Affine Arithmetic Self Organizing Map

Tony Bazzi
Department of Electrical and Systems Engineering
Oakland University
Rochester, MI 48309, USA
Email: tbazzi [AT] oakland.edu

Jasser Jasser
Department of Computer Science and Informatics
Oakland University
Rochester, MI 48309, USA

Mohamed Zohdy
Department of Electrical and Systems Engineering
Oakland University
Rochester, MI 48309, USA

Abstract— **This paper presents an arithmetic affine robust method to improve the performance of the self-organizing feature map which further preserves the similarities between data inputs and the weights matrix. The method presented herein targets the pre-processing and validation steps in the iterative process by filtering sensory uncertainties ensuing in data inaccuracy and large standard deviation affecting cluster affinity. The method introduces tolerances on incoming inputs to mitigate insignificant clustering creating computational burden and biasing the end result embedded in the topological map. The new technique utilizes mathematical means to modify both the competitive and adaptive stages of the conventional self-organizing map. To test the new algorithm, a simulation study was conducted to cluster Fisher's Iris dataset to improve the performance and robustness of the resulting map.**

Keywords-- Self-Organizing Feature Maps, Affine SOM, Neural Networks, Unsupervised Learning, Robust Map, Inputs and Weights Error Mitigation.

I. INTRODUCTION

Kohonen's Self-Organizing Feature Maps – Kohonen Map or SOFM - is a nonlinear projection of high dimensional data items as a quantized two dimensional image or lattice [1] [2]. SOFM have been adopted and applied in many scientific and industrial fields to help cluster large and complex data structures. The applications include clustering GPS inputs and outputs [3], detecting and removing malware [4], sound classification for hearing assistance [5], classification of mathematical curves [6], engine health diagnostics [7], intrusion detection systems [12] etc...

However, the theoretical research frontier has yet to determine the best learning mechanisms for the SOFM neural network based algorithms to mitigate learning problems such as good understanding and careful preparation in the data dissemination process [9]. The learning process is divided into six sequential iterative steps. The steps include problem definition, data acquisition, data pre-processing, data modeling, data evaluation, and knowledge deployment [9]. In this work, both the data acquisition/pre-processing and data modeling steps are targeted in order to obtain a visual topological cluster map that takes into account robustness and convergence time. This step has to be carefully handled to verify and validate that the training input samples do not contain any errors or faults propagating in the algorithm and affecting the clusters accuracy thus the output results [1]. Multiple techniques are utilized to study the effect of noise or uncertainty on the predictability of "SOFMs" using standard statistical methods such as PLS (Partial Least Squares), GFA (Genetic Function Approximation), GPLS (Genetic Partial Least Squares), and swMLR (step-wise Multiple Linear Regression) [10]. On the other hand, Zadakbar et al. [11] utilized PCA (Principle Component Analysis) to detect faults in process control systems while Bryant et al. utilized PCA to pre-process the data before running the SOFM algorithm for an engine health evaluation application [7]. Albouq et al. employed DOD (Detection of Outlier Data) and DOC (Detection of Outlier Clusters) utilizing statistical methods to detect anomalies, outliers and inconsistencies in intrusion detection system coupled to a self-organizing feature map for interconnected vehicles networks modifying the algorithm followed by a post analysis phase [8]. However, incorporating robust noise, input uncertainty mitigation or cluster outlier qualification techniques in the SOFM algorithm in itself to improve its predictability while sustaining or improving convergence time have not been demonstrated vigorously in the literature thus far. In this paper, we discuss an adjustment to the original SOFM algorithm to mitigate inaccuracies and variation in the input data. The new algorithm can also be applied to group clusters with very similar features and characteristics in order to make the SOFM robust and descriptive of the input vectors in their 2D topological projection.

