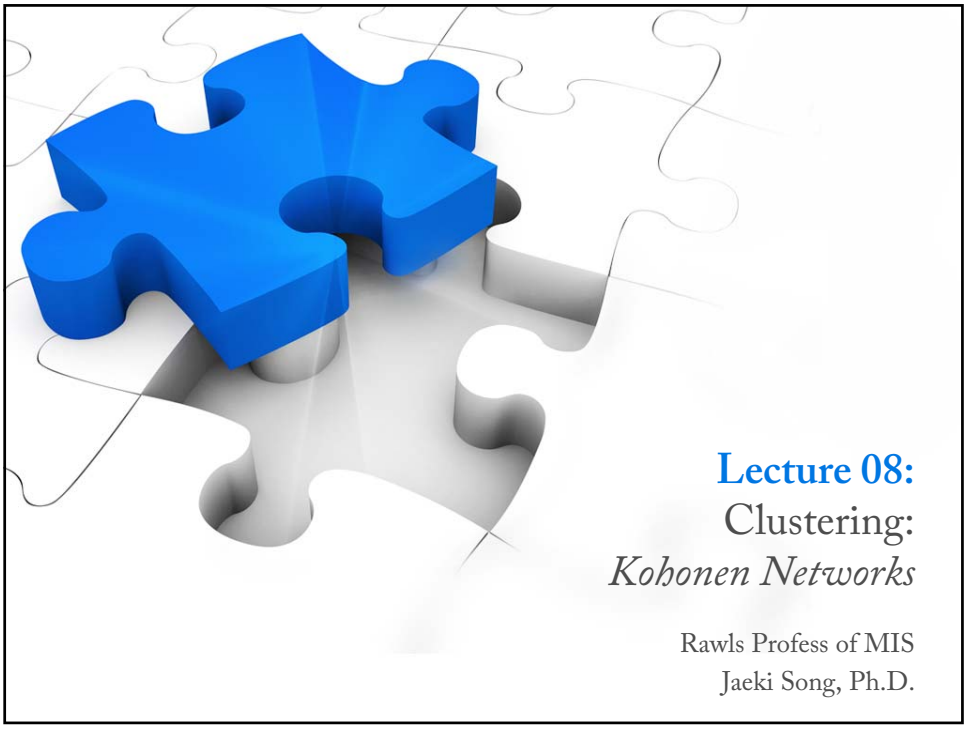


Lecture 08:
Clustering:
Kohonen Networks

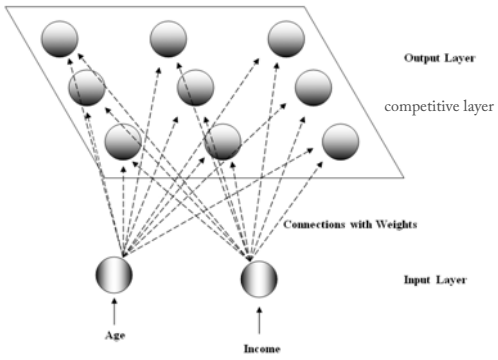
Rawls Profess of MIS
Jaeki Song, Ph.D.

Self-Organizing Maps (SOM)

- Applied initially to image and sound analysis by Finnish researcher Tuevo Kohonen in 1982
- Special Class of Neural Network
- Goal of SOMs is to convert high-dimensional signal into low-dimensional discrete map
- Output is structured as cluster of nodes where nodes in closer proximity are more similar to each other

SOM Topology

- Clustering Records by Age and Income
- Input Layer at bottom with one input node for each field



Neural Networks?...

- SOM's are feedforward, they do not allow looping or cycling
- SOM's are completely connected, every node in a given layer is connected to every node in the next layer but not those in the same layer
- Each connection has a weight initialized randomly to between zero and one
- Variable values need to be normalized and standardized

Characteristics of SOMs

- No hidden layer
- Competition
  - Output nodes compete to produce best value for a scoring function, typically Euclidean distance.
- Cooperation
  - The winning node is center of a neighborhood of excited neurons which all share in the "reward" of adaptation.
- Adaptation
  - Neighboring nodes to the winning node participate in this learning by having their weights adjusted to improve the scoring function

Kohonen Networks

- SOMs that exhibit Kohonen learning in which nodes in the neighborhood of winning node adjust weights using

$$w_{ij,\text{new}} = w_{ij,\text{current}} + \eta(x_{ni} - w_{ij,\text{current}})$$

