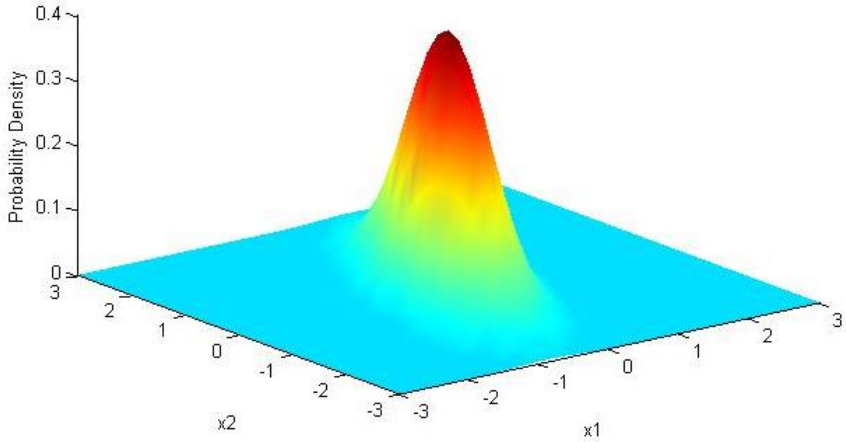
Clustering: Part 2

Taehoon Ko (taehoonko@snu.ac.kr)

Table of Contents

Density estimation approach – Gaussian density estimation

- **Density estimation approach**
  - 데이터 객체 분포를 다변량 통계모형으로 추정
- **Gaussian density estimation**
  - Gaussian distribution = Normal distribution (정규분포)



Gaussian distribution for 2 dimensions

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right],$$

$$\text{where } \boldsymbol{\mu} = \frac{1}{n}\sum_{\mathbf{x}_i \in \mathbf{X}^+} \mathbf{x}_i \text{ is the mean vector and}$$

$$\mathbf{\Sigma} = \frac{1}{n-1}\sum_{\mathbf{x}_i \in \mathbf{X}^+} (\mathbf{x}_i-\boldsymbol{\mu})(\mathbf{x}_i-\boldsymbol{\mu})^T \text{ is the covariance matrix.}$$

Density estimation approach – MoG

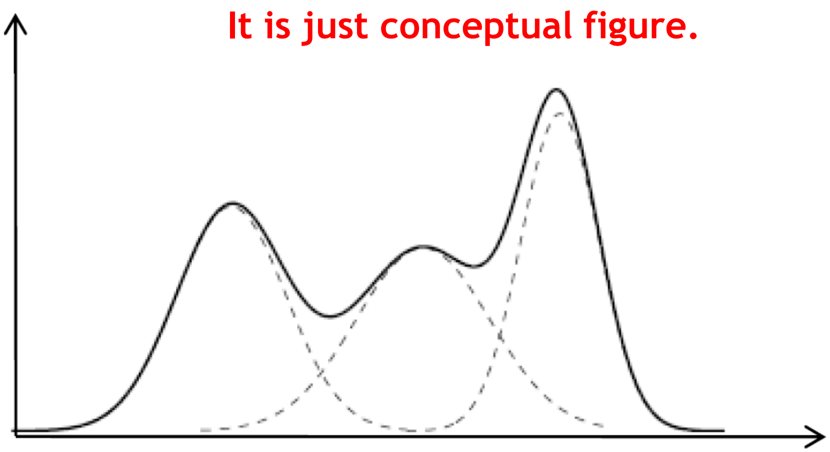
- **Mixture of Gaussian (or Gaussian Mixture Model)**
  - Mixture of *K* Gaussian

$$p(\mathbf{x}) = \sum_{k=1}^{K} P(k) p_k(\mathbf{x}),$$

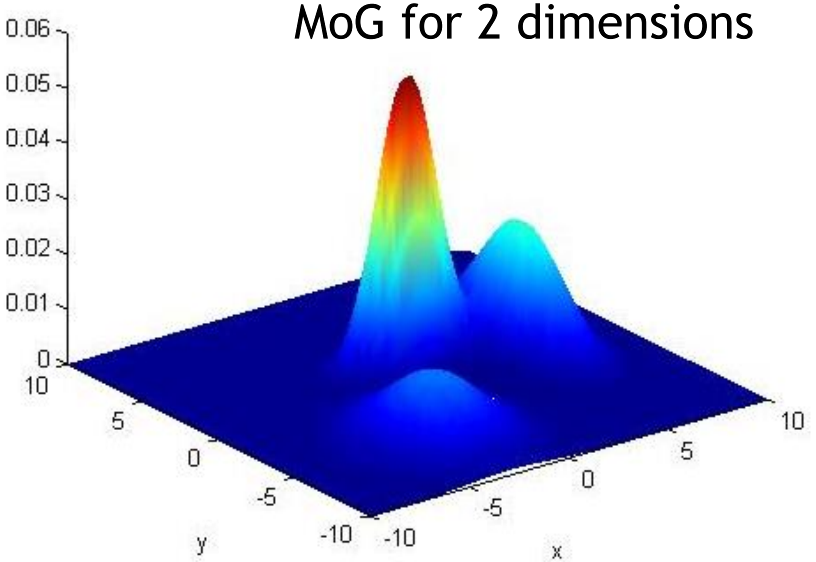
$$p_k(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right]$$

MoG for 1 dimensions

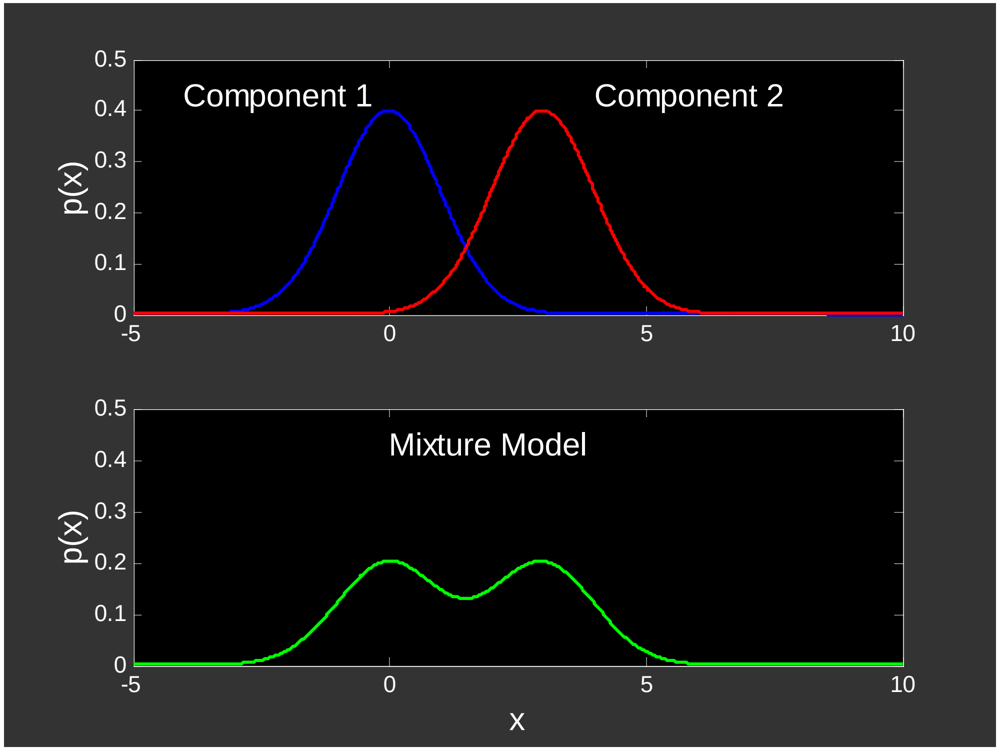
It is just conceptual figure.



