Unsupervised ML for Classification Problems

John Branch Bedoya
Pedro Atencio Ortiz

November 24, 2020

UNIVERSIDAD NACIONAL DE COLOMBIA

ITM Institución Universitaria Reacreditada en Alta Calidad

pedroatencio@itm.edu.co

Course Sesions

1. **Sesion 1:** Image Segmentation (24/11/2020)
2. **Sesion 2:** Image Classification (01/12/2020)
3. **Sesion 3:** Sequence Summarization (08/12/2020)

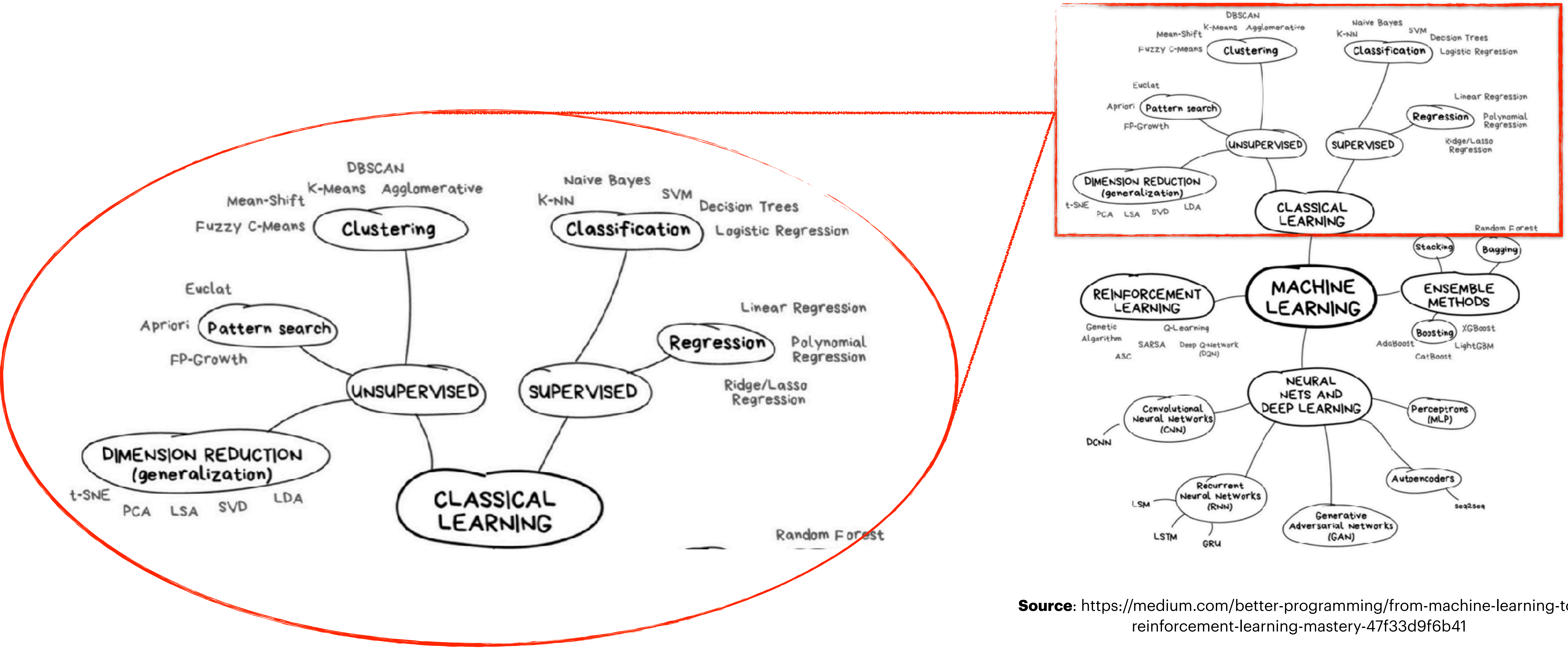
Agenda

Sesion 1: Image Segmentation

1. ML Landscape.
2. Unsupervised Learning Fundamentals
    1. Taxonomy
    2. Applications
3. Clustering
    1. Partitioning-based clustering
        1. K-Means Clustering.
    2. Density-based clustering
        1. Mean-Shift.
        2. DBScan
4. Python practice: Image Segmentation.
5. Conclusions.

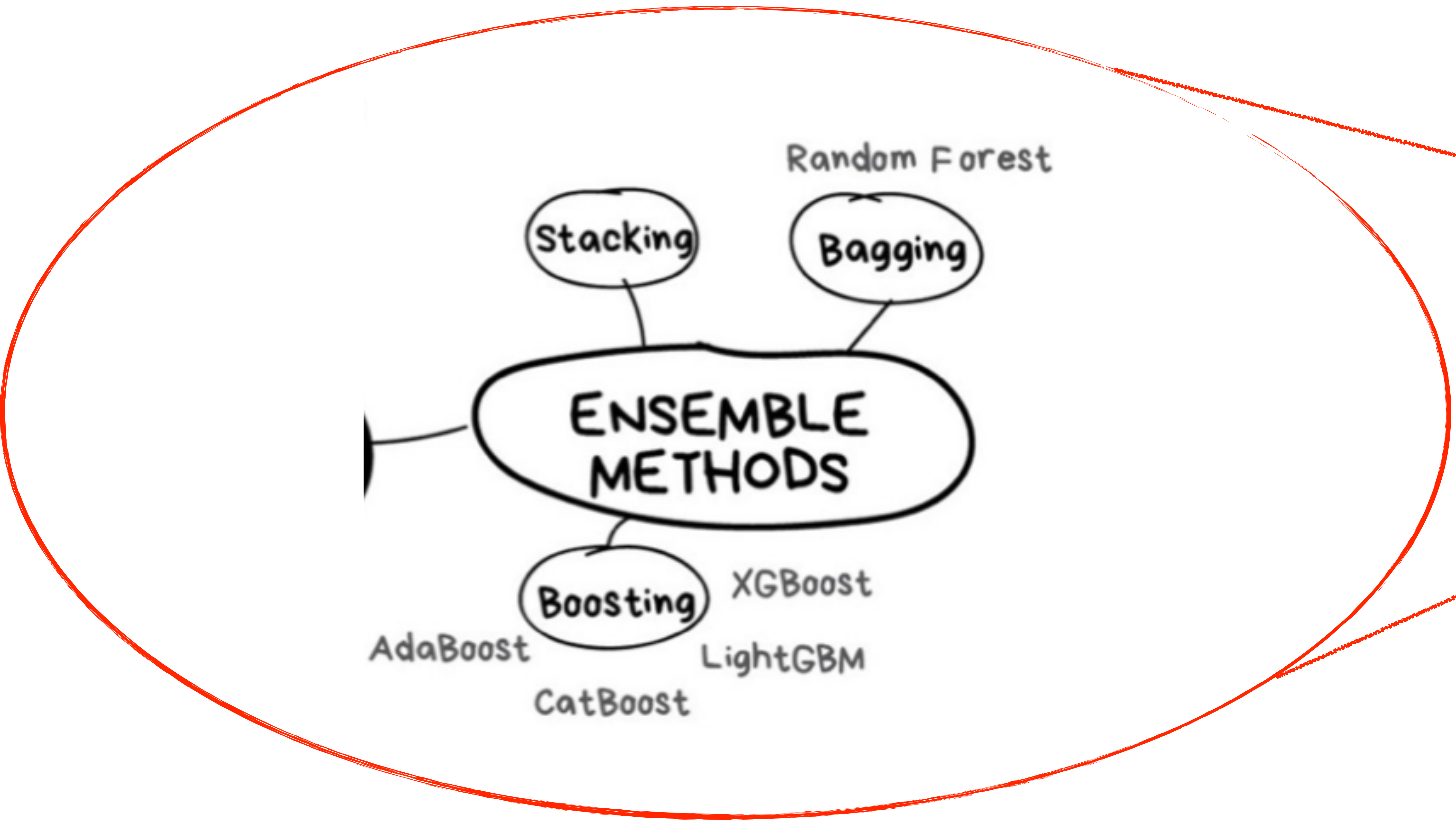
1. Machine Learning Landscape

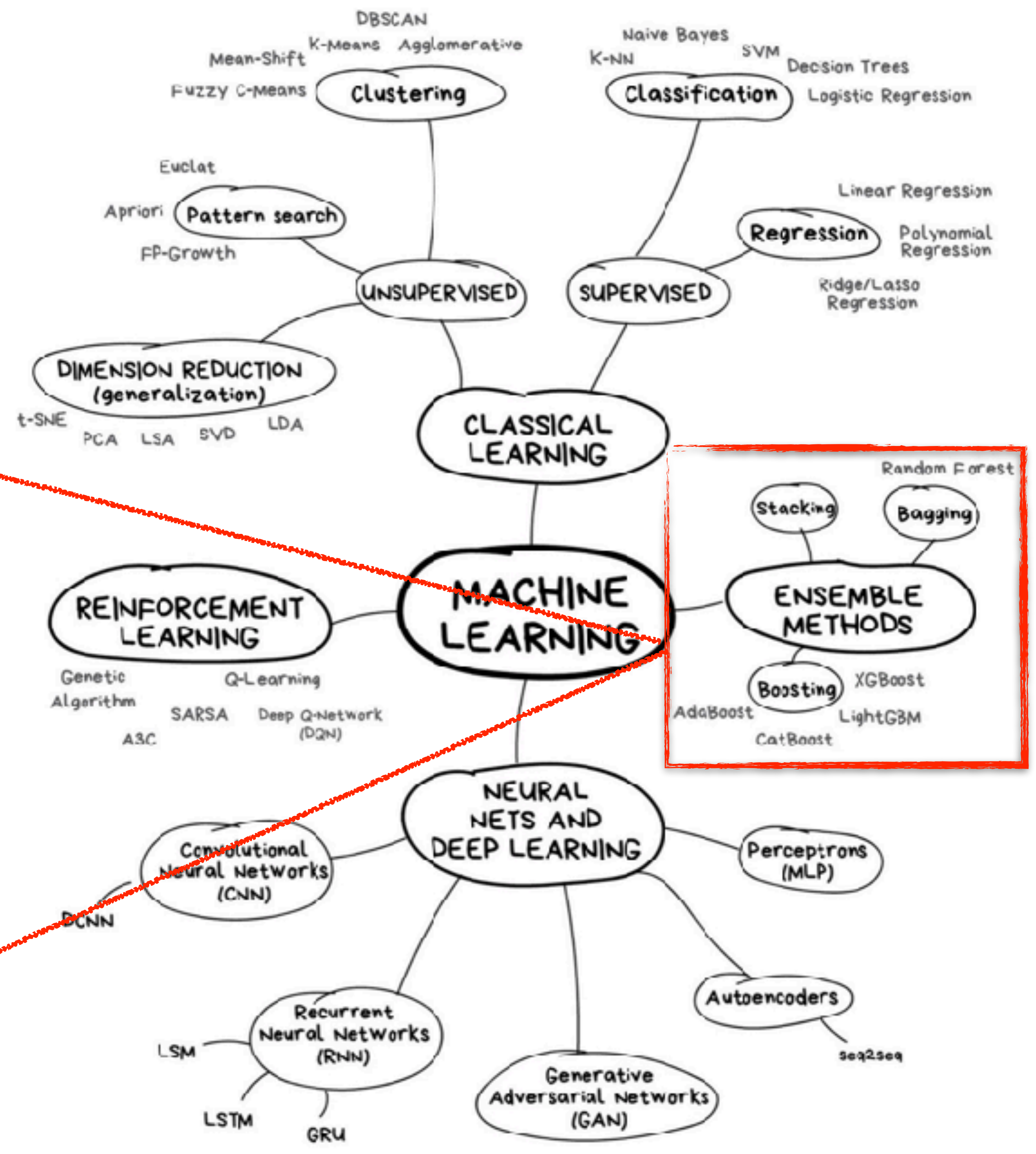
ML Landscape



Source: https://medium.com/better-programming/from-machine-learning-to-reinforcement-learning-mastery-47f33d9f6b41

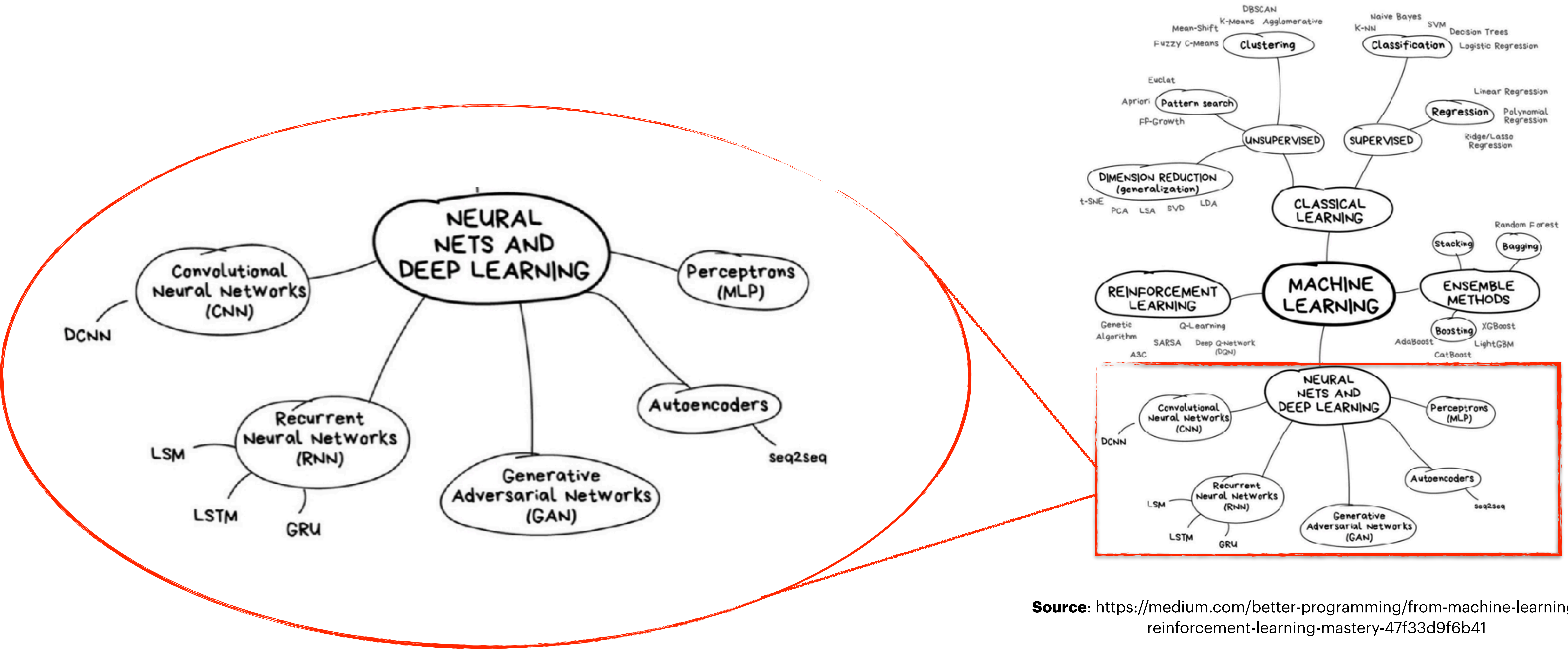
ML Landscape





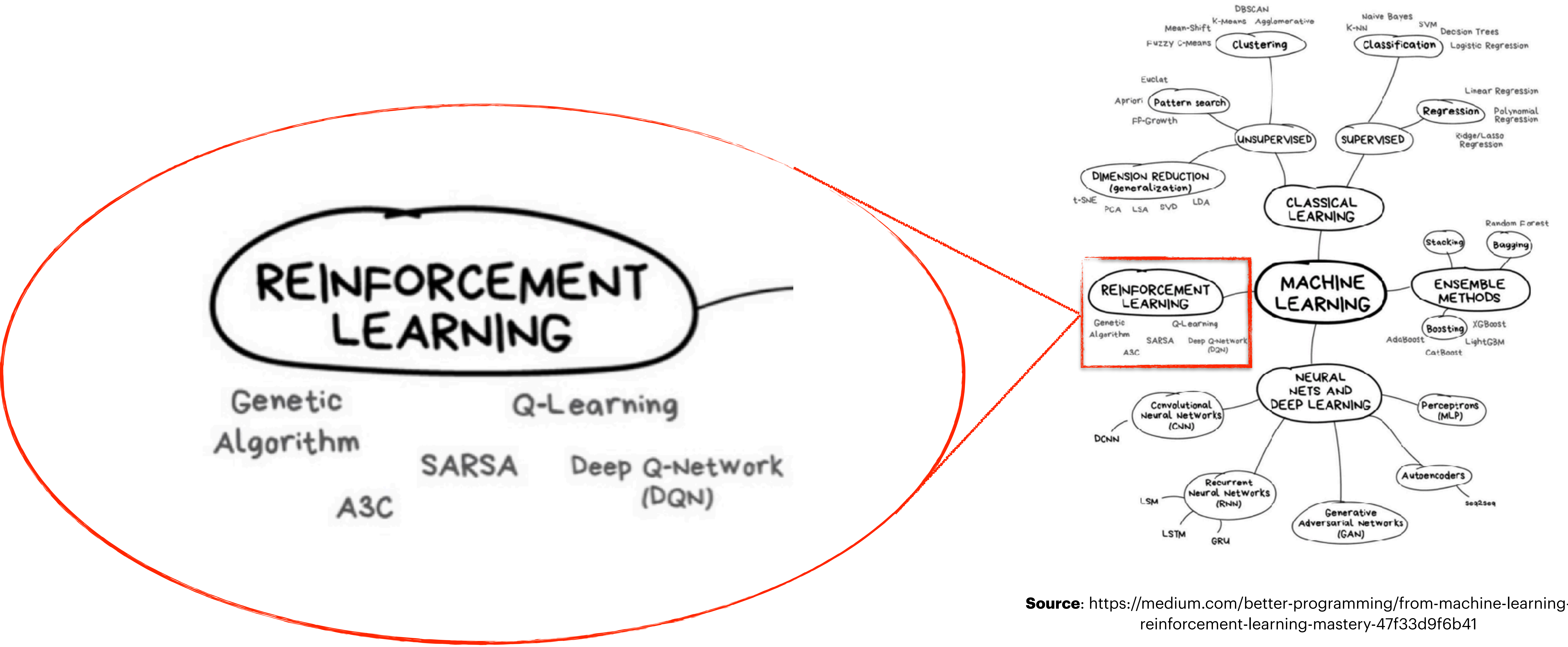
Source: https://medium.com/better-programming/from-machine-learning-to-reinforcement-learning-mastery-47f33d9f6b41

ML Landscape



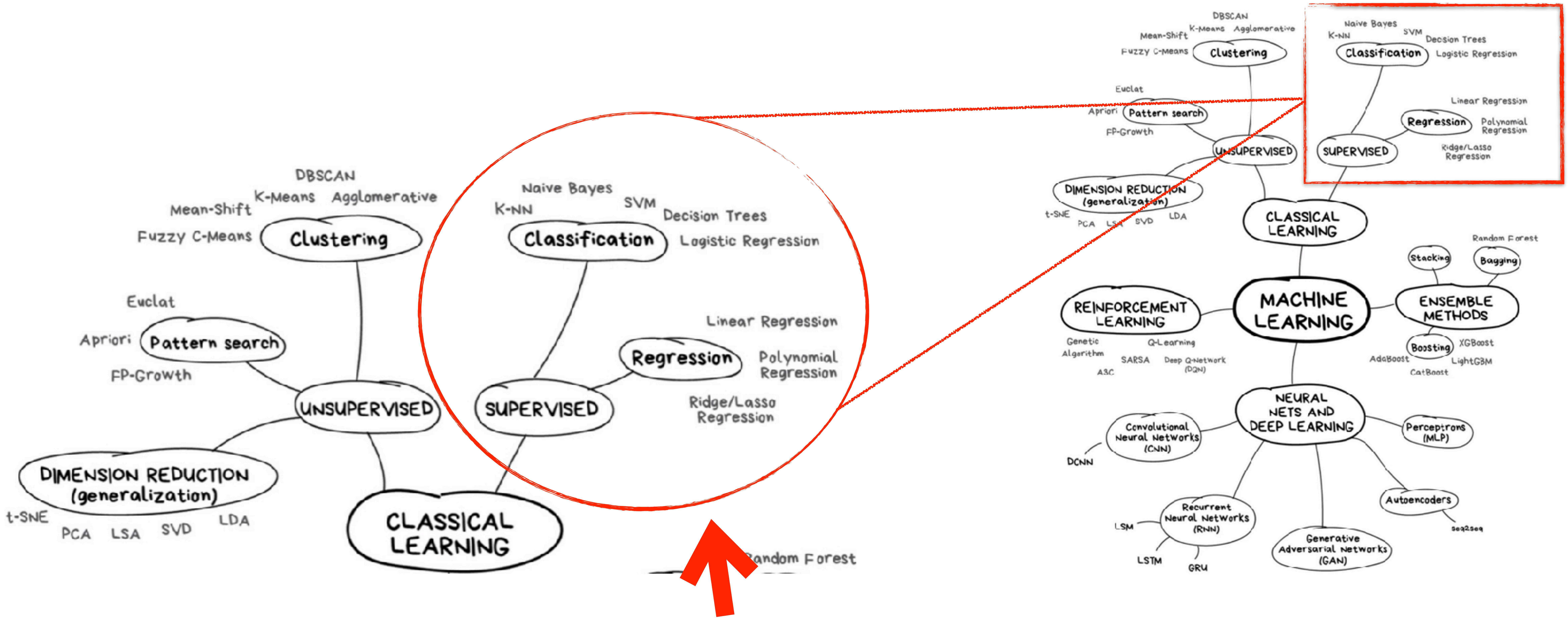
Source: https://medium.com/better-programming/from-machine-learning-to-reinforcement-learning-mastery-47f33d9f6b41

ML Landscape



Source: https://medium.com/better-programming/from-machine-learning-to-reinforcement-learning-mastery-47f33d9f6b41

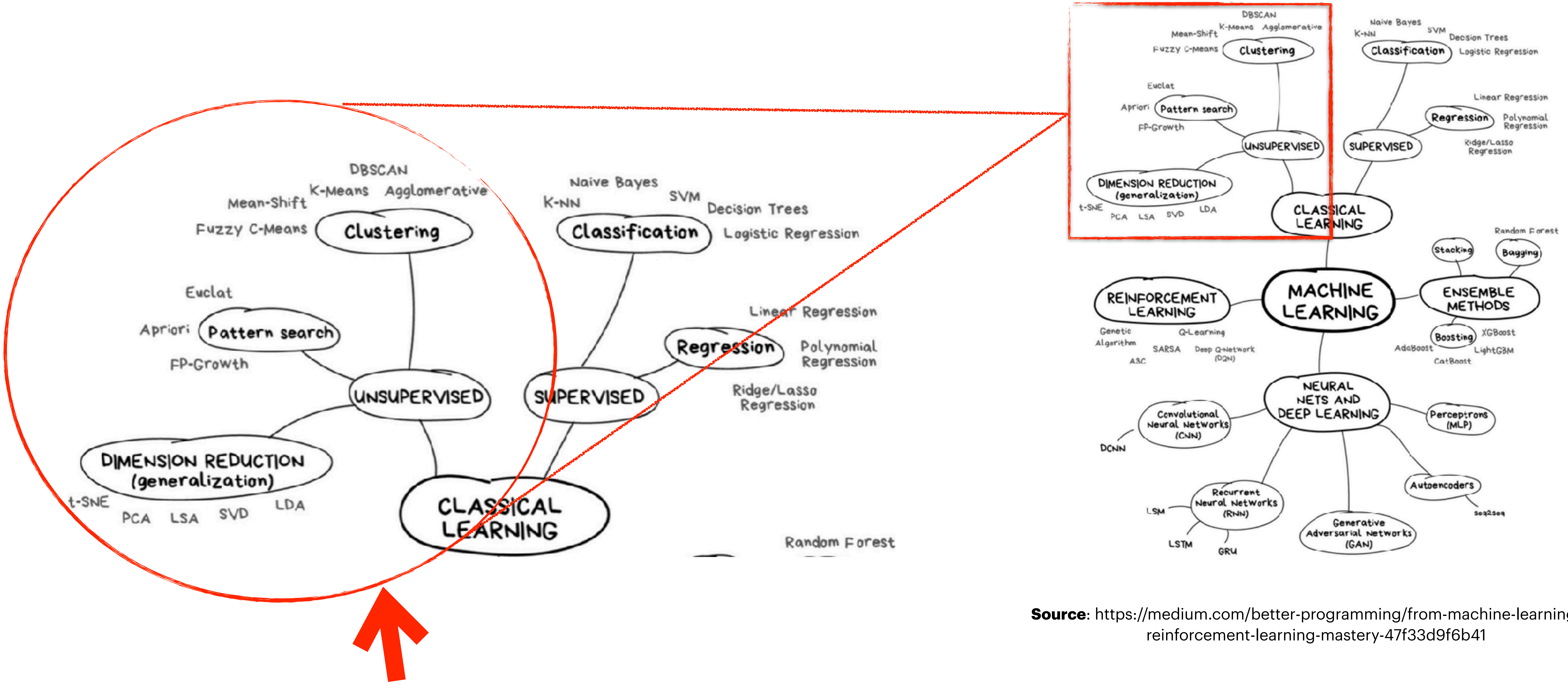
ML Landscape



This is about **Model Approximation**

Source: https://medium.com/better-programming/from-machine-learning-to-reinforcement-learning-mastery-47f33d9f6b41

ML Landscape



Source: https://medium.com/better-programming/from-machine-learning-to-reinforcement-learning-mastery-47f33d9f6b41

This is about **Pattern Description**

Classes of ML Problems

Supervised Learning

Goal: Maps inputs to outputs

This is a **mango**.
Mangoes **can be eaten**.
Birds eat mangoes.

Unsupervised Learning

Goal: Learn the underlying data patterns.

Looks like a mango.
Birds also eat it.
Maybe **I can eat it**.

Reinforcement Learning

Goal: Maximize future rewards over many time-steps.

When I eat peach, **i feel strong**.
I will **continue eating** peach.

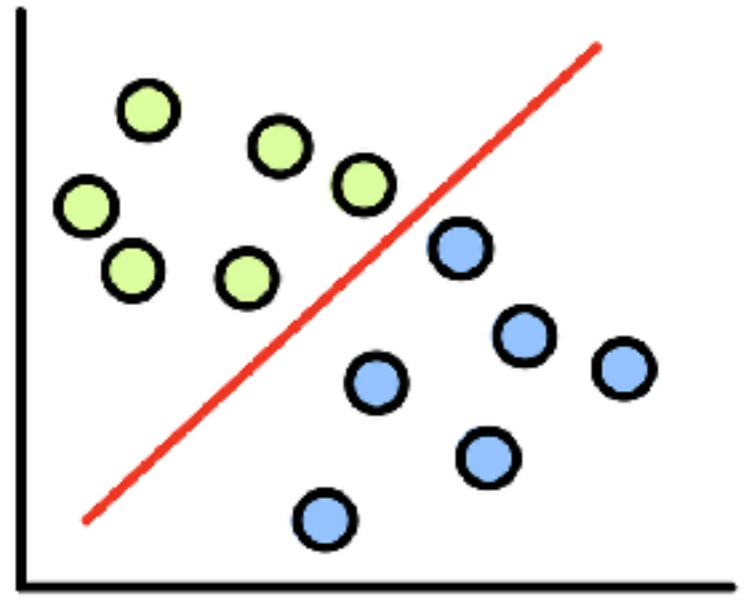
Classes of ML Problems

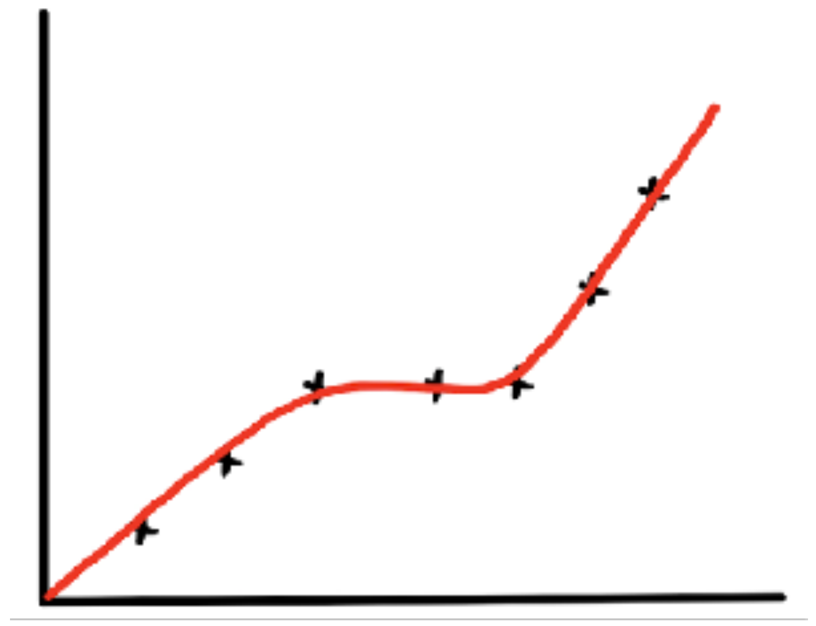
Supervised Learning

Data: (X, Y)

X is a set of input features, Y is a set of output values, labels or scalars.

Goal: $X \to Y$



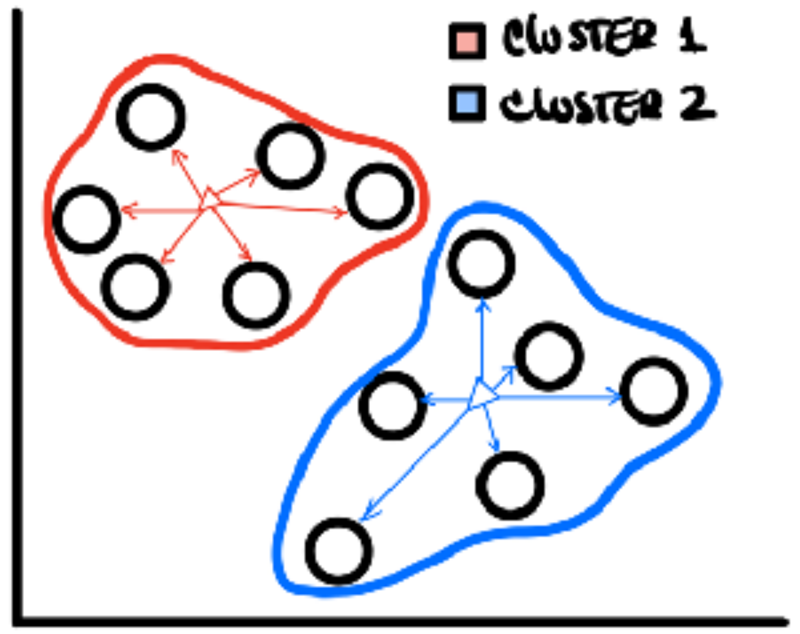


Unsupervised Learning

Data: X

X is a set of input features.

Goal: Learn the underlying data patterns.

