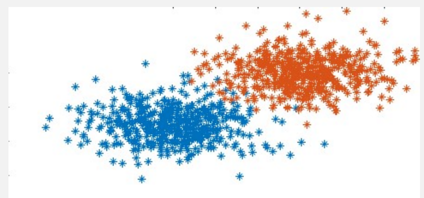
Techniques for K-Means Clustering Initialization

Michael Janov
Northeastern University
Department of Electrical and Computer Engineering
December 14, 2017

Background and Preface

- Randomized initialization K-Means clustering
  - Low stability
  - Inelegant
- All evaluated with a known number of k means
- All distances considered Euclidean
- Linear Assignment (LAKM) [1]
- Density [2]

Data Sets Used

2-Means Gaussian	Iris
	
Wine	**Glass**
	

From UCI Machine Learning Repository [3] [4] [5]

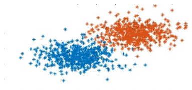
Background and Preface

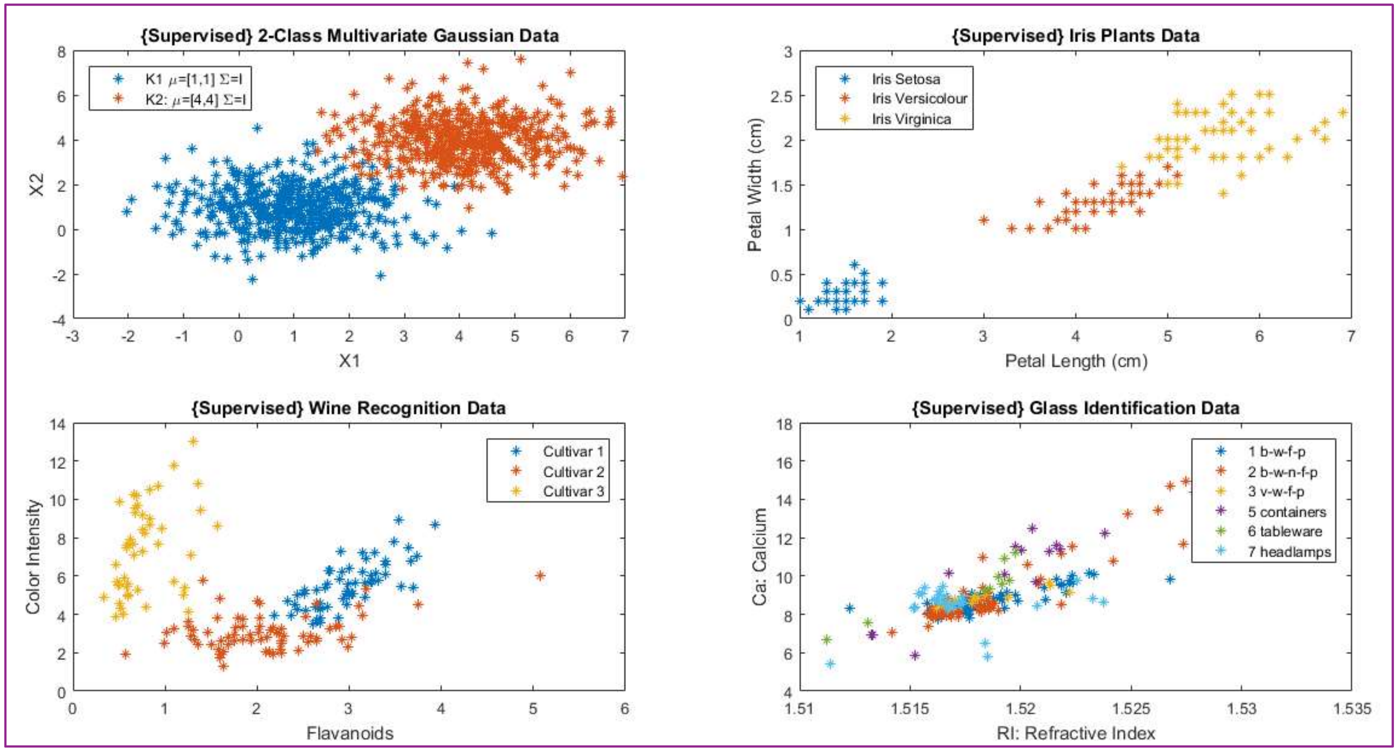
- Randomized initialization K-Means clustering
  - Low stability
  - Inelegant
- All evaluated with a known number of k means
- All distances considered Euclidean
- Linear Assignment (LAKM) [1]
- Density [2]

Data Sets Used	
2-Means Gaussian 2 classes 2 attributes 1000 data points	**Iris** 3 classes 4 attributes 150 data points
Wine 3 classes 13 attributes 178 data points	**Glass** 7 classes 11 attributes 214 data points

From UCI Machine Learning Repository [3] [4] [5]

