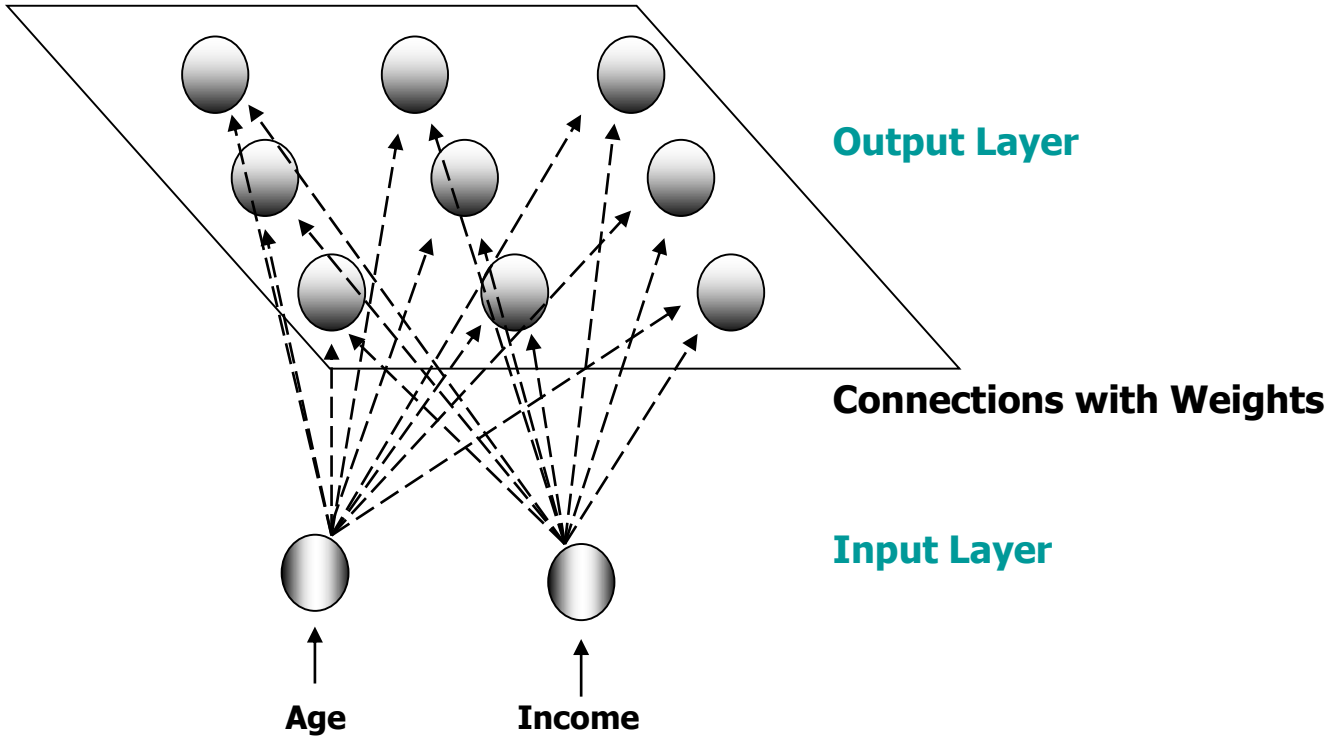
동국대학교
dongguk university

Lecture 4
자기조직화지도

Excerpts the examples from Larose “Discovering Knowledge in Data”

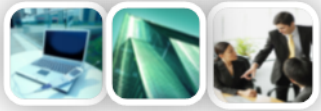
자기조직화지도

- Self-organizing maps (SOM)
  - Tuevo Kohonen
  - Kohonen networks



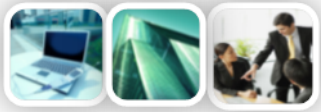
기본 작동원리

- SOMs are Feedforward
- Each node in given layer, completely connected to every node in next layer
- Every connection between two nodes has weight
- Weight values initialized randomly 0 ~ 1
- Adjusting weights is key feature of learning process
- Attribute values are normalized or standardized
- SOMs do not have hidden layer
- Data passed directly from input layer to output layer



