

Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: [www.tjmcs.com](http://www.tjmcs.com)



## A New Method for Clustering in Credit Scoring Problems

Mohammad Reza Gholamian<sup>a</sup>, Saber Jahanpour<sup>b</sup>, Seyed Mahdi Sadatrasoul<sup>a1</sup>,

<sup>a</sup> School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

<sup>b</sup> Department of financial management, Shahid Beheshti University, Tehran, Iran  
[gholamian@iust.ac.ir](mailto:gholamian@iust.ac.ir), [jahanpour.saber@gmail.com](mailto:jahanpour.saber@gmail.com)

### Article history:

Received December 2012

Accepted February 2013

Available online February 2013

### Abstract

Due to the recent financial crisis and regulatory concerns of Basel II, credit risk assessment has become one of the most important topics in the financial risk management. Quantitative credit scoring models are widely used to assess credit risk in financial institutions. In this paper we introduce Time Adaptive self organizing Map Neural Network to cluster creditworthy customers against non credit worthy ones. We test this Neural Network on Australian credit data set and compare the results with other clustering Algorithm's include K-means, PAM, SOM against different internal and external measures. TASOM has the best performance in clusters customers.

**Keywords:** Credit Scoring; Banking Industry; Clustering, Time adaptive neural network

## 1. Introduction

A credit granting decision needs an accurate decision support system because even a little improvement in Accuracy translates to great money saving for financial companies. Credit scoring is the most widely used technique that helps lenders make credit granting decisions. Its main idea is to estimate the applicant's probability of default in terms of the characteristics recorded on the application form or credit bureau. "Estimation is done by a quantitative model that is built on the basis of historical data of past applicants.

---

<sup>1</sup>Corresponding author

Tel: +98(21) 7322-5067 Fax: +98(21) 7322-5098 E-mail addresses: [sadatrasoul@iust.ac.ir](mailto:sadatrasoul@iust.ac.ir)

Different quantitative methods with different disciplines have been used for building credit scoring models, include statistical, mathematical, machine learning and other models. Some researchers have shown that hybridizing the clustering methods with the mentioned methods could improve the accuracy of the models[2-4].

Self Organizing Map Neural Network (SOM), which was first introduced by Kohonen [5], is a basic neural network for clustering[26]. There are some papers that use the SOM Neural Network to cluster credit scoring data or use it as an auxiliary for the main clustering method to support its weaknesses[2, 4]. According to[6], In the SOM neural network the learning rate is decreasing over time, so it is good for static environments. However, for dynamic and unstable environments that have many changes over time, this method is not appropriate. Using adaptive learning parameters whose values change based on the changes of environment and behavior of input variables as time passes can be very useful for dynamic environments[7].

This paper extends the third method's ability by introducing a new Neural Network that produces better results than other Clustering Algorithms and helps banks to analyze customer credit worthiness better. The results are experienced against the UCI Australian dataset from University of California Irvine Machine Learning Repository

Next section introduces Clustering algorithms. The used dataset is introduced in section three. Section four discusses about Internal and External cluster performance measurements. Section five displays experimental results of all algorithms, finally, the paper is concluded in section six.

## 2. Clustering methods

The aim of clustering methods is to group patterns on the basis of a similarity (or dissimilarity) criterion where groups (or clusters) or set of similar patterns. Clustering techniques can be roughly divided into five categories[6]:

- Hierarchical;
- Partitioning;
- Model-Based;
- Density-Based;
- Grid-Based.

In this paper a new model based clustering method is compared with hierarchical and partitioning methods in the domain of credit scoring. So we just investigate the first three categories, and introduce some of the methods in each of them, then we introduce the new model-based method TASOM.

### 2.1. Hierarchical

Hierarchical clustering methods are able to find structures which can be further divided into substructures recursively[8-11]. The result is a hierarchical structure of groups known as dendrogram as shown in figure (1). The hierarchical algorithms have two main types: agglomerative which starts with points as individual clusters and merges the two closest ones in each iteration until only one cluster remains and divisive which starts

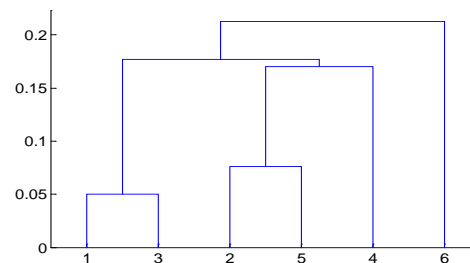


Fig 1. Dendrogram in hierarchical

with one cluster and splits the most dissimilar pairs in each iteration until each cluster contains a point[6].In this paper we use three types of agglomerative hierarchical algorithms single link, complete line, group average.

