Fundamentals / Foundation

Data Science 9CFU

Computer Science 6CFU

# of Data Science

Indro Spinelli

Sapienza University of Rome

Linear Transformation (Ax)

Given a vector $x \in \mathbb{R}^n$and a matrix $A \in \mathbb{R}^{m \times n}$,the product $y = Ax$ transforms x into a new vector $y \in \mathbb{R}^m$.

The effect depends on the structure of A:

- **Scaling** – If A is diagonal, each axis is stretched or shrunk by the diagonal entries.
- **Rotation / Reflection** – If A is orthonormal ($A^\top A = I$,(the transformation is a pure rotation or reflection (no scaling).
- **Shearing** – If A has off-diagonal entries, directions can get tilted.
- **Projection** – If A has rank k$< n$, the transformation “collapses” vectors into a lower-dimensional subspace.

So geometrically, Ax is a **linear transformation** of space.

Rotation

Given an orthonormal basis C → $\mathbf{C}^T\mathbf{C} = \mathbf{I}_{N\times N}$.

We can project the points from the original basis to the new one as simply as

$$\mathbf{w}_p = \mathbf{C}^T\mathbf{x}_p \qquad p = 1...P.$$

The inverse process is straightforward too:

$$\mathbf{C}\,\mathbf{C}^T\mathbf{x}_p = \mathbf{x}_p \qquad p = 1...P.$$

While C^T encodes C decodes.

Projection

If our orthonormal basis is composed of $K \leq N$ spanning vectors. We are going going to be able to represent perfectly the original points in the new space.

Meaning that while $\mathbf{w}_p = \mathbf{C}^T \mathbf{x}_p$ still holds. We are going to lose information and we are not going to be able to perfectly represent the point $\mathbf{C}\,\mathbf{C}^T \mathbf{x}_p \approx \mathbf{x}_p$ which is the orthogonal projection of x_p on C

Reconstruction Error

For each point we want to minimize: $h\left(\mathbf{w}_p\right) = \left\|\mathbf{C}\mathbf{w}_p - \mathbf{x}_p\right\|_2^2.$

Which leads to the following objective: $\underset{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_P}{\text{minimize}} \ \frac{1}{P} \sum_{p=1}^{P} \left\|\mathbf{C}\mathbf{w}_p - \mathbf{x}_p\right\|_2^2.$

Plugging in the solution we found above into the optimization objective:

$$g\left(\mathbf{C}\right) = \frac{1}{P} \sum_{p=1}^{P} \left\|\mathbf{C}\ \mathbf{C}^T \mathbf{x}_p - \mathbf{x}_p\right\|_2^2.$$

Orthogonal Projection $CC^{T}x_p$

The (unreducible) error is now the perpendicular distance from x_p to this subspace spanned by C

Basis $C = [c_1, c_2]$ with orthonormal columns.

