Visualization of Multivariate Health Data Using Self-Organizing Maps

MARK LESTER GHANY
University of the Philippines Manila
Department of Physical Sciences and Mathematics
Padre Faura, Ermita, Manila
PHILIPPINES
comsci12chester@gmail.com

GEOFFREY SOLANO
University of the Philippines Manila
Department of Physical Sciences and Mathematics
Padre Faura, Ermita, Manila
PHILIPPINES
gasolano@upm.edu.ph

Abstract: Data that are multivariate in nature may contain subtle patterns. However, these types of data are considered to be an obstacle in reasearch most of the time since classical statistics may find it encumbering to analyze. The Self-Organizing Map is an artificial neural network trained using unsupervised learning. It enables the organization of data that unveils subtle patterns by representing it in much lower dimensional representations, typically two to three dimensions. Data Visualization on the other hand, gives the human brain a visual representation of the knowledge about the data. Together, they provide an image of the map together with the possible hidden patterns. The paper gives an application of the self-organizing map in the extraction of patterns and relationships in a large data set on the Philippine regional health data which includes thirty health and demographic variables.

Key–Words: Self-Organizing Maps, Data Visualization, Multivariate Data, Computational Statistics, Artificial Neural Networks

1 Introduction

The advent of database systems has tremendously increased the amount of data that are made available for analysis. Through automation, data have been properly indexed, and hence, are easily retrieved through queries. However, because database systems entail an easier way of gathering and storing data, most of the dataset became large and dimension while containing subtle patterns. Because of these characteristics, it may be encumbering for classical statistics to interpret and analyze these data alone. Thus, computational statistics may be put into use.

Knowledge Discovery and Data Mining are concepts in computer science, as well as computational statistics, that aim to search large datasets for patterns that may be considered as knowledge about the data [1]. These patterns may include anomalies, associations, clusters, and classes and to effectively present these knowledge, one of the tasks included in data mining is Data Visualization - giving the human brain a visual representation of the data [2].

A Self-Organizing Map (SOM) is a type of artificial neural network that is capable of discovering patterns in datasets. It is trained using unsupervised learning and employs a data compression technique known as vector quantization to reduce multidimensional data to a low-dimensional representation [3].

Visualizing a SOM may be considered as a post-processing step. A common algorithm used to visualize the lattice is the Unified Distance Matrix (U-matrix) - a technique which delivers a "landscape" of distance relationships of the input data in the data space [4]. Applying a clustering algorithm to a SOM is also a common choice in visualizing defined clusters in the data space [5]. For associations, we use component planes to see relationships between different variables. For other knowledge, further methods such as threshold classification may be applied to the map. And in cases where applicable, SOM and GIS may work hand in hand to present a better visualization of spatial data [6].

Geographic Information Systems (GIS) are systems designed to capture, store, manipulate, analyze, manage, and present all types of geographically relevant data [7]. In most cases, pattens that exist in spatial datasets are easier to uncover in geographic maps. Thus, there is also a need for the use of the GIS in the study.

The Regional Health Data obtained through sur-

ISBN: 978-1-61804-106-7

veys conducted by the Philippine Census is an example of a multidimensional data. It is considered as a complex system because several factors influence health including social, economic, and environmental variables that may differ per region. Data used in this study include thirty health and demographic variables in categories such as Fertility, Family Planning, Maternal Health, Childhood Mortality, Children's Health and Nutrition, HIV/AIDS - related Knowledge and Behavior, and Violence Against Women. The data is available through the NSO Website (http://census.gov.ph).

The main objective of this study is to produce a tool that would enable the visualization of the Philippine Regional Health Data to be able to uncover several patterns that may exist within the dataset. Thus, upon proper visualization, we may be able to use the results for further statistical tests. The tool should be able to make use of the Self-Organizing Map as well as its visualizations.

