DSC5001: Statistical Machine Learning I

Week 12: Unsupervised Learning: PCA and Clustering

Kangning Cui
City University of Hong Kong (Dongguan)

Semester A, 2026/27

Project Submission

Final report and code

- **ZIP option:** Submit one ZIP containing the final report (PDF) and the complete source code.
- **GitHub option:** Submit the report (PDF) and an accessible GitHub repository link containing the complete source code.
- Include the files and instructions needed to run the project.

Presentation

- Upload group presentation slides to Canvas by the **evening of 27 November 2026**.
- In-class group presentations: **30 November 2026**.

Learning Objectives

- PCA and explained variance
- Projection, reconstruction, and SVD
- K-means clustering
- Using PCA with clustering

Dimension Reduction

DSC5001 ▪ Semester A, 2026/27

High-Dimensional Data

A dataset may contain many correlated measurements. For example, several sensors can respond to the same underlying change. Examining every feature separately then becomes difficult.

$$\boxed{\boldsymbol{x}_i \in \mathbb{R}^p \quad \longrightarrow \quad \boldsymbol{z}_i \in \mathbb{R}^r, \qquad r < p.}$$

Here p is the number of original features, r is the number retained, and i identifies an observation.

PCA chooses the reduced coordinates to preserve as much variation in the observed data as possible.

Reduced Coordinates

A lower-dimensional representation stores each observation using coordinates along a few shared directions rather than all p original features.

$$\boxed{\boldsymbol{z}_i = \boldsymbol{V}_r^\top (\boldsymbol{x}_i - \bar{\boldsymbol{x}}), \qquad \boldsymbol{V}_r \in \mathbb{R}^{p \times r}.}$$

The r columns of $\boldsymbol{V}_r$ are the directions to be selected; $\bar{\boldsymbol{x}}$ is the mean of the training observations.
Before choosing directions, we must first describe variation *around that mean*.

Centering

If every sensor reading increases by a constant, the relationships between observations have not changed. PCA should therefore measure variation relative to the mean rather than relative to zero.

$$\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i, \qquad \widetilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \bar{\boldsymbol{x}}.$$

Here n is the number of observations. The centered vectors sum to zero.
For a new observation, subtract the **training mean**, not a mean recomputed from that observation.

Centering Example

Consider three observations with two features:

$$\boldsymbol{x}_1 = (2, 4)^\top, \quad \boldsymbol{x}_2 = (4, 6)^\top, \quad \boldsymbol{x}_3 = (6, 8)^\top.$$

Their mean is $\bar{\boldsymbol{x}} = (4, 6)^\top$.
Subtracting it from each row gives

$$\boxed{\widetilde{\boldsymbol{X}} = \begin{pmatrix} -2 & -2 \\ 0 & 0 \\ 2 & 2 \end{pmatrix}.}$$

Both centered features rise together. The cloud lies along a diagonal; centering has made its direction of variation easy to see.

Feature Scaling

Centering does not remove differences in measurement units. A feature measured in thousands may dominate the covariance matrix, even if the numerical scale is arbitrary.

$$\boxed{u_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j},}$$

where $\bar{x}_j$ and s_j are the training mean and standard deviation of feature j (assuming $s_j > 0$). **Covariance PCA** preserves the original units; **correlation PCA** uses standardized features u_{ij}. This choice determines which variation receives more weight.

Scaling Example

Suppose two centered features have covariance

$$\boldsymbol{S} = \begin{pmatrix} 100 & 0 \\ 0 & 4 \end{pmatrix}.$$

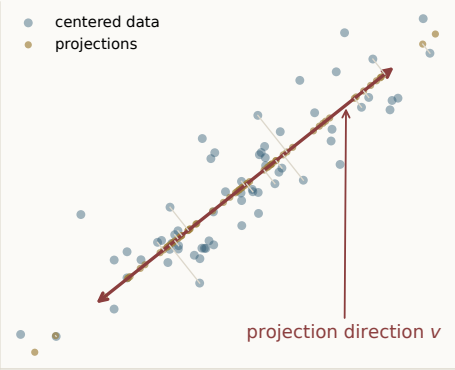
On the original scale, PCA selects the first feature because its variance is 100 rather than 4. Dividing by standard deviations 10 and 2 produces

$$\boxed{\boldsymbol{R} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.}$$

After standardization, neither coordinate has more variance; there is no unique leading direction.

Scaling changes the PCA question.

Projection Geometry



Each original point is represented by a point on the selected line. The segments between them show information lost in the projection.
PCA will choose a line that minimizes the *total squared* loss.

Projection Coordinates

Let $\boldsymbol{v}$ be a unit vector along a chosen line, and let $\widetilde{\boldsymbol{x}}_i$ be a centered observation.
Its coordinate on the line is the inner product

$$\boxed{z_i = \boldsymbol{v}^\top \widetilde{\boldsymbol{x}}_i.}$$

We can return to the original feature space using

$$\boxed{\widehat{\boldsymbol{x}}_i = \boldsymbol{v} z_i = \boldsymbol{v}\boldsymbol{v}^\top \widetilde{\boldsymbol{x}}_i.}$$

The difference $\widetilde{\boldsymbol{x}}_i - \widehat{\boldsymbol{x}}_i$ is the part discarded by this one-dimensional representation.