3가지 특징

- 경쟁 (Competition)
  - Output nodes compete with one another for “best” score
  - Winning node produces smallest distance between inputs and connection weights
- 협동 (Cooperation)
  - Winning node becomes center of neighborhood
  - Output nodes in neighborhood share “excitement” or “reward”
- 적응 (Adaptation)
  - Neighborhood nodes participate in adaptation (learning)
  - Weights adjusted to improve score function



가중치 조정

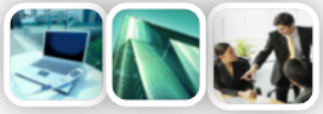
- 현재 가중치 벡터와 입력 벡터와의 선형 결합을 통한 조정

$$w_{ij,NEW} = w_{ij,CURRENT} + \eta(x_{ni} - w_{ij,CURRENT}), \text{ where}$$

$\mathbf{x}_n = x_{n1}, x_{n2}, \ldots, x_{nm}$ m field values for nth record

$\mathbf{w}_j = w_{1j}, w_{2j}, \ldots, w_{mj}$ current set of m weights, for particular output node j

$\eta, 0 < \eta < 1$ learning rate



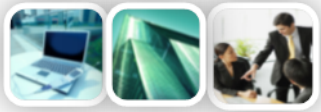
코호넨 네트워크 알고리즘

- 초기화
  - 가중치를 난수로 부여
- 경쟁
  - 각 출력노드 **j** 에 대해 거리 함수 **$D(w_j, x_n)$**를 계산

$$\text{Euclidean Distance} = D(w_j, x_n) = \sqrt{\sum_i (w_{ij} - x_{ni})^2}$$

  - **$D(w_j, x_n)$**를 최소화 시키는 승리 노드 **j** 를 탐색
- 협동
  - **Identify output nodes j, within neighborhood of J defined by neighborhood size R**
- 적응
  - **Adjust weights of all neighborhood nodes j:**

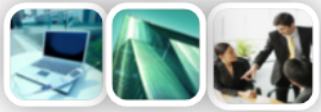
$$w_{ij,NEW} = w_{ij,CURRENT} + \eta(x_{ni} - w_{ij,CURRENT})$$



사례

- **2 x 2 Kohonen Network**
- 이웃 크기 **= 0,** 학습율 **= 0.5**
- 정규화된 나이와 소득수준 변수로 구성된 **4**명의 데이터

1	x11 = 0.8	x12 = 0.8	높은 소득수준을 가진 노인
2	x21 = 0.8	x22 = 0.1	낮은 소득수준을 가진 노인
3	x31 = 0.2	x32 = 0.9	높은 소득수준을 가진 청년
4	x41 = 0.1	x42 = 0.1	낮은 소득수준을 가진 청년

