Korea Aerospace University

2AI4203: Machine Learning

Lec07: Dimensionality Reduction & Clustering

Jiyoon Shin

KAU College of AI Convergence

Oct. 14, 2026

Korea Aerospace University

Recap:

Support Vector Machines & Boosting

Feature Scaling

- Why scaling matters
  - Large-scale features can dominate distance calculation
  - SVM, PCA, and K-Means are sensitive to feature scaling
- Standardization

$$x_j' = \frac{x_j - \mu_j}{\sigma_j}$$

- Min-Max Scaling

$$x_j' = \frac{x_j - x_j^{\min}}{x_j^{\max} - x_j^{\min}}$$

- Avoiding data leakage
  - Fit the scaler on training data only and apply it to validation/test data

Support Vector Machines

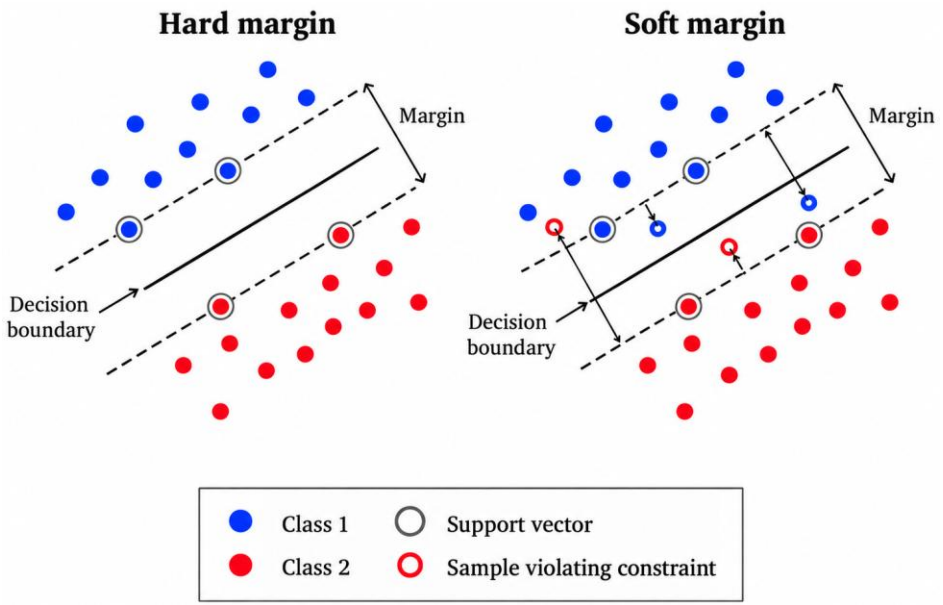
- Maximum margin: finds the decision boundary with the largest margin
- Support vectors: closest samples that determine the decision boundary

- Hard-Margin SVM
  - Requires linearly separable data
  - Maximizes the margin by minimizing

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2$$

subject to

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$



Support Vector Machines

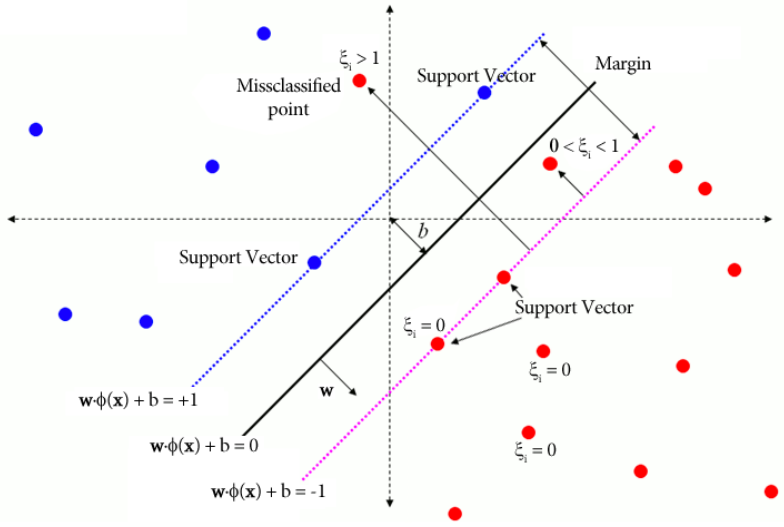
- Soft-Margin SVM
  - Allows margin violations for overlapping or noisy data
  - Balances margin size and violation penalties

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

subject to

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0$$

  - ξ_i: slack variable (margin violation)
  - C: penalty for violations
  - The role of C
    - Small C: more violations allowed, stronger regularization
    - Large C: violations penalized more, weaker regularization

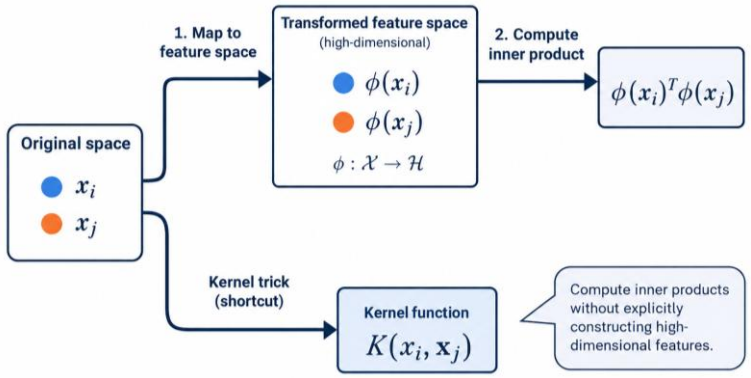


Kernel Methods

- Feature mapping

$$\mathbf{x} \longrightarrow \phi(\mathbf{x})$$

  - Maps data to a new feature space for linear separation



- Kernel Trick

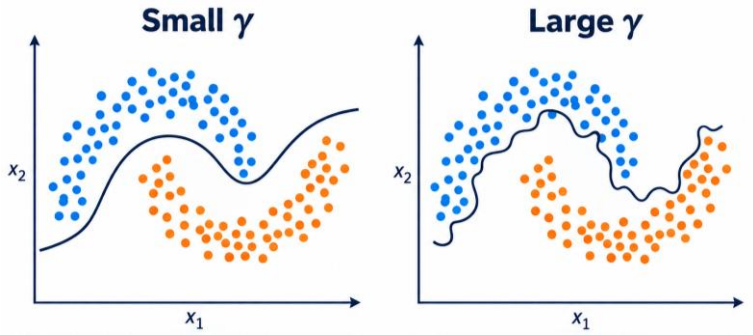
$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