**2.1.1. Hierarchical Agglomerative Clustering Algorithm:** The main issue in this algorithm is updating the proximity matrix. The algorithm in each method (single link, complete link, group average), has its own procedure. This difference is met in the 4th step. Figure (2) shows nested clusters which can build using these methods. The steps of the algorithm are shown in the following.

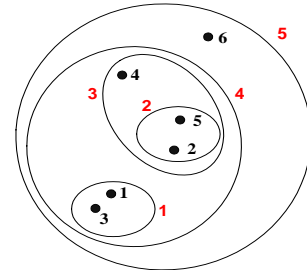


Fig 2. Set of nested clusters in hierarchical

1. Compute the proximity matrix, (we assume Euclidian Distance  $D(C_i, C_j) = \sqrt{\sum_{i=1}^n (x_i - x_j)^2}$ , also we can use the Manhattan and etc.).
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters,(objects that have the lowest dissimilarity measure)
  - a. Single link similarity is based on the two most similar (closest) points in the different clusters.
  - b. Complete linkage is based on the two least similar (most distant) points in the different clusters.
  - c. Group average is based on the average of pairwise proximity between points in the two clusters.  $(proximity)(c_i \times c_j) = \frac{\sum_{p_i \in c_i \text{ and } p_j \in c_j} proximity(p_i, p_j)}{|c_i| \times |c_j|}$ .
5. Update the proximity matrix.
6. **Until** only a single cluster remains.

**2.2. Partitioning**

Partitioning clustering methods are often based on the optimization of an appropriate objective function and try to obtain a single partition of data without any other sub-partition as hierarchical algorithms do. The result is the creation of separated hyper surfaces among clusters. Figure (3) shows a partitioned clustering which can built using these methods. k-means and Partition around medoids (PAM) are used in this paper from this category of clustering algorithms.

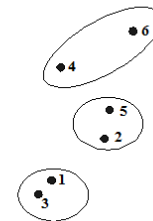


Fig3.Partitioned clustering sample

**2.2.1. K-means :**K-means is one of the famous clustering methods due to its easy and rapid usability [4, 12]. One of the main challenges in using the k-means is its local optimality problem. This problem yields

to the clusters which are heavily influenced by the initial solutions which can fed to the algorithm. k-means steps are shown in the following steps[13]:

1. Select k objects randomly in the dataset X as initial seeds for the cluster's centers.
2. Assigns each object x to the closest center  $c_i$ ,  $i=1,2,..K$ , based on the mean value of the objects in the cluster.
3. Recompute the center of each cluster(calculate the mean value of objects for each cluster).
4. Repeat steps 2 and 3 until centers don't change.

Typical convergence criteria are: no reassignment of patterns to new cluster centers, or minimal decrease in squared error. The iterative K-Means minimizes an objective function, commonly a squared error function defined as the sum of distances of the n data points  $x_j$ ,  $j=1,..,n$ , from their respective cluster centers  $c_i$ ,  $i=1,..,k$ .

Many works have been done to improve the k-means weaknesses[14, 15]. However the basic algorithm is the most widely used one. So we use the simple k-means for comparison in our experiments.

**2.2.2. Partitioning Around Medoids (PAM):** Because k-means is sensitive to outlier, K-medoids chooses actual objects to represent the data and the remaining objects are clustered according to their similarity to the representative object [16].PAM is one of the first K-medoids algorithms which tries to determine K clusters for n objects. K-medoids steps are as follows[16]:

1. Select k objects randomly in the dataset X as initial representatives for the cluster's centroids.
2. Assigns each object x to the closest cluster  $c_i$ ,  $i=1,2,..K$ , with nearest representative objects.
3. Randomly select a non representative object,  $O_{randomselected}$  from X dataset.
4. Compute the total cost of swapping a representative object  $O_i$ , with  $O_{randomselected}$  (the cost function varies and computes the average dissimilarity, so we can use different distance functions to compute it).
5. If the cost of swapping was negative then swap  $O_i$ , with  $O_{randomselected}$ .
6. Repeat steps 2 to5 until centroids don't change.