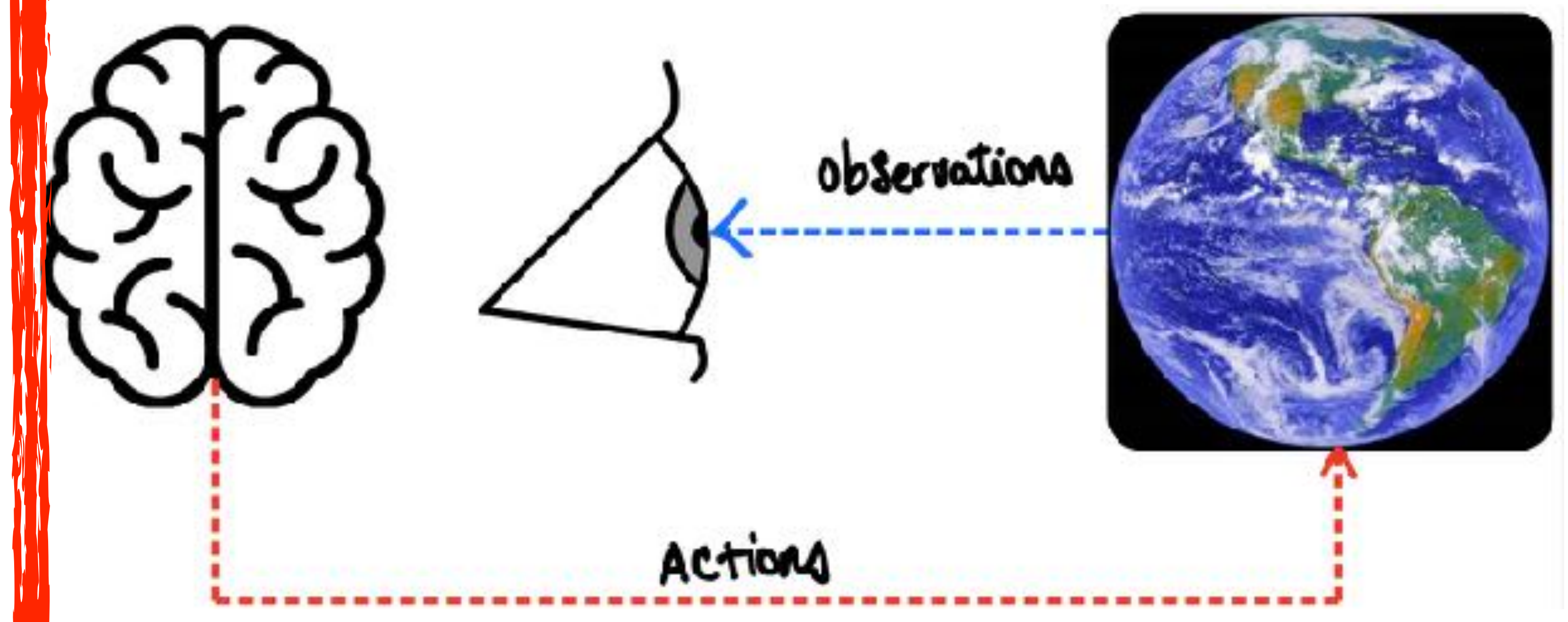


Reinforcement Learning

Data: (S, A)

S is a set of states, A is a set of actions.

Goal: Maximize future rewards over many time-steps.



2. Unsupervised Learning Fundamentals

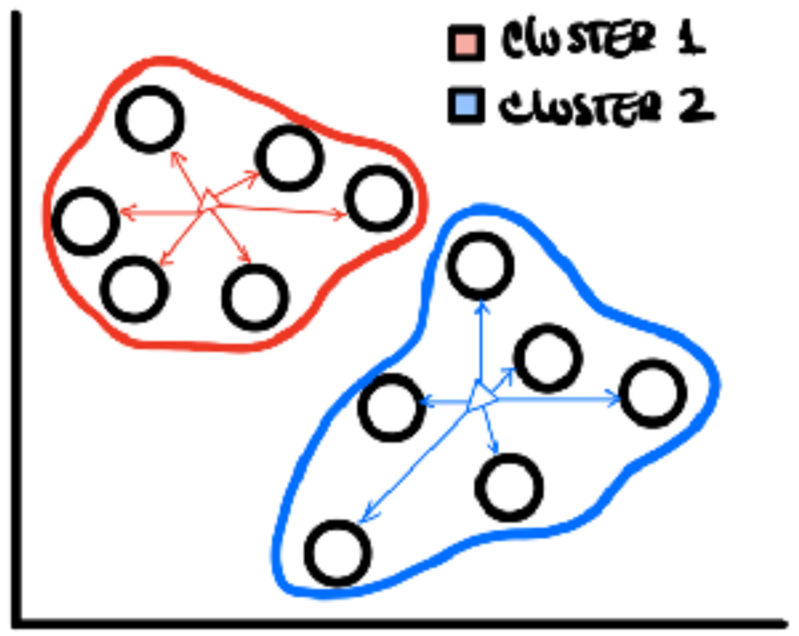
Unsupervised Learning Taxonomy

Clustering

Data: $X = \{x_1, x_2, ..., x_n\}$

X is a set of elements with a fixed feature size.

Goal: $X \rightarrow C_i$

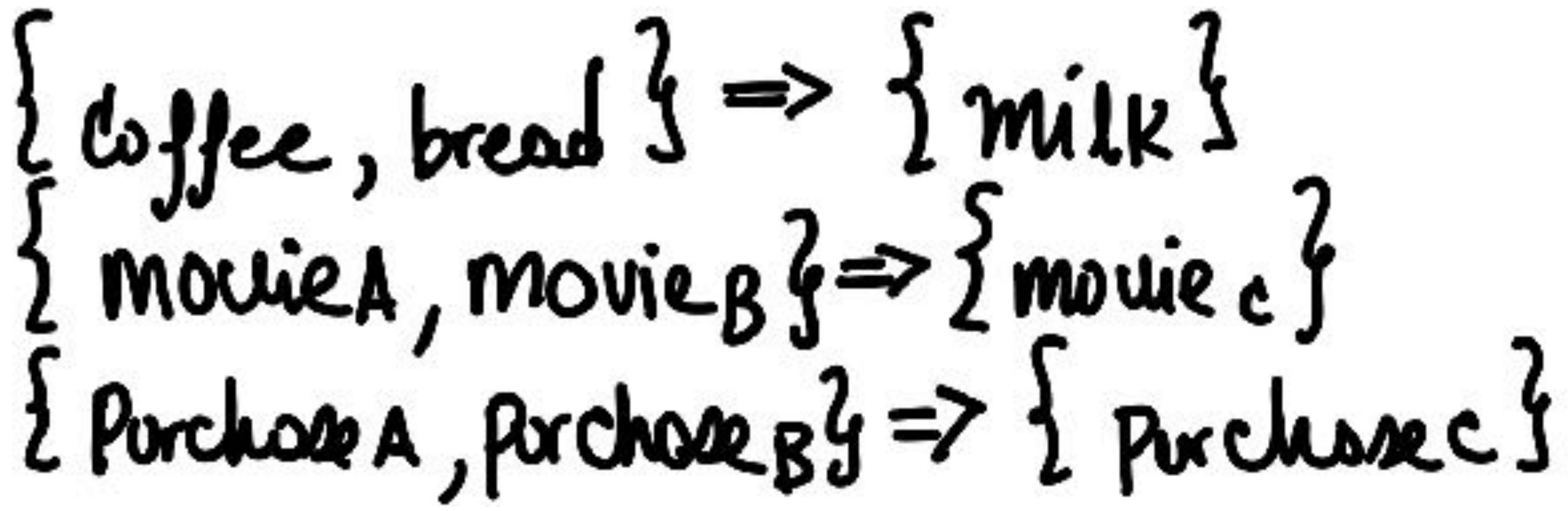


Pattern Search

Data: $X = \{t_1, t_2, ..., t_n\}$

X is a set of transactions composed of items $\in I = \{i_1, i_2, ..., i_m\}$.

Goal: $X \rightarrow Y \subseteq I$

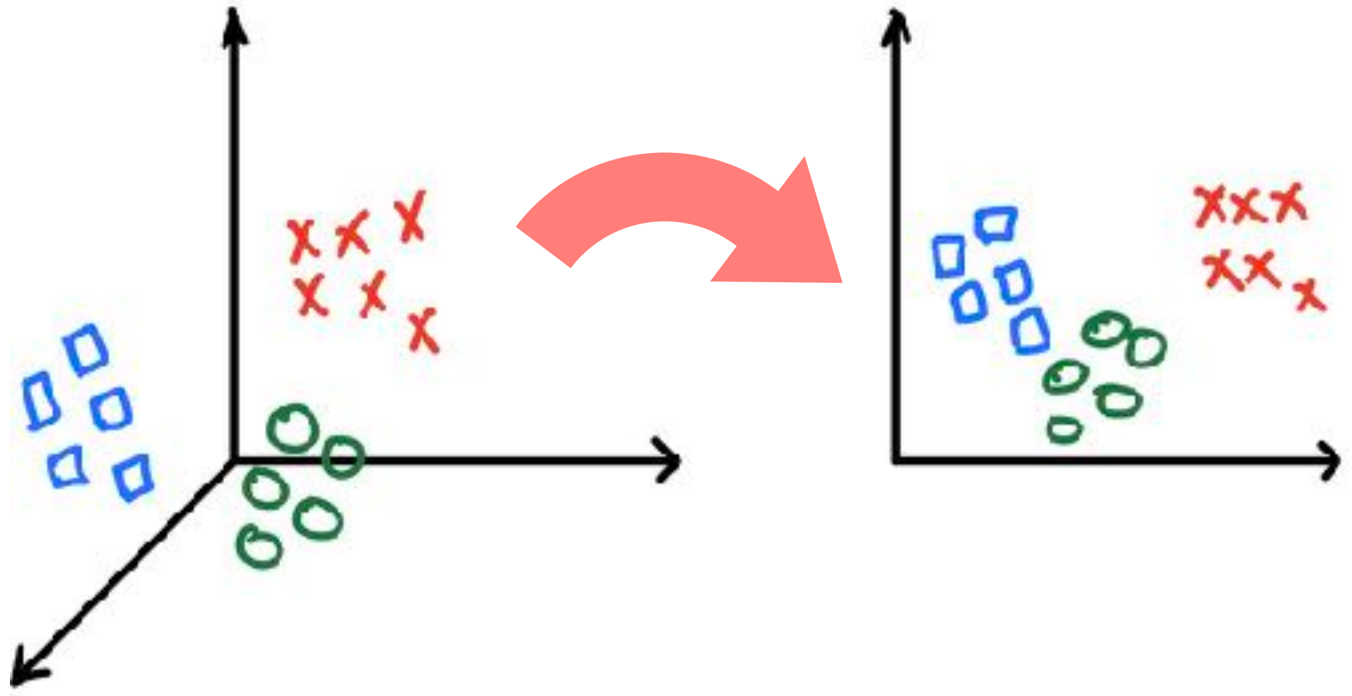


Dimensionality Reduction

Data: X

X is a set of elements with a fixed feature size.

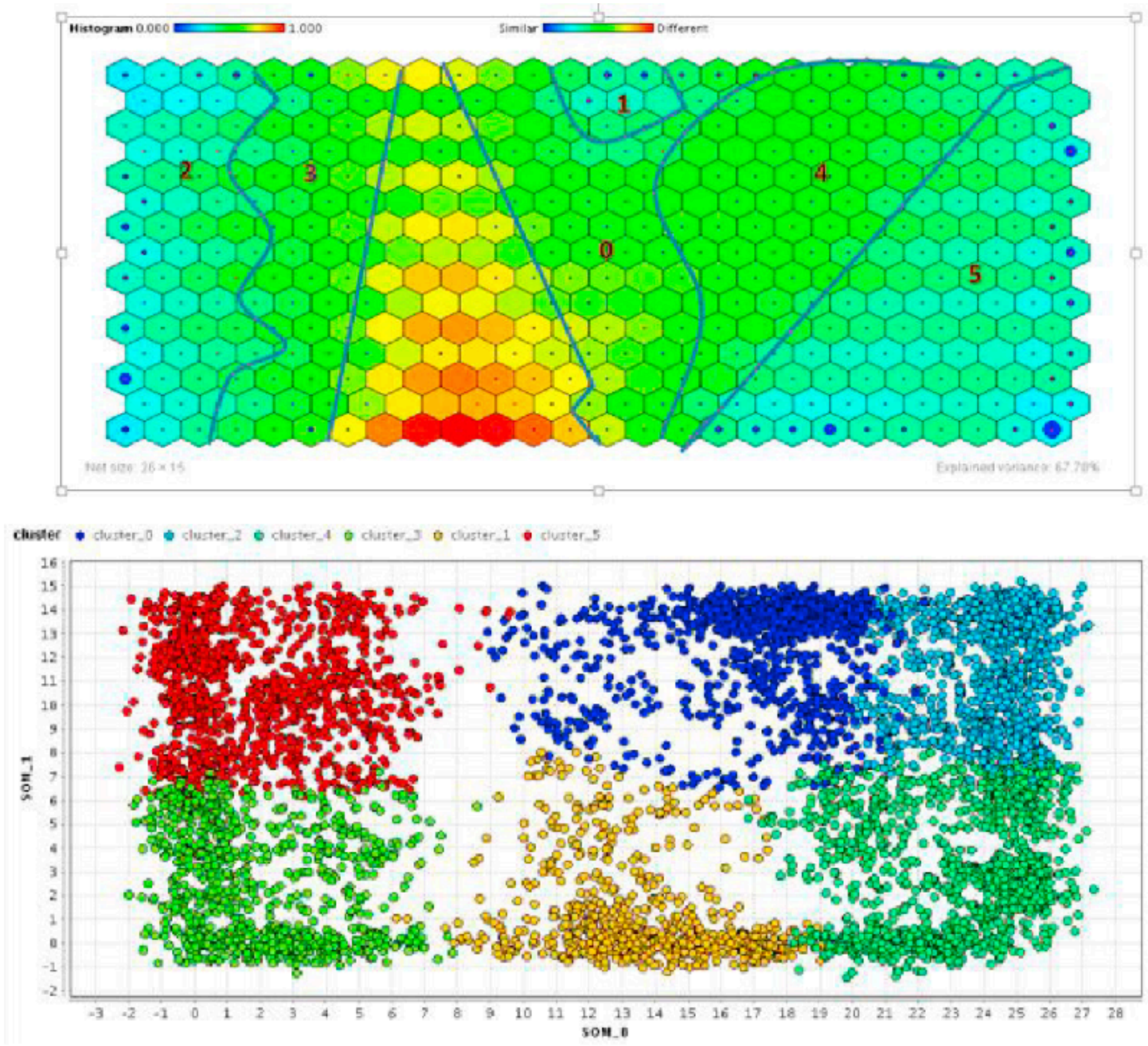
Goal: $X \rightarrow F(X) = X'$ where $\mathbb{R}^l > \mathbb{R}^m$ for $X \in \mathbb{R}^l, X' \in \mathbb{R}^m$



Applications

Customer segmentation

- Use of Self Organizing Maps (SOM) and K-Means to find user groups with high similarity.
- Design of targeted marketing campaigns.
- Support decision taking through cluster interpretation.
- Data can be static (customer-features) or behavioral (Netflix).

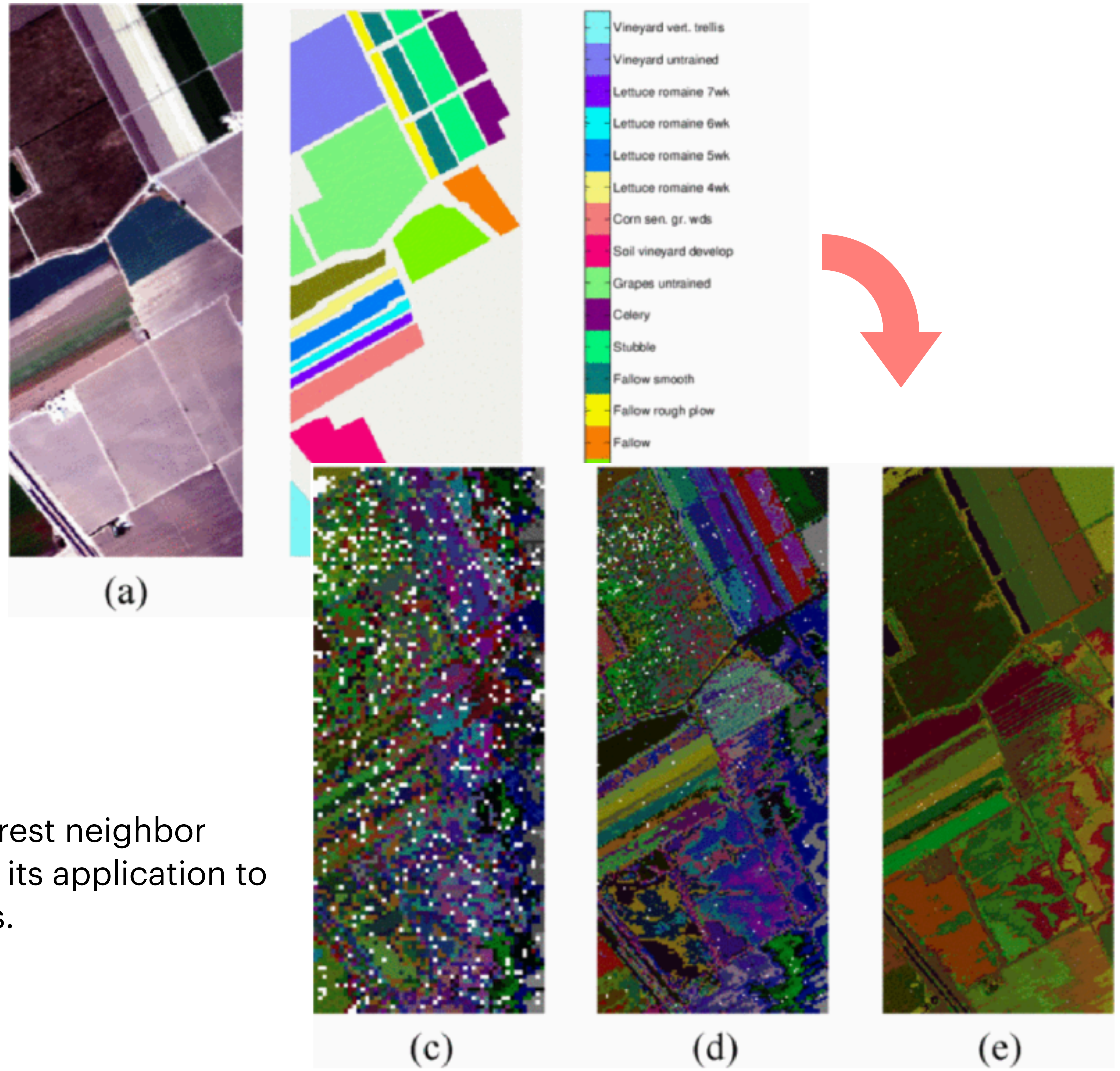


(Qadadeh et. al., 2019). Customer segmentation in the Insurance Company (TIC) Dataset.

Applications

Remote Sensing Image Analysis

- Segmentation of crop types.
- Automatic terrain elements detection: road, river, crop, etc.
- Hyperspectral imaging == massive data.

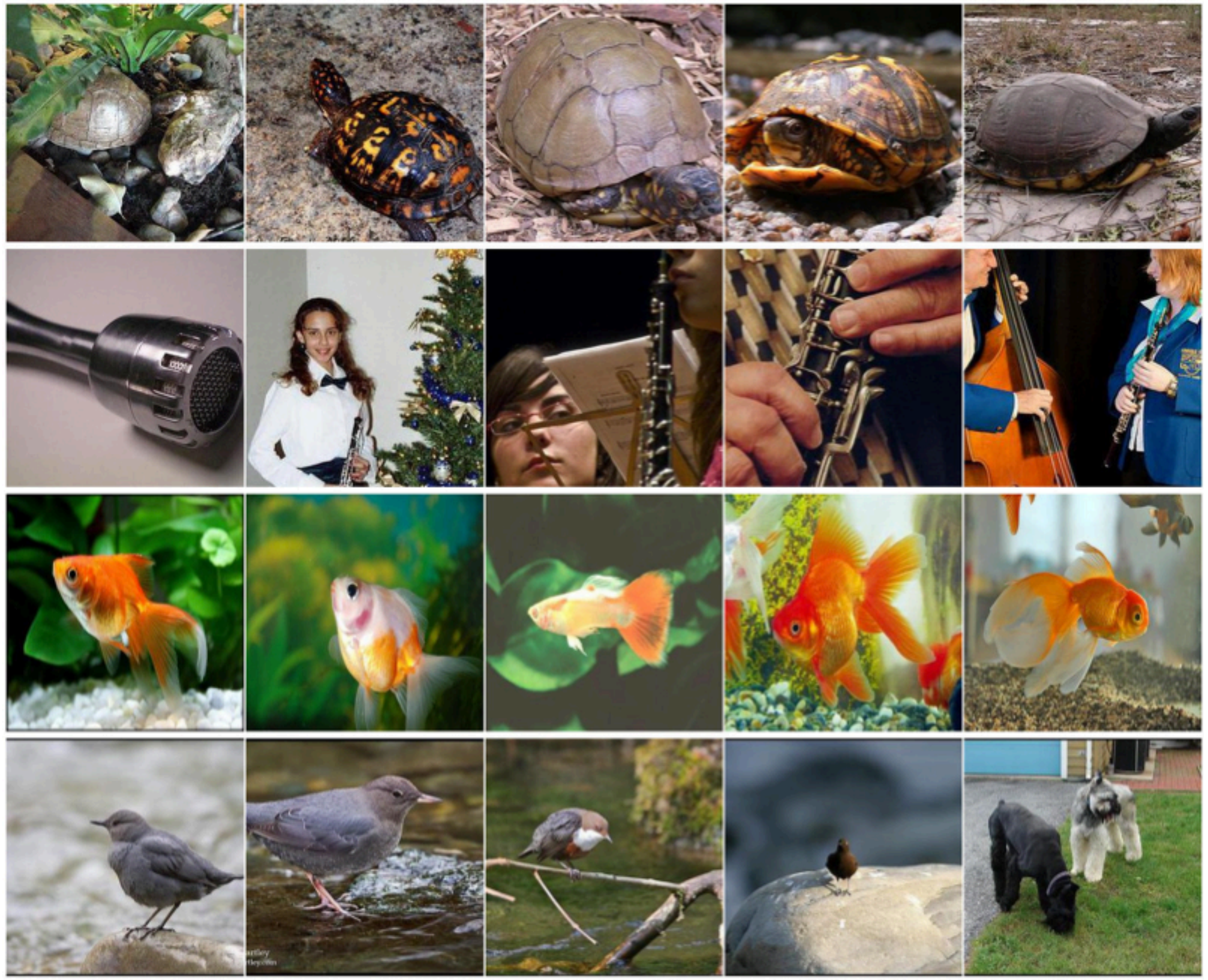


(Cariou et. al., 2016). A new k-nearest neighbor density-based clustering method and its application to hyperspectral images.

Applications

Unsupervised Image Classification

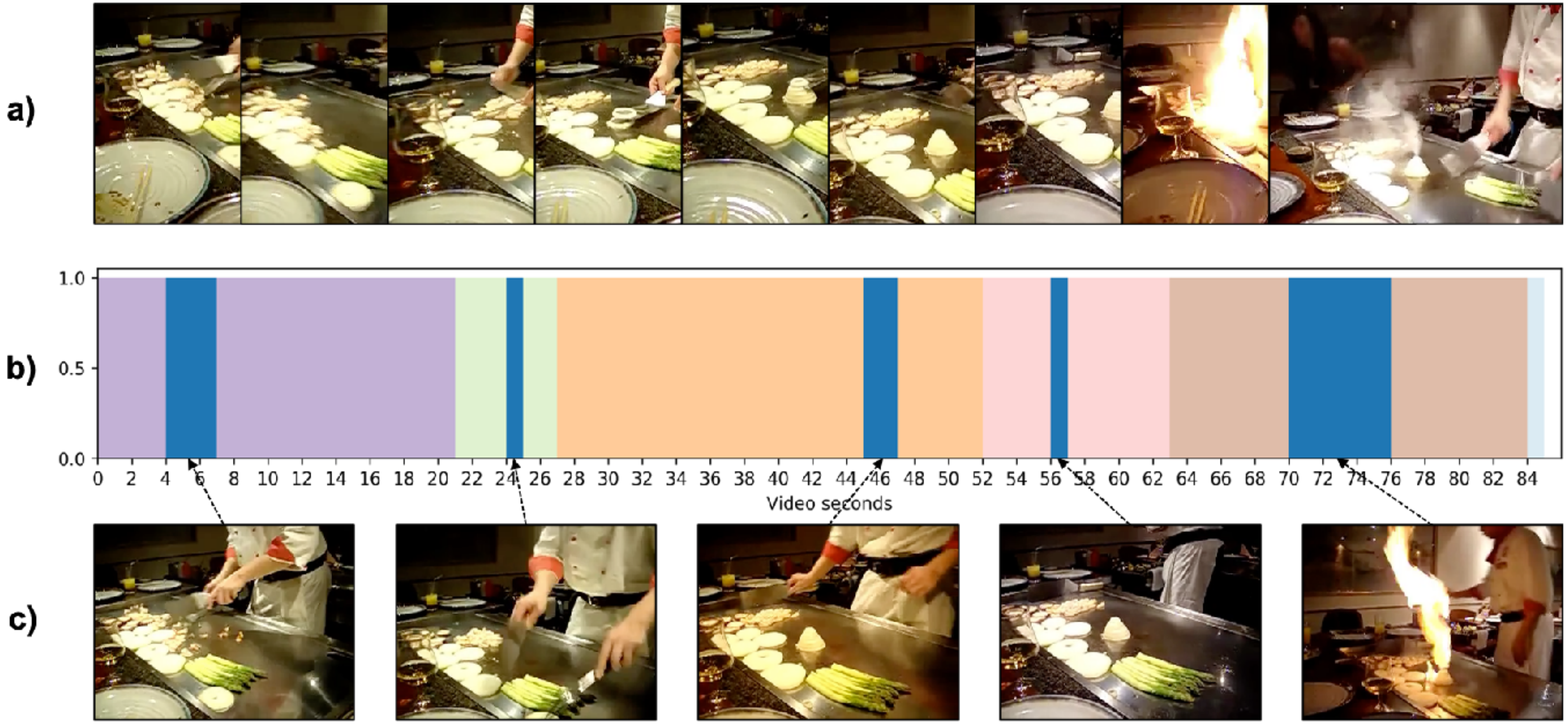
- Supervised annotations impose an important constraint in image classification.
- Authors proposed a combination of feature-learning and clustering approach.
- Self-supervised learning is used to learn image features.
- Cluster learning non dependent of low level features.



(Gansbeke et. al., 2020). SCAN: Learning to Classify Images without Labels.

Applications

Text / Video Summarization



(Atencio et. al., 2020). Query-based Video Summarization Using Machine Learning and Coordinated Representations.

3. Clustering

Clustering

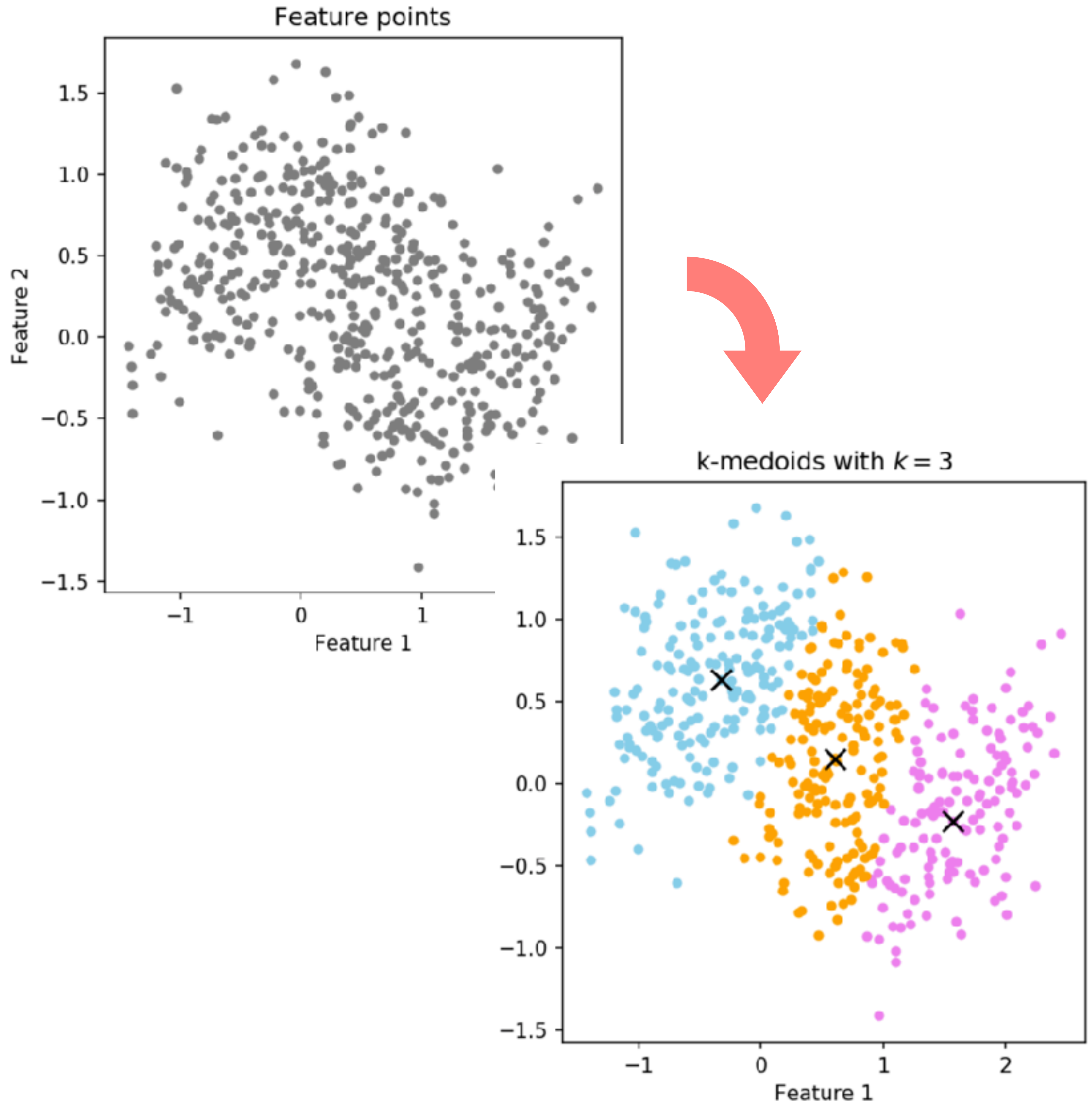
Input data

Clusters

Clustering

What is clustering

- The grouping of unlabeled data $X = \{x_1, x_2, \ldots, x_n\}$ into a similar sets or clusters.
- A cluster is a collection of data elements with high similarity or close distance in some feature space.
- Each cluster is expected to have high inter-variance and low intra-variance.



