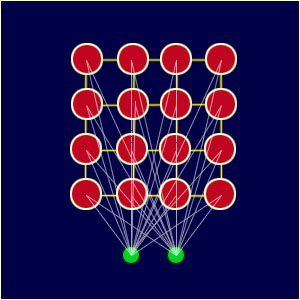
**Implementation of Kohonen Self Organizing Maps and Constrained Topological Maps**

NIKHIL LOBO

University of Wisconsin River-Falls

(14th June 2020)



**Source**:<https://medium.com/@valentinerutto/selforganizingmaps-in-english-35574f95b0ac>

**Abstract:** Kohonen Self Organizing Maps (SOMs) has applied to data reduction and density approximation. It is a type of artificial neural network. SOMs have trained using unsupervised, competitive learning to produce a low-dimensional, discretized representation of the training samples' input space called a feature map. Self-organizing maps are known for their clustering, visualization, and classification capabilities. In some examples, Kohonen's algorithm performs poorly for regression problems of even low dimensionality. A modification to the KSOM algorithm called constrained topological mapping algorithm proposed for regression applications. This paper presents both Self-organizing maps and the Constraint Topology algorithm. They are generalized to an arbitrary number of associated activities and simulation results to find out the capabilities of both the maps and the ability to generalize to new inputs that they have not trained. The results were very encouraging and confirmed the strength of both SOM and CTM to learn to associate the representations of its input space. In this paper, the dataset has generated randomly using python NumPy library.

**Keywords:** Self Organizing Map, Constrained Topological Mapping, Dimensionality reduction, CTM, KSOM, Clustering algorithm, Artificial Neural Network, Kohonen Maps, Nonparametric regression.

**INTRODUCTION**

Self-organizing maps are a powerful tool for data-mining, classification, analysis, and visualization. SOM is a type of artificial neural network which reduces the input from high dimensional space to a low dimensional representation without supervision. Unlike supervised training via error-correction (backpropagation), self-organizing maps employ

unsupervised competitive learning to map similar input vectors to nearby nodes in the map layer. Dimensionality reduction has been one of the essential concepts in data analysis. SOM algorithm represents high dimensional input data with a 1D and 2D map.

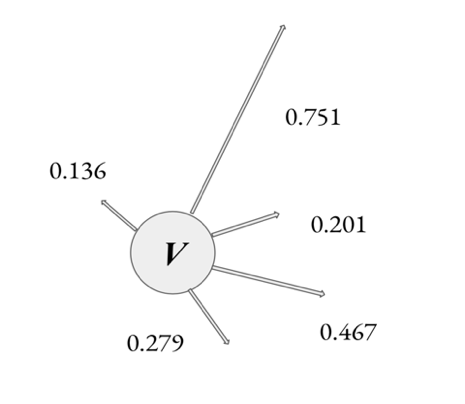


Fig 1: Representation of input vector.

N features describe input space. Each input instance from the data set has referred to as the input vector.

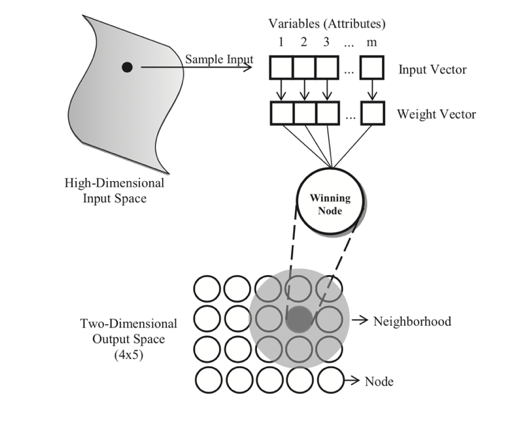


Fig 2: Graphical illustration of Self-organizing maps.

The map node weights match the input vector to find the Best Matching Unit, and the area of the lattice is selectively optimized. As the network performs unsupervised training, it does not require the knowledge of target values. The nodes in the system converge to form a cluster representing a group of entities with similar features. It has used as a classification tool for various problem domains, including speech recognition, image data compression, and image recognition.

A simple example shows that initially Kohonen's algorithm performs poorly for regression problems, since the topologically correct units in N-dimensional space may violate the natural topological ordering of projections of those units onto(N-1) dimensional subspace of independent variable. A modification of the original algorithm is called the constrained topological mapping algorithm is proposed for regression analysis applications.

**Self-organizing maps Algorithm**:

1. Each node's weights of the map have randomly initialized.
2. An input vector *Z* has chosen at random from the set of training data.
3. Every node weight compared with the input vector to calculate which one's weights are most like the input vector. The winning node is known as the Best Matching Unit (BMU).
4. The neighborhood of the BMU has calculated. The neighbors decrease over time.
5. The BMU has rewarded for becoming more likely to sample vector. The neighbors also become more like the sample vector. The nodes closer to the winning node, the more weights get altered, and the farther away the neighbor is from the BMU, the less it learns.
6. Repeat step 2 for *N* iterations.

