R Shiny's Self-Organizing Map

Zury Betzab Marroquin[1], Joshua Walsh[2], Trenton Wesley[3]
Scripps College[1], University of Alaska Southeast[2], Harvey Mudd College[3]

Center for Collaborative Studies in Mathematical Biology
Intercollegiate Biomathematics Alliance
Scholarship, Teaching, and Advanced Research Development

Introduction

There are similarity and dissimilarity measurements in high-dimensional data reduction to two-dimensions. R Shiny's Self-Organizing Map (SOM) app is for those who want to analyze and display similarities in high-dimensional data by creating a two-dimensional map of the data.

What is R Shiny?

R Shiny runs on R software, which is for statistical computing and graphics [1]. R Shiny allows users to share their data, share their analysis, and make an interactive app for others to experience [2].

What is a Self-Organizing Map (SOM)?

The SOM measures similarity in high-dimensional data with Euclidean distance and displays the similarities on a two-dimensional square map of nodes. The nodes have two features: the nodes are fixed into the square map, and the nodes are given values in the same dimension as the data. The nodes are *Self-Organizing* (the nodes *learn*) in the sense that each node and the nearby nodes (neighbors) slowly takes on values of the most similar data observation, until all the nodes of the *Map* mirror the data. So, the SOM reproduces the topology of the data with a discrete square map of nodes, whereby the data observations represented by the nodes are similar, and nearby nodes are like each other.

Application Outline

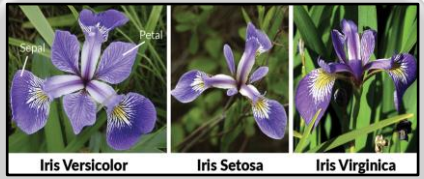
R Shiny's SOM app functions from R package '*kohonen*' [5] and has a three-tab interface: introduction, import data, and visualize data, with an option for complexity.

1.) Introduction Tab - explains what a SOM does and what the graphics display.

2.) Import Data Tab - uploads your data.

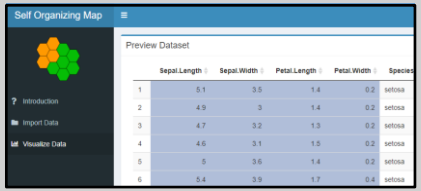
3.) Visualize Data Tab – displays graphics for similarity analysis and can retrain the map.

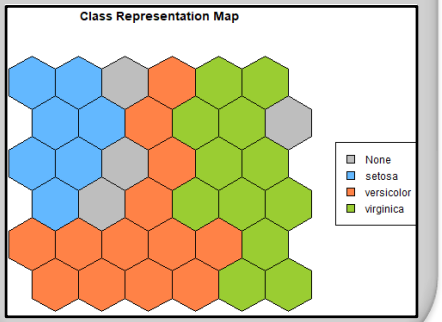
The parameters and methods that can be changed in the SOM algorithm are variable selection, toroidal map, map size, and clustering.



Iris Dataset

The Iris flower dataset provides a standard measurement for SOM efficacy. There are three types of iris flower with 50 observations each (150 observations total): iris setosa, iris virginica, and iris versicolor; additionally, four variables measure each flower: sepal length, sepal width, petal length, and petal width. The similarity in the Iris dataset is the two species iris virginica and iris versicolor, which are separated from the iris setosa.





How does the SOM work?

Data values are scaled between 0 and 1, then the nodes are given values from the data. Fig.(1)

Euclidean distance $\sqrt{o^2 + n^2}$ measures the distance between a *single* observation '*o*' from the data and *every* node '*n*' on the map. Fig.(1)

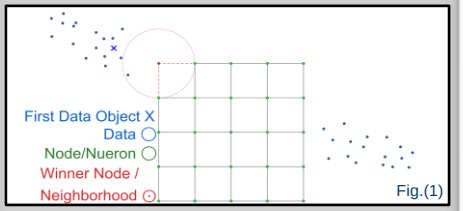


Fig.(1)

The Best Matching Unit (BMU) *learns*. The BMU is the node with the least distance to the observation, so $\min\sqrt{o^2 + n^2}$. *Learning* is adding value to the node such that the node becomes more like the observation. For example, say the BMU is the numeric pair (0,0) and the observation is (1,1), then learning can be shown for the BMU as (0.05,0.05), whereby the BMU's numeric inputs are closer to the observation's numeric inputs by 0.05. Fig.(2)

The BMU's neighborhood learns. The neighborhood are the nodes nearest the BMU. Fig.(2)

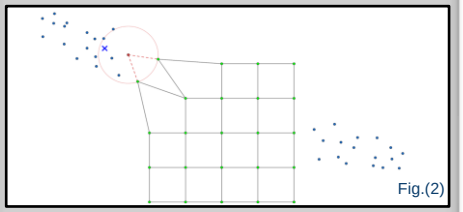


Fig.(2)

An *epoch* occurs when every observation is compared to the map of nodes. There are 100 epochs.

Only the BMU learns after $\frac{1}{3}$ of the total epochs. Fig.(3)

Multiple graphics display the data. Fig.(4)

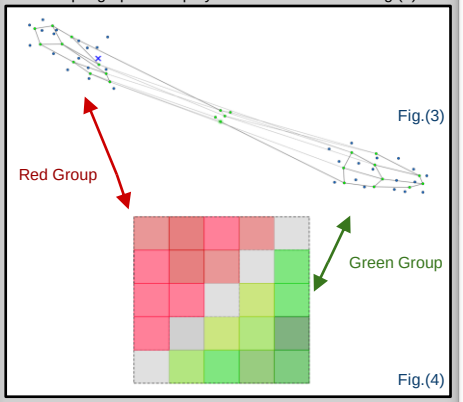


Fig.(3)
Fig.(4)

What next?

There are numerous articles and guides for constructing a SOM. The power is in extracting and displaying the similarities after the SOM has been built. Future contributions could add: a reduction in data dimensions before using the SOM (PCA or t.SNE), or an automatic cluster optimization package (Spectrum).

Acknowledgements

The authors appreciate CURE's research opportunities, and thank **Dr. Olcay Akman**, ISU; and **Dr. Christopher Hay-Jahans**, UAS, for mentoring and guiding our project.

Citations

1. The R Project for Statistical Computing. Retrieved from https://www.r-project.org/
2. Shiny from R Studio. Retrieved from https://shiny.rstudio.com/
3. Geogebra. Figures created using https://www.geogebra.org/geometry
4. Willems, K. (2020). Machine Learning in R for beginners. Image retrieved from https://www.datacamp.com/community/tutorials/machine-learning-in-r
5. Kruisselbrink, J. and Wehrens, R. (2019). Package 'kohonen'. Reference manual retrieved from https://cran.r-project.org/web/packages/kohonen/kohonen.pdf