Methods Paper Outline

Lee Panter

Introduction

Necessary background of method

- Introduced in 1982 by Dr. Kohonen Teuvo in his paper titled: "Self-organized formation of topologically correct feature maps" [1]
- SOMs emulate biological information compartmentalized processing in the human brain.
  - Inputs are organized into similar stimuli prior to processing
  - Sensory/memory/functional components of the brain
- Used extensively as a clustering, classification, pattern recognition, segmentation and visualization tool [2]
- Training method is distinct from reinforment & optimization based algorithms
  - Learning by observation rather than example [2]

Main objective of method

- Clustering, also useful for visualizations formed from data dimension reduction methods that preserve the original topological characteristics found in high dimensions in the low-dimensional representations.
- Maping, following training the algorithm will have defined a map that can classify high dimensional inputs into low-dimensional clusters.

How to identify that the method may be useful

- Requirements for use:
  - Objects in the predictor space can be "ordered" or structured
- Use when:
  - not sure what predictor variables mean
  - interested in partitioning variance into a specified number of groups
  - no outcome to guide a supervised process [2]
  - visualization in reduced dimension is needed with preserved topological relationships
  - visualization in reduced dimension is needed with non-linear dimensional reduction techniques
  - $>$ 20 observations (two or more output nodes)

What information the results of the method provide

- Will ultimately depend on the parameter specifications
- Can be stochastic in nature from random weight initialization
- Unsupervised (Training Portion)
  - How the whole data set, represented as a large, highly variable population can be represented as a set of smaller, more homogenous sub-populations.
  - How each group's average high-dimensional variability represents the same information as each individuals high-dimensional variability
- Supervised (Testing Portion)
  - Organized output to the input of a Neural Net
  - Classify unseen observations
  - Inferential & Hierarchical Modeling
    - Mapping Outcome nodes to Heirarchy of clusters within outcome
    - Mapping heirarchy nodes to further classifications based upon meta-data

What questions are addressed by the method

- How is my predictor space related to an outcome variable that I have not been able to measure?
- How can I visualize my predictor space, knowing that the order of the observations in the high-dimensional representation is going to be important?
- I have no idea what any of these covariates mean... but maybe they all relate to each other, and maybe that relationship can be used to explain my outcom. HOW?
- WOW, jeez... there sure are a lot of observations here. How about we just get into groups and pick someone to represent each group?
- Why not? Sure go for it, cant hurt.

Case Study Description

Details reguarding data set to be analyzed

- Data set is called IRIS
- Available in R Package `datasets`
  - use `data(iris)` to obtain after intsalling package
- 150 observations of 3 different species of Iris measurements
- 4 different quantitative measurements included:
  - Sepal length (cm)
    - min = 4.300, mean = 5.843, max = 7.900
  - Sepal Width (cm)
    - min = 2.000, mean = 3.057, max = 4.400

  - Petal length (cm)
    - min = 1.000, mean = 3.758, max = 6.900
  - Petal Width (cm)
    - min = 0.100, mean = 1.199, max = 2.500
- 1 Qualitative measure:
  - Species
    - Setosa: qty = 50
    - Versicolor: qty = 50
    - Virginica: qty = 50

Goals of the analysis

- Separate data into test/train sets using a random sample
- Train a SOM
- Obtain cluster-visualizations of data set
- Perform a heirarchical clustering of the output nodes
- Use heirarchical clustering to define mapping:

$$(\text{S.Len},\ \text{S.Wid},\ \text{P.Len},\ \text{P.Wid}) \longrightarrow \text{Species}$$

- Use the defined SOM to quantify accuracy using test data

Model and Methods

Clear mathematical specification of the method

- Topology Preserving Maps:
  - Let (X, d_X) and (Y, d_Y) be metric spaces
  - Supponse that each observation has p components.

    $$\mathbf{x}_i = (x_{i1}, \ldots, x_{iP})$$

    for $i = 1, \ldots, N$ where N is the number of observations
  - Also assume that each component has a numerical representation in the rea
  - A Self Organizing Map is a continuous function:

    $$\mathbf{F} : (X, d_X) \subseteq \mathbb{R}^P \longrightarrow (Y, d_y) \subseteq \mathbb{R}^Q$$

    where $Q = 1, 2$
- Algorithm:
  - Let the outcome space, (Y, d_Y) be an $(M \times K)$ lattice structure with nodes enumerated sequentially, so that node $j \in \{1, \ldots, M \times K\}$
  - For each node there will be $N \times M \times K$ weight vectors:

    $$\mathbf{W}_{ij} = (W_{ij1}, \ldots, W_{ijP})$$

    for $i = 1, \ldots, N$ and $j = 1, \ldots, M \times K$

- Competition
  - for a fixed i
  - weights are also calculated for each value of $j = 1, \ldots, M \times K$