II. SUMMARY OF THE SELF ORGANIZING MAP

Conventional Self Organizing Feature Map

The SOFM algorithm can be described by a competitive stage followed by an adaptive stage that is iterated through until convergence is achieved. In the competitive stage, the best matching unit –BMU- [winner node] is selected generally based on Euclidean distance, P-norm or other techniques between a randomly picked input vector and the weights

matrix. On the other hand, in the adaptive phase, the respective BMU and neighboring neurons gets updated recursively based on a stochastic mathematically defined formula, neighborhood function, and learning factor. The algorithm stages and various steps are described in the following subsections. The SOFM model proposes that a selected patch of models in the SOFM is tuned towards a given input vector in an iterative manner [1]. The pseudo-code for the conventional SOFM algorithm is presented in the following two stages adopted from what is presented for the original, stepwise recursive SOM algorithm in [1]. It must be realized that the conventional SOFM may be implemented as a standalone algorithm and/or as a neural net structure.

Competitive Stage

Let X(t) be a vector from the input data set and represents the dimensional real features that are successively computed approximations of the input model. The winner node is based on the smallest Euclidean distance to the selected input vector , in accordance with the formula in the following equation (1).

$$\begin{aligned} &\operatorname{argmin}\left\|X(t)-m_i(t)\right\| \\ &m_i \text{ is the BMU weight.} \end{aligned} \tag{1}$$

Adaptive Stage

Once the best matching unit or node is selected, the update process is activated according to the recursive formula in equation (2), where the winning unit and its neighboring nodes weights are modified. The modification process converges and is governed by the selection of the neighborhood function and the learning rate. The neighborhood function used herein is shown on Figure (1).

$$\begin{aligned} &m_i(t+1)=m_i(t)+h_{c,i}(t)\left[X(t)-m_i(t)\right] \\ &h_{c,i}=\alpha(t)e^{\frac{-\left\|r_c-r_{l,k}\right\|}{2\sigma(t)^2}} \\ &r_c = \text{ BMU Index.} \\ &\mathrm{r}_{l,k} = \text{ Neighbouring Neuron Index.} \\ &m_i \text{ and } X(t) \text{ are the weight and input vectors.} \end{aligned} \tag{2}$$

$$\alpha(t)=\eta_0 e^{\frac{-t}{\sigma(\tau)}}$$

$\alpha(t)$ is the learning rate.

$\sigma(t)$ is the variance for current iteration.

For the conventional SOFM, convergence is achieved when the final optimal values of the model take place where the learning rate attains values on the order of .01 [1].

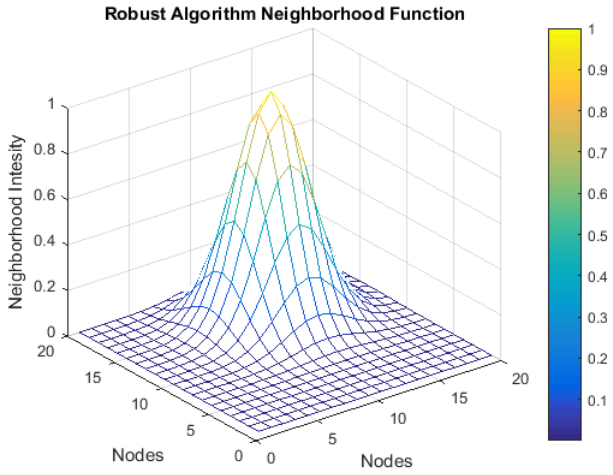

Figure 1: The neighborhood function taking on Gaussian form or similar around the best matching unit.

Affine Aithmetic Self Organizing Map

In the ASOFM (Affine Self-Organizing map) illustrated in Figure 2, the competitive stage is adjusted to mitigate inaccuracies in the input samples, eliminating outlier data, and grouping clusters with strong similarities thus increasing the robustness of the conventional SOFM.

Consider an input vector m_1 and a weight vector m_2 where α and β are radii for the circles depicted around each input and weight vectors representing the uncertainty interval in the input vector or data acquisition system and the tolerance parameter signifying cluster similarities respectively. Figure 2 illustrates the representation of both the input and weight vectors modification in the ASOFM pre-processing/validation substitute step. The radii for both circles around each weight are now parametrized and added to both and respectively as follows in equation (4) below:

$$\begin{aligned} &s_1 = m_1 + \alpha\left(2t-1\right) \\ &s_1 = m_2 + \beta\left(2t-1\right) \\ &\text{Where, } 0 \le t \le 1 \end{aligned} \tag{4}$$