  - Computes inner products without explicitly constructing transformed features

- RBF Kernel

$$K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \left|\left|\mathbf{x} - \mathbf{z}\right|\right|^2)$$

  - Small γ: broad influence, smoother boundary
  - Large γ: local influence, more complex boundary

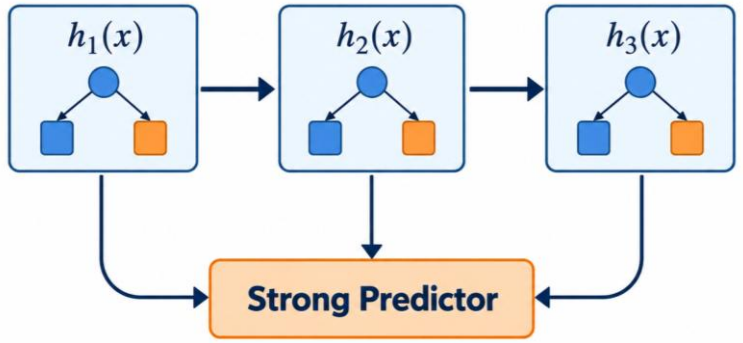


Boosting

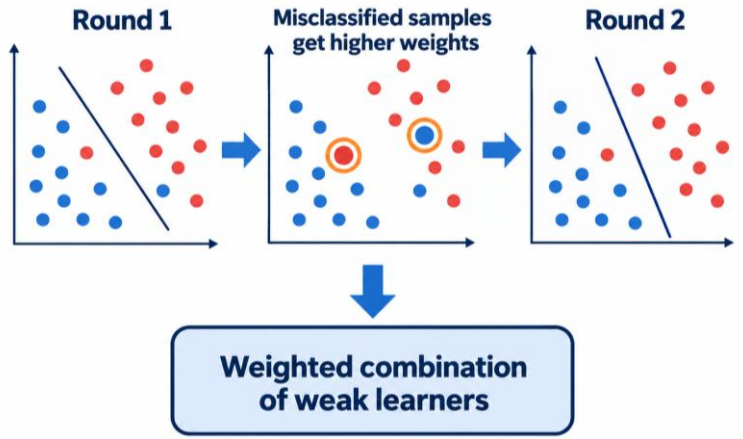
- Combines weak learners sequentially to correct errors

$$F(x) = \sum_{m=1}^{M} \alpha_m h_m(x)$$

  - h_m: weak learner
  - α_m: weight of learner m



- AdaBoost
  - Assigns higher weights to misclassified samples
  - Combines learners through weighted voting



Gradient Boosting

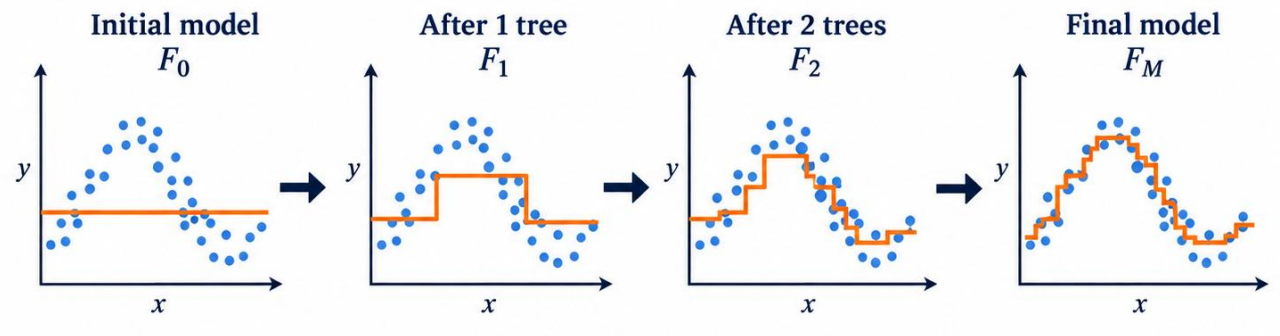
- Builds trees sequentially to correct previous errors
- For squared-error regression, learns residuals

$$r_i = y_i - F_{m-1}(\mathbf{x}_i)$$

- Additive learning

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \eta h_m(\mathbf{x})$$

  - F_{m-1}: current model
  - h_m: new tree
  - η: learning rate



- Methods
  - XGBoost: regularization, efficient tree construction
  - LightGBM: histogram-based splitting, leaf-wise growth