**Constrained topological mapping Algorithm**:

1. Initialize the M dimensional topological map to the random weights in N-dimensional sample space.
2. Given an input vector *Z* in N-dimensional sample space, consider the projection sub-space of independent variables. In this projection, find the closed polygon region, which contains the input's projection.
3. In M dimensional projection subspace, find the best matching unit (BMU), among those units that form the closed region containing the projection of Z.
4. Define asymmetric neighborhood of units around the BMU and update the weights of the nodes which are inside the neighborhood region.
5. Adjust the learning factor and neighborhood function and return to step 2.

**Comparison of SOM and CTM**:

The self-organizing map is a model of mapping the input vectors onto a low dimensional discrete lattice of units. The SOM aims to generate topology-preserving mapping, where the neighborhood relations in the input space have preserved. The self-organizing mapping consists of two steps. The first step includes finding the best matching unit to the input vector Z drawn from the input space. The second step includes the forming of topological maps is to modify the weight vector of the nodes in the neighborhood. Self-organizing maps have used for various classification problems, but it performs poorly to regression problems.

The original algorithm is modified so that it can use regression analysis problems, the modified algorithm called Constrained topological mapping (CTM). The proposed algorithm performs correct topological mapping of units and, simultaneously, preserves the topological ordering of projections of these units onto (N-1) dimensional subspace of independent coordinates.

**Overview of the GUI:**

The below screenshot shows the GUI that the user will interact. It has various fields to get input from the user and read the dataset file. In the GUI, the user needs to enter the name of the dataset file, Dimension of the map wants to produce, several nodes, learning rate, epochs, and dependent value, which is used for the Constrained topologically mapping algorithm. When the user wants to train the CTM algorithm, this entry can be filled with dependent value or for SOM; it can have kept as None.

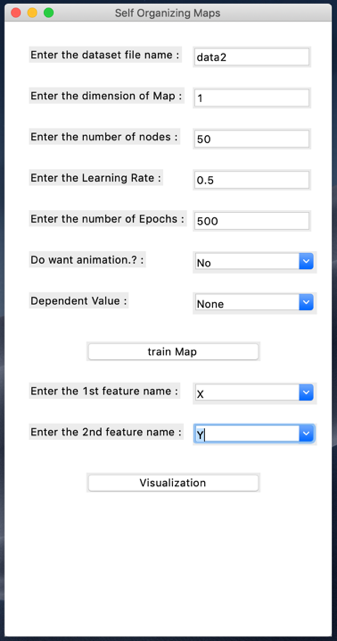
****

Fig 3: Graphical User Interface (GUI)

Once the user enters the information, later can click in the train map wherein background the input has read, and processing has done on the data given. Next is to select the features from the given dataset for visualization. The map will be plotted based on the input provided by the user.

**Experiments and Results:**

**SOM:**

**1-D Map results:**

This map has plotted by providing a numeric dataset, nodes:50 learning rate:0.3 epochs:500. By looking at

The figure 4, can see that the data has plotted in black dots in the background. The final map is plotted on data using green dots, and neighbors are connected using the red line. There is an option for animation while entering the inputs in the GUI. If the animation has selected as accurate, then the map will start animating from its random weights assigned to match the dataset. While animating the map finds the Best Matching Unit (BMU) and its neighborhood. BMU and neighborhood plotted used green dots, whereas remaining nodes displayed using red dots for better visualization. In figure 3 the map has not correctly fitted to the dataset.

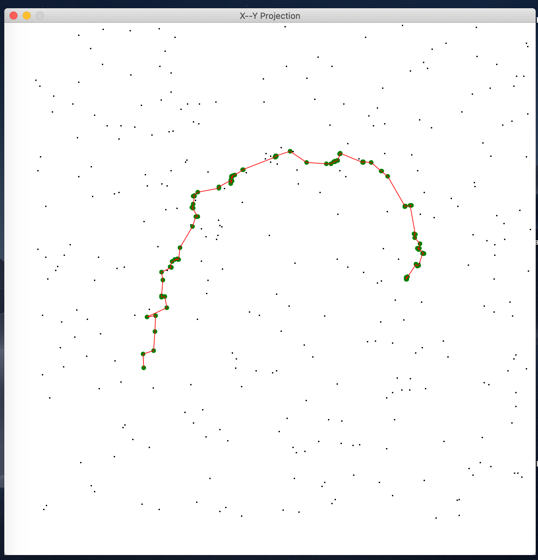
****

Fig 4: One-dimensional SOM map

In figure 4, map has plotted for the same data provided above. However, only changing a few parameters, like learning rate:0.5 and epochs:1000. By comparing the above map with this can say that this map fits perfectly to data.