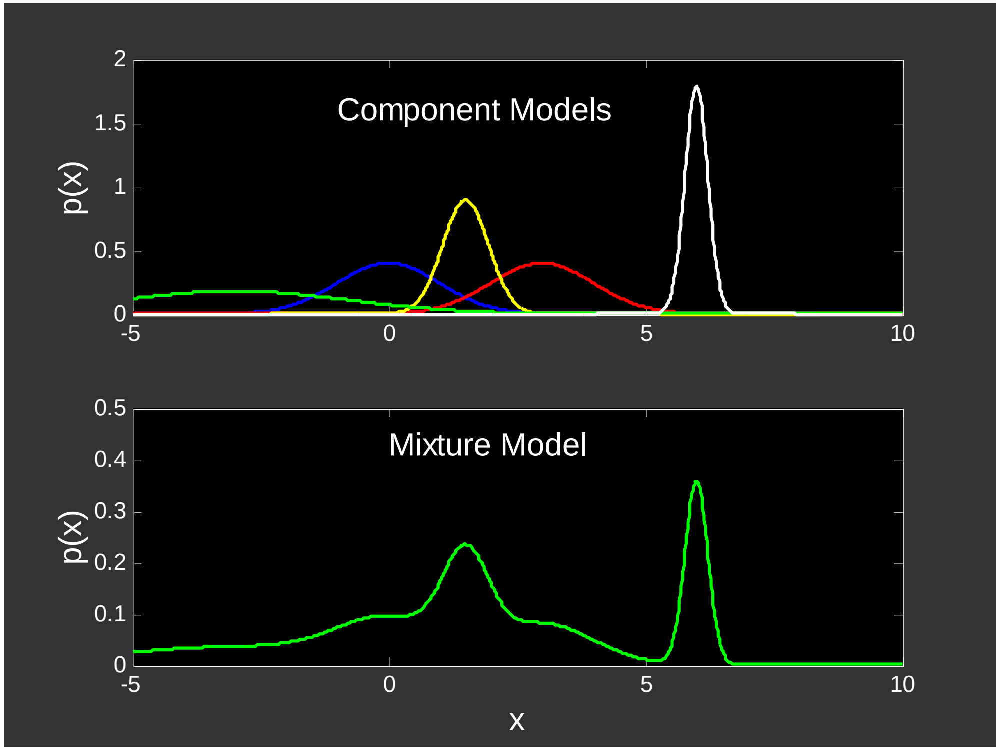
MoG for 2 dimensions



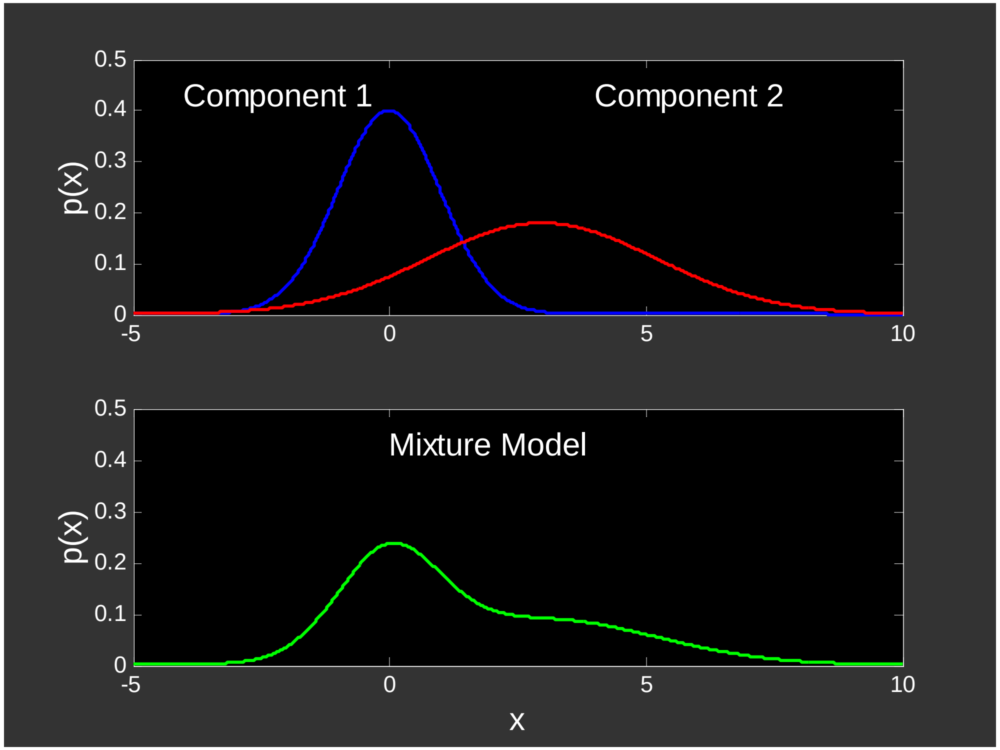
Density estimation approach – MoG



Density estimation approach – MoG



Density estimation approach – MoG

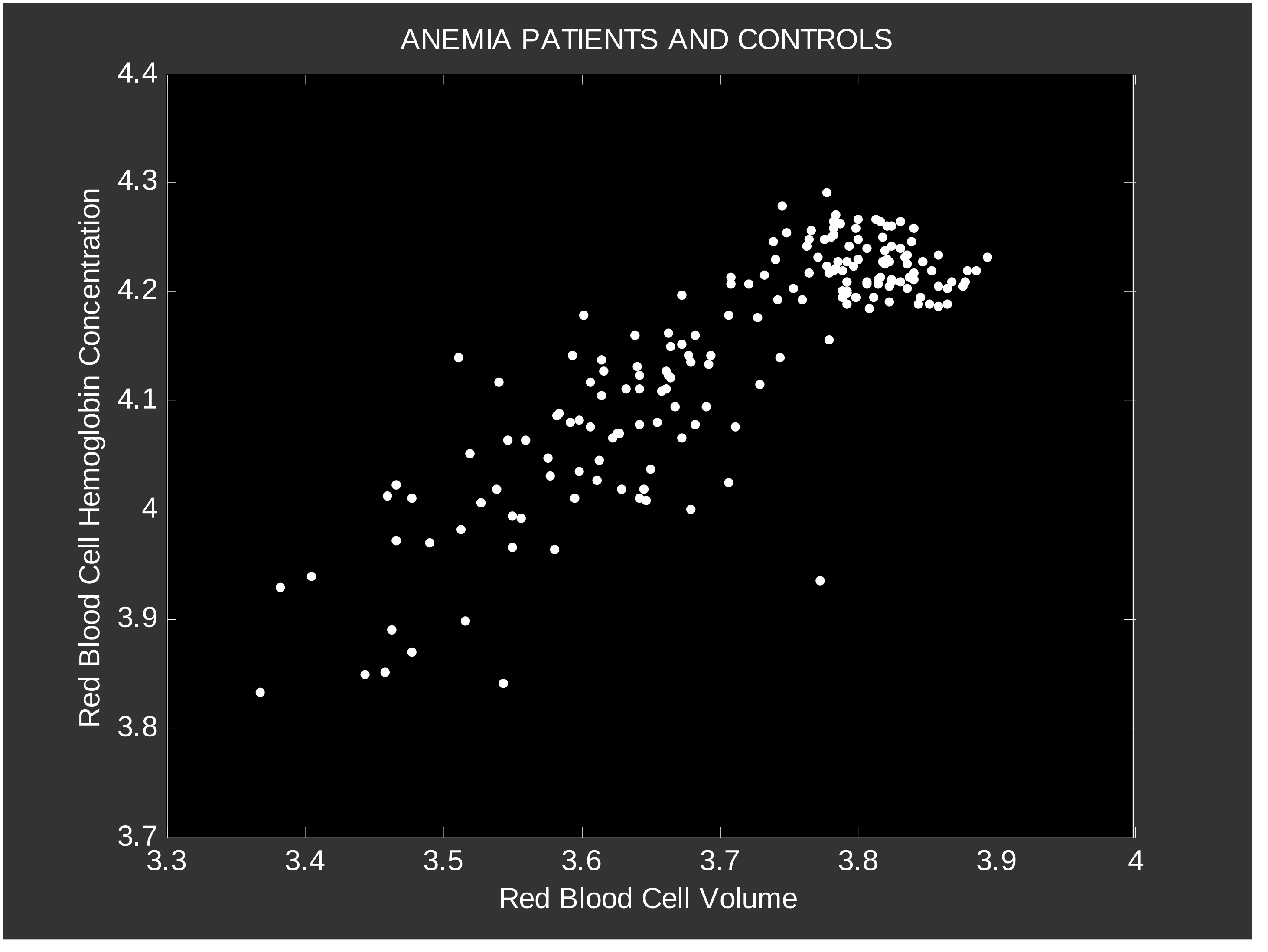


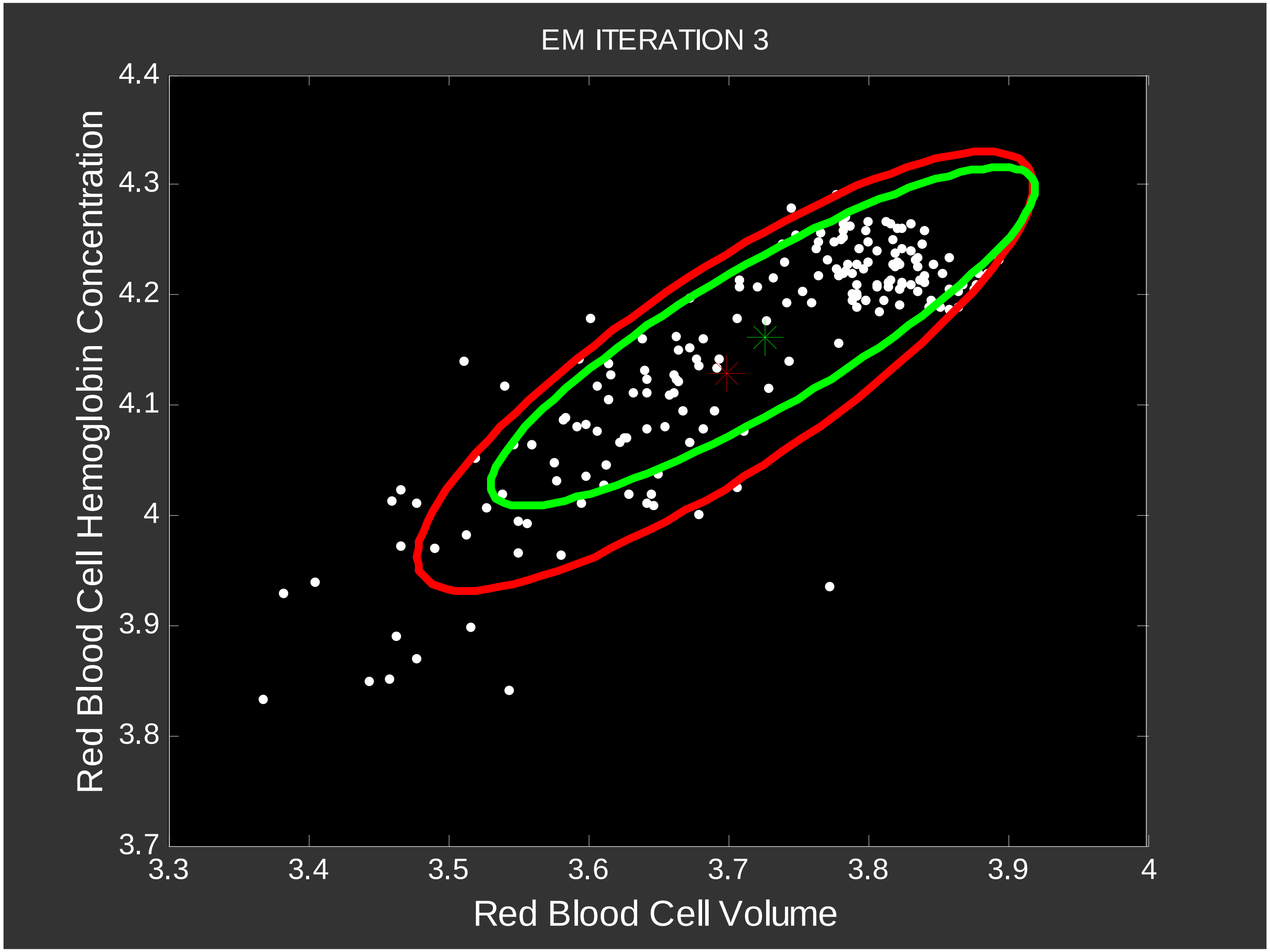
Density estimation approach – MoG

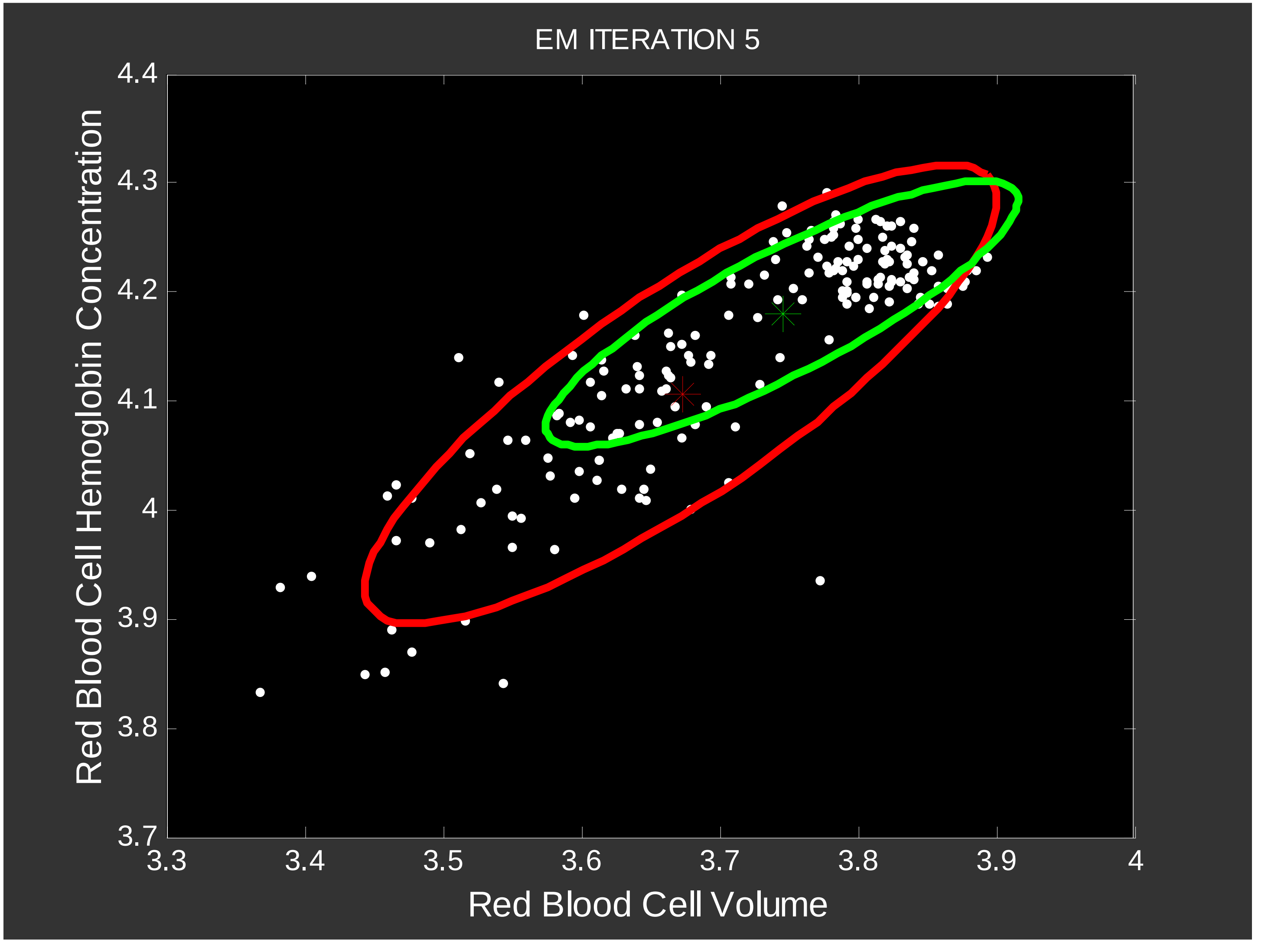
- **How to fit MoG to data**

  : ***The Expectation Maximization algorithm (EM algorithm)***

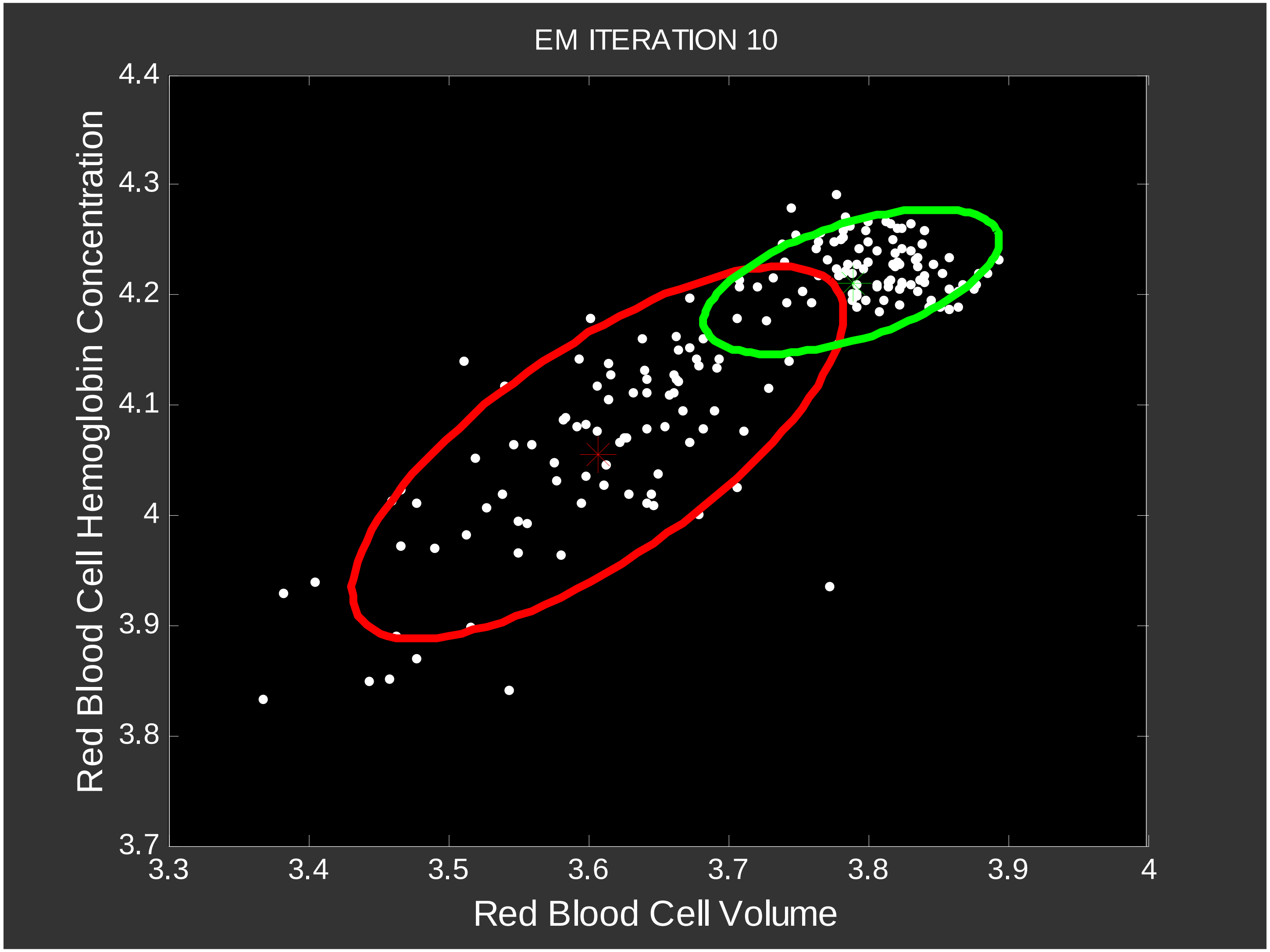
  - General framework for likelihood-based parameter estimation with missing data
    - start with initial guesses of parameters
    - E step: estimate memberships given params
    - M step: estimate params given memberships
    - Repeat until convergence
  - Converges to a (local) maximum of likelihood
  - E step and M step are often computationally simple
  - Generalizes to maximum a posteriori (with priors)


ANEMIA PATIENTS AND CONTROLS
Red Blood Cell Hemoglobin Concentration
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
Red Blood Cell Volume