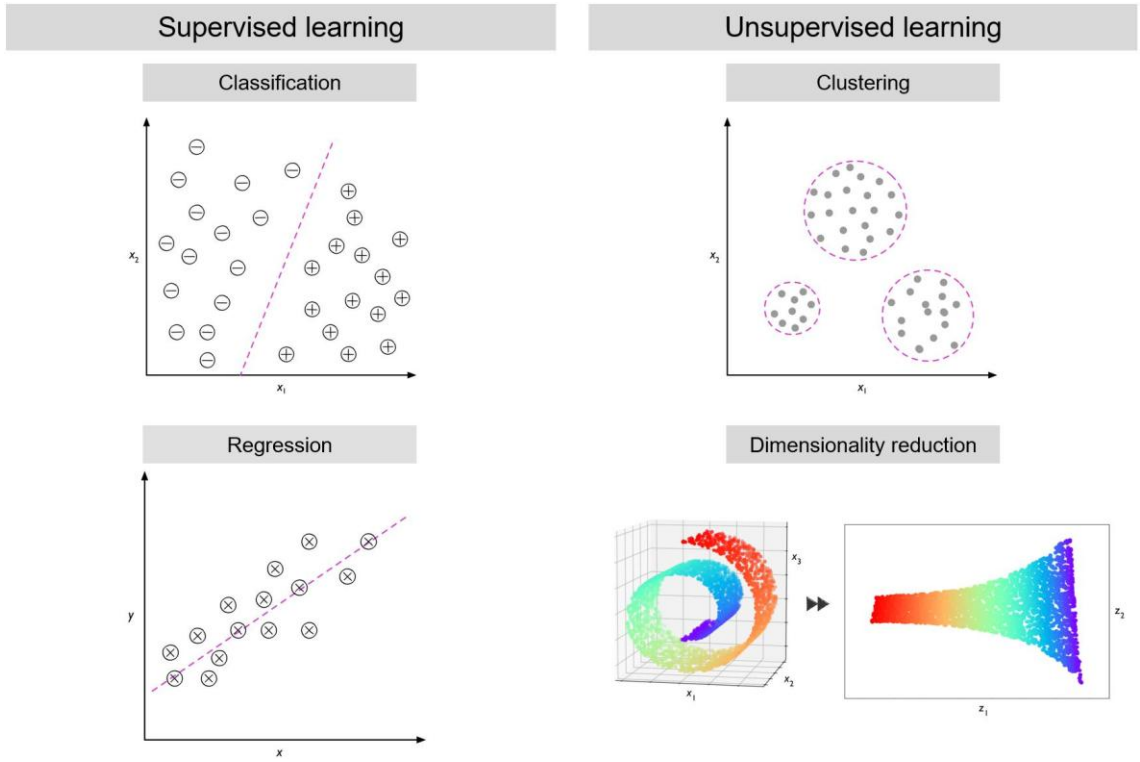
Korea Aerospace University

Part 1. Unsupervised Learning

Supervised vs. Unsupervised Learning

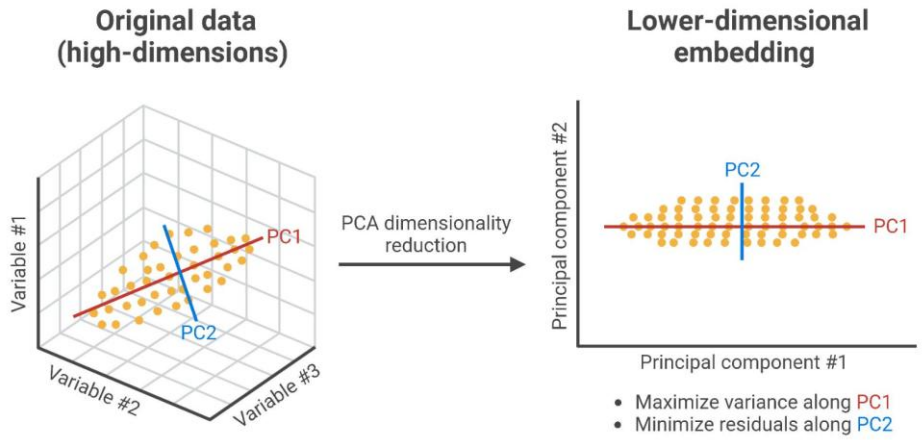
- Supervised Learning
  - Learns from labeled data
  - Predicts target values or labels
  - Examples: classification, regression

- Unsupervised Learning
  - Learns from unlabeled data
  - Discovers patterns or structures
  - Examples: dimensionality reduction, clustering



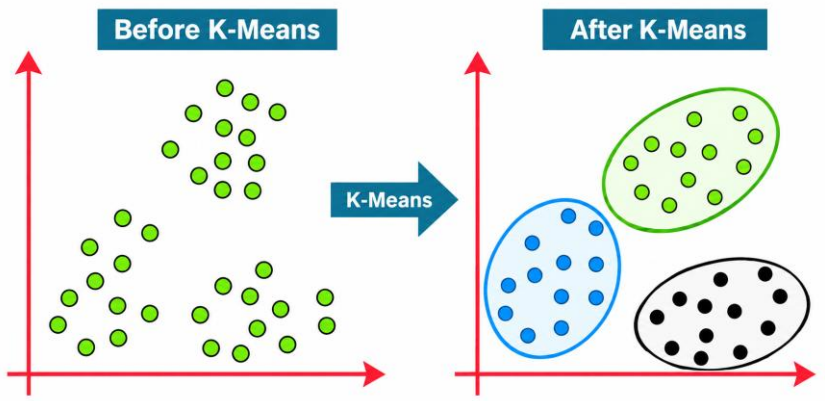
Dimensionality Reduction

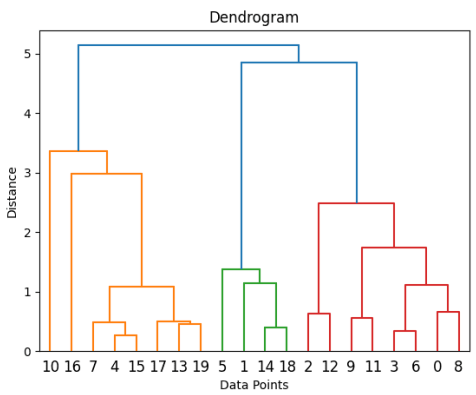
- Why reduce dimensions?
  - Real-world datasets often contain many features
  - High-dimensional data can be difficult to visualize and analyze
- Dimensionality Reduction
  - Transforms data into fewer dimensions
  - Aims to preserve important information or structure
- Example
  - Principal Component Analysis (PCA)



Clustering

- What is clustering?
  - Groups similar data points without labels
- Key idea
  - Within a cluster: data points are similar
  - Between clusters: clusters should be well separated
- Common methods
  - K-means clustering
  - Hierarchical clustering



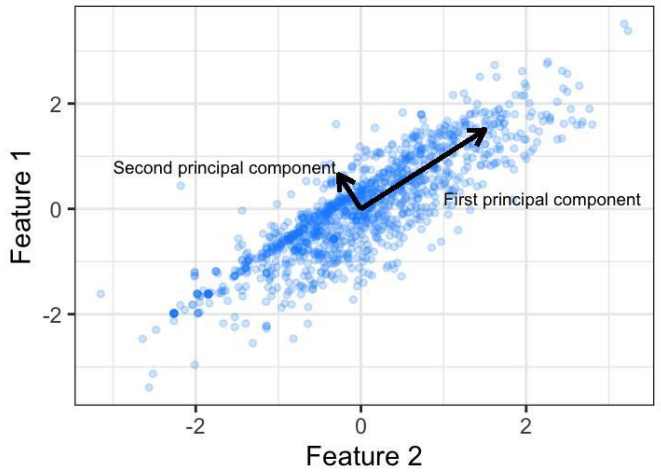


Korea Aerospace University

Part 2. Principal Component Analysis (PCA)

Principal Components

- Key idea of PCA
  - Finds new directions that capture maximum variance in data
  - Reduces dimensionality by retaining the leading components
- Principal Components
  - PC1: direction of maximum variance
  - PC2: direction of maximum remaining variance, perpendicular to PC1