Clustering

Distance

- A dissimilarity measure between a pair of elements x_i and x_j
- Common used distance metrics are: *Manhattan*, *Euclidean* and *Cosine*

Manhattan

$$d(x^{[1]}, x^{[2]}) = \sum_{i=1}^{n} |x_i^{[1]} - x_i^{[2]}|$$

Euclidean

$$d(x^{[1]}, x^{[2]}) = \sqrt{\sum_{i=1}^{n} (x_i^{[1]}) - x_i^{[2]})^2}$$

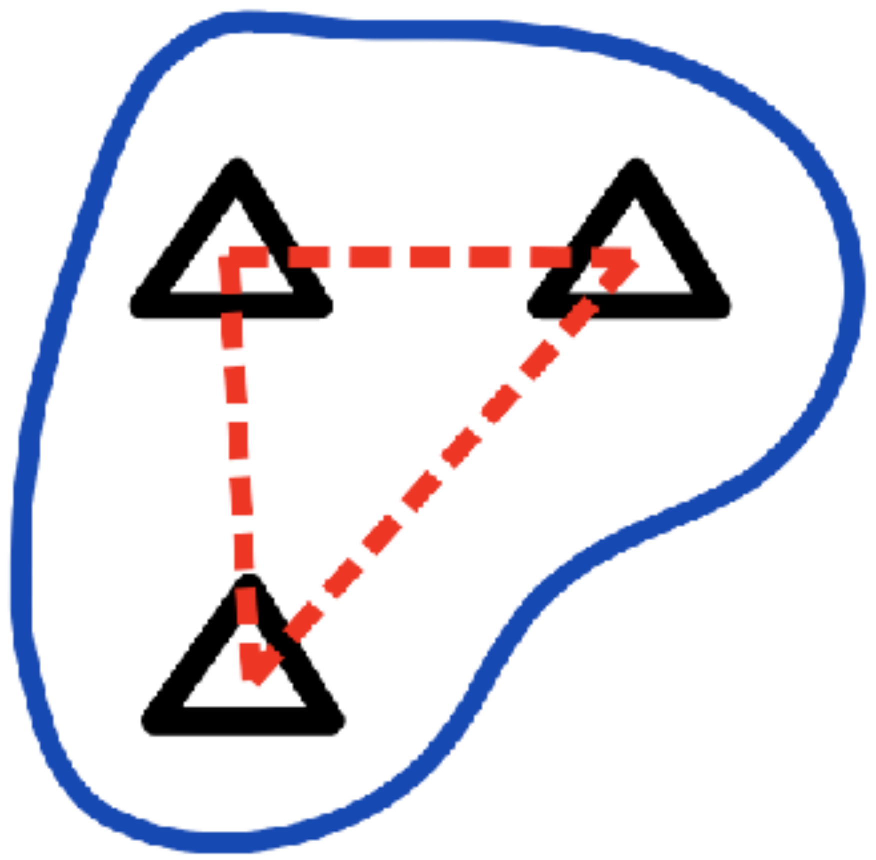
Cosine

$$cos\theta = \frac{x^{[1]} \cdot x^{[2]}}{\| x^{[1]} \| \ \| x^{[2]} \|}$$

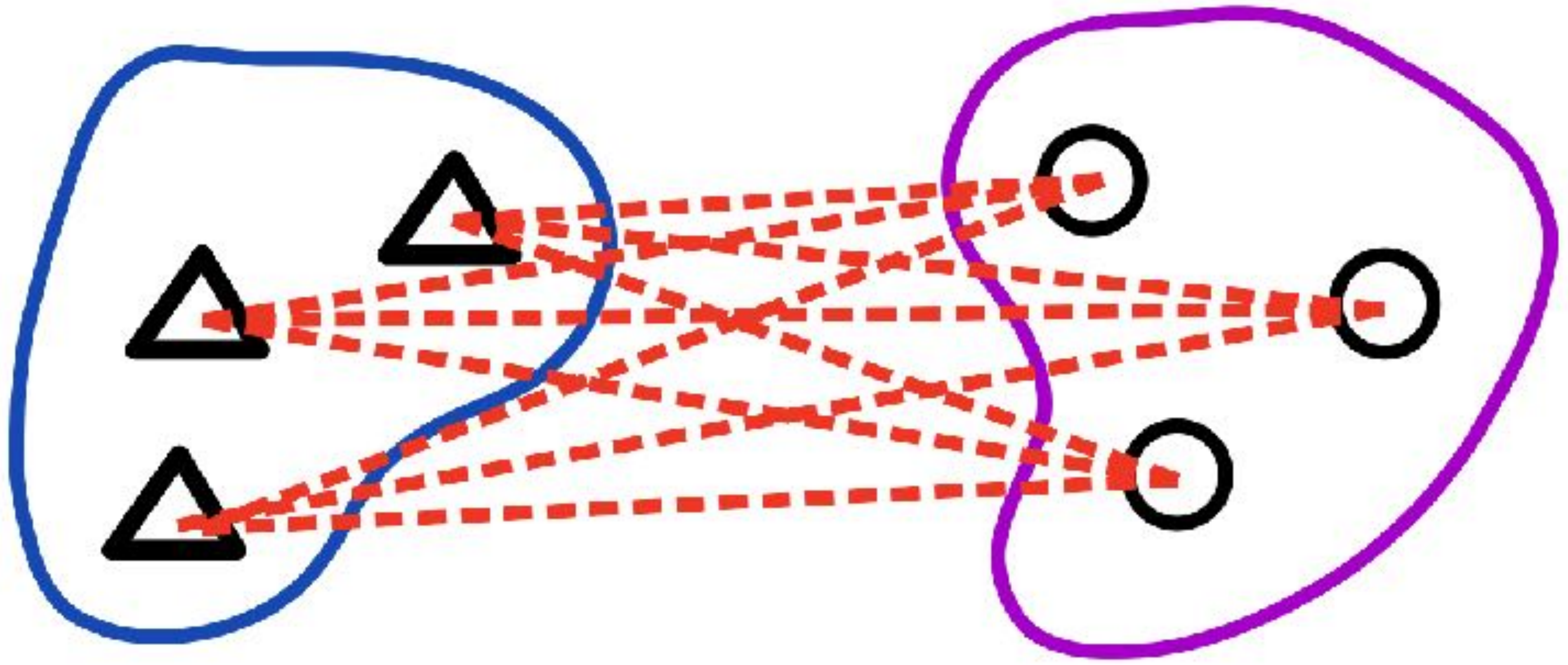
Clustering

Cluster quality

- **Intra-cluster cohesion (compactness)**:
  - How close are samples inside each cluster?
  - How close are samples to the cluster centroid?



- **Inter-cluster separation (isolation)**:
  - How far are samples from different clusters?



Clustering

Cluster quality

- **Intra-cluster cohesion (compactness)**:
  - How close are samples inside each cluster?
  - How close are samples to the cluster centroid?

$$SSE(C_i) = \sum_{x \in C_i} d(C_i, x)^2 = \frac{1}{2m} \sum_{x \in C_i} \sum_{y in C_i} d(x, y)^2$$

Sum of Squared Errors

- **Inter-cluster separation (isolation)**:
  - How far are samples from different clusters?

$$SSB = \frac{1}{2K} \sum_{i=1}^{K} \sum_{j=1}^{K} \frac{m}{K} d(c_i, c_j)^2$$

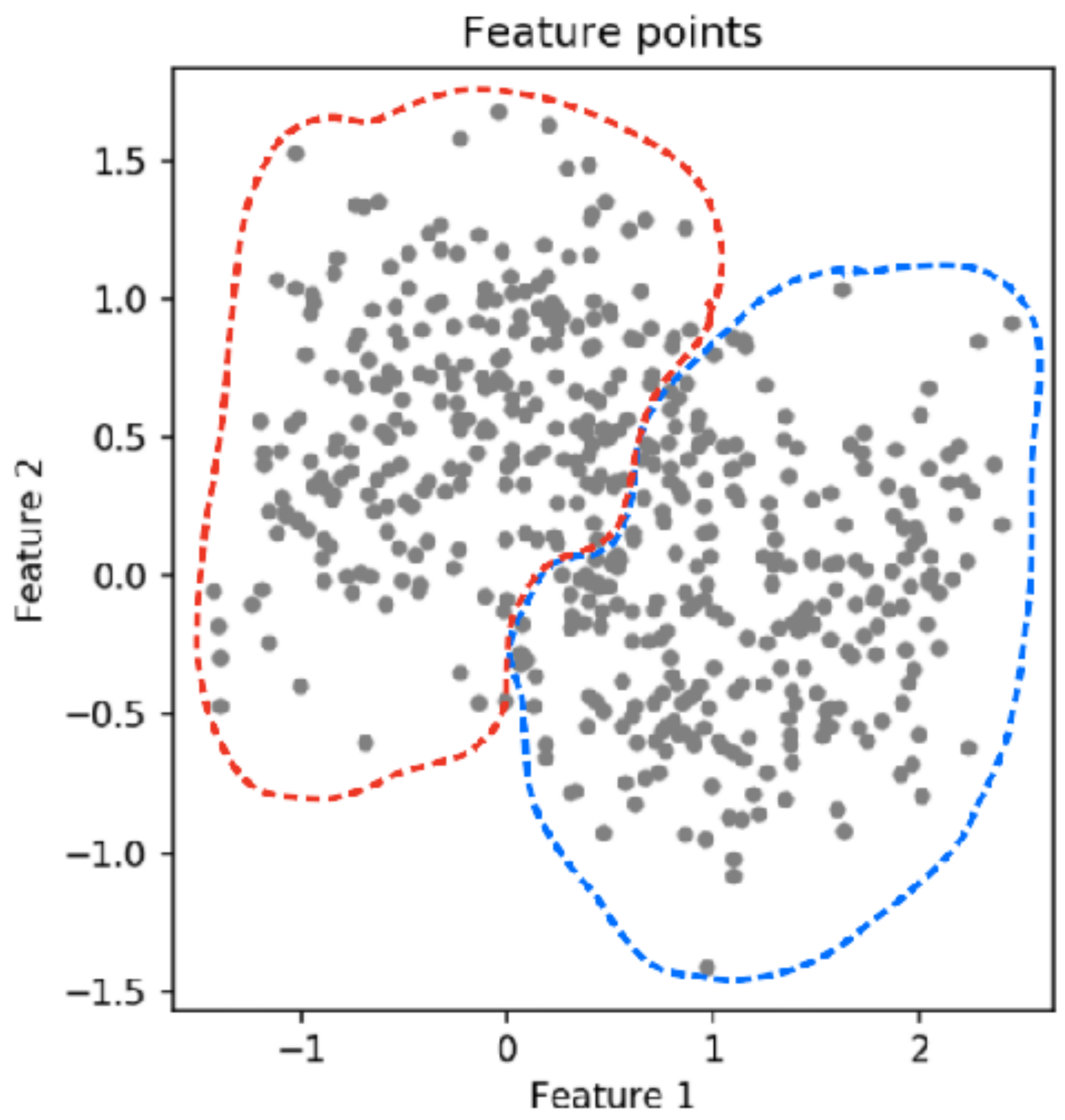
Sum of Squares Between groups

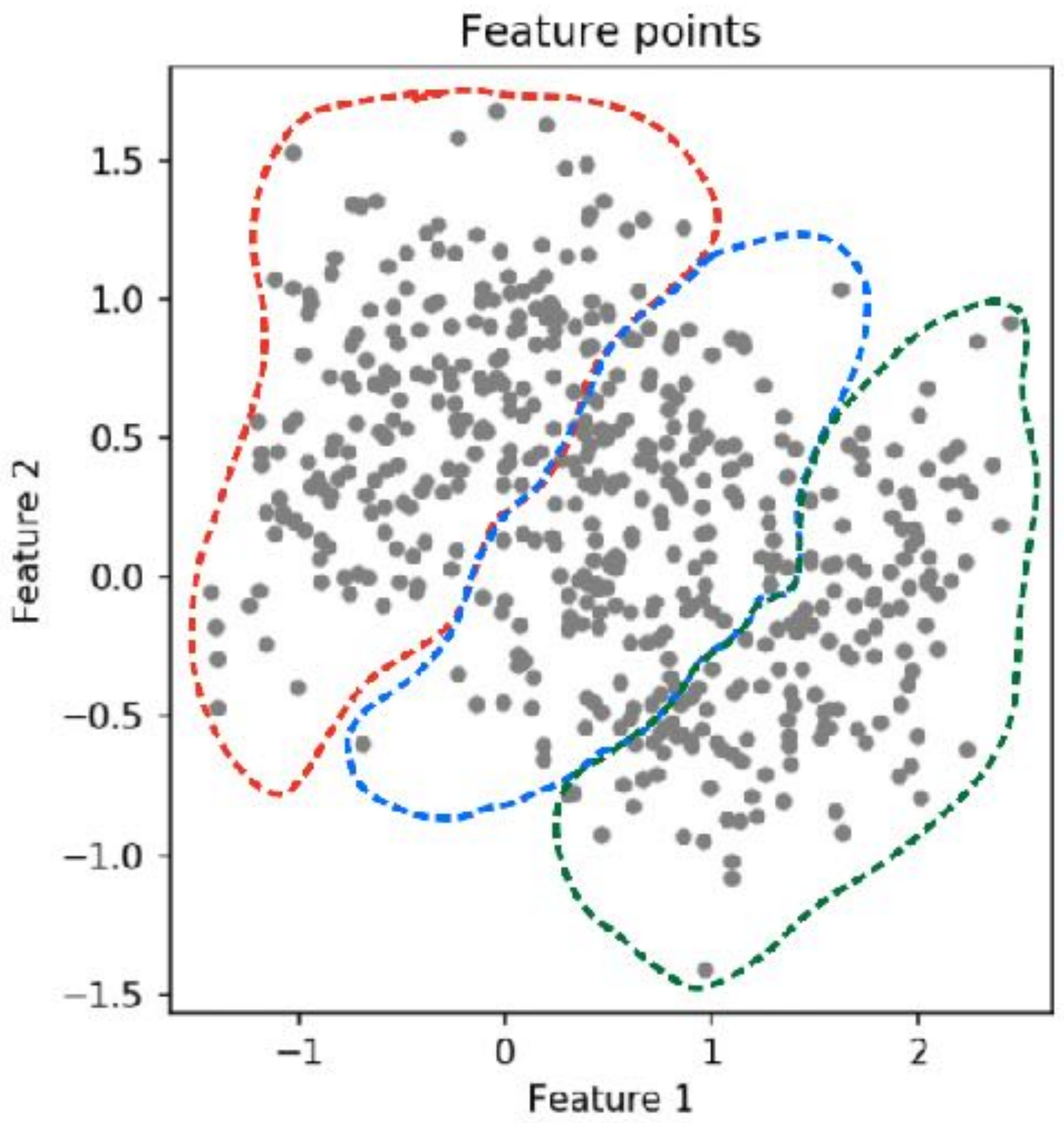
(Palacio and Berzal, 2019). Evaluation Metrics for Unsupervised Learning Algorithms.

Clustering

How many clusters?

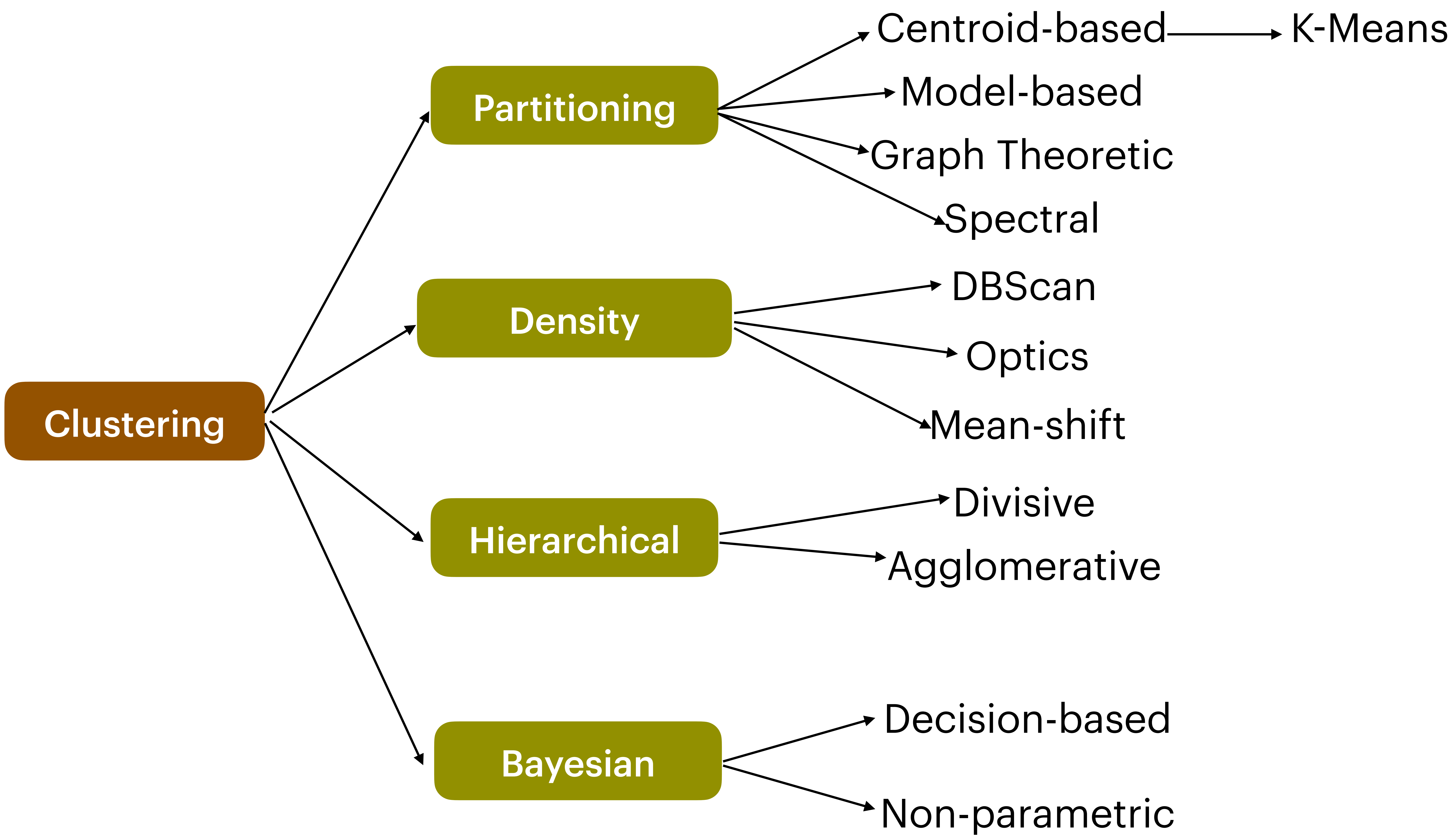
- Problem-dependent.
- **Approach 1)** To fix the number of clusters.
- **Approach 2)** To find the best number of clusters according to a criterion and a search method.





Clustering

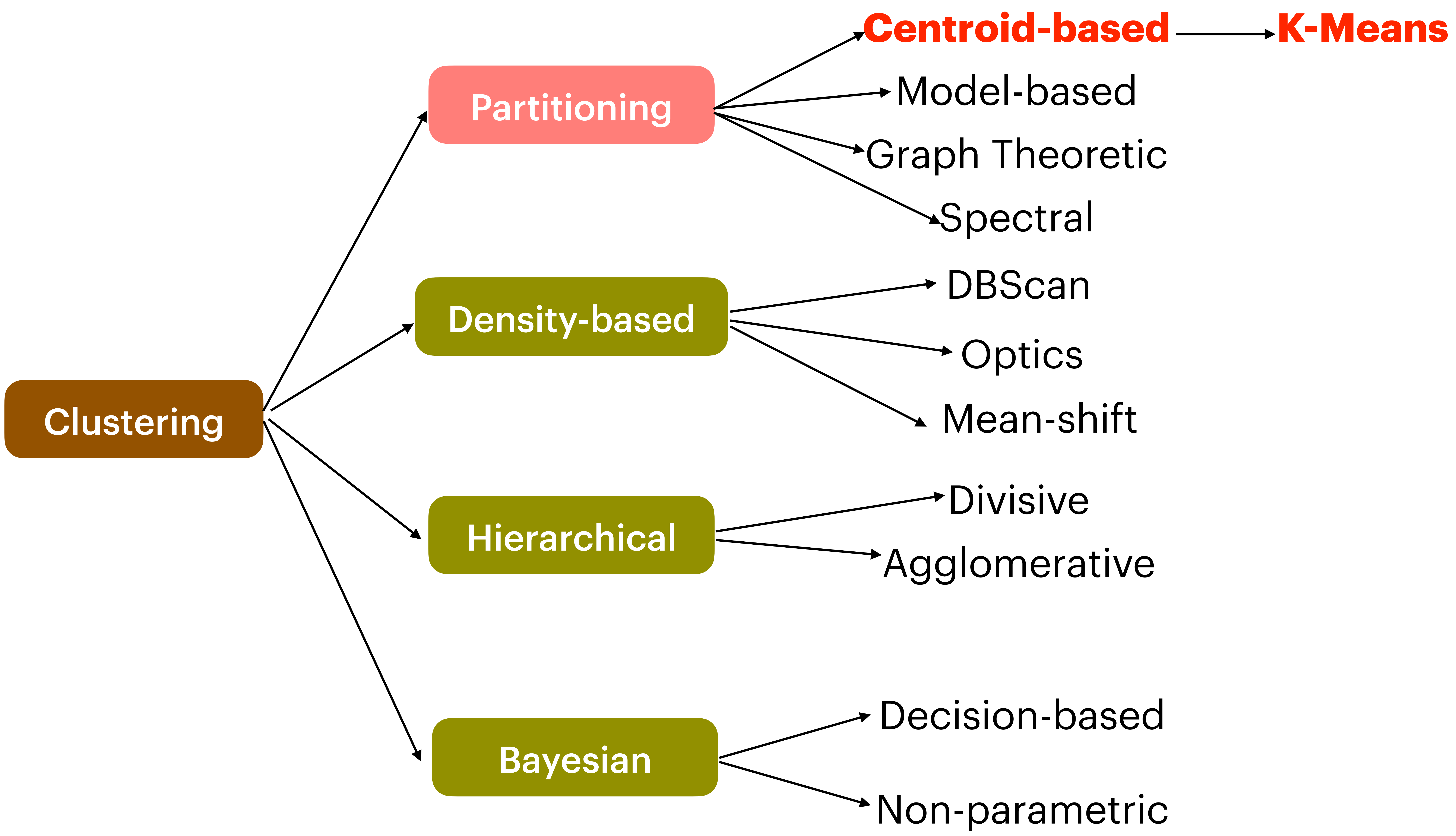
Taxonomy



3.1. Partitioning-based Clustering

Partitioning Based Clustering

Taxonomy



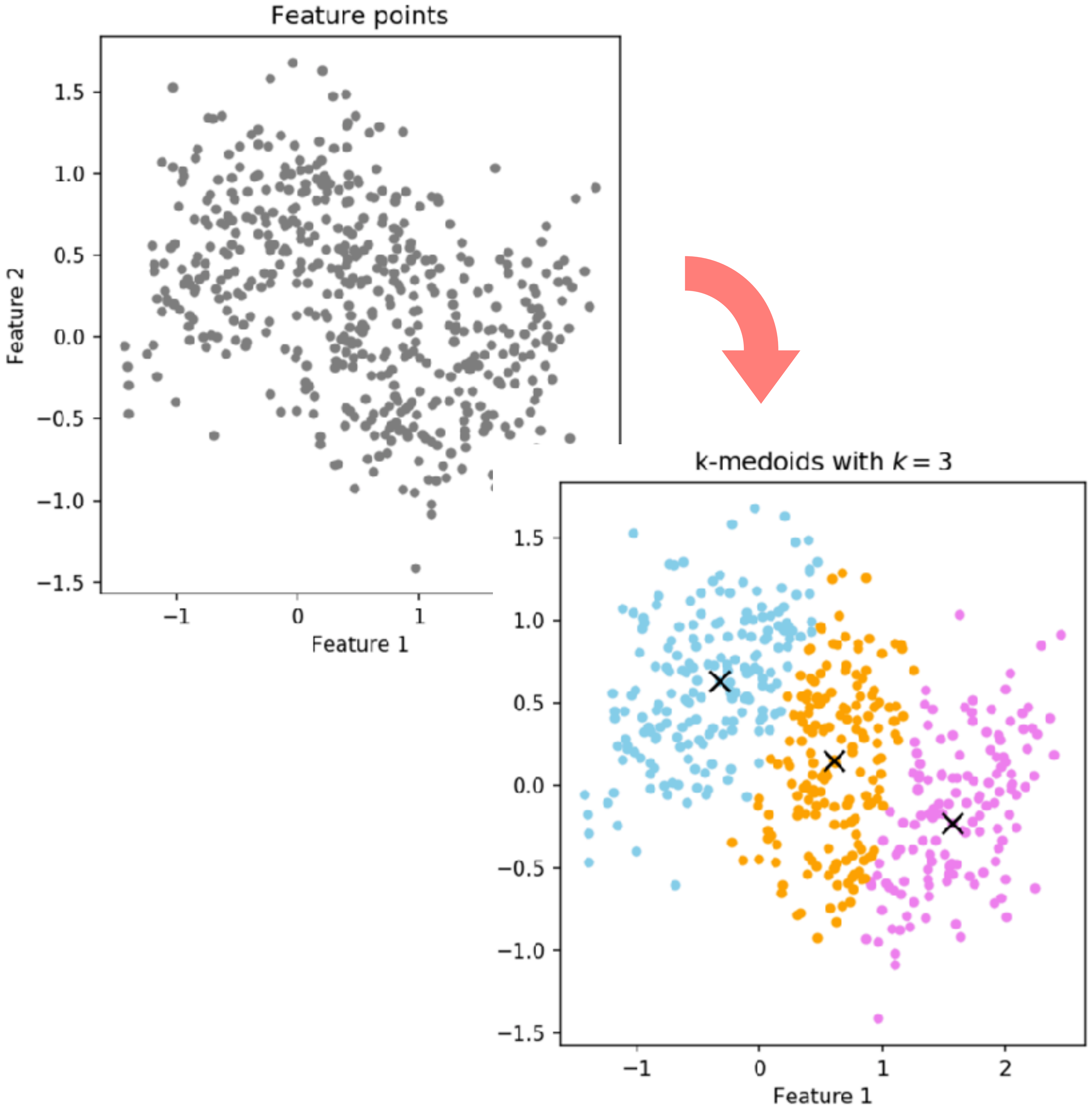
Partitioning Based Clustering

✓Divide the dataset into initial K clusters and iteratively improve the clustering quality based on cohesion or separation degree.

✓K-means is an example of a partitioning based clustering algorithm.

✓The objective function in K-means is the SSE.

✓Partitioning based algorithms are sensitive to initialization.



3.1.1. K-Means Clustering

Algorithm

1. Input Data $X = x_1, x_2, \ldots x_N$ and number of clusters K

2.Centroids $c_1, c_2, \ldots c_K$ = random K points of X

3.foreach data point x_i

- Compute distance $d_{ij} = d(x_i, c_j)$

  $i = \{1,...,N\}, j = \{1,\ldots, K\}$

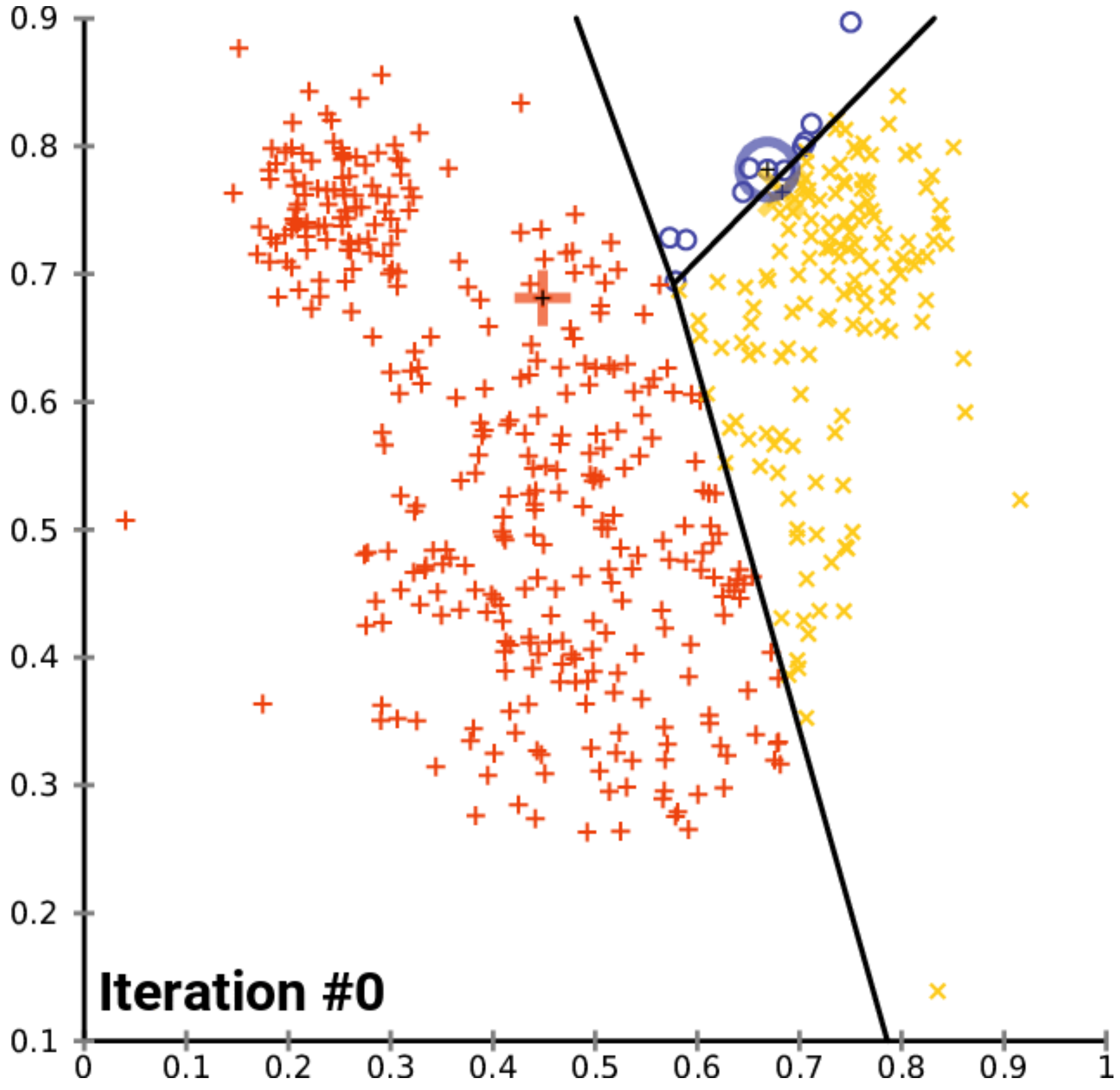
- Assign x_i to the nearest centroid: $y_i = argmin_j(d_{ij})$

4. Compute the new centroids of each cluster

   $c_j^* = mean(x_i)$ for $y_i = j$

5. if $c_j^* \neq c_j$ then $c_j = c_j^*$ goto step 3

6.Output: $c_1^*, c_2^*, \ldots c_K^*$ and y_i for $i = \{1,...,N\}$



Source: https://en.wikipedia.org/wiki/K-means_clustering

Algorithm

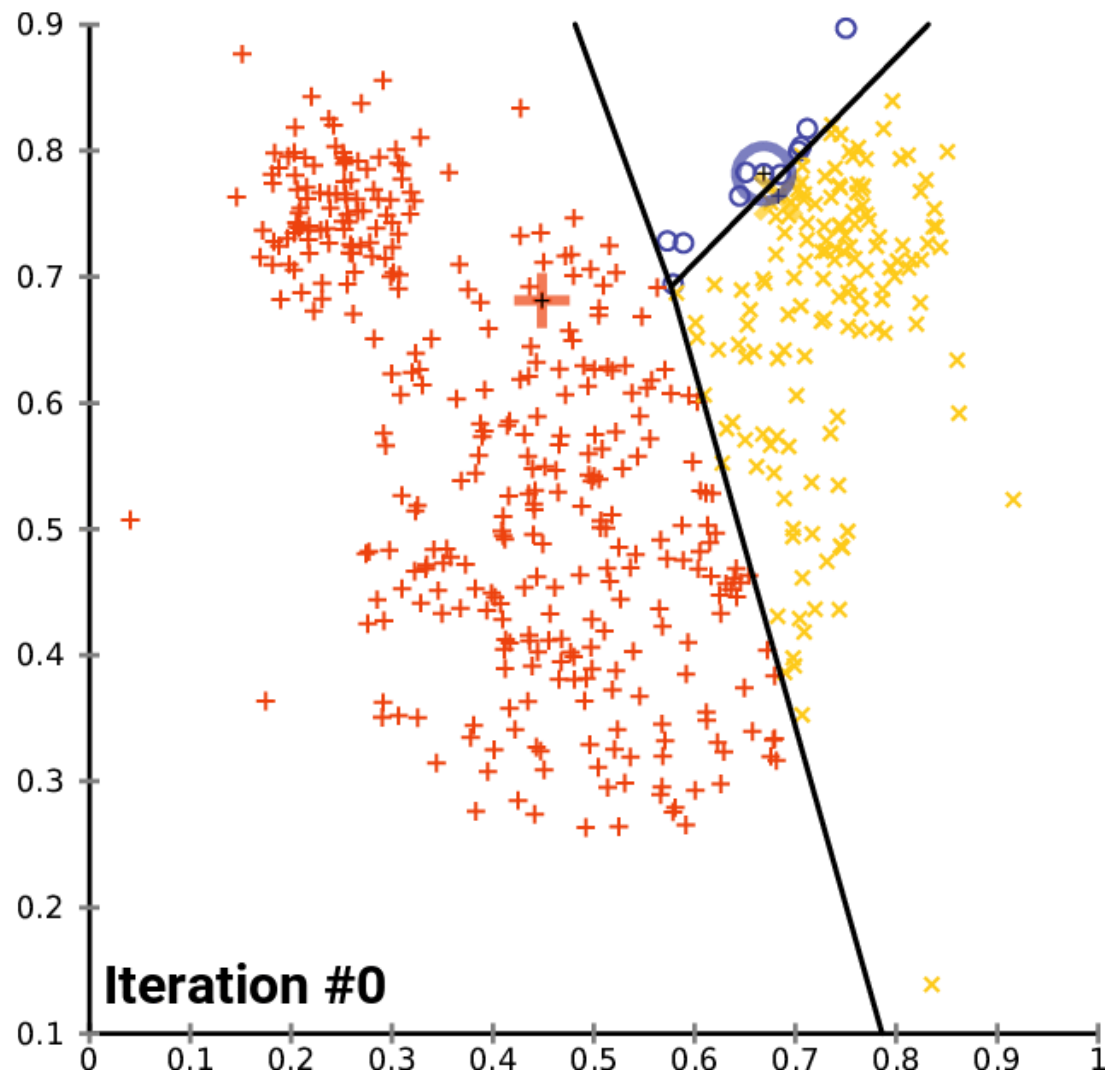
Stopping criteria

✓ No re-assignments of data $x_i \in X$ to different clusters.