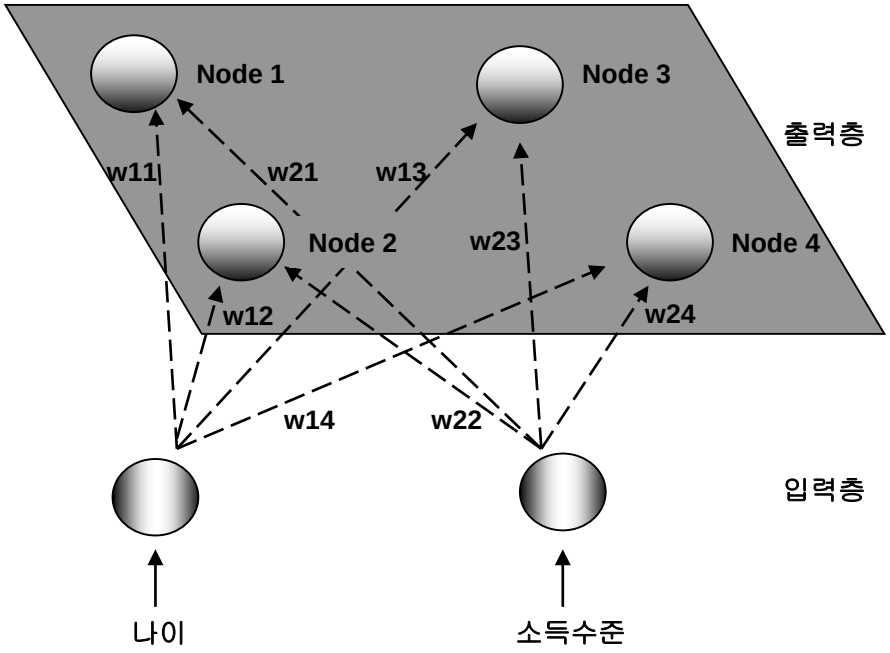


연산과정

- 네트워크 가중치 초기화 (랜덤값):

w11 = 0.9	w21 = 0.8	w12 = 0.9	w22 = 0.2
w13 = 0.1	w23 = 0.8	w14 = 0.1	w24 = 0.2

- 사례에 사용된 네트워크 구조



연산과정

- 첫 번째 데이터 **x1 = (0.8, 0.8)**
  - **Competition Phase**
    - **Compute Euclidean Distance between input and weight vectors**

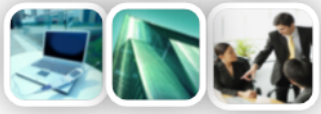
$$Node1: \quad D(w_1, x_1) = \sqrt{(0.9-0.8)^2 + (0.8-0.8)^2} = 0.10$$

$$Node2: \quad D(w_2, x_1) = \sqrt{(0.9-0.8)^2 + (0.2-0.8)^2} = 0.61$$

$$Node3: \quad D(w_3, x_1) = \sqrt{(0.1-0.8)^2 + (0.8-0.8)^2} = 0.70$$

$$Node4: \quad D(w_4, x_1) = \sqrt{(0.1-0.8)^2 + (0.2-0.8)^2} = 0.92$$

- **The winning node is Node 1 (minimizes distance = 0.10)**
- **Note, node 1 weights most similar to input record values**
- **Node 1 may exhibit affinity (cluster) for records of "older persons with high income"**

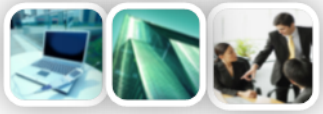


연산과정

- **Cooperation Phase**
  - **Neighborhood Size R = 0**
  - **Therefore, nonexistent "excitement" of neighboring nodes**
  - **Only winning node receives weight adjustment**
- **Adaptation Phase**
  - **Weights for Node 1 adjusted, where j = 1 (Node 1), n = 1 (First record), and learning rate = 0.5:**

$$Age: \quad \Rightarrow w_{11,NEW} = w_{11,CURRENT} + 0.5(x_{11} - w_{11,CURRENT})$$
$$= 0.9 + 0.5(0.8 - 0.9) = 0.85$$

$$Income: \quad \Rightarrow w_{21,NEW} = w_{21,CURRENT} + 0.5(x_{12} - w_{21,CURRENT})$$
$$= 0.8 + 0.5(0.8 - 0.8) = 0.8$$



연산과정

- 두 번째 데이터 **x2 = (0.8, 0.1)**
  - **Competition**
    - **Compute Euclidean Distance between input and weight vectors**

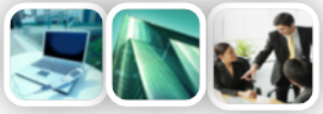
$$Node1: \quad D(w_1, x_2) = \sqrt{(0.9 - 0.8)^2 + (0.8 - 0.1)^2} = 0.71$$

$$Node2: \quad D(w_2, x_2) = \sqrt{(0.9 - 0.8)^2 + (0.2 - 0.1)^2} = 0.14$$

$$Node3: \quad D(w_3, x_2) = \sqrt{(0.1 - 0.8)^2 + (0.8 - 0.1)^2} = 0.99$$

$$Node4: \quad D(w_4, x_2) = \sqrt{(0.1 - 0.8)^2 + (0.2 - 0.1)^2} = 0.71$$

- **Node 2 is the winning node with distance = 0.14**
- **Node 2 weights (0.9, 0.2) most similar to input record values (0.8, 0.1)**
- **Records of "older persons and low income" may cluster to Node 2**



