

## A Comparative Performance Analysis of Self Organizing Maps on Weight Initializations using different Strategies

H.Haripriya  
Amrita CREATE,  
Amrita Vishwa Vidyapeetham,  
Amritapuri,  
Kollam, Kerala, India  
haripriyagireesh@gmail.com

R DeviSree  
Department of Computer Science and Applications  
Amrita Vishwa Vidyapeetham,  
Amritapuri,  
Kollam, Kerala, India  
devikarai90@gmail.com

Dinesh.Pooja  
Department of Computer Science and Applications  
Amrita Vishwa Vidyapeetham,  
Amritapuri,  
Kollam, Kerala, India  
mail2poojadinesh@gmail.com

Prema Nedungadi  
Amrita CREATE,  
Amrita Vishwa Vidyapeetham,  
Amritapuri,  
Kollam, Kerala, India  
prema@amrita.edu

**Abstract**— Self Organizing Maps perform clustering of data based on unsupervised learning. It is of concern that initialization of the weight vector contributes significantly to the performance of SOM and since real world datasets being high-dimensional, the complexity of SOM tend to increase tremendously leading to increased time consumption as well. Our work focuses on the analysis of different weight initialization strategies and various dimensionality reduction measures with the intent to make SOM flexible for handling high-dimensional datasets. We use two methods of comparison, one on projected space and another before projection. The datasets used are real world datasets taken from UCI repository.

**Keywords**—Unsupervised learning, SOM, PCA, NLPCA, FCM, Weight Initialization.

### I. INTRODUCTION

Self-organizing Maps [1] organizes a way to represent higher dimensional data into much lower dimensional space. SOMs use unsupervised learning, meaning they categorize datasets without supervision. The interesting features of SOMs are its simplicity and its ease of understanding.

The weight initialization of SOMs has a major role in its performance [2], [3]. The initialization of the weight vector along with learning parameter and training inputs is what makes the network prone to learn. Considering the very aspect, we focus on initializing the weight vectors using different methods to analyze and compare SOMs performance. The different methods are Principal Component Initialization (PCI), Kernel Principal Component Initialization (KPCI), Fuzzy C-means Clustering (FCMI) along with the much used Random Initialization (RI) technique.

Principal Component Analysis (PCA)[4], [5] is the simplest form of dimensionality reduction technique and is a

linear projection method. It works well for linear datasets in reducing the number of dimensions. Kernel Principal Component Analysis (KPCA) [6]–[8] is a nonlinear extension to PCA and includes the kernel trick that allows it to work well with non-linear datasets. Another, Fuzzy C-means Clustering (FCM)[9] is based on a fuzzy approach. Also known as soft clustering, it varies from K-means in the sense that it can put an element into two clusters with each having an associated membership level.

The dimensionality reduction techniques like Principal Component Analysis (PCA) and Kernel Principal Component Analysis (KPCA) or Non-Linear Principal Component Analysis (NLPCA) are integrated with the intention to reduce the dimensions of data given to SOM. Along with the weight initializations using PCI and KPCI, the dimensionally reduced data will now be the input to SOM. Fuzzy C-means (FCMI) is also used to initialize the weight vector before dimensionality reduction and, after dimensionality reduction with dimensionally reduced data as input to SOM.

So, a comparative analysis of the SOM's performance on assigning different weight initializations is being observed in this paper with the datasets being real. The results obtained thus can be used for the detection of type of datasets provided too.

### II. RELATED WORKS

There are different clustering algorithms[10]–[12]and dimensionality reduction[13], [14] techniques existing.

#### A. Clustering Algorithms

K-means[10], [11] is one of the efficient and popular unsupervised method for clustering. Its performance is determined by initialization and appropriate distance measure. The choice of initial partition can greatly affect the resultant final clusters. Since for different initial partitions it produces

different outcomes, it would be difficult to compare the clusters formed and does not work well if the clusters in the original data are of different size and density.