✓ No change of centroids $c_i \in C$ values.

✓Minimum decrease in the SSE value or intra-cluster cohesion.

✓Minimum increase un the SSB value or intra-cluster separation.



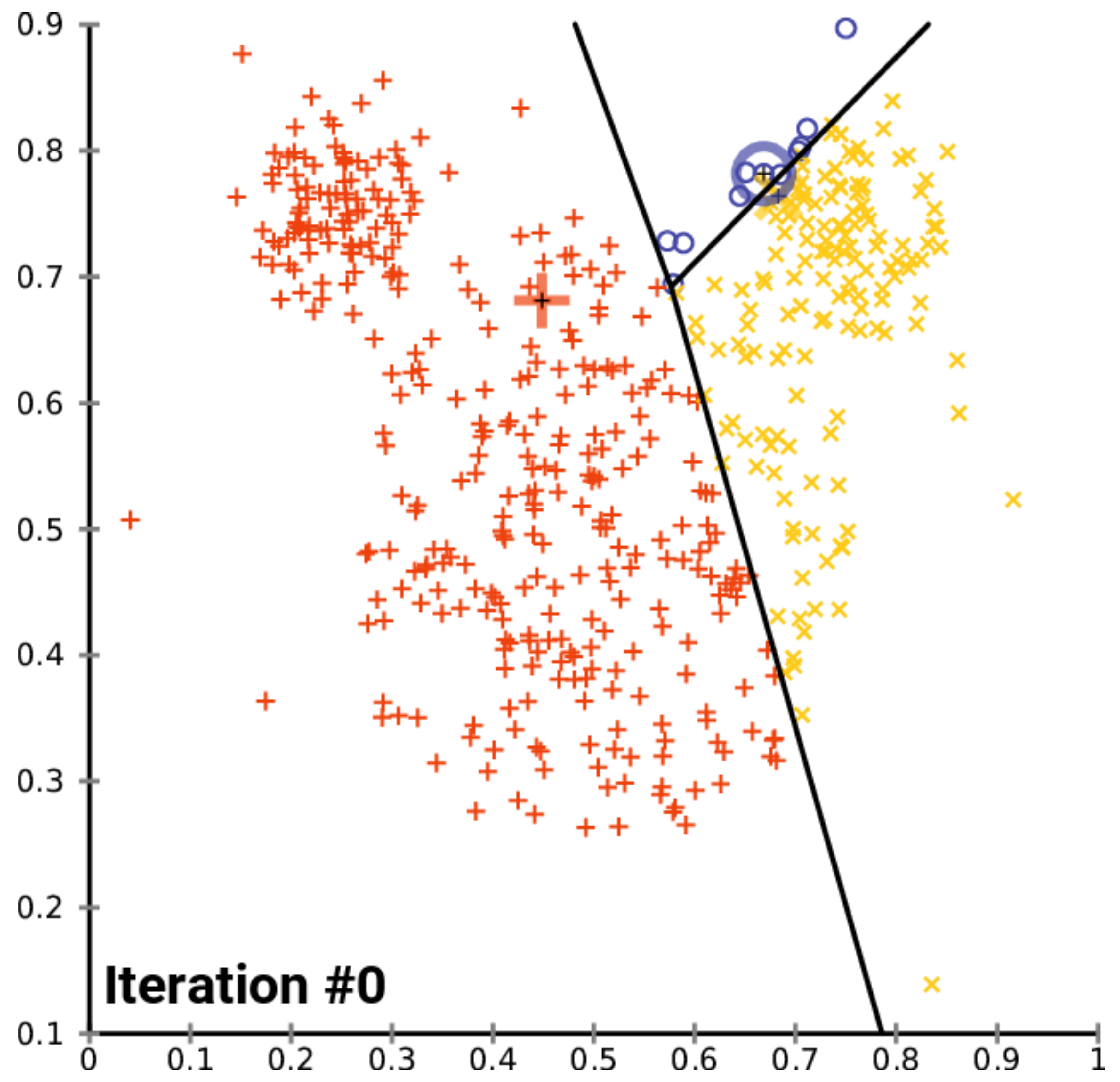
Source: https://en.wikipedia.org/wiki/K-means_clustering

Strengths

✓A simple algorithm. Simple Implementation.

✓Complexity $O(tkn)$ where t is the number of iterations, k the number of clusters, and n the number of data samples.

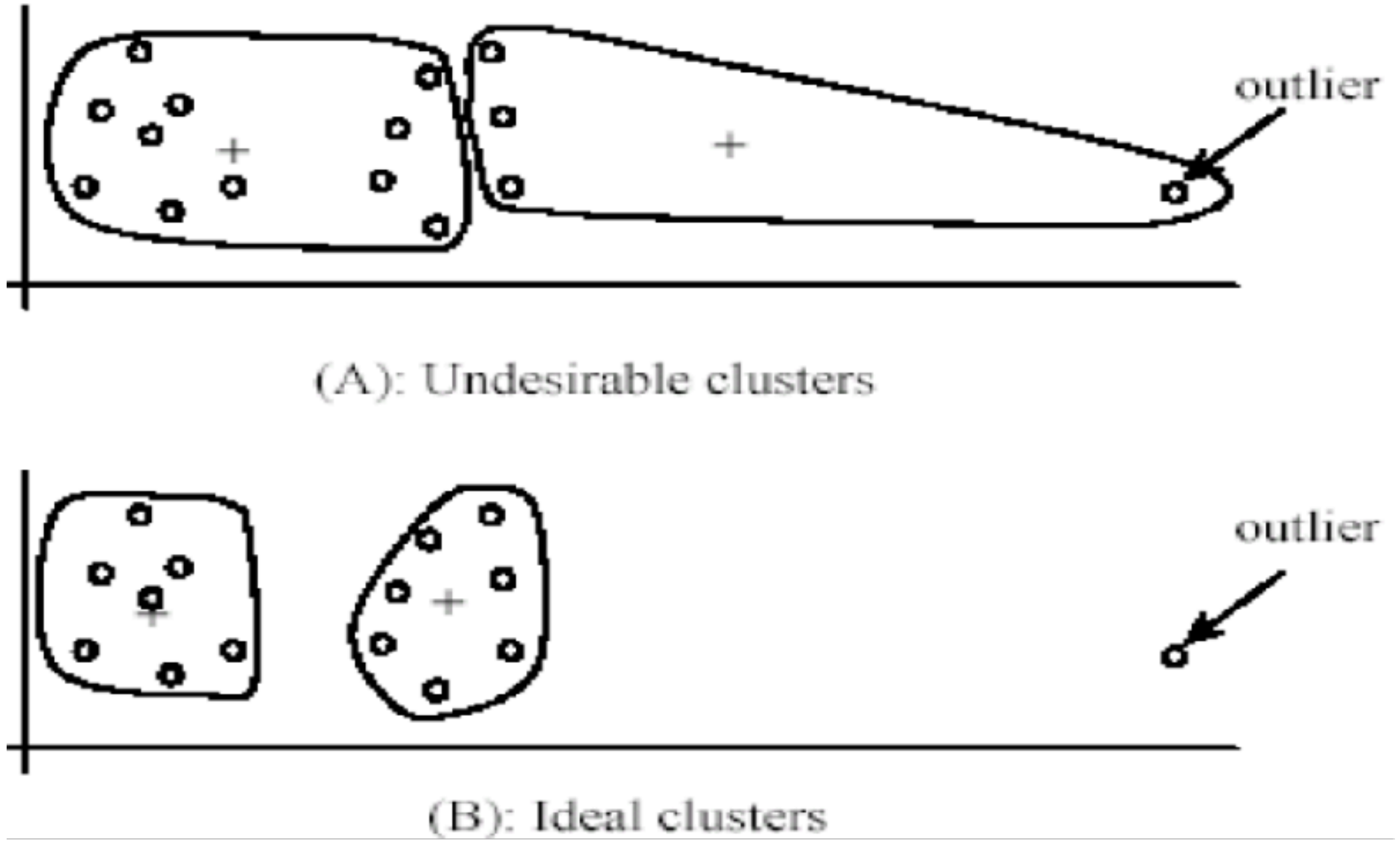
✓Since t and k are relatively small, K-Means is considered a linear algorithm.

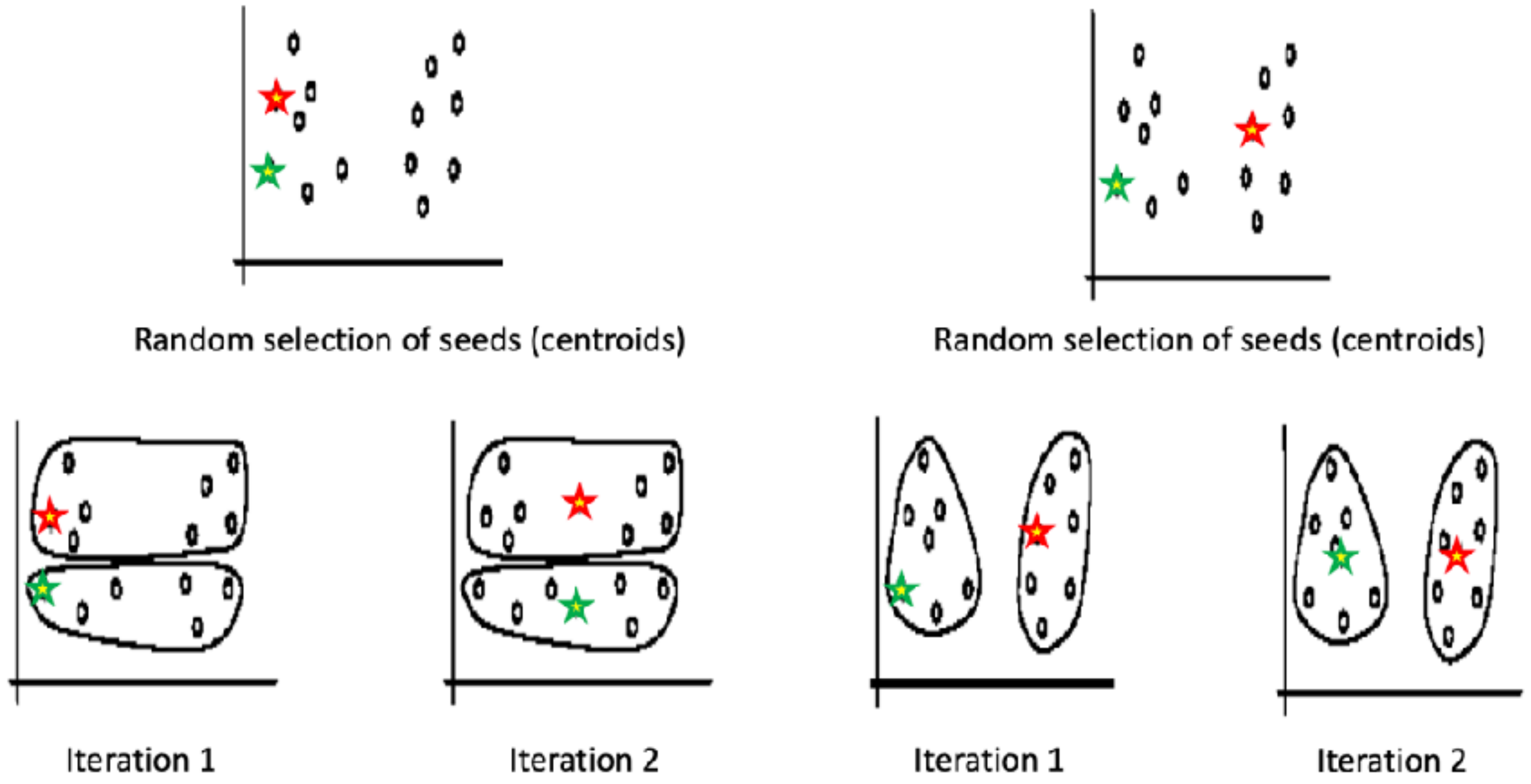


Source: https://en.wikipedia.org/wiki/K-means_clustering

Limitations

✓The user needs to specify the number of centroids k.
✓The mean estimation requires numerical data. For categorical data, k-mode or most frequent value can be used.
✓ Outliers can have a negative effect in the performance of the algorithm.
✓Different initial centroids can lead to different clusters.
✓Not good for arbitrary shaped clusters (spatial).



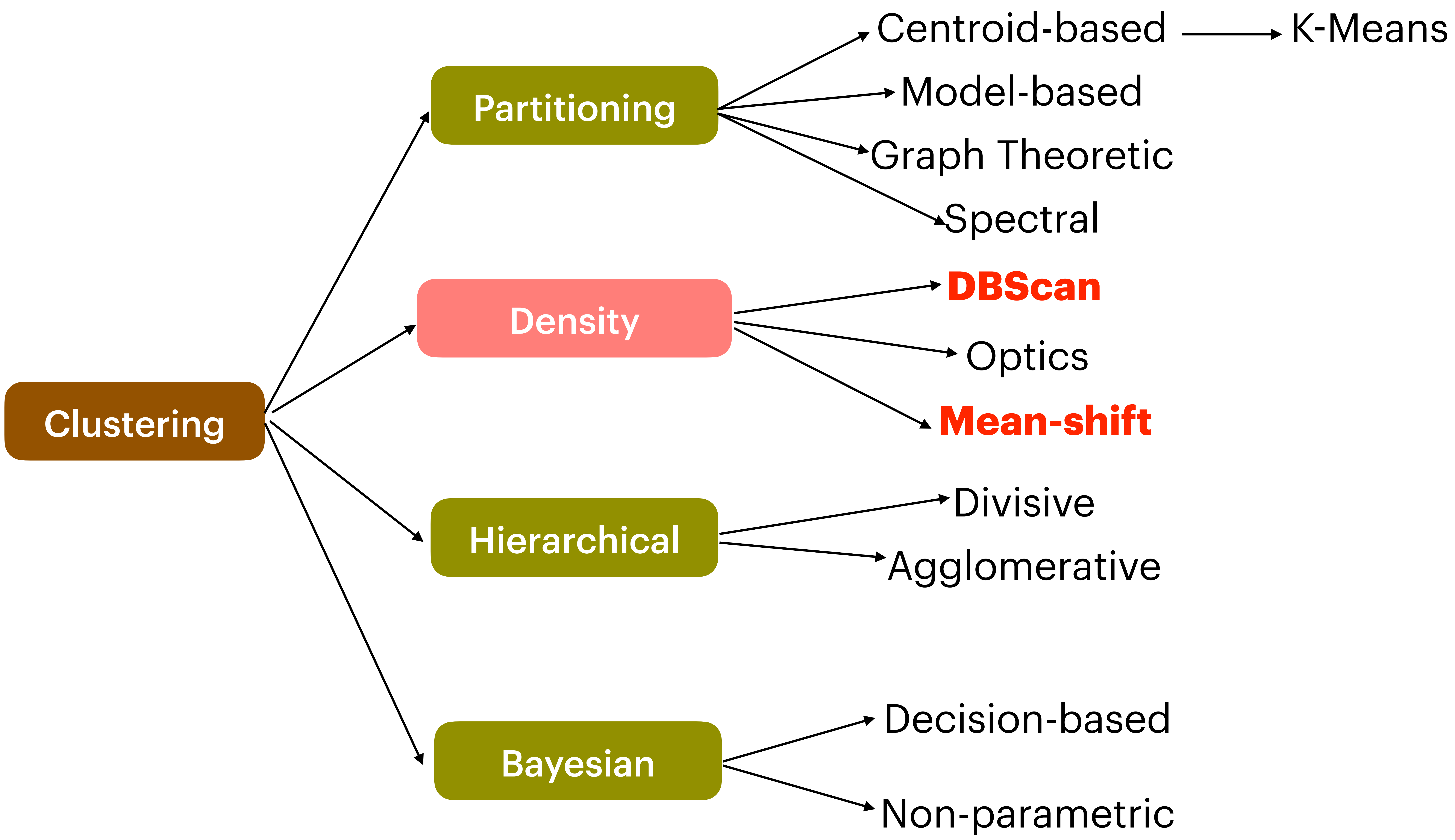


Source: http://www.mit.edu/~9.54/fall14/slides/Class13.pdf

3.2. Density-based Clustering

Density Based Clustering

Taxonomy



Density Based Clustering

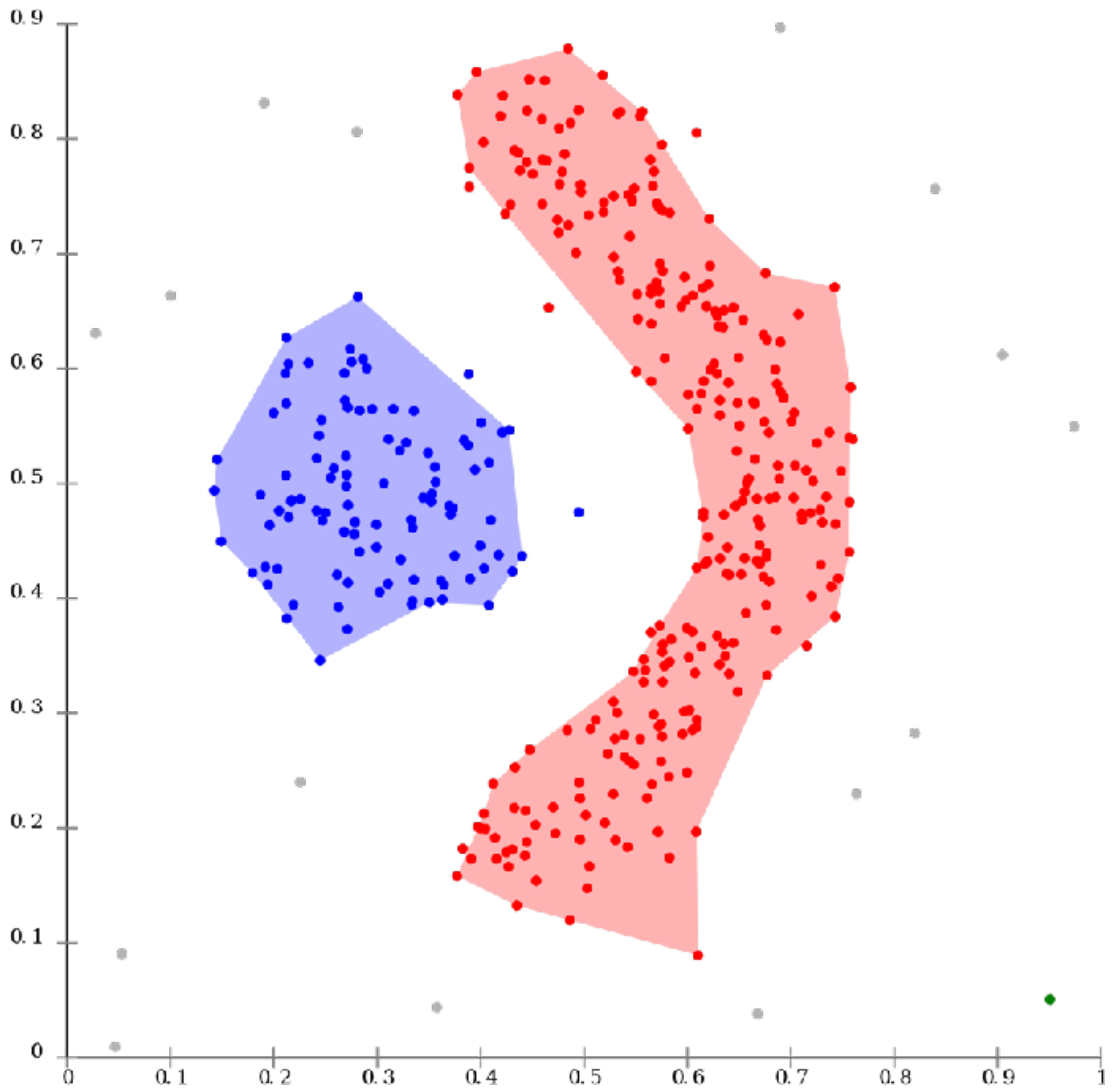
✓Good for spatial data.

✓Considers high-density regions (clusters) and low-density regions (void or outliers).

✓A cluster is a region which is density connected, i.e., the density of points in that region is greater than a minimum.

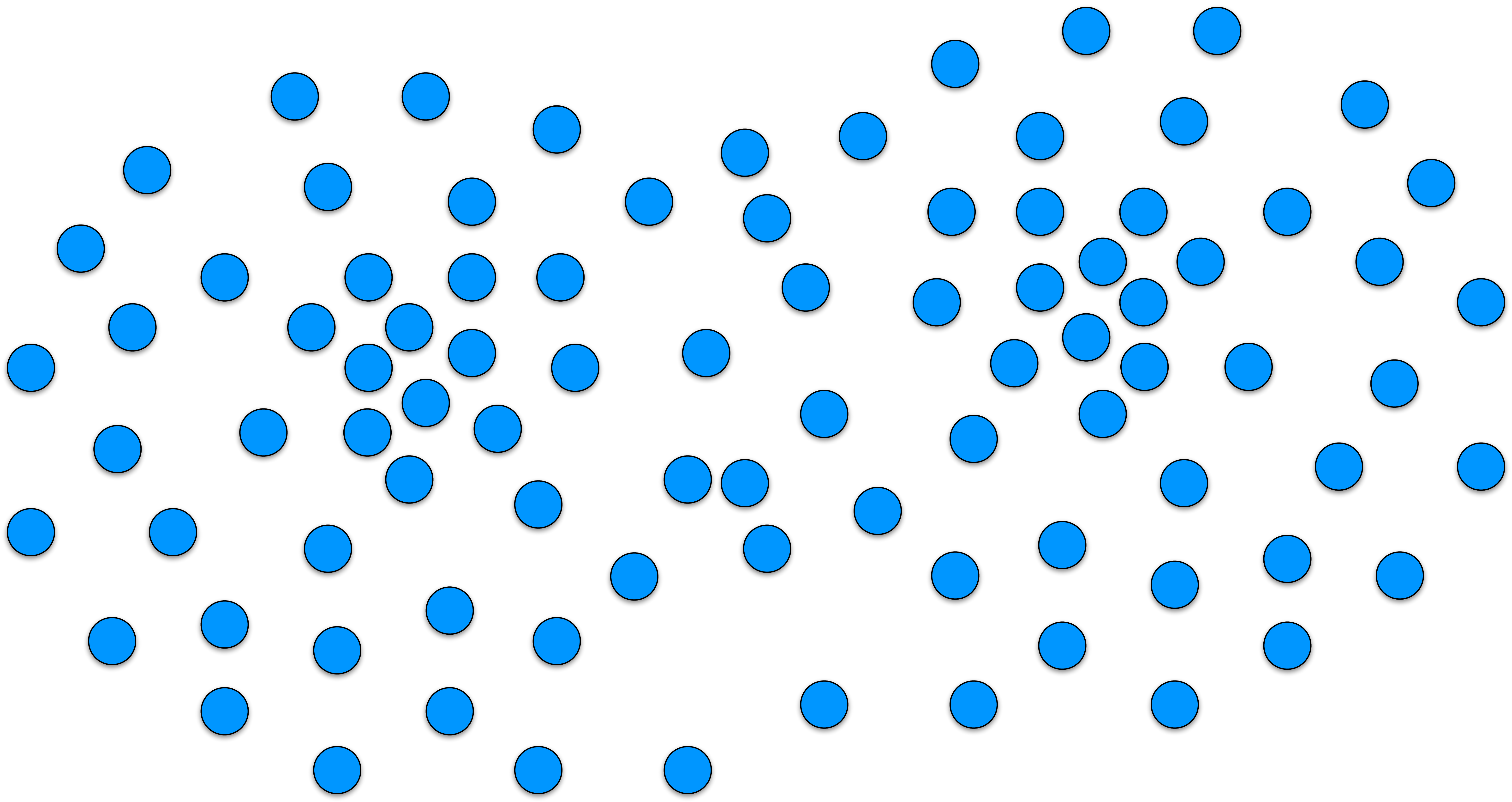
✓Since density is the criterion for cluster expansions, this algorithms works well for arbitrary-shaped clusters.

✓DBScan is an example of this clustering category.



