Computational Intelligence: Methods and Applications

Lecture 10
SOM and Growing Cell Structures

Włodzisław Duch
SCE, NTU, Singapore
Google: Duch

Italian olive oil

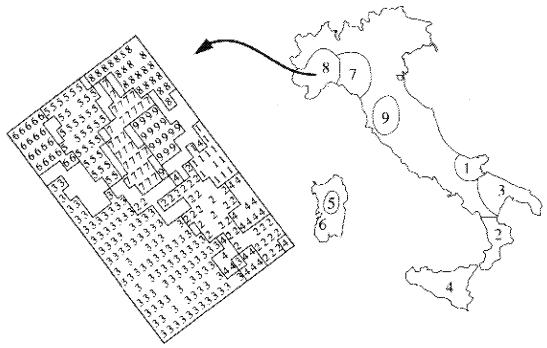
An example of SOM application:

572 samples of olive oil were collected from 9 Italian provinces.

Content of 8 fats was determine for each oil.

SOM 20 x 20 network, Maps 8D => 2D.

Classification accuracy was around 95-97%.



Note that topographical relations are preserved, region 3 is most diverse.

SOM properties

General properties of the algorithm:

- Hard to prove anything in d=2 or more dimensions about convergence properties of SOM maps.
- In d=1 analytical results for continuous time (infinitely many small step iterations) show that correct ordering of points is obtained by the SOM algorithm.
- Reducing the neighborhood too quickly leads to twisted configurations – forming quite wrong representations of data ("strange opinions"?).
- To avoid such effects convergence should be very slow, typically 10^4-10^6 iterations.
- Most adjacent neurons point to adjacent feature space regions, but kinks are possible since d-dimensional topography cannot be represented in 2D – try to do this with a map of a globe.

More SOM properties

- Complexity of the algorithm is O($KndN_{it}$) for K nodes (processors), d dimensions, N_{it} iterations n vectors in d-dimensions: all distances have to be computed and compared, therefore maps are not too large $K \sim 10^2$–10^4.
- Parallelization: easy to compute distances with K physical processors, but N_{it} iterations is still large.
- **Classification quality is poor**.
  Kohonen: SOM is mainly for visualization but ... visualization is also poor, there is no estimation of map distortion.
- SOM algorithm does not optimize any cost function measuring distortion of visualization!

Quantization error

How to evaluate the errors that SOM maps make?

Data in feature space region where node *c* is the winner is replaced by average W_c of winner node. For *m* nodes in the SOM network

local quantization error is a sum over all X vectors for which W_c is the winner (the algorithm does the opposite, finds a winner to update):

$$\varepsilon_c(\mathbf{W})^2 = \sum_{\mathbf{X}\in O(\mathbf{W}_c)} \|\mathbf{X}-\mathbf{W}_c\|^2$$

Map of local quantization errors gives an idea where this approximation is acceptable. The total quantization error is a sum over all SOM nodes:

$$E_q(\mathbf{W}) = \left(\sum_{c=1}^{m} \varepsilon_c(\mathbf{W})^2\right)^{1/2}$$

Hierarchical SOM maps: rough description of all data, separate maps for subsets of data, and more precise maps for even smaller subsets.

Growing Cell Structures (GCS)

Constructive SOM (Fritzke 1993), many versions exist.

Initial topology: for k-D maps (*k*=1, 2, 3) (*k*+1)-dim. simplex , i.e. connect *k*+1 nodes with each other: line, triangle, pyramid.

Idea: add new nodes whenever quantization error is large.
Calculate average quantization error for each node for all {X};
find the worst node,
find its least similar (in the data space) neighbor
add new node between these two nodes, interpolating their parameters.

SOM algorithm is used for training, but adaptation is always only for the winner and its nearest neighbors (linked in physical space).

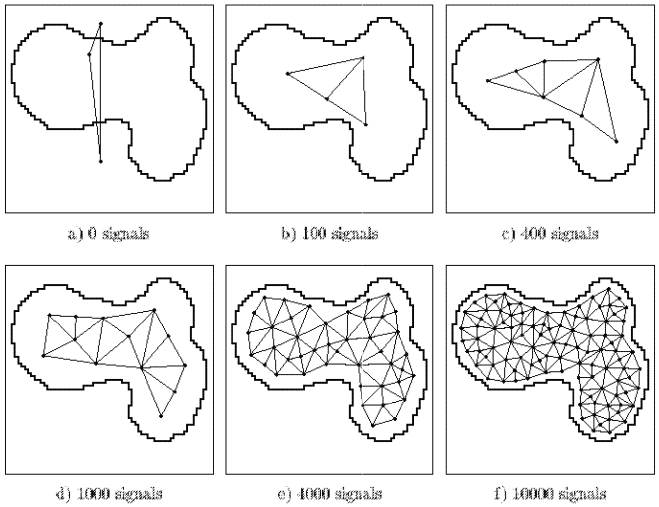
Many other competitive learning algorithms like that have been invented and are suitable for visualization.

See the Growing Self-Organizing Networks demo and the report “Some Competitive Learning Methods”.