2 Self-Organizing Map

Self-organizing map (SOM), also known as Kohonen map, is a type of neural network based on competitive learning. It can be used for clustering and for visualization of high dimensional data by representing them in much lower dimensional spaces. It provides a topology preserving mapping from the high-dimensional space to map units. Map-units, or neurons, usually form a two-dimensional lattice and thus the mapping is from high-dimensional space onto a plane. The property of topology preserving means that the mapping preserves relative distance between the points. Points that are near each other in the input space are mapped to nearby map units in the map. Thus, the SOM can serve as a cluster analyzing tool of high-dimensional data [8].The Self-Organizing Map is a two dimensional array of neurons:

$$M = \{m_1, m_2, ..., m_{p \times q}\}$$

One neuron is a vector called the codebook vector:

$$m_i = [m_{i1}, m_{i2}, ..., m_{in}]$$

These neurons have the same dimension as the input vectors (n-dimensional) which contains the elements called weights. They are connected to adjacent neurons by a neighborhood relation. This dictates the topology, or the structure of the map. Usually, neurons are connected to each other either via rectangular or hexagonal topology.

Distance between the map units may also be defined according to their topology relations. Immediate neighbors (adjacent neurons) belong to the neighborhood N_c of the neuron m_c. The neighborhood function is a decreasing function of time: $N_c = N_c(t)$. Like any other artificial neural network, the self-organizing map has a learning algorithm. However, a self-organizing map does not need a target output to be specified unlike many other types of network. Training a SOM occurs in several steps and iterations:

1. Each node's weights are initialized.

   Prior to training the data, each node in the lattice should be initialized, typically set to small standardized random values. The nodes may be randomly initialized or randomly take samples from the data and use it for initialization.

2. An input vector is chosen at random from the set of data and presented to the lattice.

3. Every node is examined to calculate which one's weights are most like the input vector. The winning node is known as the Best Matching Unit (BMU).

   To determine the best matching unit, we commonly iterate through all the nodes in the lattice and recommend to calculate the Euclidean distance between the each node's weight vector and the current input vector. The Euclidean distance is given by:

   $$Dist = \sqrt{\sum_{i=0}^{n} (V_i - W_i)^2} \qquad (1)$$

   where V_i is an element of the current input vector and W_i is an element of the node's weight vector. However, in the tool, other distance measures that the user prefers may be used.

4. Determine the BMU's local neighborhood. The radius of the neighborhood is calculated through the neighborhood function. Generally, this is a value that starts large, typically set to the radius of the lattice, but diminishes each time-step. Any

ISBN: 978-1-61804-106-7

node found within this radius are said to be inside the BMU's neighborhood.

A unique feature of the Kohonen learning algorithm is that the area of the neighborhood shrinks over time that it is accomplished by reducing the radius over time. The default neighborhood function for the tool is given by:

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda}\right) \qquad t = 1, 2, 3, ... \quad (2)$$

Where σ_0 denotes the width of the lattice at time t_0 and λ denotes a time constant which is computed based on the number of iterations set and the current size of the neighborhood; t is the current time step which in implementation is the current iteration of the loop. The tool allows for revisions in the neighborhood function but generally it should shrink over time up to the size of one node.

5. Each neighboring node's weights are adjusted to make them more like the input vector. The closer a node is to the best matching unit; the more its weights get altered.

   Basically, to determine whether a node is within the neighborhood, we iterate through all the nodes to determine if they lay within the radius or not. Every node within the BMU's neighborhood has its weight vector adjusted according to the following equation:

   $$W(t+1) = W(t) + \Theta(t) L(t) (V(t) - W(t)) \quad (3)$$

   Where t represents the current time step, L is a variable called learning rate which decreases with time, and Θ is the amount of influence a node's distance from the BMU is on its learning. The equation states that the new adjusted weight for the node is equal to its current weight W, plus a fraction of the difference $\Theta \times L$ between the current weight and the input vector. This function is given by Dr. Kohonen in his paper [3].

   The preset decay of the learning rate is computed using the following equation:

   $$L(t) = L_0 \exp\left(-\frac{t}{\lambda}\right) \qquad t = 1, 2, 3, ... \quad (4)$$

   This is basically similar to the neighborhood function but is used to decay the learning rate. However, the tool allows the users to set this function according to their preference.