연산과정

- **Adaptation**
  - **Weights for Node 2 adjusted, where j = 2 (Node 2), n = 2 (Second record), and learning rate = 0.5:**

$$Age: \quad = w_{12,NEW} = w_{12,CURRENT} + 0.5(x_{21} - w_{12,CURRENT})$$
$$= 0.9 + 0.5(0.8 - 0.9) = 0.85$$
$$Income: \quad = w_{22,NEW} = w_{22,CURRENT} + 0.5(x_{22} - w_{22,CURRENT})$$
$$= 0.2 + 0.5(0.1 - 0.2) = 0.15$$

  - **Again, weights move towards input field values**
  - **Initial w_{12} = 0.9, adjusted to 0.85 (direction of x_{12} = 0.8)**
  - **Initial w_{22} = 0.2, adjusted to 0.15 (direction of x_{22} = 0.1)**
  - **Node 2 develops affinity for records of "older, lower income" persons**

연산과정

- 세 번째 데이터 **x3 = (0.2, 0.9)**
  - **Competition**
    - **Compute Euclidean Distance between input and weight vectors**

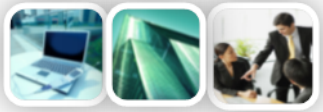
$$Node1: \quad D(w_1, x_3) = \sqrt{(0.9-0.2)^2 + (0.8-0.9)^2} = 0.71$$

$$Node2: \quad D(w_2, x_3) = \sqrt{(0.9-0.2)^2 + (0.2-0.9)^2} = 0.99$$

$$Node3: \quad D(w_3, x_3) = \sqrt{(0.1-0.2)^2 + (0.8-0.9)^2} = 0.14$$

$$Node4: \quad D(w_4, x_3) = \sqrt{(0.1-0.2)^2 + (0.2-0.9)^2} = 0.71$$

- **Node 3 is the winning node with distance = 0.14**
- **Node 3 weights (0.1, 0.8) most similar to input record values (0.2, 0.9)**
- **Records of "younger persons and high income" may cluster to Node 3**



연산과정

- **Adaptation**
  - **Weights for Node 3 adjusted, where j = 3 (Node 3), n = 3 (Third record), and learning rate = 0.5:**

$$Age: \quad = w_{13,NEW} = w_{13,CURRENT} + 0.5(x_{31} - w_{13,CURRENT})$$
$$= 0.1 + 0.5(0.2 - 0.1) = 0.15$$
$$Income: \quad = w_{23,NEW} = w_{23,CURRENT} + 0.5(x_{32} - w_{23,CURRENT})$$
$$= 0.8 + 0.5(0.9 - 0.8) = 0.85$$

  - **Again, weights move towards input field values**
  - **Initial w_{13} = 0.1, adjusted to 0.15 (direction of x_{12} = 0.2)**
  - **Initial w_{23} = 0.8, adjusted to 0.85 (direction of x_{22} = 0.9)**
  - **Node 3 develops affinity for records of "younger, high income" persons**

연산과정

- 네 번째 데이터 **x4 = (0.1, 0.1)**
  - **Competition**
    - **Compute Euclidean Distance between input and weight vectors**

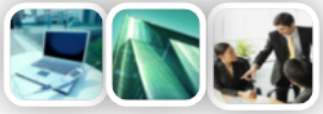
$$Node1: \quad D(w_1, x_4) = \sqrt{(0.9-0.1)^2 + (0.8-0.1)^2} = 1.06$$

$$Node2: \quad D(w_2, x_4) = \sqrt{(0.9-0.1)^2 + (0.2-0.1)^2} = 0.81$$

$$Node3: \quad D(w_3, x_4) = \sqrt{(0.1-0.1)^2 + (0.8-0.1)^2} = 0.70$$

$$Node4: \quad D(w_4, x_4) = \sqrt{(0.1-0.1)^2 + (0.2-0.1)^2} = 0.10$$

- **Node 4 is the winning node with distance = 0.10**
- **Node 4 weights (0.1, 0.2) most similar to input record values (0.1, 0.1)**
- **Records of "younger persons and low income" may cluster to Node 4**