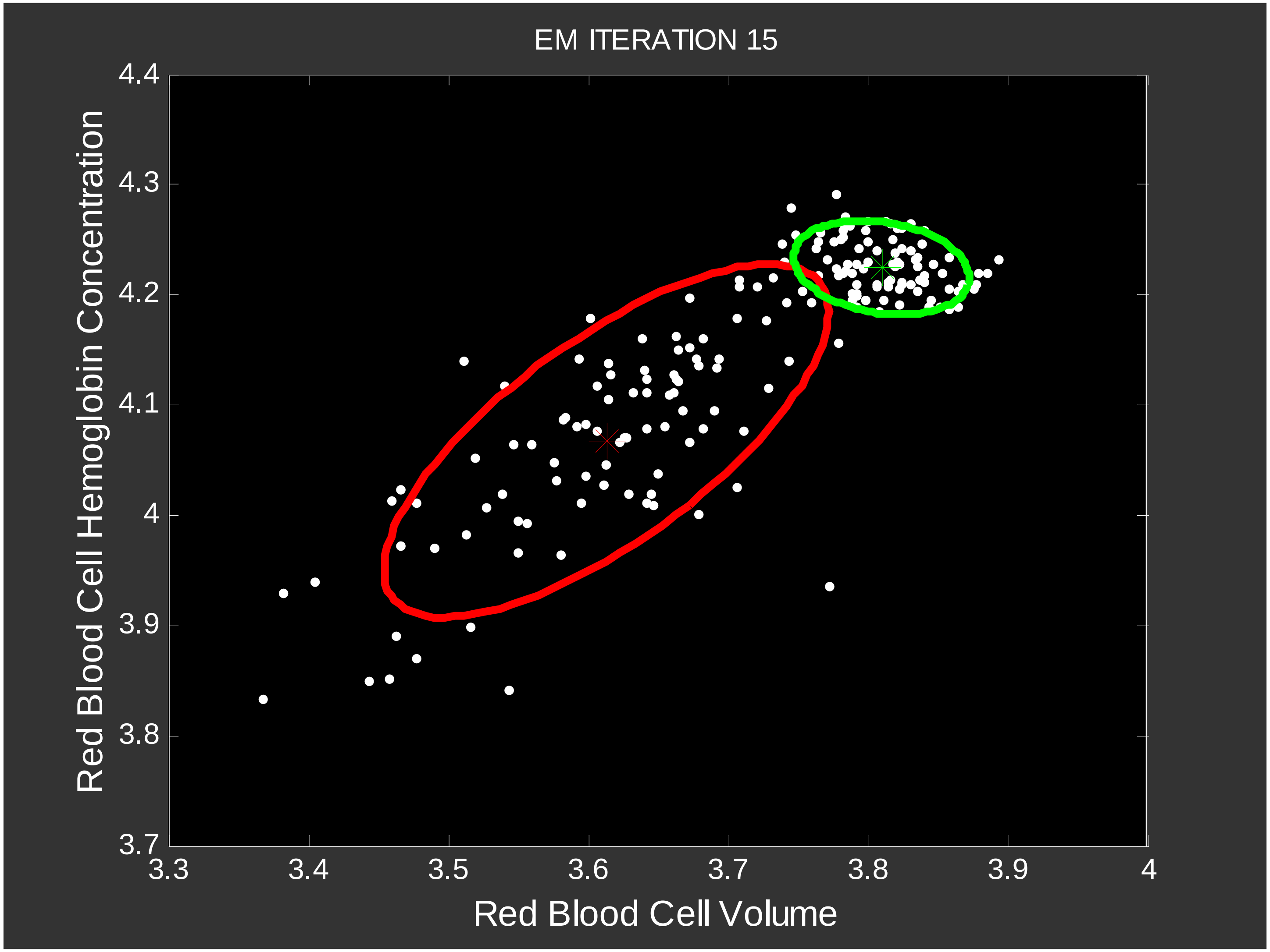

EM ITERATION 3
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4

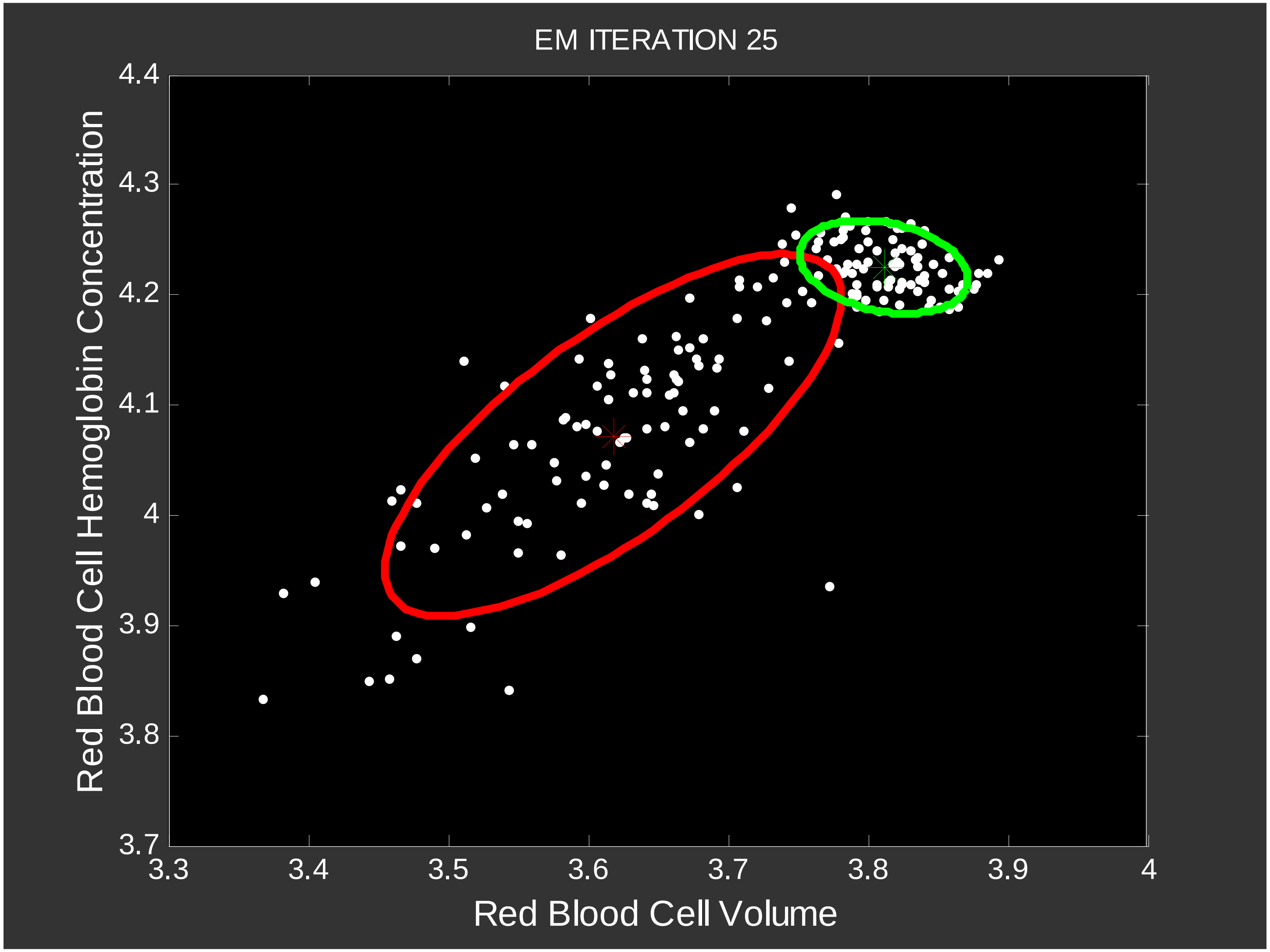


EM ITERATION 5
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4



EM ITERATION 10
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


EM ITERATION 15
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


EM ITERATION 25
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4

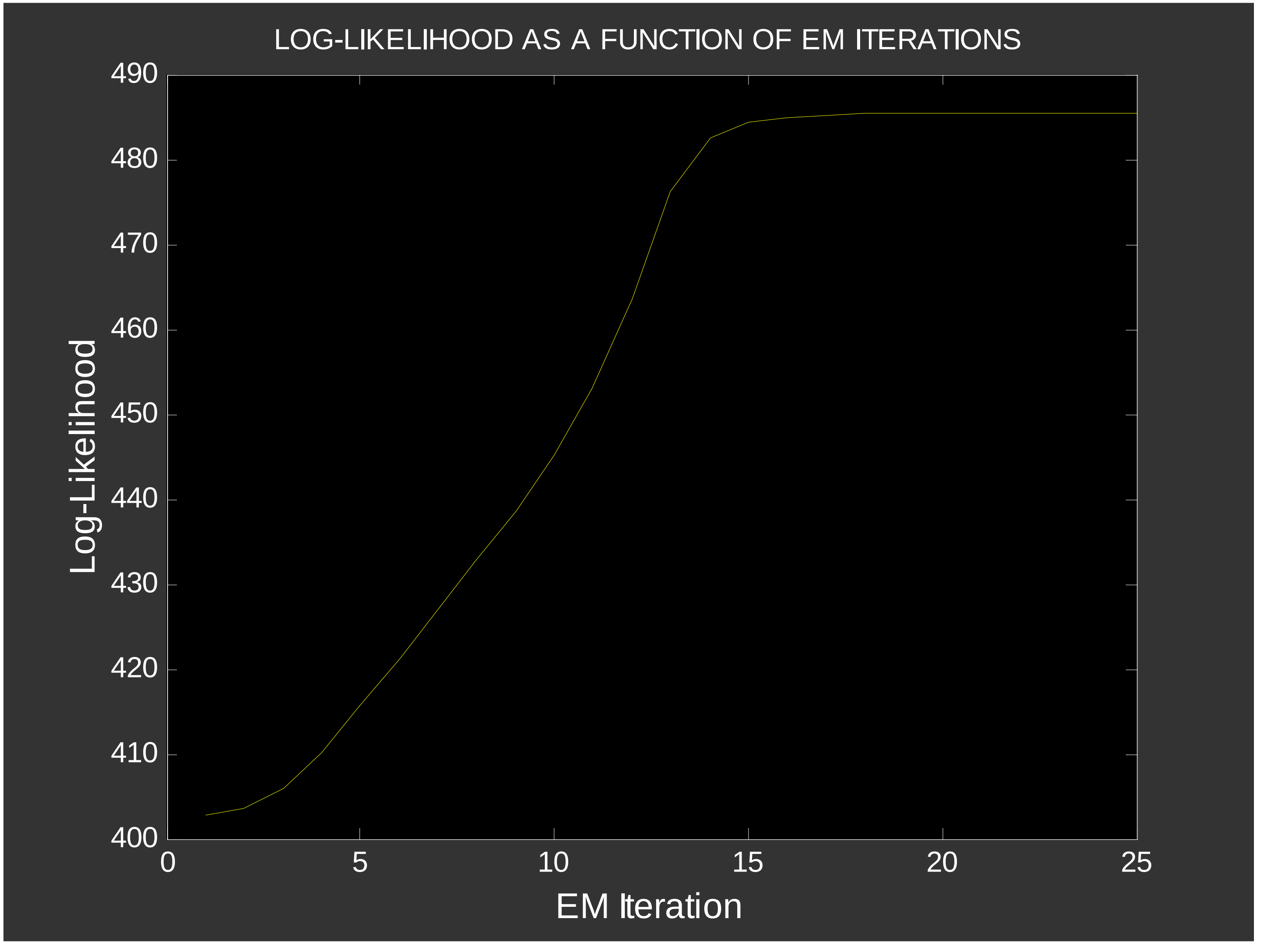
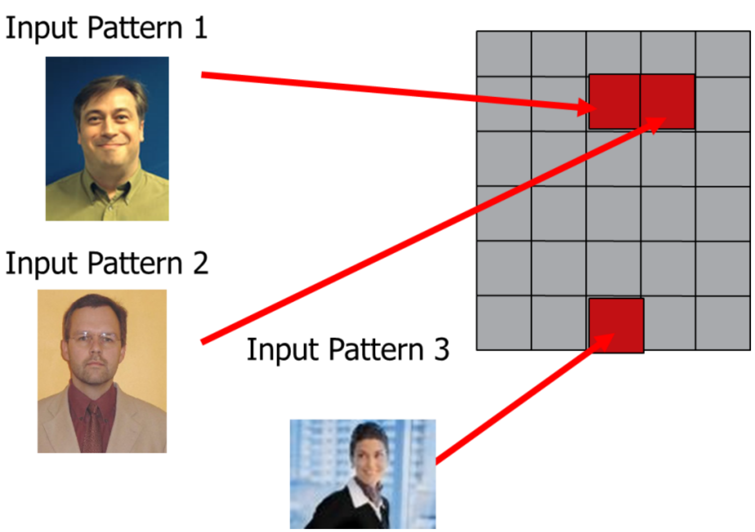
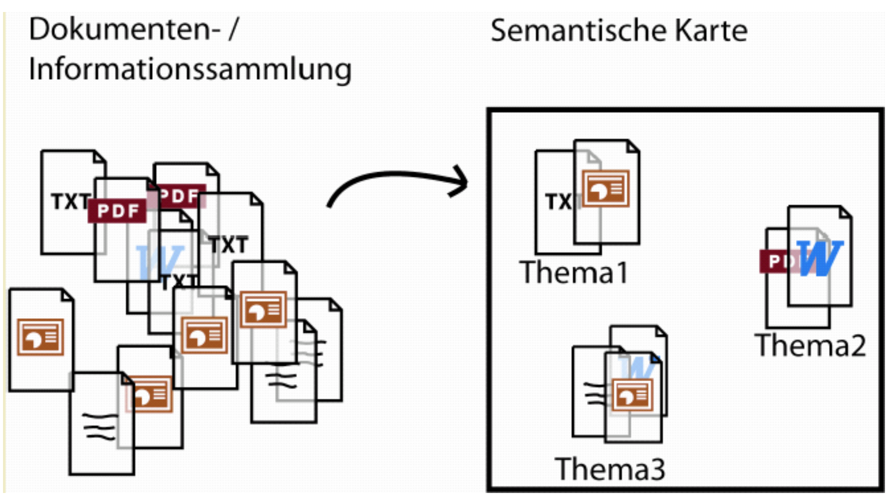

LOG-LIKELIHOOD AS A FUNCTION OF EM ITERATIONS
Log-Likelihood
EM Iteration
490
480
470
460
450
440
430
420
410
400
0
5
10
15
20
25

Table of Contents

자기조직화 지도: Self-Organizing Map (SOM)

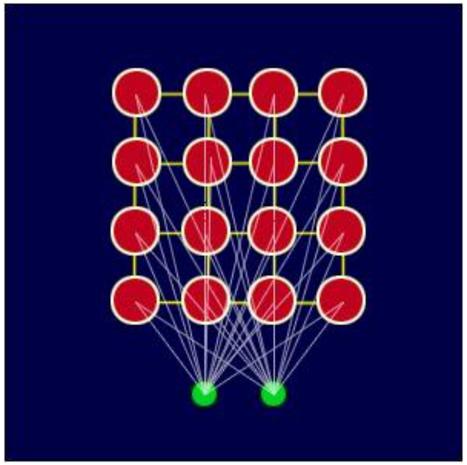
- **자기 조직화 지도: Self-Organizing Map (SOM)**
  - 고차원의 데이터를 사람이 시각적으로 이해할 수 있는 저차원(2차원 또는 3차원) 격자에 표현하는 방식
    - 고차원에서 유사한 개체들은 저차원에 인접한 격자들과 연결됨
    - 인공신경망 학습 알고리즘을 차용: 비지도적 경쟁학습
  - **저차원의 격자에서의 유사도는 고차원 입력 공간에서의 유사도를 최대한 보존하도록 학습**