Perceptron [15], [16] consists of input layer connected by paths with fixed weights to other neurons. This learning rule iteratively adjusts the weight. It reproduces all of the training input and target output pairs correctly, but the process is error correcting as long as the errors do not occur too often.

Fuzzy C-Means (FCM) [17], [18] is similar to K-Means. It is most commonly known as soft k-means algorithm. Here, each of the datapoints has got a membership probability of belonging to each clusters. The main drawbacks of FCM include the need to define the number of clusters and also need to determine membership cutoff value. Also the clusters are sensitive to initial assignment of centroids

Self-Organizing Maps (SOMs) [19], [20] represent higher dimensional data in much lower dimensions. The most interesting aspects of SOM is that it uses unsupervised learning. Advantages and disadvantages include it is very easy to understand and are simple and they work well. One of the main draw-back of SOM is getting the right data. They are computationally expensive in high dimensional domain.

### B. Dimensionality reduction methods

Dimensionality Reduction is the technique that converts data of very high dimension into data of lower dimensions. These lower dimensions convey much more information than that of higher dimensions. This is typically used in solving machine learning problems. There are both linear [21] and nonlinear [22]–[24] dimensionality reduction techniques:

Singular Value Decomposition (SVD) [25] algorithm can be used to better understand a database by presenting only the number of essential dimensions and also to simplify it, by reducing of the number of attributes that are used in a data mining process. There may be fields that are totally dependent of other fields and SVD could easily show us those fields that are not necessary for the data analysis and removes these unnecessary data.

Principal Component Analysis (PCA) [21] is a specific case of Singular Value Decomposition (SVD). PCA is a simplest form of unsupervised learning that maps the data from higher dimensionality into a lower dimensionality. It is a linear projection method for reducing the number of dimensions. Here a set of correlated variables are transferred into a new set of uncorrelated variables. Where graphical representation is difficult, PCA is efficient for analyzing data and finding patterns in it. There is a difficulty in unfolding complex data structures and only linear transformation is possible. Non-Linear PCA (NLPCA) [22]–[24] extends conventional principal component analysis (PCA) to a new feature space which is dimensionally high using the “kernel trick” and it works well for complex datasets.

Sammons Non Linear Mapping (NLM) [26] is one among the dimensionality reduction techniques that employs Non-Linear Transformations. This algorithm aims to preserve the innate structure of the input data when mapping from high

dimensional space to low dimension. As the number of input patterns increases, computational load grows to  $O(n^2)$ .

## III. DEFINITIONS AND NOTATIONS

A. *Min-Max Normalization-It transforms data inputs it to a value between 0.0 and 1.0.*

$$\text{Normalized}(e_i) = \frac{(e_i - E_{\min})}{(E_{\max} - E_{\min})} \quad (1)$$

where  $e_i$  is the column(E) values,  $E_{\min}$  the minimum value of E,  $E_{\max}$  = the maximum value of E.

B. *Euclidean Distance-It gives the distance between two points in Euclidean space.*

$$D_{ij} = \sum (W_{ij} - x_{ij})^2 \quad (2)$$

$D_{ij}$  is the distance between the  $i$  th input layer neuron to  $j$  th output layer neuron,  $W_{ij}$  is the weight associated with each  $i$  th input layer neuron and the  $j$  th output layer neuron,  $x_{ij}$  represents the  $j$  th feature of the  $i$  th input.

C. *Weight Updation*

$$W_{ij}^{(new)} = W_{ij}^{(old)} + \alpha (x_i - W_{ij}^{(old)}) \quad (3)$$

$W_{ij}^{(old)}$ ,  $W_{ij}^{(new)}$  is the weight associated with each  $i$  th input layer neuron and the  $j$  th output layer neuron before and after updation.