연산과정

- **Adaptation**
  - **Weights for Node 4 adjusted, where j = 4 (Node 4), n = 4 (Fourth record), and learning rate = 0.5:**

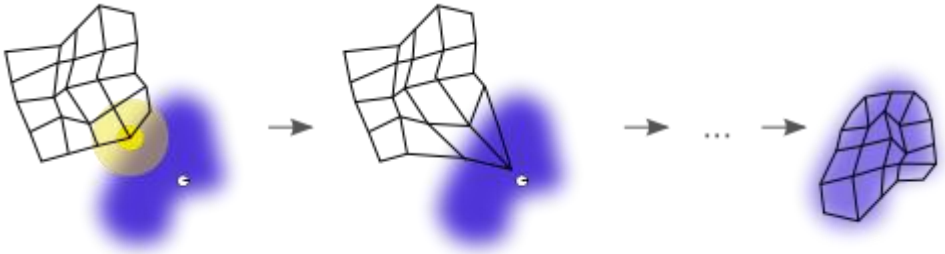
$$Age: \quad = w_{14,NEW} = w_{14,CURRENT} + 0.5(x_{41} - w_{14,CURRENT})$$
$$= 0.1 + 0.5(0.1 - 0.1) = 0.10$$
$$Income: \quad = w_{24,NEW} = w_{24,CURRENT} + 0.5(x_{42} - w_{24,CURRENT})$$
$$= 0.2 + 0.5(0.1 - 0.2) = 0.15$$

  - **Again, weights move towards input field values**
  - **Initial w_{14} = 0.1, adjusted to 0.10 (direction of x_{12} = 0.1)**
  - **Initial w_{24} = 0.2, adjusted to 0.15 (direction of x_{22} = 0.1)**
  - **Node 4 develops affinity for records of "younger, lower income" persons**

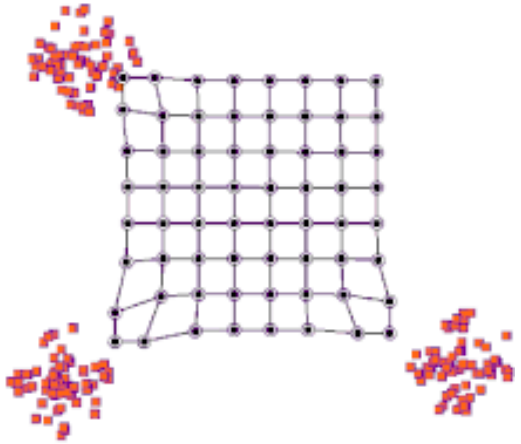
분석 결과

- **Four output nodes represent distinct clusters**

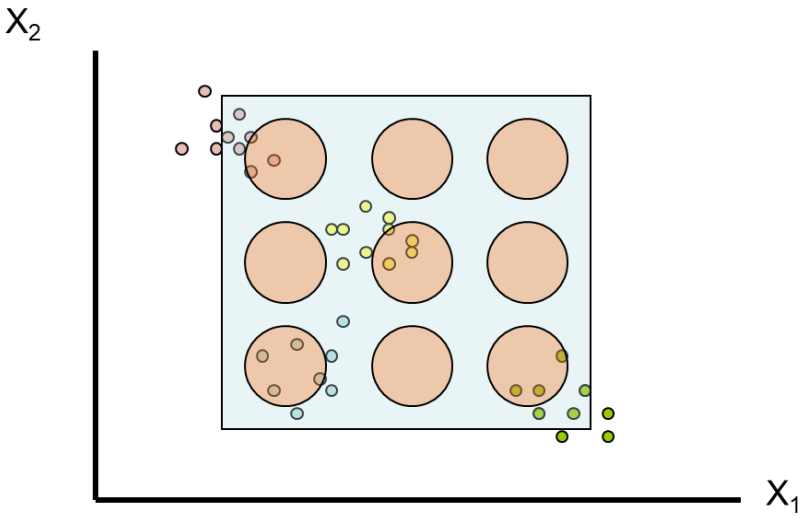
Cluster	Associated With	Description
1	Node 1	Older person with high income
2	Node 2	Older person with low income
3	Node 3	Younger person with high income
4	Node 4	Younger person with low income