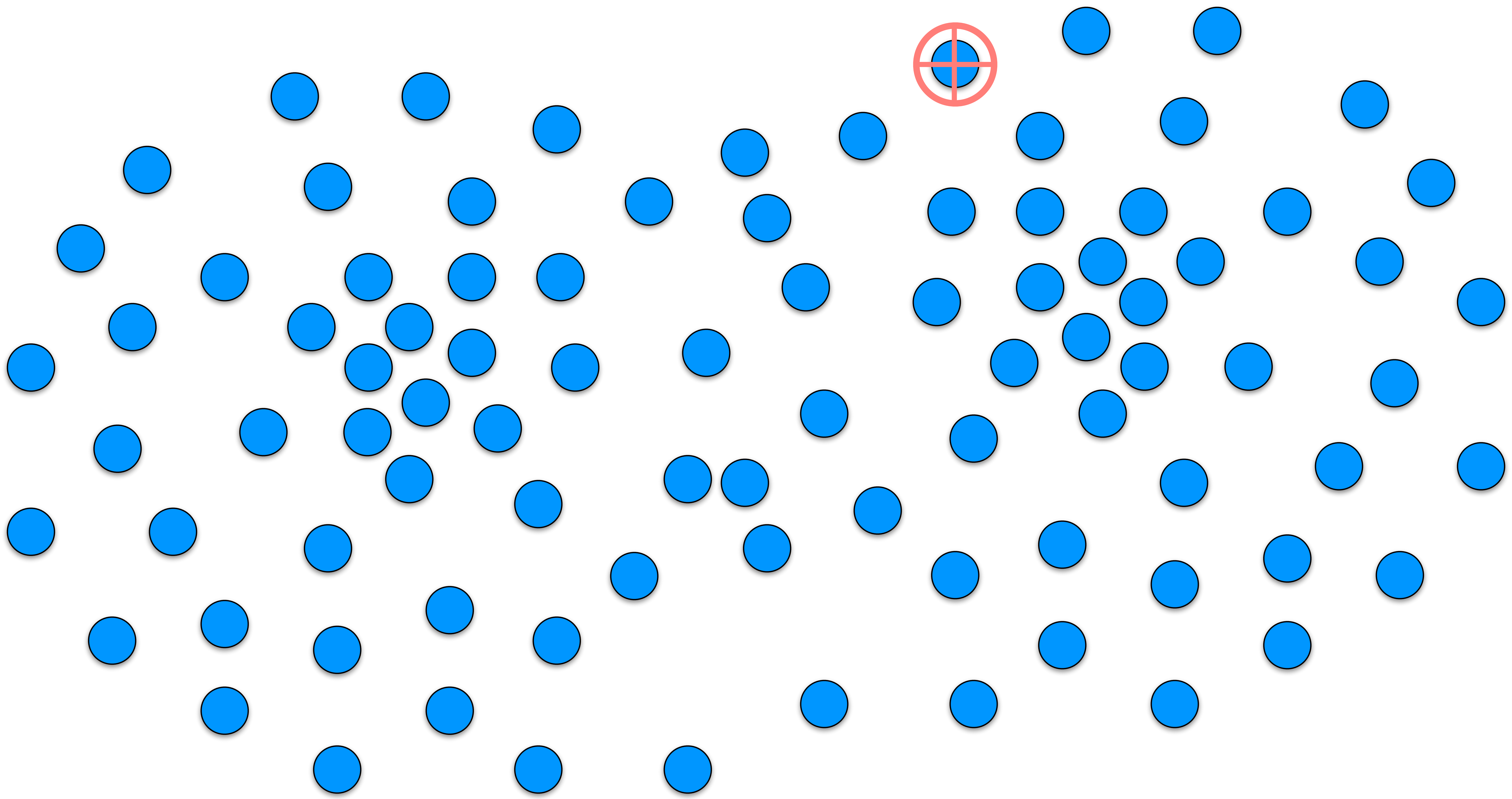
3.2.1. Mean-Shift

Algorithm



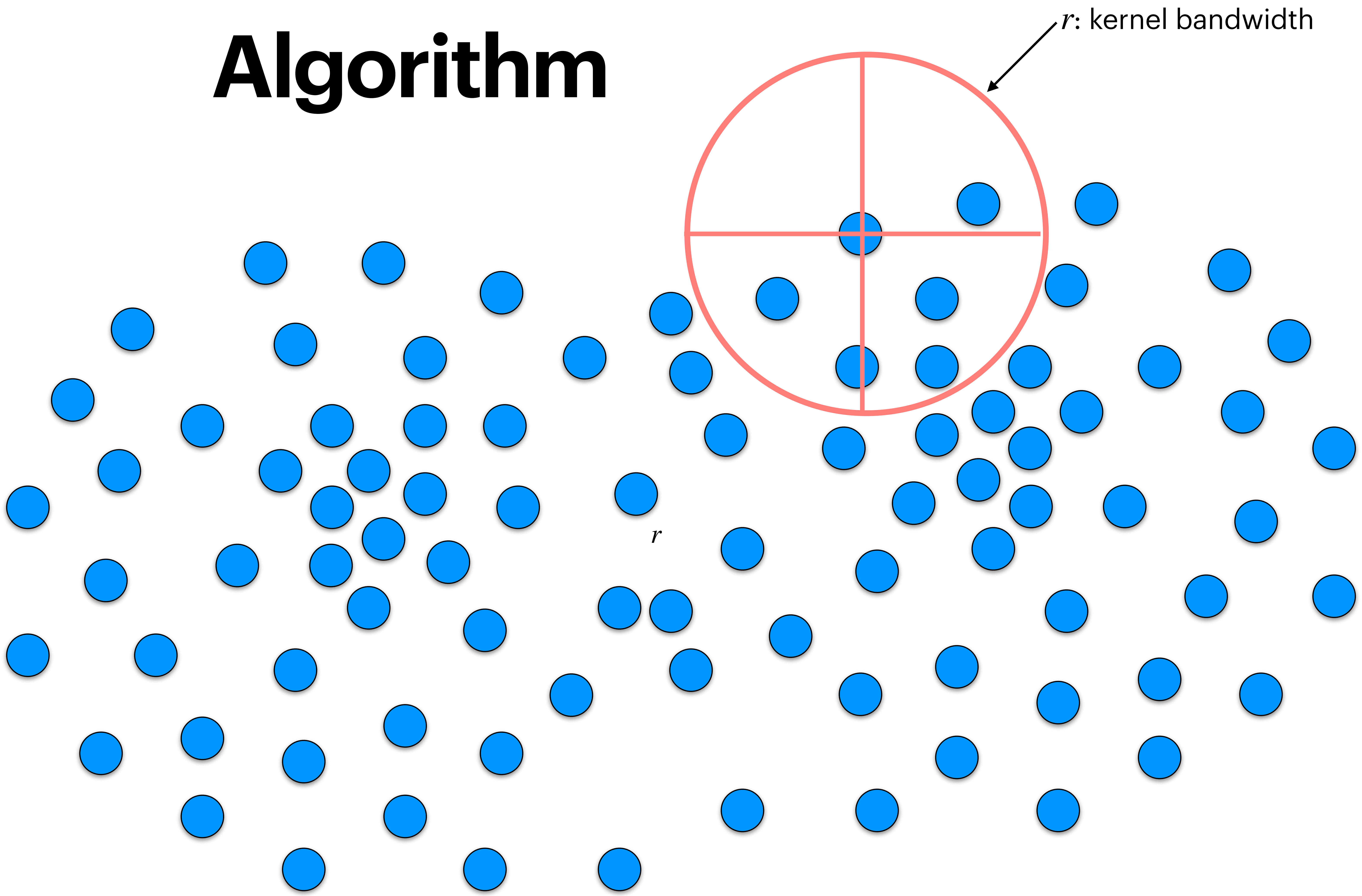
Algorithm

1. Select a point (random)



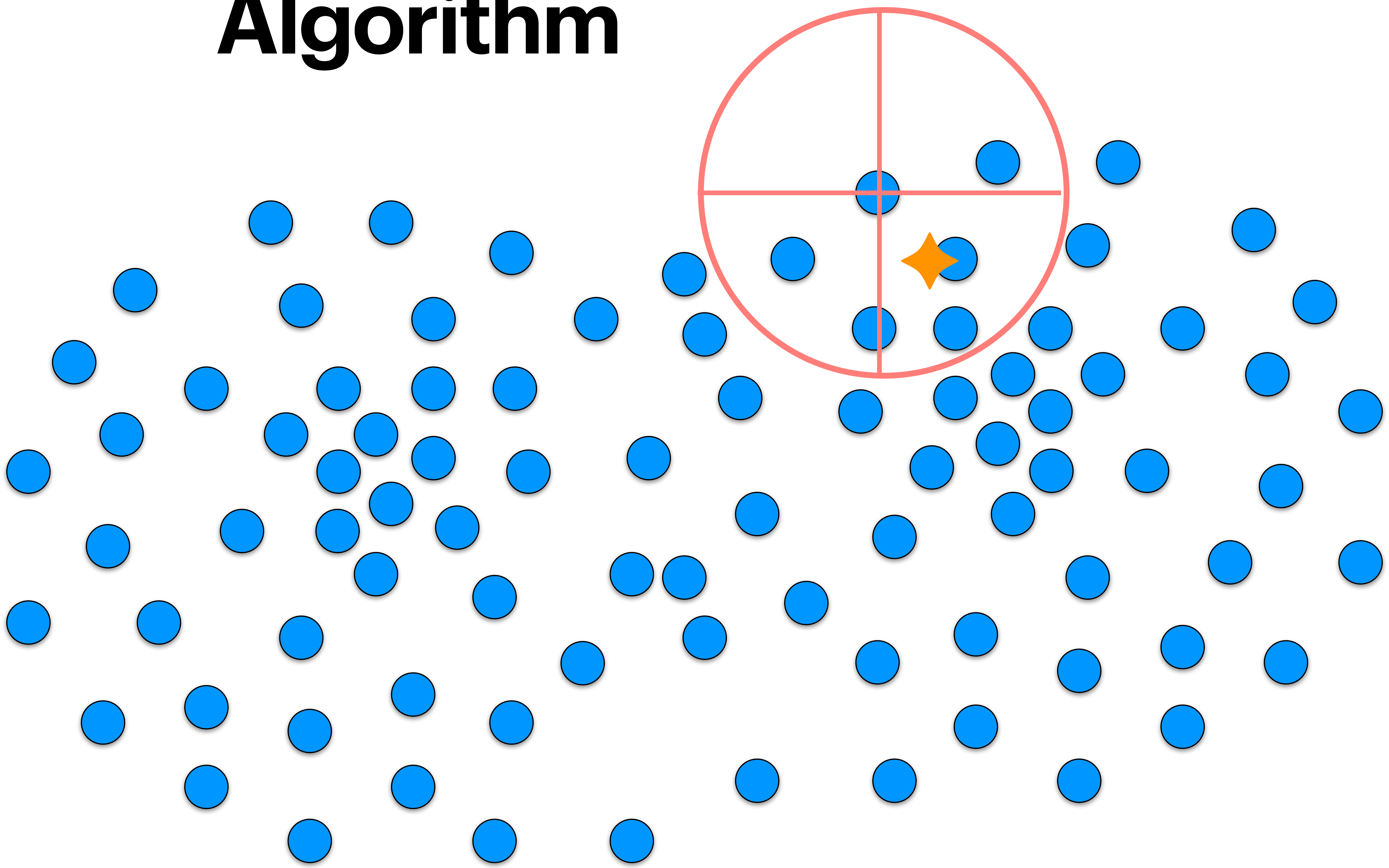
Algorithm

1. Select a point (random)
2. Compute neighborhood



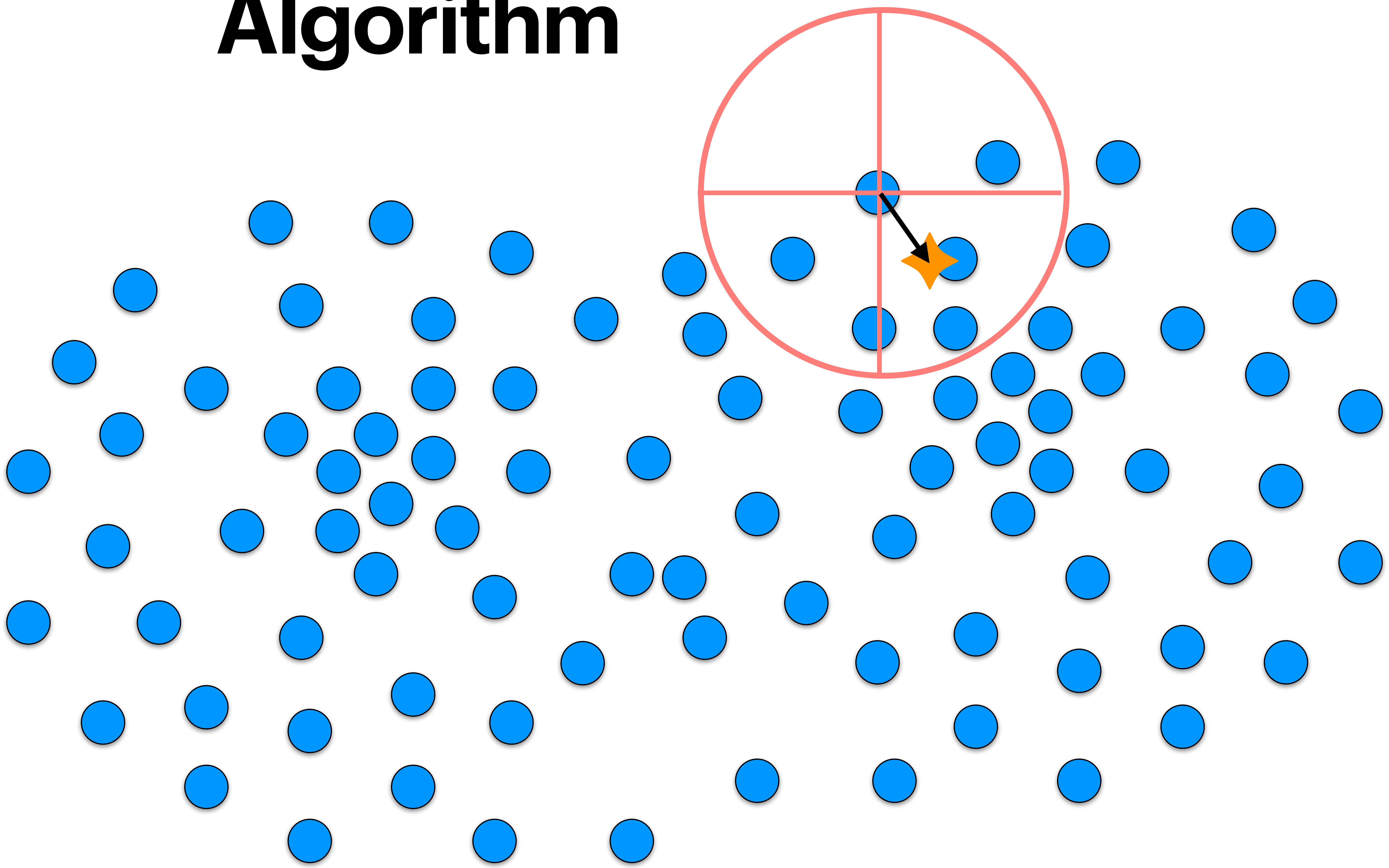
Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)



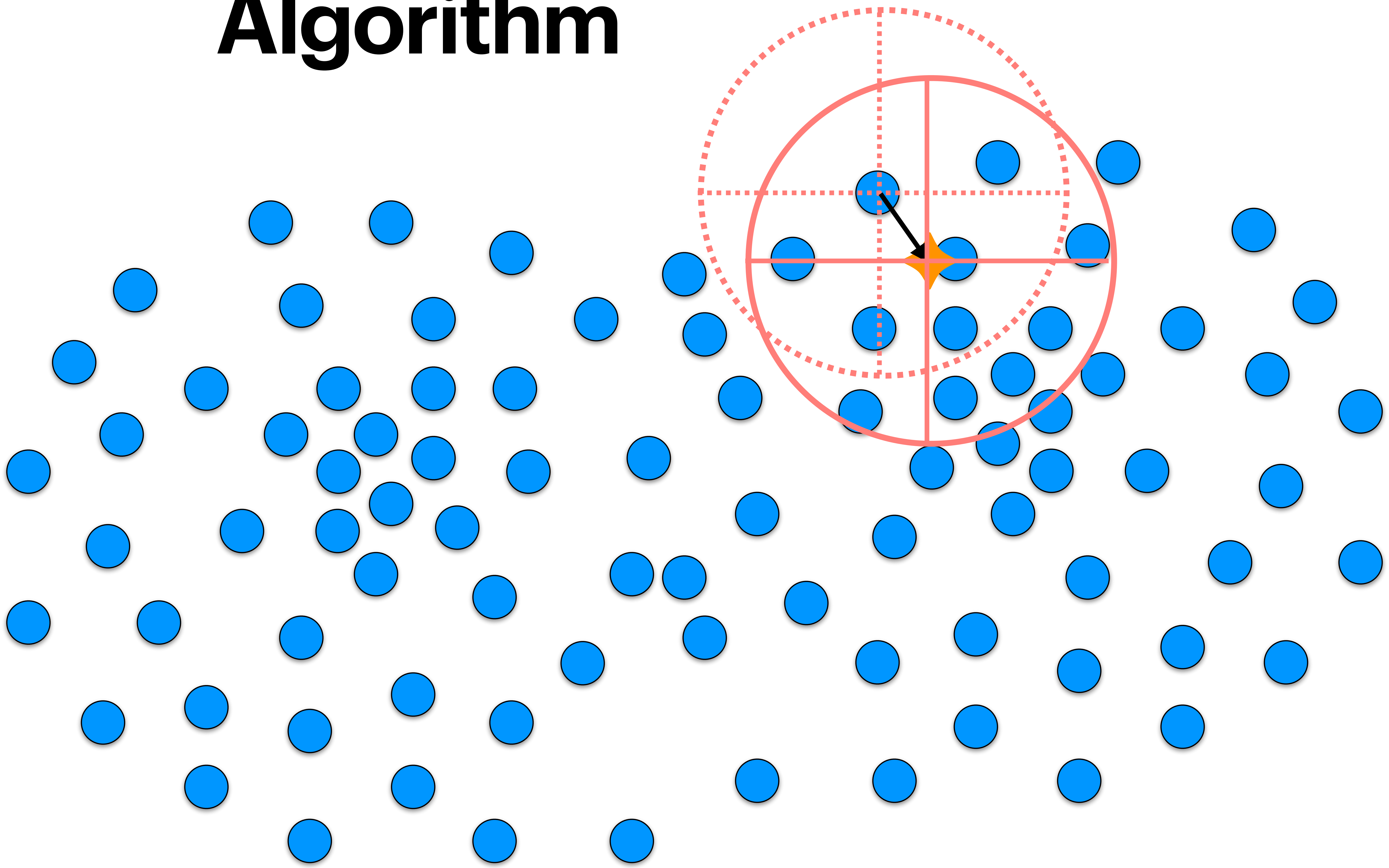
Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift



Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

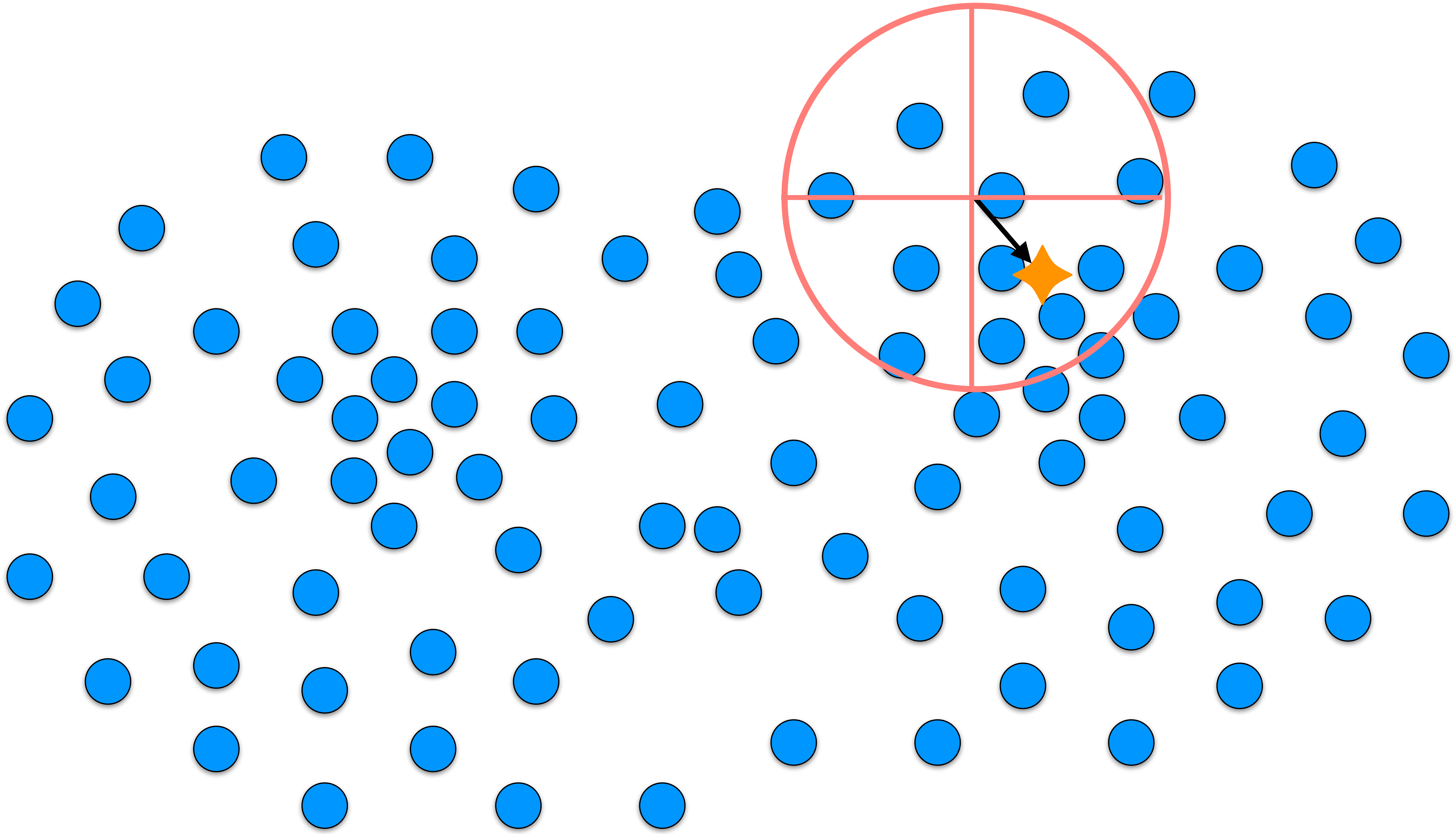


Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

   ...

6. Repeat until convergence

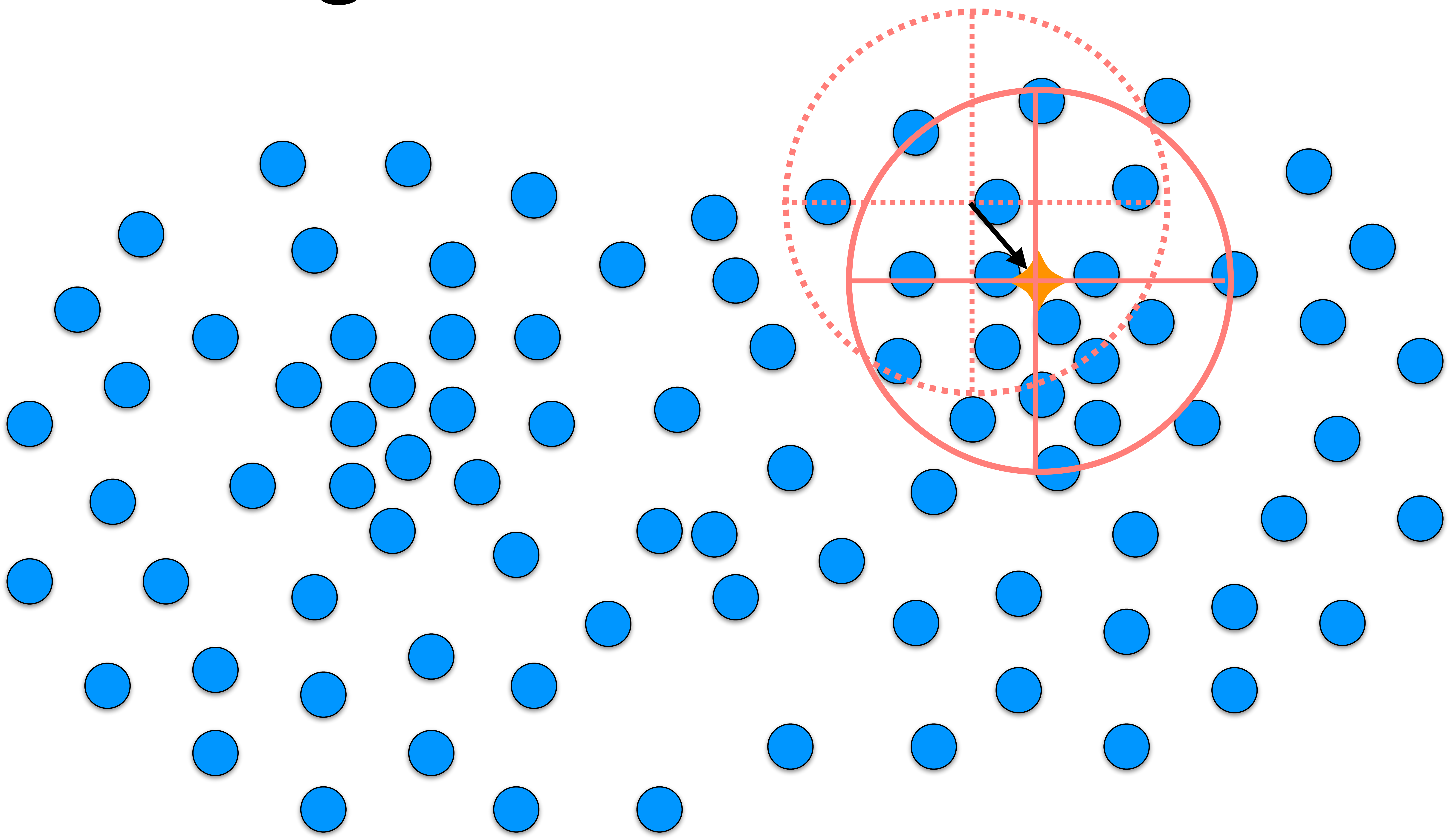


Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

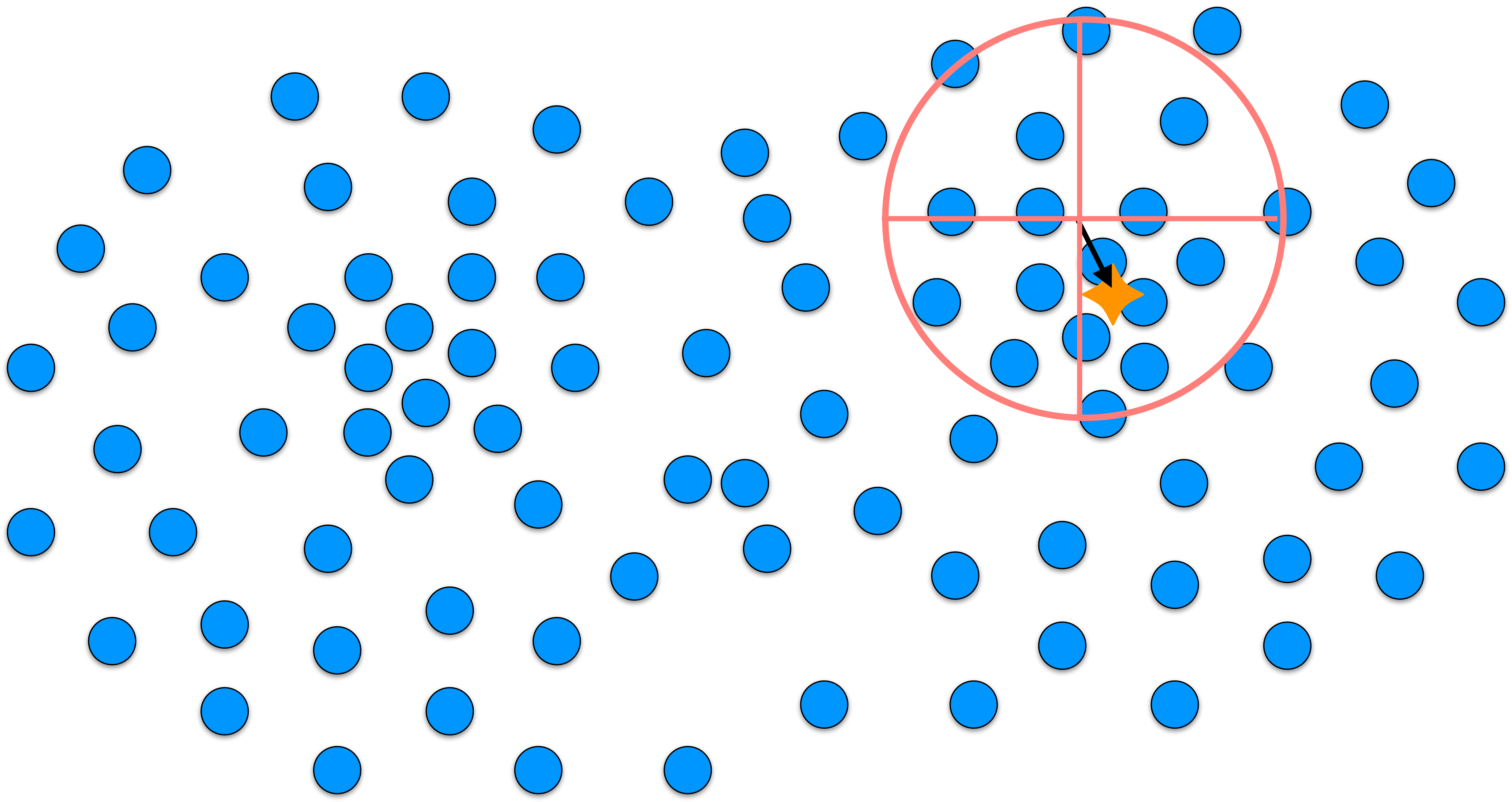
   ...

6. Repeat until convergence



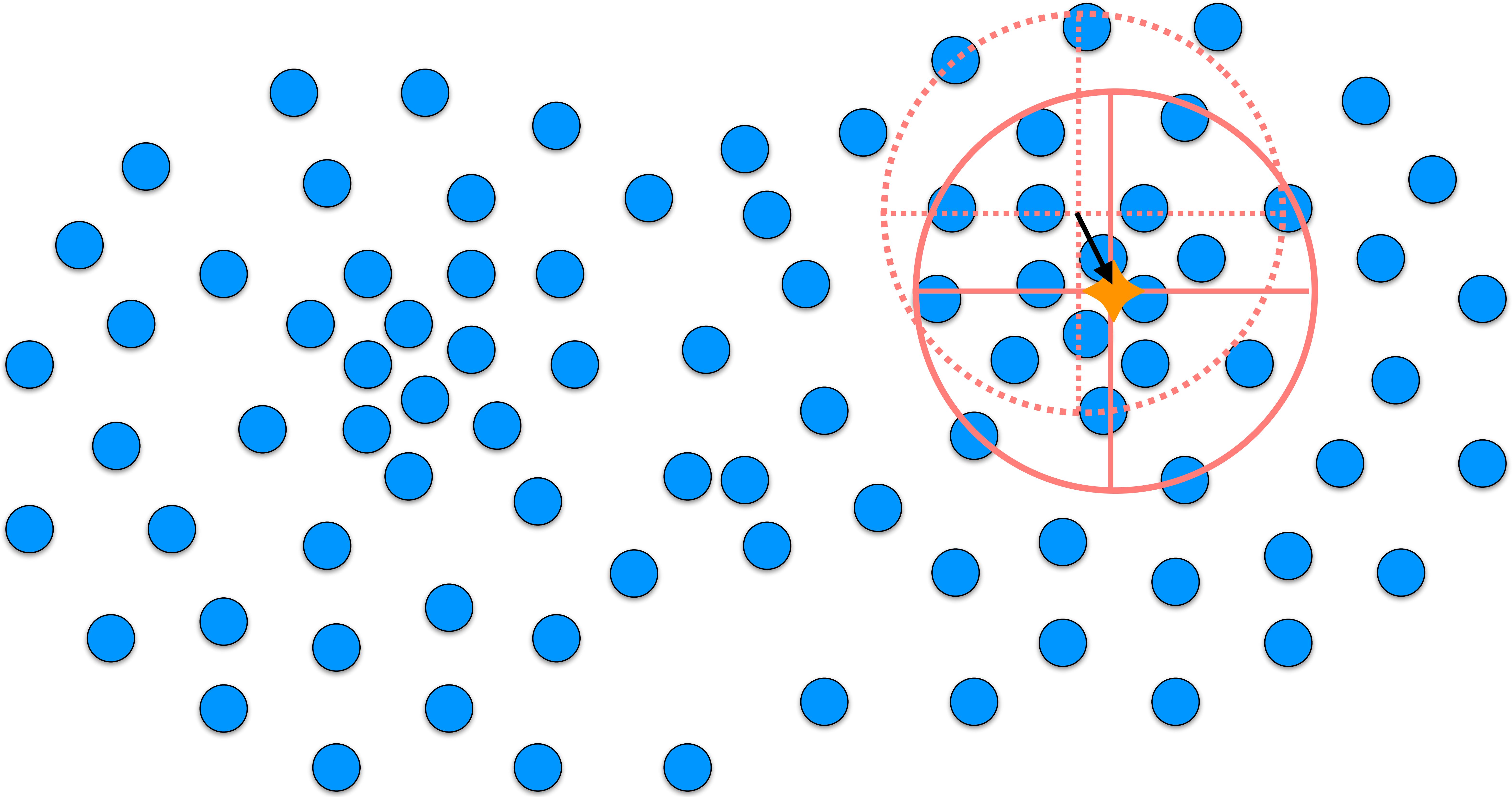
Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood
   ...
6. Repeat until convergence



Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood
...
6. Repeat until convergence

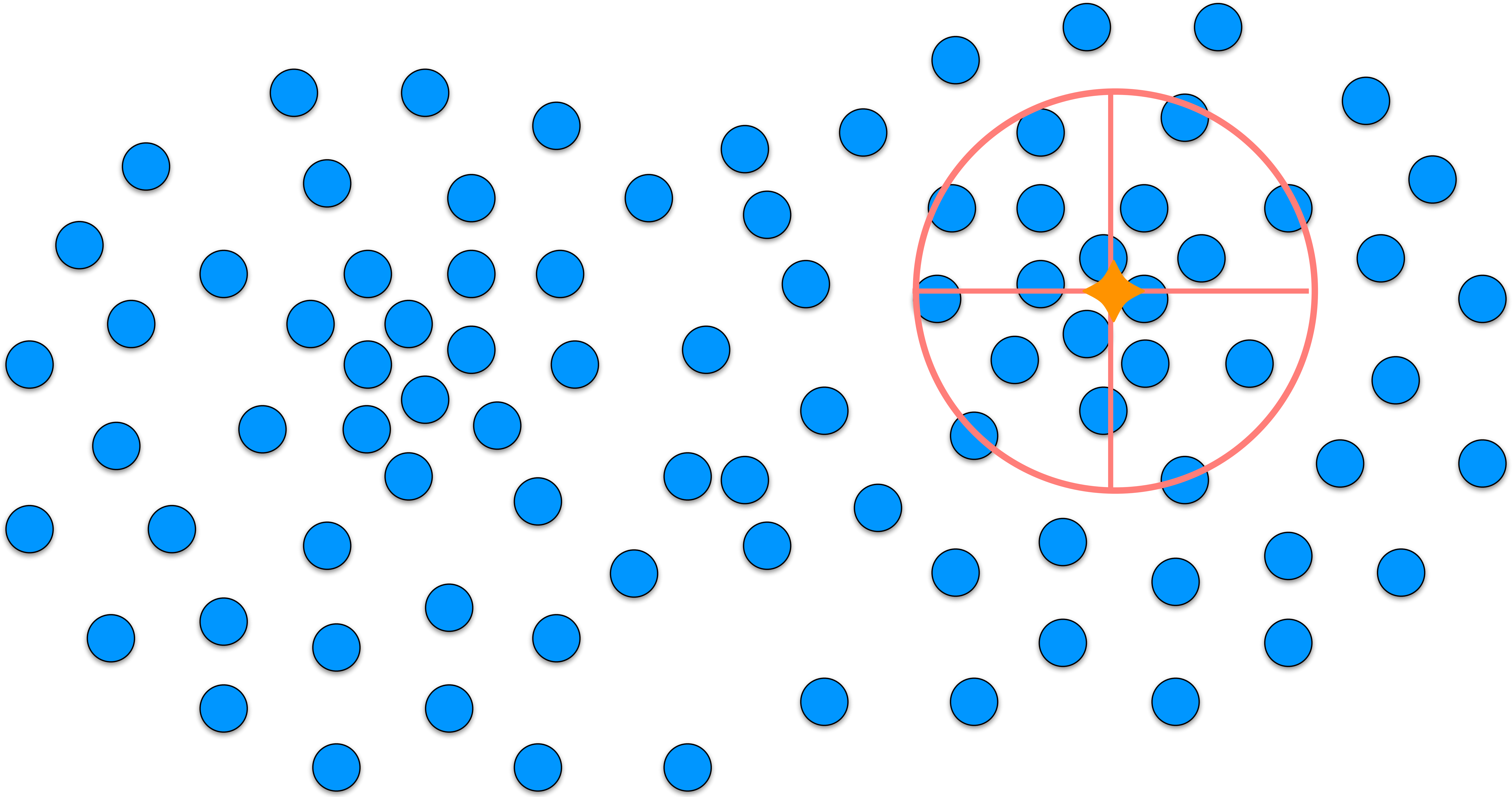


Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

   ...