   The tool also allows for other learning decay functions including the power series:

   $$L(t) = L_0 \left(\frac{L_f}{L_0}\right)^{t/T} \qquad t = 1, 2, 3, ... \quad (5)$$

   Where L_f is the final learning rate and T is the total number of iterations.

   On the other hand, Θ by default is computed using:

   $$\Theta(t) = \exp\left(-\frac{dist^2}{2\sigma^2(t)}\right) \qquad t = 1, 2, 3, ... \quad (6)$$

   Where $dist$ is the spatial distance of the node from the BMU and σ is the width of the neighborhood function as calculated using equation (2).

6. Steps 2-5 are repeated for N iterations.

Though referred to as a neural network, self-organizing maps work differently from most known neural network algorithms which are commonly supervised. With regard to the SOM's convergence criterion, the SOM limits its training algorithm to a specific number of iterations. Since SOM is an unsupervised learning technique, there is no target vector and thus, we cannot obtain a difference between a target vector and the value at the current iteration [9]. And since the learning is a stochastic process, the statistical accuracy of the mapping depends on the number of iterations which must be reasonably large, Kohonen has devised a "rule of thumb" for the number of iterations in training[3].

3 Methodology

The development of the tool is the priority in this study. Thus, the first step is to develop the software

ISBN: 978-1-61804-106-7

using a specific platform and language. Majority of the tool's modules were developed using the Java Programming Language, however, the GIS part of the tool uses PHP, HTML, CSS, Javascript, and mySQL.

3.1 Training Parameters

1. *Map Width and Height:* A rule of thumb in deciding the map's size states that the number of neurons should be at least $10\times$ the number of dimensions of the dataset. For the Philippine Regional Health Data which has 30 variables, we used an 18×18 map for a total of 324 neurons.

2. *Initial Learning Rate:* Kohonen suggests that this be near the value of unity. We choose this value to be 0.8.

3. *Number of Iterations:* A rule of thumb suggests that the number of iterations be reasonably large from 10000 to possibly $500\times$ the number of neurons. We use 10000 iterations for this particular study.

The learning rate and neighborhood decay functions used for training the map are exponential functions that were default values for the tool developed. While the node influence used the Gaussian decay function.

3.2 Visualization

Several visualizations were used for this study. This includes the Similarity Matrix, Clusters, Component Planes, and GIS.

3.2.1 Similarity Matrix

The similarity matrix is a lattice in black and white which represents possible similarities and differences in the dataset. Light colored nodes that are near each other exhibit similarities while those with a dark border in between represents a difference.This provides a hint of the clustering of the data. It computes the node's distances from each other and give them colors on the grayscale color scheme. Through the U-matrix, we are able to detect which data entries are similar through each other and which are very different. The rule-of-thumb is data entries positioned near each other on the map that are light colored possibly forms a cluster while those with dark borders around them represent differences.

3.2.2 Clusters

Clusters in the dataset are made by clustering the map using the k-means clustering algorithm. The user may use the similarity matrix to predict the number of clusters that exist in the lattice and use it as an input for the k-means algorithm. Based on the results gathered, the similarity matrix proved to be an effective way to predict the number of clusters as they both suggest similar clusters. For accuracy, the algorithm was run 30 times and each visualization was examined. The cluster that appeared the most number of times is chosen as the dataset's cluster.

3.2.3 Component Planes

The Component Plane visualization shows the values of each variable by using a visual representation. Component planes were examined to be able to look for possible patterns and explanation. Using this visualization, the user may be given an idea of correlations that exist in the data space as well as assess the similarities that exist in each cluster of data given. It is also effective to use this type of visualization to look for trends that exist on the data space.

3.2.4 GIS

Lastly, the GIS visualization module provides opportunity for users to look at the visualization on a geographic map. This uses the values given by the Cluster and Component Planes, which are plotted in the Geographic Map, and thus, creating a geographic visualization. Since this is on a on the geographic map, this visualization provides a more intuitive presenation on trends and patterns that exist between geographic locations as compared with the lattice visualizations.