Supervised Data

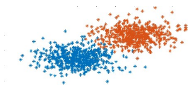


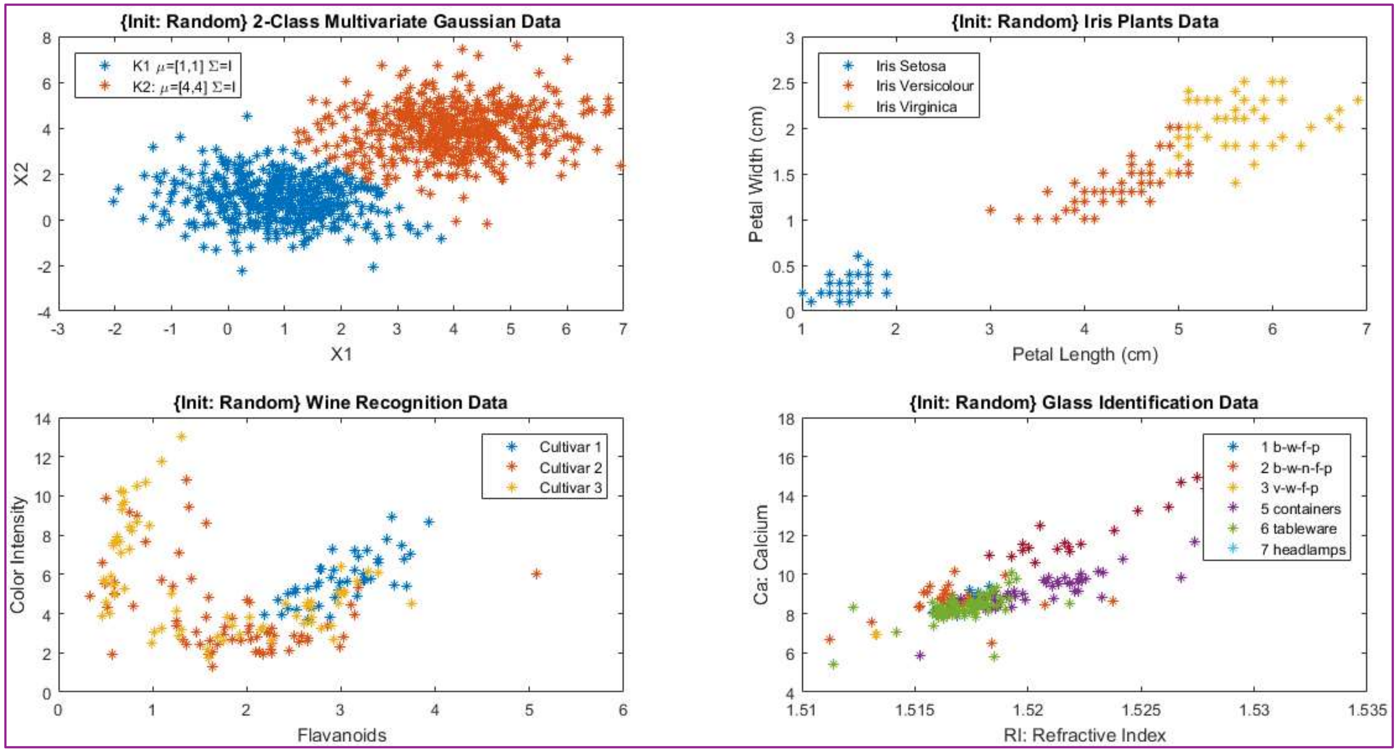


Randomized Initialization: Algorithm

1. Randomly select a k data points to serve as class means
2. Assign label based on distance between data point and class mean
3. Store sum-squared-error of current solution
4. Repeat steps 1-3 ten times
5. Keep the solution with the lowest sum-squared-error

Randomized Initialization: Results





Randomized Initialization: Results

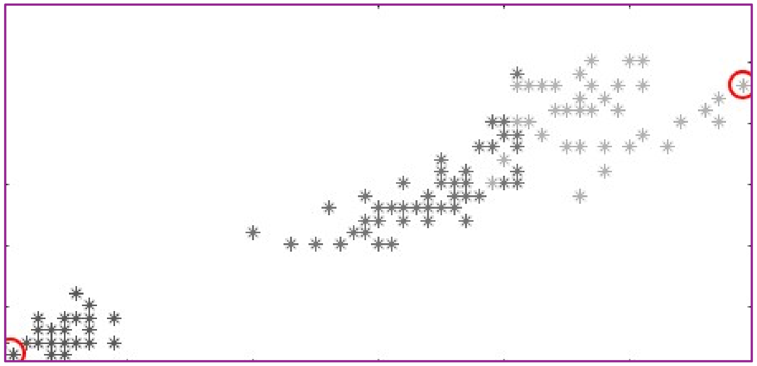
	Gauss2	Iris	Wine	Glass
Accuracy	98.70%	84.67%	71.35%*	2.80%*
Execution Time (ms)	44.8	8	9.9	15.6
Iterations	10	10	10	10

Linear Assignment Initialization: Algorithm

1. Compute distance between each data point
2. Choose the two farthest points as the initial **cluster representatives**
3. For each additional class, choose the farthest point from currently-existing representatives
4. Classify data based on distance from representatives
5. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
6. Repeat 5 until convergence