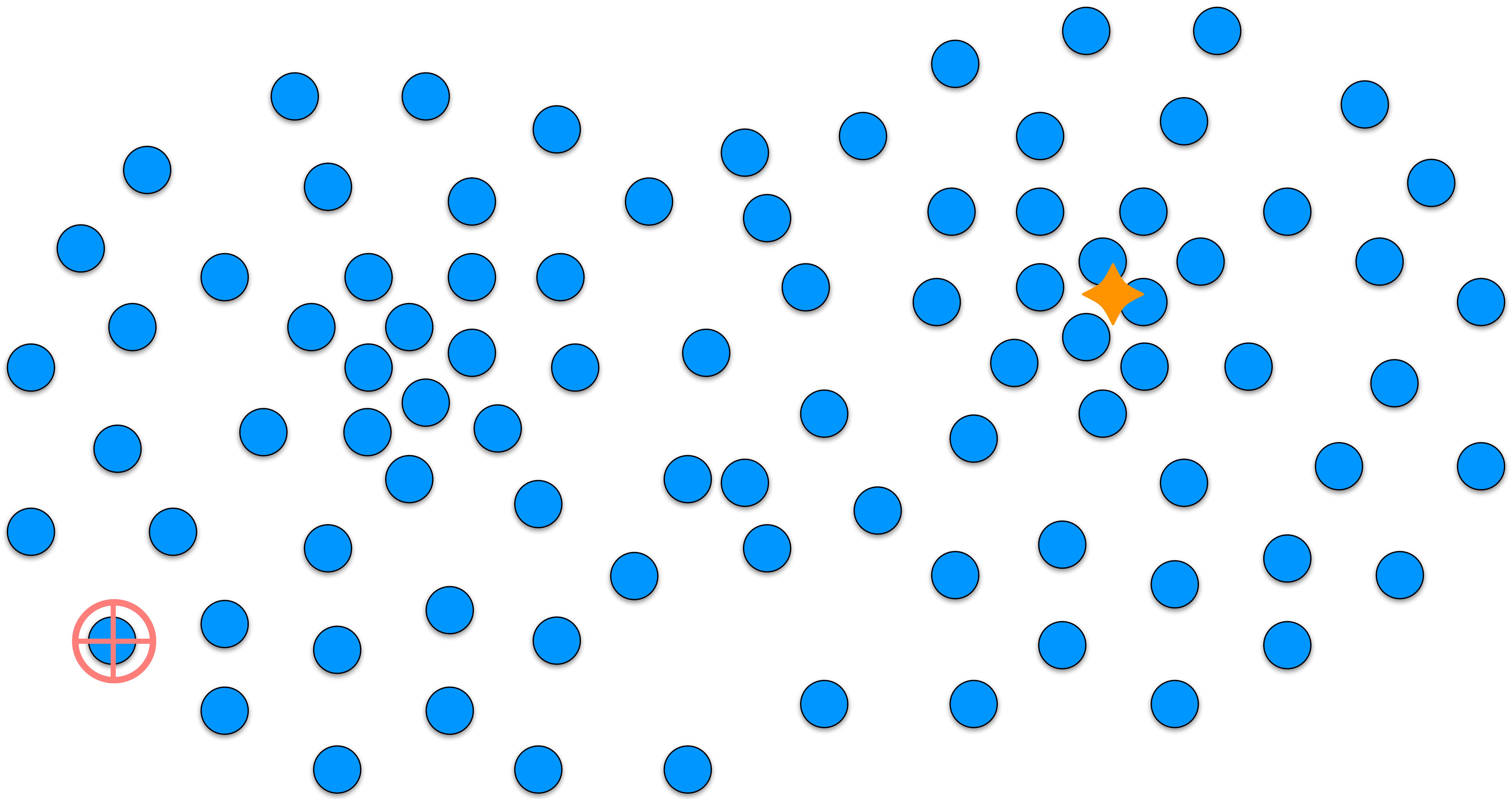
6. Repeat until convergence



Algorithm

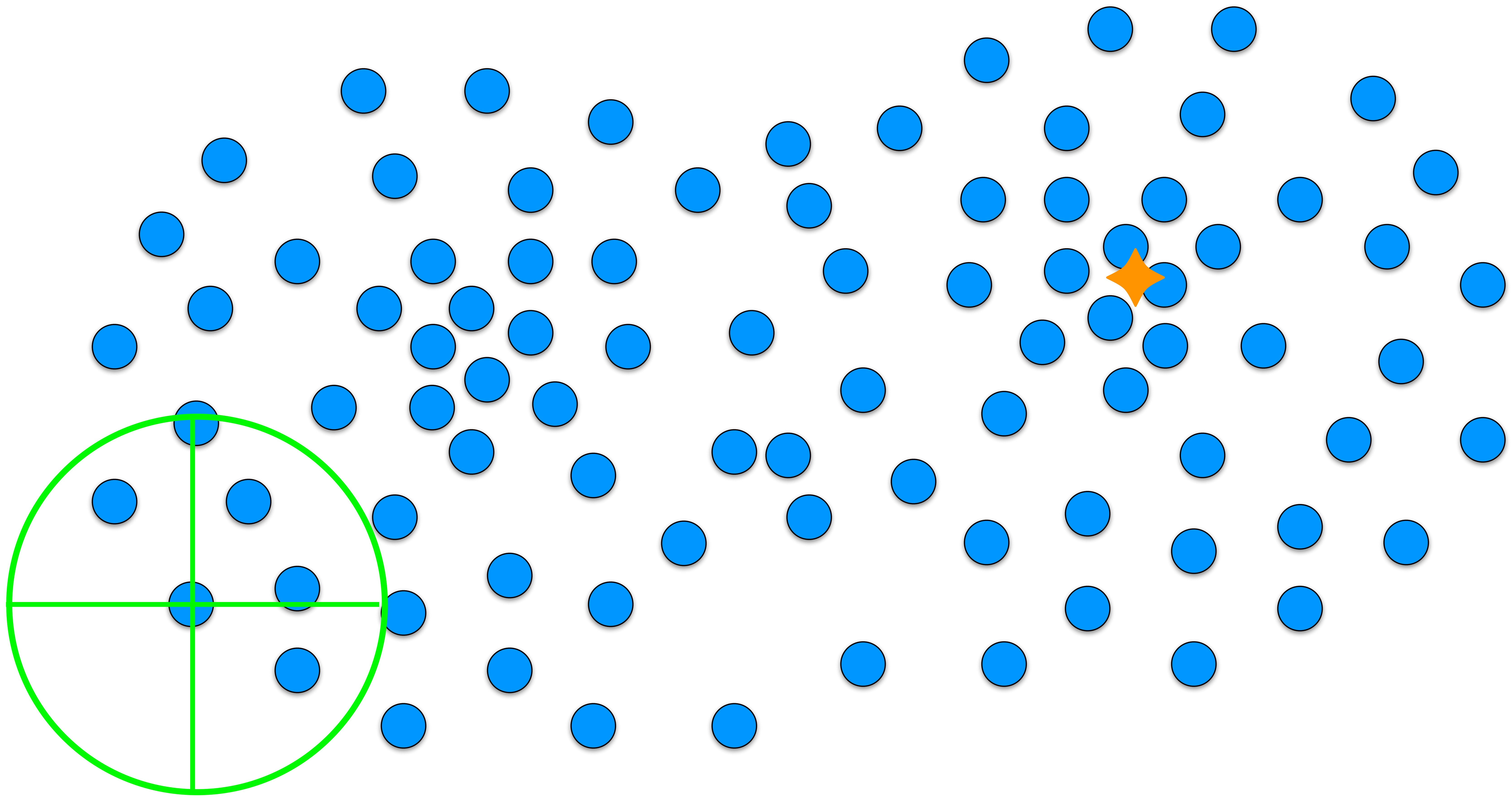
1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

   ...
6. Repeat until convergence



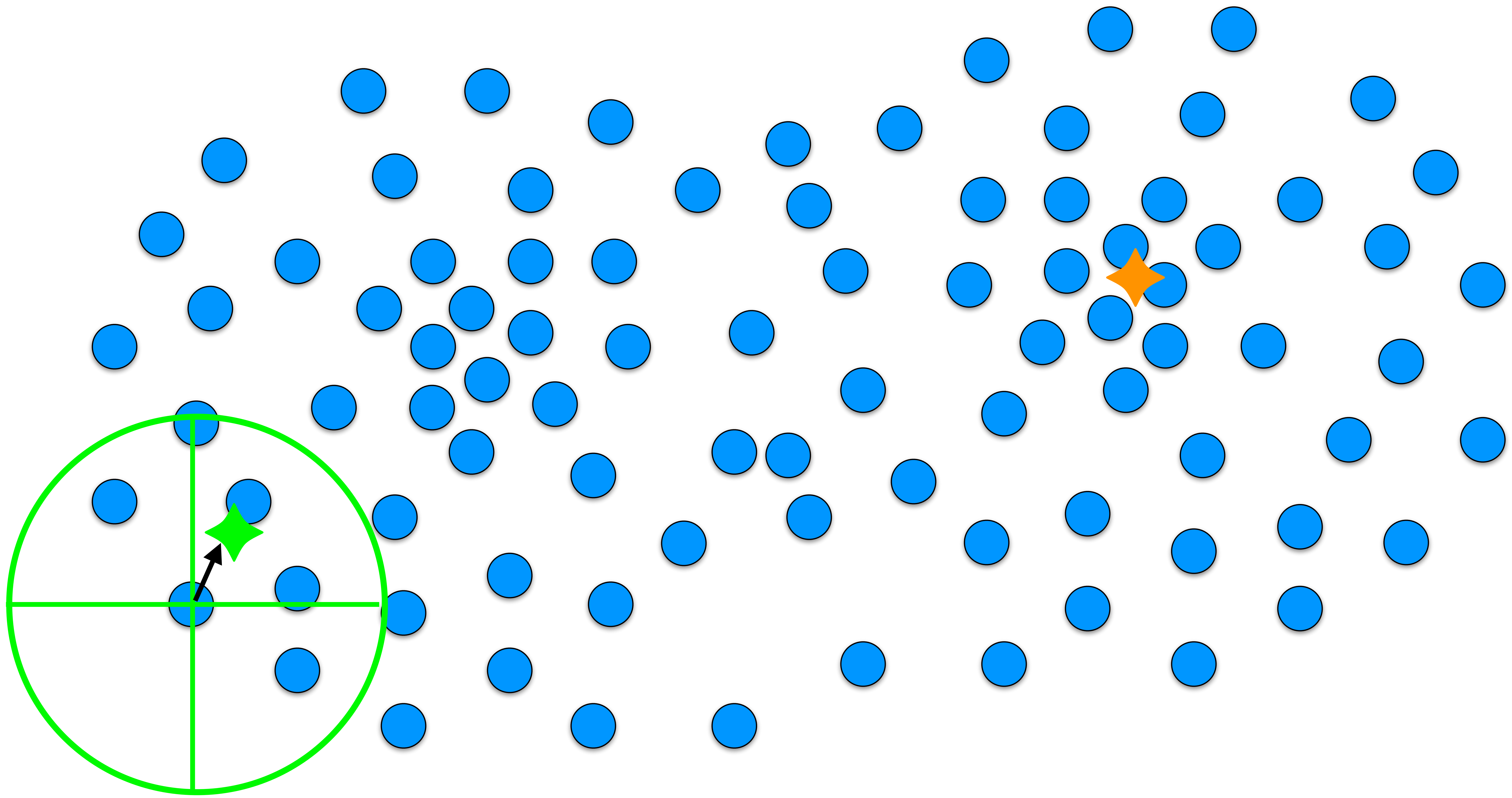
Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood
   ...
6. Repeat until convergence



Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood
...
6. Repeat until convergence

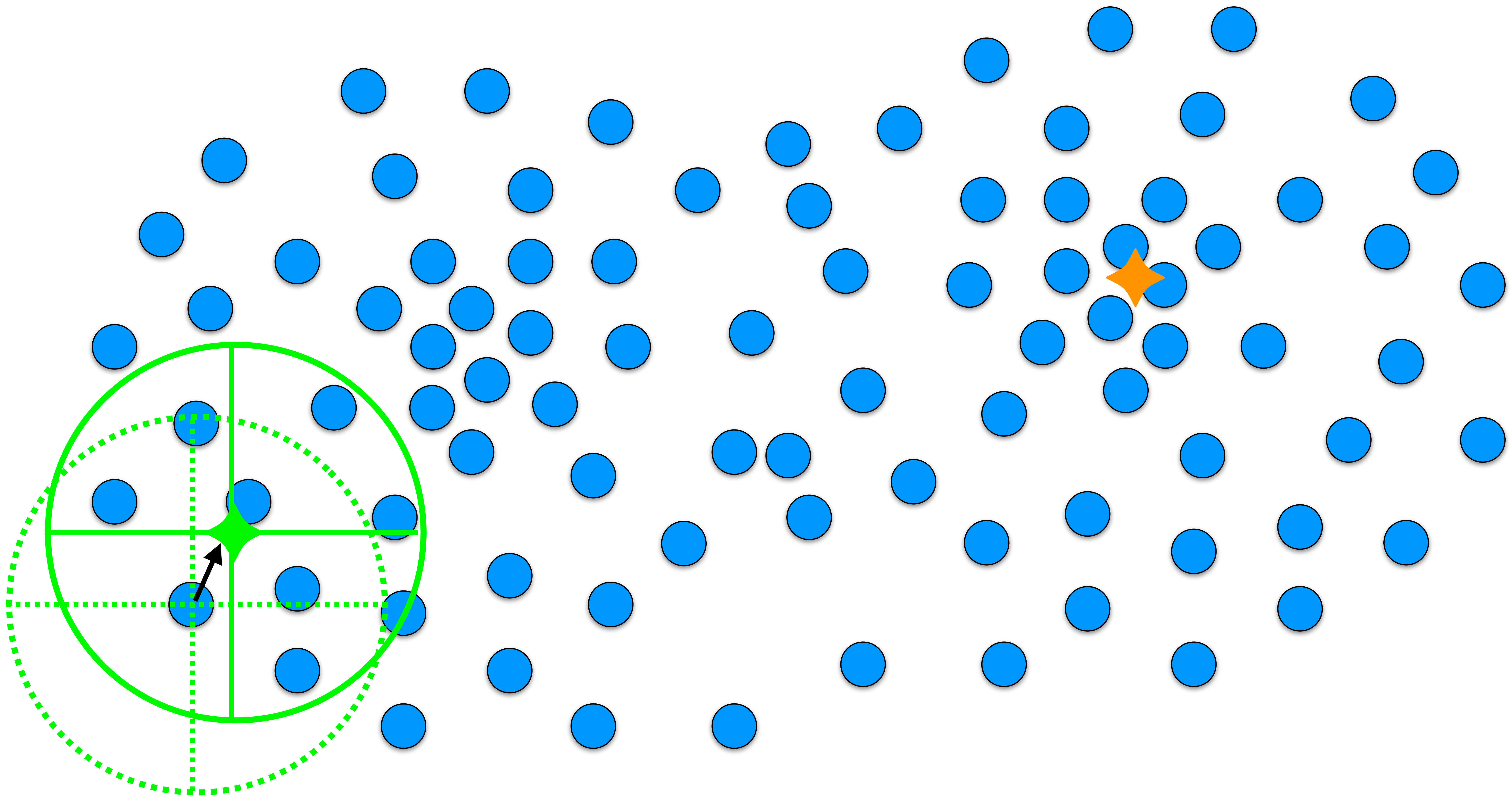


Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

...

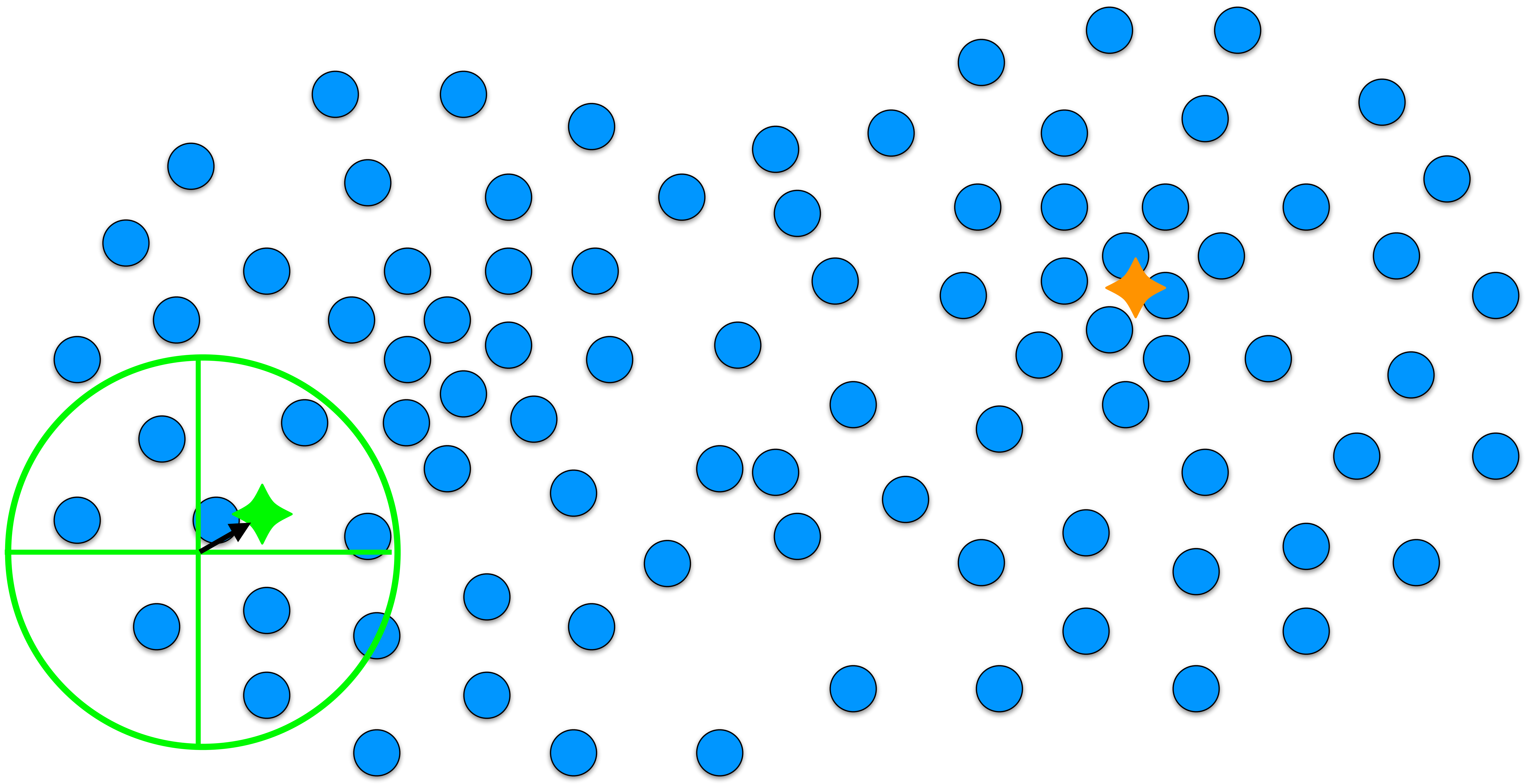
6. Repeat until convergence



Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

   …
6. Repeat until convergence

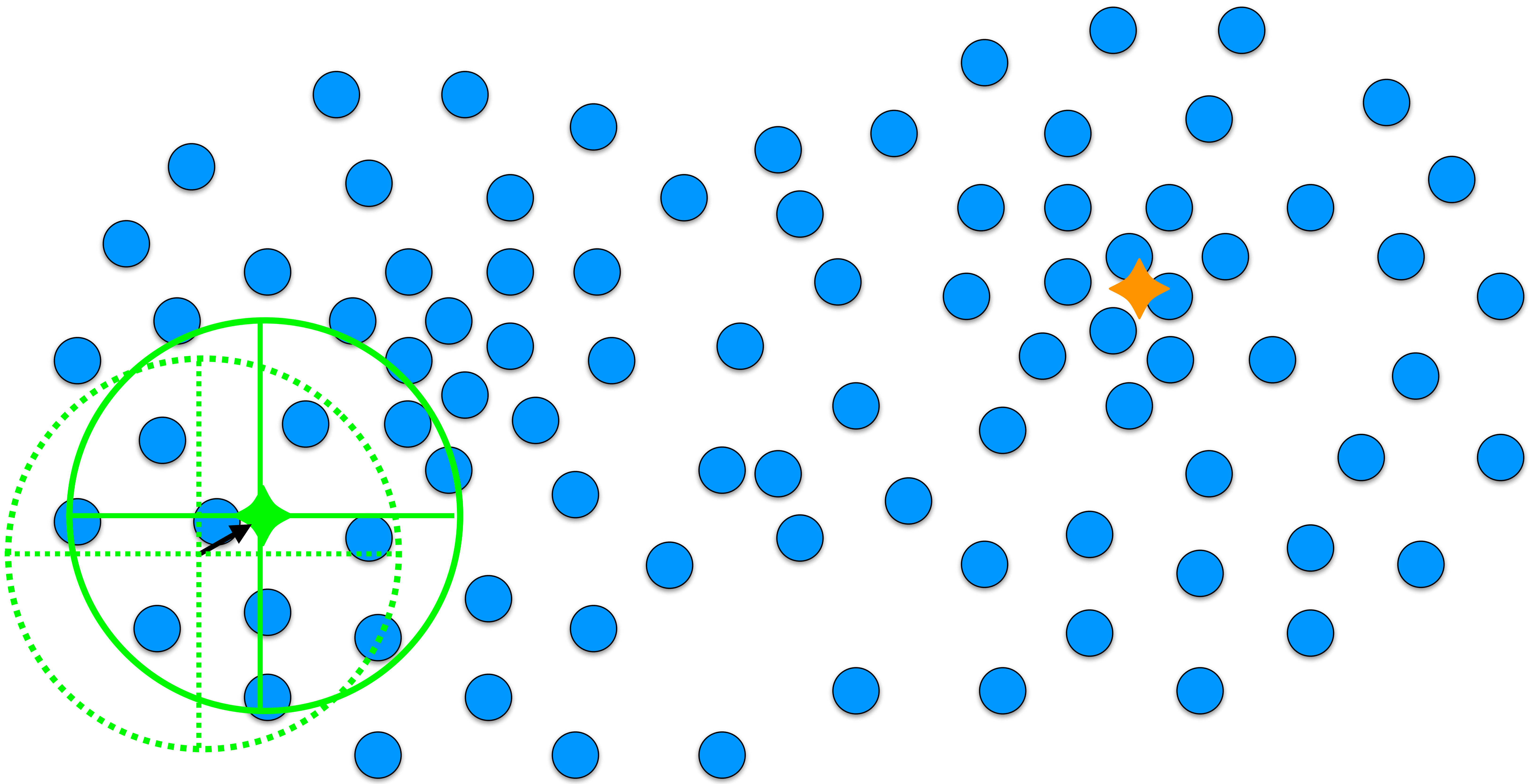


Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

   …

6. Repeat until convergence

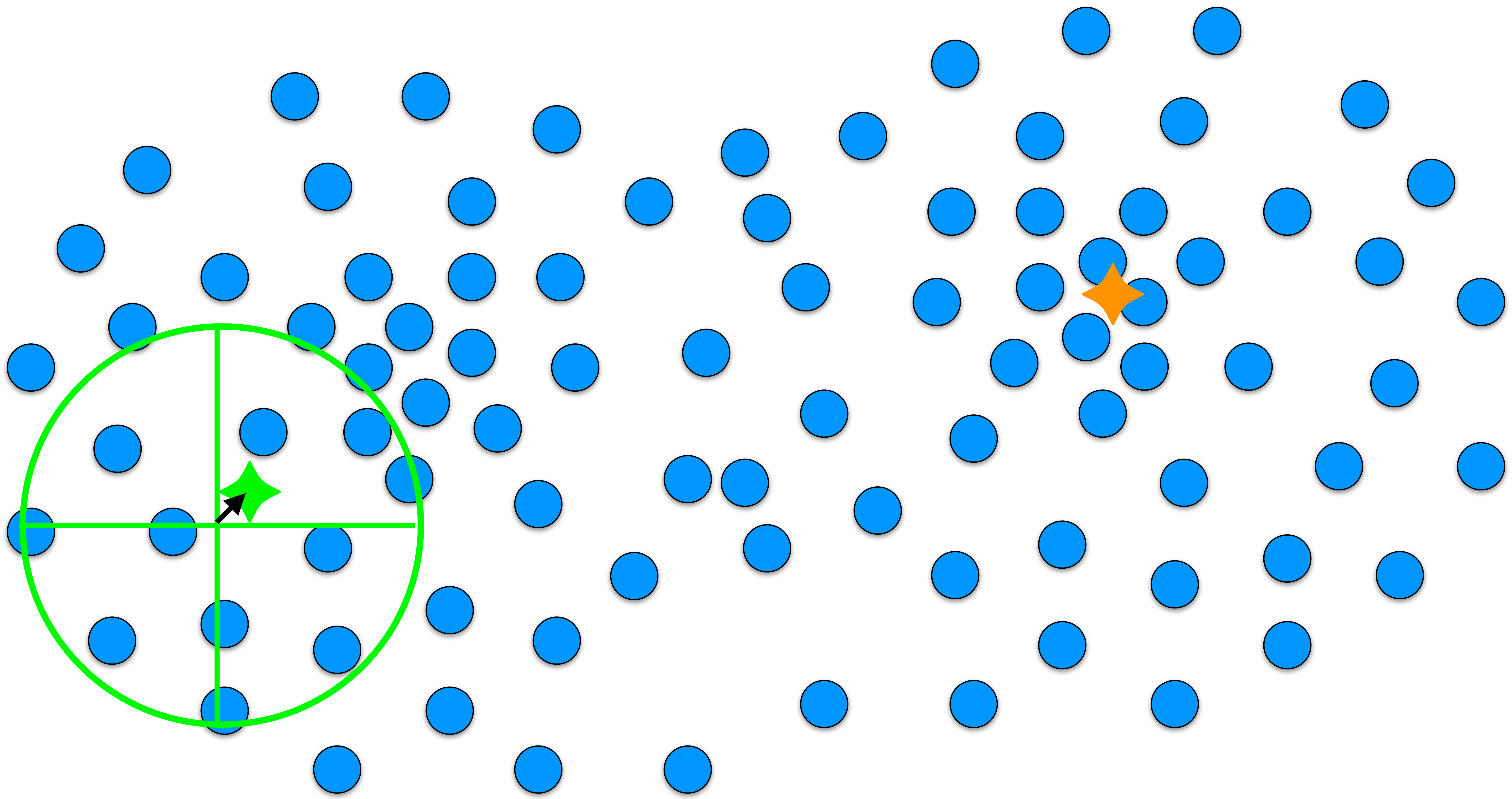


Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

   ...

