K-Means Representational Bias, Search Bias and Loss Function

- Represent clusters by their centroids
- Search using gradient descent
- Distortion/Vector Quantization Error
  \[ E = \frac{1}{2} \sum_{i} D(x_i, C_j), \text{where } x_i \in C_j \]
- Minimize distortion must manage information-modeling trade-off

\[ E_{\text{mse}}[\mathcal{X}(x)] = \omega_0 KL(Q_0 \Vert P_0) + \omega_1 KL(Q_1 \Vert P_1) + I(Q \mid F) \]

Properties of K-Means Algorithm - 1

1) The Effect Of Exclusive Assignment
   Can not model overlapping classes, Outliers
2) The Inconsistency of the learning algorithm
   \[ \lim_{n \to \infty} P(\theta_{\text{true}}) = 1 \]
   where \( n \) is the number of observations
3) The learning algorithm is not invariant to non-linear transformations
4) The learning algorithm is not invariant to scale transformations
Properties of K-Means Algorithm - 2

5) The learning algorithm finds the local minima of its loss function, which is the vector quantization error (distortion).

6) The learning algorithm provides biased class parameter estimates

![Graph showing two distributions]

Properties of K-Means Algorithm - 3

7) The learning algorithm requires the a-priori specification of the number of classes

8) Euclidean distance measures can unequally weight attributes

9) Non-parametric modeling of continuous attributes
Kohonen SOMs - Applications

- A different type/style of machine learning
  - Preprocesses and presents the data in a meaningful way
  - Does not explicitly make any decisions
- Useful in situations where:
  - Human expert monitoring is desirable:
    • Sensitive, complex tasks etc.
- Can’t be used in situations where:
  - Model is very complex
  - Need real-time decisions etc.

Mapping

A function \( f: A \to B \) is a mapping or total function if the domain of \( f \) is the whole of the set \( A \),

A function \( f: A \to B \) is surjective (or onto) if \( \text{Ran}(f) = B \). A surjective mapping is known as a surjection.

A function \( f: A \to B \) is injective (or 1-1) if its inverse \( f^{-1} \) is also a function. If \( f \) is not injective, then its inverse is a relation, not a function. An injective mapping is known as an injection.

A function is bijective if it is both injective and surjective. A bijective mapping is known as a bijection. Note that if \( f: A \to B \) is injective (1-1) and \( a \in \text{Dom}(f) \) then \( f^{-1}(f(a)) = a \);
also if \( b \in \text{Ran}(f) \) then \( f^{-1}(f^{-1}(b)) = b \).

SOM’s are mapping from a higher to lower dimensional space.
Other Types of Dimension Reduction

- Multi-dimensional scaling (MDS)
  - Next lectures discusses representing a model using MDS

Mapping that SOM Performs

- The SOM mapping is complex. We can say:
  - It can be surjective
  - It is not injective and therefore not bijective.
  - Preserves some topological features
    - “The property of topology preserving means that the mapping preserves the relative distance between the points. Points that are near each other in the input space are mapped to nearby map units in the SOM.”
SOM - Functionality

- Input: a series of instances
- Processing: Each neuron competes to represent the instance.
- Output: A maps that compresses the data and a list of instances won for each neuron.

SOM Structure

- One layer neural network.
- Neurons usually arranged in 2D grid.
- Neurons are completely connected to each other.
- Number of inputs into each neuron is the number of attributes.
- Neuron performs no computation but have a model/memory for each attribute.
Training a SOM - 1

Initialize model/memory/weights

- Step 1: Provide an instance to each neuron
- Step 2: Determine the neuron that represents the instance.
- Step 3: Calculate the winner’s neighborhood
- Step 4: Neurons adjust their model/memory

Size of neighborhood and adjustment decreases with time.

Training a SOM - 2

The Neuron that wins, $c$, has the model that is closest to the instance in Euclidean distance

$$
||x(t) - m_c(t)|| \leq ||x(t) - m_i(t)|| \forall i. \quad (2)
$$

The neighborhood of a Neuron depends on time and spatial location

$$
h_{c(x),i} = \alpha(t) \exp \left( -\frac{||r_i - r_c||^2}{2\sigma(t)^2} \right), \quad (3)
$$

A neuron’s model is updated depending on distance to the winner and the instance

$$
m_i(t + 1) = m_i(t) + h_{c(x),i}(x(t) - m_i(t)), \quad (1)
$$

$\alpha$ is the learning rate that decreases monotonically with time, $t$.

$r_i$ and $r_c$ are models and $||r_i - r_c||$ the Euclidean distance between them.
Properties of SOM

• Takes a long time to train
  – Complexity is $O(s^2)$, $s$ is the number of neurons.
  – Use batch training
    • Parse all the data
    • Collect the list of instances won for each neuron
    • For each neuron calculate the centroid for the list
    • Update neuron’s memory/model using the centroid.
• Sensitive to initial model.

Using a SOM

• For each class determine its centroid.
• Need to determine $k$, (user discernable)
• Determining quality of SOM
• Construct a B/W distance map.
  – For each neuron calculate the mean distance between it’s model/memory and its neighbors.
  – Map a distance of 0 to white and a (normalized) distance of 1 to black
Black and White Distance Map - 1

Color Distance Map - 1