Mean-Shift Clustering: Implementation and Understanding

Govind Joshi

April 15, 2025

1 Overview

I have studied and implemented Mean-Shift Clustering (MSC) in my project. All derivations and commentary appear in the Jupyter notebook `mean_shift_clustering.ipynb`, and the final implementation resides in `my_rignet_utils.py`. Below is a structured summary.

2 Epanechnikov Kernel

We use the finite-support Epanechnikov kernel to weight neighbor contributions:

$$K\big(\|x_i - x_j\|\big) = \max\Big(0,\ 1 - \frac{\|x_i - x_j\|^2}{h^2}\Big).$$

This kernel assigns weights near 1 to points within radius h, tapering smoothly to zero at the boundary, and ignoring points beyond h.

3 Density Estimate

Define an (unnormalized) density estimate at point x_i by summing kernel weights:

$$p(x_i) \;\propto\; \sum_{j=1}^{N} K\big(\|x_i - x_j\|\big).$$

Although a true kernel density estimate requires a normalizing constant, MSC only uses relative densities, so the constant can be omitted.

4 Mean-Shift Update

At each iteration, every point x_i is "pulled" toward regions of higher density via:

$$x_i \;\leftarrow\; \frac{\displaystyle\sum_{j=1}^{N} a_j\, K\big(\|x_i - x_j\|\big)\, x_j}{\displaystyle\sum_{j=1}^{N} a_j\, K\big(\|x_i - x_j\|\big)},$$

where a_j is an optional attention weight (defaults to 1). Repeating this update for a fixed number of iterations or until the maximum shift falls below a tolerance causes points to converge to local density modes.

5 Mode Extraction

After convergence, we extract cluster centers (modes) as follows:

1. Compute final densities
$$p_i = \sum_{j=1}^{N} a_j \, K\big(\|x_i - x_j\|\big).$$
2. Maintain a boolean mask of *unused* points.
3. **Loop:**
   - Set densities of *used* points to $-\infty$.
   - Find $\arg\max_i p_i$ among unused points; record that x_i as a mode.
   - Mark all points within Euclidean distance h of x_i as used.
4. Repeat until no unused points remain.
5. Return the list of mode coordinates as an $M \times D$ tensor.

6 Implementation and Testing

- All code is documented in `mean_shift_clustering.ipynb`, including derivations and inline comments.
- The final, optimized implementation resides in `my_rignet_utils.py`
- I validated the implementation on synthetic data generated by `sklearn.datasets.make_blobs`, experimenting with various bandwidths h and iteration counts to confirm correct clustering behavior.