Projection Calculation

Take a unit direction $\boldsymbol{v} = (1,1)^\top/\sqrt{2}$ and a centered point $\widetilde{\boldsymbol{x}} = (2,0)^\top$.
The retained score is

$$z = \boldsymbol{v}^\top \widetilde{\boldsymbol{x}} = \frac{2+0}{\sqrt{2}} = \sqrt{2}.$$

Moving z units along $\boldsymbol{v}$ gives the reconstruction:

$$\boxed{\widehat{\boldsymbol{x}} = z\boldsymbol{v} = (1,1)^\top.}$$

We have replaced two original coordinates with one number, but the reconstructed point is no longer exactly $(2,0)^\top$.

Projection Residual

The information missing from the previous reconstruction is

$$\boldsymbol{e} = \widetilde{\boldsymbol{x}} - \widehat{\boldsymbol{x}} = (2,0)^\top - (1,1)^\top = (1,-1)^\top.$$

Its squared Euclidean reconstruction error is

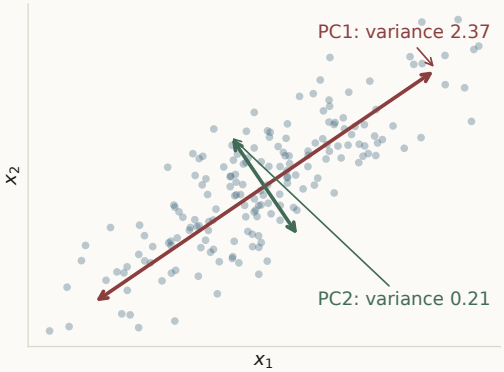
$$\boxed{\|\boldsymbol{e}\|_2^2 = 1^2 + (-1)^2 = 2.}$$

A different line would produce a different residual.
PCA chooses a projection that performs well across *all* training observations, not only this single point.

Principal Components

DSC5001 ▪ Semester A, 2026/27

Principal Directions



PC1 follows the direction in which the observations vary most.
PC2 records the remaining variation along a perpendicular direction.
We now turn this geometric choice into an optimization problem.

Unit Directions

If we multiplied a direction vector by 10, every projected score would become ten times larger, although the projection *axis* would be unchanged. We need a common scale for comparison.

$$\boxed{\boldsymbol{v}^\top \boldsymbol{v} = \|\boldsymbol{v}\|_2^2 = 1.}$$

PCA searches only among unit-length directions.
The vectors $\boldsymbol{v}$ and $-\boldsymbol{v}$ describe the same axis, so the sign of a principal component is arbitrary.

Covariance Matrix

Instead of evaluating every possible line by projecting all points, we summarize how the centered features vary together.
With n training observations, define

$$\boxed{\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^{n} \widetilde{\boldsymbol{x}}_i \widetilde{\boldsymbol{x}}_i^{\top} = \frac{1}{n}\widetilde{\boldsymbol{X}}^{\top}\widetilde{\boldsymbol{X}}.}$$

$\boldsymbol{S}$ is the $p \times p$ covariance matrix under the $1/n$ convention.
Its diagonal measures feature variances; its other entries measure covariation.

Projected Variance

For a unit direction $\boldsymbol{v}$, a centered observation has score $z_i = \boldsymbol{v}^\top \widetilde{\boldsymbol{x}}_i$.
The mean squared score is its empirical variance:

$$\frac{1}{n}\sum_i z_i^2 = \frac{1}{n}\sum_i \boldsymbol{v}^\top \widetilde{\boldsymbol{x}}_i \widetilde{\boldsymbol{x}}_i^\top \boldsymbol{v}$$
$$= \boxed{\boldsymbol{v}^\top \boldsymbol{S} \boldsymbol{v}}.$$

This single quadratic expression measures how much variation would survive projection onto $\boldsymbol{v}$.

Variance Maximization

To keep the most variation in one coordinate, maximize projected variance over all unit directions:

$$\max_{\boldsymbol{v}} \ \boldsymbol{v}^\top \boldsymbol{S}\boldsymbol{v} \quad \text{subject to} \quad \boldsymbol{v}^\top \boldsymbol{v} = 1.$$

Without the constraint, simply increasing $\|\boldsymbol{v}\|$ would increase the objective for the wrong reason.
The constrained solution provides the first principal direction.

Eigenvector Equation

Use a Lagrange multiplier λ to impose unit length:

$$\mathcal{L}(\boldsymbol{v}, \lambda) = \boldsymbol{v}^\top \boldsymbol{S}\boldsymbol{v} - \lambda(\boldsymbol{v}^\top \boldsymbol{v} - 1).$$

Setting the derivative with respect to $\boldsymbol{v}$ to zero,

$$2\boldsymbol{S}\boldsymbol{v} - 2\lambda\boldsymbol{v} = \boldsymbol{0},$$

$$\boxed{\boldsymbol{S}\boldsymbol{v} = \lambda\boldsymbol{v}.}$$

Thus the candidate principal directions are eigenvectors of the covariance matrix; λ is the corresponding eigenvalue.