Data Preprocessing

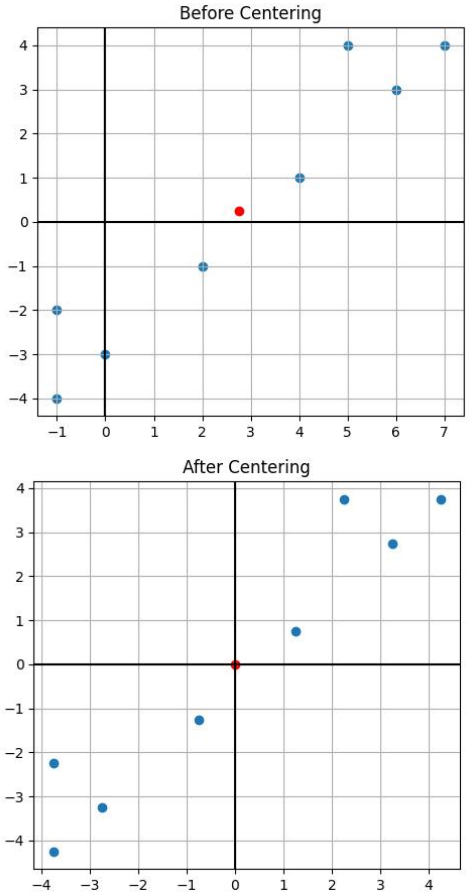
- Data matrix

$$X = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$$

- Mean centering
  - Subtract the mean of each feature to center the data around zero

$$X_c = X - \boldsymbol{\mu}\mathbf{1}^T$$

  - $\boldsymbol{\mu} \in \mathbb{R}^D$: feature-wise mean vector
  - $\mathbf{1} \in \mathbb{R}^N$: all-ones vector



Data Preprocessing

- (optional) Feature scaling

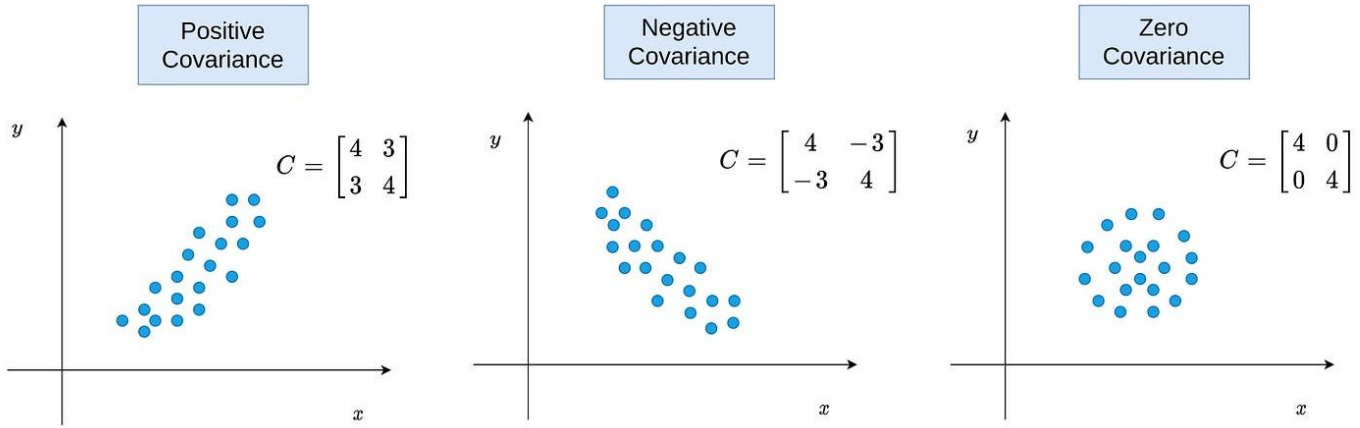
$$x'_{d,n} = \frac{x_{d,n} - \mu_d}{\sigma_d}$$

  - Standardization is useful when features have different scales
  - Prevents large-scale features from dominating PCA

Covariance Matrix

- Variance & Covariance
  - Variance: measures the spread of a feature
  - Covariance: measures how two features vary together
- Covariance Matrix

$$S = \frac{1}{N} X_c X_c^T, \qquad S = \begin{bmatrix} \mathrm{Var}(f_1) & \mathrm{Cov}(f_1, f_2) \\ \mathrm{Cov}(f_1, f_2) & \mathrm{Var}(f_2) \end{bmatrix}$$



Eigenvectors & Eigenvalues

- Finding principal directions
  - PCA finds directions that maximize projected variance

$$\mathrm{Var}(\mathbf{b}^T \mathbf{x}_c) = \mathbf{b}^T S\mathbf{b} \qquad (||\mathbf{b}|| = 1)$$

- Eigenvalue Decomposition

$$S\mathbf{b}_j = \lambda_j \mathbf{b}_j$$

  - Eigenvectors ($\mathbf{b}_j$): principal directions
  - Eigenvalues (λ_j): variances along those directions

Eigenvectors & Eigenvalues

$$\mathrm{Var}(\mathbf{b}_j^T \mathbf{x}_c) = \mathbf{b}_j^T S \mathbf{b}_j = \lambda_j$$

- Ordering principal components

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$$

  - Larger eigenvalues indicate greater variance
  - Keep the eigenvectors with the largest eigenvalues

Explained Variance

- Explained variance ratio

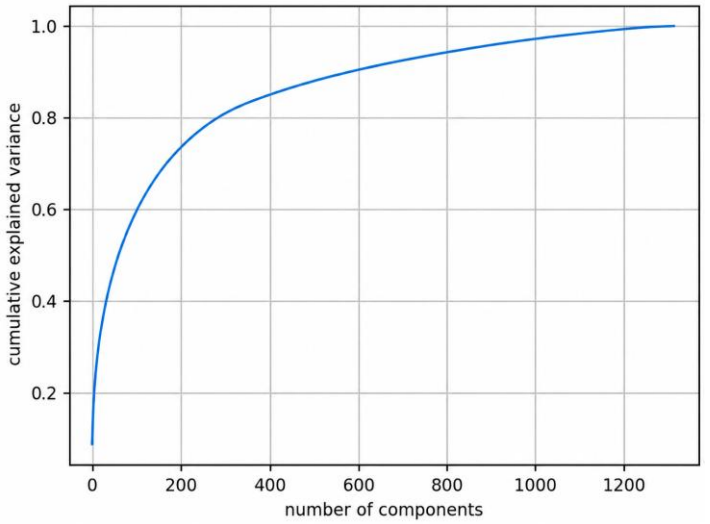
$$\mathrm{EVR}_j = \frac{\lambda_j}{\sum_{i=1}^{D} \lambda_i}$$

  - Proportion of total variance explained by each component

- Choosing the number of components

$$\text{Cumulative EVR}(M) = \frac{\sum_{j=1}^{M} \lambda_j}{\sum_{i=1}^{D} \lambda_i}$$

