Journal of Theoretical and Applied Information Technology
20th April 2013. Vol. 50 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SELF-ORGANIZING MAP-BASED K-MEANS CLUSTERING FOR STABILITY ANALYSIS OF PRODUCT QUALITY IN PACKAGING IN SEMICONDUCTOR MANUFACTURING

[1]JIN DING, [2]YONG-ZAI LU, FELLOW, IEEE, [3]JIAN CHU

1Ph. D., 2Professor, 3Professor

State Key Laboratory of Industrial Control Technology, Institute of Cyber-System and Control,

Zhejiang University, Hangzhou, 310027, China

E-mail: [1]jding@zju.edu.cn, [2]yongzailu@yahoo.com, [3]chj@csc.zju.edu.cn

ABSTRACT

The "packaging" plays an important role in semiconductor manufacturing. Among the multiple test-passed steps in packaging production, the step of "wire-bonding" is the most complicated and critical one since there are many tuning parameters, such as force, current and time, etc. needing to be set-up by operators in order to perform good bonding. Several key perform indexes, for example, wire pull, ball shear and ball height, are used to measure the product quality. This paper presents the development of applying SOM (self-organizing map) based k-means to the data-based classification for product quality. Namely, the product quality can be divided into a number of categories in terms of huge historical quality related data. In addition, SOM based K-mean clustering model, algorithms and the case studies with real-production data are addressed. The concluding remarks and future work are given in the final section.

Keywords: *Stability Analysis; Clustering; Self-Organizing Map; K-Means; Grid-Based Heuristic Initialization; Wire Bonding; Packaging*

1. INTRODUCTION

In packaging in semiconductor manufacturing, the test-passed silicon wafers go through slicing, loading, wire-bonding, molding, curing, electroplating, printing and testing, and finally become packages. Among these eight steps, production in wire-bonding is the most complicated. In the bonding step, using the thermosonic wire-bonding technique, metal wire, usually Au wire, connects the chip I / O terminals with corresponding package pins or weld zone on the substrate. The joint heat, force, and ultrasonic energy can remove the surface oxides and contamination, create the plastic deformation, make the interfaces intimately contacted by electron sharing and atomic diffusion, and form the solder joint [1]. In this paper, we perform stability analysis of product quality in wire-bonding step. Several quality indexes, for example, wire pull, ball shear and ball height, are commonly used to describe the product quality in wire bonding. Low fluctuations in these indexes mean good stability. With the product quality data from wire bonding, we first cluster these data into several product categories, and then perform stability analysis on each product category.

The most well known method for clustering is K-means, developed by Mac Queen in 1967 [2], which separates samples into k disjoint groups, termed clusters, aiming to minimize the sum of distance between the sample and the centroid of the cluster to which this sample belongs. Due to the simplicity of k-means, its applications range from CRM [3], to color image segmentation [4] to document clustering [5]. If the number of cluster is unknown, k-means method can be repeated for a set of different number of clusters, typically from two to $\sqrt{N}$ [6], where N is the number of samples in the data set. To select the best one among these different partitionings, each of these can be evaluated using some kind of validity index [7], [8], [9]. Under this scenario, we can roughly estimate the computational complexity of K-means on N samples is proportional to $\sum_{k=2}^{\sqrt{N}} Nk$. When operating on a large-scale data set, such as in wire bonding, each wire bonder yields thousands of products per day, time consumed is considerable.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Self-Organizing Map (SOM) is an unsupervised neural network developed by Kohonen [10] for visual cluster analysis. The neurons of the map are located on a regular grid in a low (usually 2) dimensional space, and associated with the cluster prototypes by the linked weight vectors. In the course of learning process, the neurons compete with each other through the best-matching principle in such way that the input is projected to the nearest neuron under a given distance metric. The weight vectors of winning neuron and its neighbors are then adjusted towards the input in proportion with the neighborhood distance, and consequently the neighboring neurons likely represent the similar patterns of the input space. Due to the data clustering and spatialization through the topology preserving projection, SOM is widely used in the context of visual clustering applications [11], [12], [13]. In this paper, we combine SOM with k-means to cluster product quality data in wire bonding [14], [15], [16]. SOM-based k-means includes two phase. At the first phase, SOM is used to preprocess the data samples. After SOM training, the data samples are mapped into a two-dimensional grid, and every neuron in the grid represents part of data samples which are closer to the weight vector associated with this neuron than others; at the second phase, k-means performs on the result of SOM. It is clear that SOM-based k-means is more computationally efficient than the traditional k-means which operates on the whole data samples. The total complexity of SOM-based k-means is proportional to $NM + \sum_k Mk$. Assuming the number of neurons $M = 5\sqrt{N}$, the number of the data samples N=10000, and without many practical considerations, we can roughly estimate the reduction of computational load is six-fold, compared to the traditional k-means [6]. In addition, at the second phase, we propose a grid-based heuristic method to initialize the k cluster centroids [17], [18], [19]. From the training process of SOM, we know each neuron in the grid represents some similar data samples, and the similarity between data samples represented by two neurons is proportional to the distance between these two neurons. This prompts us that if we make the selected k neurons as evenly scatter across the grid as possible, in general, the weight vectors (centroids) associated with these k neurons will fit the distribution of data samples well. In return this will accelerate convergence while maintaining a good solution quality.