## IV. CLUSTERING IN SOM

SOM algorithm is based on unsupervised learning. SOMs run in two phases: Training phase where the input vectors are input to SOM along with specifying alpha rate and initializing weight vector in-order to train the network. The winner node calculated using distance function will be the corresponding category or cluster of the given input. Several iterations are conducted in this manner updating the weight and decaying the alpha value in-order to fine tune the network. Next is the clustering phase where new input vectors are quickly forwarded to their corresponding cluster once the system gets trained, easily classifying or categorizing the new data.

A. *Algorithm SOM*

*Input : { Real Dataset }, Output { Clusters }*

**Step 1:** Randomly initialize Weight vector ( $W$ ) and set learning rate ( $\alpha$ ) as 0.9.

**Step 2:** Select input vector where  $x_i$  in  $(x_1, x_2, x_3, \dots, x_n)$  that is to be given to the algorithm.

**Step 3:** Perform Min-Max normalization using equation (1) to each column.

**Step 4:** For each output cluster unit  $j$ , compute winner using Euclidean Distances from equation(2).

**Step 5:** Update the weight using equation (3) of the winner node for use by further input vectors.

**Step 6:** After completing the set of input vectors adjust the learning rate parameter ( $\alpha$ ) for the next iteration

**Step 7:** Repeat from step (2) till a pre-defined no. of iterations

*B. Principal Component Analysis and Non-Linear Principal Component Analysis*

Principal Component Analysis (PCA) is a simplest form of dimensionality reduction. It finds the mean, mean centered data, covariance matrix, corresponding eigen values and eigen vectors. The eigen vector corresponding to the highest eigen value represent the first principal component as shown in Fig.1. The dimensionally reduced data is obtained by projecting the original data onto the eigen vectors.

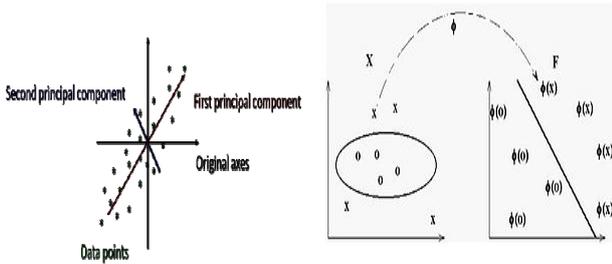


Fig. 1. Principal Components in PCA Fig.2. Mapping data points to a high dimension in KPCA

Non-Linear PCA (NLPCA) extends conventional PCA to a high dimensional feature space using the “kernel trick” overcoming the incapability of PCA to work with non-linear datasets. The data here first gets transformed to a high dimension using some pre-defined function where the data now is linearly separable. The kernel matrix helps in identifying the eigen values and eigen vectors or principal components. The transformed data after kernel trick is represented using a symbol  $\Phi$  as shown in Fig 2.

V. SYSTEM ARCHITECTURE

Our experimentations are carried out in two steps. One is by directly giving the input dataset to SOM as shown in Fig.3. The input dataset is first split into two that is 80 for training the algorithm and 20 for testing. A weight vector is initialized using either the principal components obtained after PCA/KPCA or by initializing using RI/FCMI. We have experimented also by varying  $\alpha$  value within the range 0 and 1.

The train data is normalized using Min-Max Normalization and then given to compute function which calculates the Euclidean distance between the weight vector and the input vector. The winner node will be determined and the weight of the corresponding node is updated. This is an iterative process and continues till a specified value of alpha  $\alpha$ . A final weight matrix is obtained which will then be used in the testing phase. Next, test data is normalized and the distance between the updated weight vector and the input vector is computed for

determining the winner node. The winner node will be the category of the input dataset which we call as the cluster.