Largest Eigenvalue

If $\boldsymbol{S}\boldsymbol{v} = \lambda \boldsymbol{v}$ and $\|\boldsymbol{v}\|_2 = 1$, then

$$\boldsymbol{v}^\top \boldsymbol{S}\boldsymbol{v} = \lambda \boldsymbol{v}^\top \boldsymbol{v} = \boxed{\lambda}.$$

The eigenvalue equals the variance along that eigenvector.
Therefore **PC1 uses the largest eigenvalue**, not just any solution of the eigenvector equation.
The later principal components use the remaining ordered eigenvectors and are orthogonal to earlier ones.

Covariance Example

We can now compute a PCA direction instead of choosing it visually.
Consider two centered features with covariance

$$\boxed{\boldsymbol{S} = \begin{pmatrix} 5 & 2 \\ 2 & 5 \end{pmatrix}.}$$

Both features have variance 5, and their positive covariance 2 means they tend to move together.
We expect a principal direction near $(1, 1)^\top$; the eigenvalue calculation will confirm it.

Eigenvalue Calculation

Find the eigenvalues by solving $\det(\boldsymbol{S} - \lambda \boldsymbol{I}) = 0$, where $\boldsymbol{I}$ is the identity matrix:

$$(5 - \lambda)^2 - 2^2 = 0,$$
$$(3 - \lambda)(7 - \lambda) = 0.$$

Hence

$$\boxed{\lambda_1 = 7, \qquad \lambda_2 = 3.}$$

The leading axis will retain variance 7, while the perpendicular axis retains variance 3.

Eigenvector Calculation

For $\lambda_1 = 7$, the equation $(\boldsymbol{S} - 7\boldsymbol{I})\boldsymbol{v} = 0$ gives $v_1 = v_2$.
For $\lambda_2 = 3$, it gives $v_1 = -v_2$.
Normalizing both directions yields

$$\boxed{\boldsymbol{v}_1 = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}, \qquad \boldsymbol{v}_2 = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}.}$$

PC1 measures the features moving together; PC2 measures their difference. Both are unit length and orthogonal.

Principal Scores

The principal directions are shared across observations; each observation receives its own coordinates along them.
For centered point $\widetilde{\boldsymbol{x}}_* = (2, 0)^\top$,

$$z_{*,1} = \boldsymbol{v}_1^\top \widetilde{\boldsymbol{x}}_* = \boxed{\sqrt{2}},$$
$$z_{*,2} = \boldsymbol{v}_2^\top \widetilde{\boldsymbol{x}}_* = \boxed{\sqrt{2}}.$$

If we retain only PC1, we store the first score.
The second score describes variation lost in that one-component representation.

Orthogonal Components

The two directions are perpendicular:

$$\boxed{\boldsymbol{v}_1^\top \boldsymbol{v}_2 = \tfrac{1}{2}(1-1) = 0.}$$

For centered training data, their score covariance is

$$\boldsymbol{v}_1^\top \boldsymbol{S} \boldsymbol{v}_2 = \lambda_2 \boldsymbol{v}_1^\top \boldsymbol{v}_2 = 0.$$

Thus the component scores are uncorrelated in the sample.
This avoids duplicate *linear* information, but uncorrelated does not necessarily mean independent.

Variance Partition

The total variance equals the sum of the covariance eigenvalues:

$$\operatorname{tr}(\boldsymbol{S}) = 5 + 5 = 7 + 3 = 10.$$

Keeping only the first component retains

$$\boxed{\frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{7}{10} = 70\%.}$$

The other 30% of the sample variance lies along PC2.
We can now decide whether this saving in dimensionality is worth the information discarded.

Reconstruction and SVD

DSC5001 ▪ Semester A, 2026/27

PCA Reconstruction

Principal scores store fewer coordinates, but can we recover the original measurements from them?
Collect the first r principal directions as $\boldsymbol{V}_r = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r]$.

$$\boxed{\boldsymbol{z} = \boldsymbol{V}_r^\top \widetilde{\boldsymbol{x}}, \qquad \widehat{\boldsymbol{x}} = \boldsymbol{V}_r \boldsymbol{z} = \boldsymbol{V}_r \boldsymbol{V}_r^\top \widetilde{\boldsymbol{x}}.}$$

Here $\widehat{\boldsymbol{x}}$ reconstructs the *centered* observation.
Add the original training mean to recover the original feature units. When $r < p$, some detail is generally lost.

