International Journal of Computer Applications (0975 – 8887)
Volume 50– No.21, July 2012

Weight Optimize by Automatic Unsupervised Clustering using Computation Intelligence

C. Lowongtrakool
KMITL, Dept. of CS
Ladkrabang, Bangkok
Thailand, 10520

N. Hiransakolwong
KMITL, Dept. of CS
Ladkrabang, Bangkok
Thailand, 10520

ABSTRACT
Several techniques are applied to the unsupervised clustering data analysis. The entered data is dataset of input without aclass of answer. Besides, the beginning weight and the values of cluster groups of answers are defined. However, the most important parameter among these three factors (unsupervised clustering, weight, and the number of clusters) is the determination of beginning weight for the system. If the weight is well determined after the starting point, the system will be able to calculate track and figure out the answers more rapidly and precisely. Therefore, this paper proposes the method to optimize the weight of the system by conducting a technique of computational intelligence to manage the unsupervised clustering data analysis. The experiment starts from finding the value of beginning weight and then it is processed later by using sample datasets from UCI Machine Learning Repository including iris, balance and wine. The result shows that the efficiency of data classification increases to 99.3%, 83.6% and 47.0%, respectively, and finding automatically the initial number of cluster k. Consequently, the outcome reduces the number of predicting clusters to discover approximate answer as well.

Keywords
Weight Optimize, Unsupervised Clustering, Computation Intelligence.

1. INTRODUCTION
The weight determination of neural network is very significant and necessary because it is the first step of algorithm calculation. Actually, the weight determination must be stemmed from learning suitable values obtained from system practice or expert’s experience in specific fields. The optimized weight value will affect a good and precise output of the system. Nevertheless, it is quite difficult to optimize the appropriate value because it does not mean that the proper value will be suitable for all kinds of data. On the other hand, it should be the appropriate value for the data used for analysis. The weight must be adjustable and flexible following the data while the required unsupervised clustering algorithm must apply the weight value with input pattern to process the outcome, for instance, Kohonen’s Winner-Take-All’s network,Kohonen self-organizing feature maps and incremental learning fuzzy neural network.

2. REVIEW OF BACKGROUND KNOWLEDGE
An artificial neural network of unsupervised learning can be called a “competitive network”, based on input data classification with no a priori knowledge in group or type of data. The network will search for particularities and profiles of the data which will be utilized to help identify groups and possible boundary between groups. The network of unsupervised learning will categorize the groups of data with same attributes and separate the groups of data with different attributes to others.

2.1 Kohonen’s Winner-Take-All’s network
Kohonen’s artificial network determines that winner takes all network. It divides the input into S groups as previously defined.The artificial neural networks winner-take-all networks consider sets of S weight sectors. The value of parameter vector is adjusted before learning and the weight vector gathered from randomization must be converted to normalization as described below.

Step1,Define the structure of winner-take-all networks and divide the number of groups into S groups. After that, choose learning rate: η whereη= value near zero and then fix L_{max} (maximum epoch).

Step 2,Randomize the weight w_i while i =1, 2,..., S to convert to normalization

Step 3,Receive training packages on teaching $x = x_k$ to the network to calculate a distance between the weight value of each artificial neural network and input

$$d_i = \| x - w_i \| \quad ; i = 1, 2, \ldots, S.$$

Step 4,Find a neural code winner which is the artificial neural network with similar weight vector close to the input.

Step 5,Adjust the weight vector of artificial neural network winner.

Step 6,If k < Kandk = k + 1, repeat the calculation from Step 3. Untilk => K, then go to Step7.

Step 7,Ifm < L_{max}, define k = 1, m = m + 1 and restart the training process from Step 3. Until m $\geq$ L_{max}, then finish the training.

