

Lecture 22

Clustering (3)

Outline

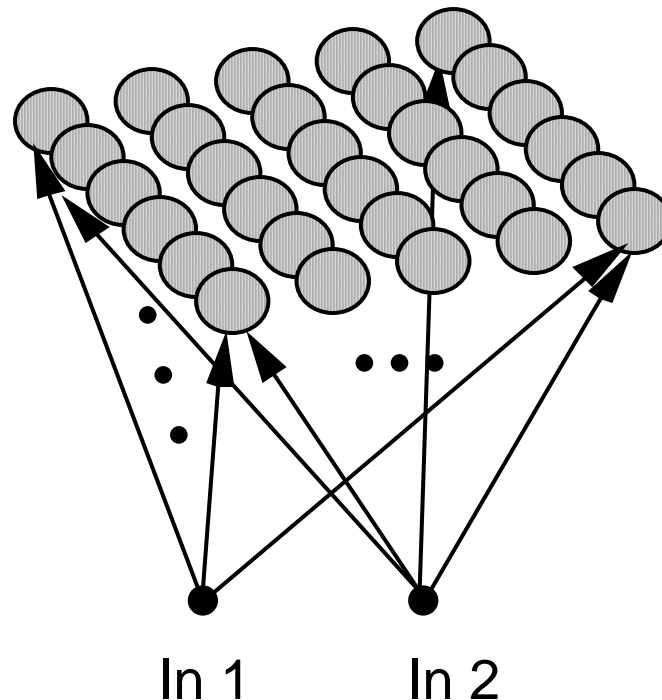
- Self Organization Map:
Structure
Feature Map
Algorithms
Examples

Introduction to SOM

- Both SOM and LVQ are proposed by T. Kohonen.
- **Biological motivations:** Different regions of a brain (cerebral cortex) seem to tune into different tasks.
Particular location of the neural response of the "map" often directly corresponds to specific modality and quality of sensory signal.
- **SOM** is an unsupervised clustering algorithm which creates spatially organized "*internal representation*" of various features of input signals and their abstractions.
- **LVQ** is a supervised classification algorithm which is obtained by fine-tuning the result obtained using SOM initially.

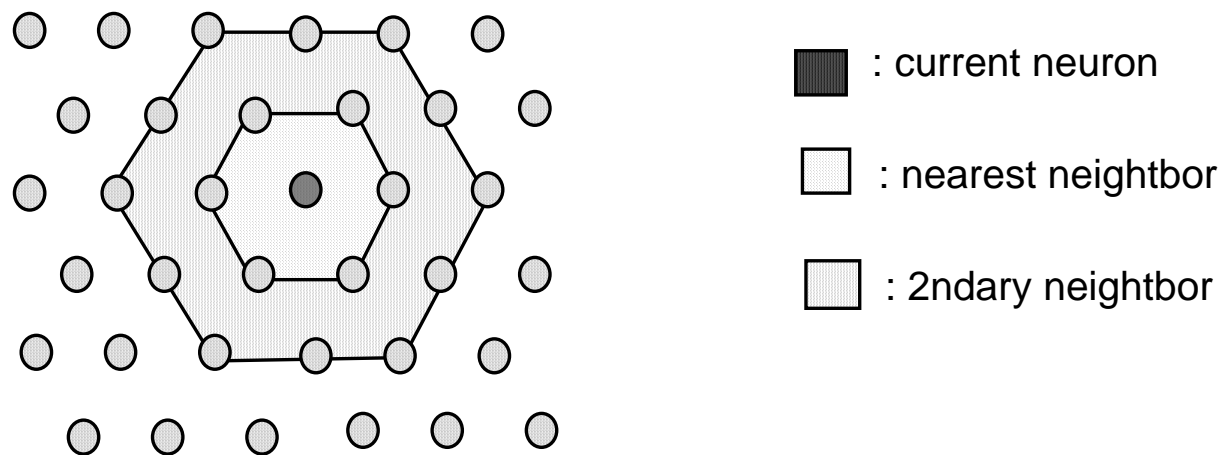
SOM Structure

- Neurons are spatially organized (indexed) in an 1-D or 2-D area.
- Inputs connect to every neurons.



Neighborhood Structure

- Based on the topological arrangement, a "neighborhood" can be defined for each neuron.



- Linear and higher dimensional neighborhood can be defined similarly.

Feature Map

- The neuron output form a low dimension map of high dimensional feature space!
- Neighboring features in the feature space are to be mapped to neighboring neurons in the feature map.
- Due to the gradient search nature, initial assignment is important.

SOM Training Algorithm

Initialization: Choose weight vectors $\{w_m(0); 1 \leq m \leq M\}$ randomly. Set iteration count $t = 0$.

While Not_Converged

Choose the next x and compute $d(x, w_m(t)); 1 \leq m \leq M$.

