Machine Learning: Module 4 Notes

Dr. Likhit Nayak

December 10, 2025

1 Clustering Algorithms

1.1 Key Concepts

- **Clustering:** The task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar to each other than to those in other groups. It is a primary task of exploratory data analysis and a common technique for statistical data analysis.
- **K-means Clustering:** A non-probabilistic, iterative algorithm that partitions a dataset into a predetermined number (k) of clusters. It aims to make the intra-cluster data points as similar as possible while also keeping the clusters as different as possible.
- **Centroid:** The arithmetic mean of all the points belonging to a cluster. In K-means, each cluster is represented by its centroid.
- **Within-Cluster Sum of Squares (WCSS):** The objective function that K-means aims to minimize. It is the sum of the squared Euclidean distances between each data point and the centroid of its assigned cluster.
- **Spectral Clustering:** A clustering technique that uses the spectrum (eigenvalues and eigenvectors) of a similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions.
- **Similarity Graph:** A graph representation of the dataset where nodes are the data points and edges connect similar points. The weight of an edge, W_{ij}, represents the similarity between points x_i and x_j.
- **Graph Laplacian (L):** A matrix representation of a graph, defined as $L = D - W$. Its properties are fundamental to spectral clustering.

1.2 K-means Clustering

K-means is a popular centroid-based clustering algorithm. Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ and a desired number of clusters k, the algorithm partitions the data into k sets $C = \{C_1, C_2, \ldots, C_k\}$ by minimizing the Within-Cluster Sum of Squares (WCSS).

1.2.1 Objective Function

The goal of K-means is to find a set of cluster assignments and centroids that minimize the WCSS:

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

where C_i is the i-th cluster and μ_i is the centroid of cluster C_i.

1.2.2 Algorithm

The K-means algorithm alternates between two steps:

1. **Initialization:** Select k initial centroids (e.g., using K-means++).
2. **Assignment Step:** Assign each data point x_j to the cluster C_i whose centroid is the closest.

$$C_i^{(t)} = \{x_j : \|x_j - \mu_i^{(t)}\|^2 \leq \|x_j - \mu_l^{(t)}\|^2 \quad \forall l\}$$

3. **Update Step:** Recalculate the centroids for each cluster.

$$\mu_i^{(t+1)} = \frac{1}{|C_i^{(t)}|} \sum_{x_j \in C_i^{(t)}} x_j$$

4. **Convergence:** Repeat until centroids stabilize.

1.3 Spectral Clustering

Spectral clustering identifies clusters with complex, non-convex shapes by transforming the data into a spectral domain.

1.3.1 Algorithm and Derivation

1. **Construct Similarity Graph:** Compute adjacency matrix W, often using a Gaussian kernel:

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

2. **Compute Graph Laplacian:** $L = D - W$, where D is the degree matrix.
3. **Eigendecomposition:** Compute the first k eigenvectors of L corresponding to the smallest eigenvalues.
4. **Low-Dimensional Embedding:** Form matrix U using these eigenvectors.
5. **Cluster:** Apply K-means to the rows of U.

2 Feature Extraction

Feature extraction transforms raw data into a more manageable group of features that are informative and non-redundant.

2.1 Principal Component Analysis (PCA)

PCA finds an orthogonal linear transformation that maximizes the variance of the projected data.

- **Step 1:** Center data $X_c = X - \mu$.
- **Step 2:** Compute covariance $\Sigma = \frac{1}{n-1} X_c^T X_c$.
- **Step 3:** Eigendecomposition $\Sigma v = \lambda v$.
- **Step 4:** Project data onto top k eigenvectors.

2.2 Kernel PCA (kPCA)

kPCA extends PCA to non-linear data using the kernel trick $\kappa(x_i, x_j)$. It diagonalizes the centered Gram matrix K rather than the covariance matrix in the input space, implicitly mapping data to a high-dimensional feature space.

2.3 Independent Component Analysis (ICA)

While PCA seeks uncorrelated variables by maximizing variance, Independent Component Analysis (ICA) seeks variables that are **statistically independent**. It is a computational method for separating a multivariate signal into additive, non-Gaussian source signals.

2.3.1 Motivation: The Cocktail Party Problem

The classic motivating example for ICA is the "Cocktail Party Problem." Imagine a room with two people speaking simultaneously and two microphones recording the audio at different locations.

- Each microphone records a weighted mixture of the two voices.
- The goal of ICA is to take these mixed recordings (observed variables) and separate them back into the two original distinct voices (latent source variables).

2.3.2 Mathematical Formulation

Let x be a vector of observed signals and s be a vector of independent source signals. We assume a linear mixing model:

$$x = As$$

where A is an unknown mixing matrix. The goal of ICA is to find an unmixing matrix W (essentially A^{-1}) such that:

$$\hat{s} = Wx$$

recovers the original independent sources.

2.3.3 Core Assumptions

To solve this problem (which is otherwise ill-posed), ICA relies on two strong assumptions:

1. **Statistical Independence:** The source signals s_i are statistically independent of each other. That is, $p(s_1, s_2) = p(s_1)p(s_2)$.
2. **Non-Gaussianity:** The source signals must have non-Gaussian distributions.
    - *Why?* By the Central Limit Theorem, the sum of independent random variables tends toward a Gaussian distribution. Therefore, a mixture of signals is usually "more Gaussian" than the individual source signals.
    - To recover the sources, ICA algorithms iterate to find a projection $w^T x$ that maximizes **non-Gaussianity**.

2.4 Non-Negative Matrix Factorization (NMF)

Non-negative Matrix Factorization (NMF) is a dimensionality reduction technique used when the data is inherently non-negative (e.g., pixel intensities, word counts). Unlike PCA or ICA, NMF constrains the resulting factors to be non-negative.

2.4.1 Key Concept: Parts-Based Representation

In PCA, the components (eigenvectors) can have positive or negative values. This leads to a "holistic" representation where patterns are built by complex cancellations (adding and subtracting).

- **NMF Constraint:** Since NMF features are non-negative, the original data must be reconstructed by **adding** components together (no subtraction allowed).
- **Result:** This forces the algorithm to learn a "parts-based" representation. For example, if applied to facial images, NMF might learn basis vectors that look like specific parts: a nose, an eye, or a mouth.

2.4.2 Mathematical Formulation

Given a data matrix $V \in \mathbb{R}^{n \times m}$ where $V_{ij} \geq 0$, NMF seeks to approximate it as the product of two non-negative matrices $W \in \mathbb{R}^{n \times r}$ and $H \in \mathbb{R}^{r \times m}$:

$$V \approx WH$$

- W: The Basis matrix (features).
- H: The Coefficient matrix (encodings/activations).
- r: The rank (number of components), usually $r \ll \min(n, m)$.