2.2 Kohonen Self-organizing Feature Maps
The Kohonen’s self organizing feature maps (Kohonon's SOFM or SOM) [11 12 18] are the network of unsupervised learning. It aims to discover the structure of data which SOM calls a map with self-organized dimensions. They are compared to the dimensions of neighboring artificial neural network by building the type of input as the two-dimensional map. SOM depends on neighboring function to maintain the type of weight of input structure. However, it is different from the method of winner that takes all network which only adjusts the weight of the winner. SOM works in 2 modes; practice and calibration. The practice mode is used to generate

a calibrating map by applying the data of practice mode which is a competitive process. On the contrary, the calibration mode classifies new data which has never seen before in any group. SOM consists of node or neuron and each node includes consistent weight vector. Its dimension is equal to an input vector and position on the calibrating map. Moreover, the node layout can be defined as rectangular grid or hexagonal grid. It can be said that this approach calibrates input space from higher dimensional space to lower dimensional space. The process oflaying vector of data space on the calibrating map aims to find the node with the weight close to the data vector of data space and target the position of node for the input data on self-organizing map. It is helpful to force different parts of the network to response the types of input data. The weight of artificial neural network is defined by randomizing from the small values or data training. Then the weight value is adjusted to the teaching data. Furthermore, the competitive learning is utilized for training process whereas the practice data is entered to the network to figure out the distance between the data and the neural weight value of each neuron. The neuron with the least distance between the data and the weight value will be regarded as the winner. After that, the winner and neighbors must be regulated to such data following the defined type meanwhile other neurons still remain the same. The volume of adjustment will decrease upon the distance between the weight value and the data. When the round is higher, the value of adjustment will diminish, too.

2.3 Incremental Learning Fuzzy Neural Network

The incremental learning fuzzy neural network (ILFN) [9 10] is applied to learn the data for one round including both supervised learning and unsupervised learning. To the supervised learning, the algorithm is required for both input and target parts. On the other hand, the unsupervised learning only consists of input part. The algorithm will learn the relationship between the input and the weight value for clustering. This process comprises of certain parts as follows.

Input Subsystem – It is used to enter the parameters to the system by measuring the distance and the weight value which may be randomized from the input values. The weight value must be adjusted by conducting Gaussian function.

Target Subsystem – Each target value is linked to the neural system by matching with the weight value adjusted for each round. The target value is the number of answer cluster of the processed system which can be analyzed to compute membership value by conducting pruning module.

Controller Module – It is beneficial to increase the amount of cluster node in hidden layer of the input subsystem and target subsystem for the supervised learning. AND gate is also utilized to compare both input and target.

Decision Layer – It includes a soft decision; a degree of each membership value, and a hard decision; a calculation to find only one probability value.

3. THE PROPOSED OF AUTOMATIC WEIGHT OPTIMIZE

3.1 Automatic Unsupervised Clustering Computation Intelligence (AUCCI)

Automatic unsupervised clustering computational intelligenceis applied to analyze the data without randomization or the classwithout master data. The system is allowed to learn from the real data by using main techniques as described below.

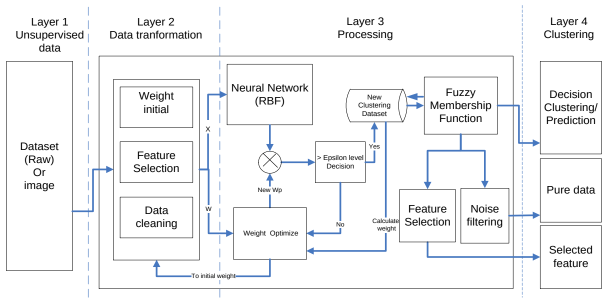


Fig 1: Automatic Unsupervised Clustering Using Computational Intelligence Model

Layer 1: Unsupervised data – The data is entered without identifying its class or type or beginning class. The obtained data may be raw data with some useless data for processing. Feature selection technique is applied and the weight is automatically defined for more effective calculation.
Layer 2: Data transformation – This step helps prepare unfiltered raw data for more effective output. The feature selection technique is utilized to cut the data of duplicated or unnecessary attributes to reduce processing time. The weight value is automatically determined as well [2 6 14 15 17] which will be used for the calculation of neural network in the main system. Moreover, the weight value will be adjusted by learning imported data [16 19 20 21] to automatically find the suitable weight value.
Layer 3: Processing – This stage consists of 3 important algorithms. The first layer of radial basis function [13] is an input layer in the neural network requiring the adjusted weight value from previous layer. The cluster membership value of each imported record must be properly determined for end class. During the measure of membership value, the weight will be adjusted to figure out the appropriate weight value to return to the system. Next, it is an interpretation step of cluster membership value applying fuzzy logic.
Layer 4: Clustering – The results are gained from membership decision of each cluster classified in various classes. These outcomes may help discover new classes of unseen data.
As stated above, the unsupervised clustering of Kohonen's winner-take-all's network, Kohonen's self-organizing feature maps, incremental learning fuzzy neural network and automatic unsupervised clustering computational intelligent needs the weight value to compute the pattern of relationship between the input data and the weight. In addition, some algorithms demand the numbers of answer cluster to learn the relationship of system.