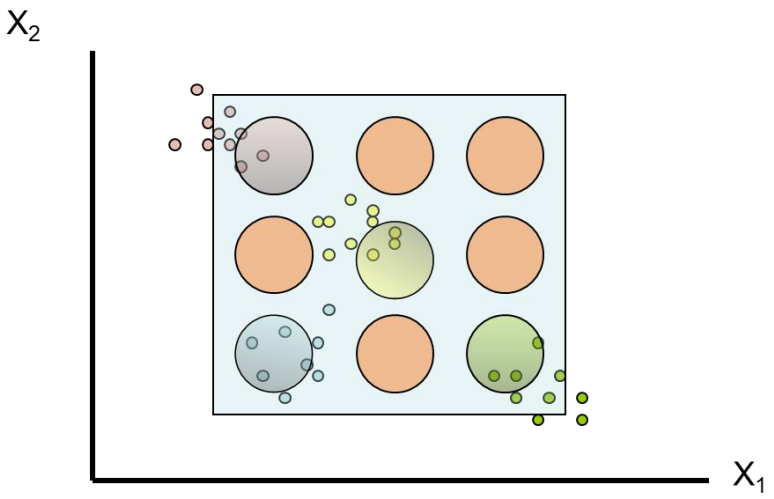


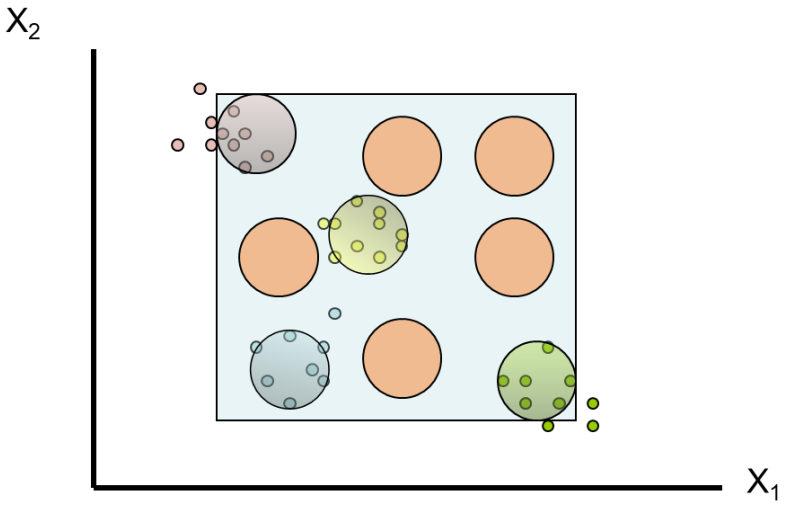
From Wikipedia



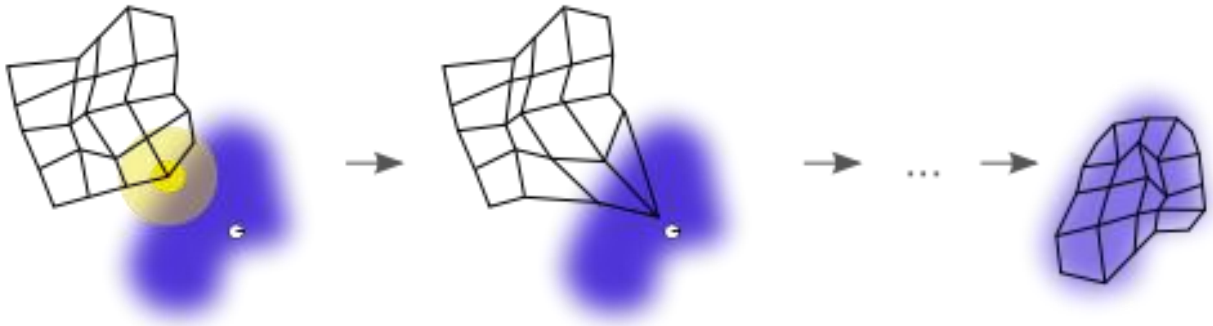
SOM 진화과정

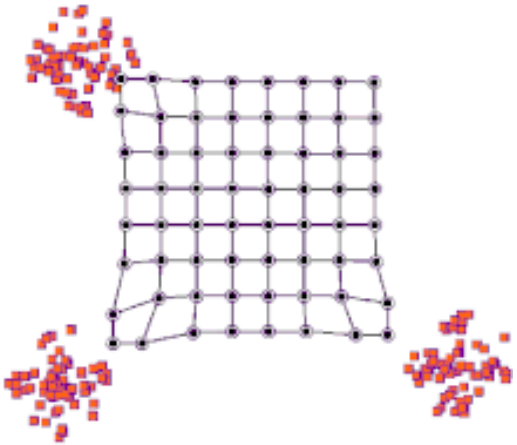


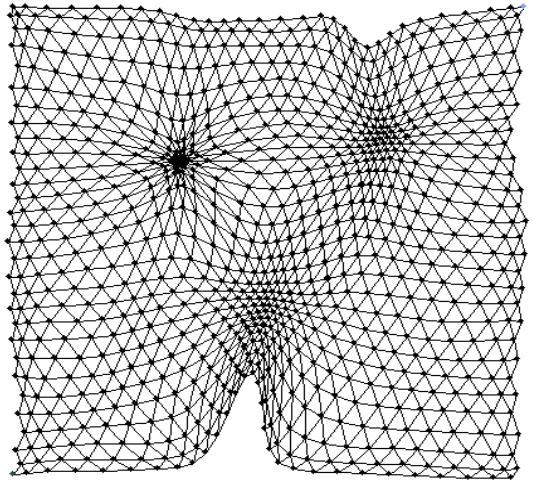




학습 후 SOM