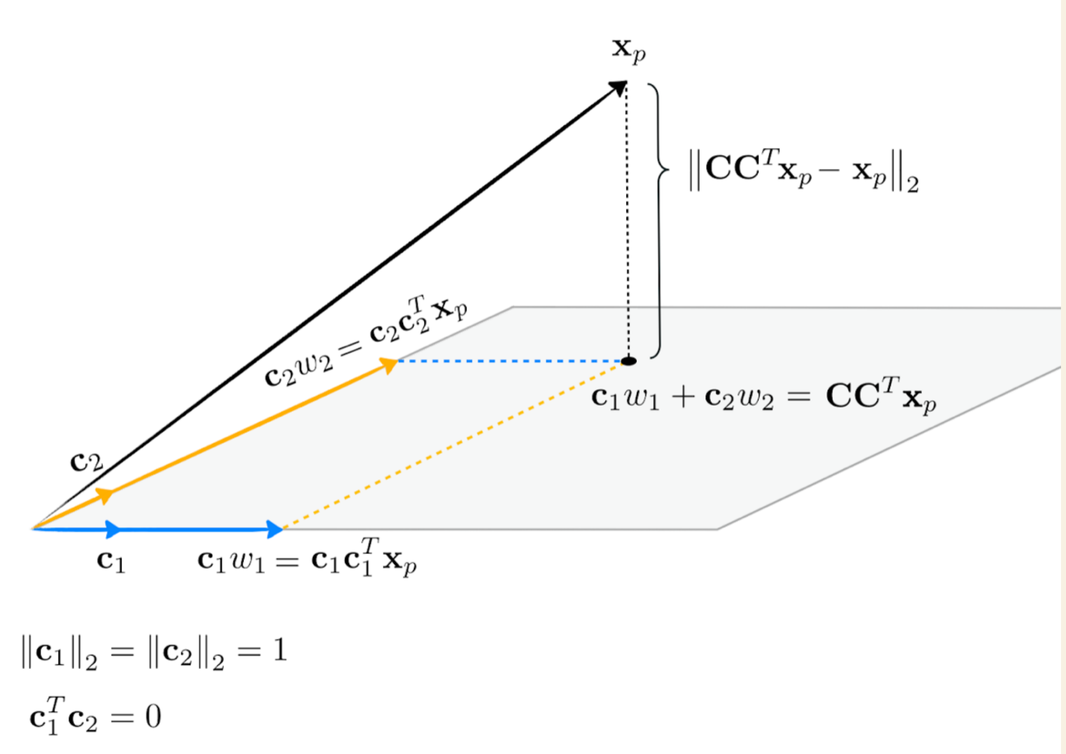
Optimal weights: $w_1 = c_1^\top x_p,\ w_2 = c_2^\top x_p.$

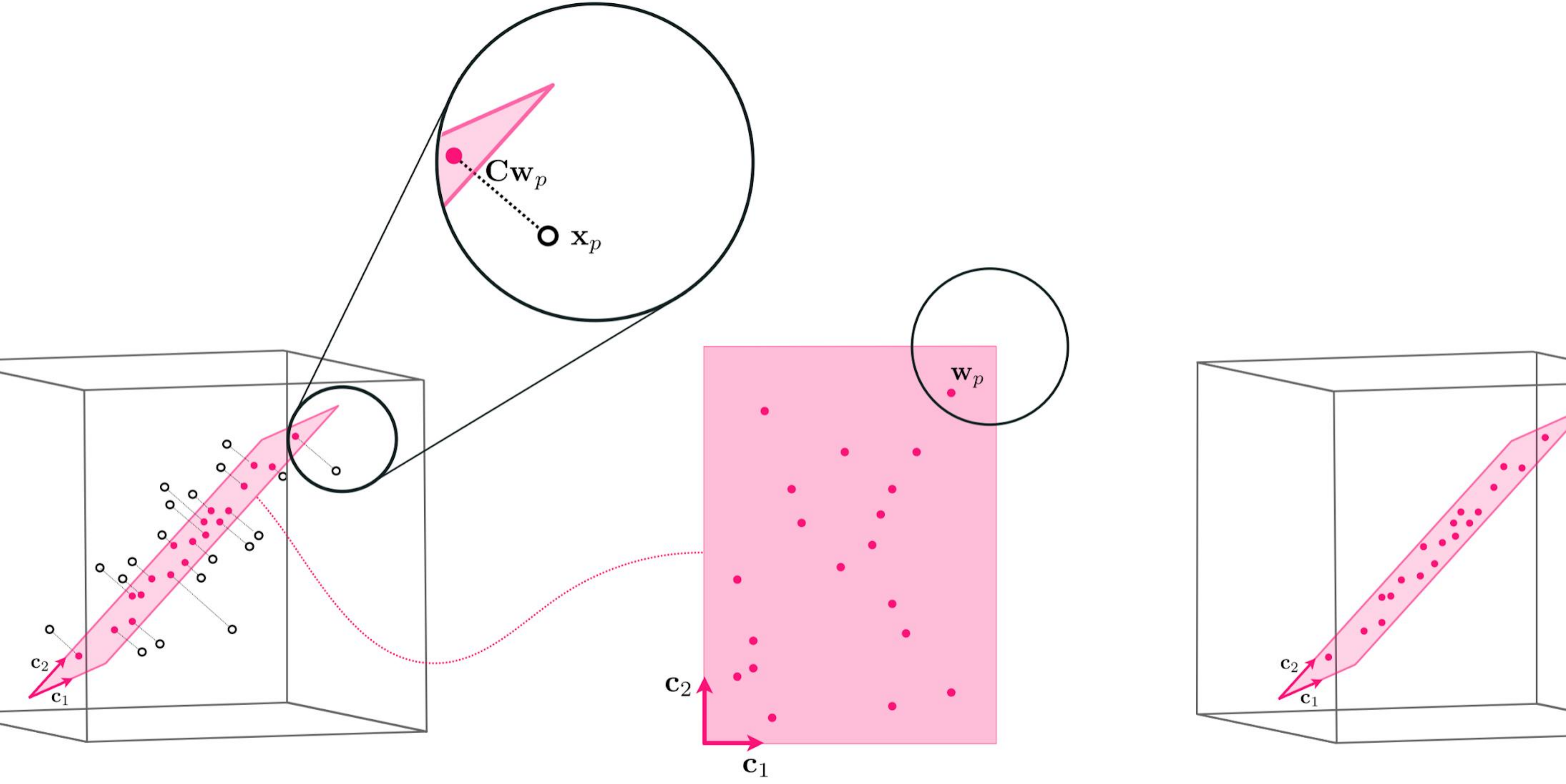
Reconstruction:

$$CC^\top x_p = c_1(c_1^\top x_p) + c_2(c_2^\top x_p).$$

(the sum of the blue and orange arrows in the figure).

Error: $\|CC^\top x_p - x_p\|_2$ (black dotted line from x_p to the plane).





$\mathbf{C}\mathbf{w}_p$
$\mathbf{x}_p$
$\mathbf{w}_p$
$\mathbf{c}_2$
$\mathbf{c}_1$
original
encoded
decoded

Learning the Basis

For now we always assumed that the basis C was given and we were only interested into learing the new representation for each point w.

Now we will talk about the most fundamental unsupervised learning method known as Principal Component Analysis or PCA for short.

Instead of just learning the proper weights to best represent input data over a given fixed spanning set we learn a proper spanning set as well.

Learning the Basis

This simplified cost function is known as the autoencoder because we learn both the encoding (via the learned weights w_p) and the decoding (via the projection Cw_p) for each data point.

The optimization problem becames: $g\left(\mathbf{w}_1, \ldots, \mathbf{w}_P, \mathbf{C}\right) = \frac{1}{P} \sum_{p=1}^{P} \left\|\mathbf{C}\mathbf{w}_p - \mathbf{x}_p\right\|_2^2.$

This cost function can be properly minimized using any number of standard approaches.

However, we have seen that if C is orthonormal the objective can be simplified to:

$$g\left(\mathbf{C}\right) = \frac{1}{P} \sum_{p=1}^{P} \left\|\mathbf{C}\,\mathbf{C}^T\mathbf{x}_p - \mathbf{x}_p\right\|_2^2.$$

Importantly, we need not constrain the minimization to enforce the orthonormality because it can be shown that the minima is always orthonormal.

Expanding the Cost Function

Consider one summand of the Autoencoder cost function, assuming C is orthogonal and expand the objective:

$$\left\|\mathbf{C}\,\mathbf{C}^T\mathbf{x}_p - \mathbf{x}_p\right\|_2^2 = \mathbf{x}_p^T\mathbf{C}^T\mathbf{C}\mathbf{C}\mathbf{C}^T\mathbf{x}_p - 2\mathbf{x}_p^T\mathbf{C}\mathbf{C}^T\mathbf{x}_p + \mathbf{x}_p^T\mathbf{x}_p$$

Since $C^TC = I$

$$= -\mathbf{x}_p^T\mathbf{C}\mathbf{C}^T\mathbf{x}_p + \mathbf{x}_p^T\mathbf{x}_p = -\left\|\mathbf{C}^T\mathbf{x}_p\right\|_2^2 + \left\|\mathbf{x}_p\right\|_2^2.$$

Because x_p is fixed with respect to the optimization over C, minimizing the original expression is equivalent to **maximizing** $\| C^\top x_p \|_2^2$

Expanding the Cost Function

If we expand across the basis elements: $\left\|\mathbf{C}^T\mathbf{x}_p\right\|_2^2 = \sum_{n=1}^{N}\left(\mathbf{c}_n^T\mathbf{x}_p\right)^2,$

which leads us to the following optimization objective

$$g\left(\mathbf{C}\right) = -\frac{1}{P}\sum_{p=1}^{P}\sum_{n=1}^{K}\left(\mathbf{c}_n^T\mathbf{x}_p\right)^2.$$

There are no cross terms between different c_i, c_j when $i \neq j$.
Therefore, we can optimize each basis element independently.