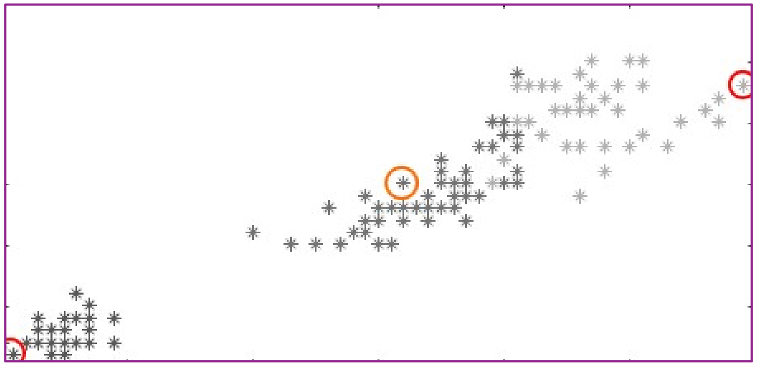
Linear Assignment Initialization: Algorithm

1. Compute distance between each data point
2. Choose the two farthest points as the initial **cluster representatives**
3. For each additional class, choose the farthest point from currently-existing representatives
4. Classify data based on distance from representatives
5. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
6. Repeat 5 until convergence



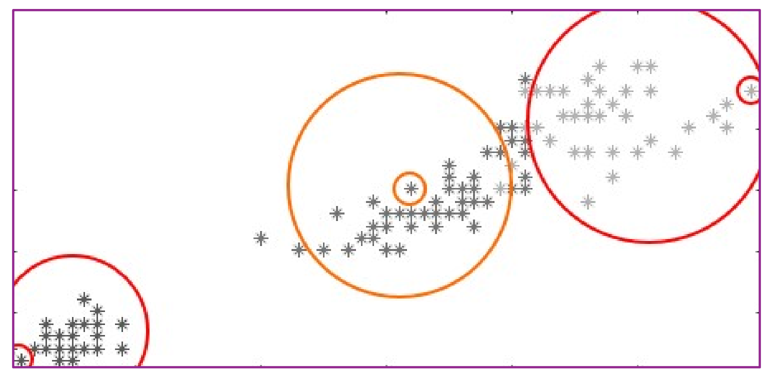
Linear Assignment Initialization: Algorithm

1. Compute distance between each data point
2. Choose the two farthest points as the initial **cluster representatives**
3. For each additional class, choose the farthest point from currently-existing representatives
4. Classify data based on distance from representatives
5. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
6. Repeat 5 until convergence



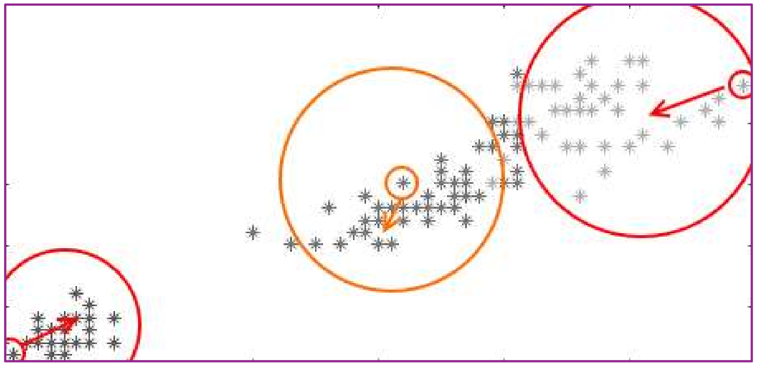
Linear Assignment Initialization: Algorithm

1. Compute distance between each data point
2. Choose the two farthest points as the initial **cluster representatives**
3. For each additional class, choose the farthest point from currently-existing representatives
4. Classify data based on distance from representatives
5. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
6. Repeat 5 until convergence



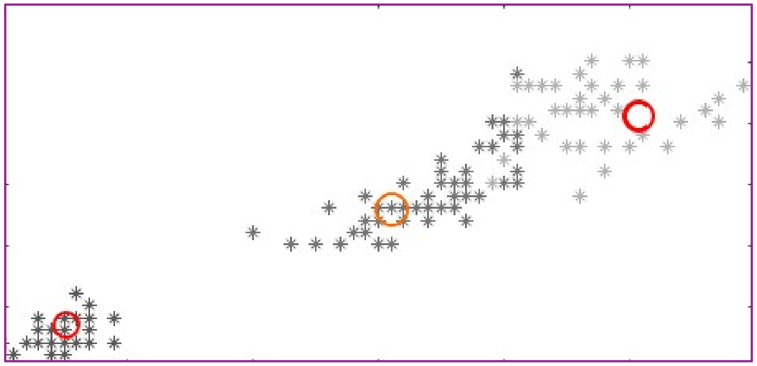
Linear Assignment Initialization: Algorithm

1. Compute distance between each data point
2. Choose the two farthest points as the initial **cluster representatives**
3. For each additional class, choose the farthest point from currently-existing representatives
4. Classify data based on distance from representatives
5. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
6. Repeat 5 until convergence



