Applied Machine Learning
Class 5

Master 2 Physique Fondamentale et Applications

Daniel Förster
daniel.forster@univ-orleans.fr
UFR Sciences et Techniques Université d'Orléans

Chapter 4: Unsupervised Learning

- Dimensionality reduction
- Clustering (k-means clustering, Gaussian mixture models)

Chapter 4: Unsupervised Learning
A: Dimensionality reduction

Principal Component Analysis (PCA)

- **Goal:** Reduce the dimensionality of data while explaining as much variance as possible.
- **Main idea:**
  - Transform data into a new coordinate system (shift and rotate to align with principal axis).
  - The axes corresponding to the largest moments are retained, as they account for the most variance.
- **Applications:**
  - Data visualization in 2D or 3D.
  - Noise reduction.
  - Feature extraction for downstream tasks.

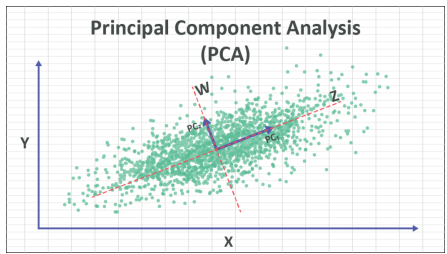


Image source [1]

How PCA Works

Steps of PCA:

1. Center the Data: Subtract the mean of each feature. You may also normalize first and standardize afterward if deemed helpful.
2. Compute covariance matrix ("tensor of inertia" if data points are considered point masses)
3. Compute Eigenvectors and Eigenvalues: Eigenvectors define the directions of principal components; eigenvalues give their magnitude.
   - Eigenvectors with the highest eigenvalues are the principal components.
4. Choose Principal Components: Select components based on cumulative explained variance.
5. Project data onto the new axes defined by the retained eigenvectors.

Eigenvalue Decomposition

Covariance matrix ($k \times k$):

$$\mathsf{S} = \frac{1}{n-1} \sum_{i=1}^{n} \mathsf{x}_i^T \mathsf{x}_i$$

where n is the number of observations, and k the number of dimensions.
Eigenvalue equation:

$$\mathsf{S}\mathsf{v}_i = \lambda_i \mathsf{v}_i$$

- v_i are eigenvectors (principal components).
- λ_i are eigenvalues (variance along each component).

Solve for v_i and λ_i for S. Sorting eigenvectors by eigenvalues in descending order.

Applying the Transformation (Rotation)

Project the data onto the new coordinate system:

$$Z = XW$$

where:

- X is the centered data.
- W contains the top k eigenvectors (principal components).
- Z is the transformed data.

Reconstruction (if needed):

$$X_{\text{reconstructed}} \approx ZW^T + \bar{x}$$

Remark: W is orthonormal.

Deciding How Many Components to Keep

Methods:

- **Explained Variance:** Keep components until a certain percentage of total variance is captured.
- **Scree Plot:** Look for the 'elbow' in the plot of eigenvalues vs. component number.

Example:

- If 95% of variance is explained by the first 3 components, you might choose to keep these 3.

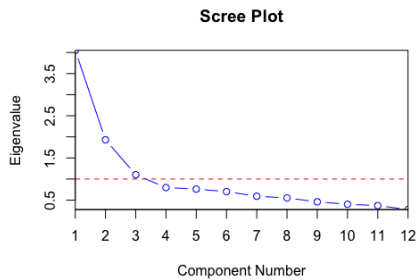


Image source [2]

Chapter 4: Unsupervised Learning
B: Clustering
1: k-means Clustering

k-Means Clustering

Unsupervised Learning: No labeled output; the structure is learned from the input data.

- **Objective:** Partition data into k clusters.
- **Cluster:** Group of data points with high intra-cluster similarity and low inter-cluster similarity.
- **Centroid:** The center of a cluster, the mean of all data points in the cluster.

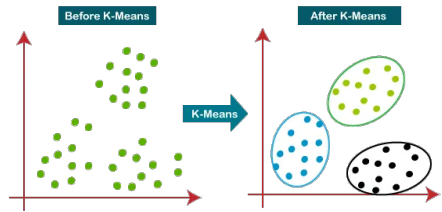


Image source [3]

k-Means Algorithm: Steps

Algorithm:

1. **Initialize:** Randomly place k centroids.
2. **Assignment Step:** Assign each data point to the nearest centroid.
3. **Update Step:** Calculate the new centroids as the mean of the points in each cluster.
4. Repeat steps 2 and 3 until centroids stabilize or a maximum number of iterations is reached.

Common distance metrics (reminder):

- **Euclidean Distance:** $d(x, y) = \sqrt{\sum_{j=1}^{m} (x_j - y_j)^2}$.
- **Manhattan Distance:** $d(x, y) = \sum_{j=1}^{m} |x_j - y_j|$.
- **Cosine Similarity:** $s(x, y) = \frac{\sum_{j=1}^{m} x_j y_j}{\sqrt{\sum_{j=1}^{m} x_j^2} \sqrt{\sum_{j=1}^{m} y_j^2}}$.

Convergence Criterion

- Stop when centroids stabilize (no significant movement).
- Alternatively, stop after a fixed number of iterations.

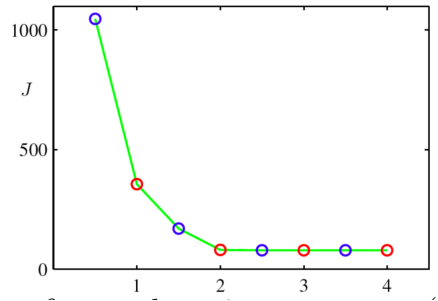


Image source [4]

K-means cost function after each assignment step (blue) and refitting step (red).