Reconstruction Example

Return to the covariance example with $\boldsymbol{v}_1 = (1,1)^\top/\sqrt{2}$ and $\widetilde{\boldsymbol{x}}_* = (2,0)^\top$.
Keeping only PC1 gives score $z_{*,1} = \sqrt{2}$ and

$$\begin{aligned}\widehat{\boldsymbol{x}}_* &= \boldsymbol{v}_1 z_{*,1} \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}\sqrt{2} \\ &= \boxed{\begin{pmatrix}1\\1\end{pmatrix}}.\end{aligned}$$

The reconstruction must lie on the PC1 axis.
It differs from $(2,0)^\top$ because PC2 was discarded.

Reconstruction Error

The difference between a centered observation and its reconstruction is the residual:

$$\boldsymbol{e} = \widetilde{\boldsymbol{x}}_* - \widehat{\boldsymbol{x}}_* = \begin{pmatrix} 2 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

For this observation, squared reconstruction error is

$$\boxed{\|\boldsymbol{e}\|_2^2 = 1^2 + (-1)^2 = 2.}$$

Keeping both orthonormal components would reconstruct it exactly.

Orthogonal Decomposition

The retained projection and its discarded residual are perpendicular.
By the Pythagorean theorem,

$$\boxed{\|\widetilde{\boldsymbol{x}}_i\|_2^2 = \|\boldsymbol{V}_r^\top \widetilde{\boldsymbol{x}}_i\|_2^2 + \|\widetilde{\boldsymbol{x}}_i - \widehat{\boldsymbol{x}}_i\|_2^2.}$$

Total centered energy is fixed for the dataset.
Therefore, increasing retained projected energy decreases total squared reconstruction error by the same amount.
This explains the connection between two PCA objectives.

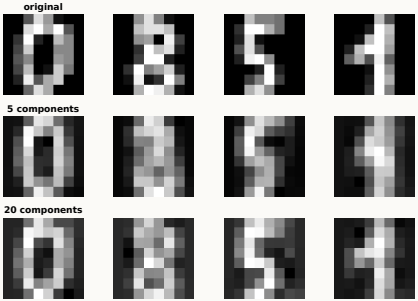
Optimal Subspace

Among all r-dimensional orthonormal linear subspaces, PCA minimizes total squared reconstruction error:

$$\min_{\boldsymbol{V}_r^\top \boldsymbol{V}_r = \boldsymbol{I}_r} \sum_{i=1}^{n} \|\widetilde{\boldsymbol{x}}_i - \boldsymbol{V}_r \boldsymbol{V}_r^\top \widetilde{\boldsymbol{x}}_i\|_2^2.$$

The best $\boldsymbol{V}_r$ contains the directions associated with the r largest covariance eigenvalues. Thus **maximizing variance and minimizing squared reconstruction error give the same PCA subspace**.

Digit Reconstruction



Compare the same digit across rows. Five components preserve its basic shape; twenty recover finer detail.
More PCs improve reconstruction but require more stored coordinates.

Explained Variance

Retaining every component preserves all variance but provides no compression.
To assess the amount kept by the first r, calculate the cumulative **explained-variance ratio**:

$$\boxed{\mathrm{EVR}(r) = \frac{\lambda_1 + \cdots + \lambda_r}{\lambda_1 + \cdots + \lambda_p}.}$$

The eigenvalues λ_j are sorted from largest to smallest.
EVR measures variation retained, *not* predictive accuracy or the importance of features for a later task.

Variance Ratio Example

For a separate four-feature dataset, suppose the eigenvalues are $(7, 3, 1, 1)$, totaling 12.

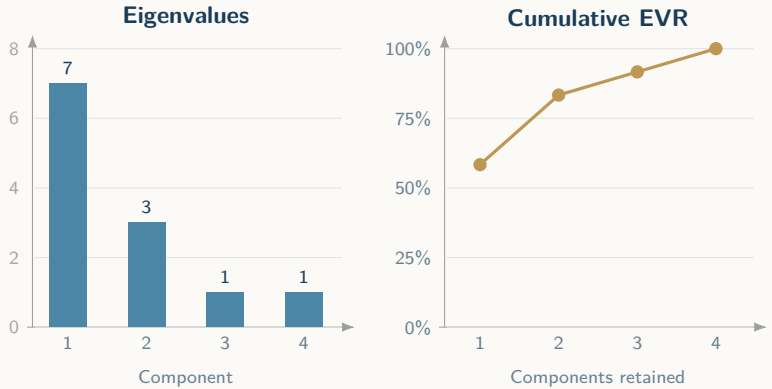
Retained PCs r	1	2	3
$\mathrm{EVR}(r)$	$7/12$	$10/12$	$11/12$

If the aim is to preserve at least 80% of the variance,

$$\boxed{\mathrm{EVR}(2) = \frac{10}{12} = 83.3\%.}$$

Two components are sufficient for this variance criterion.
A scree plot shows how much each additional component contributes.

Scree Plot



For eigenvalues $(7, 3, 1, 1)$, two components retain $10/12 = 83.3\%$ of the variance.
The bars show the additional variance from each component; the cumulative curve shows the total retained.

Component Selection

A scree plot makes it easier to see where additional components begin to contribute little extra variance.