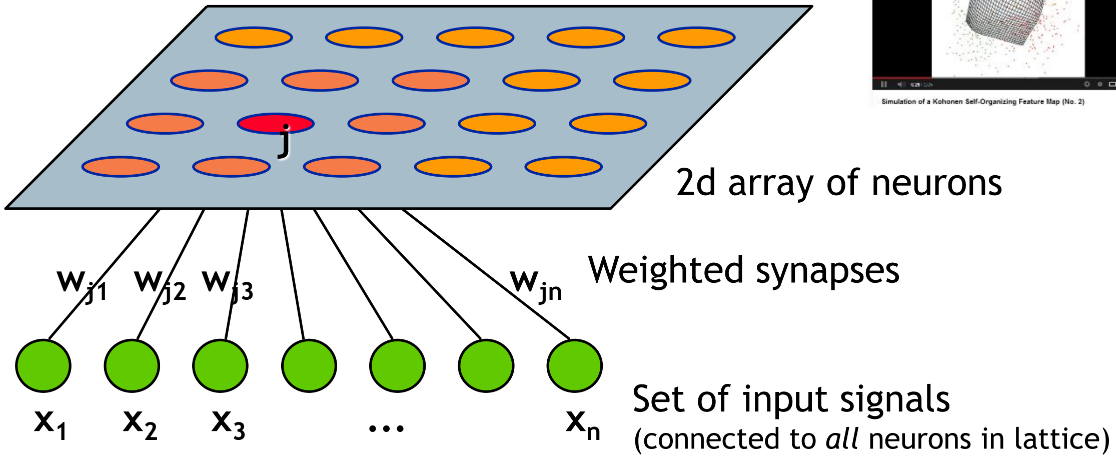




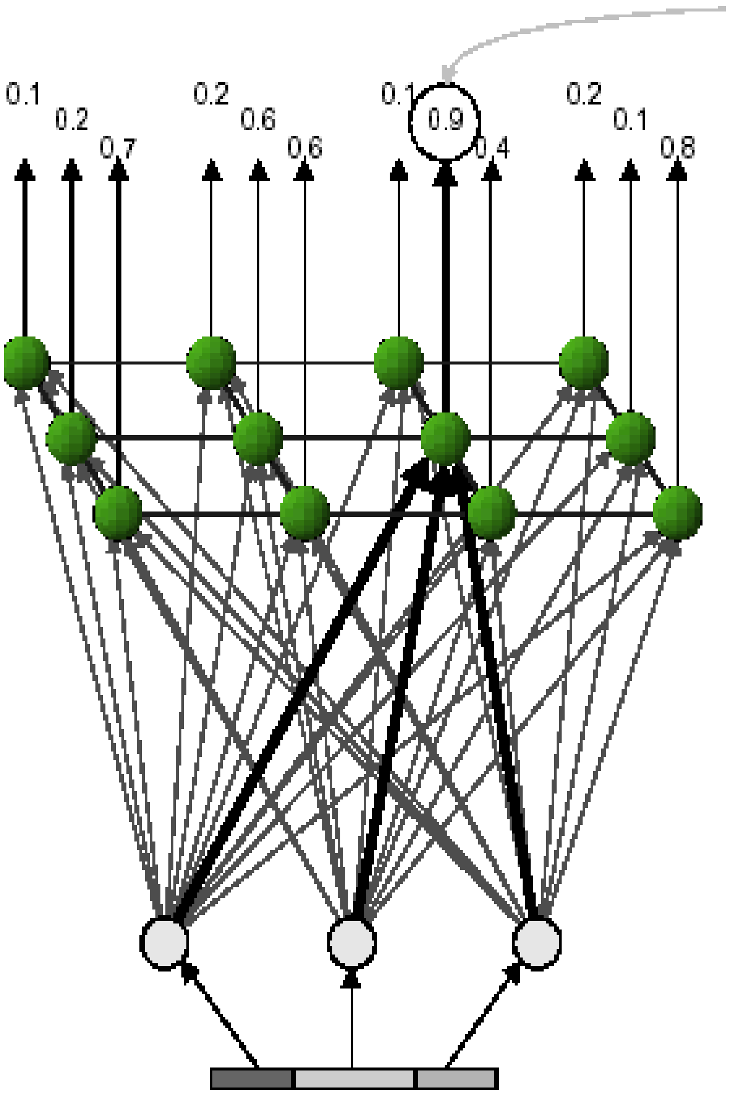
자기조직화 지도: Self-Organizing Map (SOM)

- 자기조직화 지도: 구조
  - 저차원 격자의 모든 노드는 원 공간의 모든 개체들과 가중치 w로 연결되어 있음
  - 저차원 격자의 노드들은 서로 위치적인 유사도 관계를 가짐
  - 원 공간의 각 개체와 가장 유사한 형태의 가중치를 갖는 **Winning 노드**가 선택됨
  - 선택된 노드 및 근처 노드들이 **활성화**되어 원 공간의 개체와 유사하도록 가중치를 조정함





자기조직화 지도: Self-Organizing Map (SOM)



Structure

- Input nodes: Input variables
- Output nodes: centroids, usually in 2-dim grid
- Synapses: Fully connected
- $y_j = w_j * x$, simple inner product between input vector and weight vector
- y_j is large when w_j is close to x.
- The inner product functions as a distance measure between centroid w_j and input vector x
- "Choose largest y_j = ?"

자기조직화 지도: Self-Organizing Map (SOM)

- **Topological ordering on the centroids (maintaining neighborhood relation)**
  - Input data vectors $\mathbf{x}^1$, $\mathbf{x}^2$
  - locations r^1, r^2 of corresponding centroids, respectively

    - if $|\mathbf{x}^1 - \mathbf{x}^2| \to 0$, then $|r^1 - r^2| \to 0$

    - if $|\mathbf{x}^1 - \mathbf{x}^2| \to \infty$, then $|r^1 - r^2| \to \infty$

- **Self-Organizing Map(SOM) is a neural network that implements the property.**

자기조직화 지도: Self-Organizing Map (SOM)

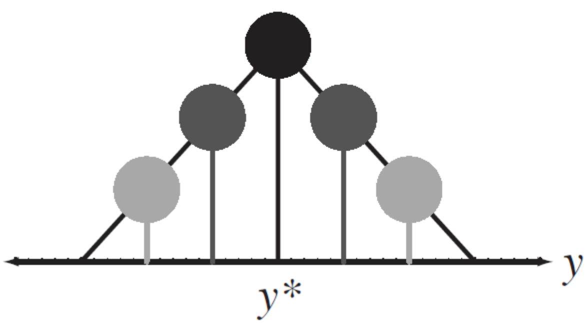
❖ 자기조직화지도: 학습

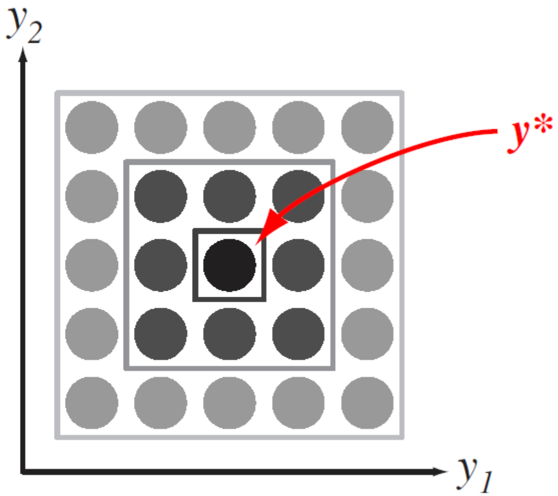
1. 자기조직화지도 격자 설정
2. 각 노드의 가중치를 설정 (usually at random)
3. 학습 데이터의 한 개체에 대해 모든 격자와의 유사도를 평가하여 **Best Machine Unit** (BMU) 선택
4. BMU와 이웃 노드와의 유사성 계산
5. BMU는 학습 데이터와 유사하도록 가중치를 업데이트하고, 이웃노드들도 일정 수준 가중치 업데이트를 수행
6. 가중치의 변화가 없을 때까지 Step 3부터 Step 5까지를 반복

자기조직화 지도: Self-Organizing Map (SOM)

❖ 자기조직화지도: 학습

- 이웃노드와의 유사성 계산





자기조직화 지도: Self-Organizing Map (SOM)

❖ **자기조직화지도: 학습**

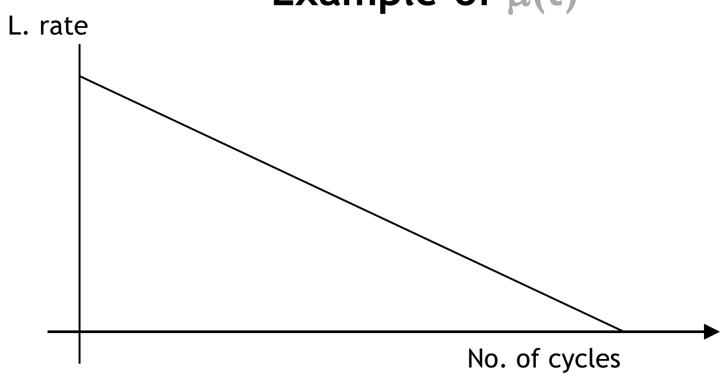
- BMU는 학습 데이터와 유사하도록 가중치를 업데이트하고, **이웃노드들도 일정 수준 가중치 업데이트를 수행**

<SOM Weight Update Equation>

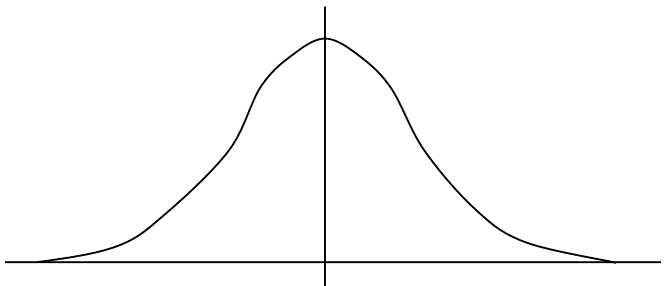
$$w_j(t+1) = w_j(t) + \mu(t)\, \lambda_{\omega(x)}(j,t)\, [x - w_j(t)]$$

"The weights of every node are updated at each cycle by adding Current learning rate × Degree of neighbourhood with respect to winner × Difference between current weights and input vector to the current weights"

Example of $\mu(t)$



Example of $\lambda_{\omega(x)}(j,t)$



-x-axis shows distance from winning node

-y-axis shows 'degree of neighbourhood' (max. 1)

자기조직화 지도: Self-Organizing Map (SOM)

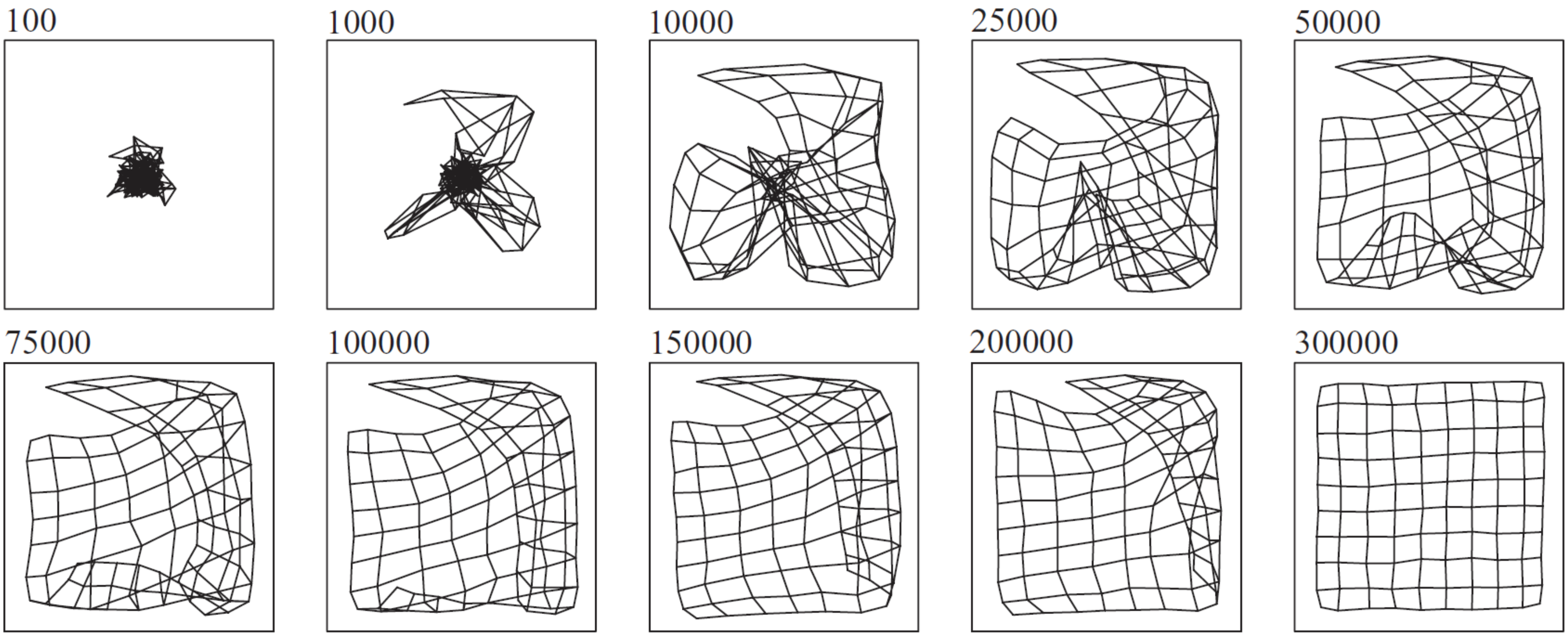
❖ 학습 방법

- Step 1 : Choose the winner
  - j* = arg min distance between x and w_j = arg max y_j
- Step 2 : Update weights $\Delta \boldsymbol{w_j} = \boldsymbol{\eta \Lambda}(\boldsymbol{j}, \boldsymbol{j}^*)(\boldsymbol{x} - \boldsymbol{w_j})$
  - Convergence :$\boldsymbol{\eta}(\boldsymbol{t}) \propto \boldsymbol{t}^{-\boldsymbol{a}}, (\mathbf{0} < \boldsymbol{a} < \mathbf{1})$
  - $\boldsymbol{\Lambda}(\boldsymbol{j}, \boldsymbol{j}^*)$: Neighborhood function
  - The further j is located from j*, the smaller the value”
  - ex) Gaussian $\boldsymbol{\Lambda}(\boldsymbol{j}, \boldsymbol{j}^*) = \exp(-\frac{\left|r_j - r_{j^*}\right|^2}{2\sigma^2})$
    - r_j : location of unit j
    - σ : width parameter(diffuse-->sharp) / $\sigma(t) = 1/t$
  - Result: Nodes around j* become similar to j*

