the k **eigenvectors** corresponding to the k largest **eigenvalues** of the matrix $\mathbf{M} = \frac{1}{N}\mathbf{A}\mathbf{A}^\top$, where $\mathbf{A}$ is the matrix whose columns are $\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2, \ldots \vec{\mathbf{v}}_N$:

$$\mathbf{A} = \begin{bmatrix} | & | & \cdots & | \\ \vec{\mathbf{v}}_1 & \vec{\mathbf{v}}_2 & \cdots & \vec{\mathbf{v}}_N \\ | & | & \cdots & | \end{bmatrix} \text{ and } \mathbf{A}^\top = \begin{bmatrix} - & \vec{\mathbf{v}}_1^\top & - \\ - & \vec{\mathbf{v}}_2^\top & - \\ - & \cdots & - \\ - & \vec{\mathbf{v}}_N^\top & - \end{bmatrix}$$

3.1. 2D Example ($k = 1$ reconstruction).

Say we have 3 data points in $\mathbb{R}^2$ centered around the origin:[1]

$$\vec{\mathbf{v}}_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad \vec{\mathbf{v}}_2 = \begin{bmatrix} -1 \\ -2 \end{bmatrix}, \quad \text{and} \quad \vec{\mathbf{v}}_3 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

These vectors form the matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & -1 \\ 1 & -2 & 1 \end{bmatrix}$$

Our matrix $\mathbf{M}$ from the claim is constructed as:

$$\mathbf{M} = \frac{1}{3}\mathbf{A}\,\mathbf{A}^\top = \frac{1}{3}\begin{bmatrix} 2 & -1 & -1 \\ 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} 2 & 1 \\ -1 & -2 \\ -1 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 6 & 3 \\ 3 & 6 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Computing the eigenvalues of this matrix, we get:[2]

$$\lambda_1 = 3, \quad \lambda_2 = 1$$

Finding the corresponding eigenvectors (and normalizing), we obtain:

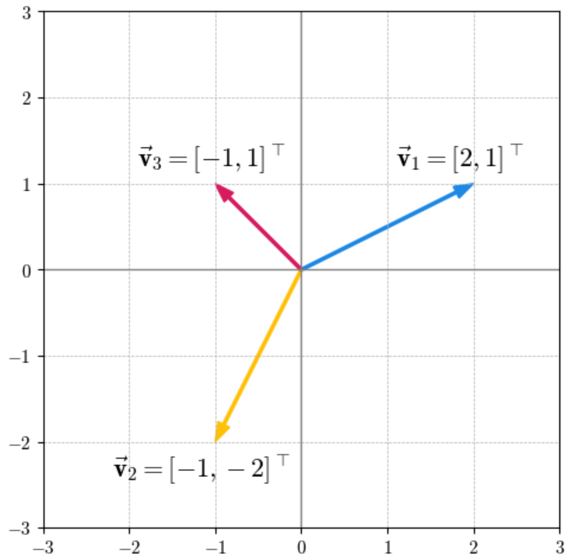
$$\vec{\mathbf{u}}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \vec{\mathbf{u}}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

where the first eigenvector $\vec{\mathbf{u}}_1$ is the best **1D projection direction**. If you project all the vectors onto $\vec{\mathbf{u}}_1$, you will minimize the average reconstruction error. If you use both eigenvectors, you span the full space and incur zero reconstruction error.[3] For students interested in eigenvector/eigenvalue computations, please see Example 4.5 on page 107 of the MML textbook. We visually show this below in Figure 1:

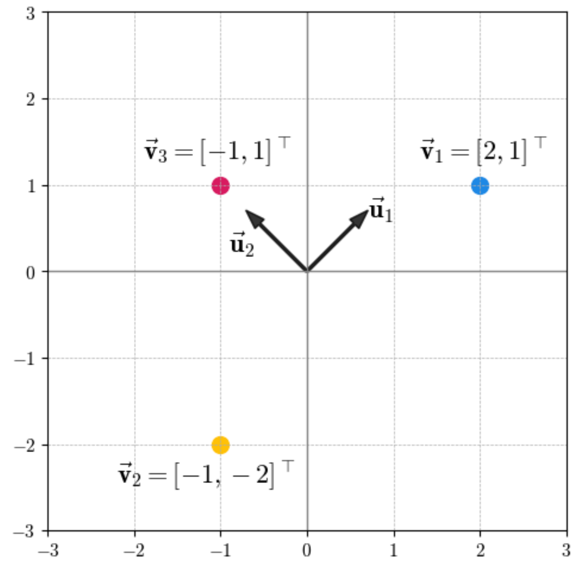
[1] The dataset needs to be centered so that when we later look for important directions in the data (eigenvectors), those directions reflect the way the points are spread out *relative to each other*, not where they happen to sit in space.

