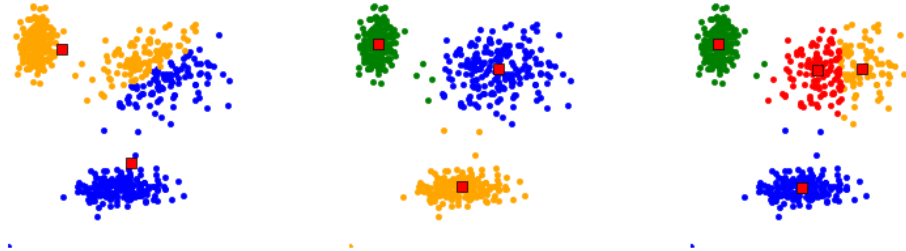
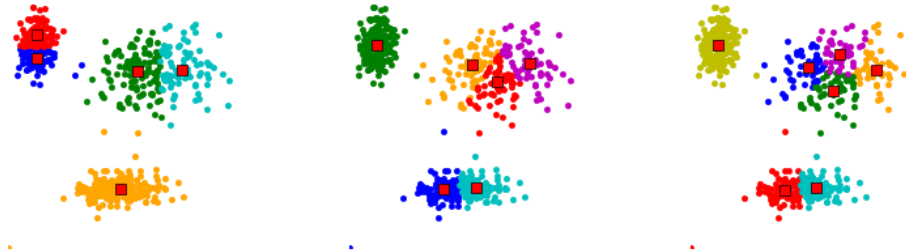


Fuzzy C-means (FCM) Clustering Algorithm

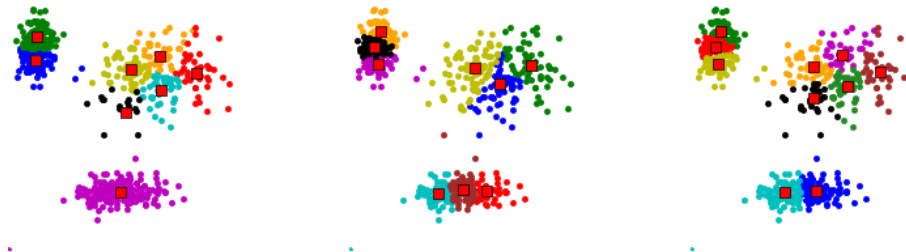
Centers = 2; FPC = 0.79 Centers = 3; FPC = 0.88 Centers = 4; FPC = 0.81



Centers = 5; FPC = 0.72 Centers = 6; FPC = 0.71 Centers = 7; FPC = 0.69



Centers = 8; FPC = 0.64 Centers = 9; FPC = 0.58 Centers = 10; FPC = 0.58



출처: https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_cmeans.html

1. FCM 개요

- 퍼지 군집(Fuzzy Clustering)에서는 퍼지 이론에 기반해서 각 관측치가 여러 군집에 동시에 속할 수 있음
- 각 군집 별로 속할 가능성(degrees of possibility)을 제시

Hard Clustering (ex : K-centroid clustering)				vs.	Soft Clustering (ex : Fuzzy clustering)			
	Cluster 1	Cluster 2	Cluster 3			Cluster 1	Cluster 2	Cluster 3
Obs. 1	1	0	0		Obs. 1	0.8	0.15	0.05
Obs. 2	0	0	1		Obs. 2	0.05	0.05	0.9
Obs. 3	0	0	1		Obs. 3	0.3	0.1	0.6
Obs. 4	0	1	0		Obs. 4	0.0	0.9	0.1
Obs. 5	1	0	0		Obs. 5	0.75	0.05	0.2
Obs. 6	⋮				Obs. 6	⋮		

→ "관측치3번은 군집3에 속함. 끝."

"관측치3번은 군집1에 속할 가능성이 30%, 군집2에 속할 가능성(possibility)이 10%, 군집3에 속할 가능성이 60%임"

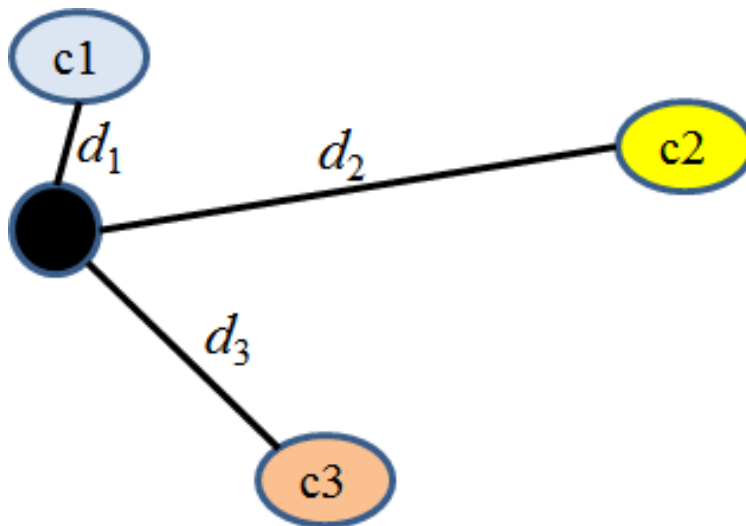
[R 분석과 프로그래밍] <http://rfriend.tistory.com>