Simulation results on real product quality (wire pull and ball shear) data in wire bonding from semiconductor packaging and testing vendor, show the quality stability is differentiated over product categories on a wire bonder. Based on this, for to-be-processed products, operators can choose the suitable wire bonder to run, which will reduce the product quality fluctuation at large and improve relationship with customers.

The rest of this paper is organized as follows. In the section 2, we first briefly review the k-means method and SOM method. Then we introduce the SOM-based k-means method and the novel grid-based heuristic initialization method for k cluster centroids; Section 3 gives the simulation results on the real industrial quality data from wire bonding; and finally, in section 4, concluding remarks are made.

2. SELF-ORGANIZING MAP-BASED K-MEANS CLUSTERING MODEL

2.1. K-means

The *k*-means algorithm partitions a given set of data in a manner such that the squared-error function is minimized for a pre-specified number of clusters. The squared error function (*E*) is defined as:

$$E = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \| \mathbf{x} - \mathbf{m}_i \|^2 \quad (1)$$

Where k is number of specified clusters, the d-dimensional $\mathbf{m}_i$ denotes the center of ith cluster, and $\mathbf{x}$ represents a d-dimensional data vector belonging to the cluster C_i. The computational steps of *k*-means algorithm that aim to minimize the sum of squared distances between all points and the cluster centers are described below:

Step1: Randomly choose k number of initial cluster centers $\mathbf{m}_i$, i=1, 2, ..., k.

Step2: Assign a point $\mathbf{x}_j$, j=1, 2, ..., N, to the cluster q, if

$$\| \mathbf{x}_j - \mathbf{m}_q \| \leq \| \mathbf{x}_j - \mathbf{m}_i \|, \quad i = 1, 2, ..., k \quad (2)$$

Step3: Compute new cluster centers as follows,

$$\mathbf{m}_i^{new} = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x} \quad i = 1, 2, ..., k \quad (3)$$

Where $|C_i|$ is the number of data points assigned to the cluster C_i.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step4: If the iteration number reaches the maximum number of iterations or the distances between the new cluster centers and the previous cluster centers are less than pre-specified value, then terminate; otherwise, go to Step2.

If the number of cluster is unknown, the *k*-means can be repeated for a set of different number of clusters, typically from two to $\sqrt{N}$. To select the best one among different partitionings, each of these can be evaluated using some kind of validity index. Here in our application, we use the Davies-Bouldin (DB for short) index [9],

$$\frac{1}{Q}\sum_{i=1}^{Q}\max_{l\neq i}\left\{\frac{\frac{1}{|C_i|}\sum_{\mathbf{x}\in C_i}\|\mathbf{x}-\mathbf{m}_i\|+\frac{1}{|C_l|}\sum_{\mathbf{x}\in C_l}\|\mathbf{x}-\mathbf{m}_l\|}{\|\mathbf{m}_i-\mathbf{m}_l\|}\right\} \quad (4)$$

Where Q is the number of clusters. From the definition of this index, we can see, as a function of the ratio of the within-cluster scatter to the between-cluster separation, a lower value will mean that the clustering is better.

