Self-Organizing Maps
Basics

• Unsupervised
• Competitive learning
• Clustering technique
• Reduce dimensions
• Display similarities
Why am I interested in SOM?

• Classify electronic tongue data
  – PCA

Vector Quantization (VQ)

- Data compression method
- Data of approximately same value are grouped together and assigned a single index/codeword value.
- Voronoi tessellation, illustrate VQ

* codebook or reference vectors
VQ

- Input vector $x$
- Codebook vector $m_i$, $i = 1, 2, \ldots, k$
Human Brain

- Organized spatially into regions according to sensory functions
  - performing special tasks, speech, analysis of sensory signals
- Structure in brain
  - minimize the wiring between functions in close contact
  - separated (in location) responses to minimize “crosstalk” between function
Self-organization

- Simultaneous development of both structure and parameter in learning
- The evolution of a system into an organized form, under the control of the system itself.
Unsupervised, competitive learning
SOM

• Nonparametric regression process
• Unsupervised classification
• Used for:
  data compression, process analysis, machine perception, control and communication
• Resembles the VQ
• Difference from VQ
  – $m_i$ is defined in such way that the mapping is ordered and descriptive of the distribution of $x$
Initial

- Input samples \( x = [\xi_1, \xi_2, \ldots, \xi_n]^T \in \mathbb{R}^n \)
- A set of units \( i \) in a grid
  - usually two-dimensional array
- Each unit \( i \) is assigned a weight
  \[ m_i = [\mu_{i1}, \mu_{i2}, \ldots, \mu_{in}]^T \in \mathbb{R}^n \]
  - \( m_i \) is selected randomly, same dimensions as the input data
  - \( x(t) \) and \( m_i \) can be vectors, strings of symbols, etc.
Winner

- Compare each $x(t)$ with all the $m_i$
- Best matching unit, $c$
  - Euclidean distance

$$c = \arg \min_i \{ \| x - m_i \| \}$$
Learning

• Learning, non-linear projection.

\[ m_i(t + 1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \]

• Neighborhood function \( h_{ci}(t) \)

• For convergence, \( h_{ci}(t) \to 0 \) when \( t \to \infty \)
Neighborhood

- Neighborhood set of array point around node $c$, $N_c$
  
  $h_{ci}(t) = \alpha(t)$ if $i \in N_c$
  
  $h_{ci}(t) = 0$ if $i \notin N_c$

- Where $\alpha$ is the learning rate factor
  
  $(0 < \alpha(t) < 1)$
Neighborhood

- Gaussian function

\[ h_{ci}(t) = \alpha(t) \cdot \exp \left( -\frac{\| r_c - r_i \|^2}{2\sigma^2(t)} \right) \]

- \( \| r_c - r_i \|^2 \) denotes the distance between the winner node \( c \) and input vector \( i \)

- Varies with time
  - \( \sigma \) width of the kernel
  - \( \alpha \) learning rate factor
Quality of SOM

- Quantization error
  \[ E\left\{ \| x - m_c \| \right\} \]

- Topology
  - neighborhood relations
Variants of SOM

• $x$ can be normalized,
  – may improve numerical accuracy because the resulting reference vectors then tend to have the same dynamical range.

• Different matching
  – weighted Euclidean distance
  – not Euclidean

• Initialization of $m_i$
  – random/linear initialization
Learning Vector Quantization

- Supervised learning
- Purpose: define class regions in the input data space.
LVQ

Machine Learning 2005, Malin Lindquist
Drift

• A single map quickly becomes useless due to drift.
• The codebook vectors are moving towards the latest input when the same odor is presented or even towards another cluster.
mSOM

- Classification task.
- Each map associated to a single odor
- Enables a self adjustable process to the neurons in each local map and adapts to the new situation.
- Number of maps = the number of odors.
Procedure

- Each map is trained with observations belonging to the same odor.
- Refining process.
- Maximum quantization error for each map is found.
- Testing.
- Drift?
  - The distribution of the data stored changes and self-train, to re-adapt.
LVQ

- Refine each maps codebook vector to reduce the high uncertainty accumulated at the borders of different clusters

  Input data is presented to the different maps.
  In each map there is winning neuron and also one winning map.
  If the winning map label and input label match, the corresponding winning neuron will move towards the input vector. If the labels do not match the neuron will move away.
“Self-Organizing Maps” T. Kohonen, 2001