4 Results

The tool developed using Java was named SOM Visualize. It was then used to visualize the Philippine Regional Health Data, and the following shows the results.

The tool's User Interface is shown in the Figure 1.

Upon training the map using the tool, each visualization may be seen. First, the Similarity SOM. Figure 2 shows the Similarity SOM for the Philippine Regional Health Data. The figure has shown that among

ISBN: 978-1-61804-106-7

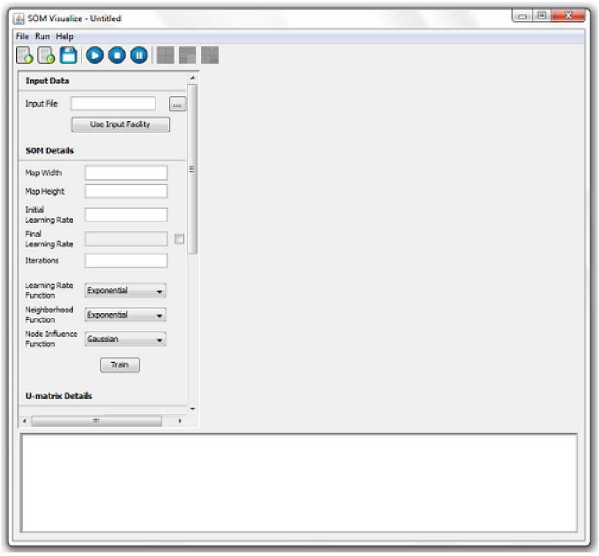

Figure 1: User Interface, SOM Visualize

the 16 regions of the Philippines, ARMM, was significantly different in terms of health status. It is shown by the black border that separates it from other nodes.

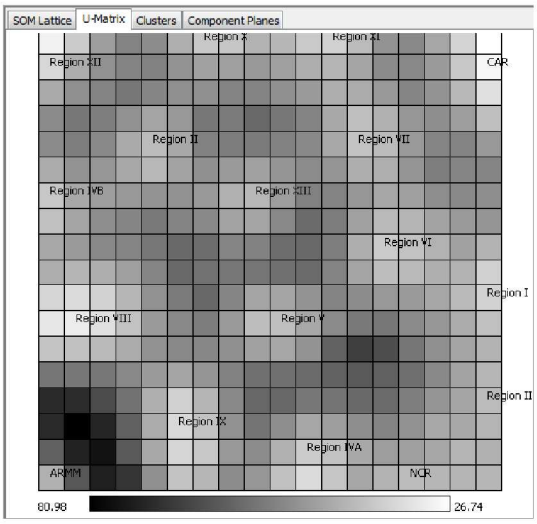

Figure 2: Similarity SOM, SOM Visualize

Then, the lattice was used as an input for the k-means clustering algorithm where k was set to 3. There were 30 different clusters. Figure 3 shows the cluster that appeared the most number of times.

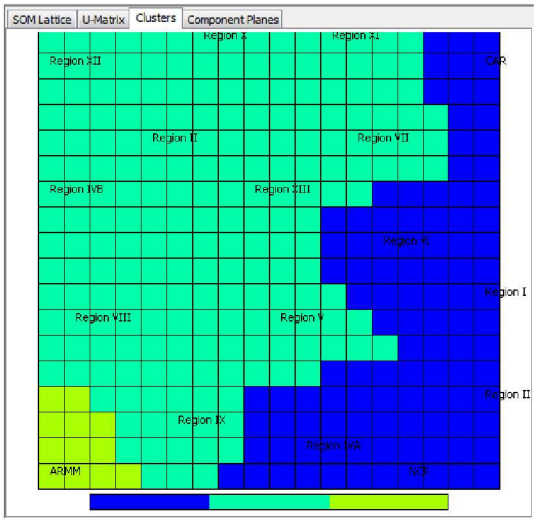

Figure 3: Clustering the SOM, SOM Visualize

Through the clusters, it was noticed that ARMM still formed its own cluster while another group of cluster was formed. NCR, Regions III, IVA, I, CAR, and VI formed their own cluster. This can be supported by the fact that these regions except for regions VI are near each other geographically.