Example

- Data set with two attributes, age and income and would like to use a 2 x 2 Kohonen network to uncover clusters
- Set neighborhood size to R = 0 and learning rate to 0.5
- Assume random initial weights as

$$w_{11} = 0.9 \quad w_{21} = 0.8 \quad w_{12} = 0.9 \quad w_{22} = 0.2$$
$$w_{13} = 0.1 \quad w_{23} = 0.8 \quad w_{14} = 0.1 \quad w_{24} = 0.2$$

- For the first input vector x_1 = (0.8, 0.8) we perform the competition, cooperation, and adaptation steps

Example

- Competition: Compute Euclidean distance between input vector and the weight vectors for each node:

Node 1: $D(w_1, x_1) = \sqrt{\sum_i (w_{i1} - x_{1i})^2} = \sqrt{(0.9 - 0.8)^2 + (0.8 - 0.8)^2} = 0.1$
Node 2: $D(w_3, x_1) = \sqrt{(0.9 - 0.8)^2 + (0.2 - 0.8)^2} = 0.61$
Node 3: $D(w_3, x_1) = \sqrt{(0.1 - 0.8)^2 + (0.8 - 0.8)^2} = 0.70$
Node 4: $D(w_4, x_1) = \sqrt{(0.1 - 0.8)^2 + (0.2 - 0.8)^2} = 0.92$

Node 1 wins since it minimizes score function D between input record and vector of weights over all nodes.

Example

- Cooperation
  - Neighborhood size R = 0 means cooperation is nil and only winning node 1 is rewarded
- Adaptation
  - For node 1, for the first record for each field

For *age*: $w_{11,\text{new}} = w_{11,\text{current}} + 0.5(x_{11} - w_{11,\text{current}})$
$= 0.9 + 0.5(0.8 - 0.9) = 0.85$

For *income*: $w_{21,\text{new}} = w_{21,\text{current}} + 0.5(x_{12} - w_{21,\text{current}})$
$= 0.8 + 0.5(0.8 - 0.8) = 0.8$

  - Weights nudged in the direction of fields' values of input record

Example

- For the second input vector x2 = (0.8, 0.1)
- Node 2 wins competition because its weights are more similar to the field values for this record

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

- Adaptation: For node 2, adjust the weights

For *age*: $w_{12,\text{new}} = w_{12,\text{current}} + 0.5(x_{21} - w_{12,\text{current}})$
$= 0.9 + 0.5(0.8 - 0.9) = 0.85$

For *income*: $w_{22,\text{new}} = w_{22,\text{current}} + 0.5(x_{22} - w_{22,\text{current}})$
$= 0.2 + 0.5(0.1 - 0.2) = 0.15$

Example

- For the third input vector x3 = (0.2, 0.9)
- Node 3 wins competition because its weights are more similar to the field values for this record

$$\textit{Node 3}:\ D(w_3, x_3) = \sqrt{(0.1-0.2)^2 + (0.8-0.9)^2} = 0.14$$

- Adaptation: For node 3, adjust the weights

$$\text{For } \textit{age}:\ w_{13,\,\text{new}} = w_{13,\,\text{current}} + 0.5(x_{31} - w_{13,\,\text{current}})$$
$$= 0.1 + 0.5(0.2 - 0.1) = 0.15$$

$$\text{For } \textit{income}:\ w_{23,\,\text{new}} = w_{23,\,\text{current}} + 0.5(x_{32} - w_{23,\,\text{current}})$$
$$= 0.8 + 0.5(0.9 - 0.8) = 0.85$$

Example

- For the fourth input vector x4 = (0.1, 0.1)
- Node 4 wins competition with minimum distance to the fourth record's field values

$$\textit{Node 4}:\ D(w_4, x_4) = \sqrt{(0.1-0.1)^2 + (0.2-0.1)^2} = 0.10$$

- Adaptation: For node 4, adjust the weights

$$\text{For } \textit{age}:\ w_{14,\,\text{new}} = w_{14,\,\text{current}} + 0.5(x_{41} - w_{14,\,\text{current}})$$
$$= 0.1 + 0.5(0.1 - 0.1) = 0.10$$

$$\text{For } \textit{income}:\ w_{24,\,\text{new}} = w_{24,\,\text{current}} + 0.5(x_{42} - w_{24,\,\text{current}})$$
$$= 0.2 + 0.5(0.1 - 0.2) = 0.15$$

Kohonen Networks Example Clusters

- Four output nodes represent four distinct clusters uncovered by Kohonen Network

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