  - Select the smallest M that preserves sufficient variance
  - Common thresholds: 90% or 95%



Data Projection

- Projection Matrix

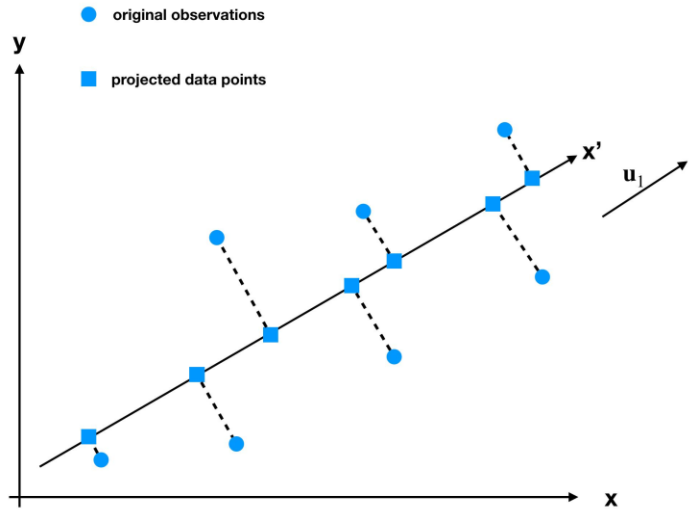
$$B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_M] \in \mathbb{R}^{D \times M}$$

  - Choose from the top M eigenvectors

- Projecting Data

$$Z = B^T X_c$$

  - Projects data onto the selected principal components
  - Reduces dimensionality from D to M



PCA Example

- Original Data

$$X = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}$$

- Step 1. Mean centering

$$X_c = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

- Step 2. Covariance Matrix

$$S = \frac{1}{3} X_c X_c^T = \frac{1}{3} \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

PCA Example

- Step 3. First principal component

$$\lambda_1 = \frac{4}{3}, \qquad \lambda_2 = 0$$

$$\mathbf{b}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, \qquad \mathbf{b}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$$

- Step 4. Projection

$$Z = \mathbf{b}_1^T X_c = [-\sqrt{2} \quad 0 \quad \sqrt{2}]$$

Strengths & Limitations

- Strengths
  - Reduces dimensionality
  - Produces uncorrelated components
  - Enables visualization
  - May reduce computational cost

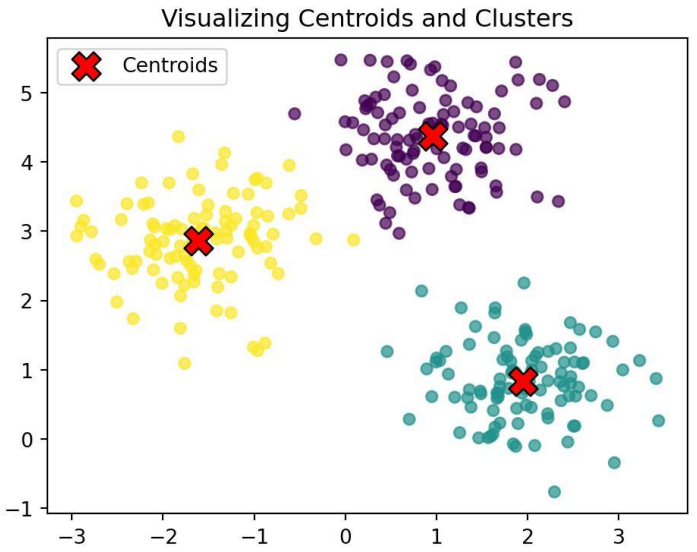
- Limitations
  - Uses linear projections
  - Sensitive to feature scaling
  - Components can be hard to interpret
  - High variance does not guarantee useful information

Korea Aerospace University

Part 3. K-Means Clustering

K-Means Clustering

- Key idea
  - Divides data into K clusters
  - Each cluster is represented by a centroid (mean of its points)
  - Assigns each point to the nearest centroid
- Initialization
  - Choose the number of clusters K
  - Initialize K centroids (e.g., randomly selected data points)

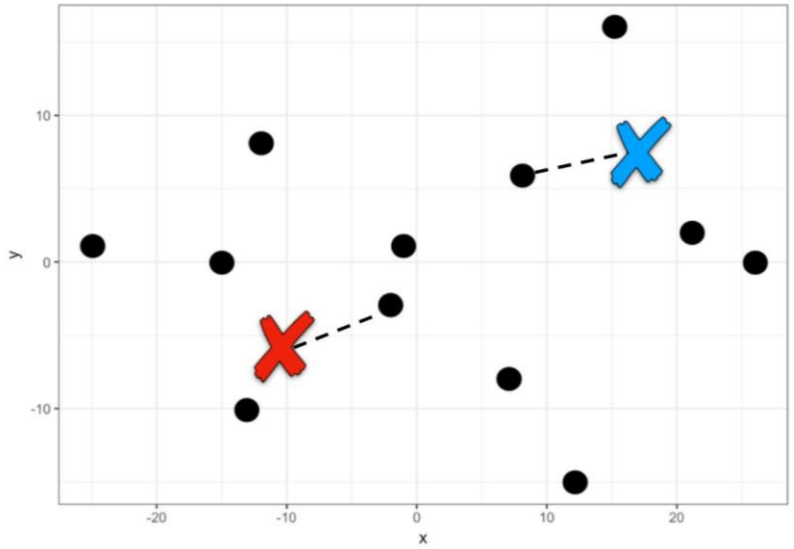


K-Means Objective

- Objective
  - Keep data points close to their assigned centroids
  - Minimize the within-cluster sum of squares (WCSS)