Yet selecting r requires knowing how the representation will be used.

- For compression, compare reconstruction error with storage saved.
- For prediction, compare performance on validation data.
- For clustering, check whether relevant group structure survives.

A high explained-variance ratio alone cannot guarantee useful downstream results.

Loadings and Scores

The PCA directions describe *features*; the projected coordinates describe *observations*.
It helps to give these different names:

$$\boxed{\text{loadings} = \boldsymbol{V}_r, \qquad \text{scores} = \widetilde{\boldsymbol{X}}\boldsymbol{V}_r.}$$

The entries of each direction are its **loadings**.
For PC1 in our two-feature example, both loadings equal $1/\sqrt{2}$: the features contribute equally with the same sign.
Interpret component loadings using the meanings and units of the features.

SVD Motivation

So far we have computed PCA from the $p \times p$ covariance matrix.
For large datasets, explicitly forming $\widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{X}}$ can be unnecessary and may make numerical calculations less stable.

SVD works directly with the centered data matrix.

The singular value decomposition produces the same principal directions and gives a direct way to construct low-rank reconstructions.

Singular Value Decomposition

Let $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{n \times p}$ be the centered data matrix.
Its singular value decomposition is

$$\boxed{\widetilde{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top.}$$

Columns of $\boldsymbol{V}$ are feature directions (**loadings**); $\boldsymbol{U}\boldsymbol{D}$ contains training scores, and the diagonal entries $d_j \geq 0$ of $\boldsymbol{D}$ are **singular values**.
Because $\boldsymbol{S} = \widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{X}}/n$,

$$\boxed{\lambda_j = d_j^2/n.}$$

Singular-Value Calculation

Suppose $n = 8$ centered observations have covariance eigenvalues $\lambda_1 = 7$ and $\lambda_2 = 3$, using the $1/n$ convention.
The corresponding singular values are

$$d_1 = \sqrt{n\lambda_1} = \sqrt{56} = \boxed{2\sqrt{14}},$$
$$d_2 = \sqrt{n\lambda_2} = \sqrt{24} = \boxed{2\sqrt{6}}.$$

The larger singular value indicates greater variation along its associated principal direction.
Singular values themselves are not covariance eigenvalues: they must be squared and divided by n.

Truncated SVD

If only r components are needed, keep the first r singular values and their left and right singular vectors:

$$\boxed{\widetilde{\boldsymbol{X}}_r = \boldsymbol{U}_r \boldsymbol{D}_r \boldsymbol{V}_r^\top.}$$

This reconstructs the centered data using a matrix of rank at most r.
When r is small, the factorization can be stored more compactly than the original $n \times p$ matrix.
The next question is how much reconstruction error truncation creates.

Discarded Variation

Truncation discards singular directions numbered $r+1, r+2, \ldots$.
The squared reconstruction error equals the energy of those directions:

$$\boxed{\|\widetilde{\boldsymbol{X}} - \widetilde{\boldsymbol{X}}_r\|_F^2 = \sum_{j>r} d_j^2 = n \sum_{j>r} \lambda_j.}$$

Here $\|\cdot\|_F^2$ is the sum of squared matrix entries.
For our two-feature covariance with eigenvalues $(7, 3)$, keeping PC1 discards total squared training error $3n$.

Best Rank Approximation

Could another rank-r matrix reconstruct the same centered data with a smaller sum of squared entrywise errors?
For this measure, the truncated SVD is optimal:

$$\widetilde{\boldsymbol{X}}_r \in \underset{\operatorname{rank}(\boldsymbol{A})\leq r}{\arg\min} \|\widetilde{\boldsymbol{X}} - \boldsymbol{A}\|_F.$$

This is what **best rank-r approximation** means.
It does not claim that PCA is best for classification, clustering, or every other downstream objective.

Compression Trade-Off

Keeping fewer PCs reduces storage and can remove noise, but it increases reconstruction error.
Keeping more PCs preserves detail while reducing the savings.

Choose the retained dimension r for the intended task.

PCA tells us how to represent each observation with fewer coordinates.
Our next task is different: **discovering groups** when no class labels are provided.

K-means Clustering

DSC5001 ▪ Semester A, 2026/27

Clustering Motivation

Suppose customers or sensor records have measurements but no known category labels. We may still want to find cases with similar behavior.

$$\boxed{\{\boldsymbol{x}_i\}_{i=1}^n \longrightarrow \{c_i\}_{i=1}^n, \qquad c_i \in \{1, \ldots, K\}.}$$

The label c_i is a group assigned by an algorithm, and K is the chosen number of groups.
Unlike supervised classification, we have no true labels to check directly.
Therefore group meaning must be assessed separately.

Distance Choice

A grouping method must first define what *similar* means.
K-means uses squared Euclidean distance between two p-feature vectors:

$$d(\boldsymbol{x}, \boldsymbol{z})^2 = \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 = \sum_{j=1}^{p} (x_j - z_j)^2.$$

Large numerical feature scales can dominate this sum.
Before interpreting clusters, choose feature units and transformations that reflect the comparisons you actually want to make.