1. FCM 개요

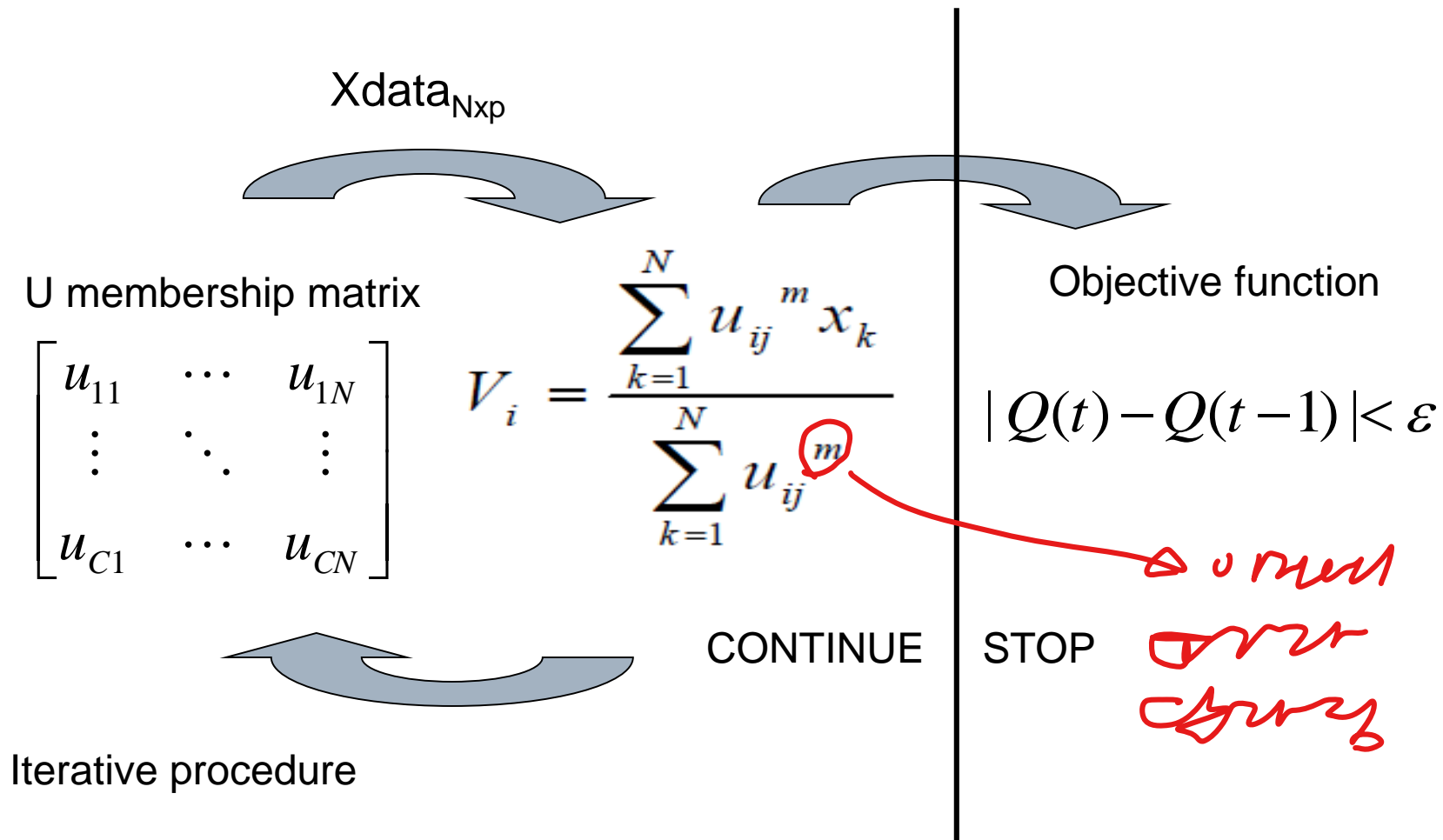
- 퍼지 군집 알고리즘으로 가장 많이 사용되는 것
 - ✓ **Fuzzy C-means(FCM) Clustering Algorithm**
- FCM은 1973년 J. C. Dunn이 개발하였고, 1981년 J. C. Bezdek 이 발전시켰음
- FCM algorithm은 K-means algorithm과 매우 유사
- 군집 내 관측치들 간의 유사성을 최대화하고(즉, 군집의 중심과 관측치 간 거리의 합을 최소화), 군집 간 비유사성을 최대화(즉, 군집의 중심 간 거리의 합을 최대화) 하는 최적 해(optimal solution)를 반복적인 연산을 통해 찾는 개념은 동일