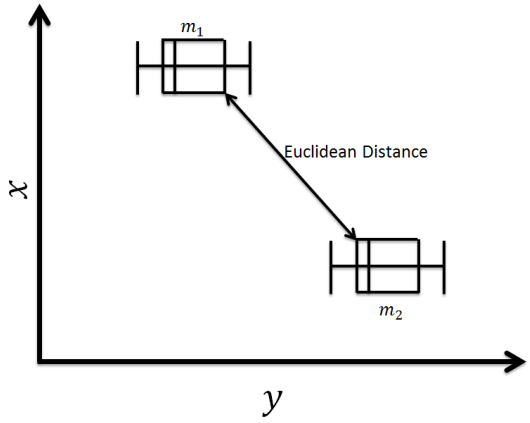

Figure 2: ASOFM 2-D sample illustration of the uncertainty in terms of a box plot around both the input m_1 and a weight vector m_2 . The box plot shows the upper/lower extremes, the upper/lower quartile and the median from the normal data around our input and weight values.

The competitive stage Euclidean distance calculation measure is thus modified as follows in equation (5) below:

$$d = \int_0^1 s_{12}{}^2 dx$$

$$d = \int_0^1 \left[\left(m_1 - m_2 \right) + \left(\alpha - \beta \right) \left(2t - 1 \right) \right]^2 dx$$

$$d = \left(m_1 - m_2 \right)^2 + \frac{1}{3} \left(\alpha - \beta \right)^2 \qquad (5)$$

$\text{eucDist} = \sqrt{d}$

α and β are radii of uncertainty.

Note that when uncertainty reduces to zero, we get back the uncorrected Euclidean distance measure.

The adaptive stage will be modified in accordance to the following equation (6).

$$m_i(t+1) = m_i(t) + h_{c,i}(t) \left[s_1(t) - m_i(t) \right] \qquad (6)$$

III. Results

To test the Affine Self Organizing Feature Map (ASOFM), the Iris Flower dataset is employed to study the performance characteristics in terms of classification accuracy utilizing the proposed algorithm.

Iris Flower Classification

Mohebi's work in [14] describes both the specifications and statistics in the Iris database as presented in Tables 1 and 2. The Fisher's Iris data set is probably the most popular database used and referenced frequently in pattern recognition to this day. It contains three classes of flowers with 50 instances each where two of the classes are linearly separable. Utilizing the Iris flower dataset will facilitate our testing of the modified algorithm to quantitatively assess its effects on classifying the different flowers based on their features. Table 1 presents the specifications of the employed dataset while Table 2 depicts a statistical analysis on the dataset.

Number of attributes	4
Number of classes	3
Missing data	No
Number of instances	150
Percent of Virginica flowers	33.34
Percent of Setosa flowers	33.33
Percent of Versicolor flowers	33.33

Table 1: Fisher's Iris Flower dataset specifications.

Attribute name	Min	Max	SD	Class correlation
Sepal length	4.3	7.9	0.83	0.7826
Sepal width	2.0	4.4	0.43	-0.4194
Petal length	1.0	6.9	1.76	0.9490
Petal width	0.1	2.5	0.76	0.9565

Table 2: Fisher's Iris data set statistics where a class correlation closer to 1.0 indicates that the two attributes are class correlated introducing ambiguity in the data instances and overlapping in the features.

Affine Arithmetic Clustering Case Study

To test ASOFM algorithm, the Iris flower data set is randomly contaminated with $\pm 8\%$ uncertainty on all features cell values. The contaminated data set is inputted to the ASOFM algorithm skipping any pre-processing steps that may be performed to filter measurement noise. The parameters used in the ASOFM algorithm are given in Table 3.

Figures 3 through 5 presents the hexagonal Euclidean distance matrix based 20 x 20 nodes topology for all α and β values possible combinations. The two dimensional maps are labeled and colored to designate clusters of different flowers based on our input data set. On the other hand Figures 6, 7 and 8 represent the conversion rate of the algorithm versus the number of iterations as a function of α and β while the accuracy numbers designate the quality of the topological map and how well it represents the input data set. Ideally, an accuracy number of 33.33% for each of the classes in the topological map illustrates that the ASOFM algorithm is able

to mitigate uncertainty or misspecified tunable parameters in the inputs or weights vectors respectively. The convergence of the algorithm to accurately represent the input data set is strictly governed by the proper choices of α and β. In our case study, the best achieved accuracy occurred with $\alpha = 0.01$ and $\beta = 0.04$ as shown in Figure 7. Results of clustering quality or flower class accuracy as function of chosen α and β values are summarized quantitatively in Table 4.