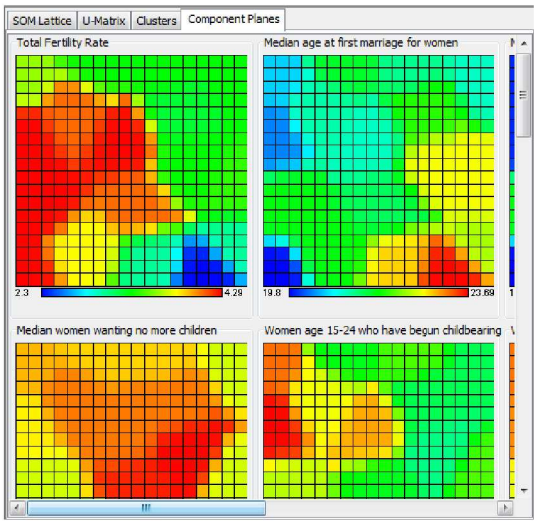

Figure 4: Component Planes, SOM Visualize

Figure 4 shows another visualization provided by the tool, the Component Planes. Throught the component planes, it was found out that ARMM exhibited the said characteristic because most of its variable values are either at the maximum or minimum, which is, most of the time, significantly different from the others.

The last figure shows the GIS visualization for the

ISBN: 978-1-61804-106-7

Figure 5: GIS Visualization, SOM Visualize

dataset. Through the GIS, it was confirmed that those clustered together were near each other geographically. Through the GIS, it was also seen that ARMM possibly exhibited those qualities because of its geographical characteristics which were tiny little islands on the southernmost part of the country.

5 Conclusion

SOM Visualize is a tool that uses the Self-Organizing Map algorithm for analysis of multi-dimensional data. Furthermore, it gives several visualization schemes such as the U-matrix, Clusters, and Component Planes to enable the discovery of subtle patterns in the dataset such as correlations, trends, and clusters. The tool is powerful considering its capabilities to handle noisy, skewed, and large datasets. It also enables viewing geographic data in a geographic map for spatial analysis and thus, we are able to gather new hypotheses that can be further tested statistically to discover new knowledge. SOM Visualize is interactive and user-friendly, making it a viable tool even for researches who are not very knowledgable about neural networks.

Through the Philippine Regional Health Data that has been processed using SOM Visualize, it can be seen that the tool is indeed capable of uncovering patterns in multidimensional data. Therefore, we may consider the tool to be a success and may further be improved to allow for a better experience in Data Visualization. In turn, the knowledge and information derived from the data set may be used in formulating hypothesis which can be further studied through statistical tests.

References:

[1] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *American Association for Artificial Intelligence - AI Magazine*, pp. 37–54, 1997.

[2] M. Friendly, "Milestones in the history of thematic cartography, statistical graphics, and data visualization.," August 2009.

[3] T. Kohonen, "The self-organizing map," in *Proceedings of the IEEE*, vol. 78, September 1990.

[4] A. Ultsch and H. P. Siemon, "Kohonen's self organizing feature maps for exploratory data analysis," in *Intern. Neural Networks*, 1990.

[5] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Transactions on Neural Networks*, vol. 11, pp. 586–600, May 2000.

[6] T. Fincke, V. Lobo, and F. Bao, "Visualizing self-organizing maps with gis," in *GI Days*, 2008.

[7] K. E. Foote and M. Lynch, "Geographic information systems as an integrating technology: Context, concepts, and definitions."

[8] T. Germano, "Course on self organizing maps," March 1999.

[9] S. Marsland, *Machine Learning: An Algorithmic Perspective*. CRC Press, 2009.

ISBN: 978-1-61804-106-7