1. FCM 개요

- k 개의 군집을 전문가가 사전적으로 지정해 주는 것도 동일
- 유사성의 척도로 거리(distance)를 사용하는 것도 동일
- 다만, 퍼지 군집에서는 각 관측치가 특정 군집에 속할 가능성, 가중치 w 를 계산하는 것이 다름



1. FCM 개요



2. FCM 처리 알고리즘

1. 군집의 개수 K 를 선택함
2. 각 관측치가 특정 군집에 속할 가중치(가능성) 값을 무작위로 할당

3. Repeat

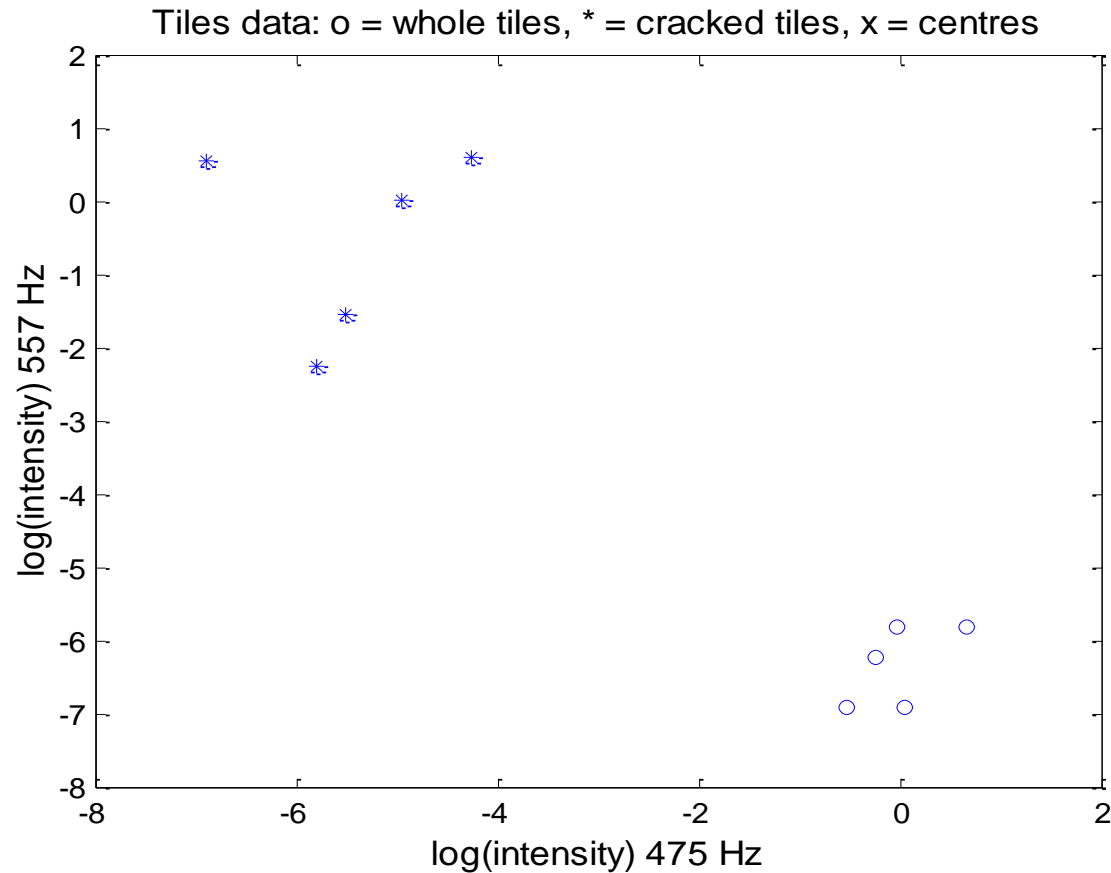
- 각 군집의 중심을 계산함
- 각 관측치에 대해 특정 군집에 속할 가중치(가능성) 값을 다시 계산

4. Until 알고리즘이 수렴할 때

- ✓ 즉, 3번의 2회 반복에서 더 이상 가중치 값의 변화가 주어진 민감도 기준치 미만일 때(또는 각 군집의 중심 변화가 주어진 기준값 미만일 때)