[2] Computations for the eigenvalues given here.

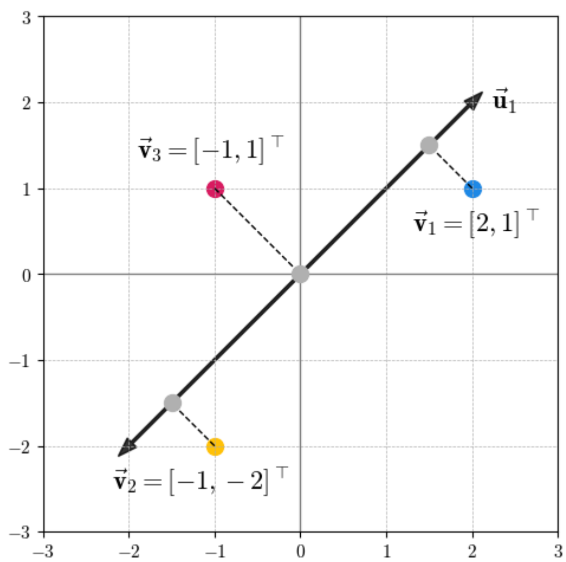
[3] Spanning the full space would mean that we can exactly construct each of the data points using linear combinations of the eigenvectors.



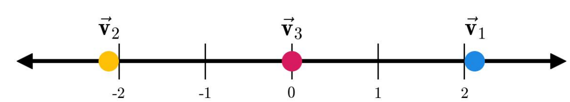
(a) Original data vectors $\vec{\mathbf{v}}_1$, $\vec{\mathbf{v}}_2$, and $\vec{\mathbf{v}}_3$ in $\mathbb{R}^2$.



(b) Top two eigenvectors $\vec{\mathbf{u}}_1$ and $\vec{\mathbf{u}}_2$ of $\mathbf{M} = \frac{1}{N}\mathbf{A}\mathbf{A}^\top$.



(c) Projection of original data onto $\vec{\mathbf{u}}_1$ with dashed reconstruction lines.



(d) 1D space (span of $\vec{\mathbf{u}}_1$) showing projected points.

FIGURE 1. Principal Component Analysis (PCA): from 2D input vectors to projection onto the top principal component.

Note: The top eigenvector $\vec{\mathbf{u}}_1$ is scaled in the illustration (1c) for visual clarity. Since projection only depends on direction, not magnitude, scaling the eigenvector does not affect the projection results or the interpretation.

Let's compute the average reconstruction error using the formula given in Equation (1). We project each vector $\vec{\mathbf{v}}_i$ onto the top eigenvector $\vec{\mathbf{u}}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ using: $\widehat{\vec{\mathbf{v}}}_i = (\vec{\mathbf{u}}_1^\top \vec{\mathbf{v}}_i) \cdot \vec{\mathbf{u}}_1$. We see that $\vec{\mathbf{v}}_1$ is projected onto $\begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}$, giving us a reconstruction error of:

$$\|\vec{\mathbf{v}}_1 - \widehat{\vec{\mathbf{v}}}_1\|_2^2 = \left\| \begin{bmatrix} 2 \\ 1 \end{bmatrix} - \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix} \right\|_2^2 = (0.5)^2 + (-0.5)^2 = 0.5$$

Computing this for the other vectors, we see $\vec{\mathbf{v}}_2$ has a reconstruction error of 0.5 and $\vec{\mathbf{v}}_3$, which projects to $\vec{\mathbf{0}}$, has reconstruction error 2. Thus, the average reconstruction error is: $\frac{1}{3}(0.5 + 0.5 + 2) = \frac{3}{3} = 1$

This section illustrates the foundational idea behind dimensionality reduction using PCA: the top k (in the above case above $k = 1$) eigenvectors of the matrix $\mathbf{M} = \frac{1}{N}\mathbf{A}\mathbf{A}^\top$ define the optimal k-dimensional space that minimizes the average reconstruction error when the data is projected onto it.

