Problem 1: Clustering Synthetic Data

(a) Algorithms Implementation

• Completed in `Assigment2_Part1.ipynb` and `Assigment2_Part2.ipynb`
• Implemented using `numpy, scipy, matplotlib` , and other libraries

(b) Algorithm Performance on Three Datasets

Qualitative Analysis of Clustering Algorithms

K-means Algorithm

• Performance Across Datasets:

- Dataset A (spherical clusters): Moderate performance (ARI=0.693)
- Dataset B (non-spherical clusters): Poor performance (ARI=0.366)
- Dataset C (complex structures): Best performance among algorithms (ARI=0.711)

• Key Advantages:

- High computational efficiency and straightforward implementation
- Suitable for datasets with compact, uniformly distributed clusters
- Serves as an effective baseline method

• Main Limitations:

- Assumes spherical cluster shapes, performing poorly on non-spherical structures
- Shows moderate stability with significant performance variations across datasets

Gaussian Mixture Models (GMM)

• Performance Across Datasets:

- Dataset A: Excellent performance (ARI=0.965)
- Dataset B: Moderate performance (ARI=0.673)
- Dataset C: Poor performance (ARI=0.442)

• Key Advantages:

- Optimal for Gaussian-distributed data
- Provides probabilistic cluster assignments through soft clustering
- Excellent performance when data matches distribution assumptions

• Main Limitations:

- Strong dependence on Gaussian distribution assumption
- Least stable algorithm with high sensitivity to initialization
- Performance degrades significantly on non-Gaussian data

Mean Shift Algorithm

• Performance Across Datasets:

- Dataset A: Excellent performance (ARI=0.940)
- Dataset B: Best performance (ARI=0.874)
- Dataset C: Moderate performance (ARI=0.655)

• Key Advantages:

- Non-parametric approach with no distribution assumptions
- Most stable algorithm across different datasets
- Automatically determines number of clusters
- Adapts well to both spherical and non-spherical clusters

• Main Limitations:

- High sensitivity to bandwidth parameter selection
- Higher computational complexity compared to K-means
- Performance decline on extremely complex structures

Performance Summary Table

Algorithm	Dataset Performance	Advantages	Limitations
K-means	A: Moderate (0.693) B: Poor (0.366) C: Best (0.711)	Efficient implementation; suitable for compact clusters	Spherical shape assumption; poor non-spherical performance; variable stability

Algorithm	Dataset Performance	Advantages	Limitations
GMM	A: Excellent (0.965) B: Moderate (0.673) C: Poor (0.442)	Optimal for Gaussian data; probabilistic assignments	Gaussian distribution dependency; initialization sensitivity; unstable
Mean Shift	A: Excellent (0.940) B: Best (0.874) C: Moderate (0.655)	No distribution assumptions; automatic cluster count; most stable	Bandwidth sensitivity; high computation cost; complex structure limitations

(c) Sensitivity Analysis of Mean Shift to Bandwidth Parameter

!../images/part1/img.png

Bandwidth Parameter Sensitivity Analysis

1. **Sensitivity Patterns:**
   - High sensitivity range (h=0.1-1.0): ARI increases dramatically from 0.005 to 0.968
   - Stable range (h=1.0-2.0): Consistent performance with ARI between 0.937-0.968
   - Cluster count decreases sharply with increasing bandwidth
2. **Optimal Parameter Identification:**
   - Optimal bandwidth: h≈1.0 (ARI=0.968, 5 clusters detected)
   - Suboptimal range: h=0.7-0.9 (ARI=0.891-0.949, 7-16 clusters)
3. **Quantitative Sensitivity:**
   - ARI increases approximately 0.1 per 0.1 bandwidth increase in sensitive range
   - Cluster count stabilizes at 4 clusters for h>1.0
   - Performance remains stable even with large bandwidth values
4. **Theoretical Consistency:**
   - Small bandwidth causes over-segmentation (theoretical expectation confirmed)
   - Large bandwidth maintains good performance (unexpected stability)
   - Optimal bandwidth provides near-optimal clustering results
5. **Practical Recommendations:**

- Recommended bandwidth range: h=0.9-1.1
- Avoid h<0.5 to prevent severe over-segmentation
- Use systematic parameter search strategies for optimal results

Conclusion: The Mean Shift algorithm demonstrates significant sensitivity to bandwidth parameters, particularly in lower ranges. However, proper parameter selection within the optimal range enables excellent clustering performance across diverse dataset types.

Problem 2: Image Segmentation

(a) Comparative Analysis of Segmentation Algorithms

Algorithm Performance and Application Scenarios

1. Segmentation Quality Assessment

• **K-means:** Produces clear boundaries with high computational efficiency, but may oversimplify texture details. Ideal for applications requiring fast, straightforward segmentation.

• **GMM:** Captures gradient information and preserves detailed textures, though with increased computational requirements. Best suited for scenarios where detail preservation is critical.

• **Mean Shift:** Naturally adapts to image structures with excellent edge preservation, but requires careful parameter tuning. Optimal for boundary-sensitive applications.

2. Feature Space Impact Analysis

• **Color-only features:** K-means and GMM may lack spatial continuity; Mean Shift can cause over-segmentation.

• **Color + Position features:** All algorithms show improved spatial coherence. Mean Shift demonstrates superior object integrity preservation.

3. Parameter Sensitivity Ranking

• **K-means:** Moderate sensitivity to cluster count (K)
• **GMM:** Lowest sensitivity; most stable across parameters
• **Mean Shift:** Highest sensitivity to bandwidth parameters

4. Practical Application Guidelines

1. **Algorithm selection strategy:**
    - Simple scenarios: K-means (speed priority)
    - Complex textures: GMM (detail preservation)
    - Boundary-critical applications: Mean Shift
2. **Feature engineering recommendations:**
    - Include position features for object segmentation
    - Consider texture features for detailed segmentation
3. **Implementation priorities:**
    - Balance computational resources and quality requirements
    - GMM generally provides best quality-stability balance

(b) Feature Scaling Effects on K-means and Mean Shift

K-means Feature Scaling Analysis

Spatial feature scaling (divided by λ) directly influences feature weighting:

• **λ > 1:** Increases spatial feature importance, generating spatially compact clusters
• **λ < 1:** Emphasizes color features, producing color-uniform segments

Mean Shift Feature Scaling Effects

Independent bandwidth control for different feature types:

• **Color bandwidth (h_c):** Controls color feature sensitivity
• **Spatial bandwidth (h_p):** Governs spatial feature influence
• **Smaller scaling factors:** Increase feature sensitivity for finer segmentation

Expected Outcomes and Applications

Feature scaling enables:

• **Balanced feature importance** in K-means through λ adjustment
• **Independent smoothness control** for color and spatial features in Mean Shift
• **Scenario-optimized segmentation:**

- Complex textures: Emphasize color features
- Complex spatial structures: Prioritize spatial features
- Fine segmentation requirements: Reduce scaling factors (increase bandwidth)

This parameter flexibility allows tailored segmentation approaches for specific image characteristics and application requirements.