2.2. Self-Organizing Map

The principal goal of the self-organizing map (SOM) algorithm developed by Kohonen [10] is to transform an incoming signal pattern of arbitrary dimension into a one- or two-dimensional discrete map, and to perform this transformation adaptively in a topological ordered fashion. Many activation patterns are presented to the network, one at a time. Typically, each input presentation consists simply of a localized region of activity against a quiet background. Each such presentation causes a corresponding localized group of neurons in the output layer of the network to be active.

The essential ingredients of the neural network embodied in such an algorithm are as follows [20]:

1. A one- or two-dimensional lattice of neurons that computes simple discriminant functions of inputs received from an input of arbitrary dimension;
2. A mechanism that compares these discriminant functions and selects the neuron with the largest discriminant function value (also called "best matching unit", BMU);
3. An interactive network that activates the selected neuron and its neighbors simultaneously;
4. An adaptive process that enables the activated neurons to increase their discriminant function values in relation to the input signals.

The SOM comprises an array of units (also known as "nodes" or "neurons") arranged in the form of a grid. A *d*-dimensional weight vector (prototype) vector is associated with each node in the grid, where *d* refers to the dimensionality of an input data pattern. Let $\mathbf{x}_i$, i=1, 2,..., N, be the *d*-dimensional vectors to be clustered and $\mathbf{w}_j$ be the *d*-dimensional weight vector associated with the node j, j=1, 2,..., M. The stepwise procedure for training the SOM network is as given below.

Step1 (*Initialization*): Choose small random values for the initial weights, $\mathbf{w}_j(0)$, and fix the initial learning rate (α_0) and the neighborhood.

Step2 (*Determining the BMU*): Select a sample pattern, $\mathbf{x}$ (t), from the data set and determine the BMU ($i(\mathbf{x})$) at training iteration t, using the minimum Euclidean distance criterion.

$$i(\mathbf{x}) = \arg\min_j \|\mathbf{x}(t)-\mathbf{w}_j\|, \qquad j=1,2,...,M \quad (5)$$

Where ||·|| refers to the Euclidean norm.

Step3 (*Weight updating*): Updating all the weights according to the Kohonen learning rule,

$$\mathbf{w}_j(t+1)=\begin{cases}\mathbf{w}_j(t)+\alpha(t)[\mathbf{x}(t)-\mathbf{w}_j(t)], & j\in N_{i(\mathbf{x})}(t)\\ \mathbf{w}_j(t), & otherwise\end{cases} \quad (6)$$

Where t denotes iteration index, $N_{i(\mathbf{x})}$ (t) is the neighborhood of $i(\mathbf{x})$, the BMU at iteration t, and α (t) =α_0/ (1+t) is the learning rate.

Step4: Increment the iteration index, t, by unity and decrease the magnitude of the learning rate, $\alpha(t)$, accordingly; shrink the neighborhood, $N_{i(\mathbf{x})}(t)$ of the BMU.

Step5: Repeat steps 2-4 until the change in the weight magnitudes is less than the specified threshold or the maximum number of iterations has been reached.

2.3. SOM-Based K-Means

2.3.1. Framework of som-based k-means

SOM-based k-means includes two phase. At the first phase, SOM is used to preprocess the data samples. After SOM training, the data samples are mapped into a two-dimensional grid, and every neuron in the grid represents part of data samples which are closer to the weight vector associated with this neuron than others; at the second phase, k-means performs on the result of SOM.

The workflow of SOM-based k-means method is depicted as follows, using pseudo code. Dataset

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

represents d-dimensional data samples and the number of data samples is N.

SOM_BASED_KMEANS (*Dataset*)