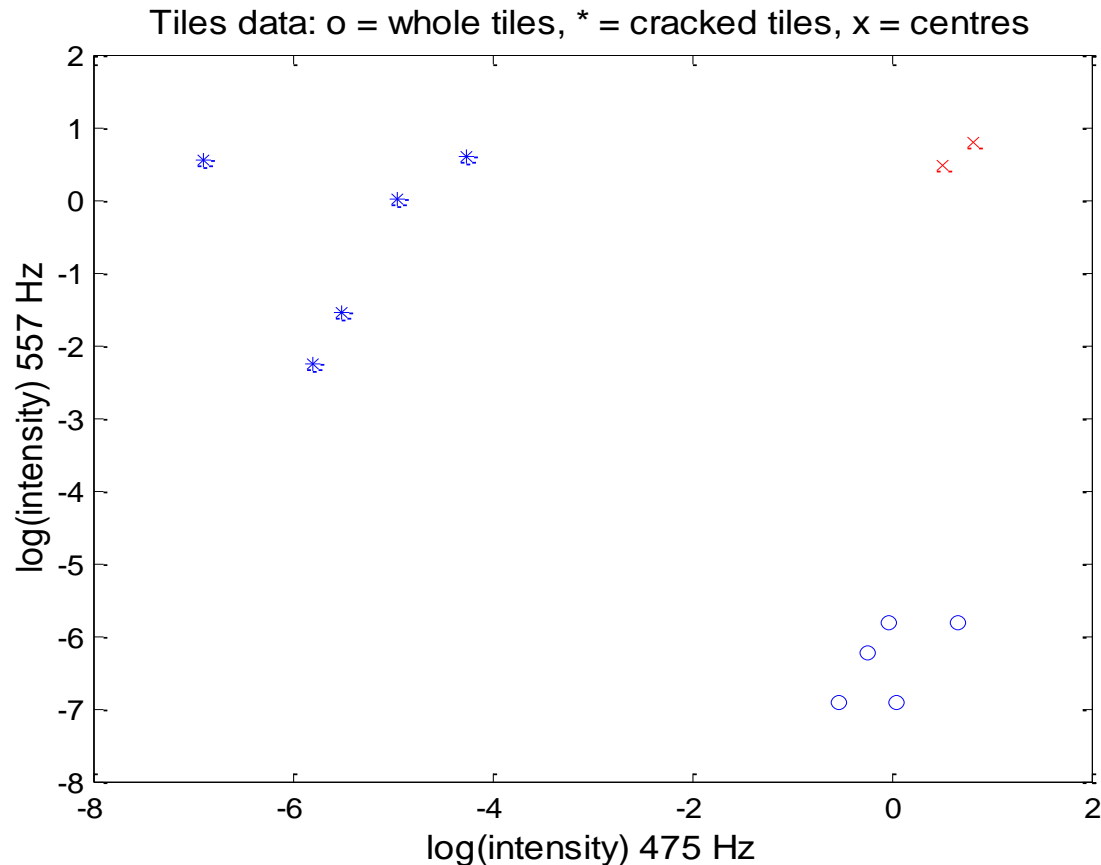
3. FCM 처리 과정

- Each data point belongs to two clusters to different degrees



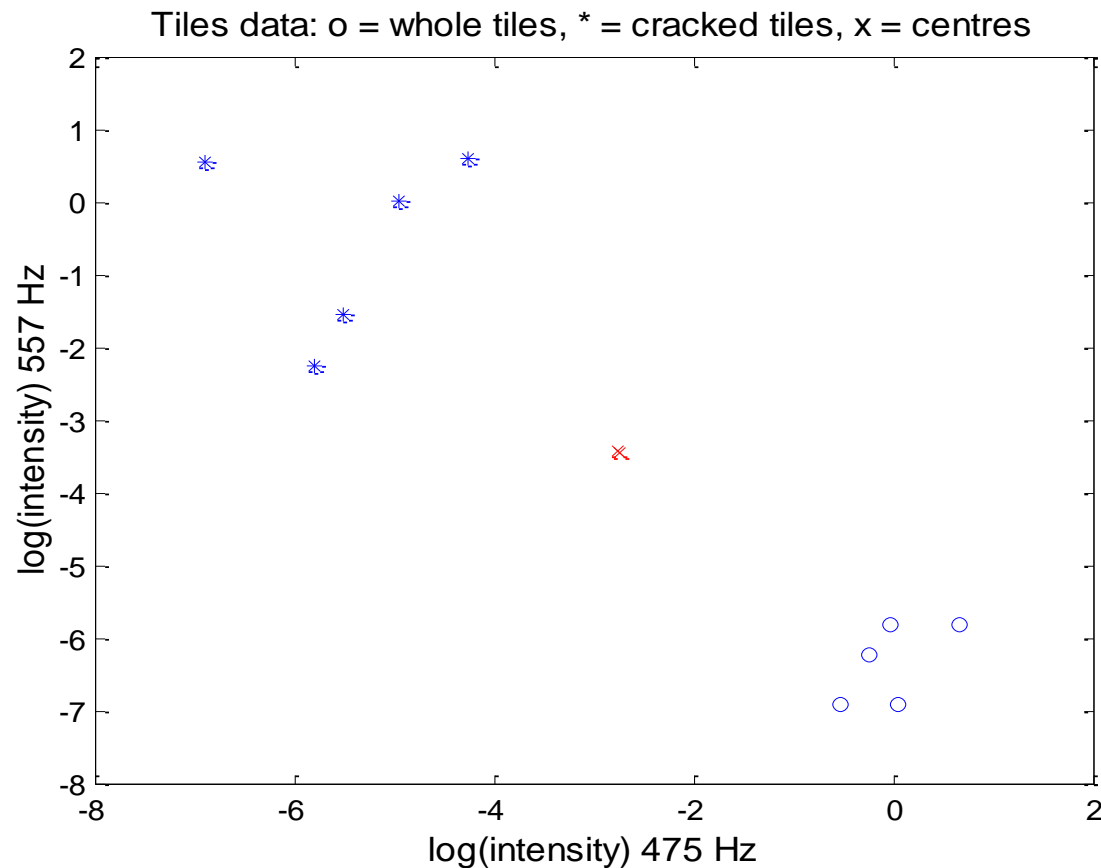
3. FCM 처리 과정

- Place two cluster centers
- Assign **a fuzzy membership** to each data point depending on distance