Distance Example

Compare two observations, $\boldsymbol{x} = (2, 100)^\top$ and $\boldsymbol{z} = (4, 110)^\top$.
On the original scale,

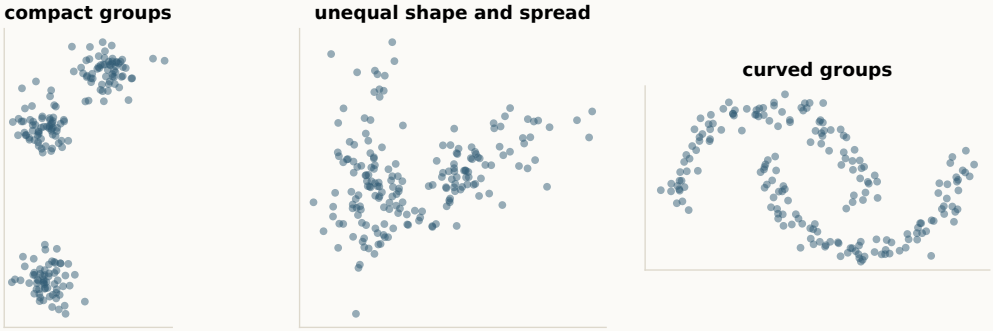
$$\boxed{d(\boldsymbol{x}, \boldsymbol{z})^2 = 2^2 + 10^2 = 104.}$$

If the second feature is expressed in tens, the same records become $(2, 10)^\top$ and $(4, 11)^\top$:

$$\boxed{d(\boldsymbol{x}', \boldsymbol{z}')^2 = 2^2 + 1^2 = 5.}$$

The underlying cases have not changed. Their distance has, so scaling can change their cluster assignments.

Cluster Shapes



K-means represents each group by a single centroid and favors compact groups under Euclidean distance.
Curved or unequal-density patterns may require a different clustering method.

Assignments and Centers

K-means describes each group by a **centroid**, the mean of its members.
Two quantities change during fitting:

- c_i: the current group assigned to observation i;
- $\boldsymbol{\mu}_k$: the centroid of group k.

First assign *all* observations using the current centers.
Then calculate each new center from the fixed memberships.
Keeping these two operations separate lets us see why the algorithm works.

K-means Objective

A useful centroid should lie close to the observations it represents.
K-means formalizes this using the within-cluster sum of squared distances:

$$J(\boldsymbol{c}, \boldsymbol{\mu}) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{\mu}_{c_i}\|_2^2.$$

Here $\boldsymbol{\mu}_{c_i}$ is the center assigned to observation i.
The number K is fixed before fitting; assignments and centroids are then updated alternately to lower J.

Assignment Step

Hold all centroids fixed. Each observation can lower its contribution to J by selecting its nearest center:

$$\boxed{c_i \leftarrow \underset{k=1,\dots,K}{\arg\min} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_2^2.}$$

All observations are assigned against the *same current centers*.
A consistent rule resolves exact ties.
Only after assignments are complete should the centers be moved.

Centroid Update

With assignments fixed, choose a representative for each group that minimizes the sum of squared distances to its points:

$$J_k(\boldsymbol{\mu}_k) = \sum_{i:c_i=k} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_2^2.$$

The minimum occurs at the sample mean:

$$\boxed{\boldsymbol{\mu}_k \leftarrow \frac{1}{n_k} \sum_{i:c_i=k} \boldsymbol{x}_i,}$$

where n_k is the number of observations in cluster k.
This mean update is correct because K-means uses *squared* distance (assuming the cluster is nonempty).

One-Dimensional Data

We can follow both K-means steps completely on six one-dimensional observations:

$$\boxed{x = (0, 2, 3, 8, 9, 10), \qquad K = 2.}$$

Start from deliberately imperfect centers

$$\mu_1^{(0)} = 0, \qquad \mu_2^{(0)} = 4.$$

A superscript indicates the iteration; it is not an exponent.
For a point exactly halfway between centers, assign it to the lower-indexed cluster.

Initial Assignments

The midpoint between initial centers 0 and 4 is 2.
Using the specified tie rule at $x = 2$,

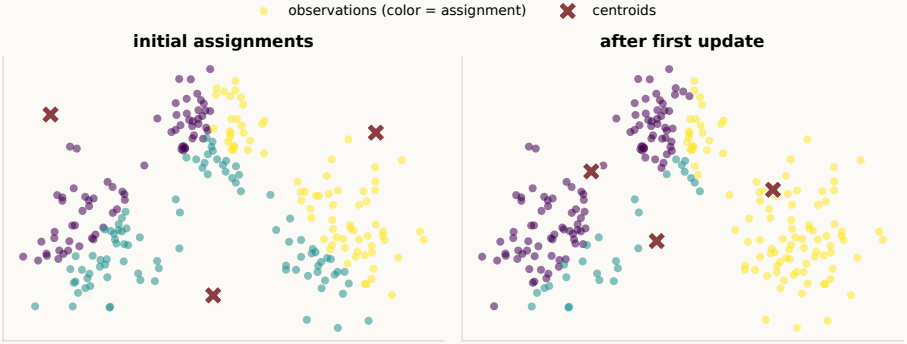
$$\boxed{C_1^{(0)} = \{0, 2\}, \qquad C_2^{(0)} = \{3, 8, 9, 10\}.}$$

The initial within-cluster sum of squares is

$$\boxed{J_0 = 0^2 + 2^2 + (-1)^2 + 4^2 + 5^2 + 6^2 = 82.}$$

The second center is much too far left for most of its points.
We therefore expect the centroid update to improve the objective.