Compared to k-means, PAM has the advantage of robustness against noisy data and outliers, but its more costly than k-means[16, 17]. PAM works well for small datasets but not for large datasets.

### 2.3. Model-Based

Model based clustering methods try to optimize the fitness between data and a mathematical model[16]. In fact in the model-based clustering algorithms, one uses certain models for clusters and then tries to optimize the fitness between the models and the data [18]. It is assumed that the data are generated by combining different probability distributions and each component in an algorithm represents a new cluster. In the model based clustering group SOM and TASOM are investigated in this paper.

**2.3.1. Self Organizing Map (SOM):** Self

Organizing Map (SOM) is a type of competitive learning networks that defines a spatial neighborhood for each output unit [5, 19]. In fact one of the main properties of SOM is preserving the topology[20]. In the topologic map, neighborhood input patterns activate nearby output patterns in a periodical manner. SOM topologic maps consist of some two dimensional arrays of units which are connected to all N input nodes. Figure (4) shows topologic map and input patterns. The algorithm steps are as follows [20]:

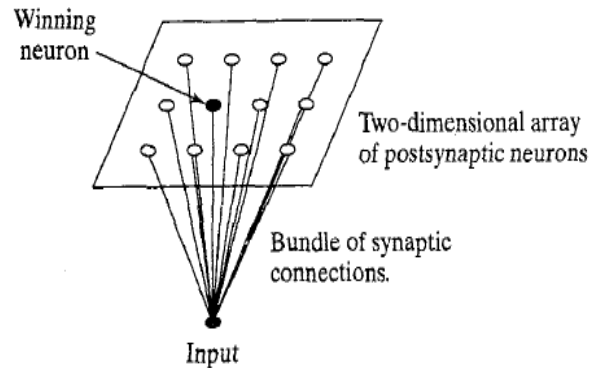


Fig 4.Kohonenself organizing map [1]

The algorithm steps are as follows [20]:

1. Set initial learning rate and neighborhood (And the size of the neighborhood can set from one half to two third of the network size.) and initial weights to small numbers.
2. Use a topologic pattern and evaluate the network outputs(the neighborhood shape can be rectangular, square and circular[12].).
3. Assuming Euclidean distance as the dissimilarity measure, select the wining unit  $(a_i, a_j)$ :

$$\|x - w_{a_i a_j}\| = \min_{i,j} \|x - w_{i,j}\|$$

4. update the weight:

$$w_{i,j}(t + 1) = \begin{cases} w_{i,j}(t) + \eta[x(t) - w_{i,j}(t)] & \text{if } (i,j) \in N_{a_i a_j}(t) \\ w_{i,j}(t) & \text{otherwise} \end{cases}$$

Where  $\eta(t)$  is the leaninging rate at t and  $N_{a_i a_j}(t)$  is the neighborhood of  $(a_i, a_j)$  at time t.

5. Decrease  $\eta(t)$  and the neighborhood  $N_{a_i a_j}(t)$ .
6. Repeat step 2 to 5 until a maximum iterations reached or the change in the value of weights become less than a pre specified threshold.

**2.3.2. TASOM Neural Algorithm with Adaptive Learning Rates and Neighborhood Sizes:** The time adaptive self organizing map neural (TASOM)[7] network is a modified self organizing map (SOM) neural network with adaptive learning rates and neighborhood sizes. Every neuron in the TASOM has its own learning rate and neighborhood size. For each new input vector, the neighborhood size and learning rate of the winning neuron and the learning rates of its neighboring neurons are updated. A scaling vector is also employed in the TASOM algorithm to compensate for scaling transformations. Analysis of the updating rules of the algorithm reveals that the learning parameters may increase or decrease to be adapt to a changing environment, such that the minimum increase or decrease is achieved according to a specific measure.

The TASOM algorithm with adaptive learning rates and neighborhood sets may be specified in the following eight steps:

1. Initialization: Choose some values for the initial weight vectors,  $W_j(0)$  where  $j=1,2,\dots,N$ ;

And  $N$  is the number of neurons in the lattice. The learning rate parameters  $\eta_i(0)$  should be initialized with values close to unity. The constant  $\alpha, \beta, \alpha_s$  and  $\beta_s$  can have any values between zero and one.