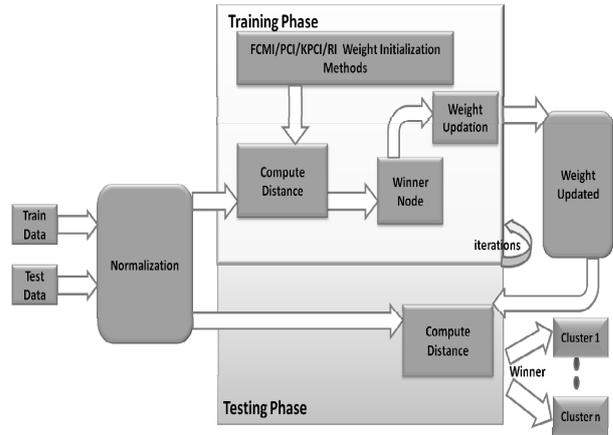


Fig.3. First Method : SOM's architecture with FCMI/PCI/KPCI/RI weight initializations

Another method is by reducing the input data dimensionally before giving it to SOM for clustering refer Fig.4. The train and test data are dimensionally reduced here using PCA or KPCA. Selection of ‘k’ principal components is an important part in dimensionality reduction. The reduced data is now set as the input to SOM, along with PCI, KPCI, RI or FCMI initializations and performance is then observed.

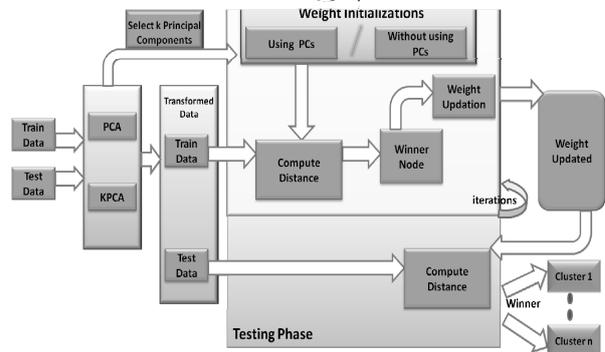


Fig.4. Second Method : SOM's architecture with weight initializations using PCs or without PCs

VI. EXPERIMENTAL EVALUATION

A. Evaluation Measure

We evaluated our results using precision and recall. Precision is the fraction of retrieved instances that are relevant. Recall is the fraction of relevant instances that are

retrieved. If the data set contains  $C$  classes [10] for a given clustering, let  $a_i$  represent the number of data objects that are correctly assigned to the class  $C_i$ , let  $b_i$  denote the data objects that are not correctly assigned to the class  $C_i$ , and let  $c_i$  denote the data objects that are incorrectly rejected from the class  $C_i$ . The performance of SOM is calculated by finding the precision and recall of each class using the equations below

$$\text{Precision, } p_i = a_i / (a_i + b_i), \quad 1 \leq i \leq C \quad (4)$$

$$\text{Recall, } r_i = a_i / (a_i + c_i), \quad 1 \leq i \leq C \quad (5)$$

The performance of SOM is calculated by finding micro-precision (micro-p) or micro-recall (micro-r). It will average over precision values.

$$\text{micro-p} = \text{micro-r} = \left( \sum_{i=1}^C a_i \right) / n \quad (6)$$

where,  $n$  is the number of data objects in dataset.

### B. Datasets

We have used real world datasets taken from The UCI (University of California – Irvine) Machine Learning Repository. The datasets include Wine recognition data set, Wisconsin prognostic breast cancer, Blood transfusion service center data set, Iris plants data and User Knowledge Modelling Data Set (Computer) dataset.

TABLE I: WEIGHT INITIALIZATIONS USING PCI, KPCI, RI, FCMI ON REAL DATASETS BEFORE DIMENSIONALITY REDUCTION

Datasets	Weight Initializations			
	PCI	KPCI	RI	FCMI
Breast Cancer	0.4398	<b>0.5</b>	<b>0.5</b>	0.404040
Blood Transfusion	<b>0.4481283</b>	0.396256684	0.080695187	0.240641
Wine	0.3971910	0.274719101	0.397752809	<b>0.410112</b>
Iris	0.336	<b>0.381333333</b>	0.338666667	0.24
Computer	0.224806	0.351550388	0.235658915	<b>0.465116</b>

In Table I, the weight initializations using PCI showed better performance for Blood dataset indicating the dataset to be linear. KPCI initializations served better for Cancer and Iris datasets, hence concluding the nature of the dataset to be non-linear. RI seems good only for cancer dataset sharing with KPCI .FCMI seems to outperform all others in this approach. It is to note that FCM is a clustering method

and it is the clusters formed from the input data that is set as weights in FCMI hence coming with better performance.