6. Repeat until convergence

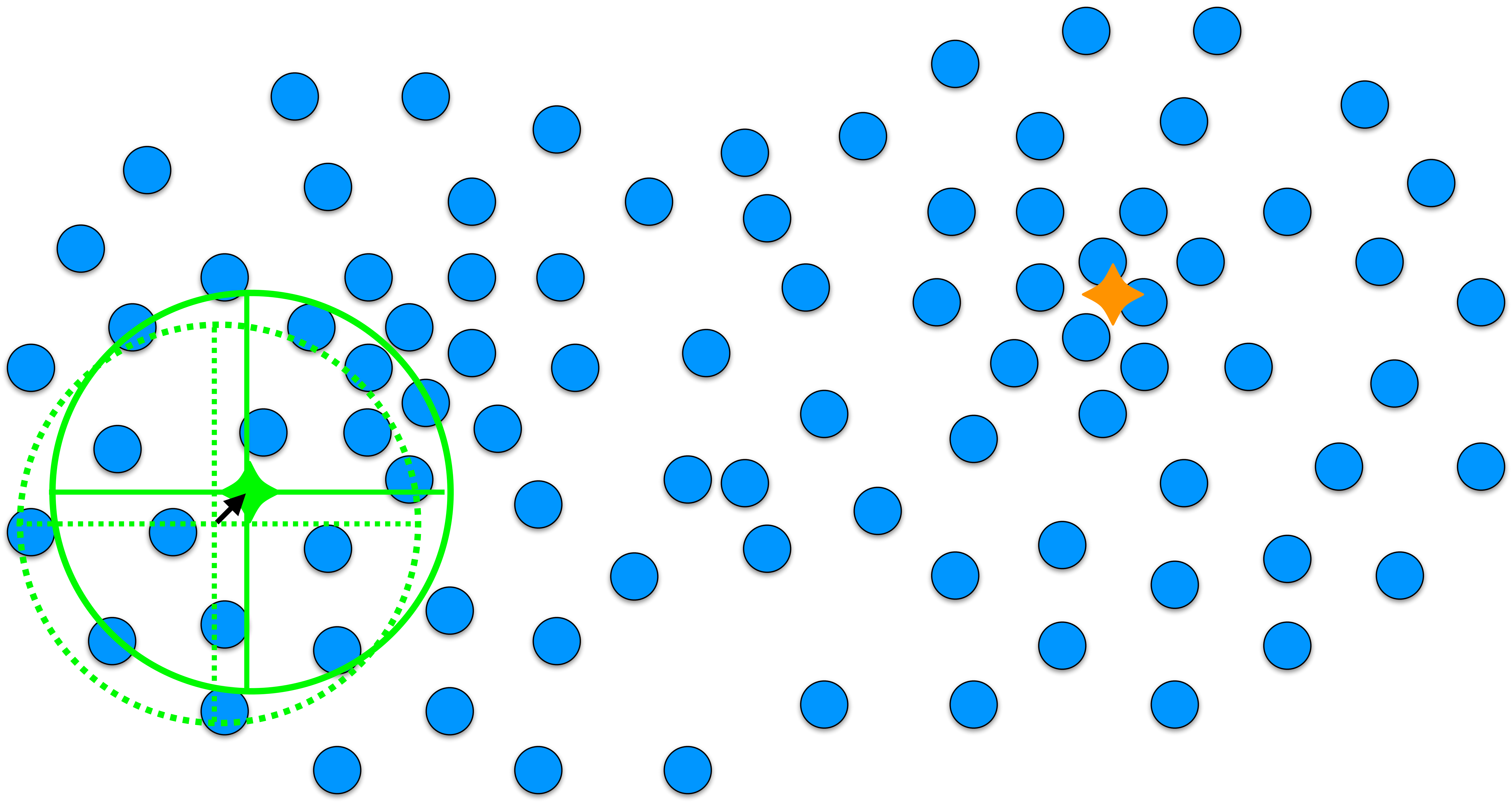


Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

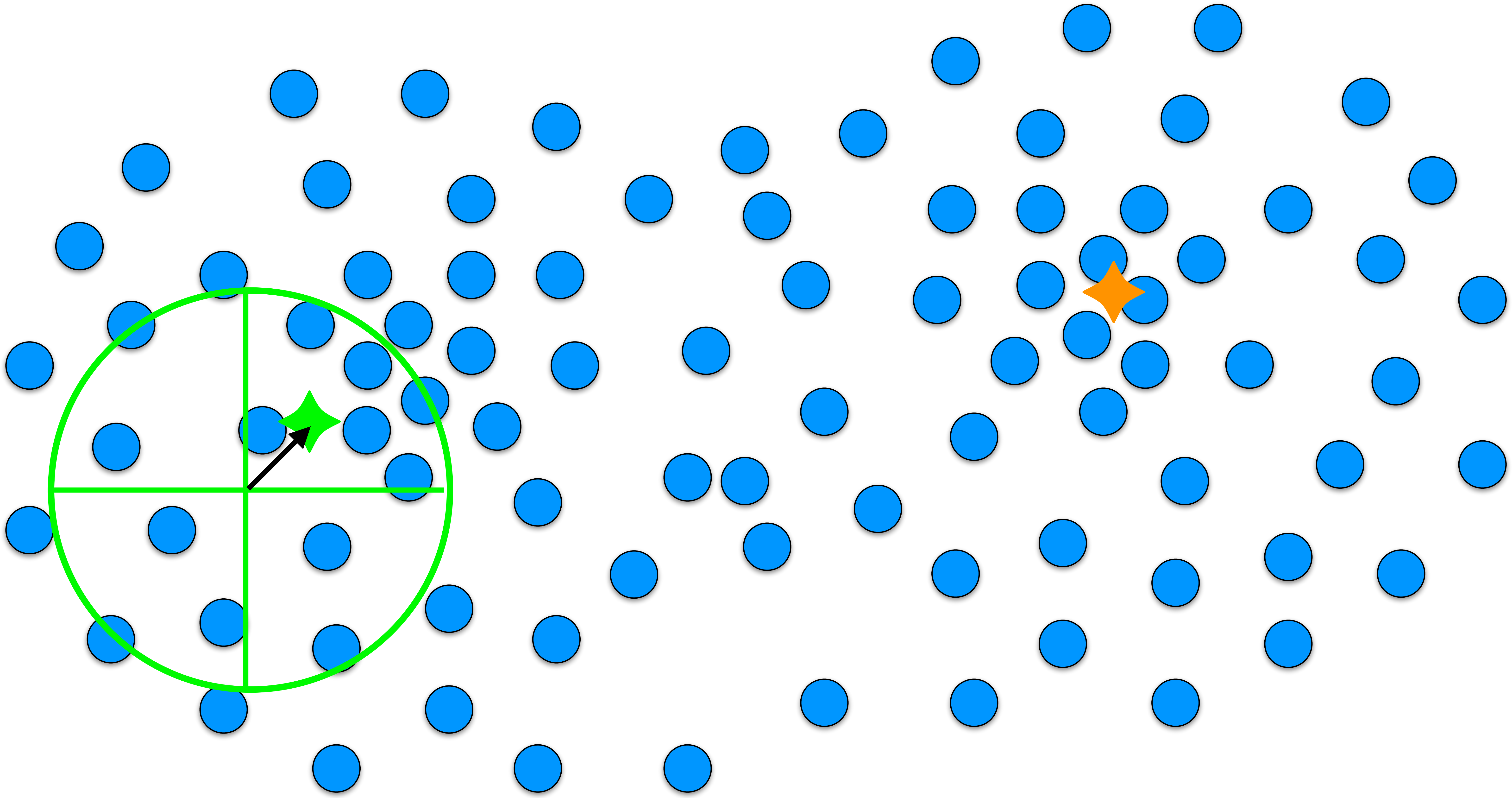
   …

6. Repeat until convergence



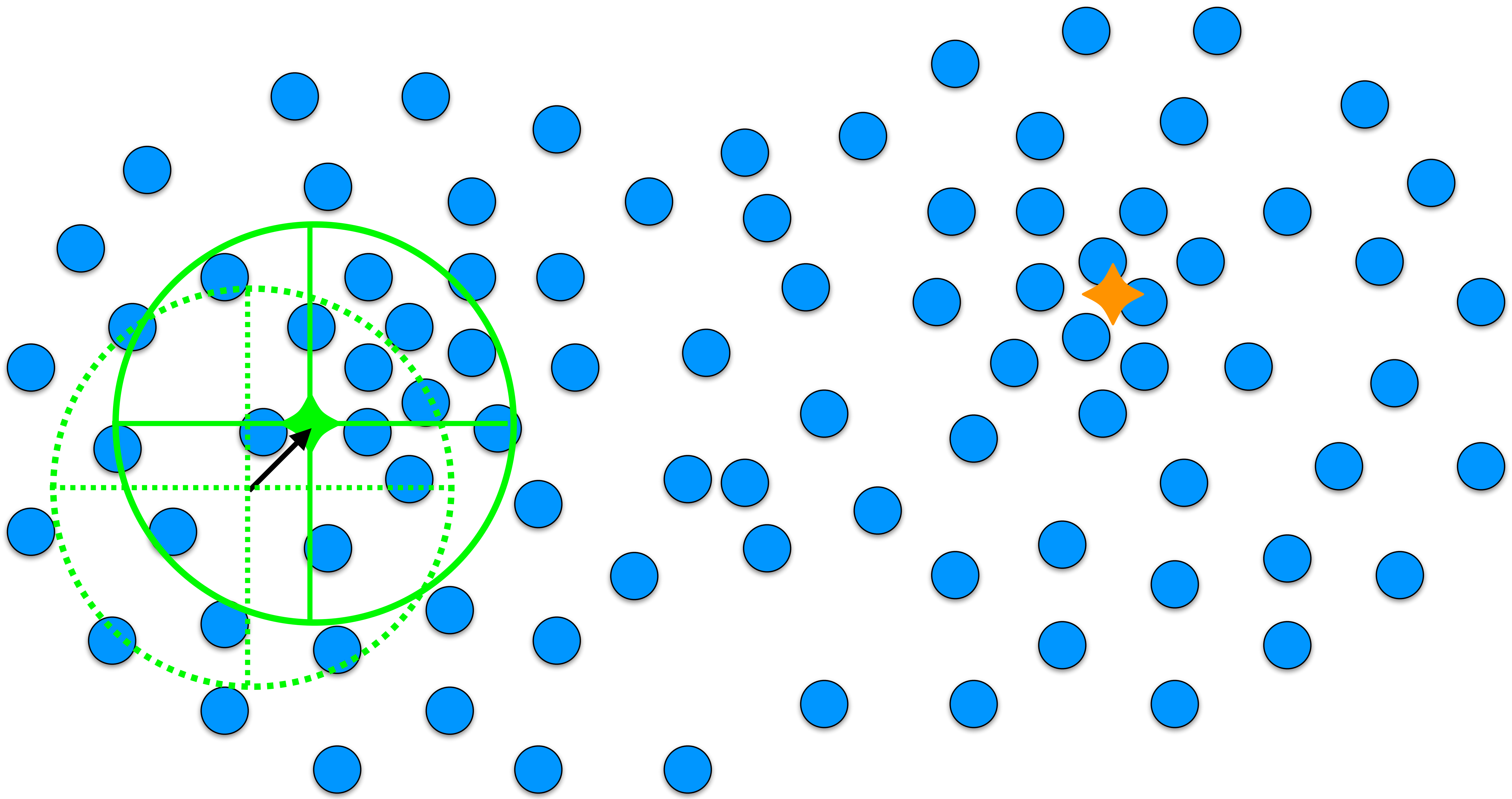
Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood
   ...
6. Repeat until convergence



Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood
   ...
6. Repeat until convergence

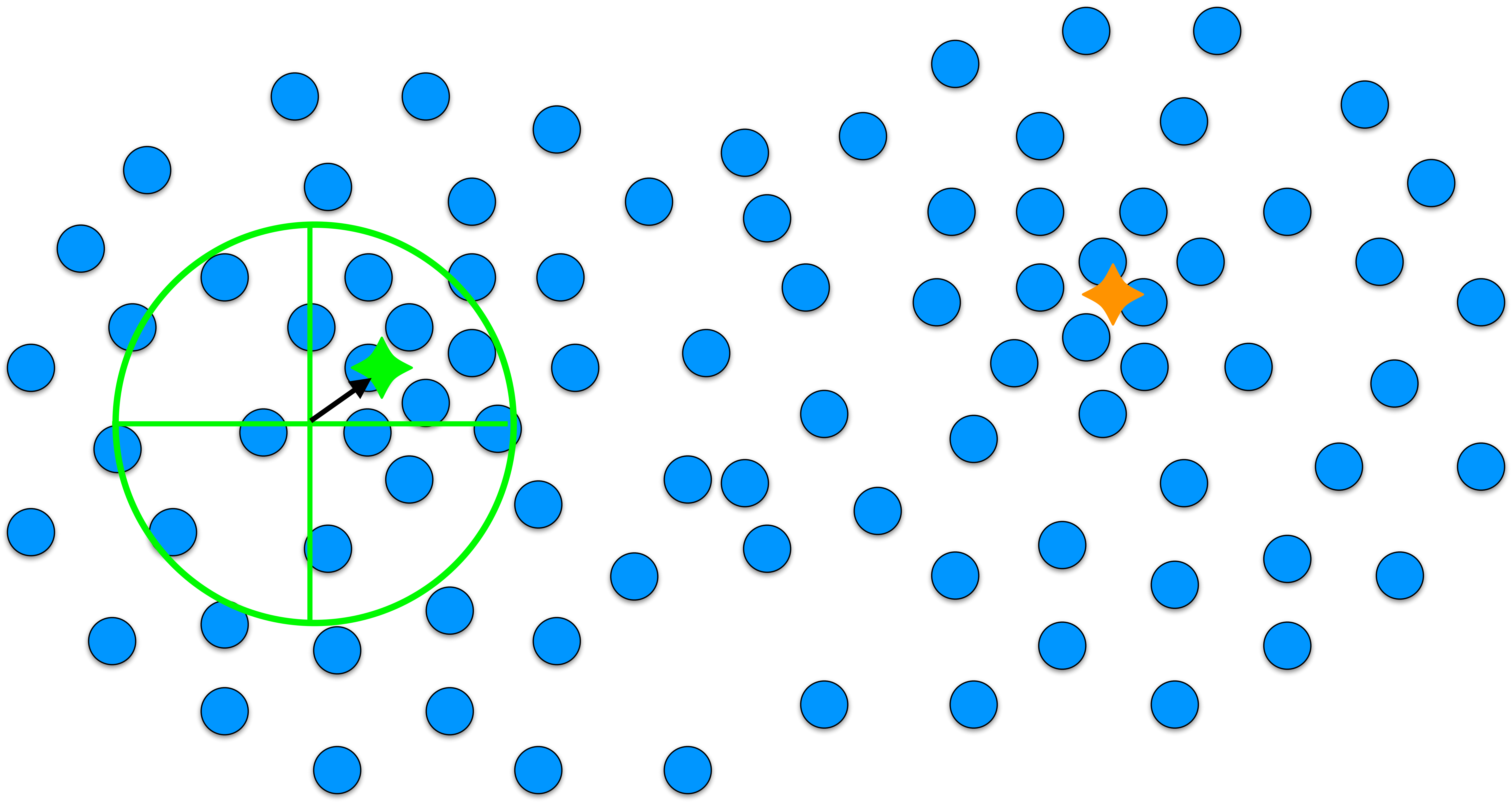


Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

   ...

6. Repeat until convergence

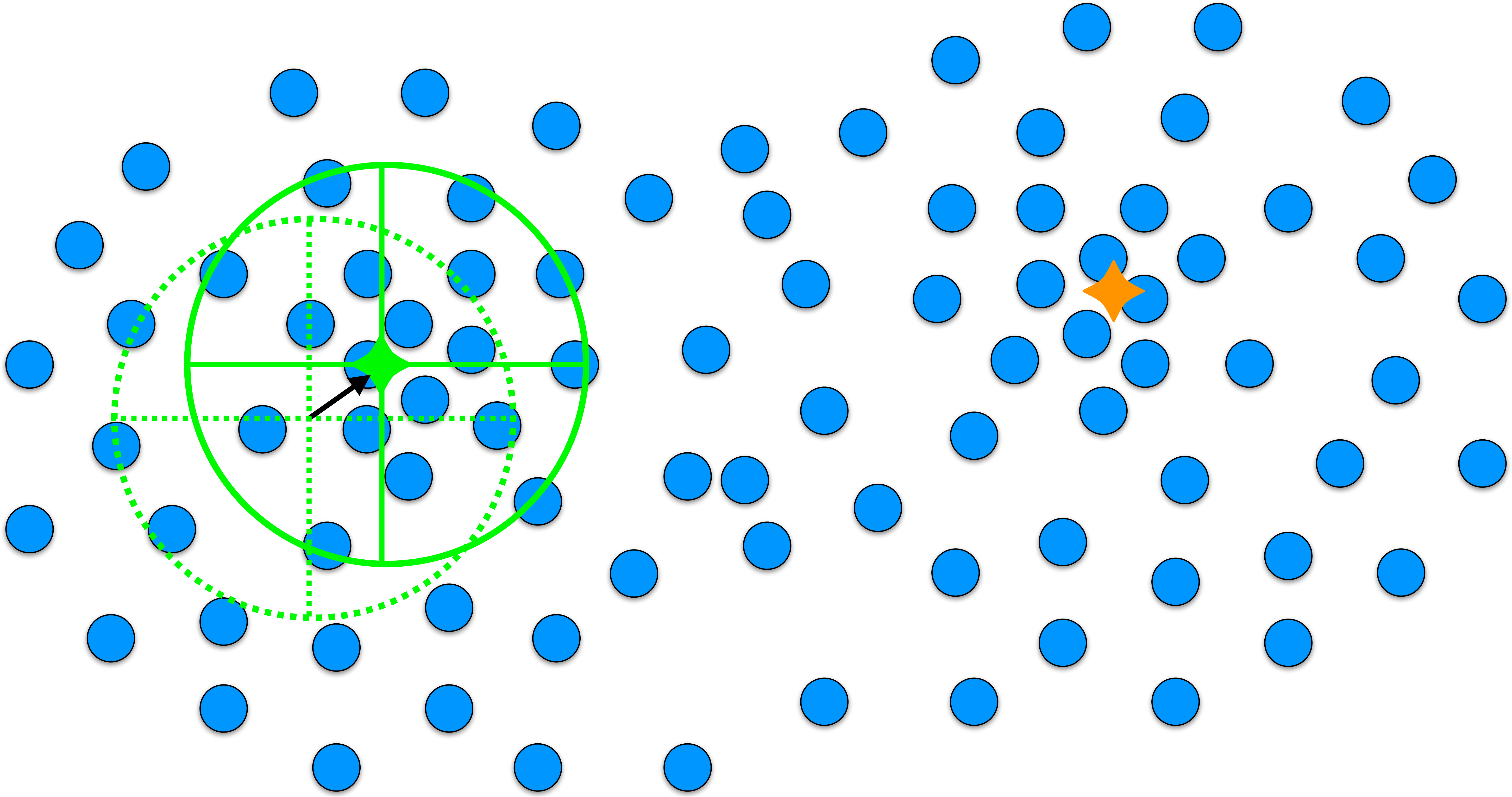


Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

...

6. Repeat until convergence

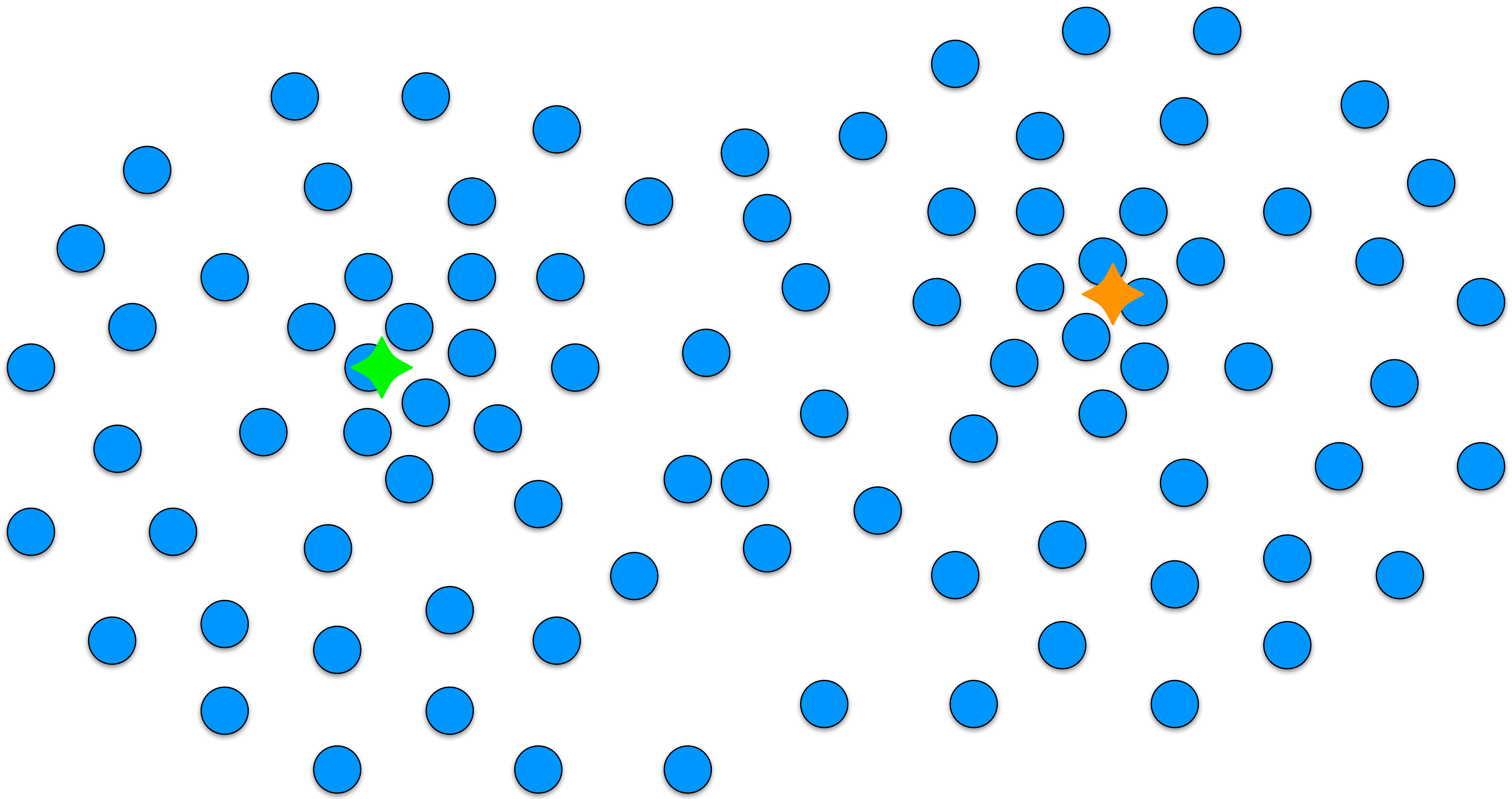


Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

   …

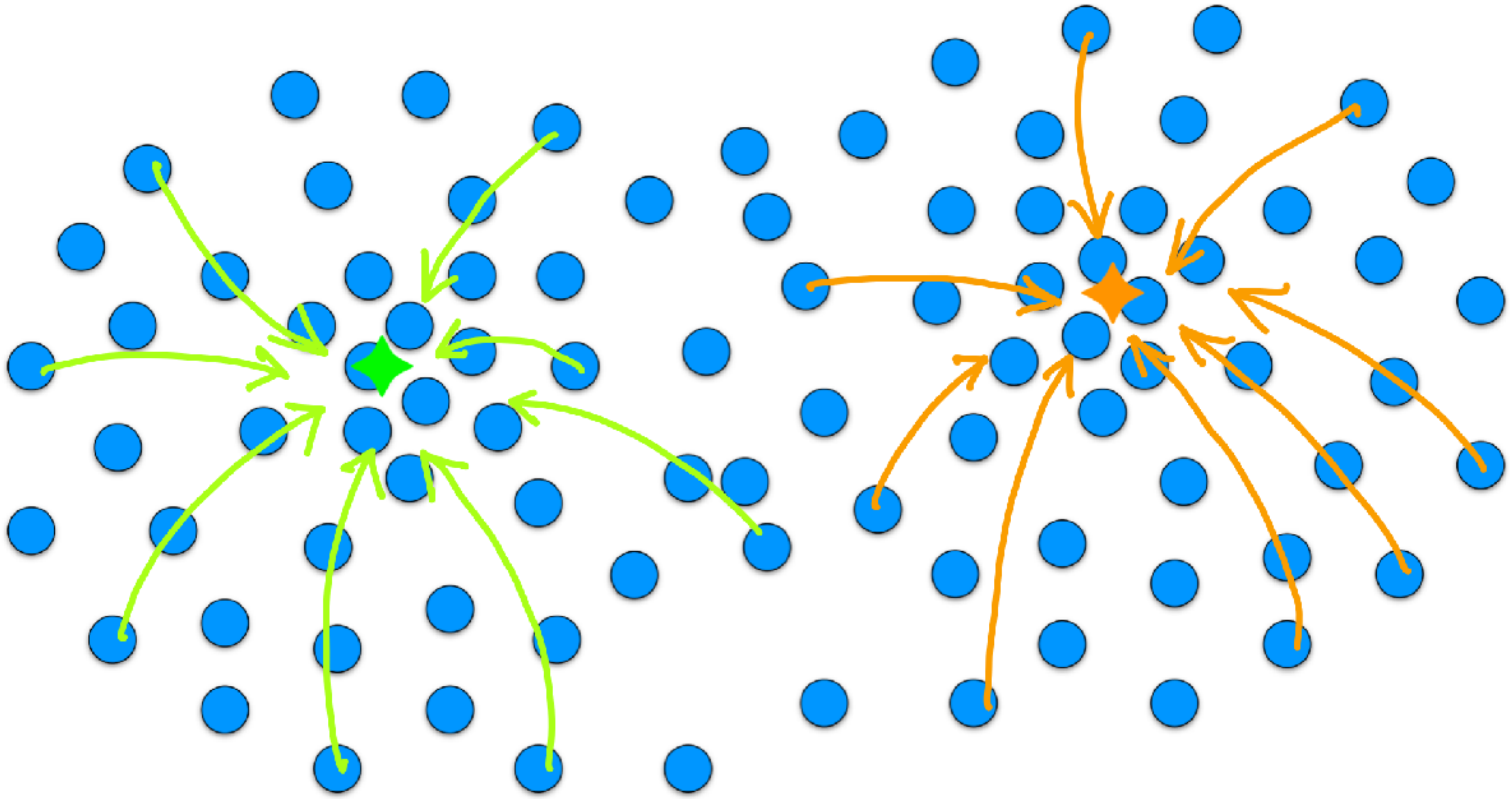
6. Repeat until convergence



Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

   ...
6. Repeat until convergence

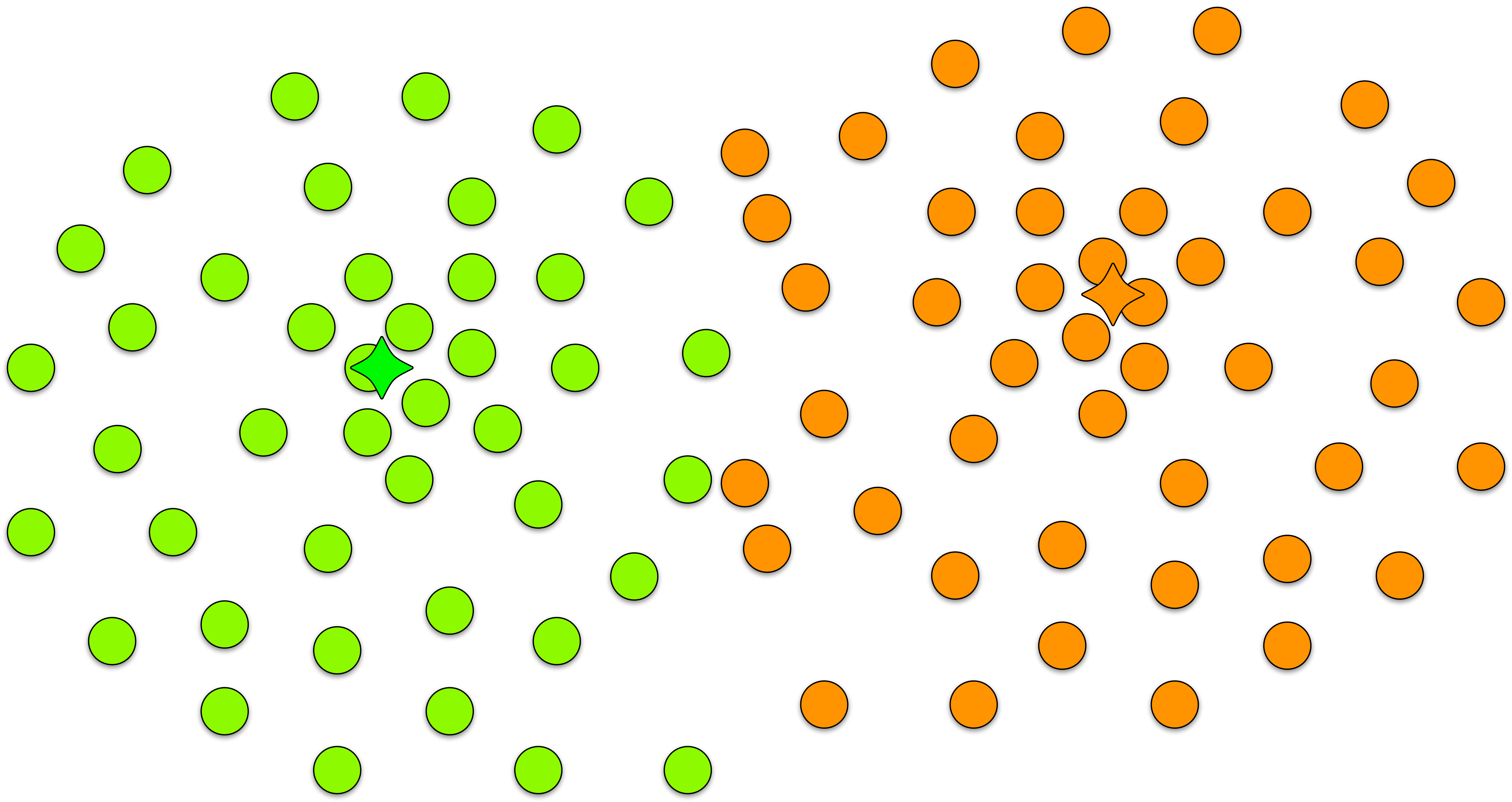


Algorithm

1. Select a point (random)
2. Compute neighborhood
3. Compute mean (centroid)
4. Compute mean-shift
5. Update neighborhood

   ...

6. Repeat until convergence



Seminal Paper

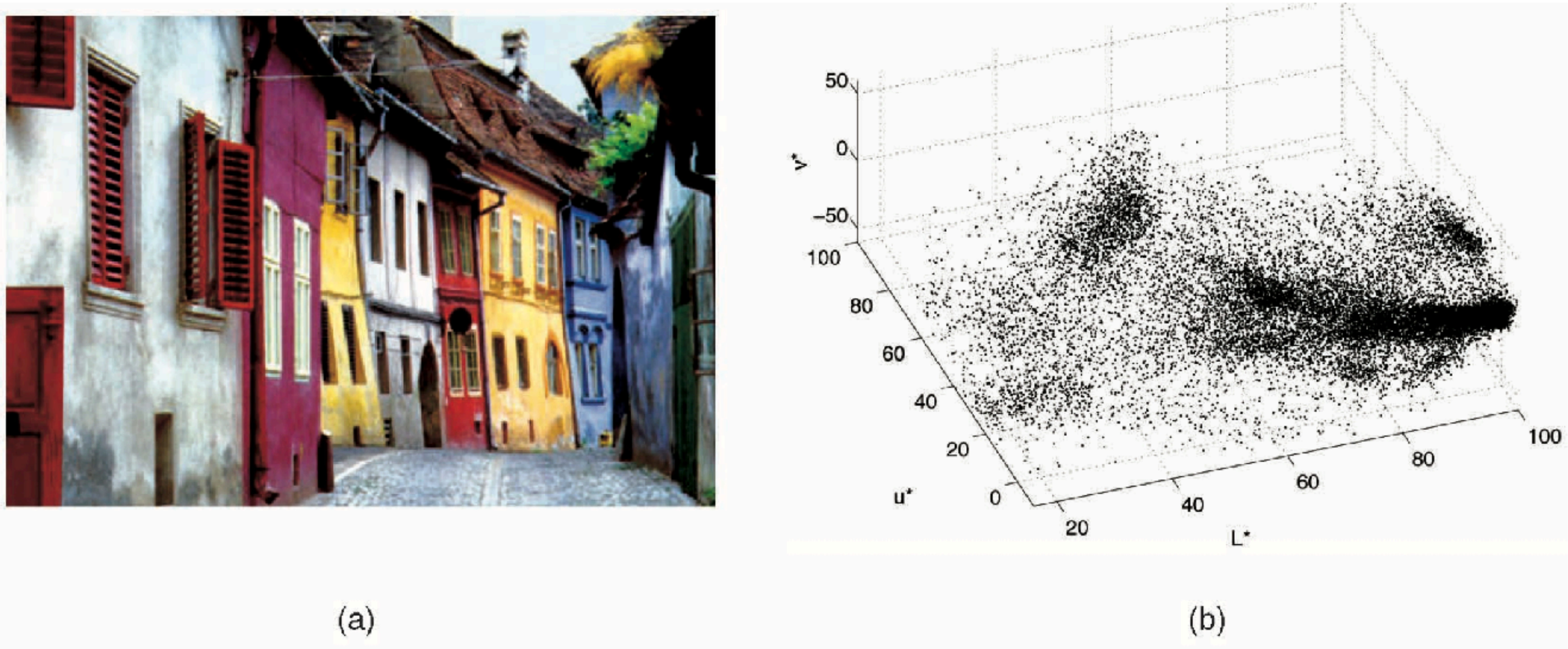
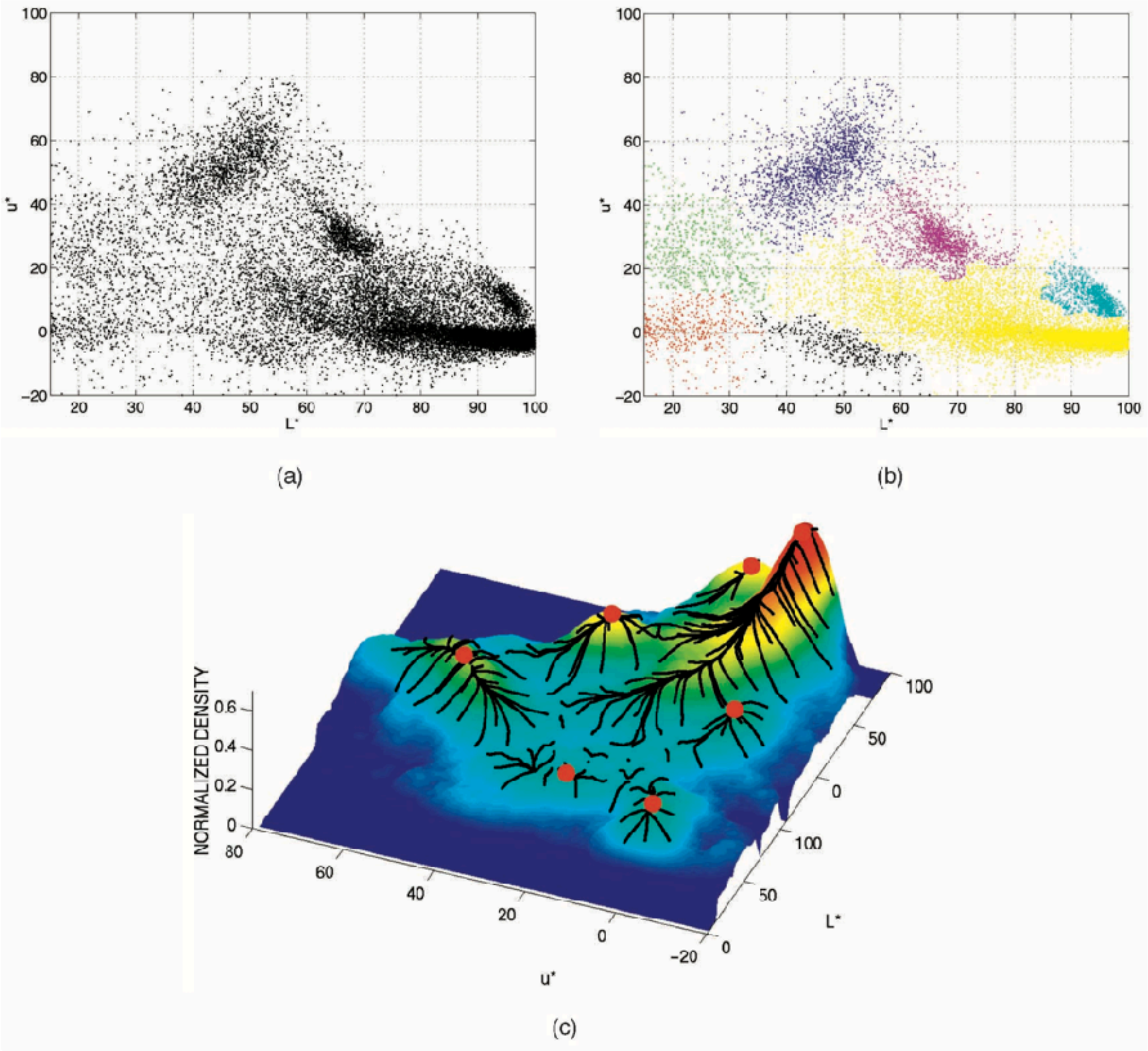


Fig. 1. Example of a feature space. (a) A 400×276 color image. (b) Corresponding L*u*v* color space with $110,400$ data points.