Parameter	Value
Initial learning rate	0.5
Initial sigma	15
Weights initialization	Random Gaussian
Number of iterations	500
Number of nodes	400
Input error thresholds	0.01/0.04/0.08
Weight tolerance thresholds	0.01/0.04/0.08

Table 3: ASFOM algorithm parameter table.

α	β	Setosa %	Versicolor %	Virginica %
0.08	0.08	30.5	37.5	32
0.08	0.04	30.25	39.75	30
0.04	0.08	29.25	40.75	30
0.04	0.04	29	43.25	27.75
0.01	0.08	31	45	24
0.01	0.04	34.25	32.75	33
0.08	0.01	26.75.5	44.75	28.5
0.04	0.01	25.25	41.5	33.25
0.01	0.01	29	44	27

Table 4: ASFOM algorithm classification accuracy as a function of α and β.

IV. CONCLUSION

In this paper, the new ASOFM algorithm does indeed mitigate uncertainty in the input data and allows the end user to specify allowable tolerances resulting from inherited sensory accuracy ranges and deviations in the input data as well as insignificant superfluous differences between clusters thus quality of the output results or topological map is improved. Utilizing the Iris flower machine learning sample dataset, it was demonstrated that the map had converged and was able to cluster the 3 classes while improving the accuracy with the addition of $\pm 8\%$ uncertainty on each of the input instances for every attribute. The accuracy and convergence results are listed in Figures 6, 7, 8 and Table 4 respectively. The convergence rates and demonstrated accuracies are globally dependent on the choice of both α and β values. Further work is needed to develop scientific or heuristic methods for choosing and optimizing the ASOFM tunable parameters whereas for now they are end user supplied values.

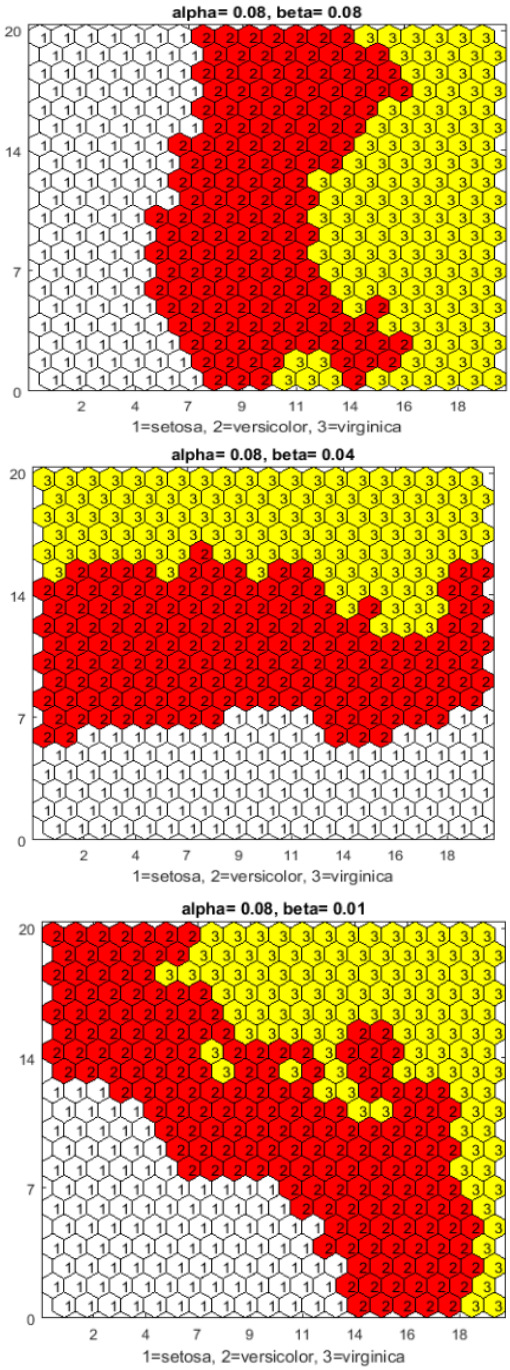

Figure 3: 2-D hexagonal topological map for respective α and β. specified in the title of the plot.