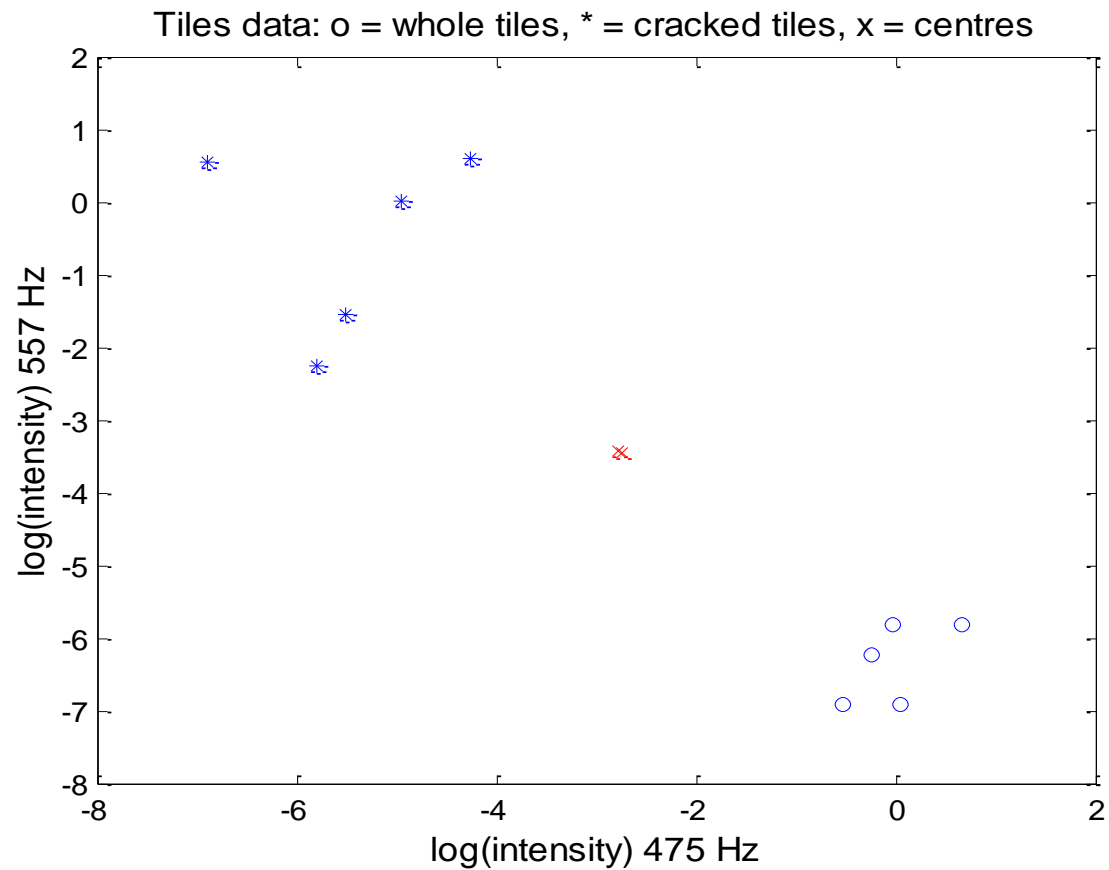
3. FCM 처리 과정

- Compute the new center of each class
- Move the crosses (x)