$$J = \sum_{k=1}^{K} \sum_{x_i \in C_k} ||x_i - \mu_k||_2^2$$

  - C_k: set of points in cluster k
  - μ_k: centroid of cluster k

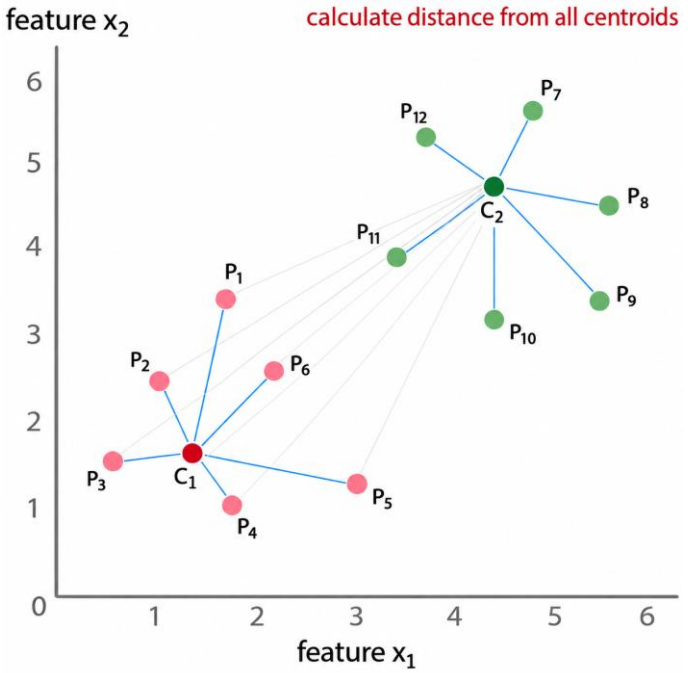


Cluster Assignment

- Step 1: Assign data points
  - Compute the distance from each point to every centroid
  - Assign each point to its nearest centroid

$$c_i = \arg\min_k ||x_i - \mu_k||_2^2$$

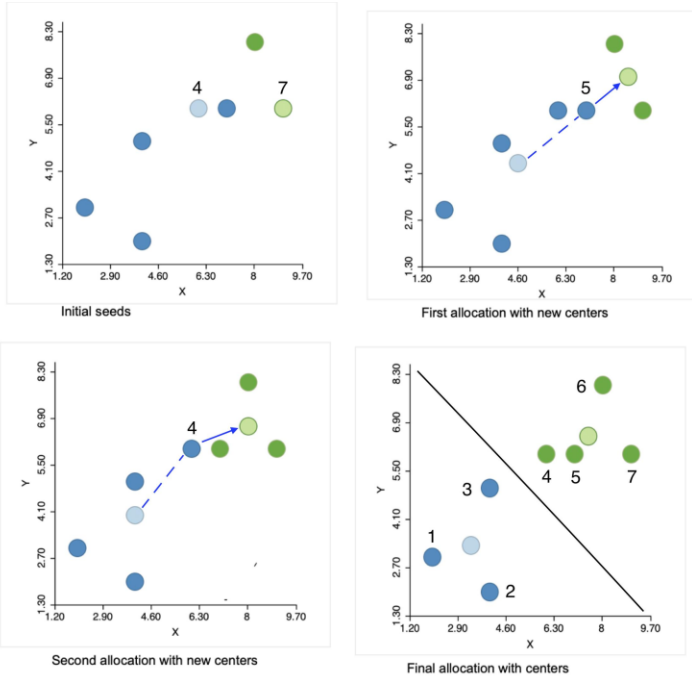
  - c_i: cluster assignment of point x_i



Centroid Update

- Step 2: Update centroids
  - Calculate the mean of all points assigned to each cluster
  - Move each centroid to the mean of its assigned points

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$



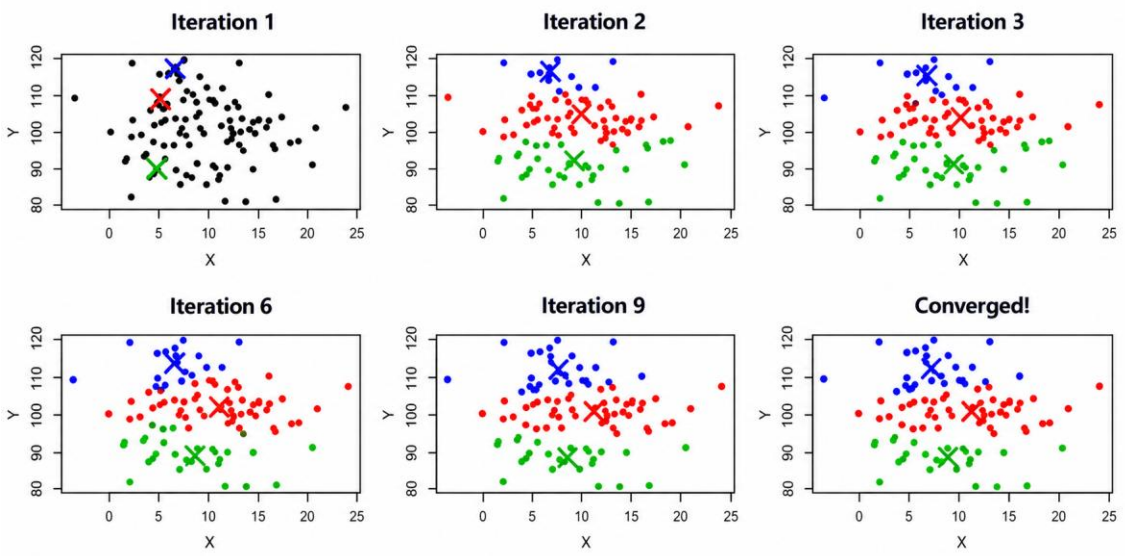
- Example

$$C_1 = \{2, 4, 6\} \implies \mu_1 = \frac{2 + 4 + 6}{3} = 4$$

Iteration & Convergence

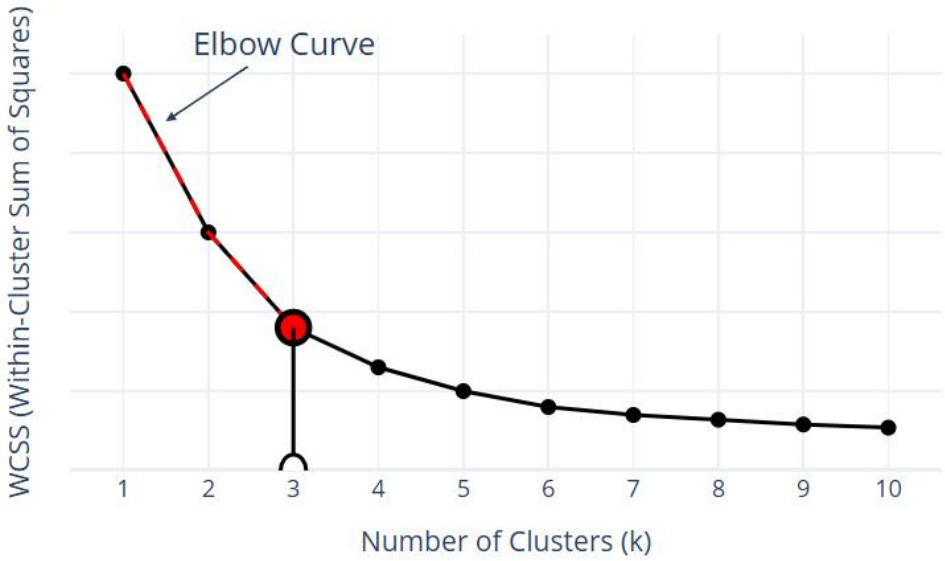
- Iteration
  - Repeat cluster assignment and centroid update
  - WCSS does not increase across iterations