TABLE II: WEIGHT INITIALIZATIONS USING PCI, KPCI ON REAL DATASETS AFTER DIMENSIONALITY REDUCTION

Datasets	Dimensionality Reduction with Weight Initializations	
	PCA+PCI	KPCA+KPCI
Breast Cancer	<b>0.464646465</b>	0.446464646
Blood Transfusion	0.455347594	<b>0.552139037</b>
Wine	0.329213483	<b>0.370724</b>
Iris	0.257333333	<b>0.332</b>
Computer	<b>0.270542636</b>	0.245736434

Table II shows that PCA contributes good performance for cancer, computer and diabetics dataset whereas KPCA outperforms for blood, wine and iris datasets. The hint being that cancer dataset may be containing a slight linear property as well. Same is with blood dataset, which was better for PCI before, is now better for KPCI after dimensionality reduction indicating it to have non-linear property. For Iris dataset, the non-linear property still sustains.

TABLE III: RANDOM WEIGHT INITIALIZATIONS ON REAL DATASETS AFTER DIMENSIONALITY REDUCTION

Datasets	Dimensionality Reduction with Random Initializations	
	PCA+RI	KPCA+RI
Breast Cancer	0.464646465	<b>0.553535354</b>
Blood Transfusion	0.544652406	<b>0.552406417</b>
Wine	<b>0.52247191</b>	0.31011236
Iris	<b>0.442666667</b>	0.333333333
Computer	<b>0.193023256</b>	0.168992248

Table III contains the result of random weight initializations and the dimensionality reduction methods. The performance is comparatively seen well for PCA on Wine, Iris and Computer datasets. Random initialization contributes PCA for performance improvement.

TABLE IV: FCM WEIGHT INITIALIZATIONS ON REAL DATASETS AFTER DIMENSIONALITY REDUCTION

Datasets	Dimensionality Reduction with FCM weight Initializations	
	PCA+FCMI	KPCA+FCMI
Breast Cancer	<b>0.570707071</b>	0.448484848
Blood Transfusion	<b>0.67860962</b>	0.547326203
Wine	0.347752809	<b>0.351123596</b>
Iris	0.308	<b>0.334</b>
Computer	<b>0.250387597</b>	0.211627907

The clusters obtained after FCM is used for weight initializations in SOM. The results are shown in Table IV. Also, since FCM does not form non-convex clusters, PCA, the linear projection method will work efficiently on the convex clusters obtained and thus the performance in PCA.

## VII. CONCLUSION

In our paper, a comparative study of SOM's performance under different weight initialization strategies is presented. The strategies are applied to real datasets taken from UCI repository. The initialization strategies are applied to datasets after dimensionality reduction and before dimensionality reduction as well. By looking whether PCA or KPCA outperformed for a particular dataset we can identify whether the input dataset is linear or non-linear. It is also visible that even after reducing dimensionality the datasets sustained the linear or non-linear properties of the dataset. Our comparative analysis presented can serve as a reference for future researchers of SOM.

## ACKNOWLEDGMENT

We would like to express our fullest gratitude to the Chancellor of Amrita University, Sri Mata Amritanandamayi Devi, who is our guiding force and inspiration. We also express our gratitude to Dr. M.R Kaimal, Chairman, Department of Computer Science and Engineering and to the faculties of Computer Science and Application Department for their constant support and guidance. Also we are grateful to Deity for the support of eGAP project .

## REFERENCES

[1] T. Kohonen, "The Self-Organizing Map," *Proceeding of the IEEE*, vol. 78, pp. 1464–1480, 1990.