Select $m^* = \text{mim}_m d(x, w_m(t))$

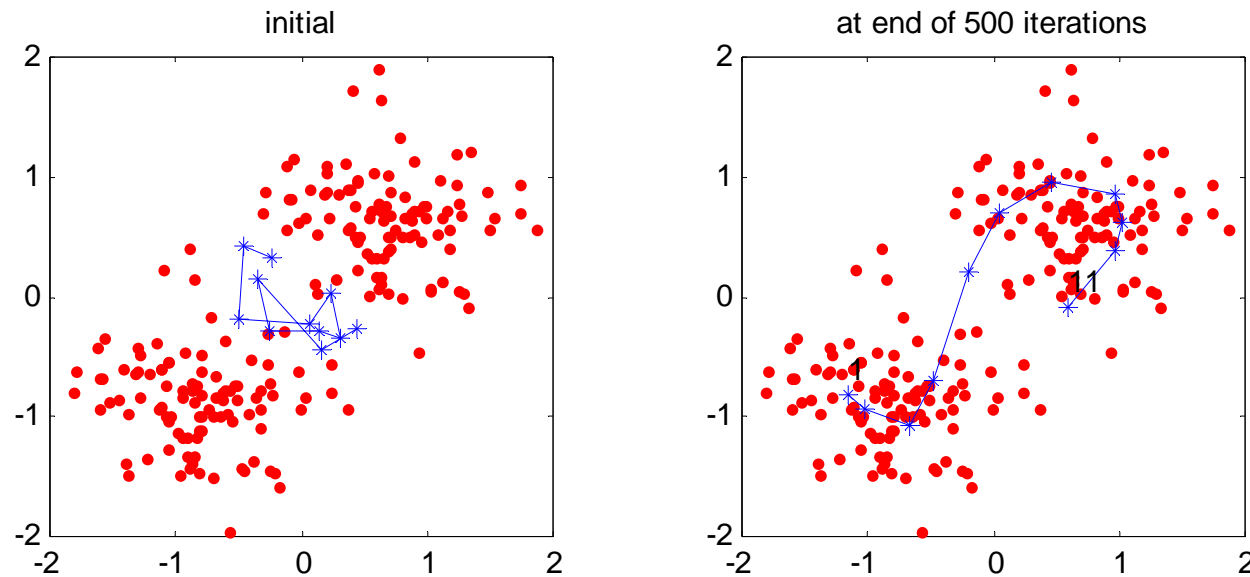
Update node m^* and its neighborhood nodes:

$$w_m(t+1) = \begin{cases} w_m(t) + \eta(x - w_m(t)) & m \in N(m^*, t); \\ w_m(t) & m \notin N(m^*, t) \end{cases}$$

If Not_converged, then $t = t+1$

End % while loop

Example



- Initially, the code words are not ordered as shown in tangled blue lines.
- At the end, the lines are stretched, and the wiring is untangled.

SOM Algorithm Analysis

- Competitive learning – neurons competes to represent the input data. Winner Takes it All!
- Neighborhood updating: the weights of neurons fall within the winning neighborhood will be updated by pulling themselves toward the data sample. Let $\mathbf{x}(t)$ be the data vector at time t , if all the data vectors $\mathbf{x}(t)$ have been nearest to $W_m(t)$, then

$$\begin{aligned} W_m(t+1) &= W_m(t) + \eta \cdot (\mathbf{x}(t) - W_m(t)) = (1-\eta)W_m(t) + \eta \cdot \mathbf{x}(t) \\ &= (1-\eta)^{t+1} \mathbf{w}_m(0) + \eta \sum_{k=0}^t (1-\eta)^k \mathbf{x}(t-k) \end{aligned}$$

for $\mathbf{x}(t)$'s that m is in the winner neighborhood.

- The size of the neighborhood is reduced as t increase. Eventually, $N(m^*,t) = m^*$.

More SOM Algorithm Analysis

- A more elaborate update formulation:

$$w_m(t+1) = w_m(t) + \eta(m^*,t) (x - w_m(t))$$

where $\eta(m^*,t) = 0$ if $m \notin N(m^*,t)$, and for example,

$$\eta(m^*,t) = h_0 \exp(-|m-m^*|^2/s^2(t)) \text{ if } m \in N(m^*,t).$$

- Distance measure $d(x, w_m(t)) = \|x - w_m(t)\|$.
Other definition of distance may also be used. 24

SAMMON Mapping

- Visualization of high dimensional data structure!
- The map developed in SOM can serve such a purpose.
- In general, this is a multi-dimensional scaling problem : Distances (δ_{ij}) between low-D points $\{y_i\}$ in the map correspond to the dissimilarities (d_{ij}) between points $\{x_i\}$ in the original space.
- Optimization Problem: Given $\{x_i\}$, find $\{y_i\}$ to minimize

$$J_{\text{ef}} = \left[\sum_{i < j} \delta_{ij} \right]^{-1} \sum_{i < j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}}$$
- Optimize with gradient search for some initial selection of $\{y_i\}$.

$$\nabla_{y_k} J_{\text{ef}} = \frac{2}{\sum_{i < j} \delta_{ij}} \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}} \cdot \frac{y_k - y_j}{d_{kj}}$$
- Other criteria may also be used.