Optimization for the First Basis Vector

Consider the first basis vector c_1: $h\left(\mathbf{c}_1\right) = \frac{1}{P}\sum_{p=1}^{P}\left(\mathbf{c}_1^T\mathbf{x}_p\right)^2.$

Here $c_1^\mathsf{T} x_p$ is the **projection** (scalar) of data point x_p onto the direction c_1.

Squaring and averaging gives the **variance of the data along direction** c_1 $(assuming\ the\ data\ is\ mean\ centered)$

So $h(c_1)$ =variance of data along axis c_1. remember c_1 is unit length

Optimization for the First Basis Vector

Since c_1 is constrained to unit norm, $h(c_1)$ represents the **sample variance** of the dataset along direction c_1 .Our goal is therefore to maximize this variance.

Stacking data samples into the matrix $X = [x_1, x_2, \ldots, x_P] \in \mathbb{R}^{N \times P}$, we rewrite:

$$h\left(\mathbf{c}_1\right) = \frac{1}{P}\mathbf{c}_1^T \mathbf{X}\,\mathbf{X}^T \mathbf{c}_1 = \mathbf{c}_1^T \left(\frac{1}{P}\mathbf{X}\,\mathbf{X}^T\right)\mathbf{c}_1.$$

Covariance Matrix

The term $\Sigma = \frac{1}{P} X X^\top$ is the **empirical covariance matrix**

Each entry Σ_{ij} tells you the **average correlation** between dimension i and dimension j across the dataset:

If $i = j$: is the **variance** of the i-th feature.

If $i \neq j$: is the **covariance** between features $i$and j.

Σ compactly encodes how features **vary together**.