By projecting the original data onto a lower-dimensional space, we preserve the most important **spatial relationships** among the points while minimizing reconstruction error. This means that key geometric features, such as the relative ordering and relative distances between points, are maintained as faithfully as possible in the lower-dimensional space, enabling effective data compression and retaining meaningful structure.

3.2. **3D Example ($k = 2$ reconstruction).** Given 4 centered data points in $\mathbb{R}^3$:

$$\vec{\mathbf{v}}_1 = \begin{bmatrix} 4 \\ 0 \\ 1 \end{bmatrix}, \quad \vec{\mathbf{v}}_2 = \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix}, \quad \vec{\mathbf{v}}_3 = \begin{bmatrix} -4 \\ 0 \\ 1 \end{bmatrix}, \quad \vec{\mathbf{v}}_4 = \begin{bmatrix} 0 \\ -2 \\ -1 \end{bmatrix}$$

These data vectors form the matrix:

$$\mathbf{A} = \begin{bmatrix} 4 & 0 & -4 & 0 \\ 0 & 2 & 0 & -2 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

From this, we compute:

$$\mathbf{M} = \frac{1}{4}\mathbf{A}\mathbf{A}^\top = \frac{1}{4} \begin{bmatrix} 32 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 4 \end{bmatrix} = \begin{bmatrix} 8 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

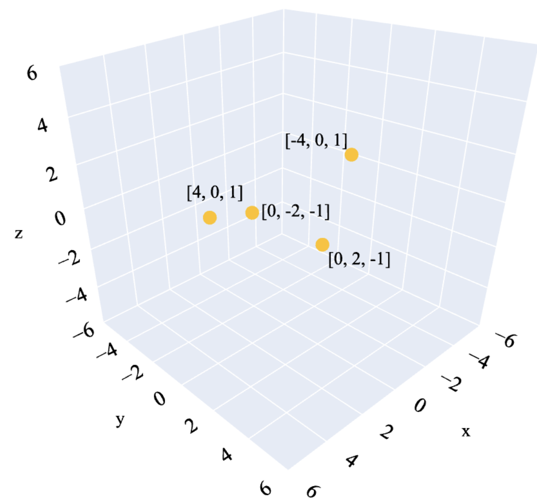
The eigenvalues of $\mathbf{M}$ are: $\lambda_1 = 8, \lambda_2 = 2, \lambda_3 = 1$, with corresponding eigenvectors:[4]

$$\vec{\mathbf{u}}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \vec{\mathbf{u}}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{\mathbf{u}}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

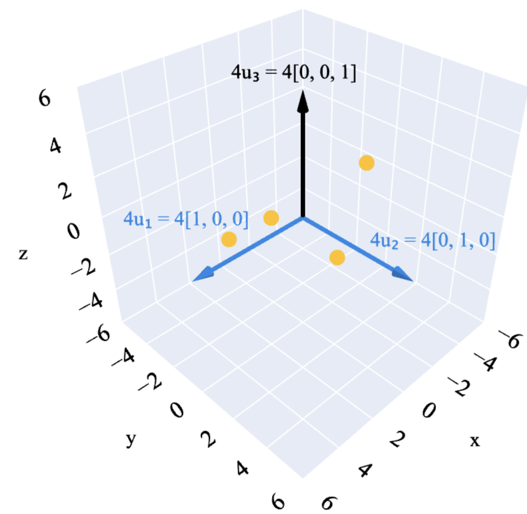
The first two eigenvectors, $\vec{\mathbf{u}}_1$ and $\vec{\mathbf{u}}_2$, define the best plane to project the data onto. This

[4]Computations for the eigenvalues given here.

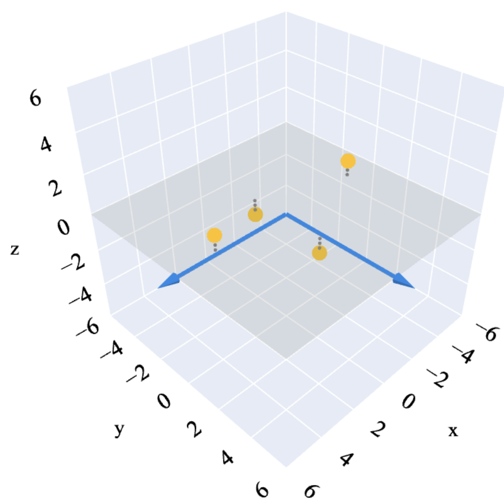
is the optimal 2D reconstruction of the original 3D data that minimizes the average reconstruction error. We illustrate all of this in Figure 2 below. To show that the average squared reconstruction error for this example is 1, *think about the projection of any point onto the plane* $z = 0$. Solutions provided in Section 6.1.