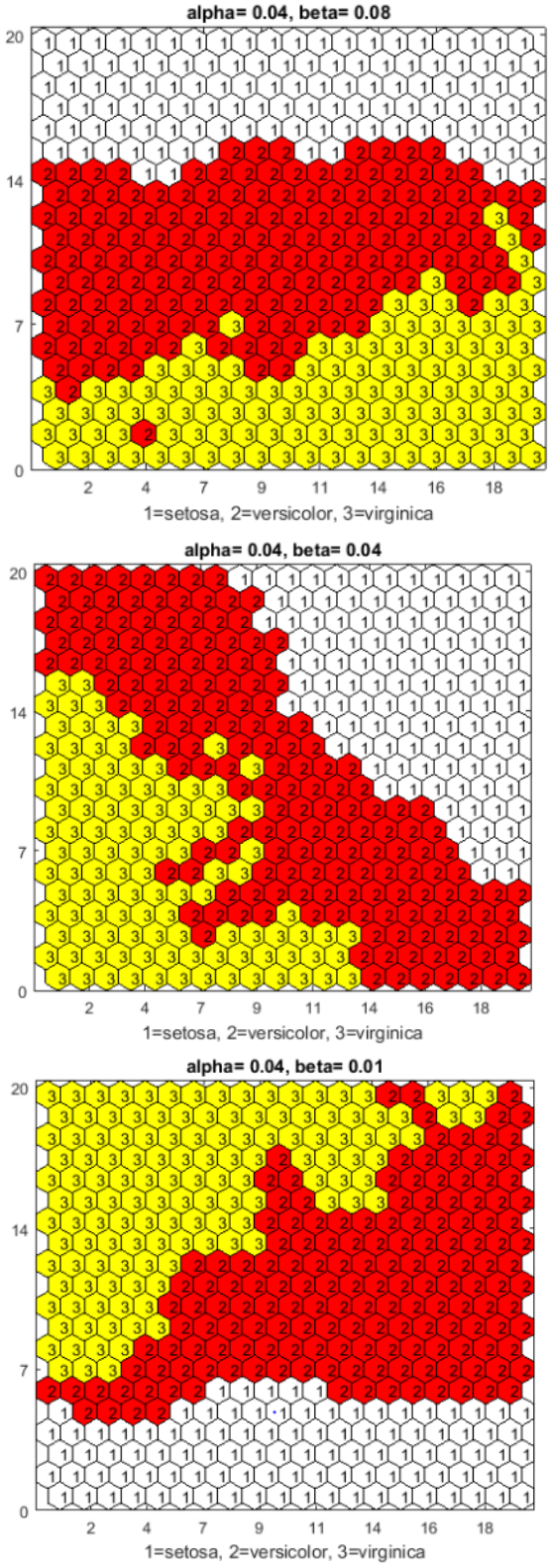

Figure 4: 2-D hexagonal topological map for respective α and β. specified in the title of the plot.