Define width as the number of neurons in the horizontal edge of the grid

Length as the number of neurons in the vertical edge of the grid

M as the number of neurons

w as weight vectors of neurons

dist as distance matrix, denoting distance between any two neurons

alpha as learning rate

$Neighbor_i$ as neurons in the neighborhood of the i_{th} neuron

m as cluster centroids

DB_INDEX as Davies-Bouldin index for partitionings

C as cluster index to which data points belong

1. SOM_TRAINING(**w**, **dist**, *width*, *length*, *alpha*, *Neighbor*);
2. *M* ← *width*length*
3. **for** *i*←2 **to** sqrt (*M*)
4. **do m** ← HEURISTIC_INITIALIZATION_KCENTROIDS(*width*, *length*, *i*, **dist**, **w**);
5. KMEANS(*i*, **m**, **w**);
6. *m_trials*[*i*] ← **m**;
7. compute *DB_INDEX*[*i*];
8. **for** *i*←2 **to** sqrt(*M*)
9. **do if** DB_INDEX[i] < *min* **then**
10. **do** *min* ← *DB_INDEX*[*i*];
11. *best_number* ← *i*;
12. **m** ← *m_trials*[*best_number*];
13. **for** *i*←1 **to** *N*
14. **do for** *j*←1 **to** *best_number*
15. **do if** distance (*Dataset*[*i*], **m**[*j*]) < *min***then**
16. **do** *min* ← distance(*Dataset*[*i*], **m**[*j*]);
17. **C**[*i*] ← *j*;
18. **return C**;

Subroutines like SOM_TRAINING and KMEANS can be called from the common packages, and HEURISTIC_INITIALIZATION_KCENTROIDS is defined in the below section.

2.3.2. Grid-based heuristic initialization of k centroids

In the traditional k-means method, with few knowledge on the distribution of data samples, random initialization for k cluster centroids is adopted. While in the SOM-based k-means method, after SOM training, the produced two-dimensional grid sheds lights on the distribution of data samples [10]. Based on this, we propose a grid-based heuristic initialization method, making the initialized k cluster centroids fit the distribution of data samples well. In general, k-means with the initialized k cluster centroids which are fitting the distribution of input space well has faster convergence than those with randomly initialized k cluster centroids. Fig.1 uses a simple one-dimensional data clustering to depict the effects of the initial cluster centroids on the convergence of k-means. In Fig.1(a), if two initial centroids made by random strategy huddle, three iterations are needed to converge; whereas in Fig.1(b), if two initial centroids made by some heuristic strategy fit the distribution of the data points well, one iteration is needed to converge.

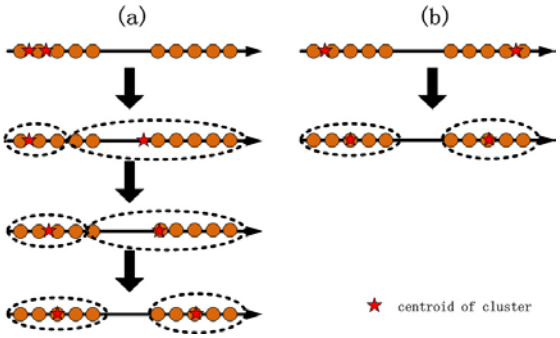


Figure.1 Simple Clustering Example Of One-dimensional Data Points. (a), If Two Initial Centroids Made By Random Strategy Huddle, Three Iterations Are Needed To Converge; (b) If Two Initial Centroids Made By Some Heuristic Strategy Fit The Distribution Of The Data Points Well, One Iteration Is Needed To Converge.

According to the SOM training process, after training, each neuron in the grid represents part of the data samples which are closer to the weight vector associated with this neuron than others. And, if two neurons are relatively close on the grid, then the data samples represented by these two neurons

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

in the input space are relatively close as well, measured in the Euclidean space. Generally speaking, the similarity between data samples represented by two neurons is proportional to the distance between these two neurons. This hints us that if the selected k neurons evenly scatter across the grid, the weight vectors (centroids) associated with these k neurons will fit the distribution of data samples well.

Taking advantage of this, we propose a grid-based heuristic initialization method for k cluster centroids. First, select a neuron, usually the first neuron of the grid, as the first cluster centroid. Second, randomly select a neuron as the second cluster centroid from candidates which distance the first neuron with d. Third, randomly select a neuron as the third cluster centroid from candidates which distance the second neuron with d. Continue this process until k cluster centroids has been selected. d is the function of the size of grid and the number of the clusters. Fig.2 elucidates the selection process of k cluster centroids. Assume the topology of the grid is 10x10, k=5 and d=2.

The workflow of grid-based heuristic initialization method is depicted using pseudo code as follows. width denotes the number of neurons in the horizontal edge of the grid; length denotes the number of neurons in the vertical edge of the grid; k is the number of the clusters; dist is a distance matrix, denoting distance between any two neurons; w denotes the weight vectors of neurons.