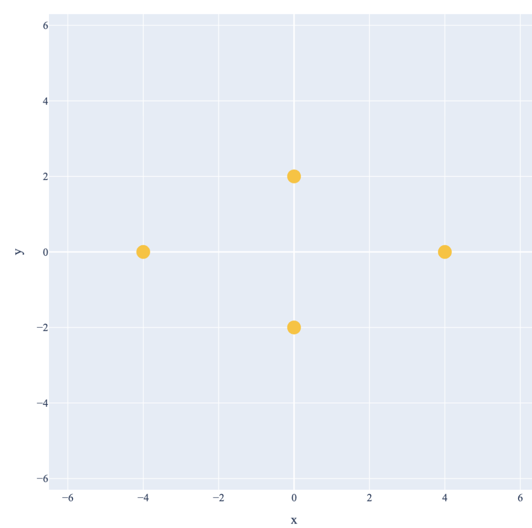
(a) Data-points in $\mathbb{R}^3$.



(b) Eigenvectors $\vec{\mathbf{u}}_1$, $\vec{\mathbf{u}}_2$, and $\vec{\mathbf{u}}_3$ of $\mathbf{M}$, scaled by a factor of 4.



(c) Projection of points onto the plane spanned by $\vec{\mathbf{u}}_1$ and $\vec{\mathbf{u}}_2$



(d) The projected data in the 2D plane spanned by $\vec{\mathbf{u}}_1$ and $\vec{\mathbf{u}}_2$.

FIGURE 2. PCA in $\mathbb{R}^3$: data projected onto a plane defined by top eigenvectors.

This 3D example reinforces the same principle from the 2D case: the top k eigenvectors of the matrix $\mathbf{M} = \frac{1}{N}\mathbf{A}\mathbf{A}^\top$ define the k-dimensional space that minimizes the average reconstruction error when the data is projected onto it.

4. Proof for best orthonormal basis

Note: Any references and citations to **MML** refer to the *Mathematics for Machine Learning* textbook, which can be found here.

Goal: Given a centered dataset $\{\vec{\mathbf{v}}_i \in \mathbb{R}^d\}_{i=1}^N$ (i.e. mean of dataset is $\vec{\mathbf{0}}$), find an orthonormal basis $\{\vec{\mathbf{u}}_j \in \mathbb{R}^d\}_{j=1}^k$ that minimizes the following reconstruction error:

$$\min \frac{1}{N} \sum_{i=1}^{N} \|\vec{\mathbf{v}}_i - \widehat{\vec{\mathbf{v}}}_i\|_2^2 \tag{14}$$

Claim: The best orthonormal basis $\{\vec{\mathbf{u}}_j \in \mathbb{R}^d\}_{j=1}^k$ that minimizes the above reconstruction error is the k eigenvectors with the largest eigenvalues of the following symmetric matrix $\mathbf{M} \in \mathbb{R}^{N \times N}$:

$$\mathbf{M} = \frac{1}{N}\mathbf{A}\mathbf{A}^\top \tag{15}$$

where $\mathbf{A} \in \mathbb{R}^{d \times N}$ and contains $\vec{\mathbf{v}}_i$ as its column vectors, and $\mathbf{A}^\top \in \mathbb{R}^{N \times d}$ and contains $\vec{\mathbf{v}}_i^\top$ as its row vectors:

$$\mathbf{A} = \begin{bmatrix} | & | & \vdots & | \\ \vec{\mathbf{v}}_1 & \vec{\mathbf{v}}_2 & \vdots & \vec{\mathbf{v}}_N \\ | & | & \vdots & | \end{bmatrix}, \mathbf{A}^\top = \begin{bmatrix} - & \vec{\mathbf{v}}_1^\top & - \\ - & \vec{\mathbf{v}}_2^\top & - \\ - & \cdots & - \\ - & \vec{\mathbf{v}}_N^\top & - \end{bmatrix} \tag{16}$$