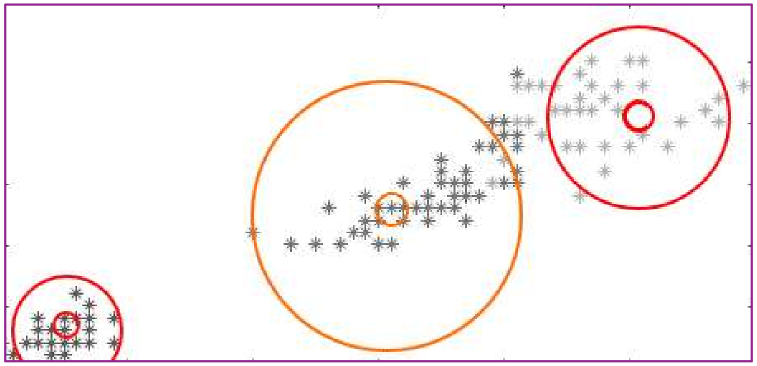
Linear Assignment Initialization: Algorithm

1. Compute distance between each data point
2. Choose the two farthest points as the initial **cluster representatives**
3. For each additional class, choose the farthest point from currently-existing representatives
4. Classify data based on distance from representatives
5. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
6. Repeat 5 until convergence

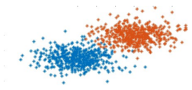


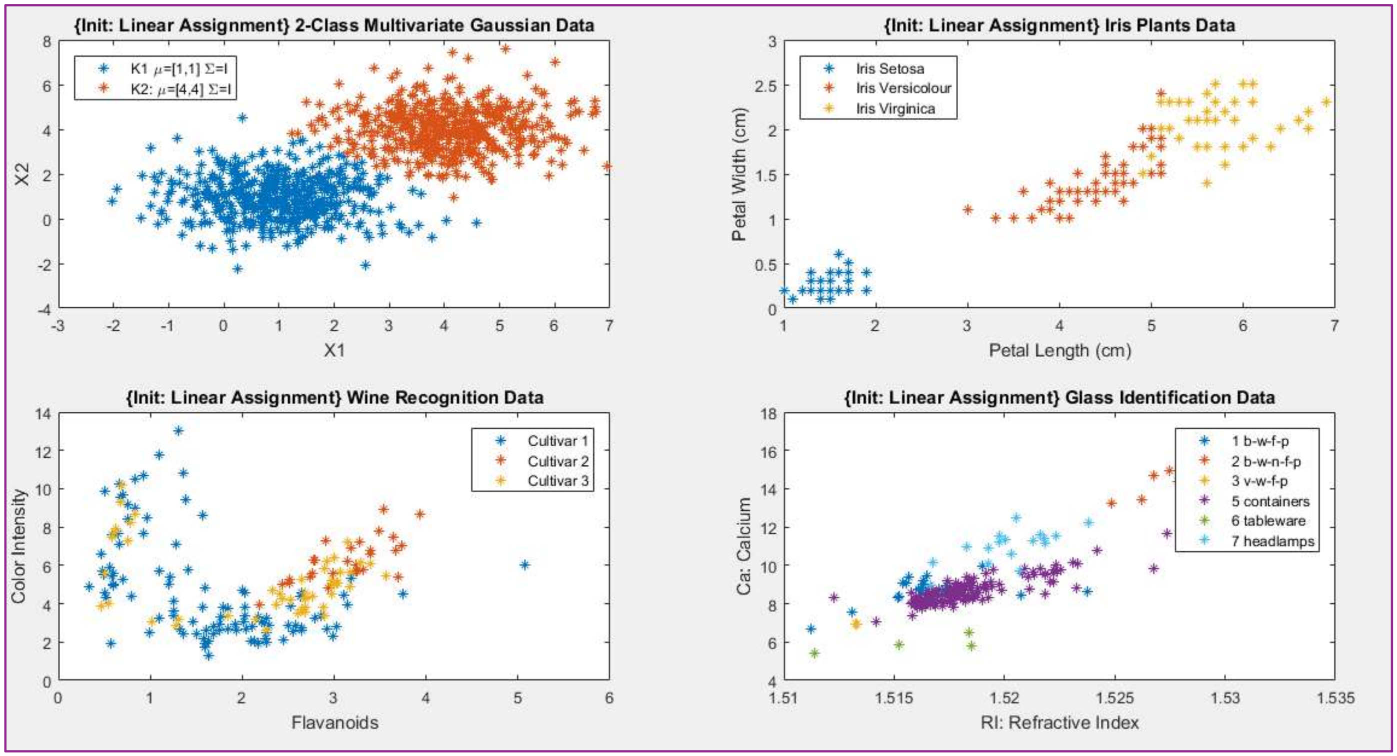
Linear Assignment Initialization: Algorithm

1. Compute distance between each data point
2. Choose the two farthest points as the initial **cluster representatives**
3. For each additional class, choose the farthest point from currently-existing representatives
4. Classify data based on distance from representatives
5. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
6. Repeat 5 until convergence