The constant parameters  $s_f$  and  $s_j$  should be set to satisfy the application's needs.  $R_i(0)$  should be set to include all the neurons. The components  $S_k(0)$  of the scaling vector  $S(0) = [S_1(0), \dots, S_p(0)]^T$  should be set to small positive values, where  $P$  is the dimension of the input and weight vectors. The parameters  $E_k(0)$  and  $E_{2k}(0)$  may be initialized with some small random values. Neighboring neurons of any neuron  $i$  in a lattice are included in the set  $NH_i$ .

2. Sampling: Get the next input vector from the input  $x$  distribution. The assumption is that the input distribution is unknown to the TASOM algorithm, and the input vectors are received in an arbitrary order.
3. Similarity matching: Find the best-matching or winning neuron  $i(x)$  at time, using the minimum distance Euclidian norm scaled by the scaling vector  $S(n)$ .
4. Updating the neighborhood size: Adjust the neighborhood set  $\Lambda_i(n)$  of the winning neuron  $i(x)$  by the following equations:

$$\Lambda_i(n+1) = \{j \in N | d(i, j) \leq R_i(n+1)\}$$

where:

$$R_i(n+1) = R_i(n) + \beta \left( g \left( (sg \cdot |NH_i|)^{-1} \times \sum_{j \in NH_i} \|w_i(n) - w_j(n)\|_s \right) - R_i(n) \right)$$

And  $\beta$  is a constant parameter between zero and one which controls how fast the neighborhood sizes should follow the local neighborhood size errors  $\sum_{j \in NH_i} \|w_i(n) - w_j(n)\|_s$ . The scalar function  $| \cdot |$  gives the cardinality of a set and the  $d(i, j)$  is the distance between two neurons  $i$  and  $j$  in the lattice.

The neighborhood sets of the other neurons do not change. Function  $g(z)$  is a scalar function for which  $(dg(z))/dz \geq 0$  for  $z \geq 0$ , and is used for normalization of the weight distances. For 1-D lattices of  $N$  neurons,  $g(0) = 0$  and  $0 \leq g(z) \leq N$ , and for 2-D lattices of  $M \times M$  neurons,

$$0 \leq g(z) \leq M\sqrt{2}.$$

5. Update the learning-rate parameters  $\eta_j(n)$  in the neighborhood  $\Lambda_i(x)(n+1)$  of the winning neuron  $i(x)$  by

$$\eta_j(n+1) = \eta_j(n) + 1 \left( f(\|x(n) - w_j(n)\|_s / s_f) - \eta_j(n) \right) \text{ for } j \in \Lambda_{i(x)}(n+1)$$

The learning rate parameters of other neurons do not change. Function  $f(\cdot)$  is a monotonically increasing scalar function such that for each positive  $z$ , we have  $0 < f(z) \leq 1$  and  $f(0) = 0$ .

6. Updating the synaptic Weights : Adjust the synaptic weight vectors of all output neuron in the neighborhood  $\Lambda_{i(x)}(n+1)$
7. Updating the Scaling vector: Adjust the scaling vector  $S(n) = [S_1(n), \dots, S_k(n), \dots, S_p(n)]^T$  by the following equation:  $S_{k+1}(n) = \sqrt{(E_{2k}(n+1) - E_k(n+1)^2)}$  where  $E_{2k}(n+1) = E_{2k}(n) + \alpha_s(x_k^2(n) - E_{2k}(n))$ ,  $E_k(n+1) = E_k(n) + \beta_s(x_k(n) - E_k(n))$ , and  $(z)^+ = \max(z, 0)$ .
8. Continuous learning: go to Step 2

### 3. Data sets

To test the performance of our proposed algorithm on real world problems, Australian credit data sets from UCI machine learning repository were used. This data set contains 14 attributes. The number of classes is one which mainly indicates the default/Non default, and the number of instances used in the data set is 690.

### 4. Clustering performance metrics

The basic idea of clustering is to group similar (or related) objects in the same group, on the other hand, objects of different groups be dissimilar (or unrelated) to each other[6].

There are mainly two types of measures:

- External measures: using the class label for cluster analysis.
- Internal measures: use the vectors for analysis.

In the following a description of the measures investigated in this paper for evaluating the results are represented.

#### 4.1. External Measures

External criteria are used either (a) for a comparison of a clustering structure  $C$ , produced by a clustering algorithm, with a partition  $P$  of  $X$  drawn independently from  $C$  or (b) for measuring the degree of agreement between a predetermined partition  $P$  and the proximity matrix,  $PM$ , of  $X$ [17].