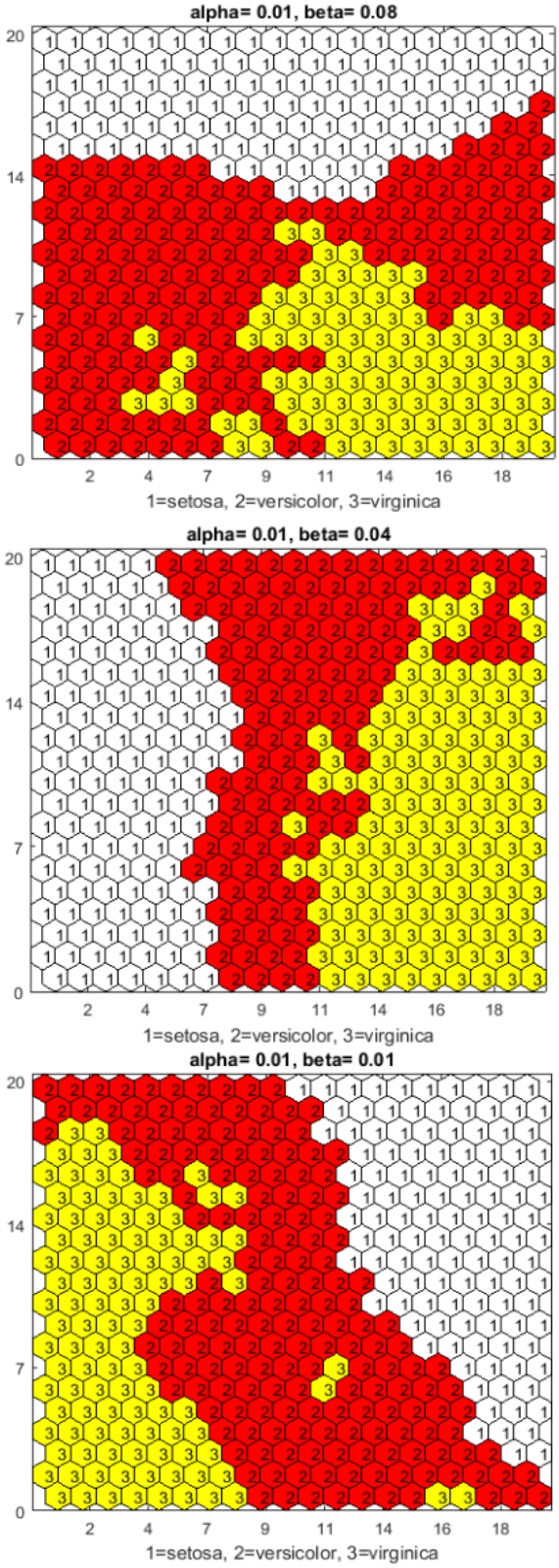

Figure 5: 2-D hexagonal topological map for respective α and β. specified in the title of the plot.