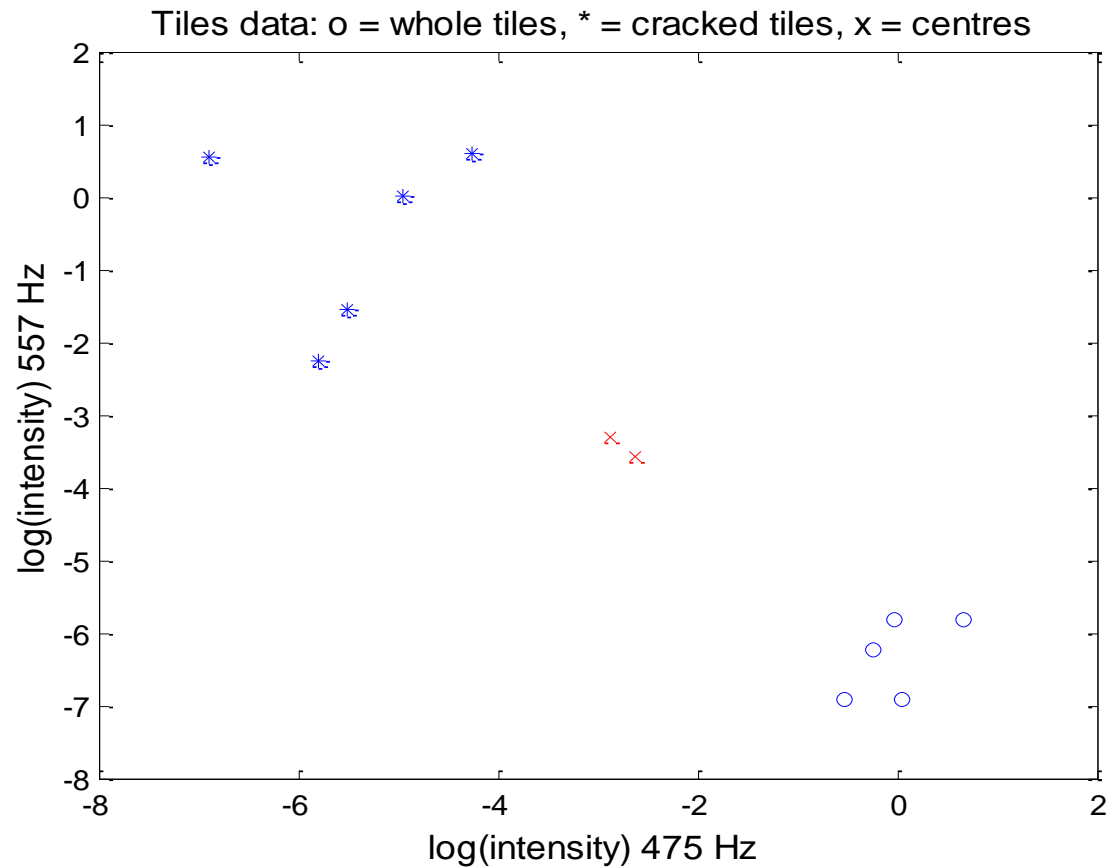
3. FCM 처리 과정

- Iteration 2



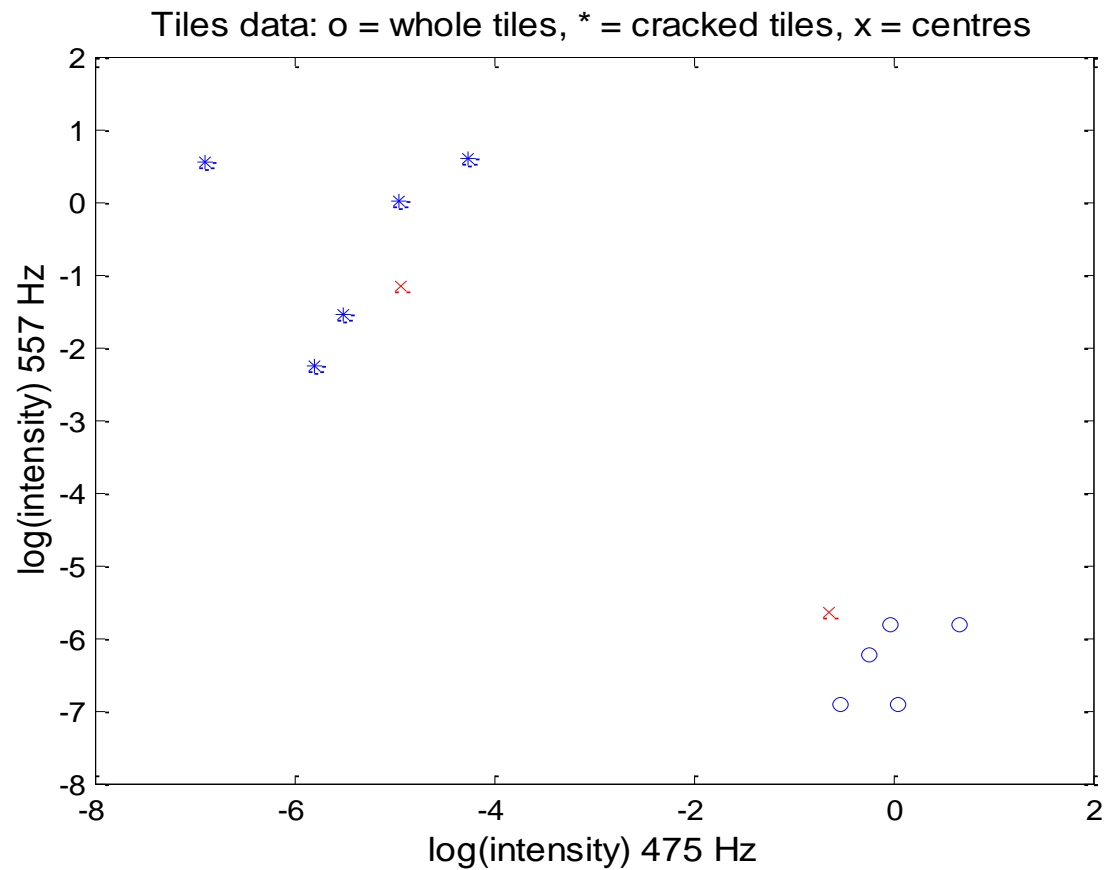
3. FCM 처리 과정

- Iteration 5



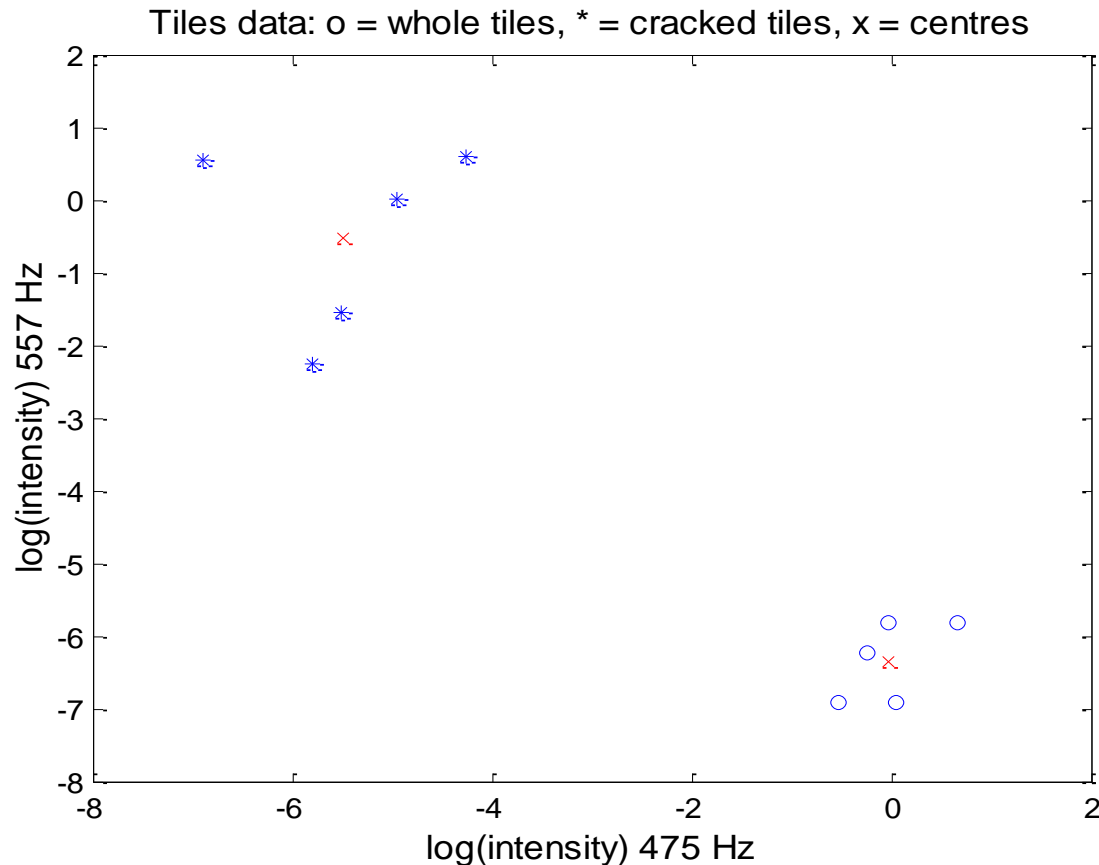
3. FCM 처리 과정

- Iteration 10



3. FCM 처리 과정

- **Iteration 13** (then stop, because no visible change)
- Each data point belongs to the two clusters to a degree



3. FCM 처리 과정

- **Iteration 13** (then stop, because no visible change)
- Each data point belongs to the two clusters to a degree

M =	0.0025	0.9975
	0.0091	0.9909
	0.0129	0.9871
	0.0001	0.9999
	0.0107	0.9893
	0.9393	0.0607
	0.9638	0.0362
	0.9574	0.0426
	0.9906	0.0094
	0.9807	0.0193