First Centroid Update



The crosses move from $0, 4$ to the means $1, 7.5$.
All memberships remain fixed during this update; the point at $x = 3$ can switch only when assignment is repeated.

Centroid Calculation

Average the values within each *current* group:

$$\mu_1^{(1)} = \frac{0+2}{2} = \boxed{1},$$
$$\mu_2^{(1)} = \frac{3+8+9+10}{4} = \boxed{7.5}.$$

With memberships still fixed, the squared errors are 2 for group 1 and 29 for group 2:

$$\boxed{J = 2 + 29 = 31 < 82.}$$

Centers are improved. Next we reconsider which center is nearest to each point.

Reassignment

For the new centers 1 and 7.5, the midpoint is 4.25.
The point $x = 3$ is now closer to center 1, so

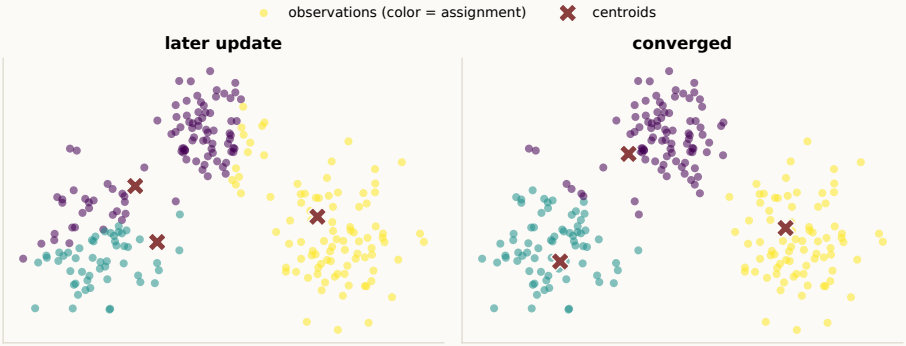
$$\boxed{C_1^{(1)} = \{0, 2, 3\}, \qquad C_2^{(1)} = \{8, 9, 10\}.}$$

Keep the centers fixed during this reassignment:

$$\boxed{J = 6 + \frac{35}{4} = \frac{59}{4} = 14.75.}$$

This reduction comes only from changing memberships; we have not recalculated the means yet.

Stable Centroids



After $x = 3$ joins the first group, the centroids move to $5/3$ and 9.
The new dividing midpoint lies between 3 and 8, so repeating the assignment step changes no memberships.

Second Centroid Update

The reassigned groups have three observations each.
Their new means are

$$\boxed{\mu_1^{(2)} = \frac{0+2+3}{3} = \frac{5}{3}, \qquad \mu_2^{(2)} = \frac{8+9+10}{3} = 9.}$$

The group squared-error contributions are $14/3$ and 2:

$$\boxed{J_2 = \frac{14}{3} + 2 = \frac{20}{3}.}$$

Their midpoint $16/3 \approx 5.33$ lies between the two groups.
No point changes cluster on the next assignment step.

Objective Decrease

The same six observations show why assignment and mean updates work together:

Stage	Within-cluster RSS
Initial centers and assignments	82
After first centroid update	31
After reassignment	$59/4 = 14.75$
After second centroid update	$20/3 \approx 6.67$

Each operation decreases or maintains the **same objective** J.
This explains the alternating nature of K-means.

Local Convergence

K-means repeats the nearest-center assignment and centroid update until memberships stabilize (or improvement becomes negligible).

$$\boxed{J_{t+1} \leq J_t.}$$

The objective cannot increase at either step.
However, the final partition can depend on the initial centers; a stable solution need not be the global minimum.
This is why practical implementations try more than one initialization.

Two-Dimensional Data

The same operations apply when each observation has several features.
Consider four two-feature points:

Case	A	B	C	D
$\boldsymbol{x}$	$(0,0)$	$(0,2)$	$(4,0)$	$(4,2)$

Start with $\boldsymbol{\mu}_1^{(0)} = (0,0)^\top$ and $\boldsymbol{\mu}_2^{(0)} = (4,2)^\top$.
Nearest-center assignment gives

$$\boxed{C_1 = \{A, B\}, \qquad C_2 = \{C, D\}.}$$

Two-Dimensional Update

Average both coordinates separately within the two groups:

$$\boxed{\boldsymbol{\mu}_1 = (0, 1)^\top, \qquad \boldsymbol{\mu}_2 = (4, 1)^\top.}$$

Before the update, B and C are two units away from their assigned centers, giving total squared error 8.
After the update, all four points are one unit from their centers:

$$\boxed{J_{\text{before}} = 8, \qquad J_{\text{after}} = 4.}$$

The labels need not change for a centroid update to improve the fit.