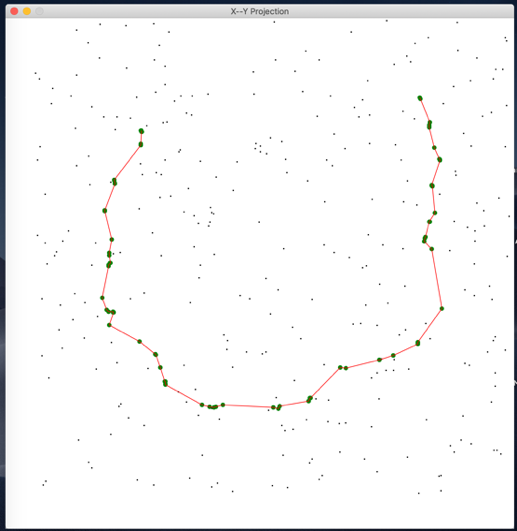
****

Fig 4: One-dimensional SOM map after tuning the parameters.

**More example for 1D map:**

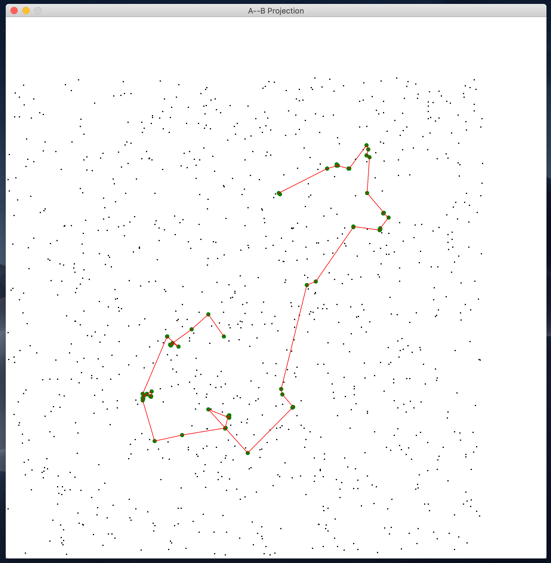
****

Fig 5: Example of 1-D map

**2-D Map results:**

The two-dimensional map is similar to the one-dimension map. The only difference is the number of nodes and the display pattern of the map. To get the below map in the GUI enter the dimension is two and the number of nodes:10. The number of nodes will form a matrix of 10\*10 nodes and map animates as a matrix to fit the data. The below map shows the final output of the SOM algorithm. Each node has connected to its neighbors. The problem with this map is that it is still not opened up entirely. The input provided to produce this map is Dimension: 2, learning rate: 0.2 and epochs: 200.

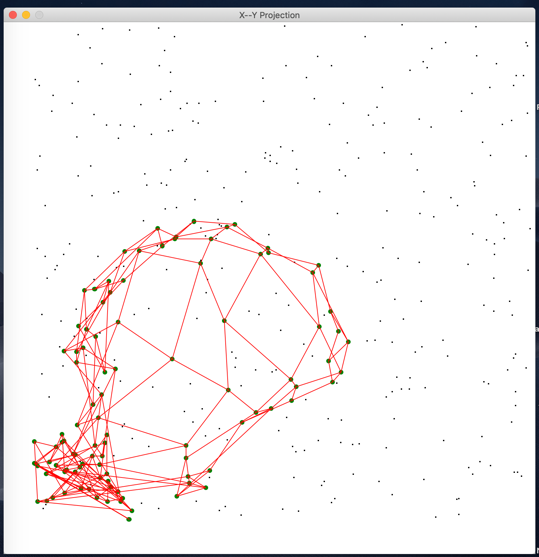


Fig 6: Two-dimensional SOM map

The map in figure 7 has also generated using the same data, which has used to produce the above map, but few parameters are changed. By looking at the map

can say the map perfectly fits the data. The parameters changed are learning rate:0.5 and epochs: 500. The conclusion is, as we tune the parameter, the map will perfectly fit the data.

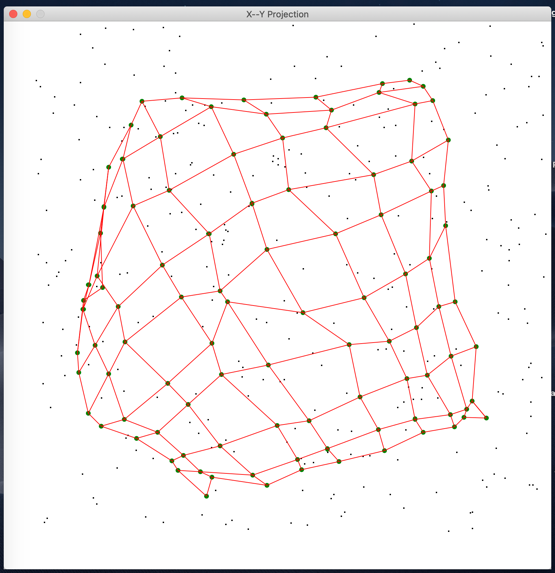


Fig 7: Two-dimensional SOM map after tuning the parameters.