자기조직화 지도: Self-Organizing Map (SOM)

❖ **자기조직화 지도: 수렴**

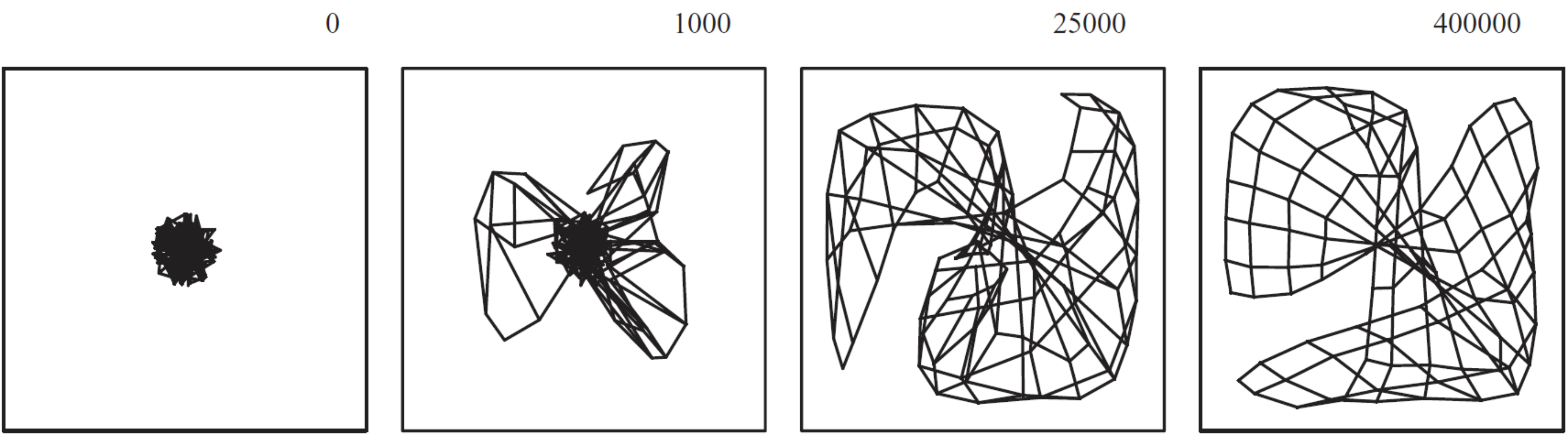
- 행복하게도…



자기조직화 지도: Self-Organizing Map (SOM)

❖ 자기조직화 지도: 수렴

- 가끔씩은…

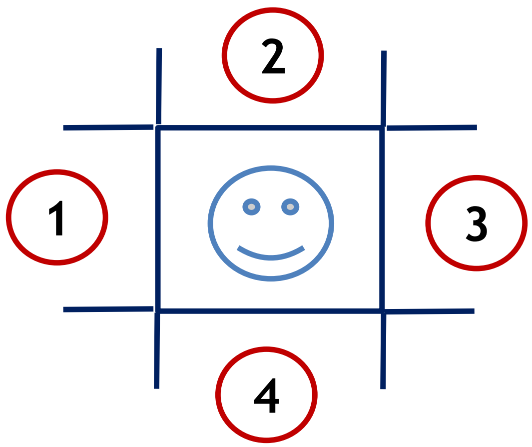


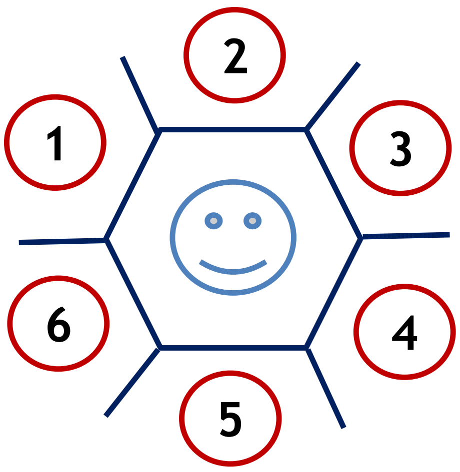
- 초기 가중치를 재설정

자기조직화 지도: Self-Organizing Map (SOM)

❖ 시각화로서의 **SOM**

- Useful for visualizing low-dimensional views of high-dimensional data.
- Property : Topological Ordering
- **6-grid based SOM** : Stronger explaining power than 4-grid.

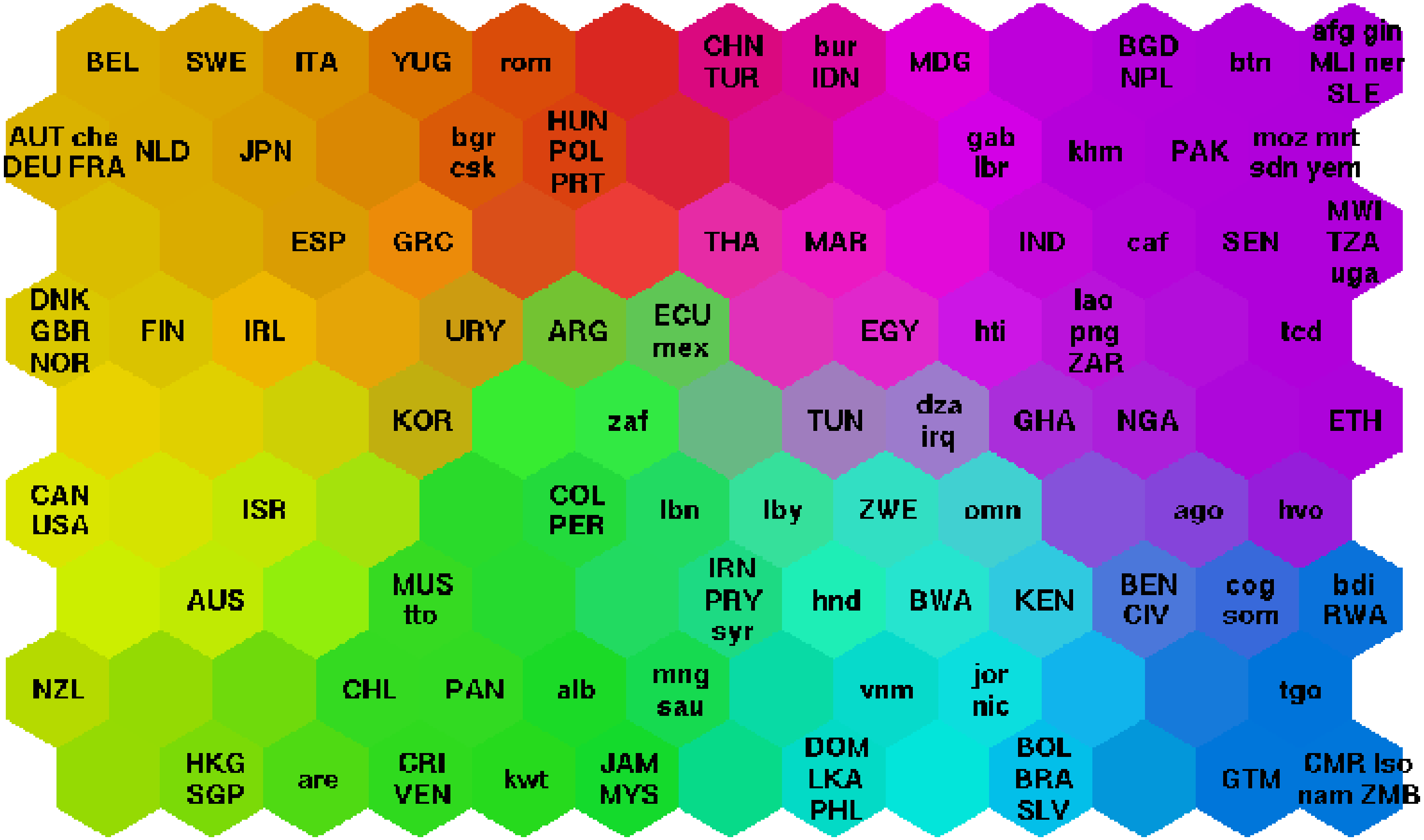




Self-Organizing Map: Application

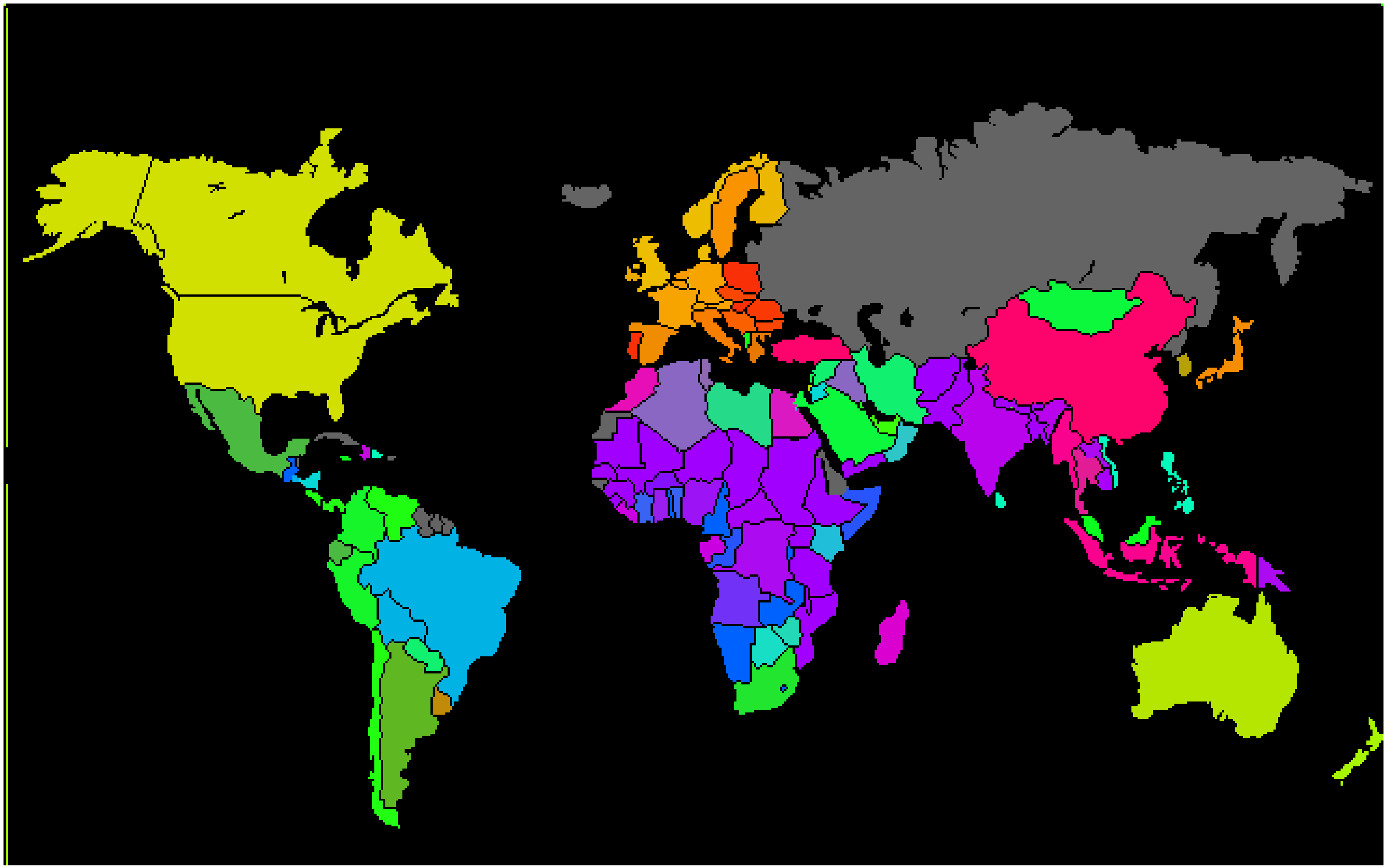
World Poverty Map

- Countries in the world mapped based on 30 Socio-economic indicators related to poverty.