Geometric Interpretation

Σ describes the **shape of the data cloud** in $\mathbb{R}^N$

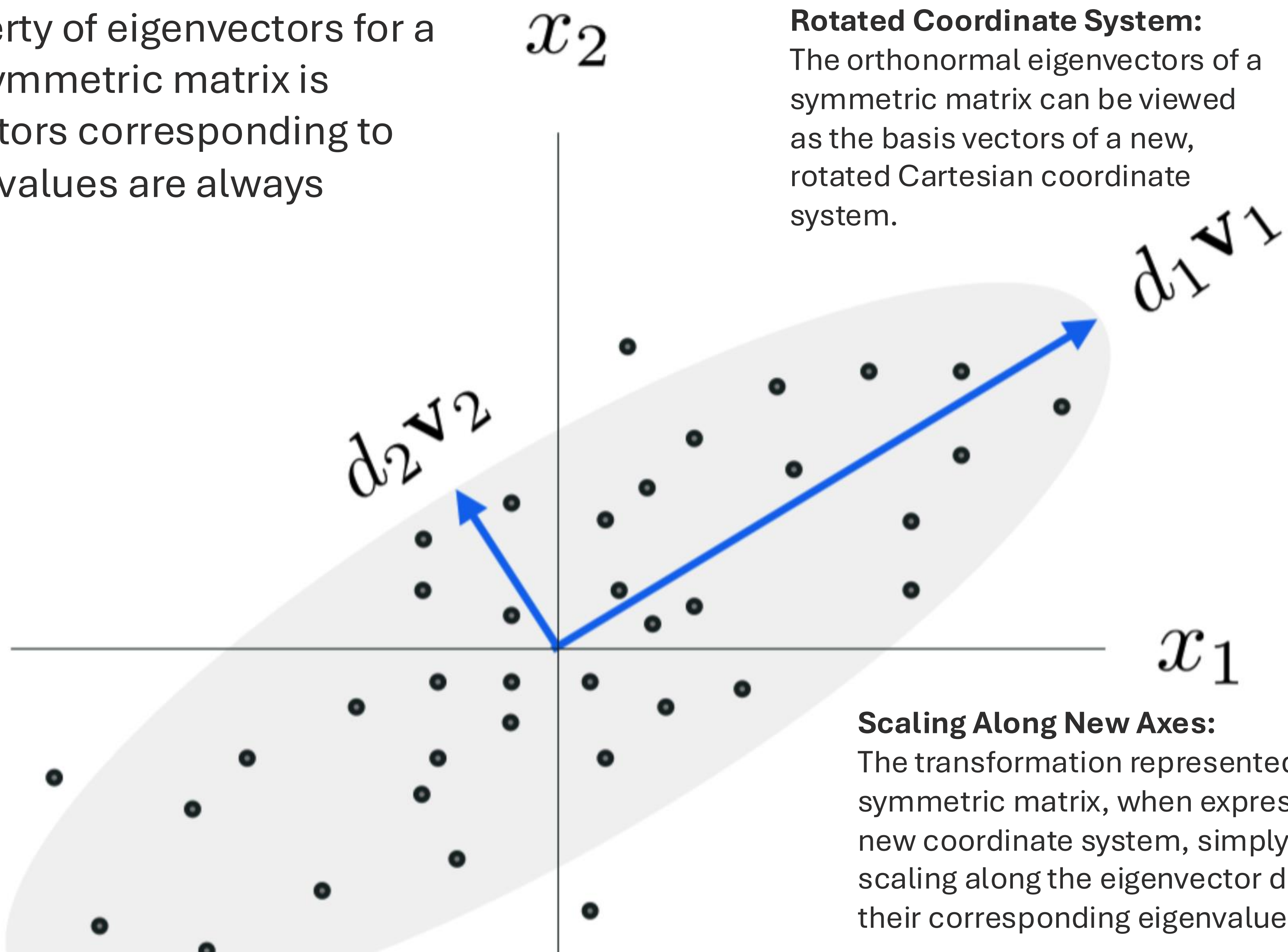
Eigenvectors of Σ point in the **principal directions** where data spreads the most.

Eigenvalues tell you **how much variance** (spread) the data has along those directions.

The covariance matrix tells you the **orientation and elongation** of the data cloud (think of it as an ellipsoid fit to the data).

The key property of eigenvectors for a real square symmetric matrix is that eigenvectors corresponding to distinct eigenvalues are always orthogonal.

Rotated Coordinate System:
The orthonormal eigenvectors of a symmetric matrix can be viewed as the basis vectors of a new, rotated Cartesian coordinate system.