HEURISTIC_INITIALIZATION_KCENTROIDS(*width*, *length*, *k*, **dist**, **w**)

define **m** as cluster centroids

1. $d \leftarrow$ round(max(*width*, *length*)/*k*);
2. **m**[1] ← **w**[1];
3. $M \leftarrow$ *width***length*;
4. *index* ← 1;
5. **for** *i* ← 2 **to** *k*
6. **do for** *j* ← 1 **to** *M*
7. **if dist**(*index*, *j*) == *d* **then**
8. **do** *candidates* ← *j*
9. *index* ← random(*candidates*);
10. **m**[*i*] = **w**[*index*];
11. return **m**;

The simulation results in the following section show this heuristic initialization method can accelerate the convergence of k-means while maintaining the quality of the solutions.

3. CASE STUDY

We apply the SOM-based k-means method to the product quality data in wire bonding from semiconductor packaging and testing vendor, and then perform the stability analysis on quality data on each product category. Here, we have 10,000 product quality records on one wire bonder, and data attributes include wire pull (WP for short) and ball shear (BS for short). The program is implemented using Matlab 2008b and simulations are executed on Intel 2.10GHz computer. The number of neurons of the grid is set to 100 and the topology of the grid is 10x10.

Fig.3(a) presents the distribution of the initialized weight vectors over the product quality data; Fig.3(b) shows after training, the distribution of the weight vectors over the product quality data. It's clear that after training, the weight vectors fit the distribution of the whole product quality data quite well.

The weight vectors are treated as the representatives of the product quality data, on which k-means performs. k ranges from 2 to 9. In Table 1 and 2, we compare the heuristic initialization (HI for short) method and the random initialization (RI for short) method on both the number of iterations for convergence and within-cluster-squared-sum (WCSS for short). Both HI and RI run 10 times, respectively. From the listed results, we can see, the HI has smaller number of iterations for convergence for all ks. Meanwhile, the WCSS on HI is competitive with RI.

Table 1: Comparison Of Number Of Iterations For Convergence Between HI And RI

Method	Best	Average	Worst	Standard Deviation
		k=2		
HI	4	5.7	**7**	**1.50**
RI	**3**	**5.6**	8	1.65
		k=3		
HI	**2**	**4.50**	**8**	**1.90**
RI	4	9.80	14	3.16
		k=4		
HI	**5**	**6.70**	**12**	**2.21**
RI	9	15.20	22	4.59
		k=5		
HI	4	**8**	**13**	**3.50**
RI	4	10.10	20	4.72
		k=6		

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HI	**5**	**7.10**	**11**	**2.08**
RI	7	13.20	19	4.42
		k=7		
HI	**3**	**8.90**	**16**	4.23
RI	7	12.20	19	3.58
		k=8		
HI	**6**	**9.90**	**16**	3.38
RI	8	12.10	18	3.11
		k=9		
HI	**4**	**9.30**	**15**	4.00
RI	8	10.60	16	2.27

Table 2: Comparison Of WCSS Between HI And RI

Method	Best	Average	Worst	Standard Deviation
		k=2		
HI	7.80	7.80	7.80	0
RI	7.80	7.80	7.80	0
		k =3		
HI	4.14	**4.14**	**4.14**	**0**
RI	4.14	5.46	5.78	0.69
		k =4		
HI	**2.98**	**3.00**	3.04	0.03
RI	3.04	3.04	3.04	0
		k =5		
HI	1.94	2.36	3.45	0.57
RI	1.94	**2.30**	**2.67**	**0.38**
		k =6		
HI	1.84	2.29	2.75	0.48
RI	**1.58**	**1.89**	**2.70**	**0.45**
		k =7		
HI	1.50	1.93	2.73	0.52
RI	**1.28**	**1.52**	**2.18**	**0.36**
		k=8		
HI	1.35	1.48	**1.69**	**0.16**
RI	**1.13**	**1.31**	2.10	0.29
		k=9		
HI	1.00	1.36	2.55	0.44
RI	**0.96**	**1.07**	**1.13**	**0.06**

Using Eq.(4), compute DB index for all *k*s and select a *k* having the lowest DB index, as shown in Table 3. In this case, *k*=4 has the lowest DB index, which means products on this wire bonder are divided into 4 categories. For each category, compute the standard deviation on WP and BS of product quality data, to perform stability analysis. Low standard deviation indicates high stability. As shown in Table 4, the stability of the third or fourth category on this wire bonder is remarkably better than the other two product categories. Products having low WP and BS belong to the third or the fourth category. Guided by this result, operators can schedule products with both low WP and BS on this wire bonder to process, such that the product quality fluctuation is reduced as a whole and the relationship with customers is improved.