**4.1.1. Rand Index:** The Rand Index measures the fraction of the total number of pairs that are either in the same cluster and partition, or in different clusters and partitions[17, 21]. Index value is between 0 and 1. Higher values means better Clustering.

**4.1.2. Jaccard Coefficient:** The Jaccard Coefficient measures the proportion of pairs that are in the same cluster and partition from those that are either in the same cluster or in the same partition[17, 21]. Index value is between 0 and 1. Higher values mean better Clustering.

**4.1.3. Fowlkes-Mallows Index:** This Index is the geometrical mean of two probabilities:

1. Probability of two random objects are in the same cluster given they are in the same group.
2. Probability of two random objects are in the same group given they in the same cluster[17, 22]. Index value is between 0 and 1, Higher values mean better Clustering.

**4.1.4. Adjusted Rand Index:** The adjusted Rand index proposed by Hubert and Arabie, 1985[23]. In this algorithm there are two sets,  $U$  is external criterion and  $V$  is a clustering result. The  $U$  and  $V$  partitions are picked at random such that the number of objects in the classes and clusters are fixed. Index value is between 0 and 1. Higher values mean better Clustering.

#### 4.2. Internal Measures

Internal Criteria measure the fitness of data in the clustering structures produced by a clustering algorithm, but using only information inherent to the data set[17].

**4.2.1. Davies-Bouldin Index:** Davis-Bouldin Index is the average similarity between each cluster and its most similar one. Small values of Index are indicative of the presence of compact and well-separated clusters[17, 24].

**4.2.2. Dunn Index:** This index measures dissimilarity between clusters. If clustering results are well-separated clusters, Dunn Index values will be large, since the distance between clusters is expected to be large and the diameter of the cluster is expected to be small[17, 24].

**4.2.3. Silhouette Index:** The Silhouette Index is useful when it is seeking compact and clearly separated clusters[17, 25]. The global silhouette value is used as a validity index for calculated clusters. In order to choose the optimal number of clusters for a data set using this index, choose the partitions with the maximum Silhouette Index.

**4.2.4. R- Squared Index:** R Squared (RS) index measures the dissimilarity of clusters. Formally it measures the degree of homogeneity degree between groups[26]. The values of RS range from 0 to 1 where 0 means that there are no differences among the clusters and 1 indicates that there are significant differences among the clusters.

## 5. Experiment Results and discussion

The Australian credit data set is clustered with 5 different algorithms, Agglomerative Hierarchical clustering, K-means, Partitioning around Medoids, Self Organizing Map neural Network and Time Adaptive Self Organizing Map Neural Network. These algorithms are tested from two to ten clusters and their best results are compared with each other. Table (1) shows the external indexes and their appropriate variances for different clustering algorithms. The higher values of external indexes indicate that the clustering algorithm has better performance. The lower values of external indexes variances indicate that the clustering algorithm has better and robust performance. The best performers for each index are bolded.

**Table 1. External indexes and variances (VAR) for different clustering algorithms**

| Algorithm    | Rand        | VAR            | Adjusted   | VAR            | JACCARD     | VAR            | Fowlkes-    | VAR           |
|--------------|-------------|----------------|------------|----------------|-------------|----------------|-------------|---------------|
| Hierarchical | 0.51        | <b>0.00002</b> | 0.02       | 0.0001         | 0.5         | <b>0.00008</b> | <b>0.71</b> | <b>0.0001</b> |
| K-MEANS      | 0.54        | 0.0005         | 0.08       | 0.0005         | 0.50        | 0.007          | <b>0.71</b> | 0.009         |
| PAM          | 0.54        | 0.0004         | 0.02       | <b>0.00005</b> | <b>0.59</b> | 0.006          | 0.63        | 0.007         |
| SOM          | 0.56        | 0.0002         | 0.11       | 0.001          | 0.49        | 0.004          | 0.69        | 0.006         |
| TASOM        | <b>0.63</b> | 0.0007         | <b>0.2</b> | 0.004          | 0.53        | 0.004          | 0.69        | 0.006         |

TASOM shows the best result at Rand and Adjusted Rand index, PAM is the best clustering algorithm from JACCARD index point of view and Fowlkes-Mallows index reported that Agglomerative Hierarchical clustering, K-means are the best performers.