4.1. **Re-write average reconstruction error objective.** Suppose we have an orthonormal basis $\{\vec{\mathbf{u}}_j \in \mathbb{R}^d\}_{j=1}^d$.
Then, we can write our datapoints $\vec{\mathbf{v}}_i$ in terms of the basis:

$$\vec{\mathbf{v}}_i = \alpha_1\vec{\mathbf{u}}_1 + \alpha_2\vec{\mathbf{u}}_2 + \cdots + \alpha_d\vec{\mathbf{u}}_d = \sum_{j=1}^{d} \alpha_j \vec{\mathbf{u}}_j \tag{17}$$

Now, let's use only the first k orthonormal vectors $\vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2, \ldots, \vec{\mathbf{u}}_k$ to approximate a datapoint $\vec{\mathbf{v}}_i$:

$$\vec{\mathbf{v}}_i \approx \widehat{\vec{\mathbf{v}}}_i = \alpha_1\vec{\mathbf{u}}_1 + \alpha_2\vec{\mathbf{u}}_2 + \cdots + \alpha_k\vec{\mathbf{u}}_k = \sum_{j=1}^{k} \alpha_j \vec{\mathbf{u}}_j \tag{18}$$

Now, we'll show that the average reconstruction error objective can be re-written as follows:

$$L = \frac{1}{N} \sum_{i=1}^{N} \|\vec{\mathbf{v}}_i - \widehat{\vec{\mathbf{v}}}_i\|_2^2 = \sum_{j=k+1}^{d} \vec{\mathbf{u}}_j^\top \mathbf{M} \vec{\mathbf{u}}_j \tag{19}$$

First, we'll focus on re-writing the squared Euclidean distance term (omitting i for now). Plugging in Equation (17) and Equation (18) for $\vec{\mathbf{v}}$ and $\widehat{\vec{\mathbf{v}}}$:

$$\|\vec{\mathbf{v}} - \widehat{\vec{\mathbf{v}}}\|_2^2 = \|\alpha_{k+1}\vec{\mathbf{u}}_{k+1} + \cdots + \alpha_d \vec{\mathbf{u}}_d\|_2^2 = \|\sum_{j=k+1}^{d} \alpha_j \vec{\mathbf{u}}_j\|_2^2 \tag{20}$$

$$= (\sum_{j=k+1}^{d} \alpha_j \vec{\mathbf{u}}_j)^\top (\sum_{j=k+1}^{d} \alpha_j \vec{\mathbf{u}}_j), \text{ using } \|\vec{\mathbf{v}}\|_2^2 = \vec{\mathbf{v}}^\top \vec{\mathbf{v}} \tag{21}$$

$$= (\sum_{j=k+1}^{d} \alpha_j \vec{\mathbf{u}}_j^\top)(\sum_{j=k+1}^{d} \alpha_j \vec{\mathbf{u}}_j) = \sum_{i=k+1}^{d} \sum_{j=k+1}^{d} \alpha_i \alpha_j \vec{\mathbf{u}}_i^\top \vec{\mathbf{u}}_j \tag{22}$$

$$= \sum_{j=k+1}^{d} \alpha_j^2 \vec{\mathbf{u}}_j^\top \vec{\mathbf{u}}_j, \text{ using } \vec{\mathbf{u}}_i^\top \vec{\mathbf{u}}_j = 0 \text{ if } i \neq j \text{ (orthonormal property)} \tag{23}$$

$$= \sum_{j=k+1}^{d} \alpha_j^2, \text{ using } \vec{\mathbf{u}}_i^\top \vec{\mathbf{u}}_j = 1 \text{ if } i = j \text{ (orthonormal property)} \tag{24}$$

$$= \sum_{j=k+1}^{d} (\vec{\mathbf{v}} \cdot \vec{\mathbf{u}}_j)^2, \text{ using } \alpha_j = \vec{\mathbf{v}} \cdot \vec{\mathbf{u}}_j \text{ (best projection)} \tag{25}$$

$$= \sum_{j=k+1}^{d} (\vec{\mathbf{u}}_j^\top \vec{\mathbf{v}})(\vec{\mathbf{v}}^\top \vec{\mathbf{u}}_j), \text{ using } \vec{\mathbf{u}} \cdot \vec{\mathbf{v}} = \vec{\mathbf{u}}^\top \vec{\mathbf{v}} = \vec{\mathbf{v}}^\top \vec{\mathbf{u}} = \vec{\mathbf{v}} \cdot \vec{\mathbf{u}} \tag{26}$$