Table 3: DB Index For All Ks

	k=2	*k=3*	*k=4*	*k=5*	*k=6*	*k=7*	*k=8*	*k=9*
DB index	*0.33*	*0.18*	*0.09*	*0.84*	*0.48*	*0.33*	*0.27*	*0.35*

Table 4: Standard Deviation Of WP And BS For Each Cluster

cluster index	WP	BS
1	0.13	0.11
2	0.12	0.12
3	0.06	0.08
4	**0.05**	**0.06**

4. CONCLUDING REMARKS

In packaging in semiconductor manufacturing, production in wire-bonding is the most complicated, because wire bonders require operators to set up many parameters, such as force, current, and time, to perform bonding. So it is worth analyzing the stability of product quality in wire bonding step. In this paper, we first apply self-organizing map (SOM)-based k-means to product quality data in wire bonding and propose a grid-based heuristic initialization to accelerate the convergence of k-means, then perform stability analysis on quality data on each product category. Simulation results on real product quality (wire pull and ball shear) data in wire bonding from semiconductor packaging and testing vendor, show the quality stability is differentiated over product categories on a wire bonder. Based on this, for to-be-processed products, operators can choose the suitable wire bonder to run, which will reduce the product quality fluctuation at large and improve relationship with customers.

5. ACKNOWLEDGEMENT

The authors want to thank the anonymous referees for their helpful comments and suggestions on this manuscript. This work was supported by the major project of the National Science and Technology (scheduling and control software research and development for the packaging and testing production line: 2011ZX02601-005).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

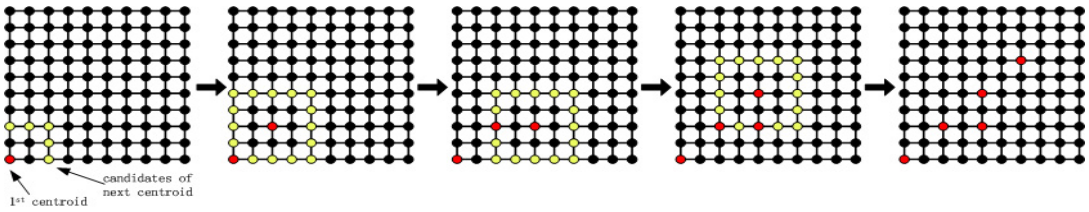
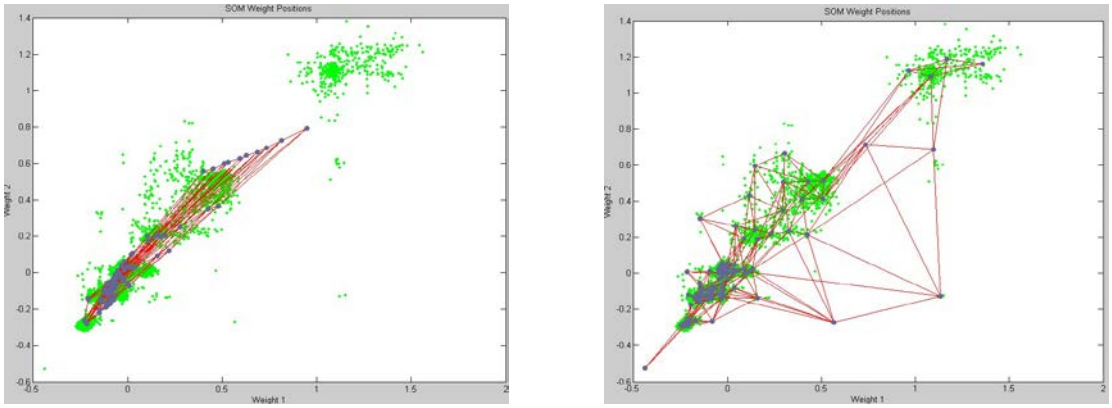


Figure 2: Procedure Of Grid-based K-centroids Initialization Method



(a) (b)

Figure 3: Distribution Of Weight Vectors. (a) After Initialization, (b) After Training

REFRENCES:

[1] George Harman, "Wire Bonding In Microelectronics", New York: McGraw-Hill, 2010, pp. 13-38.