Linear Assignment Initialization: Results





Linear Assignment Initialization: Results

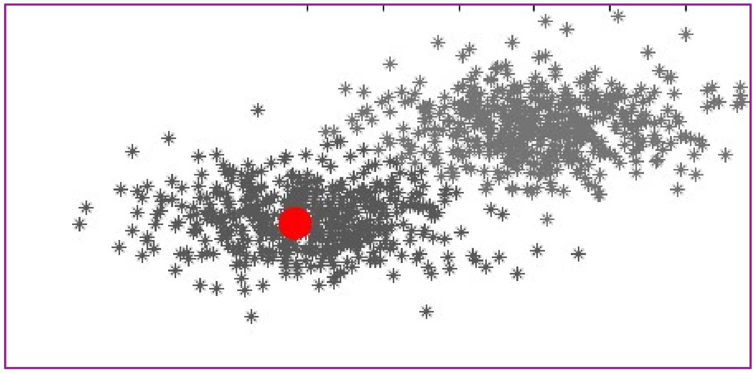
	Gauss2	Iris	Wine	Glass
Accuracy	98.6%	89.33%	6.74%*	4.67%*
Execution Time (ms)	703.6	19.1	30.4	48.1
Convergence Iterations	2	3	7	6

Density-Based Initialization: Algorithm

1. Compute distance between each data point
2. Calculate local density at all data points
   - $density(x_i) = \sum_{j=1}^{n} exp(-\frac{d(x_i,x_j)^2}{2R^2})$
   - Where x_j indicates a point in the local radius
3. Choose highest density point as first cluster mean. Remove all points it its local radius from further consideration.
4. Repeat 3 until k means are reached
5. Assign labels based on distance
6. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
7. Repeat 6 until convergence

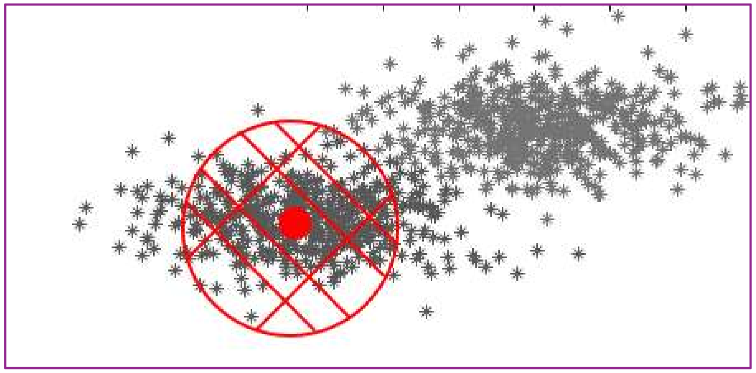
Density-Based Initialization: Algorithm

1. Compute distance between each data point
2. Calculate local density at all data points
   - $density(x_i) = \sum_{j=1}^{n} exp(-\frac{d(x_i,x_j)^2}{2R^2})$
   - Where x_j indicates a point in the local radius
3. Choose highest density point as first cluster mean. Remove all points it its local radius from further consideration.
4. Repeat 3 until k means are reached
5. Assign labels based on distance
6. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
7. Repeat 6 until convergence



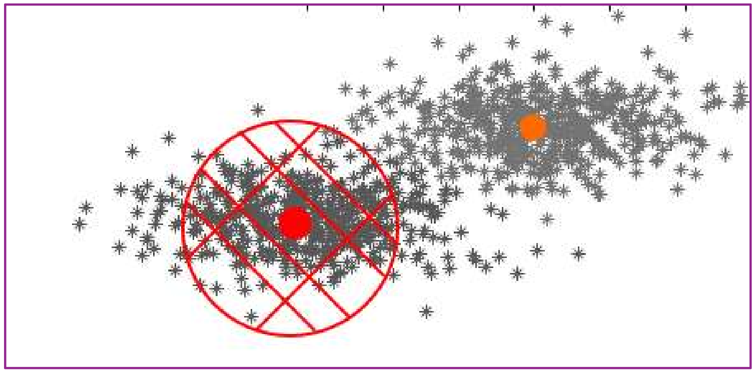
Density-Based Initialization: Algorithm

1. Compute distance between each data point
2. Calculate local density at all data points
   - $density(x_i) = \sum_{j=1}^{n} exp(-\frac{d(x_i,x_j)^2}{2R^2})$
   - Where x_j indicates a point in the local radius
3. Choose highest density point as first cluster mean. Remove all points it its local radius from further consideration.
4. Repeat 3 until k means are reached
5. Assign labels based on distance
6. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
7. Repeat 6 until convergence