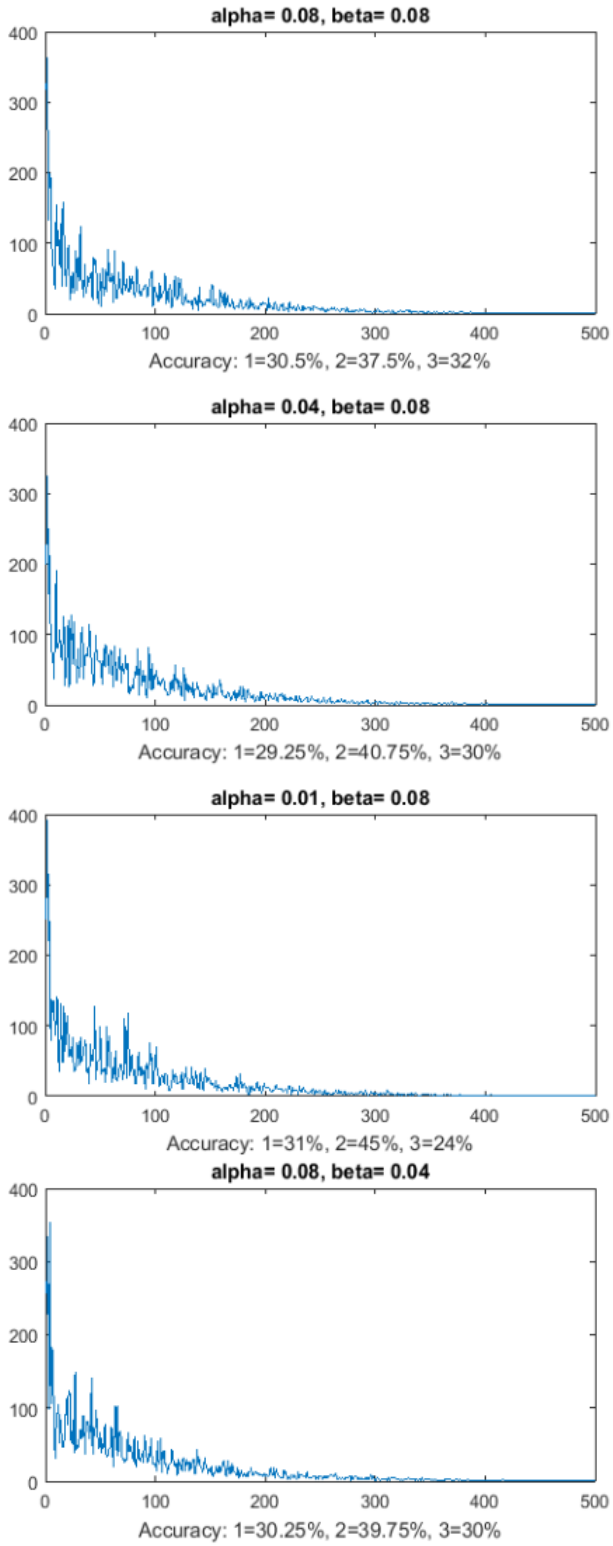

Figure 6: The x-axis for each of the plots in this Figure depicts the iteration number while the y-axis depicts the conversion rate of the self organizing map Euclidean distance matrix. The accuracy numbers represent the quality of the clustering for each of the flower classes.

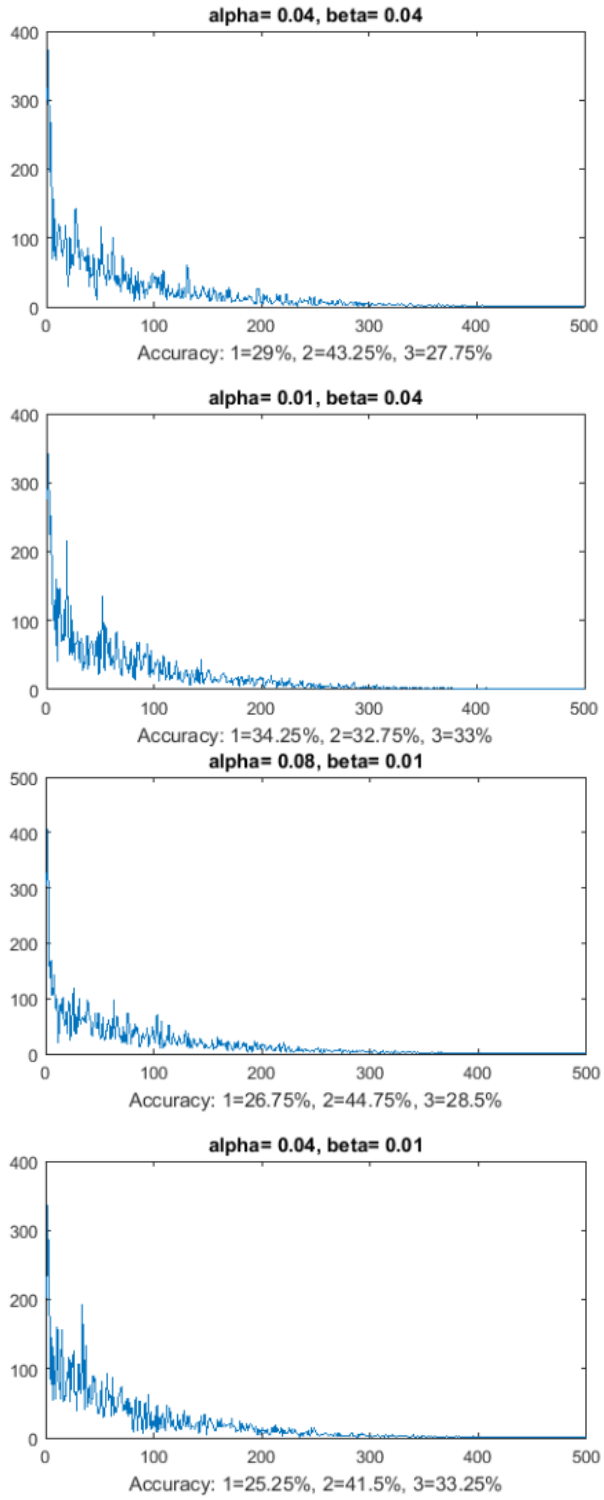

Figure 7: The x-axis for each of the plots in this Figure depicts the iteration number while the y-axis depicts the conversion rate of the self organizing map Euclidean distance matrix. The accuracy numbers represent the quality of the clustering for each of the flower classes.