Choosing k: The Elbow Method

- Plot the within-cluster sum of squares (WCSS) against the number of clusters.
- Look for an "elbow" where adding more clusters does not significantly reduce WCSS.

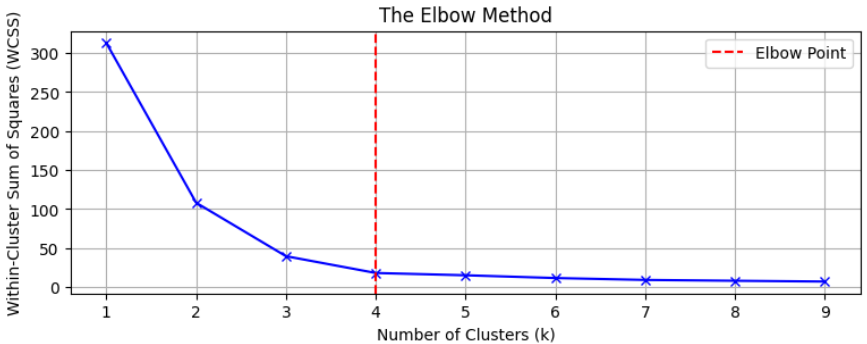


Image source [5]

Local Minima in k-Means

Challenge: Non-convex Objective Function

- The k-means objective J is non-convex.
- Coordinate descent on J is not guaranteed to converge to the global minimum.
- The algorithm may get stuck in local minima.

Mitigation Strategies

- Use multiple random initializations for centroids.
- Choose the solution with the lowest objective J after running k-means multiple times.
- Consider advanced techniques like k-means++ for better initial centroid placement.

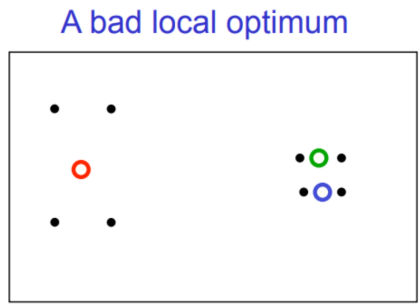


Image source [4]

Soft k-Means Clustering

- Unlike hard k-means, where each data point belongs to exactly one cluster, soft k-means assigns probabilities of membership to each cluster.
- Each data point has a fractional membership across all clusters, calculated using a **softmax function**.

Softmax Membership Function:

$$P(c_k \mid \mathrm{x}_i) = \frac{\exp(-\beta d(\mathrm{x}_i, \mu_k))}{\sum_{j=1}^{k} \exp(-\beta d(\mathrm{x}_i, \mu_j))}$$

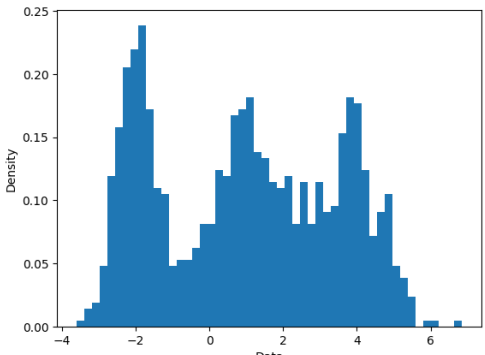
where:

- $P(c_k \mid \mathrm{x}_i)$: Probability of point x_i belonging to cluster c_k.
- $d(\mathrm{x}_i, \mu_k)$: Distance between x_i and cluster centroid μ_k.
- β: Controls the softness of cluster assignment ($\beta \to \infty$: hard assignment, $\beta \to 0$: equal membership).

Chapter 4: Unsupervised Learning
B: Clustering
2: Gaussian Mixture Models

Gaussian Mixture Models (GMMs)

- **Goal:** Cluster data by assuming the data is generated from a mixture of Gaussian distributions.
- Each cluster is represented by a Gaussian distribution, parameterized by:
  - Mean (μ): Center of the cluster.
  - Covariance (Σ): Shape and orientation of the cluster.
  - Mixing coefficient (π): Proportion of the data in each cluster.



Gaussian Mixture Model: Mathematical Formulation

Probability Density Function:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

where:

- π_k: Mixing coefficient for the k-th Gaussian, with $\sum_{k=1}^{K} \pi_k = 1$.
- $\mathcal{N}(x|\mu_k, \Sigma_k)$: Multivariate Gaussian distribution:

$$\mathcal{N}(x|\mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right)$$

- d: Dimensionality of the data.

The Expectation-Maximization (EM) Algorithm for GMMs

Steps:

1. **Initialization:**
   - Randomly initialize μ_k, Σ_k, and π_k.
2. **Expectation (E-Step):**
   - Compute the responsibility γ_{ik}:
   - γ_{ik} represents the probability that data point x_i belongs to cluster k.
3. **Maximization (M-Step):**
   - Update parameters μ_k, Σ_k, and π_k based on γ_{ik}:
4. **Repeat:**
   - Iterate between E-step and M-step until convergence.

Responsibilities in the E-Step

- **Definition:** Responsibilities represent the probability that a data point x_i belongs to cluster k.
- **Formula:**

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)}$$

- **Interpretation:** γ_{ik} is a soft assignment of x_i to cluster k, unlike hard assignments in k-means.

Updating Parameters in the M-Step

- **Mixing Coefficients (π_k):**

$$\pi_k = \frac{\sum_{i=1}^{n} \gamma_{ik}}{n}$$

- **Means (μ_k):**

$$\mu_k = \frac{\sum_{i=1}^{n} \gamma_{ik} x_i}{\sum_{i=1}^{n} \gamma_{ik}}$$

- **Covariance Matrices (Σ_k):**

$$\Sigma_k = \frac{\sum_{i=1}^{n} \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{n} \gamma_{ik}}$$