(Comaniciu and Meer, 2002). Mean Shift: A Robust Approach Toward Feature Space Analysis.

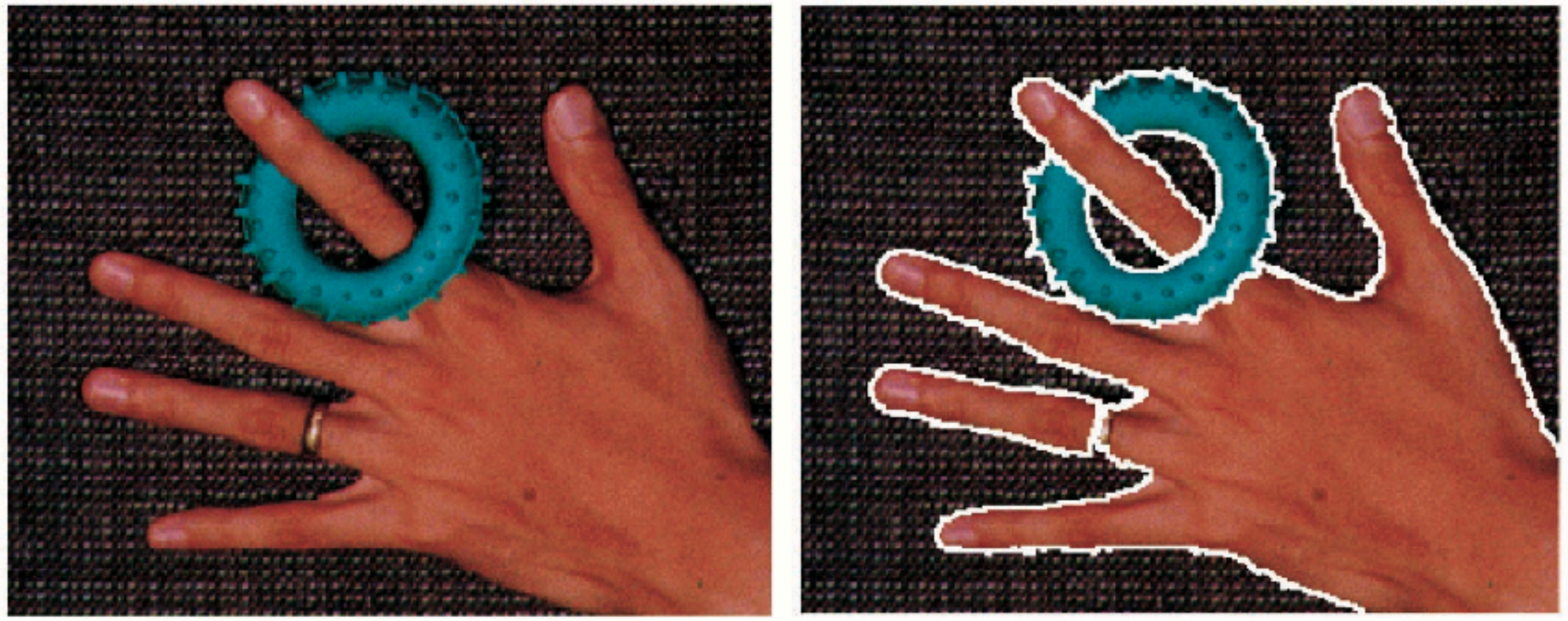
Seminal Paper



(Comaniciu and Meer, 2002). Mean Shift: A Robust Approach Toward Feature Space Analysis.

Seminal Paper

(a) (b)

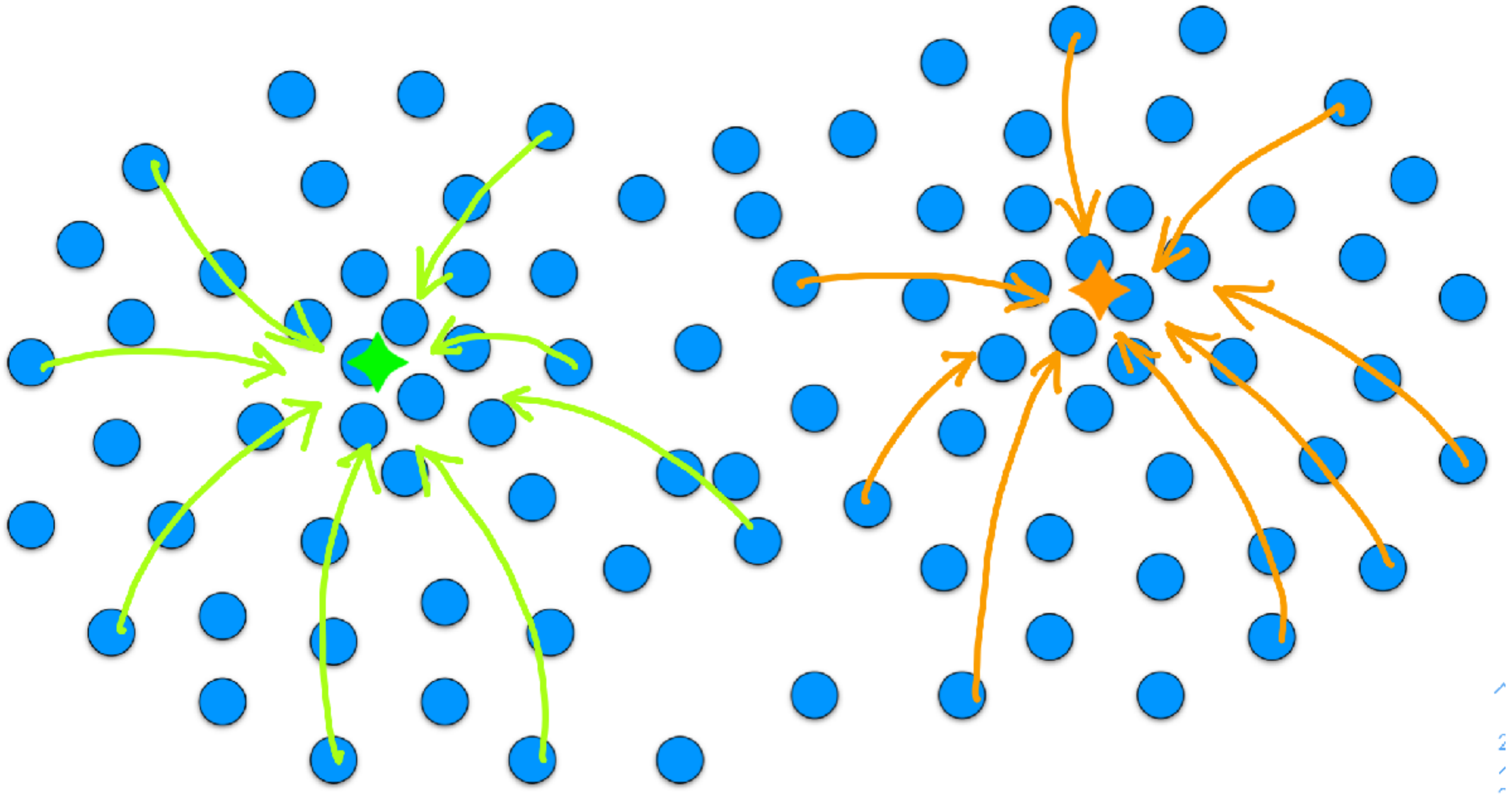


(a) (b)

(Comaniciu and Meer, 2002). Mean Shift: A Robust Approach Toward Feature Space Analysis.

Discussion

- ✓Non-parametric (no k required) algorithm.
- ✓Only needed kernel bandwidth r.
- ✓r can be estimated by grid search guided by a degree of stability of the mean shift direction.
- ✓A Gaussian kernel to compute the centroid weighting the distances of each sample according to it.
- ✓Outliers do not affect clusters shape, but Mean-Shift do not them.



3.2.2. DBScan

DBScan

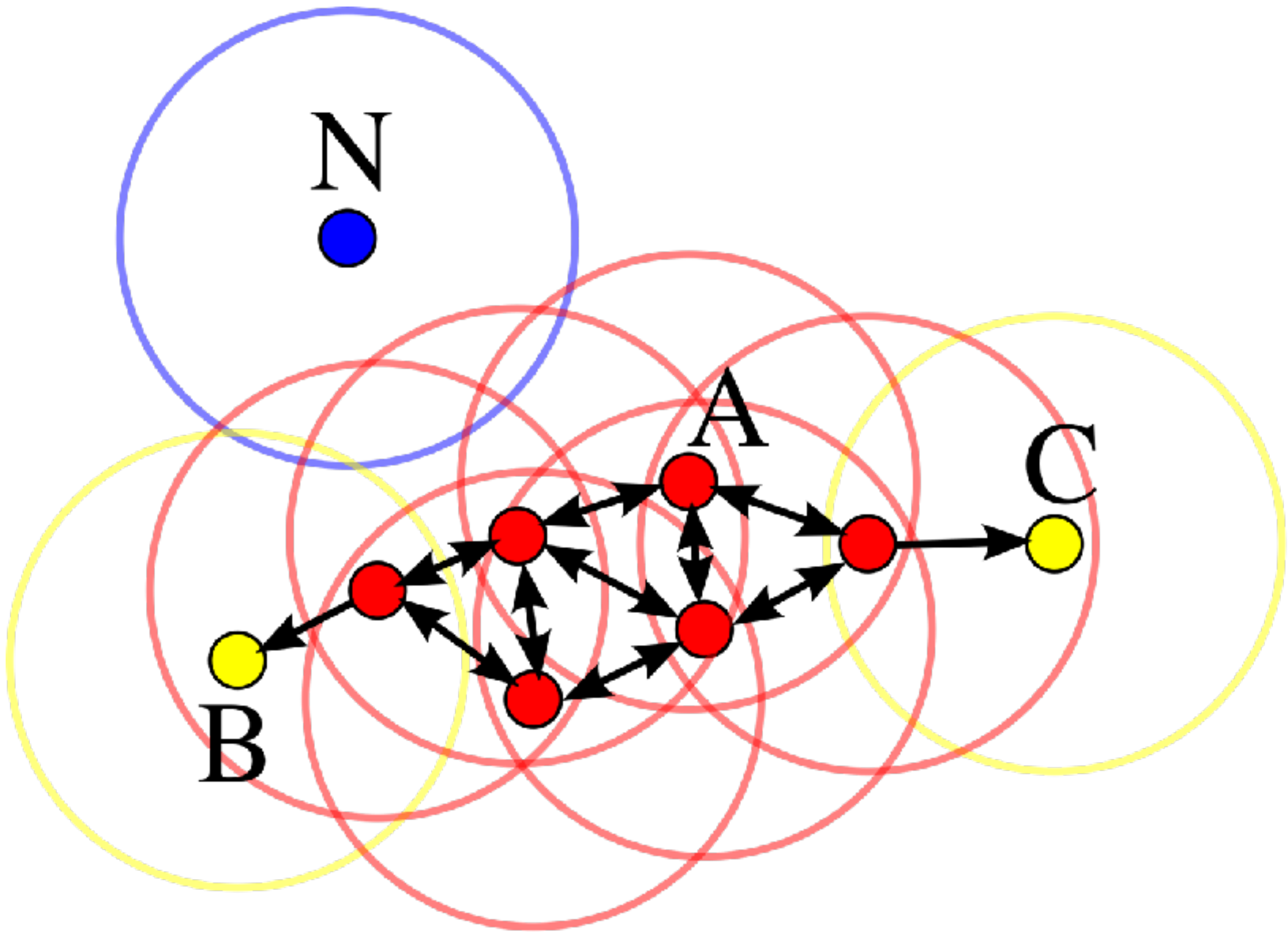
✓(Ester et. al., 1996). A Density-Based Algorithm for Discovering Clusters in LargeSpatial Databases with Noise.

✓Find dense areas and expands recursively to find dense arbitrarily-shaped clusters.

✓Does not requires K.

✓Two main parameters: kernel radius (ϵ) and the minimum number of points that should be contained within a neighborhood (τ).

✓Deals with outliers.



Source: https://es.wikipedia.org/wiki/DBSCAN

DBScan

Introduces the concept of sub-category for points inside and outside a cluster:

- ✓**Core point**: A point that has at least τ (including itself) points within it's radius ϵ.
- ✓**Direct Density Reachable (DDR)**: A point q is DDR from a point p if p is a core point and $q \in N_\epsilon$.
- ✓**Density Reachable (DR)**: Two points are DR if there exists a chain of DDR that links these two points.
- ✓**Noise**: Points that do not belong to the neighbors N_ϵ of any other point.

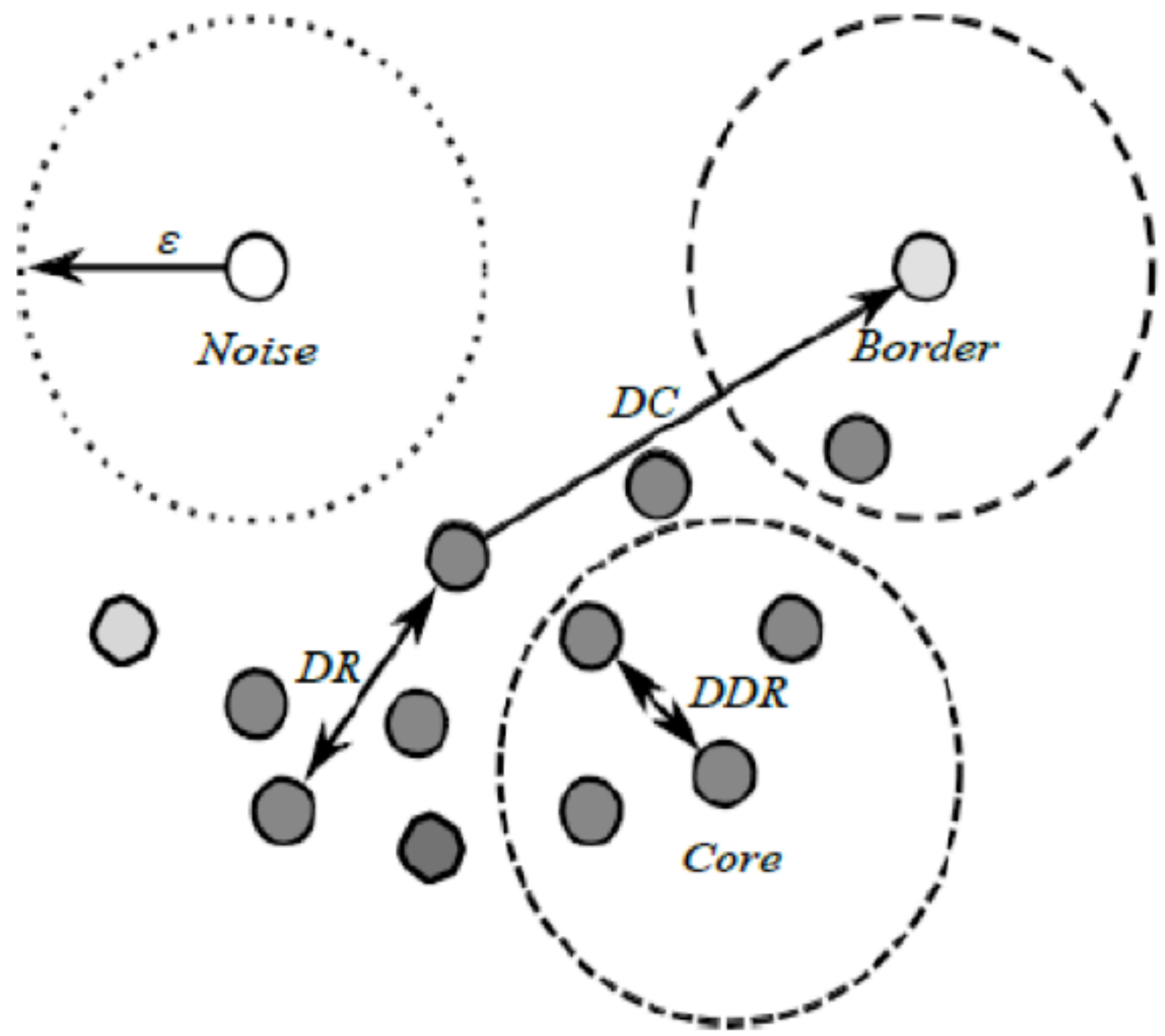


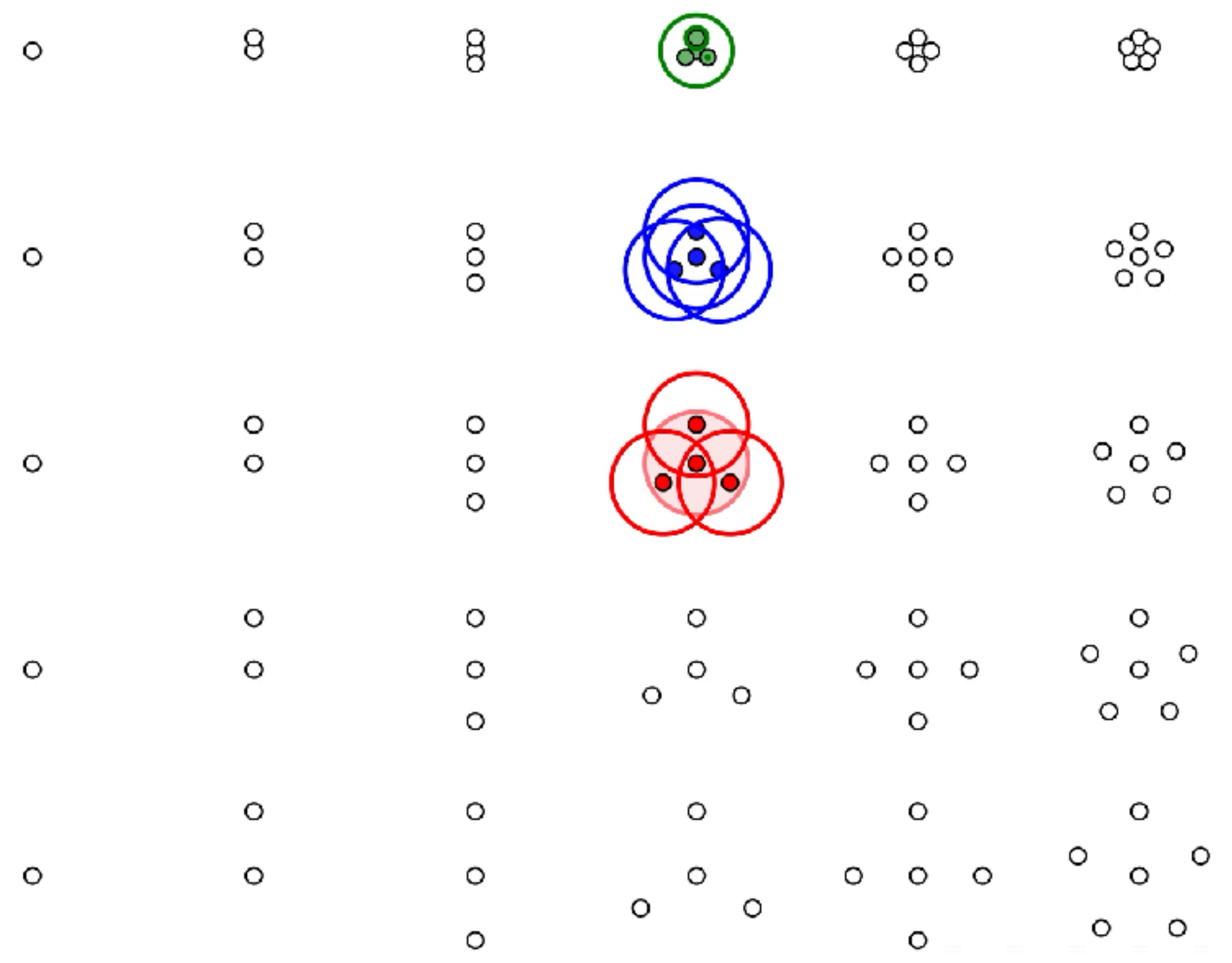
Figure 1: *DBSCAN* **clustering with** $minPoints = 4$

Algorithm

1. Arbitrarily picks a point in dataset X, until all points have been visited.

2. If there are at least τ points within a radius ϵ from the current point, all these points are considered inside the same cluster.

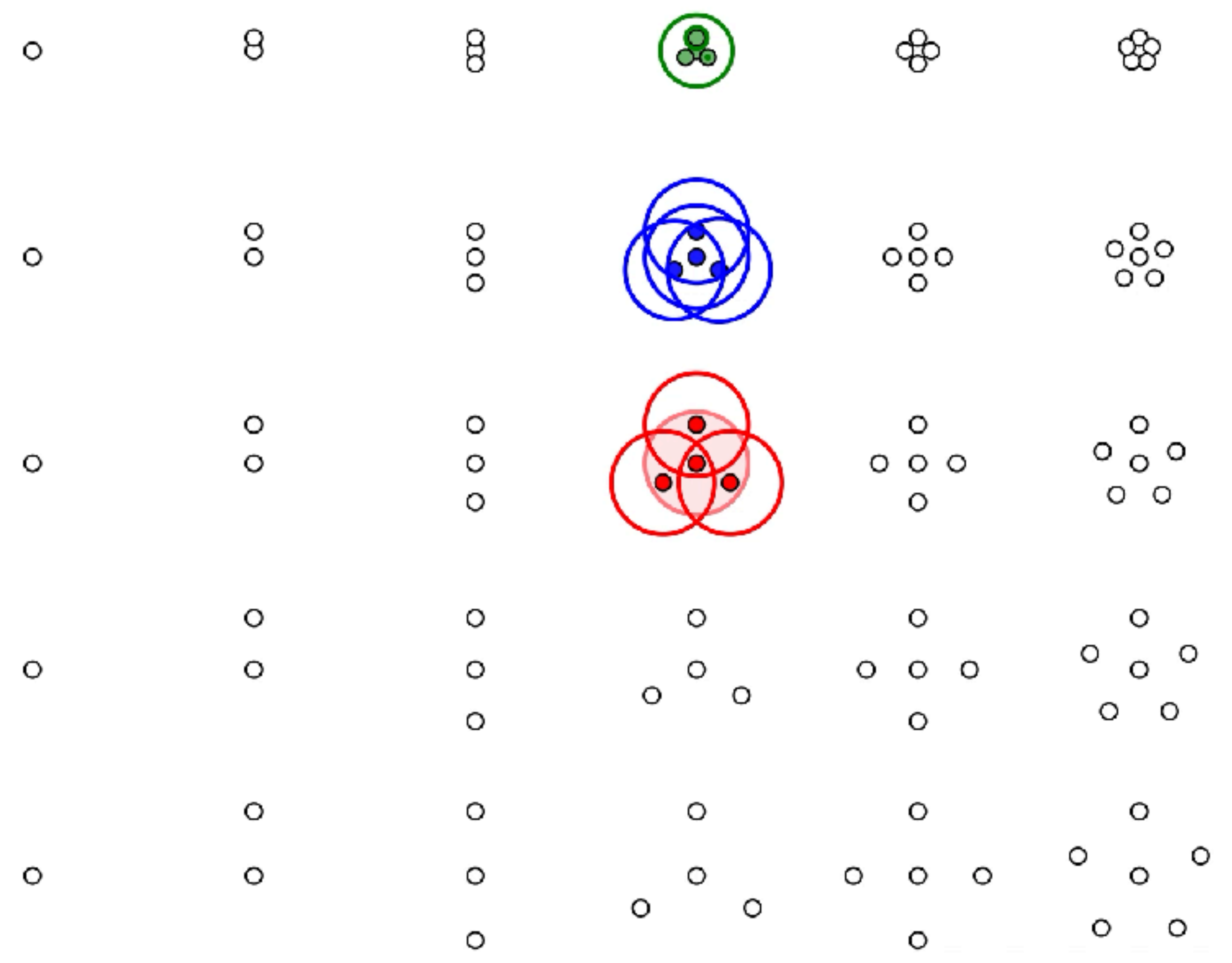
3. Expand each cluster by recursively call the latter for each neighbor point.



Source: https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

Algorithm

```
DBSCAN(DB, distFunc, eps, minPts) {
    C := 0
    for each point P in database DB {
        if label(P) ≠ undefined then continue
        Neighbors N := RangeQuery(DB, distFunc, P, eps)
        if |N| < minPts then {
            label(P) := Noise
            continue
        }
        C := C + 1
        label(P) := C
        SeedSet S := N \ {P}
        for each point Q in S {
            if label(Q) = Noise then label(Q) := C
            if label(Q) ≠ undefined then continue
            label(Q) := C
            Neighbors N := RangeQuery(DB, distFunc, Q, eps)
            if |N| ≥ minPts then {
                S := S ∪ N
            }
        }
    }
}
```



Source: https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

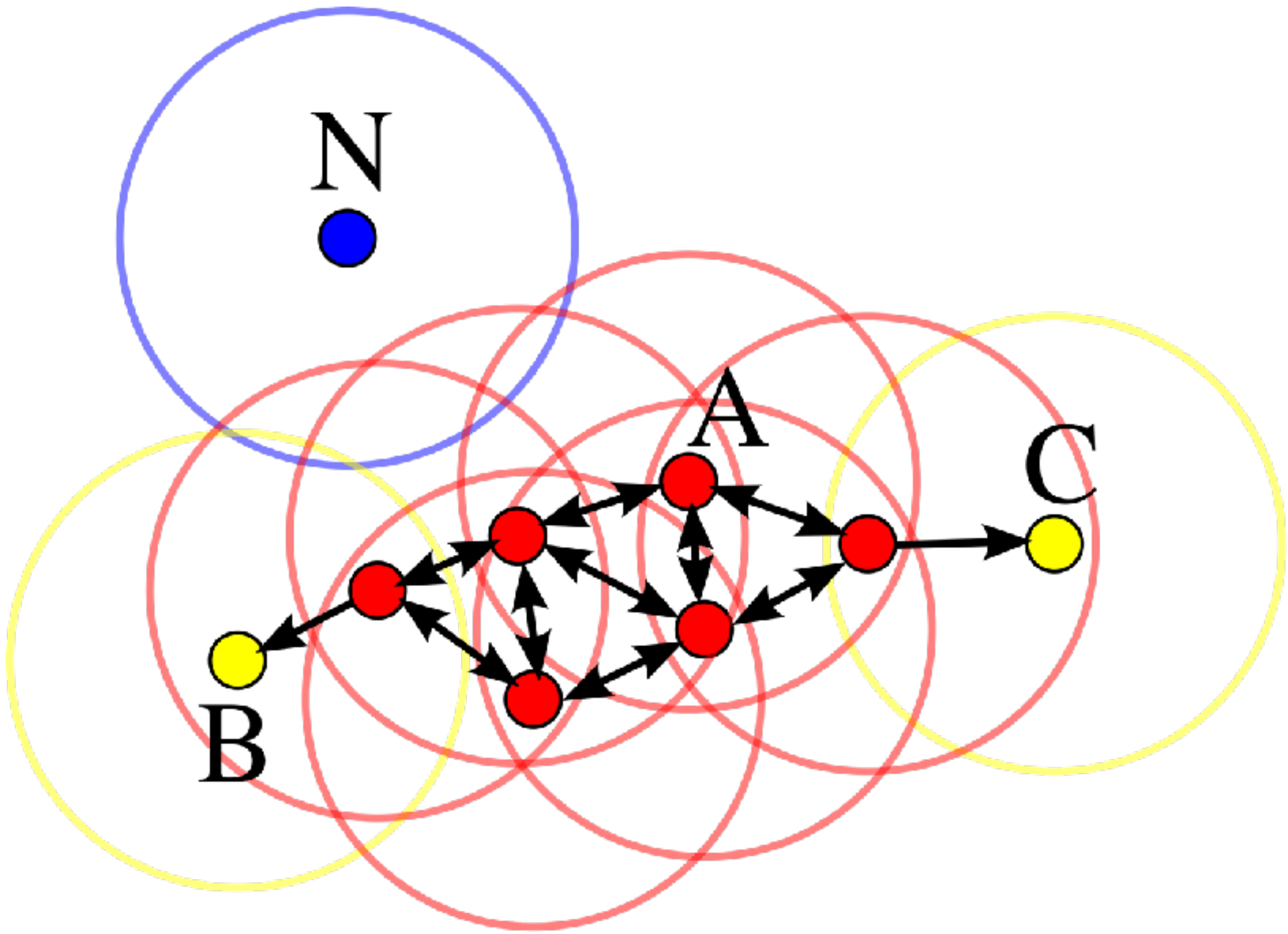
Discussion

1. Complexity is $O(n^2)$ although parallelized versions such as HPDBScan can have $O(nlog(n))$.

2. Not robust when datasets have highly variable density.

3. Does not require to specify the number of clusters k.

4. Works well for arbitrarily-shaped clusters.



Source: https://es.wikipedia.org/wiki/DBSCAN

4. Python Practice: Image Segmentation

Clustering and Image Segmentation

[Clustering with scikit-learn]

[Image Segmentation with Clustering]

4. Conclusions

Conclusions

- Unsupervised learning, specifically clustering techniques, are powerful tools to analyze data and support decision making processes.
- Image segmentation is an active field of computer vision and artificial intelligence where clustering techniques are used.
- Although K-Means has some disadvantages (outliers, clusters initialization) it is widely used due to its computational simplicity.
- For spatial data, density-based clustering techniques are better in performance.
- Performance in clustering tasks, are not as measurable as in supervised learning, so performance is problem and expert dependent.