- Convergence
  - Stop when cluster assignments stabilize
  - K-Means may converge to a local optimum



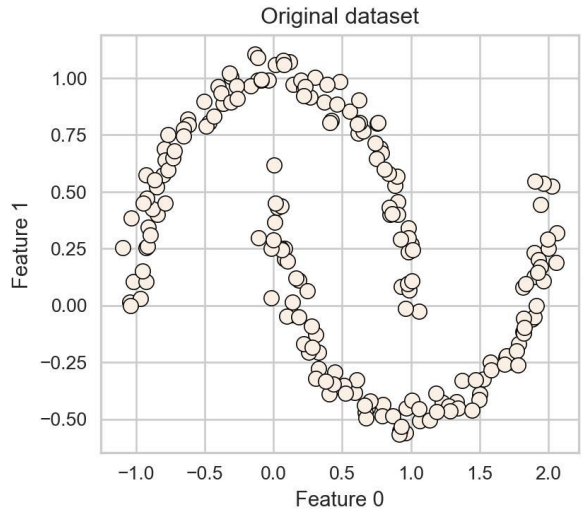
Elbow Method

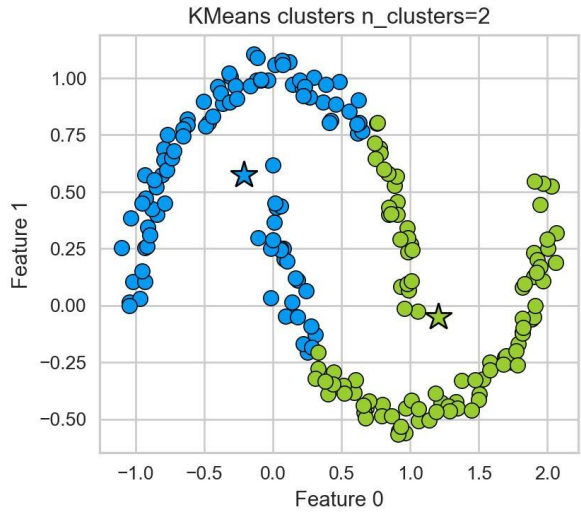
- Choosing K
  - WCSS generally decreases as K increases
- Elbow method
  - Run K-Means for different values of K
  - Plot WCSS against K
  - Find the elbow where improvement slows down
- Limitation
  - The elbow point is not always clear



Strengths & Limitations

- Strengths
  - Simple and computationally efficient
  - Scales well to large datasets
  - Works well for compact, roughly spherical clusters
- Limitations
  - Requires specifying K
  - Sensitive to initialization, outliers, and feature scaling
  - Struggles with complex cluster shapes
- Practical tips
  - Scale features when appropriate
  - Use K-Means++ and multiple initializations





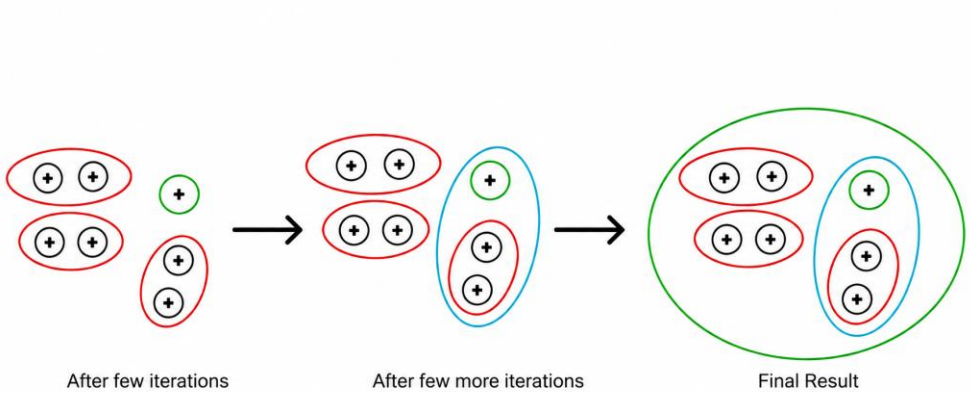
Korea Aerospace University

Part 4. Hierarchical Clustering & Evaluation

Hierarchical Clustering

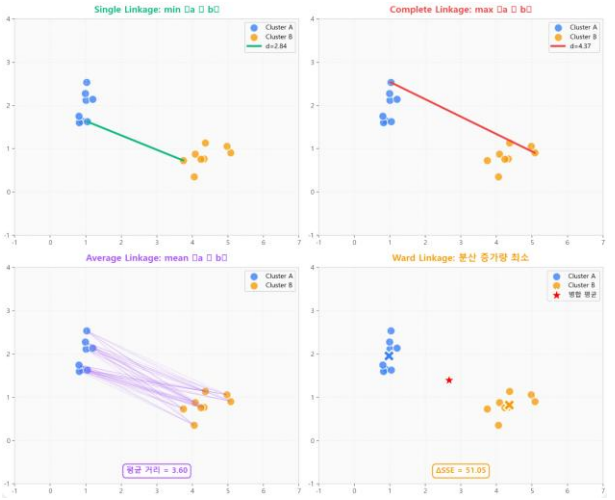
- Key Idea
  - Builds a hierarchy of clusters based on similarity
  - Represents the hierarchy as a tree structure

- Agglomerative Clustering (Bottom-Up)
  - Start with each data point as a separate cluster
  - Merge the two closest clusters
  - Repeat until all points belong to one cluster



