Machine Learning: Module 4 Notes

Dr. Likhit Nayak

Module Overview

This module focuses on Unsupervised Learning, specifically examining two major pillars: **Clustering** (grouping similar data) and **Feature Extraction** (dimensionality reduction and latent variable discovery).

1 Clustering Algorithms

1.1 Key Concepts and Metrics

- **Clustering:** An unsupervised learning task that groups objects such that objects in the same group (cluster) are more similar to each other than to those in other groups.
- **Hard vs. Soft Clustering:** In hard clustering (e.g., K-means), a point belongs to exactly one cluster. In soft clustering (e.g., Gaussian Mixture Models), a point has a probability of belonging to each cluster.
- **Within-Cluster Sum of Squares (WCSS):** A measure of cluster compactness. Lower WCSS indicates that data points are closer to their respective centroids.
- **The Elbow Method:** A heuristic used to determine the optimal number of clusters k. One plots WCSS against various k values; the "elbow" of the curve represents the point of diminishing returns.
- **Silhouette Score:** A metric ranging from -1 to 1 that measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). A high value indicates distinct, well-separated clusters.

1.2 K-means Clustering

K-means is a prototype-based, partitional clustering algorithm that attempts to separate data into k distinct, non-overlapping subgroups.

1.2.1 Objective Function

The algorithm minimizes the variance within each cluster. The objective function J (Inertia) is:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

where μ_i is the mean (centroid) of points in cluster C_i.

1.2.2 Algorithm Steps

1. **Initialization:** Choose k initial centroids.
2. **Assignment (E-Step):** Assign each data point to the nearest centroid based on Euclidean distance.

3. **Update (M-Step):** Recalculate centroids by taking the mean of all data points assigned to that cluster.

4. **Repeat:** Iterate steps 2 and 3 until convergence (centroids do not change or max iterations reached).

1.2.3 Initialization Strategies

- **Random Initialization:** Can lead to poor convergence or getting stuck in local optima.
- **K-means++:** Selects the first centroid randomly, then selects subsequent centroids with probability proportional to the squared distance from the closest existing centroid. This ensures centroids are spread out, leading to faster convergence and better results.

1.2.4 Limitations of K-means

1. **Assumption of Spherical Clusters:** K-means relies on Euclidean distance, assuming clusters are roughly spherical and of similar size. It fails on elongated, complex, or concentric shapes.

2. **Sensitivity to Outliers:** Since the centroid is a mean, outliers can significantly pull the centroid away from the true cluster center.

3. **Fixed k:** The user must specify the number of clusters in advance.

4. **Local Minima:** The result depends heavily on initialization; the algorithm is guaranteed to converge, but not necessarily to the global minimum.

1.3 Spectral Clustering

Spectral clustering treats clustering as a graph partitioning problem. It is powerful for identifying non-convex clusters (e.g., spirals or rings) where K-means fails.

1.3.1 Algorithm and Derivation

1. **Similarity Graph:** Construct a weighted adjacency matrix W representing similarity between points (often using a Gaussian Kernel/RBF):

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

2. **Laplacian Matrix:** Compute the unnormalized graph Laplacian $L = D - W$, where D is the diagonal degree matrix ($D_{ii} = \sum_j W_{ij}$).

3. **Spectral Embedding:** Compute the eigenvalues and eigenvectors of L. The eigenvectors corresponding to the k smallest eigenvalues (ignoring the trivial zero eigenvalue for connected graphs) form a new feature space.

4. **Clustering:** Project the data points into the space spanned by these eigenvectors (matrix U) and perform K-means on the rows of U.

1.3.2 Why it works

The "Spectrum" (eigenvalues) of the Laplacian describes the connectivity of the graph. The eigenvectors map data points such that points connected by high weights in the graph end up close together in the new space, making them easily separable by simple algorithms like K-means.

1.3.3 Limitations

1. **Computational Complexity:** Calculating eigenvalues for an $n \times n$ matrix is roughly $O(n^3)$. This makes spectral clustering slow for very large datasets ($n > 5000$).
2. **Parameter Sensitivity:** The choice of the similarity graph (e.g., k-nearest neighbors vs. fully connected) and the kernel width (σ) significantly affects performance.

1.4 Example Problem

Given: $X = \{(1,1), (1,2), (2,1), (5,5), (5,6), (6,5)\}$, $k = 2$. Initial centroids: $P_1(1,1)$ and $P_4(5,5)$.
Execution:

- **Iteration 1 Assignment:** Calculate distances. $(1,1), (1,2), (2,1)$ are closer to $(1,1)$. $(5,5), (5,6), (6,5)$ are closer to $(5,5)$.
- **Iteration 1 Update:**
  - New $\mu_1 = \text{mean}((1,1), (1,2), (2,1)) = (1.33, 1.33)$.
  - New $\mu_2 = \text{mean}((5,5), (5,6), (6,5)) = (5.33, 5.33)$.
- **Iteration 2 Assignment:** Re-checking distances shows no points change clusters.
- **Conclusion:** The algorithm converges. The clusters represent two distinct groups separated in space.