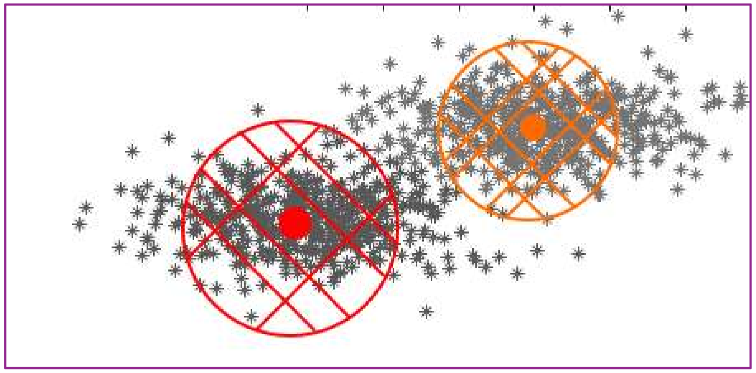
Density-Based Initialization: Algorithm

1. Compute distance between each data point
2. Calculate local density at all data points
   - $density(x_i) = \sum_{j=1}^{n} exp(-\frac{d(x_i,x_j)^2}{2R^2})$
   - Where x_j indicates a point in the local radius
3. Choose highest density point as first cluster mean. Remove all points it its local radius from further consideration.
4. Repeat 3 until k means are reached
5. Assign labels based on distance
6. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
7. Repeat 6 until convergence



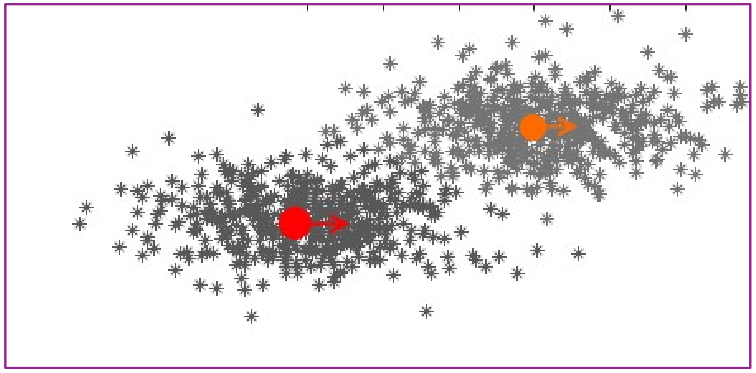
Density-Based Initialization: Algorithm

1. Compute distance between each data point
2. Calculate local density at all data points
   - $density(x_i) = \sum_{j=1}^{n} exp(-\frac{d(x_i,x_j)^2}{2R^2})$
   - Where x_j indicates a point in the local radius
3. Choose highest density point as first cluster mean. Remove all points it its local radius from further consideration.
4. Repeat 3 until k means are reached
5. Assign labels based on distance
6. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
7. Repeat 6 until convergence



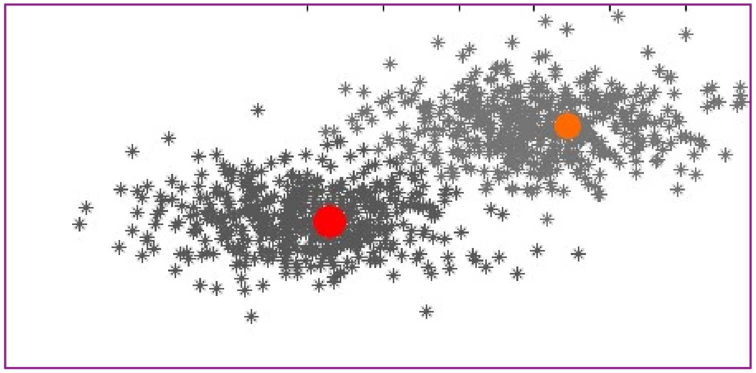
Density-Based Initialization: Algorithm

1. Compute distance between each data point
2. Calculate local density at all data points
   - $density(x_i) = \sum_{j=1}^{n} exp(-\frac{d(x_i,x_j)^2}{2R^2})$
   - Where x_j indicates a point in the local radius
3. Choose highest density point as first cluster mean. Remove all points it its local radius from further consideration.
4. Repeat 3 until k means are reached
5. Assign labels based on distance
6. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
7. Repeat 6 until convergence



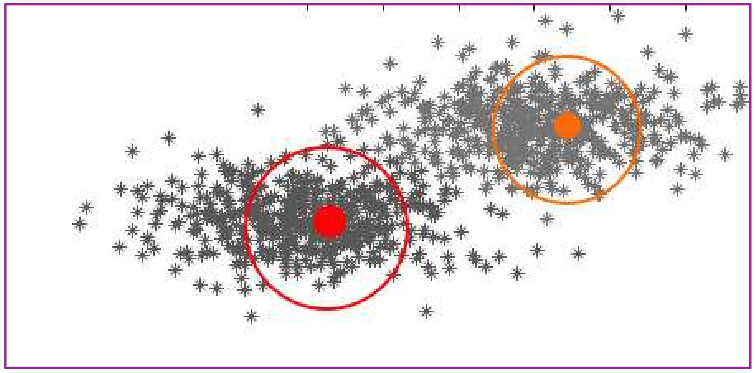
Density-Based Initialization: Algorithm

1. Compute distance between each data point
2. Calculate local density at all data points
   - $density(x_i) = \sum_{j=1}^{n} exp(-\frac{d(x_i,x_j)^2}{2R^2})$
   - Where x_j indicates a point in the local radius
3. Choose highest density point as first cluster mean. Remove all points it its local radius from further consideration.
4. Repeat 3 until k means are reached
5. Assign labels based on distance
6. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
7. Repeat 6 until convergence