Table (2) shows the internal indexes and their appropriate variances for different clustering algorithms. The best performers for each index are also bolded in this table.

**Table 2. Internal Indices and variances (VAR) for different clustering algorithms**

| Algorithm    | Silhouette  | VAR          | DAVIS BOULDIN | VAR          | DUNN       | VAR         | R-Square    | VAR          |
|--------------|-------------|--------------|---------------|--------------|------------|-------------|-------------|--------------|
| Hierarchical | <b>0.97</b> | <b>0.001</b> | <b>0.72</b>   | 0.17         | <b>0.7</b> | <b>0.09</b> | 0.79        | 0.01         |
| K-MEANS      | 0.92        | 0.05         | 0.47          | <b>0.004</b> | 0.68       | 0.94        | <b>0.84</b> | 0.01         |
| PAM          | 0.86        | 0.01         | 0.53          | 0.02         | 0.3        | 0.13        | 0.8         | 0.05         |
| SOM          | 0.92        | 0.02         | 0.47          | 0.01         | 0.60       | 0.97        | <b>0.84</b> | <b>0.004</b> |
| TASOM        | 0.96        | 0.01         | 0.63          | 0.04         | 0.62       | 0.07        | 0.8         | 0.05         |

Agglomerative Hierarchical clustering shows the best result under Silhouette and DAVIS index, TASOM has a considerable result and it has the next rank in both indexes. Agglomerative Hierarchical clustering is also the best performer in DUNN index and K-means and SOM jointly shown the best result in R-Square index.

In order to recognize the best clustering algorithm, the mean rank of index and mean rank of variances are computed for each algorithm in different index. The results of external indexes are reported in fig. 5.



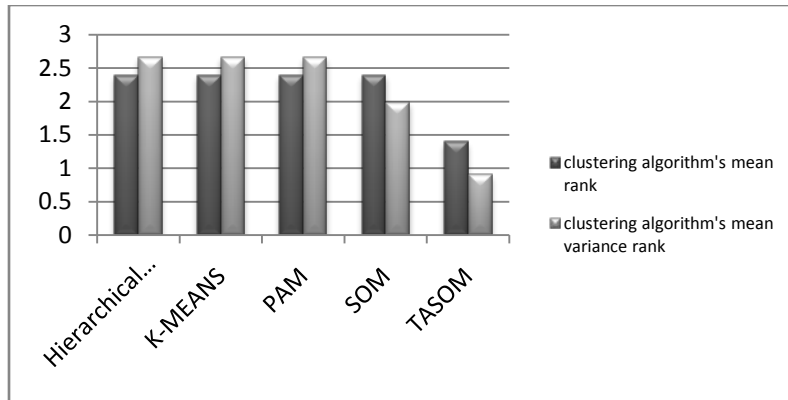


Fig.5. Clustering algorithms mean rank and mean variance rank across different external performance indexes

It can be seen that TASOM has the lowest mean rank and the lowest mean variance rank among different algorithms. The results of internal indexes are reported in fig. 6.

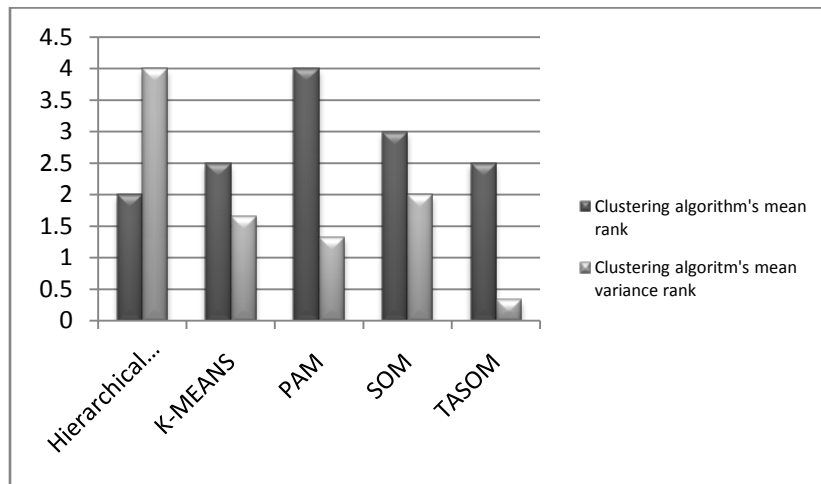


Fig.6. Clustering algorithms mean rank and mean variance rank across different internal performance indexes

As shown Agglomerative Hierarchical clustering has the best rank but its variance is the worst. K-means and TASOM are at the second level in the mean rank, but TASOM has the better variance. It can be seen that TASOM has the lowest variance at a reasonable mean rank of performance indexes.