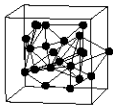
GCS algorithm

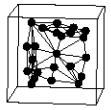
1. Create initial network topology, initialize node parameters $W^{(i)}$, for i = 1.. k+1, with small values.
2. Repeat steps 2-5 for t = 1 .. L (number of presentation epochs): Randomly select input vector X, and find the winner node c.
3. Move the winner node towards the input $\Delta W_c = \eta_c(X - W_c)$.
4. Increase frequency counter $\Delta\tau_c = \Delta\tau_c(t+1) - \Delta\tau_c(t) = 1$ of the winner node (alternatively, count quantization errors).
5. Correct parameters of its direct neighbors $\Delta W_n = \eta_n(X - W_n)$.
6. After L epochs calculate renormalized frequency for each node, $f_i = \tau_i / \Sigma_j \tau_j$
7. Find node with highest frequency W_1 (alternatively – find node with the largest quantization error);
find direct neighbor W_2 with largest distance $||W_1 - W_2||$;
add new node $W_n = (W_1 + W_2)/2$ in between, link W_1, W_2, W_n.
8. Stop if total number of epochs is too large, or quantization error is sufficiently small; otherwise repeat steps 2-7.

GCS growth

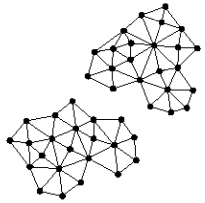


GCS - 2 clusters





In 3-D feature space there are 2 separate data clusters.



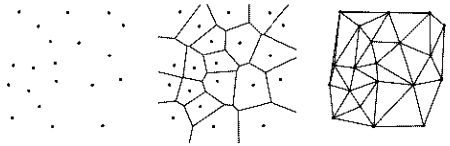
GCS network growing in 2D finds appropriate topology of the network.

Growing grid tries to preserve rectangular structure, adds rows of nodes instead of a single one.

Neural gas adds new unit between first and second winner after λ steps.

Voronoi and Delaunay

Data points | Voronoi decision borders | Delaunay triangulation

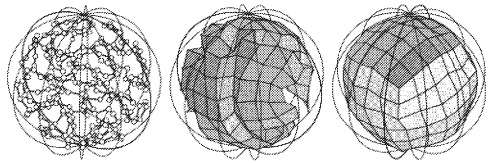


Voronoi diagram, tessellation – node in the center (codebook vector) is the winner, lines (planes) show decision borders, forming a Voronoi cell.

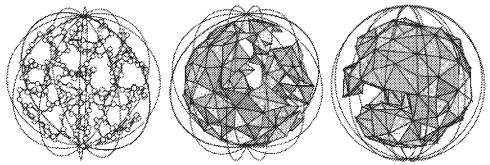
Voronoi set – all vectors inside the Voronoi cell.

Connecting nodes that have Voronoi cells with common edge **Delaunay triangulation** is obtained.

Ball in 3D



SOM: 216 nodes in 1D, 18x12 nodes in 2D and 6x6x6 nodes in 3D.



GCS: 216 nodes in 1D, 2D and 3D.

Some examples of real-life applications

Helsinki University of Technology web site

http://www.cis.hut.fi/research/refs/

has a list (old) of > 5000 papers on SOM and its applications !

- Brain research: modeling of formation of various topographical maps in motor, auditory, visual and somatotopic areas.
- AI and robotics: analysis of data from sensors, control of robot's movement (motor maps), spatial orientation maps.
- Information retrieval and text categorization.
- Clusterization of genes, protein properties, chemical compounds, speech phonemes, sounds of birds and insects, astronomical objects, economical data, business and financial data ....
- Data compression (images and audio), information filtering.
- Medical and technical diagnostics.

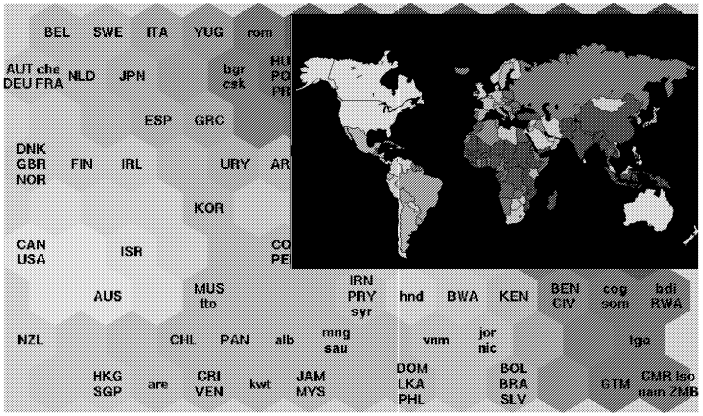
More examples

- Natural language processing: linguistic analysis, parsing, learning languages, hyphenation patterns.
- Optimization: configuration of telephone connections, VLSI design, time series prediction, scheduling algorithms.
- Signal processing: adaptive filters, real-time signal analysis, radar, sonar seismic, USG, EKG, EEG and other medical signals ...
- Image recognition and processing: segmentation, object recognition, texture recognition ...
- Content-based retrieval: examples of WebSOM, Visier, PicSom – similarity based image retrieval.

  Check the links to maps for astronomical catalogs and journals!
  http://www.ntu.edu.sg/home/aswduch/CI.html#SOM

Quality of life data

WorldBank data 1992, 39 quality of life indicators.
SOM map and the same colors on the world map.
More examples of business applications from http://www.eudaptics.com/



SOM software

- A number of free programs for SOM were written.
- Best visualization is offered by Viscovery free viewer
  http://www.eudaptics.com

  It can be used with free SOM_pack software from
  http://www.cis.hut.fi/research/som_lvq_pak.shtml
- Growing Self-Organizing Networks demo (demoGNG)