Now, plugging Equation (26) back into our objective function (and adding i back in), we get the following:

$$L = \frac{1}{N}\sum_{i=1}^{N} \|\vec{\mathbf{v}}_i - \widehat{\vec{\mathbf{v}}}_i\|_2^2 = \frac{1}{N}\sum_{i=1}^{N} \sum_{j=k+1}^{d} (\vec{\mathbf{u}}_j^\top \vec{\mathbf{v}}_i)(\vec{\mathbf{v}}_i^\top \vec{\mathbf{u}}_j) \tag{27}$$

$$= \sum_{j=k+1}^{d} \vec{\mathbf{u}}_j^\top (\frac{1}{N}\sum_{i=1}^{N} \vec{\mathbf{v}}_i \vec{\mathbf{v}}^\top) \vec{\mathbf{u}}_j \tag{28}$$

$$= \sum_{j=k+1}^{d} \vec{\mathbf{u}}_j^\top (\frac{1}{N}\mathbf{A}\mathbf{A}^\top) \vec{\mathbf{u}}_j, \text{ using matrix multiplication definition} \tag{29}$$

$$= \sum_{j=k+1}^{d} \vec{\mathbf{u}}_j^\top \mathbf{M} \vec{\mathbf{u}}_j, \text{ using } \mathbf{M} = \frac{1}{N}\mathbf{A}\mathbf{A}^\top \tag{30}$$

Thus, we've shown Equation (19), namely that our objective L can be re-written as Equation (30).

4.2. **Solve constrained optimization problem.** Recall our goal: find k orthonormal vectors $\{\vec{\mathbf{u}}_j \in \mathbb{R}^d\}_{j=1}^{k}$ that minimizes our objective L. We can write this explicitly as a constrained optimization problem ($\forall$ = "for all"):

$$\min \sum_{j=k+1}^{d} \vec{\mathbf{u}}_j^\top \mathbf{M} \vec{\mathbf{u}}_j \text{ subject to (s.t.) } \vec{\mathbf{u}}_j^\top \vec{\mathbf{u}}_j = 1, \forall j \text{ (unit vector)} \tag{31}$$

Using Lagrange multipliers (see Section 5), we can solve this constrained optimization problem by solving the following Lagrange function $\mathcal{L}$:

$$\mathcal{L} = \sum_{j=k+1}^{d} \left(\vec{\mathbf{u}}_j^\top \mathbf{M} \vec{\mathbf{u}}_j + \lambda_j (1 - \vec{\mathbf{u}}_j^\top \vec{\mathbf{u}}_j) \right) \tag{32}$$

To solve this, we set the partial derivatives $\frac{\partial \mathcal{L}}{\partial \lambda_i} = 0, \frac{\partial \mathcal{L}}{\vec{\mathbf{u}}_i} = \vec{\mathbf{0}}$ and solving for the variables.

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = 1 - \vec{\mathbf{u}}_i^\top \vec{\mathbf{u}}_i = 0 \tag{33}$$

$$\rightarrow \vec{\mathbf{u}}_i^\top \vec{\mathbf{u}}_i = 1 \tag{34}$$

$$\frac{\partial \mathcal{L}}{\vec{\mathbf{u}}_i} = \vec{\mathbf{u}}_i (\mathbf{M} + \mathbf{M}^\top) - \lambda_i \vec{\mathbf{u}}_i^\top (\mathbf{I} + \mathbf{I}^\top), \text{ using } \frac{\partial \vec{\mathbf{x}}^\top \mathbf{B} \vec{\mathbf{x}}}{\partial \vec{\mathbf{x}}} = \vec{\mathbf{x}}^\top (\mathbf{B} + \mathbf{B}^\top) \text{ (MML 5.107)} \tag{35}$$

$$= 2\vec{\mathbf{u}}_i^\top \mathbf{M} - 2\lambda_i \vec{\mathbf{u}}_i^\top \mathbf{I}, \text{ using symmetry: } \mathbf{M} = \mathbf{M}^\top, \mathbf{I} = \mathbf{I}^\top \tag{36}$$

$$= 2\vec{\mathbf{u}}_i^\top \mathbf{M} - 2\lambda_i \vec{\mathbf{u}}_i^\top, \text{ using } \mathbf{BI} = \mathbf{B} \text{ (MML 2.20)} \tag{37}$$