**CTM:**

**1-D Map Without Dependent variable:**

This is the same as the SOM 1-D map but trained with a different dataset. Where it plots the map in the same way. Looking at it can say that it is not ideally fit for the data provided. As the SOM performs poorly on regression application and the solution has discussed using of CTM algorithm.

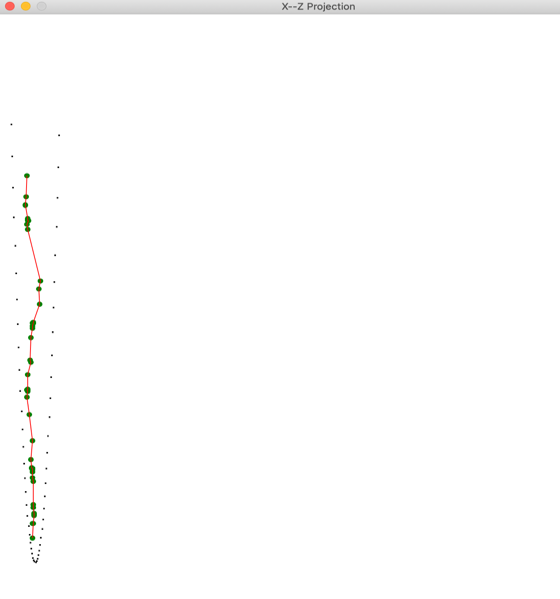


Fig 8: One-dimensional SOM map

**1D Map with the dependent variable (CTM):**

The below image shows the importance of constrained topological mapping. CTM has trained on the same data, which has used to train the previous SOM map, and the result map did not fit the data correctly. Nevertheless, when used, CTM can see how better it fits the data by providing the dependent variable as user input in the GUI.

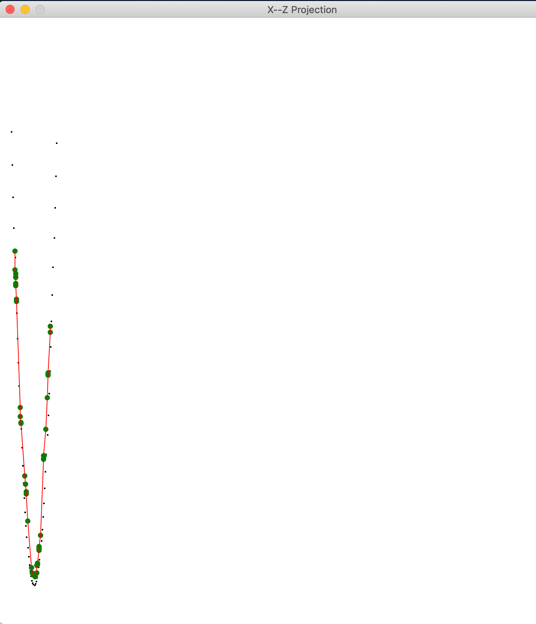


Fig 9: One-dimensional SOM map with dependent value.

**Conclusion**:

The advantages of using this algorithm over other clustering tools are flexibility in determining the number of clusters needed. The output of other techniques is only grouping and does not give the decision-maker any information about the relation between similar parts. We believe that extending the current SOM network with constrained maps will give a powerful decision-making system. As said, SOM performs poorly on regression application, but the problem has overcome using constrained topological mapping.

**References:**

[1] Melody Y. Kiang. Extending the Kohonen self-organizing map networks for clustering analysis. (2017).

[2] F Mulier. Learning Schedules for Self-Organizing Maps. Published in proceedings of the 12th IAPR Conference. (1994).

[3] Dasarathy, B.V, 1980. Nosing around the neighborhood: a new system structure and

classification rule for recognition in partially exposed environments. IEEE.

[4] Vladimir Cherkassky And Hossein Lari-Najafi. Constrained Topological Mapping for Nonparametric Regression Analysis. (1990).

[5] Eklavya, Kohonen Self-Organizing Maps. Found it on medium.com. (2019).

[6] F Murtagh & M. Hernandez-Pajares. The Kohonen self-organizing map method. (1995).

[7] Amir Ali. Self-Organizing Map with Practical Implementation. (2019).

[8] Navdeep Singh. Self-organizing Maps for Machine Learning Algorithms. Found it on medium.com (2018).

[9] Li Yuan. Implementation of Self-Organizing Maps with Python. Found in <https://digitalcommons.uri.edu/theses>. (2018).