2 Feature Extraction

Feature extraction transforms high-dimensional data into a lower-dimensional space, preserving specific properties (variance, structure, separability).

2.1 Principal Component Analysis (PCA)

PCA is a linear dimensionality reduction technique that identifies orthogonal directions (principal components) of maximum variance in the data.

2.1.1 Algorithm Steps

1. **Standardization:** Center the data (mean = 0). If features have different units, scale to unit variance.
2. **Covariance Matrix:** Compute $\Sigma = \frac{1}{n-1} X^T X$.
3. **Eigendecomposition:** Find eigenvectors and eigenvalues of Σ.
4. **Sort and Select:** Sort eigenvectors by decreasing eigenvalues. Choose the top k eigenvectors to form a projection matrix W.
5. **Transform:** New features $Z = XW$.

2.1.2 Intuition

PCA minimizes the reconstruction error (distance between original point and its projection) while simultaneously maximizing the variance of the projected data. The first PC is the line of "best fit."

2.1.3 Limitations

- **Linearity:** PCA can only capture linear correlations. It fails to unfold non-linear manifolds (e.g., a Swiss Roll dataset).
- **Orthogonality Constraint:** Principal components are forced to be orthogonal, which might not represent the true underlying structure of the data.
- **Scale Sensitive:** Highly sensitive to the scaling of the original variables.

2.2 Kernel PCA (kPCA)

kPCA extends PCA to handle non-linear data distributions by mapping the input space into a higher-dimensional feature space where the data becomes linearly separable.

2.2.1 The Kernel Trick

Instead of explicitly computing the mapping $\phi(x)$, which is computationally expensive, we use a kernel function $\kappa(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$. Common kernels include:

- **Polynomial:** $\kappa(x, y) = (x^T y + c)^d$
- **Gaussian (RBF):** $\kappa(x, y) = \exp(-\gamma \|x - y\|^2)$

2.2.2 Procedure

1. Compute the Kernel Matrix (Gram Matrix) K. 2. Center the Kernel Matrix in feature space. 3. Solve the eigenvalue problem for K. 4. Project data onto the eigenvectors of the Kernel matrix.

2.2.3 Limitations

- **Hyperparameters:** Performance depends heavily on the choice of kernel and its parameters (e.g., γ in RBF).
- **Computational Cost:** Requires storing and manipulating the $N \times N$ kernel matrix, which is expensive for large datasets.

2.3 Independent Component Analysis (ICA)

ICA is a statistical technique used to reveal hidden factors that underlie sets of random variables, measurements, or signals.

2.3.1 Concept: The Cocktail Party Problem

ICA separates a multivariate signal into additive, independent non-Gaussian sources. Unlike PCA, which decorrelates data (second-order statistics), ICA aims for statistical independence (higher-order statistics).

2.3.2 Mathematical Model

Assumes $x = As$, where:

- x: Observed mixed signals.
- s: Unknown independent source signals.
- A: Unknown mixing matrix.

ICA estimates $W \approx A^{-1}$ to recover $s = Wx$.

2.3.3 Key Assumptions

1. **Independence:** Sources are statistically independent ($p(s_1, s_2) = p(s_1)p(s_2)$). 2. **Non-Gaussianity:** Sources must be non-Gaussian. (The sum of independent variables tends toward Gaussian via CLT; therefore, ICA maximizes non-Gaussianity to "un-mix" them).

2.3.4 Limitations

- **Ambiguities:** ICA cannot determine the variance (energy) or the order of the source signals.
- **Gaussian Sources:** ICA cannot separate Gaussian sources (as the mixing matrix becomes unidentifiable for rotationally symmetric distributions).

2.4 Non-Negative Matrix Factorization (NMF)

NMF factorizes a data matrix V into two matrices W and H with the strict constraint that all matrices contain no negative elements.

$$V \approx WH$$

where $V \in \mathbb{R}^{n \times m}$, $W \in \mathbb{R}^{n \times r}$, $H \in \mathbb{R}^{r \times m}$.

2.4.1 Parts-Based Representation

Because elements are non-negative, the reconstruction $V \approx WH$ involves only additive combinations. This forces the algorithm to learn "parts" of the data rather than holistic representations.

- **Example (Images):** PCA "eigenfaces" look like ghostly complete faces. NMF basis vectors look like noses, eyes, and mouths.
- **Example (Text):** In Topic Modeling, W represents topics (clusters of words), and H represents the weight of topics in documents.