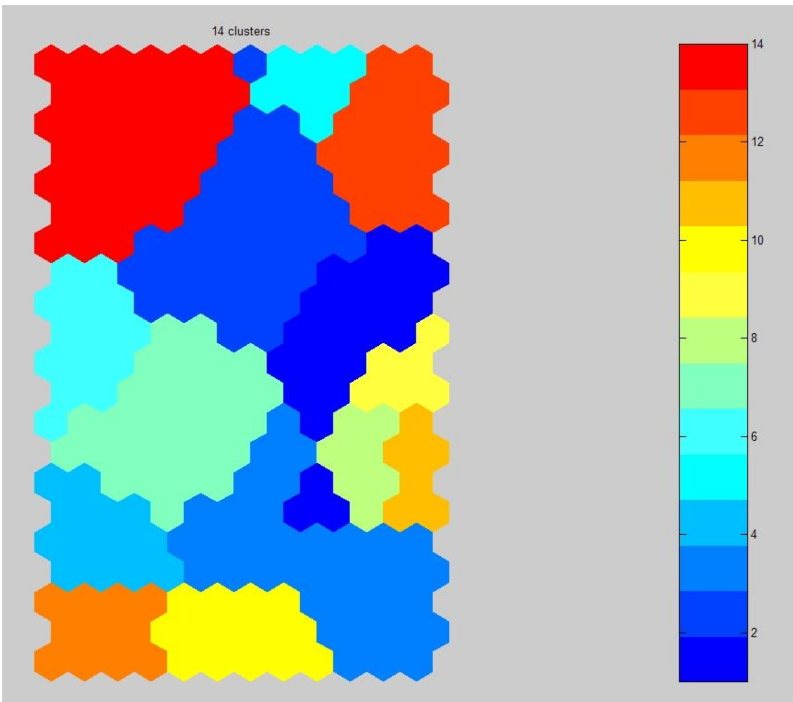
Self-Organizing Map: Application

- World Poverty Map

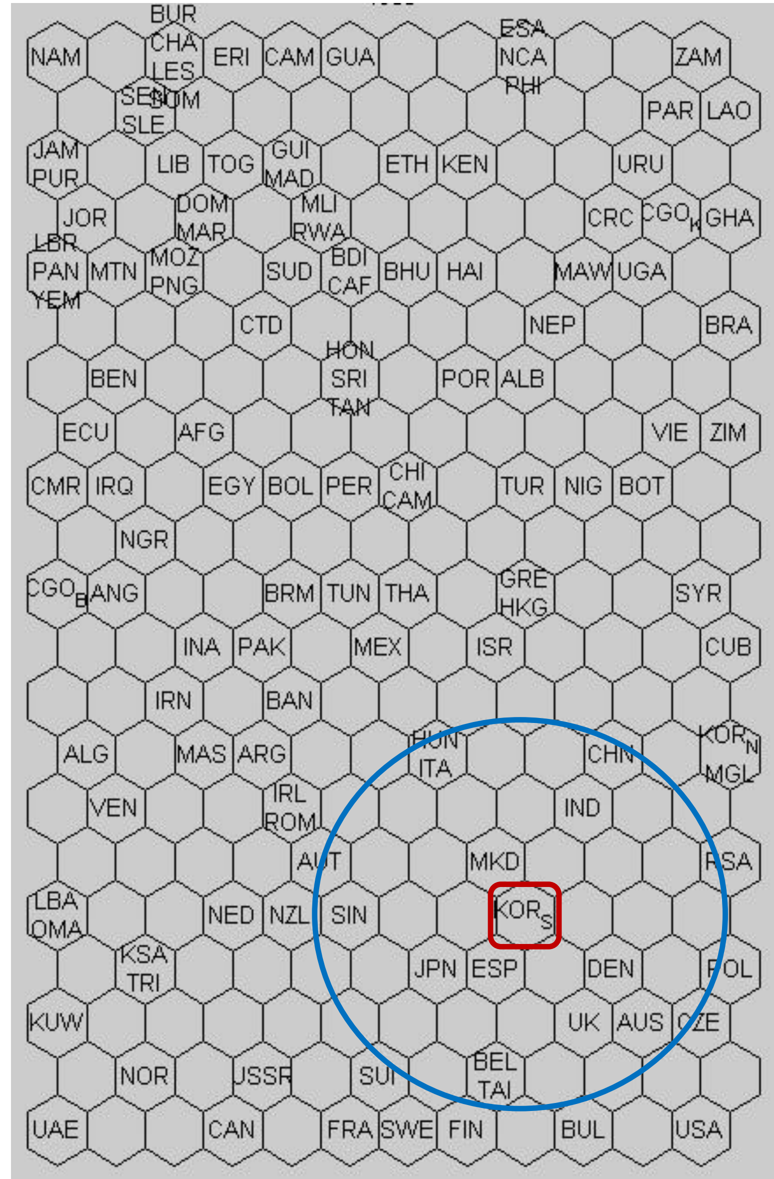


Self-Organizing Map: Application

- Energy resources consuming pattern in the word: 1985

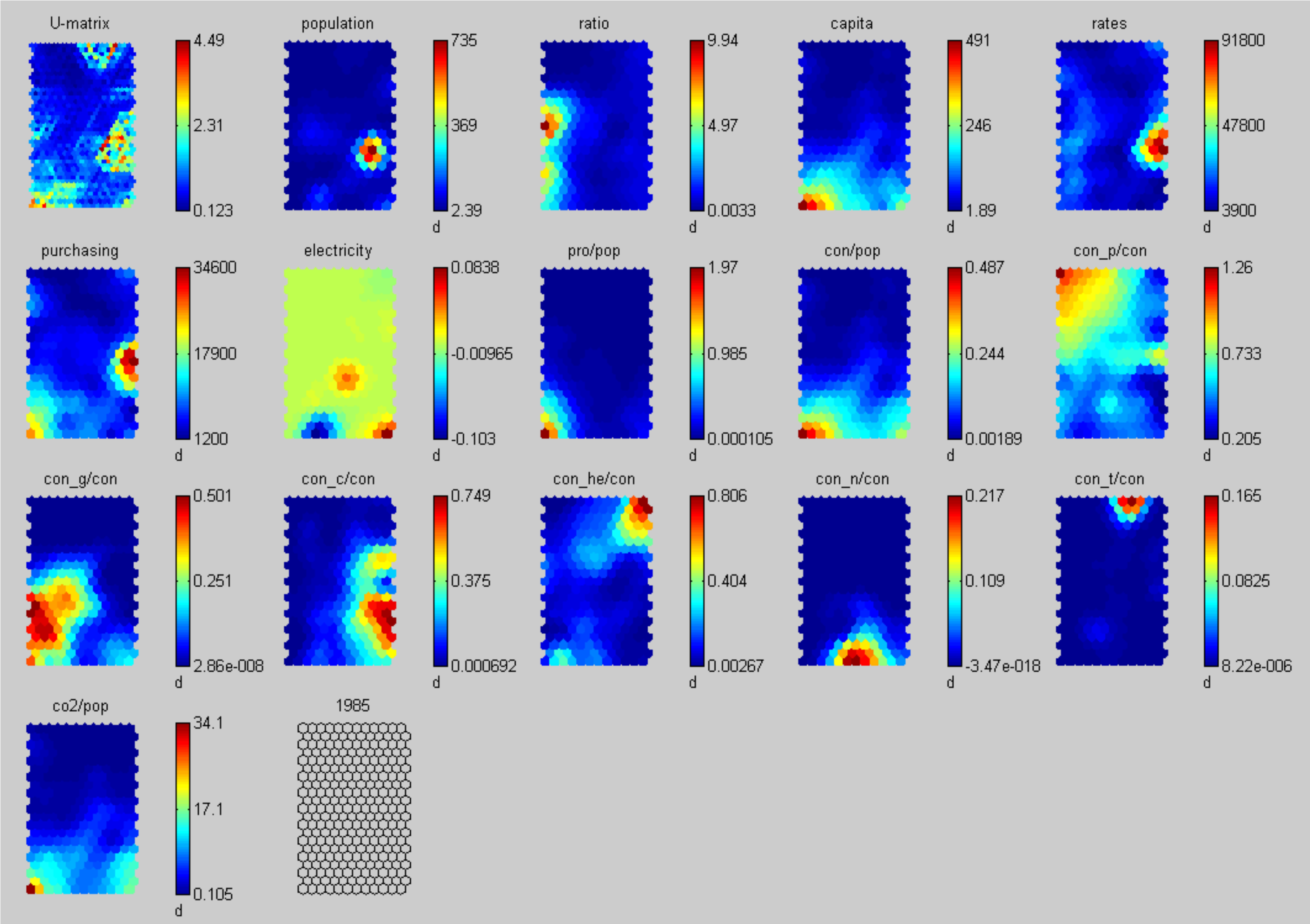


K-means Clustering : 14 Clusters



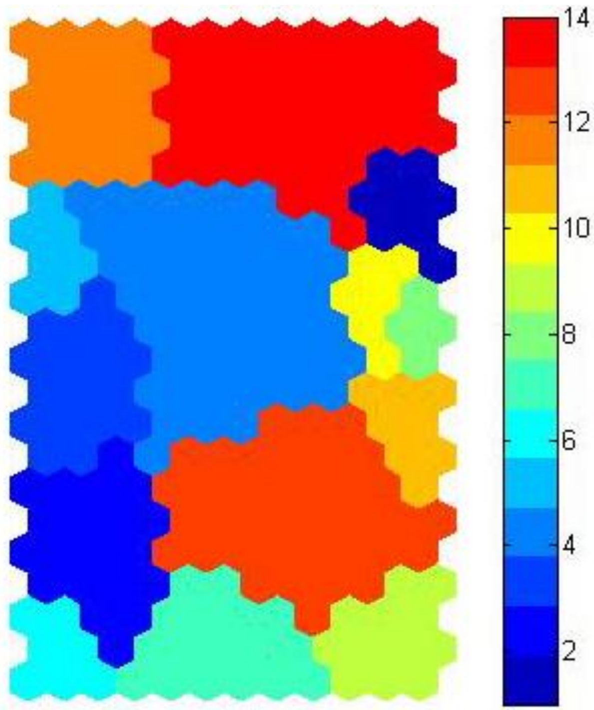
Self-Organizing Map: Application

- Energy resources consuming pattern in the word: 1985

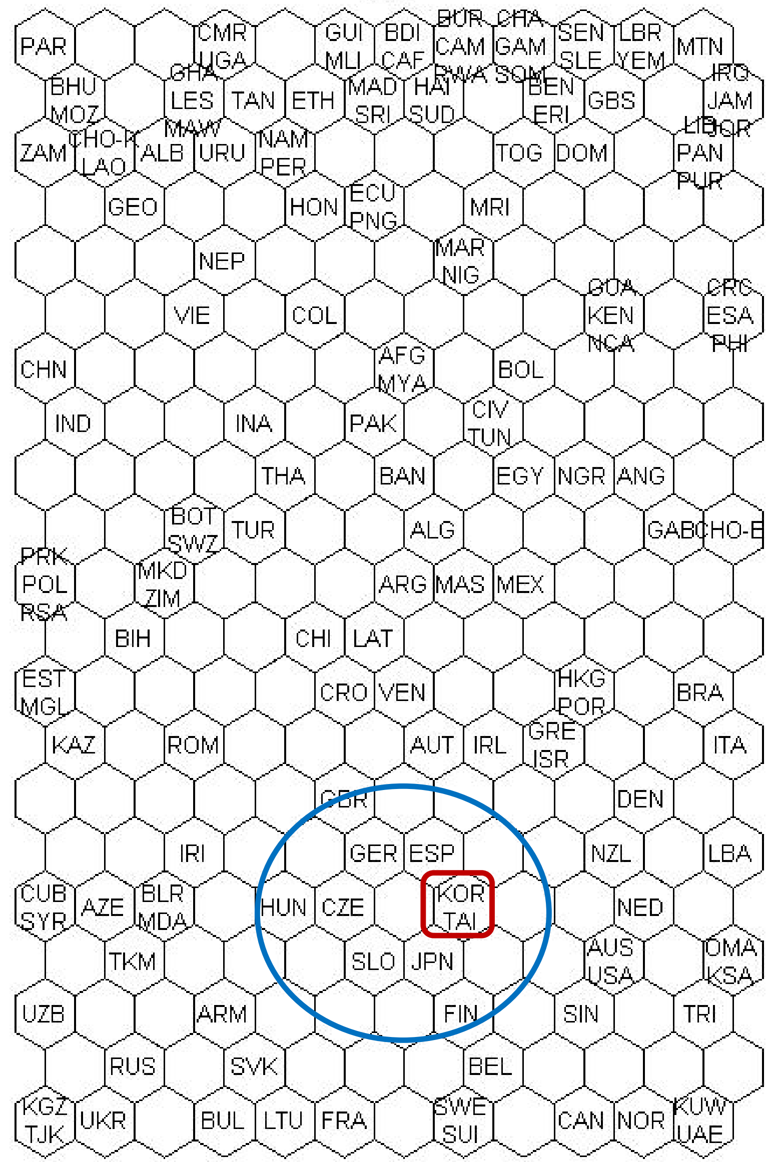


Self-Organizing Map: Application

- Energy resources consuming pattern in the word: 2000

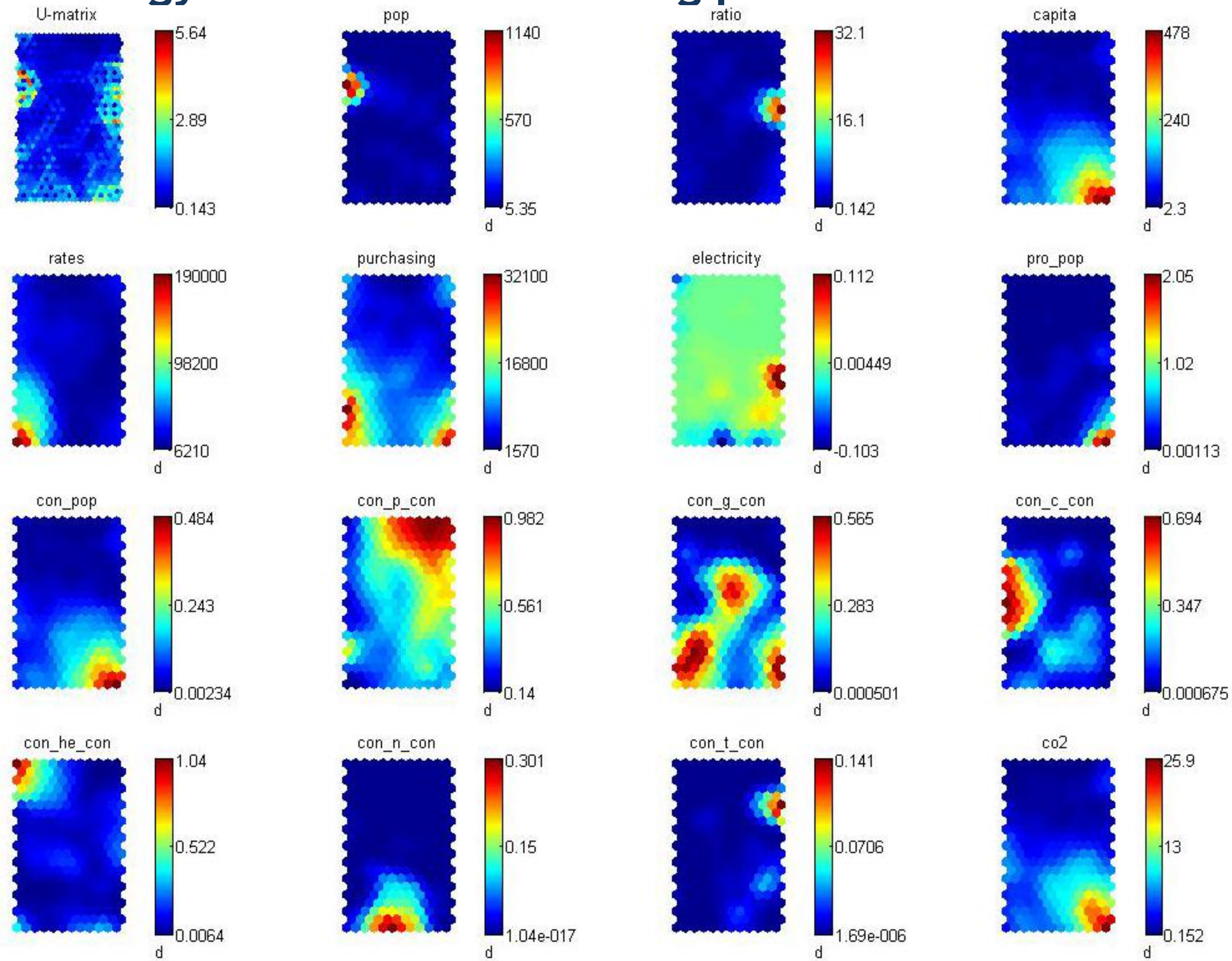


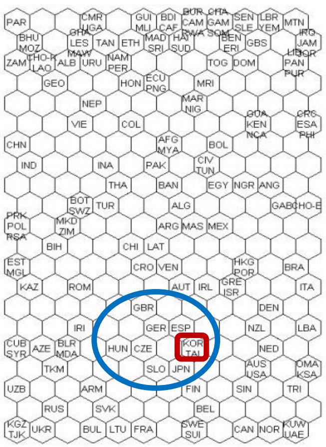
K-means Clustering : 14 Clusters



Self-Organizing Map: Application

- ❖ **Energy resources consuming pattern in the word: 2000**