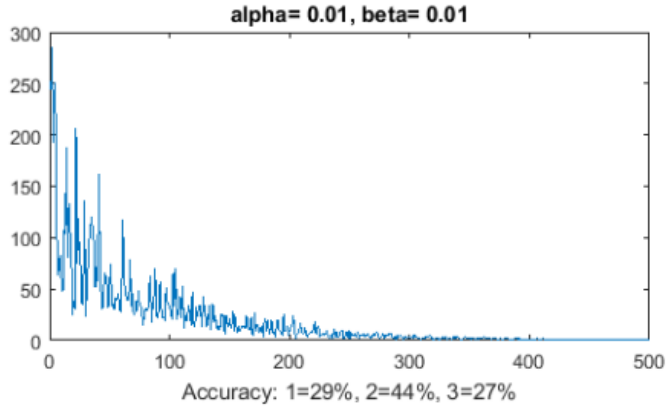

Figure 8: The x-axis for each of the plots in this Figure depicts the iteration number while the y-axis depicts the conversion rate of the self organizing map Euclidean distance matrix. The accuracy numbers represent the quality of the clustering for each of the flower classes.

V. REFERENCES

[1] Kohonen, T., MATLAB Implementations and Applications of the Self-Organizing Map. Unigrafia Oy, Helsinki, Finland, 2014, p(11-23).
[2] Kohonen, T. (1990). The Self-Organizing Map. Institute of Electrical and Electronics Engineers IEEE, Volume 78 (9), 1477.
[3] Nsour, A., Zohdy M. A. (2006), Self Organized Learning Applied to Global Positioning System (GPS) Data, Proceedings of the 6th WSEAS International Conference on Signal, Speech and Image Processing, Libson, Portigal, 203.
[4] Yang, R. Y., Kang, V., Albouq, S., Zohdy, M. A. (2015). Application of Hybrid Machine Learning to Detect and Remove Malware. Transactions on Machine Learning and Artificial Intelligence TMLAI, Volume 3 (4).
[5] Hodges, M., Zohdy, M. A. (2014). Intelligent Hearing Assistance using Self-Organizing Feature Maps. Transactions on Machine Learning and Artificial Intelligence TMLAI, Volume 2 (6), 40-52.
[6] Puengue F, L., Liu D., Zohdy M. A. (2013). Modified Self Organizing Feature Maps for Classification of Mathematical Curves. International Journal of Computer and Information Technology, Volume 2 (5).
[7] Bryant, T., Hodges, M., Zohdy, M. A. (2014). Self-Organizing Maps Applied to Engine Health Diagnostics. International Journal of Computer and Information Technology, Volume 03, issue 02, pp 205-212.
[8] Albouq, S., Zohdy M.A. (2015). Modified Self-Organizing Feature Maps for Detection Abnormal Behaviors of Connected Vehicles. International Journal of Computer and Information Technology, Volume 04, issue 05, pp 798-802.
[9] Vesanto J. Alhoniemi E. (2000). Clustering of the Self-Organizing Map. IEEE Transactions on Neural Networks, Volume 11 (3), 586-600.
[10] Xiao Y. et al., (2005). Supervised Self-Organizing Maps in Drug Discovery. Journal of Chemical Information and Modeling J. Chem. Inf. Model., 45 (6), pp 1749–1758.
[11] Zadakbar O., Imtiaz S., Khan F. (2012). Dynamic Risk Assessment and Fault Detection Using Principal Component Analysis. Ind Eng Chem Res., 52, pp. 809-816.
[12] Patole, P., A., Pachgare V., K., Kulkarni P. (2010). Self Organizing Maps to Build Intrusion Detection System. International Journal of Computer Applications, Volume 1 (8).
[13] Giraudel, J. L., Lek S. (2001). A Comparison of Self-Organizing Map Algorithm and Some Conventional Statistical Methods for Ecological Community Ordination. Elsevier Science Ecological Modeling, Volume 146, 329-339.
[14] Mohebi E., Optimized Thresholding on Self Organizing Map for Cluster Analysis, LAP LAMBERT Academic Publishing LAP, Saarbrukecken, Deutschland, 2012, p(41).