4. Fuzzy membership matrix M

Point k 's membership of cluster i

Fuzziness exponent

$$m_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(q-1)}}$$

Distance from point k to current cluster centre i

Distance from point k to other cluster centres j

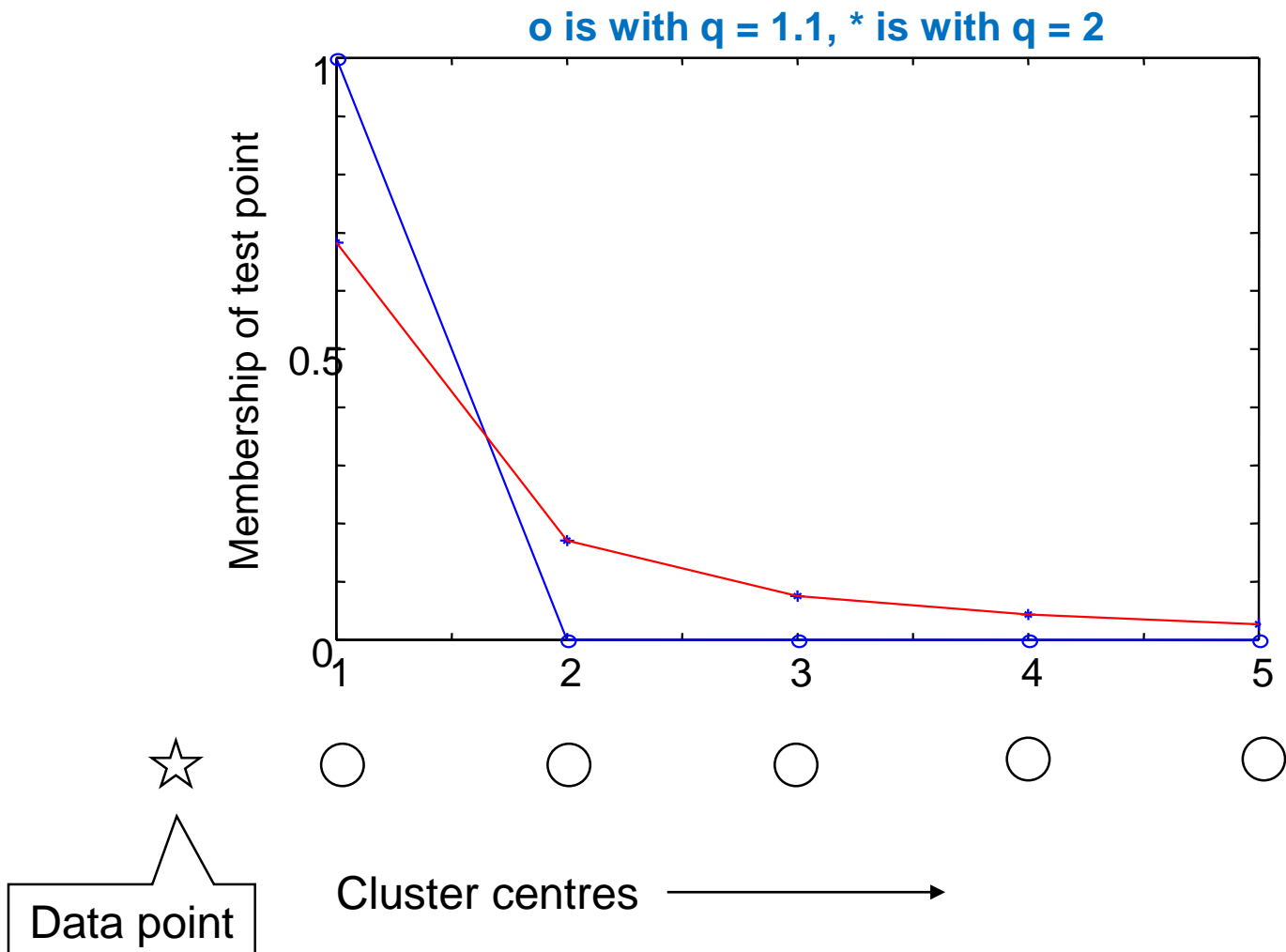
$$d_{ik} = \|\mathbf{u}_k - \mathbf{c}_i\|$$

4. Fuzzy membership matrix M

$$\begin{aligned} m_{ik} &= \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(q-1)}} \\ &= \frac{1}{\left(\frac{d_{ik}}{d_{1k}} \right)^{2/(q-1)} + \left(\frac{d_{ik}}{d_{2k}} \right)^{2/(q-1)} + \dots + \left(\frac{d_{ik}}{d_{ck}} \right)^{2/(q-1)}} \\ &= \frac{\frac{1}{d_{ik}^{2/(q-1)}}}{\frac{1}{d_{1k}^{2/(q-1)}} + \frac{1}{d_{2k}^{2/(q-1)}} + \dots + \frac{1}{d_{ck}^{2/(q-1)}}} \end{aligned}$$

Gravitation to
cluster i relative to
total gravitation

4. Fuzzy membership matrix M



5. Fuzzy c-partition

All clusters C together fill the whole universe U .

Remark: The sum of memberships for a data point is 1, and the total for all points is K

$$\bigcup_{i=1}^c C_i = U$$

$$C_i \cap C_j = \emptyset \quad \text{for all } i \neq j$$

$$\emptyset \subset C_i \subset U \quad \text{for all } i$$

$$2 \leq c \leq K$$

A cluster C is never empty and it is smaller than the whole universe U

Not valid:
Clusters do overlap

There must be at least 2 clusters in a c-partition and at most as many as the number of data points K

6. K-means 실습

◆ IRIS 데이터세트를 이용한 실습

- https://en.wikipedia.org/wiki/Iris_flower_data_set
- <https://rpubs.com/wjholst/322258>
- 3개의 클래스로 각 50개씩, 전체 150개 데이터로 이루어짐
- iris2.dat 파일 참조
- 가장 기본적인 K-means 알고리즘을 C++로 구현
- 초기값 설정 알고리즘은 아래의 두 가지 방법 사용
 - Forgy 기법
 - Random Partition 기법