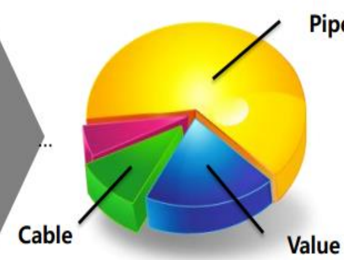
Self-Organizing Map: Application

❖ **해양 구조물 검사문서 “펀치(punch)”**

- 호선 생산에서 발생하는 결점들의 원인 유형들을 관리할 필요가 있음
- 비정형화 된 text 형태로 되어 있는 검사 보고서를 토대로 결점 유형 data mining
- 1개의 구조물에서 약 2년간 28,800여건의 검사문서

Punch 생성

Text mining



Summarization

Understanding/
Intuition

Self-Organizing Map: Application

- 유사한 검사문서를 군집으로 묶는 Clustering
  - Self-Organizing Map(SOM)을 이용한 text clustering 수행
  - 같은 문서에 나올 확률이 큰 단어들의 집합을

<u>ex) 단어 조합에 따른 시나리오</u>

valve + tag + missing

⇨ **“valve tag is missing”**

Jack bolt + install

+ spectacle + blind joint

⇨ **“install the jack bolt on spectacle blind joint”**

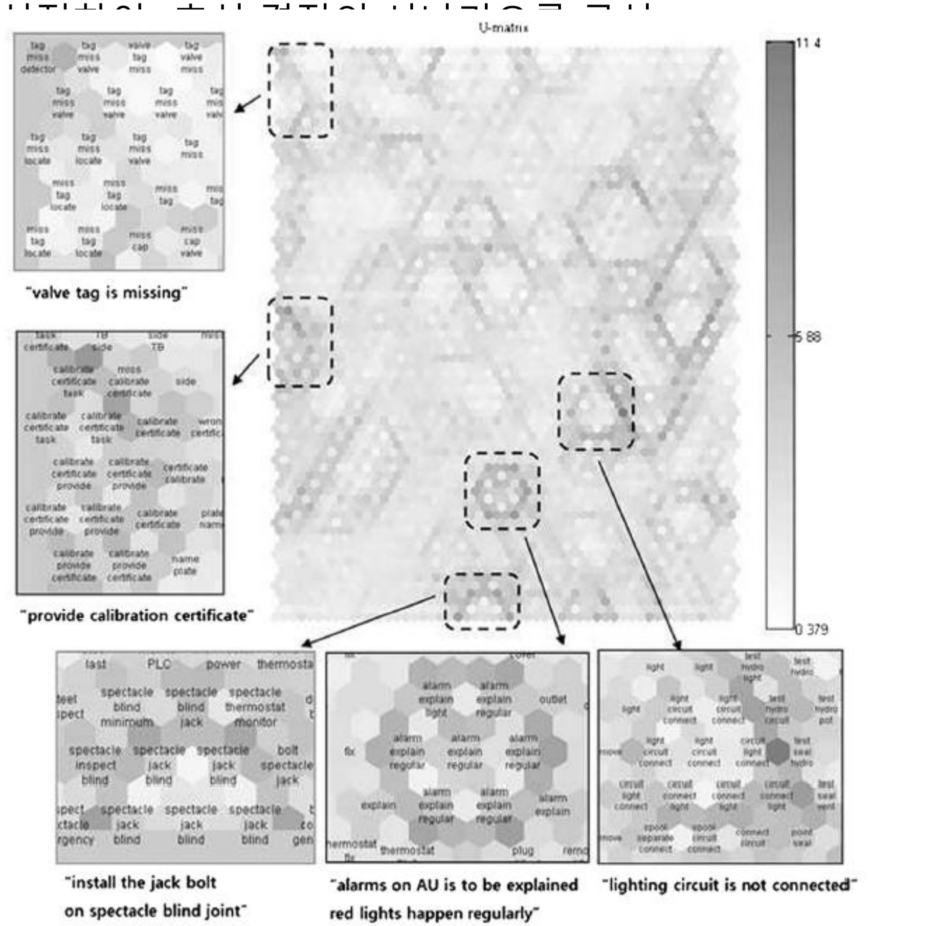


Fig. 6. U-matrix visualization of the inspection report map.