Foundations of Clustering and Regression: Lecture Notes from CS361 Machine Learning

CS361 Course Notes, IITG CSE Department
Instructor: Amit Awekar

February 20–21, 2025

Abstract

This document compiles detailed lecture notes from the CS361 Machine Learning course at IIT Guwahati's CSE Department, delivered by Amit Awekar Sir on February 20 and 21, 2025. It explores foundational clustering techniques, including density-based methods like DBSCAN (using ϵ and MinPts), Shared Nearest Neighbor (SNN) for density estimation, Jarvis-Patrick with kNN-based similarity thresholds, and centroid-based K-Means with Sum of Squared Errors (SSE) minimization. Additionally, it contrasts regression approaches—Linear (continuous outputs) versus Logistic (probabilistic)—and emphasizes critical concepts like feature scaling for Artificial Neural Networks (ANNs) and the importance of learning rate tuning in gradient descent. These notes are enriched with key mathematical insights and practical considerations.

1 Hierarchical Clustering Overview

Hierarchical clustering creates a tree of clusters (dendrogram) where leaves represent individual data points and root represents the whole dataset. Two main approaches:

Agglomerative: Bottom-up approach

Divisive: Top-down approach

- **Agglomerative**: Start with individual points as clusters and merge closest pairs
- **Divisive**: Start with one cluster and recursively split it

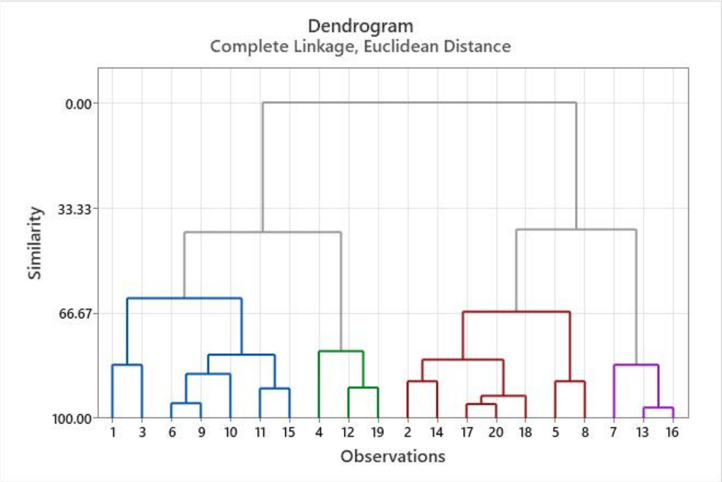

Figure 1: Hierarchical clustering dendrogram

2 Density-Based Clustering: DBSCAN

Definition 2.1: Density-Based Clustering

A clustering approach that identifies clusters as dense regions of points separated by low-density regions. Key advantages:

- Handles arbitrary shape clusters
- Robust to noise
- Doesn't require pre-specifying number of clusters

2.1 Core Concepts

- ϵ **(eps)**: Radius of neighborhood
- **MinPts**: Minimum points required to form dense region
- **Core point**: Point with $\geq$ MinPts in ϵ-neighborhood
- **Border point**: Point with ¡ MinPts but in core point's neighborhood
- **Noise point**: Neither core nor border

Algorithm 2.1: DBSCAN Algorithm Steps

1. Label all points as core, border, or noise
2. Delete noise points
3. Create edge between core points within ϵ
4. Make connected components of core points clusters
5. Assign border points to clusters

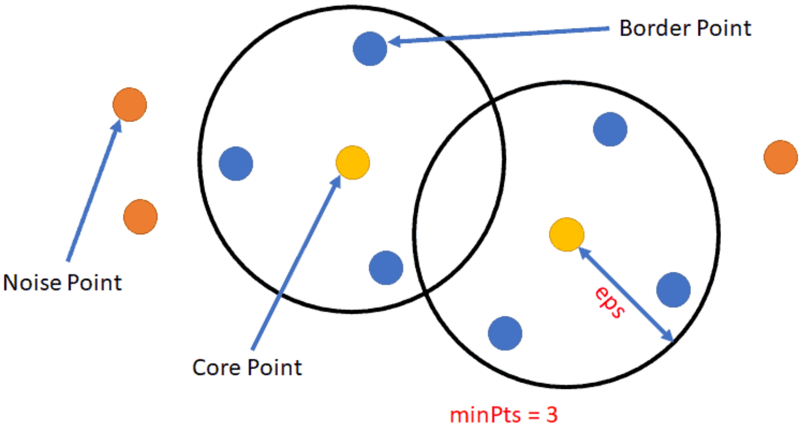

Figure 2: DBSCAN core, border, and noise points

2.2 Advantages & Limitations

Advantages:

- Handles varying density clusters
- Identifies outliers effectively

Limitations:

- Sensitive to ϵ and MinPts parameters
- Struggles with varying densities
- Border points assignment ambiguity

3 Shared Nearest Neighbor (SNN) Graphs

SNN approach addresses DBSCAN's density variation problem by considering shared neighbors:

SNN similarity = number of shared neighbors in top k neighbors

1. Create k-nearest neighbor graph
2. Compute similarity between points as number of shared neighbors
3. Apply clustering on SNN similarity matrix

Definition 3.1: SNN Density

A point's density is defined by the number of points with SNN similarity above threshold. Clusters are connected components of points with sufficient SNN density.

4 Jarvis-Patrick Algorithm

Theorem 4.1: Density Connectivity

Two points p and q are density-connected if:

$$\exists r \text{ where both } p \text{ and } q \text{ are density-reachable from } r$$

Algorithm 4.1: Jarvis-Patrick Clustering

1. Compute k-nearest neighbors for each point
2. For each pair of points:
    - If they are in each other's kNN list
    - And share ≥ s_threshold neighbors
    - Create a link between them
3. Final clusters are connected components

4.1 Advantages over DBSCAN

- More stable to parameter variations
- Better handles varying densities
- Reduces border point ambiguity

4.2 Limitations

- Sensitive to k and s_threshold choices
- May merge distinct clusters with similar neighborhoods
- Computationally expensive for large datasets

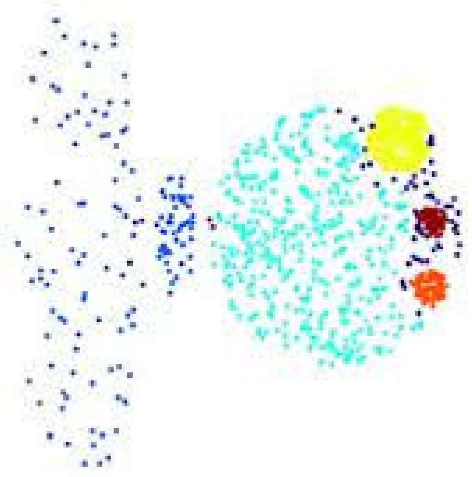

Figure 3: Jarvis-Patrick clustering result

5 K-Means Clustering

Algorithm 5.1: K-Means Algorithm

1. **Initialization**: Randomly select k initial centroids $\{\mu_1^{(0)}, ..., \mu_k^{(0)}\}$
2. **Cluster Assignment**:

$$C_i^{(t)} = \{x_p : \|x_p - \mu_i^{(t)}\|^2 \leq \|x_p - \mu_j^{(t)}\|^2 \ \forall j\}$$

3. **Centroid Update**:

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

4. Repeat steps 2-3 until convergence ($\Delta SSE < \epsilon$)

5.1 Mathematical Formulation

Definition 5.1: Sum Squared Error (SSE)

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

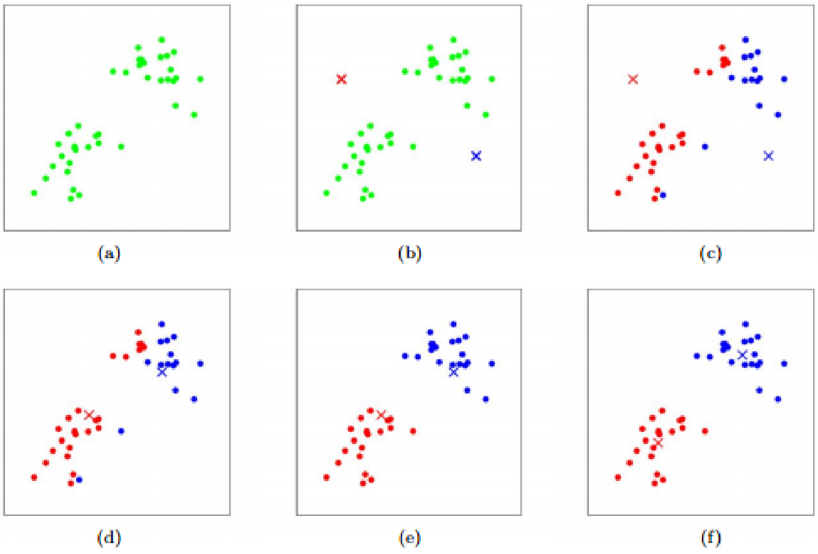

Figure 4: K-means convergence visualization

Theorem 5.1: Optimal Centroid Derivation

Minimizing SSE through differentiation:

$$\frac{\partial SSE}{\partial \mu_i} = -2 \sum_{x \in C_i} (x - \mu_i) = 0$$

$$\Rightarrow \mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

5.2 Characteristics & Complexity

Key Issues:

- Sensitive to initial centroid selection (use k-means++)
- Requires pre-specification of k (use elbow method)
- Linear complexity in each iteration: $O(nkd)$

Optimizations:

- Use triangle inequality to avoid redundant distance calculations
- Mini-batch variants for large datasets
- Multiple random initializations

6 Regression Using Artificial Neural Networks

6.1 Linear & Logistic Regression

Definition 6.1: Regression Fundamentals

- **Linear**: $y = a\mathbf{x} + b + \epsilon$
- **Logistic**: $P(y = 1|\mathbf{x}) = \sigma(a\mathbf{x} + b) = \frac{1}{1+e^{-(a\mathbf{x}+b)}}$

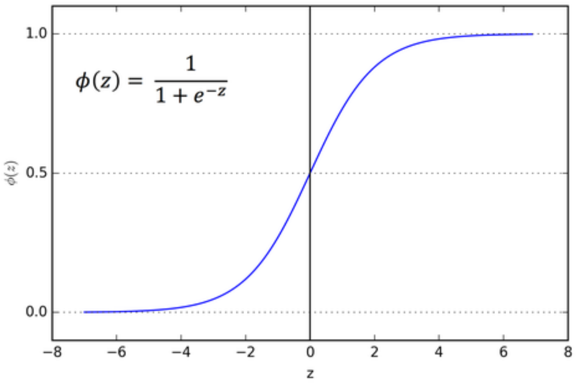

Figure 5: Sigmoid activation function

6.2 Parameter Estimation

Theorem 6.1: Gradient Descent Update

For error function E:

$$\Delta a = -\eta \frac{\partial E}{\partial a}$$
$$\Delta b = -\eta \frac{\partial E}{\partial b}$$

Where η is learning rate (0 ¡ η ¡ 1)

6.3 Likelihood Formulation

For logistic regression:

$$\text{Likelihood } L(a, b) = \prod_{i=1}^{n} P(y_i|x_i)^{y_i}(1 - P(y_i|x_i))^{1-y_i}$$
$$\text{Log-Likelihood } \ell(a, b) = \sum_{i=1}^{n} [y_i \log p_i + (1 - y_i) \log(1 - p_i)]$$

Theorem 6.2: Logistic Regression Gradients

$$\frac{\partial \ell}{\partial a} = \sum_{i=1}^{n} (y_i - p_i) x_i$$
$$\frac{\partial \ell}{\partial b} = \sum_{i=1}^{n} (y_i - p_i)$$

6.4 Feature Scaling Techniques

For variables with varying ranges:

- **Normalization**: $x' = \frac{x - x_{min}}{x_{max} - x_{min}}$
- **Standardization**: $x' = \frac{x - \mu}{\sigma}$
- **Log Transform**: $x' = \log(x + c)$ for positive-skewed data
- **Sigmoid Scaling**: $x' = \frac{1}{1+e^{-x}}$

Algorithm 6.1: Feature Transformation Pipeline

1. Handle missing values
2. Remove outliers (3σ rule)
3. Apply appropriate transformation
4. Normalize/standardize features
5. Handle categorical variables (one-hot encoding)

Summary

- DBSCAN: Density-based with ϵ and MinPts
- SNN: Shared neighbor approach for density definition
- Jarvis-Patrick: kNN-based similarity with threshold
- K-Means: Centroid-based clustering with SSE minimization
- Regression: Linear (continuous outputs) vs Logistic (probabilistic)
- Feature scaling crucial for ANN performance
- Gradient descent requires careful learning rate selection