CHAPTER 9

Unsupervised Learning Techniques

Chapter 9 surveys **unsupervised learning techniques** beyond dimensionality reduction, focusing on **clustering**, **density estimation**, and **anomaly/novelty detection**, with K-Means, DBSCAN, and Gaussian Mixture Models as core tools.

Clustering and its uses

Clustering groups similar instances into clusters without labels and is useful for customer segmentation, exploratory data analysis, recommender systems, image segmentation, semi-supervised learning (label propagation), anomaly detection, and search-by-similarity. Unlike classification, the notion of "cluster" depends on the algorithm: some (e.g., K-Means) look for spherical clusters around centroids, others (e.g., DBSCAN) for dense regions of arbitrary shape, and hierarchical methods build cluster trees.

K-Means and related methods

K-Means partitions data into k clusters by alternating between assigning each point to its closest centroid and recomputing centroids as cluster means, minimizing within-cluster squared distances ("inertia"). It is fast (roughly linear in samples, dimensions, and k when clusters exist) but sensitive to initialization and assumes roughly equal-size, spherical clusters. The chapter covers:

- **Initialization**: run the algorithm multiple times (n_init) and keep the solution with lowest inertia; K-Means++ careful seeding reduces poor local minima.
- **Hard vs soft clustering**: predict gives hard labels, while transform returns distances to each centroid, enabling soft assignments and a simple **nonlinear dimensionality reduction** by mapping each point to its distance vector in k-dimensional space.
- **Choosing k**: the **elbow method** (plot inertia vs k and find where decrease slows) and the **silhouette score**, which measures how well points fit inside their cluster vs nearest other cluster (from −1 to 1); plotting silhouette scores or diagrams helps pick a reasonable k.
  K-Means can also be used for **semi-supervised learning** by labeling cluster centroids or majority labels and propagating to cluster members.

DBSCAN and other clustering algorithms

DBSCAN identifies clusters as connected regions of high point density and marks points in low-density areas as outliers; it requires two hyperparameters: eps (neighborhood radius) and min_samples (minimum neighbor count to be a core point). Core points with overlapping neighborhoods form clusters; border points near clusters are assigned to them; isolated points become noise (label −1). DBSCAN can find any number of clusters with arbitrary shapes, is robust to outliers, and runs in about $O(m\log\ m)$, but struggles when cluster densities vary strongly and Scikit-Learn's implementation can use up to $O(m^2)$ memory if eps is large. The chapter also briefly describes agglomerative clustering (bottom-up hierarchical), BIRCH (large-scale, tree-based), Mean-Shift (mode seeking), affinity propagation (message passing), and spectral clustering (embedding similarity graph then K-Means), each with specific trade-offs in scalability and cluster shape assumptions.

Gaussian Mixture Models (GMMs)

A **Gaussian Mixture Model** assumes data is generated from a mixture of k Gaussian components, each with its own mean, covariance, and weight; clusters thus become ellipsoids that can differ in shape, size, orientation, and density. GaussianMixture uses the **Expectation–Maximization (EM)** algorithm:

- E-step: compute responsibilities (soft cluster probabilities) for each point.
- M-step: update weights, means, and covariances using those responsibilities. GMMs support:
- **Soft clustering** via predict_proba, indicating membership probabilities.
- **Sampling** via sample (generative modeling).
- **Density estimation** via scoresamples, giving log-PDF values useful for anomaly detection.

  Covariance structure is controlled by covariance_type (full, tied, diag, spherical), trading flexibility against parameter count and computational cost.

For **anomaly detection**, points with very low estimated density under the GMM (e.g., below the 4th percentile of scoresamples) can be flagged as anomalies, trading off precision and recall by adjusting the threshold. GMMs work best when clusters are roughly ellipsoidal; they can

over-split or misrepresent non-elliptical shapes (e.g., two moons), where density is still reasonable but clustering is poor.

Choosing number of clusters and Bayesian mixtures

Unlike K-Means, standard GMMs can't use inertia or silhouette scores reliably when clusters are not spherical; instead, the chapter recommends **information criteria**:

- **BIC** and **AIC**, which penalize models with more parameters while rewarding good data fit; the best k often minimizes BIC/AIC. GaussianMixture exposes .bic(X) and .aic(X), so plotting these vs k helps select k.

BayesianGaussianMixture treats component parameters and weights as random variables with priors and performs variational inference, allowing it to automatically down-weight unnecessary components (weights ≈ 0) when n_components is set higher than needed. The concentration prior (weight_concentration_prior) encodes beliefs about how many clusters to expect (low → few clusters, high → many), but its influence diminishes as data grows.

The chapter ends with an overview of other anomaly/novelty detection tools: PCA-style reconstruction error, robust covariance (EllipticEnvelope), Isolation Forest, and one-class SVM, noting which are better for anomaly detection (contaminated data) versus novelty detection (clean training data).