Initialization

The K-means objective decreases during fitting, but it can have multiple locally optimal partitions.
Different starting centers may therefore lead to different final groups.

Run K-means from several starting configurations.

A common strategy keeps the run with the smallest final within-cluster RSS.
K-means++ is a practical initialization method that spreads the initial centers apart, reducing the dependence on arbitrary starting locations.

Number of Clusters

Choosing more groups makes it easier to fit the training points.
Indeed, the best achievable within-cluster RSS cannot increase with K; when $K = n$, one center per observation gives zero error.

Training RSS alone cannot select K.

To decide whether extra groups are useful, examine separation, stability, domain interpretation, and the purpose of clustering.
A silhouette score provides one internal comparison of separation.

Silhouette Score

For observation i, let $a(i)$ be its average distance to points in its own cluster and $b(i)$ the smallest average distance to another cluster.
The silhouette score is

$$\boxed{s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}.}$$

For $a(i) = 2$ and $b(i) = 5$,

$$\boxed{s(i) = \frac{5 - 2}{5} = 0.6.}$$

A larger positive score suggests better distance-based separation.
It does not, on its own, establish a scientifically meaningful group.

K-means Limitations

The cluster-shape figure explains why K-means is not suitable for every dataset.

- Different feature scales change Euclidean distances.
- Extreme observations can pull cluster means away from most members.
- Curved or unequal-density groups may not fit centroid partitions.

These concerns become especially important if clustering is performed on PCA scores, because dimension reduction changes the distances seen by the clustering algorithm.

PCA and Clustering

DSC5001 ▪ Semester A, 2026/27

Different Objectives

PCA and K-means can be used together, but they optimize different quantities:

PCA	Retain variation and minimize reconstruction error
K-means	Reduce squared distances within assigned groups

A direction of high overall variance need not separate the groups that matter for an application. We therefore need to decide whether PCA is being used to *compress features before clustering* or merely to *display clusters found elsewhere*.

PCA and Clustering Workflow

There are two reasonable ways to combine these methods.

- **Cluster in PCA space.** Center or scale the features, retain r principal components, and fit K-means to the resulting scores. Only information in the retained components can influence the groups.
- **Use PCA for display.** Fit clusters in the original feature space, then plot their labels on the first two PCs. The plot is a view of the fitted result, not the space used for fitting.

In either case, document the chosen features, scaling, r, and K.
Then check whether the groups remain stable and are interpretable.

Information Loss Example

A short example shows why explained variance alone cannot justify a clustering representation. Consider centered points

$$(-4,-1), \quad (-4,1), \quad (4,-1), \quad (4,1).$$

Their $1/n$ covariance is

$$\boxed{\boldsymbol{S} = \begin{pmatrix} 16 & 0 \\ 0 & 1 \end{pmatrix}.}$$

PC1 retains $16/17 \approx 94.1\%$ of the variance, but completely removes the second coordinate. If the groups of interest differ by $x_2 = -1$ versus $x_2 = 1$, retaining only PC1 destroys that distinction.

Clustering in PC Space

When PCA is used for compression before clustering, form the score matrix

$$\boxed{\boldsymbol{Z} = \widetilde{\boldsymbol{X}}\boldsymbol{V}_r \in \mathbb{R}^{n \times r}.}$$

The i-th row contains the r retained scores of observation i.
K-means can then use these rows in place of the original feature vectors.
This may reduce computation and remove some noise, but important low-variance group differences could disappear.
Check whether the reduced space serves the clustering goal.

Selection and Validation

An unsupervised result depends on decisions made before and during fitting: feature selection, scaling, retained dimension r, number of clusters K, and initialization.

- Fit preprocessing and PCA using the designated training data.
- Compare partitions under sensible choices of r and K.
- Examine stability across resamples and different initial centers.

A clear two-dimensional plot can help explain a result.
It cannot replace evidence that the groups are useful or reproducible.

Cluster Interpretation

After obtaining clusters, return to the original measurements.
Describe how each group differs in the quantities that matter for the scientific or practical question.

Cluster labels depend on the representation and distance.

A useful conclusion should survive reasonable changes in preprocessing and initialization, or explain why it does not.
PCA provides a representation; K-means proposes groups; interpretation and stability checks assess their usefulness.

Summary

- **PCA** retains directions with large sample variance.
- **Reconstruction and SVD** quantify what is kept and lost.
- **K-means** alternates nearest-center assignments and mean updates.
- **Evaluation** checks whether chosen representations and groups are useful.

The complete analysis requires a clear representation, a defined objective, and evidence supporting the interpretation.

Week 12: PCA, clustering, and unsupervised learning

Questions and discussion