[2] Mac Queen, "Some Methods For Classification and analysis of multivariate observations", *Fifth Berkeley Symposium on Mathematics, Statistics and Probability*, University of California Press, 1967, pp. 281-197.

[3] X. P. Qin, S. J. Zheng, T. T. He, M. Zou and Y. Huang, "Optimizated K-means algorithm and application in CRM system", International Symposium on Computer Communication Control and Automation (3CA), May 5-7, 2010, pp: 519-522.

[4] S. Ray and R.Turi, "Determination of Number of Clusters in K-Means Clustering and Application in Color Image Segmentation", *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*, 1999, pp: 137-143.

[5] Mehrdad Mahdavi and Hassan Abolhassani, "Harmony *K*-means algorithm for document clustering", *Data Mining and Knowledge Discovery*, Vol. 18, Issue 3, 2009, pp: 370-391

[6] Juha Vesanto and Esa Alhoniemi, "Clustering of the Self-Organizing Map", *IEEE Transactions on Neural Networks*, Vol. 11, NO. 3, May 2000, pp: 586-600.

[7] J. C. Bezdek, "Some new indexes of cluster validity", *IEEE Trans. Syst., Man, Cybern. B*, Vol. 28, 1998, pp: 301-315.

[8] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set", *Psychometrika*, Vol. 50, NO. 2, June 1985, pp: 159-179.

[9] D. L. Davies and D. W. Bouldin, "A cluster separation measure", *IEEE Trans. Patt. Anal. Machine Intell.*, Vol. PAMI-1, Apr. 1979, pp:224-227.

[10] T. Kohonen, "Self-organized formation of topologically correct feature maps", *Biological Cybernetics*, Vol. 43, NO. 1, pp: 141-152.

[11] N. Chen, B. Ribeiro, A. Vieira and A. Chen, "Clustering and visualization of bankruptcy trajectory using self-organizing map", *Expert Systems with Applications*, Vol. 40, NO. 1, Jan. 2012, pp: 385-393.

[12] Kadim Taşdemir, "Vector quantization based approximate spectral clustering of large datasets", *Pattern Recognition*, Vol. 45, NO. 8, Aug. 2012, pp: 3034-3044.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[13] K. Taşdemir and E. Merenyi, "Exploiting Data Topology in Visualization and Clustering of Self-Organizing Maps", *IEEE Transactions on Neural Networks*, Vol. 20, NO. 4, April 2009, pp: 549-562.

[14] Y. Pandit, Y. Badhe, B. Sharma, S. Tambe and B. Kulkarni, "Classification of Indian power coals using K-means clustering and Self Organizing Map neural network", *Fuel*, Vol. 9, NO. 1, Jan. 2011, pp: 339-347.

[15] Qing-Shan Deng and Guo-Ping Mei, "Combining self-organizing map and K-means clustering for detecting fraudulent financial statements", *IEEE International Conference on Granular Computing*, 17-19 Aug. 2009, pp:126-131.

[16] G. H. Sun, S. Abe, O. Takei, Y. Hakozaki and T. Matsui, "A Novel Non-contact Infection Screening System Based on Self-Organizing Map with K-means Clustering", Database Theory and Application, Bio-Science and Bio-Technology, Vol.258, 2011, pp:125-132.

[17] J. M. Peña, J. A. Lozano and P Larrañaga, "An empirical comparison of four initialization methods for the K-means algorithm", *Pattern Recognition Letters*, Vol. 20, NO. 10, Oct. 1999, pp: 1027-1040.

[18] S. Redmond and C. Heneghan, "A method for initializing the K-means clustering algorithm using *kd*-trees", *Pattern Recognition Letters*, Vol. 28, NO. 8, June 2007, pp:965-973.

[19] A. Likas, N. Vlassis and J. Verbeek, "The global k-means clustering algorithm", Pattern Recognition, Vol. 36, NO. 2, Feb. 2003, pp: 451-461.

[20] Simon Haykin, "Neural Networks: A Comprehensive Foundation", New York: Macmillan College Div., 1994, pp:408-434.