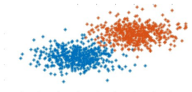


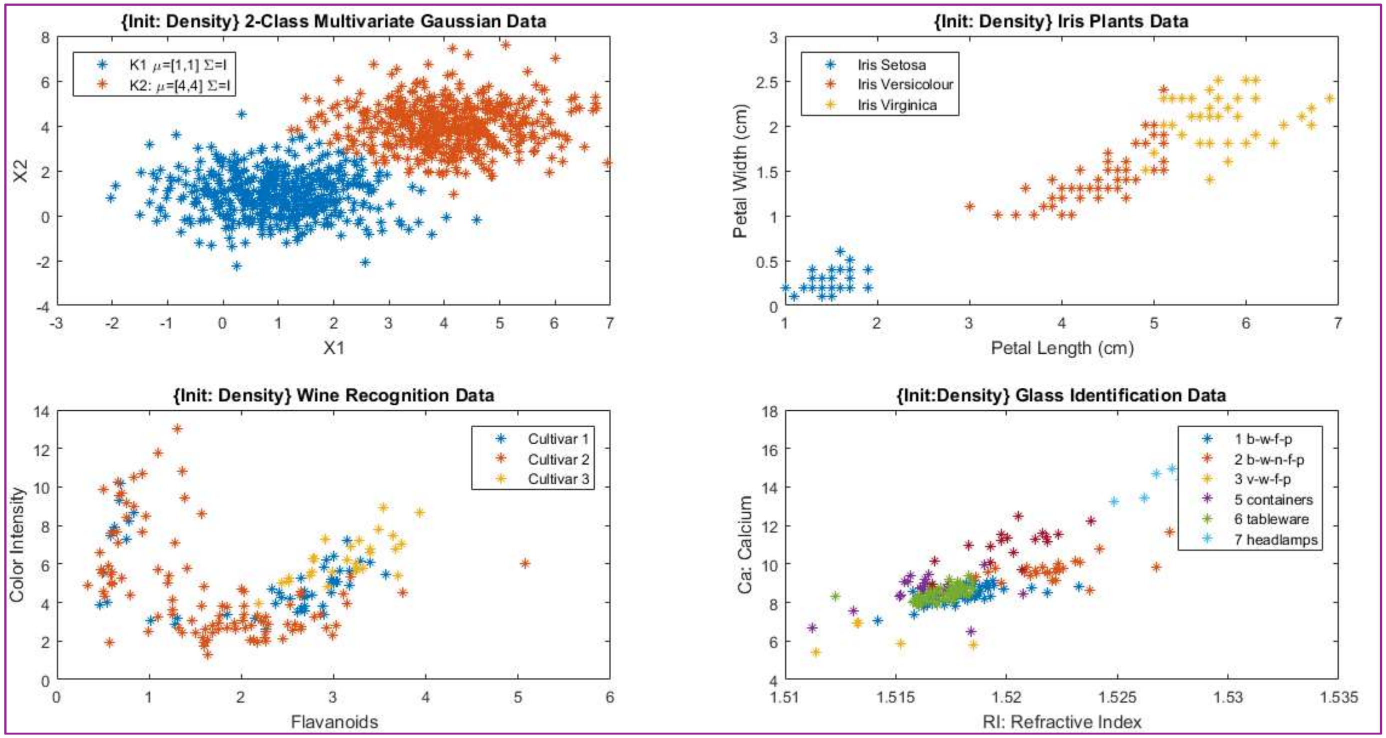
Density-Based Initialization: Algorithm

1. Compute distance between each data point
2. Calculate local density at all data points
   - $density(x_i) = \sum_{j=1}^{n} exp(-\frac{d(x_i,x_j)^2}{2R^2})$
   - Where x_j indicates a point in the local radius
3. Choose highest density point as first cluster mean. Remove all points it its local radius from further consideration.
4. Repeat 3 until k means are reached
5. Assign labels based on distance
6. Given current classifications, calculate the true cluster mean for each and reassign labels based on distance
7. Repeat 6 until convergence