Linkage Methods

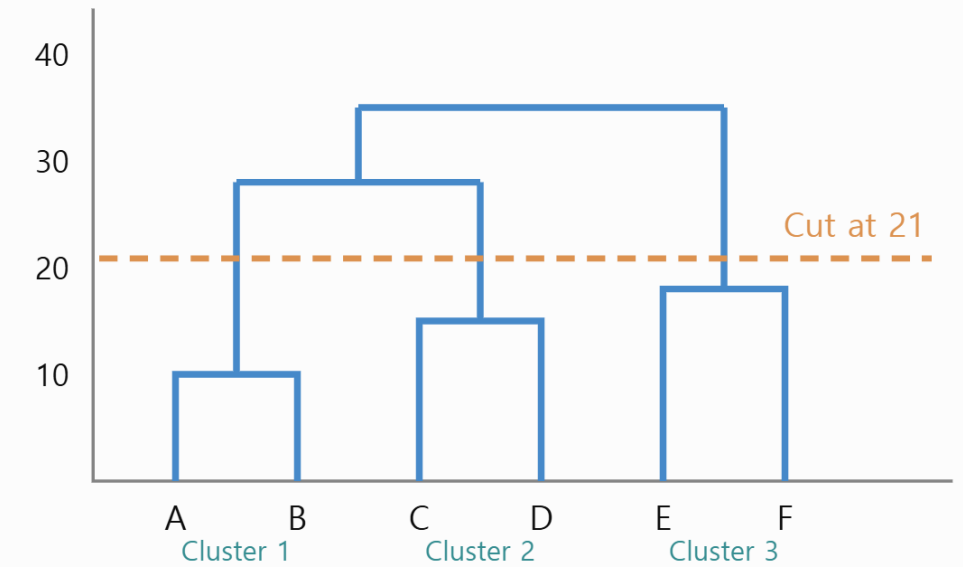
- How do we measure distance between clusters?
  - Single linkage: minimum pairwise distance
  - Complete linkage: maximum pairwise distance
  - Average linkage: average pairwise distance
  - Ward linkage: minimum increase in within-cluster sum of squares



- Key point
  - Different linkage methods can produce different clustering results

Dendrogram

- What is a dendrogram?
  - A tree diagram showing how clusters are merged
  - The height of each merge represents its dissimilarity level
- Choosing the number of clusters
  - Cut the dendrogram at a selected height
  - The resulting branches determine the clusters
- Key idea
  - A lower cut generally produces more clusters
  - A higher cut generally produces fewer clusters



K-Means vs. Hierarchical

- When to use each?
  - K-Means: large datasets with compact, well-separated clusters
  - Hierarchical: smaller datasets where hierarchical structure is useful

	K-Means	Hierarchical
Approach	Centroid-based	Hierarchy-based
Number of Clusters	Specify K in advance	Select from dendrogram
Output	Flat clusters	Cluster hierarchy
Efficiency	Generally faster	Often more expensive
Main limitation	Initialization & cluster shape	Linkage choice & scalability

Silhouette Score

- Evaluating clustering
  - Good clustering aims for high cohesion and separation
- Silhouette Score

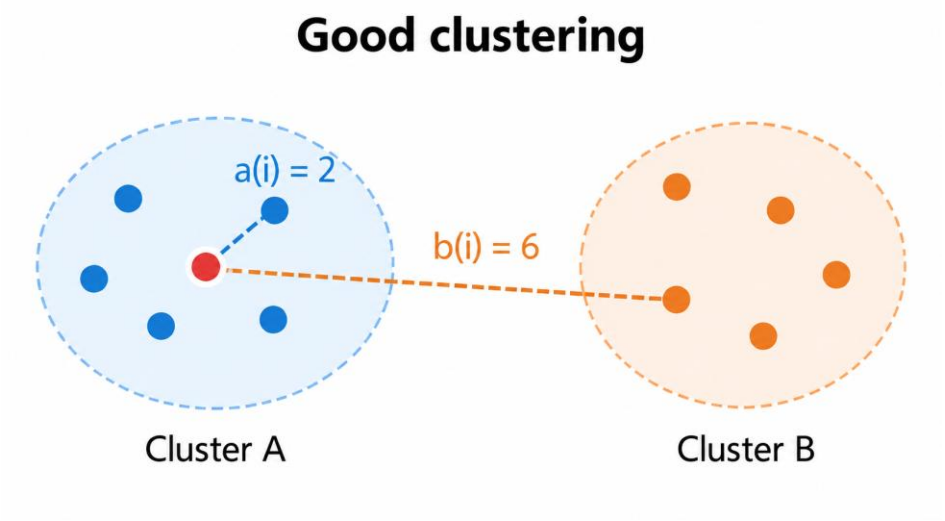
$$s(i) = \frac{b(i) - a(i)}{\max\left(a(i), b(i)\right)}$$

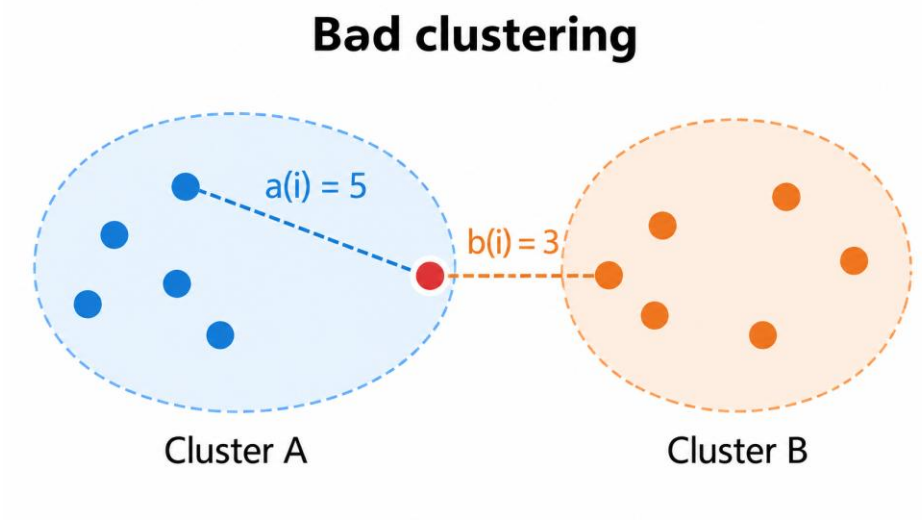
  - $a(i)$: average distance to points in the same cluster
  - $b(i)$: minimum average distance to points in another cluster
  - Overall score: average of $s(i)$ across all data points



Silhouette Score

- Interpretation
  - Close to +1: well clustered
  - Around 0: near a cluster boundary
  - Close to -1: closer to another cluster





Next Lecture

- Midterm Exam
  - When: Oct. 21, 2026 (Wed), 14:30−16:30
    - No entry after 14:30
    - 14:30−14:40: submit all belongings except writing utensils
    - 14:40−16:30: exam
  - Where: Electronics Building, Room 422
  - Coverage: Lec01−Lec07

Korea Aerospace University

Questions?

jiyoonshin@kau.ac.kr