## 6. Conclusion

In this paper several clustering algorithms are used to cluster Australian data set and their performances are compared with each other. The results of external indices compression reveals that TASOM has the best performance in clustering the data. According to RAND and Adjusted Rand Indices TASOM has the best performance and after that SOM, K-means and PAM are the next. According to JACCARD index PAM is the best and TASOM and K-means are the next. Finally, FM reveals that K-means, Hierarchical and TASOM are best Algorithms. Internal Indices results conclude that TASOM has the best clustering performance after Agglomerative Hierarchical clustering. Finally, According to both Internal and external indices The TASOM has the best performance among all, so it can be used in clustering financial data because of its variable parameters over time that makes it suitable for dynamic environments such as financial markets. TASOM can also be used in different hybrid techniques which used classifiers after clustering.

Further analysis of in credit scoring could also include using classifiers and TASOM in building hybrid algorithms to classify good applicants from the bad ones better.

## References

1. S.S. Haykin, S.S., *Neural networks and learning machines*. Vol. 3., Prentice Hall (2009).
2. N.C. Hsieh, *Expert Systems with Applications*, 28(4): p. 655(2005).
3. N.C. Hsieh, L.P. Hung, *Expert Systems with Applications*, 37(1): p. 534(2010).
4. J. Huysmans, *Expert Systems with Applications*, 30(3): p. 479(2006).
5. T. Kohonen, *Proceedings of the IEEE*, 78(9): p. 1464(1990).
6. P.N. Tan, M. Steinbach, V. Kumar, *Introduction to data mining*. Pearson Addison Wesley Boston (2006).
7. H. Shah-Hosseini, R. Safabakhsh, *IEEE Transactions on*, 33(2): p. 271(2003).
8. J.K. Jain, M.N. Murty, P.J. Flynn, *ACM computing surveys (CSUR)*, 31(3): p. 264(1999).
9. P.H.A. Sneath, R.R. Sokal, *Numerical taxonomy. The principles and practice of numerical classification*. (1973).
10. Jr. Ward J.H., *Journal of the American statistical association*, p. 236(1963).
11. M. Kuchaki Rafsanjani, Z. Asghari Varzaneh, and N. Emami Chukanlo, *The Journal of Mathematics and Computer Science*, 5(3): p. 229(2012).
12. R. Maghsoudi, *The Journal of Mathematics and Computer Science*, 2(2): p. 329(2011).
13. J.A. Hartigan, M.A. Wong, *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1): p. 100(1979).
14. B. Kövesi, J.M. Boucher, and S. Saoudi, *Pattern Recognition Letters*, 22(6): p. 603(2001).
15. T. Kanungo, *IEEE Transactions on*, 24(7): p. 881(2002).
16. J. Han, M. Kamber, *Data Mining Concepts and Techniques*, Morgan Kaufmann Publications (2011).
17. P.J. Rousseeuw, L. Kaufman, *Finding Groups in Data.*, Wiley Online Library (1990).
18. Gan, G., C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*, SIAM, 2007.
19. Kohonen, T., *Self-organization and associative memory*. Self-Organization and Associative Memory, 100 figs. XV, 312 pages.. Springer-Verlag Berlin Heidelberg New York. Also Springer Series in Information Sciences, volume 8, 1988. 1.
20. A.K. Jain, J. Mao, and K.M. Mohiuddin, *Computer*, 29(3): p. 31(1996).
21. M. Halkidi, Y. Batistakis, and M. Vazirgiannis, *ACM Sigmod Record*, 31(2): p. 40(2002).
22. E.B. Fowlkes, C.L. Mallows, *Journal of the American statistical association*, p. 553(1983).
23. M. Halkidi, Y. Batistakis, and M. Vazirgiannis, *Journal of Intelligent Information Systems*, 17(2): p. 107(2001).
24. D.L. Davies, D.W. Bouldin, *IEEE Transactions on*, (2): p. 224(1979).
25. P.J. Rousseeuw, *Journal of computational and applied mathematics*, 20,p. 53 (1987).
26. S. Sharma, *Applied multivariate techniques*. (1996).