Density-Based Initialization: Results

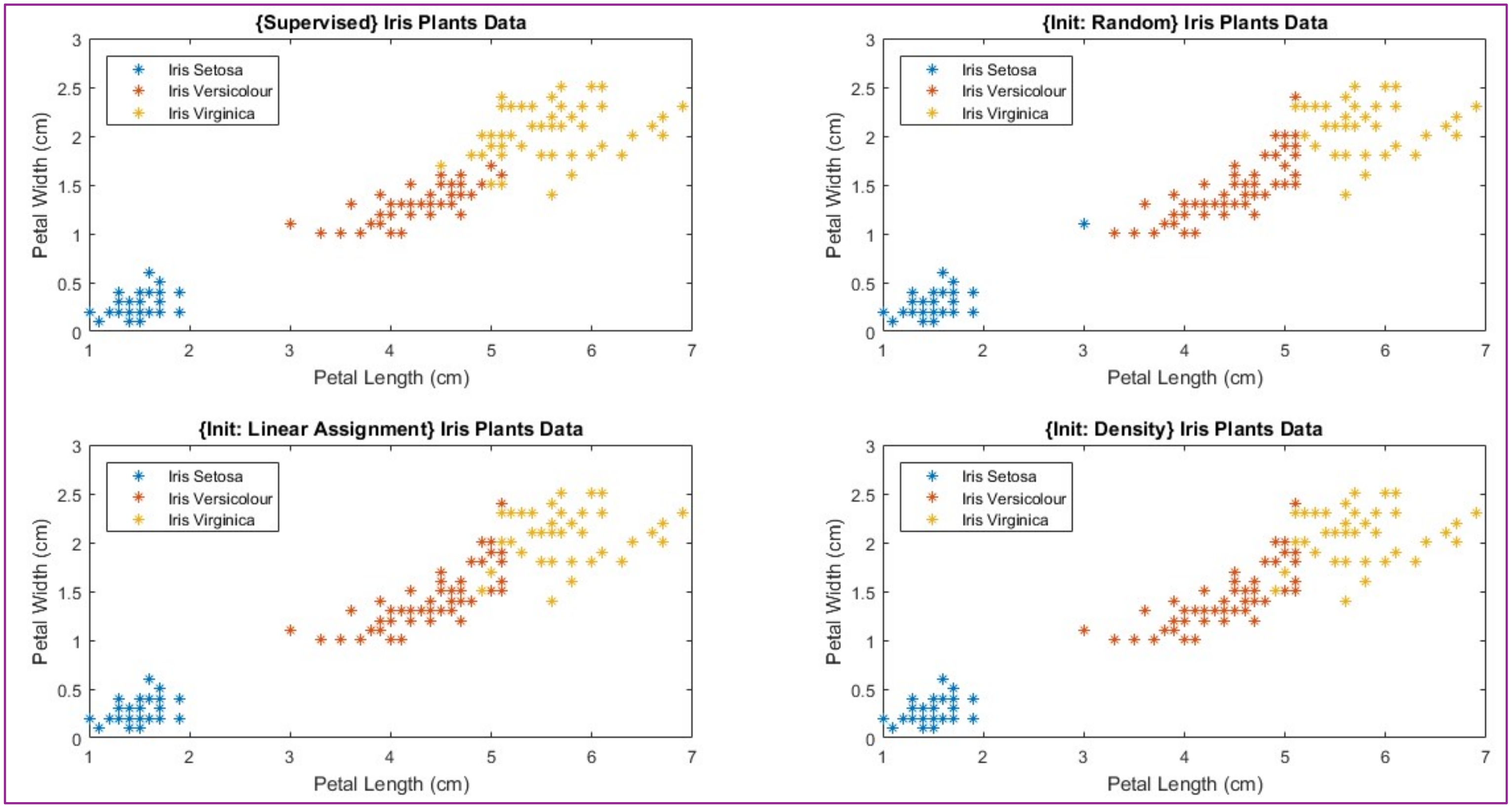




Density-Based Initialization: Results

	Gauss2	Iris	Wine	Glass
Accuracy	98.6%	89.33%	53.37%*	11.68%*
Execution Time (ms)	696.6	20.2	32.2	39.2
Convergence Iterations	6	8	12	5

Iris Results: All Four



Accuracy Results

Accuracy	Gauss2	Iris	Wine	Glass
Randomized	98.70%	84.67%	71.35%*	2.80%*
Linear Assignment	98.6%	89.33%	6.74%*	4.67%*
Density	98.6%	89.33%	53.37%*	11.68*

Time Results

Time (ms)	Gauss2	Iris	Wine	Glass
Randomized	44.8	8	9.9	15.6
Linear Assignment	703.6	19.1	30.4	48.1
Density	696.6	20.2	32.2	39.2

Convergence Iterations Results

Convergence Iterations	Gauss2	Iris	Wine	Glass
Randomized	10	10	10	10
Linear Assignment	2	3	7	6
Density	6	8	12	5

Closing Thoughts

- Density Initialization's Radius value is extremely sensitive to solution, yet little discussion from source is provided
- Highly-overlapping attribute data is poorly-suited for k-means clustering
- Correct "naming" of clusters proves problematic
- Code available on GitHub:
  - https://github.com/mikejanov/k-means-clustering-initialization

References and Sources

[1] K. L. Cheng, J. Fan and J. Wang, "A two-pass clustering algorithm based on linear assignment initialization and k-means method," 2012 5th International Symposium on Communications, Control and Signal Processing, Rome, 2012, pp. 1-5.

[2] Q. Yuan, H. Shi and X. Zhou, "An optimized initialization center K-means clustering algorithm based on density," 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, 2015, pp. 790-794.

[3] R.A. Fisher. "UCI Machine Learning Repository: Iris Data Set," 1936. Irvine, CA: University of California, School of Information and Computer Science.

[4] Forina, M. et al. "UCI Machine Learning Repository: Wine Data Set," 1991. Irvine, CA: University of California, School of Information and Computer Science.

[5] B. German. "UCI Machine Learning Repository: Glass Identification Data Set," 1987. Irvine, CA: University of California, School of Information and Computer Science.

Thank You!

Questions?