Now, by setting $\frac{\partial \mathcal{L}}{\vec{\mathbf{u}}_i} = 0$, we get the following:

$$2\vec{\mathbf{u}}_i^\top \mathbf{M} = 2\lambda_i \vec{\mathbf{u}}_i^\top \tag{38}$$

$$\vec{\mathbf{u}}_i^\top \mathbf{M} = \lambda_i \vec{\mathbf{u}}_i^\top \tag{39}$$

$$(\vec{\mathbf{u}}_i^\top \mathbf{M})^\top = (\lambda_i \vec{\mathbf{u}}_i^\top)^\top, \text{ taking the transpose on both sides} \tag{40}$$

$$\mathbf{M}^\top \vec{\mathbf{u}}_i = \lambda_i \vec{\mathbf{u}}_i, \text{ using } (\mathbf{B}^\top)^\top = \mathbf{B}, (\mathbf{BC})^\top = \mathbf{C}^\top \mathbf{B}^\top \text{ (MML 2.29, 2.31)} \tag{41}$$

$$\mathbf{M} \vec{\mathbf{u}}_i = \lambda_i \vec{\mathbf{u}}_i, \text{ using symmetry: } \mathbf{M} = \mathbf{M}^\top \tag{42}$$

From Equation (42), we get that the solution to $\mathcal{L}$ are the eigenvectors $\vec{\mathbf{u}}_i$ of $\mathbf{M}$ with eigenvalues λ_i. Because $\mathbf{M}$ is symmetric, its eigenvectors form an orthonormal basis, thereby satisfying Equation (34), and its eigenvalues are real-valued numbers, i.e. $\lambda_i \in \mathbb{R}$. Because $\mathbf{M}$ is positive semidefinite (MML Theorem 4.14), its eigenvalues are positive, i.e. $\lambda_i > 0$ (MML pg. 106).

Now, plugging Equation (42) back into our objective Equation (30), we get the following:

$$L = \sum_{j=k+1}^{d} \vec{\mathbf{u}}_j^\top \mathbf{M} \vec{\mathbf{u}}_j = \sum_{j=k+1}^{d} \vec{\mathbf{u}}_j^\top (\lambda_j \vec{\mathbf{u}}_j) \tag{43}$$

$$= \sum_{j=k+1}^{d} \lambda_j (\vec{\mathbf{u}}_j^\top \vec{\mathbf{u}}_j) = \sum_{j=k+1}^{d} \lambda_j, \text{ using } \vec{\mathbf{u}}_j^\top \vec{\mathbf{u}}_j = 1 \text{ (orthonormal property)} \tag{44}$$

Then, to minimize L, we need $\vec{\mathbf{u}}_{k+1}, \ldots, \vec{\mathbf{u}}_d$ to be the $(d-k)$ eigenvectors of $\mathbf{M}$ with the smallest eigenvalues λ_j's. Recall our original goal, which was to find the k orthonormal vectors $\{\vec{\mathbf{u}}_j\}_{j=1}^k$ that minimizes L. This set of vectors then must be the k eigenvectors of $\mathbf{M}$ with the largest eigenvalues λ_j's.

From Equation (44), the average reconstruction error L is the sum of the $(d-k)$ smallest eigenvalues of $\mathbf{M}$. Then, when choosing k, we can use the eigenvalues sorted by magnitude to decide how much error we're willing to allow in order for a small choice of k.

5. Constrained optimization

So far, we've primarily focused on unconstrained optimization problems of the following form:

$$\min_{\vec{\mathbf{x}}} f(\vec{\mathbf{x}}) \text{ or } \max_{\vec{\mathbf{x}}} f(\vec{\mathbf{x}}) \tag{45}$$

A constrained optimization adds on several constraints or equations $e_1, e_2, \ldots$ that need to be satisfied:

$$\min_{\vec{\mathbf{x}}} f(\vec{\mathbf{x}}) \text{ subject to (s.t.) } e_1, e_2, \ldots \tag{46}$$

For instance, here's a constrained problem with one constraint:

$$\min_{\vec{\mathbf{x}}} f(\vec{\mathbf{x}}) \text{ s.t. } g(\vec{\mathbf{x}}) = 0 \tag{47}$$

One way to solve constrained problems like the above is to use its analogous Lagrange function $\mathcal{L}$:

$$\mathcal{L}(\vec{\mathbf{x}}, \lambda) = f(\vec{\mathbf{x}}) + \lambda g(\vec{\mathbf{x}}) \tag{48}$$

It turns out the solution to the original constrained problem (Equation (47)) is a saddle point of $\mathcal{L}$ (Equation (48)).