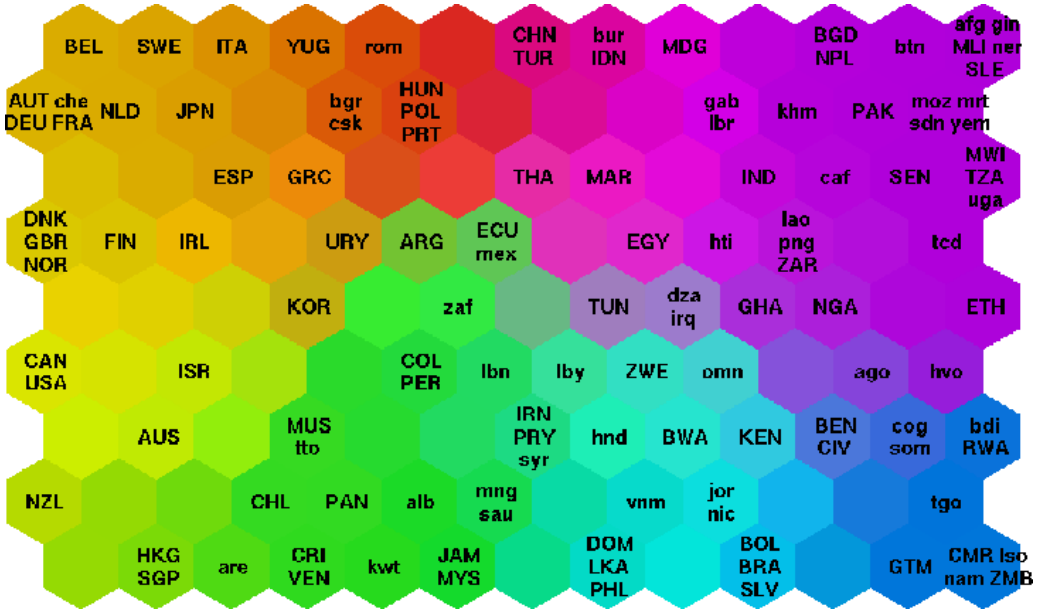




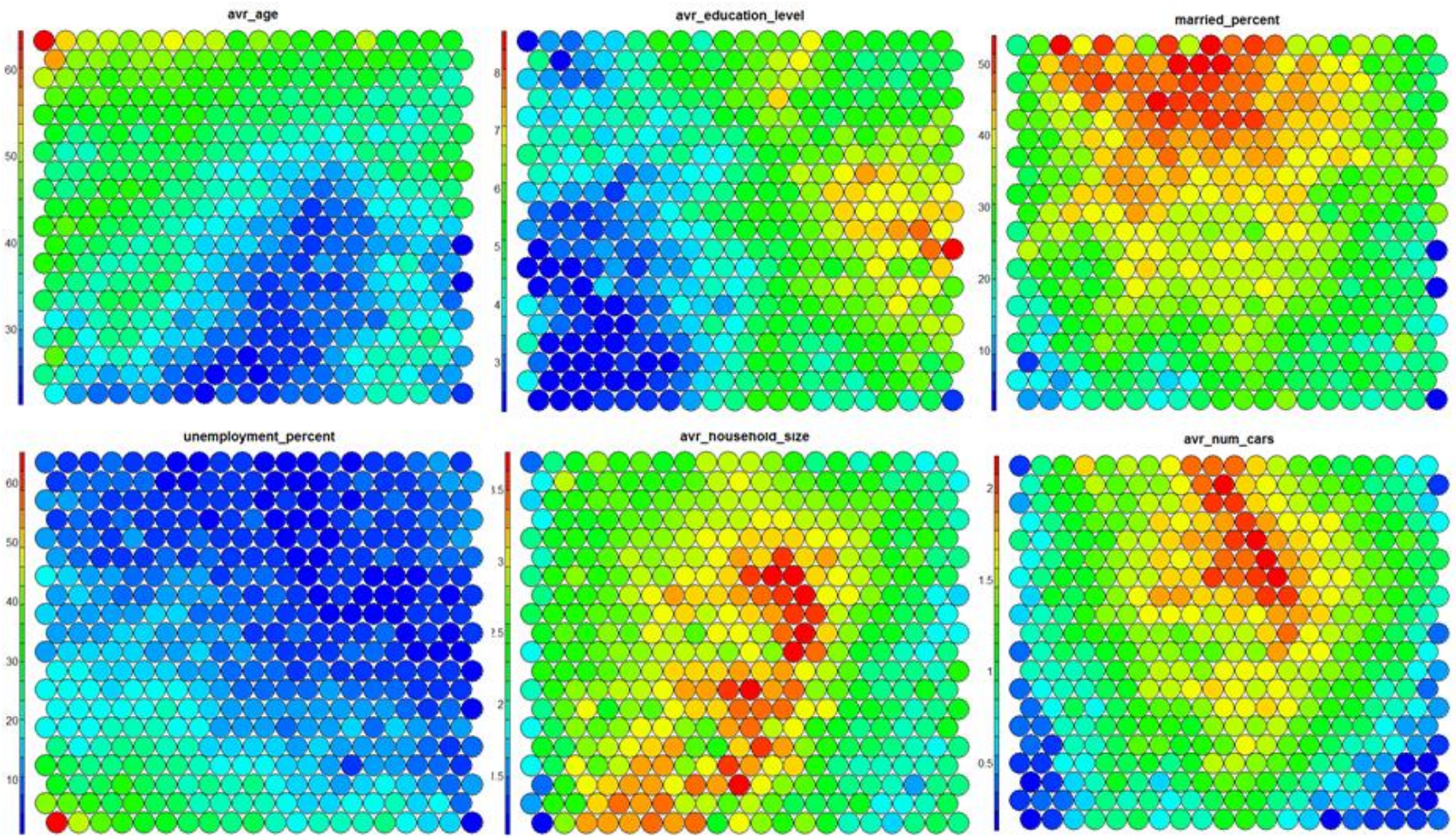
From Wikipedia

World Poverty Map

- The Self-Organizing Map (SOM) can be used to portray complex correlations in statistical data.
- Here the data consisted of World Bank statistics of countries in 1992.
- Altogether 39 indicators describing various quality-of-life factors, such as state of health, nutrition, educational services, etc, were used.



Customer segmentation: Case of Dublin



https://www.shanelynn.ie/self-organising-maps-for-customer-segmentation-using-r/