Self-Organizing Maps: An Introduction

Stephen D. Kachman
Department of Statistics

UNIVERSITY OF Nebraska
Lincoln
Introduction

- Self-Organizing maps were developed by Kohonen in the early 1980’s
- Original area was in the area of speech recognition
- Major features
  - Method of clustering high dimensional data
  - Resulting clusters are arranged on a grid
• Two areas of applications in Bioinformatics
  – Using a one dimensional grid to order the branches of a hierarchical tree
  – Using a two dimensional grid of genes clusters from microarray data.

• Algorithm
  – Mapping two dimensional onto a one-dimensional grid
  – General algorithm
Hierarchical Clustering

- Many genes measured on number of lines
  - Normal or Altered
  - Expression levels

- Interested in which lines are more closely related
  - Measure of distance (Euclidean Distance)
  - Tree is built from the bottom up
  - Two closest lines are joined . . .
Branch Order

• Both the trees on the previous slide are equally valid

• A “new tree” can be obtained by flipping the branches at a node

• While equivalent, the trees convey different visual information

• We could place the branches in “alphabetical order”
Alphabetical Order

• A to the left of B

• Cluster AB to the left of C

• If the line names had a meaningful order

• Clusters with the lowest average order to the left
One dimensional self-organizing map

- $n$ lines are assigned to $k$ clusters

- Clusters are numbered 1 to $k$

- Branches are ordered by cluster number
Individual Clusters

• A line is said to belong to the closest cluster

• Distance is measured from the center (centroid or mean) of the cluster

• Calculate the mean of a cluster by averaging
Algorithm

• This is done dynamically using a weighted average

\[ m_c(t + 1) = (1 - \alpha_t)m_c(t) + \alpha_tx_t \]

• \( m_c(t) \) mean of closest cluster at iterate \( t \)

• \( 0 < \alpha_t < 1 \) weight \( \alpha_t \to 0 \) as \( t \to \infty \)

• \( x_t \) is the gene data for a randomly selected line
Ordering the clusters

• To order the clusters we also need to adjust nearby clusters

• Nearby is defined by the cluster number

• So if \( x_t \) is closest to cluster 4 (\( m_4(t + 1) \))
  – Also update clusters 5 and cluster 6
Algorithm

\[ m_i(t + 1) = [1 - \alpha_t h_t(\|c - i\|)]m_i(t) + \alpha_t h_t(|c - i|)x_t \]

- \( h_t(d) \to 0 \) as \( \|c - i\| \to \infty \)
- \( h_t(d) \to 0 \) as \( t \to \infty \)
- \( h_t(0) = 1 \)
• The effect is cluster 4 will move toward clusters 5 and 6

• Initially $x_t$ has a large impact on both the closest cluster and nearby clusters

• Early stage results in the ordering of the clusters

• Later $x_t$ only impacts the closest cluster

• Later stage results in fine tuning the cluster centers
Simulated Data

Iteration 0
Comparisons

- Hierarchical Clustering
  - Builds the tree

- Self-Organizing Map
  - Determines the branch order
  - Does not differentiate between lines within a cluster
  - Results in ordered clusters
Two-dimensional SOM

• Clusters can be arranged in an array
  – Cluster would be indexed by row and column number
  – Cluster 2-8 would be close to Clusters 1-8, 3-8, 2-7, and 2-9

• Gene expression microarrays
  – Look at differences across time, temperature, . . .

• Cluster genes based on their expression profile
Yeast Cell Cycle

- Data from Tamayo et al. (1999)
- 6601 genes at 18 time points
- Filter out genes which show a limited amount of variability
- Standardize to a mean of zero and variance of 1
Summary

- Well suited to exploratory data analysis
- Effective use requires knowledge of the underlying biology
- Tight clusters are more informative