A saddle point for a function occurs where all partial derivatives for that function equal 0. For instance, for $\mathcal{L}$, a saddle point occurs where

$$\frac{\partial \mathcal{L}}{\partial \vec{\mathbf{x}}} = \vec{\mathbf{0}} \text{ and } \frac{\partial \mathcal{L}}{\partial \lambda} = 0 \tag{49}$$

Consider a constrained problem with multiple constraints:

$$\min_{\vec{\mathbf{x}}} f(\vec{\mathbf{x}}) \text{ s.t. } g_j(\vec{\mathbf{x}}) = 0 \text{ for } j = 1, \ldots, M \tag{50}$$

Then, its analogous Lagrange function is given as follows:

$$\mathcal{L}(\vec{\mathbf{x}}, \lambda) = f(\vec{\mathbf{x}}) + \sum_{j=1}^{M} \lambda_j g_j(\vec{\mathbf{x}}) \tag{51}$$

and a saddle point occurs when

$$\frac{\partial \mathcal{L}}{\partial \vec{\mathbf{x}}} = \vec{\mathbf{0}} \text{ and } \frac{\partial \mathcal{L}}{\partial \lambda_j} = 0 \text{ for all } j \tag{52}$$

To find a saddle point of $\mathcal{L}$ (and a solution to the original constrained optimization problem), compute the partial derivatives of $\mathcal{L}$, set them to 0, and solve for the variables.

6. Solutions

6.1. Average Reconstruction Error for 3D Example. Recall the optimization for minimizing the reconstruction error, Equation (1), copied below:

$$\min \frac{1}{N}\sum_{i=1}^{N} \|\vec{\mathbf{v}}_i - \widehat{\vec{\mathbf{v}}}_i\|_2^2$$

Since $\vec{\mathbf{u}}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix}$ and $\vec{\mathbf{u}}_2 = \begin{bmatrix}0\\1\\0\end{bmatrix}$, projecting any vector $\vec{\mathbf{v}}_i = \begin{bmatrix}x_i\\y_i\\z_i\end{bmatrix}$ onto the span of $\vec{\mathbf{u}}_1$ and $\vec{\mathbf{u}}_2$ amounts to zeroing out the z-component:

$$\widehat{\vec{\mathbf{v}}}_i = \begin{bmatrix}x_i\\y_i\\0\end{bmatrix} \quad \Rightarrow \quad \vec{\mathbf{v}}_i - \widehat{\vec{\mathbf{v}}}_i = \begin{bmatrix}0\\0\\z_i\end{bmatrix}$$

The squared reconstruction error for each point is then:

$$\|\vec{\mathbf{v}}_i - \widehat{\vec{\mathbf{v}}}_i\|_2^2 = z_i^2$$

Since the dataset is:

$$\vec{\mathbf{v}}_1 = \begin{bmatrix}4\\0\\1\end{bmatrix}, \quad \vec{\mathbf{v}}_2 = \begin{bmatrix}0\\2\\-1\end{bmatrix}, \quad \vec{\mathbf{v}}_3 = \begin{bmatrix}-4\\0\\1\end{bmatrix}, \quad \vec{\mathbf{v}}_4 = \begin{bmatrix}0\\-2\\-1\end{bmatrix}$$

We compute the total squared error:

$$\text{Total Error} = 1^2 + (-1)^2 + 1^2 + (-1)^2 = 4 \quad \Rightarrow \quad \text{Average Error} = \frac{1}{4}\cdot 4 = 1$$

So, the average reconstruction error when using a 2D projection is:

$$\boxed{\frac{1}{N}\sum_{i=1}^{N} \|\vec{\mathbf{v}}_i - \widehat{\vec{\mathbf{v}}}_i\|_2^2 = 1}$$