[2] V. Aggarwal, A. K. Ahlawat, and B. N. Pandey, "A Weight Initialization Approach for Training Self Organising Maps for Clustering Applications," pp. 1000–1005, 2012.

[3] J. Y. F. Yam and T. W. S. Chow, "A weight initialization method for improving training speed in feedforward neural network," vol. 30, pp. 219–232, 2000.

[4] L. I. Smith, "A tutorial on Principal Components Analysis Introduction," 2002.

[5] J. Shlens, "A TUTORIAL ON PRINCIPAL COMPONENT ANALYSIS Derivation , Discussion and Singular Value Decomposition," 2003.

[6] A. A. Nielsen and M. J. Canty, "Kernel principal component analysis for change detection," vol. 7109, no. 0, pp. 1–10, 2008.

[7] B. Scholkopf, A. Smola, K. R. Muller, and M. Kybernetik, "Kernel Principal Component Analysis," pp. 2–7.

[8] Y. Chen, J. Hu, Y. Zhang, and X. Li, "Traffic Data Analysis Using Kernel PCA and Self-Organizing Map," *2006 IEEE Intell. Veh. Symp.*, pp. 472–477, 2006.

[9] M.-C. Hung and D.-L. Yang, "An Efficient Fuzzy C-Means Clustering Algorithm," 2001.

[10] A. Ahmad and L. Dey, "A k -mean clustering algorithm for mixed numeric and categorical data," vol. 63, pp. 503–527, 2007.

[11] D. Pham, S. Dimov, and C. Nguyen, "Selection of K in K -means clustering," vol. 219, no. May 2004, pp. 103–119, 2005.

[12] P. Mangiameli, S. K.Chen, and D. West, "A comparison of SOM neural network and hierarchical clustering methods," vol. 2217, no. 96, 1996.

[13] A. Ghodsi, "Dimensionality Reduction A Short Tutorial," 2006.

[14] Y. Qu, G. Ostrouchov, and A. Geist, "Principal Component Analysis for Dimension Reduction in Massive Distributed Data Sets \* Notation and Background for Principal Components Anal-," 2001.

[15] M. Collins and F. Park, "Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms."

[16] G. Thimm and E. Fiesler, "High Order and Multilayer Perceptron Initialization," 1994.

[17] S. Nascimento, B. Mirkin, and F. Moura-Pires, "A fuzzy clustering model of data and fuzzy c-means," *Ninth IEEE Int. Conf. Fuzzy Syst. FUZZ- IEEE 2000 (Cat. No.00CH37063)*, vol. 1, 2000.

[18] S. A. Mingoti and J. O. Lima, "Comparing SOM neural network with Fuzzy c -means , K -means and traditional hierarchical clustering algorithms," vol. 174, pp. 1742–1759, 2006.

[19] S. M. Guthikonda, "Kohonen Self-Organizing Maps," no. December, 2005.

[20] T. Raiko and E. Oja, "Extending Self-Organizing Maps with Uncertainty Information of Probabilistic PCA."

[21] A. A. Farag, "A Tutorial on Principal Component Analysis Theoretical Discussionz Application to Face Detection and Recognition," no. September, 2009.

[22] B. Scholkopf, A. Smola, and K. R. Muller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," no. 44, 1996.

[23] M. Scholz, "Validation of Nonlinear PCA," no. March, pp. 21–30, 2012.

[24] M. Linting, J. J. Meulman, P. J. F. Groenen, and A. J. Van Der Kooij, "Nonlinear Principal Components Analysis: Introduction and Application often confronted with a large number of variables that they," vol. 12, no. 3, pp. 336–358, 2007.

[25] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems Dimensionality Reduction for."

[26] J. W. Sammon, "Nonlinear Mapping Structure Analysis," vol. C, no. 5, 1969.

[27] J. A. Lee, A. Lendasse, and M. Verleysen, "Curvilinear Distance Analysis versus Isomap," no. April, pp. 185–192, 2002.