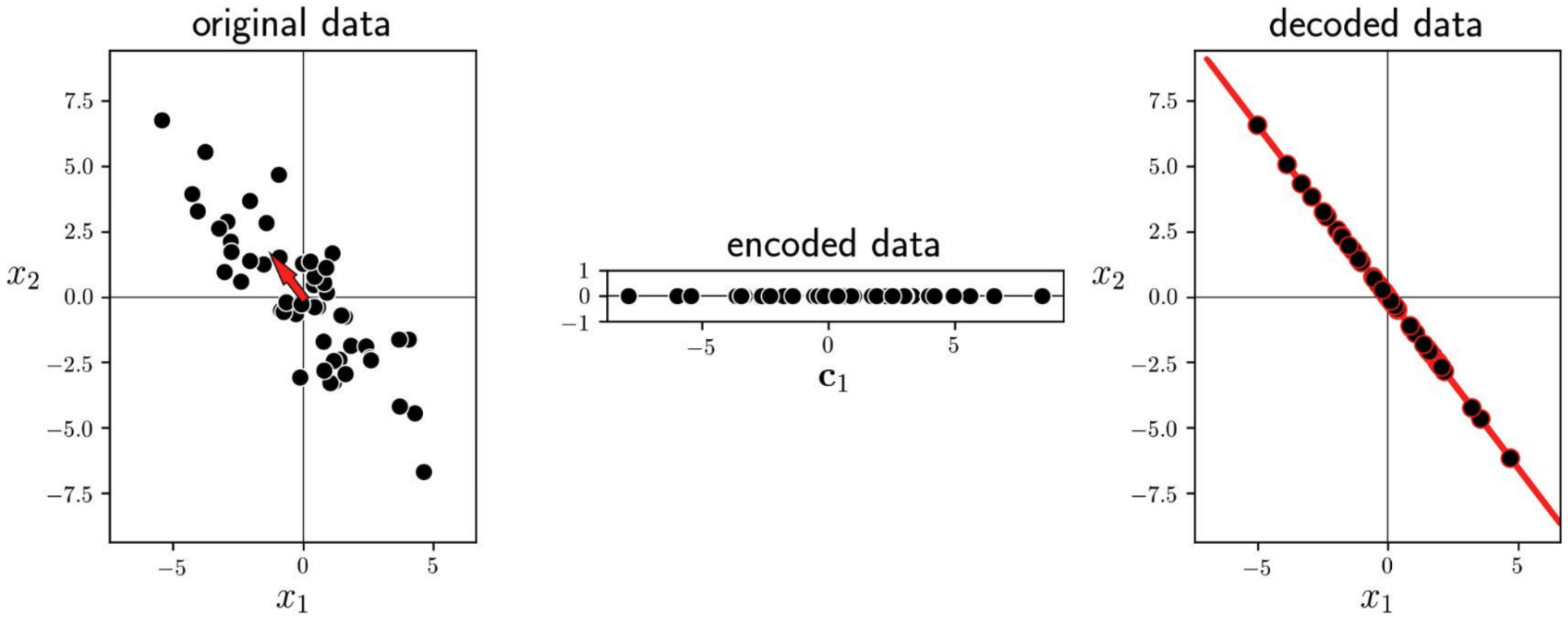
Scaling Along New Axes:
The transformation represented by the symmetric matrix, when expressed in this new coordinate system, simply involves scaling along the eigenvector directions by their corresponding eigenvalues.

Optimization for the First Basis Vector

$h(c_1) = c_1^\top \Sigma c_1$ is maxized when c_1 is the eigenvector of Σ associated with its largest eigenvalue d_1 .

Therefore: $c_1 = v_1$ and $h(v_1)=d_1$

This direction v_1 is the **first principal component** of the dataset.



original data
encoded data
decoded data
x_1
x_2
$\mathbf{c}_1$

Optimization for Subsequent Basis Vectors

Next, consider the second basis vector c_2 .The same reasoning yields:

$$h(c_2) = c_2^\top \Sigma c_2,$$

with the orthogonality constraint $c_1^\top c_2 = 0$.

By the variational characterization of eigenvalues, the maximizer is the eigenvector associated with the second largest eigenvalue d_2:

$$c_2 = v_2, \; h(v_2)=d_2.$$

This direction corresponds to the **second principal component**. The variance explained in this direction is precisely d_2

Conclusion

In summary, under the orthogonality constraint, the autoencoder's minimization problem is equivalent to maximizing variance along successive orthogonal directions. The solution is obtained by selecting the top K eigenvectors of the data covariance matrix Σ. These eigenvectors form the **principal components**, and the corresponding eigenvalues quantify the variance captured by each component.

Thus, the **classic PCA solution emerges naturally as the global minimizer of the linear autoencoder objective**.

Dimensionality Reduction

- PCA finds a new set of orthogonal axes (principal components) aligned with directions of **maximum variance** in the data.
- By keeping only the top k components ($k < d$), we reduce data from d-dimensional space to a lower-dimensional subspace.
- **Goal:** compress data while retaining as much variability (information) as possible.
- **Feature Engineering:** PCA creates new uncorrelated features (principal components) that can simplify downstream tasks, improve model performance, and reduce overfitting.

Unsupervised Learning

- PCA does **not require labels**: it learns structure directly from the data distribution.
- It is the most fundamental **unsupervised representation learning** method.
- Learns a new basis that best explains correlations in the data.
- Helps reveal hidden patterns:
  - uncover directions of strong correlation,
  - detect redundancy,
  - prepare features for clustering, anomaly detection, or other ML tasks.