$$\mathbf{W}_j = d_X\left(\mathbf{X}_i, \mathbf{W}_j\right)$$

  - a value of $\mathbf{W}_j^*$ is defined for all $j = 1, \ldots, N \times K$ based upon:

$$\mathbf{W}_j^* = \arg\min_j \; d_X\left(\mathbf{X}_i, \mathbf{W}_j\right)$$

    this is called the "winning node"
- Cooperation:
  - A neighborhood function $h_{j^*}(v, \epsilon(t), \alpha(t))$ to determine how many (and which) node weights in the vacinity of j^* will be altered in the following step, where:
  - t represents the algorithm iteration number
  - $N_{\epsilon(t)}(j^*) = \{v \in 1, \ldots, M \times K \mid d_Y(j^*, v) < \epsilon(t)\}$ is an "epsilon neighborhood" around j^* at time (t)
  - $\epsilon(t)$ is a monotonically decreasing function of t that represents the largest cross-sectional width/radius of $N_\epsilon(j^*)$. $\epsilon(t)$ is generally recommended to have a magnitude of one-half the outcome space at $t = 0$ and converge to zero
  - $\alpha(t)$ is a monotonically decreasing function of t that represents the learning rate. $\alpha(t) \in [0, 1]\ \forall t \in \mathbb{N}$ and it is recommended that $\alpha(t)$ converge to zero with algorithmic iteration.
  - neighborhood functions can be discrete as in:

$$h_{j^*}(v, \epsilon(t), \alpha(t)) = \begin{cases} h_{j^*}(v, \epsilon(t), \alpha(t)) & \text{if } v \in N_\epsilon(j^*) \\ 0 & \text{if } v \notin N_\epsilon(j^*) \end{cases}$$

    - or they can be continuous, as in:

$$h_{j^*}(v, \epsilon(t), \alpha(t)) = exp\left(\frac{-d^2(j^*, v)}{2\epsilon^2(t)}\right)$$

    $\forall v \in 1, \ldots, M \times K$
- Adaptation:
  - following the calculations of the winner $\mathbf{W}_j^*$ and weights $\mathbf{W}_j$, and the neighborhood calculations $h_{j^*}(j, \epsilon(t), \alpha(t))$ for each value of $j = 1, \ldots, M \times K$ for any given iteration value ($t = t_0$) the set of weights are updated to reflect the information gained by minimizing the distance (the competition step), using the update formula:

$$\mathbf{W}_j^{t_1} = \mathbf{w}_j^{t_0} + \alpha(t_0) * h_{j^*}(j, \epsilon(t_0), \alpha(t_0)) \left[\mathbf{X}(t_0) - \mathbf{W}_j^{t_0}\right]$$

    where for discrete time we assume that: $t_1 = t_0 + 1$

- The Competition, Cooperation, and Adaptation portions of the algorithm are repeated for each observation in the data until convergence is seen in the output layer. Recirculations through the data set may also be needed.

Assumptions of the method

- Objects in the predictor space poses comparable features
  - In order to group a set of observations those observations must be inherently comparable
- Clusters in the both the input and output space can be ordered
- Weight values are bounded
- Inputs are given equal weight

Experimental Design

Target Population

- Wide range of applications and populations of interest. Large variety of people who can benefit. Visualization of high dimensional data is very common, and a useful too. Non-linear optimization tasks common in location/resource/design allocation problems

What data is required? (quantitative, discrete, distributional assumptions, covariates, etc.)

- Covariate types should be considered in close coordination with the metrics chosen for the spaces used to represent input and output space.

- No distributional assumptions need to be made
- Missing data is interpreted as a functional part of the distribution (ie relevant information)

Commonly used pre-fitting or pre-processing steps before the method is applied

- All input variables standardized so that each variable has the same influence on the algorithm
- number of output nodes chosen to be around 10% of total observations

When should the method be applied?

Optimal scenarios

- Unknown or poorly understood predictor variables
- Observations need to be accurately classified
- Artificial Neural Network would be an appropriate method

How to identify when this method will be useful

Analysis and Results

- Interpretations of the final results/findings
- Clearly presented results in Tables & Figures

Discussion

- Discussion of results
- Next steps
- Strengths and weaknesses of the method
- Relationship to other methods
- Recommended resources

References

Appendix

- Necessary code or software istructions to carry out method
- Link to utilized data set
- Any other resources referred to or found helpful

---

1. Teuvo K (1982) Self-organized formation of topologically correct feature maps. *Biological cybernetics* 43: 59–69.

2. Asan U, Ercan S (2012) An introduction to self-organizing maps, In, *Computational intelligence systems in industrial engineering*, Springer, 295–315.