Mean Shift

Non-parametric clustering using kernel density estimation

🔄 Algorithm Process

💪 Key Features

- ✓ Automatically determines number of clusters
- ✓ Finds clusters of arbitrary shape
- ✓ Robust to outliers
- ✓ No assumption on cluster shape

📊 Detailed Calculation Example (2D Data)

📍 Input Data Points

x_1	x_2	x_3	x_4	x_5
(1, 2)	(2, 3)	(2, 2)	(8, 7)	(9, 8)

🎯 Goal

Starting from point x_1 = (1, 2), we'll calculate its mean shift and update its position iteratively until convergence. We'll use Gaussian kernel with bandwidth h = 3.

🔢 Step 1: Estimate Density using Gaussian Kernel

Gaussian Kernel Formula

```
K(x) = (1 / (2πh²)) · exp(-||x||² / (2h²))
```

```
For h = 3: K(x) = (1 / 56.55) · exp(-||x||² / 18)
```

Calculate distances from x_1 = (1, 2) to all points

Distance to x_1:

$d_1 = ||(1,2) - (1,2)|| = 0$

Distance to x_2:

$d_2 = ||(1,2) - (2,3)|| = \sqrt{((1-2)^2 + (2-3)^2)} = \sqrt{2} \approx 1.414$

Distance to x_3:

$d_3 = ||(1,2) - (2,2)|| = \sqrt{((1-2)^2 + (2-2)^2)} = 1$

Distance to x_4:

$d_4 = ||(1,2) - (8,7)|| = \sqrt{((1-8)^2 + (2-7)^2)} = \sqrt{74} \approx 8.602$

Distance to x_5:

$d_5 = ||(1,2) - (9,8)|| = \sqrt{((1-9)^2 + (2-8)^2)} = \sqrt{100} = 10$

🔢 Step 2: Calculate Kernel Weights

Apply Gaussian kernel to each distance

$K(d_1)$:

$K(0) = \exp(-0^2/18) = 1.000$

$K(d_2)$:

$K(1.414) = \exp(-2/18) \approx 0.895$

$K(d_3)$:

$K(1) = \exp(-1/18) \approx 0.946$

$K(d_4)$:

$K(8.602) = \exp(-74/18) \approx 0.015$

$K(d_5)$:

```
K(10) = exp(-100/18) ≈ 0.004
```

💡 Observation

Notice that points x_1, x_2, x_3 (which are close) have high weights (≈ 0.9-1.0), while distant points x_4, x_5 have very low weights (< 0.02). This ensures local density estimation.

🔢 Step 3: Compute Mean Shift Vector

Mean Shift Formula

```
m(x) = [Σᵢ xᵢ · K(||xᵢ - x||)] / [Σᵢ K(||xᵢ - x||)] - x
```

Calculate weighted sum of points

Numerator (weighted sum):

```
Σᵢ xᵢ · K(dᵢ) = (1,2)·1.000 + (2,3)·0.895 + (2,2)·0.946 + (8,7)·0.015 + (9,8)·0.004
= (1.000, 2.000) + (1.790, 2.685) + (1.892, 1.892) + (0.120, 0.105) + (0.036, 0.032)
= (4.838, 6.714)
```

Denominator (sum of weights):

```
Σᵢ K(dᵢ) = 1.000 + 0.895 + 0.946 + 0.015 + 0.004 = 2.860
```

Weighted mean:

```
x̄ = (4.838, 6.714) / 2.860 = (1.691, 2.347)
```

Mean shift vector:

$m(x_1) = (1.691, 2.347) - (1, 2) = (0.691, 0.347)$

Step 4: Update Position

Iteration 1 Update

New position:

$x_1^{(1)} = x_1^{(0)} + m(x_1^{(0)}) = (1, 2) + (0.691, 0.347) = (1.691, 2.347)$

Iteration 2

Starting from $x_1^{(1)} = (1.691, 2.347)$:

- Recalculate distances to all points
- Compute new kernel weights
- Calculate new mean shift vector
- Update position: $x_1^{(2)} \approx (1.85, 2.50)$

Iteration 3 and beyond

Continue iterations until $||m(x)|| < \varepsilon$ (convergence threshold)
Typically converges to a local mode around (1.9, 2.5) after 5-10 iterations

Step 5: Cluster Assignment

🎯 Final Result

After running mean shift for all points:

- Points x_1, x_2, x_3 converge to mode ≈ (1.9, 2.5) → **Cluster 1**
- Points x_4, x_5 converge to mode ≈ (8.5, 7.5) → **Cluster 2**

Points that converge to the same mode are assigned to the same cluster.

✅ Convergence Criterion

The algorithm stops when the mean shift vector magnitude falls below a threshold:
$||m(x)|| < \varepsilon$, where ε is typically set to 0.01 or 0.001