Exercise

30 minutes

Consider the 2D dataset (not mean-centered):

$$x_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Do everything **by hand** (calculator allowed for arithmetic):

1. Compute the sample mean $\bar{x}$ and mean-center the data $c_p = x_p - \bar{x}$.
2. Compute the empirical covariance matrix $\Sigma = \frac{1}{P} \sum_{p=1}^{3} c_p c_p^\top$ (use $P = 3$).
3. Find the eigenvalues and eigenvectors of Σ.
4. Identify the first principal component v_1 (unit vector) and the second v_2.
5. Project the centered samples onto v_1 (compute the scalar scores $z_p = v_1^\top c_p$).
6. Reconstruct each sample from the 1D projection and compute the mean squared reconstruction error (MSE).
7. Compute the explained-variance ratio of the first PC (as a percentage).
8. Briefly interpret the sign of the off-diagonal covariance entry.

When does it work?

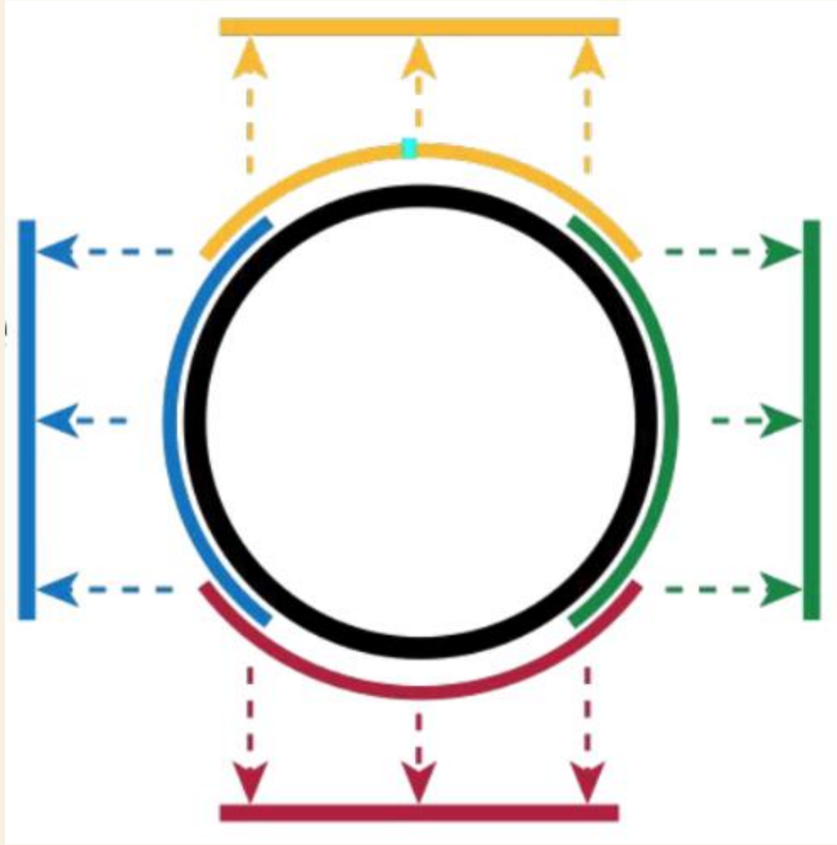
High dimensional data can often be represented using a much lower dimensional space. This happens when the data lives near a linear manifold in the high dimensional space.

If we find the manifold we can project the data in the manifold and using to represent the data without losing much information.

Manifold

A Manifold is a topological space that locally resembles the Euclidean space.

Consider the upper half of the unit circle, x2+ y2= 1, where $y \geq 0$ (yellow arc). Every point on this arc can be uniquely identified by its x-coordinate top(x,y) = x



The projection onto the first coordinate defines a smooth and invertible mapping from the upper arc to the open interval $(-1,1)$

Functions which provide a one-to-one correspondence between open regions of a surface and subsets of Euclidean space, are called charts.

Charts

Each chart can be seen as a mapping $\phi : R^1 \rightarrow S \subset R^2$.

ϕ must be smooth and invertible (diffeomorphism). The key property is that both the function and its inverse are continuously differentiable.

The domain of ϕ is the parametric space and is Euclidean.

The image of ϕ is the embedding and is a surface.

Manifolds in the end are unions of charts.