3.2 Automatic Weight Optimize Algorithm

Step1, Enter input (x) as a vector of the first record into the system.

$$X = [x_1, x_1, \ldots, x_n] \qquad (1)$$

Step2,Measure the distance between the input (x) and the value of weight $(X)^T$ by using Euclidean distant.

Step3, Define the value of standard division by applying the mean value of the first weight.

Step4, Calculate the net value which equals y/ standard division.

Step5, Calculate the e value.

$$e = \left(1+\left(\frac{1}{\max(net)}\right)\right)^{\max(net)} \qquad (2)$$

Step6,Calculate the membership value (Y) by applying an activate function.

$$Y = \qquad (3) \qquad \frac{1}{1+\exp\left(\frac{net^2}{e}\right)}$$

Step7,Determine the epsilon value = 0.0001 and the centroid of cluster is C = 1.

Step8,If the membership value > epsilon, adjust the position of weight and standard division in an output class. Then add a new center value (C) and follow the Step 2.

Where:

$$W_{new} = \frac{(w_{old}(num\_of\_cluster-1))+current\_input}{num\_of\_cluster} \qquad (4)$$

$$\text{Standard division} = \sqrt{\left(1+\frac{1}{num\_of\_cluster}\right)st_{old}^2+\frac{(st_{old}-current\_input)^2}{num\_of\_cluster}}$$

(5)

Step9,If the membership value < epsilon, adjust the position of weight and standard division again and follow Step 2.

Step10, Follow this process to complete the dataset in all records and used final weight to Automatic Unsupervised Clustering Using Computational Intelligence system.

4. EXPERIMENTAL RESULTS

A simulation of system efficiency test is programmed by applying with tools in Matlab 2010, on intel®Celeron M 1300MHz. This computer consists of 256KB memory and uses the dataset to test 10 benchmarks including iris, Glass and Thyroid. These above benchmarks are obtained from Machine Learning Repository of Center for Machine Learning and Intelligent Systems [4] as described below.

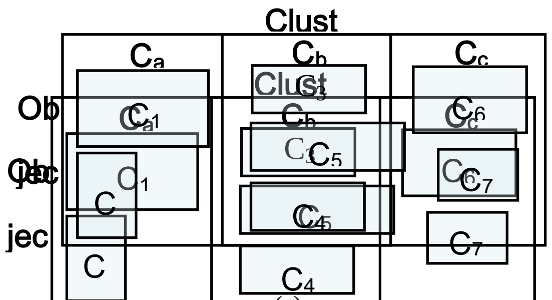
1.A dataset of iris contains 150 records with features collected from each iris such as Setosa, Versicolour and Virginica following width and length. Besides, these records are gathered from the sample groups of 50 records in each group.

2. Dataset of glass consists of 214 records and different features including refractive index, Sodium, Magnesium, Aluminum, Silicon, Potassium, Calcium, Barium and Iron. These various features are useful to create 7 products, for instance, building windows float processed, building windows non float processed, vehicle windows float processed, vehicle windows non float processed, containers, tableware and headlamps.

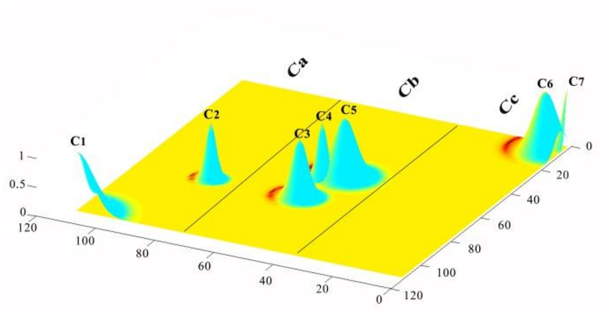
3. A dataset of Thyroid consists of 215 records with some features such as T3resin, Thyroxin, Triiodo thyronine, Thyroid stimulating and TSH_value. The classes of disease are divided into normal, hyper and hypo.

4.1 Subspace Clustering Measure

This procedure examines true value of the unsupervised clustering for data which substitutes the value of subspace cluster [3 8] with C = (O, S). An object O is a subset in the dataset while S indicates the relationship compared to the dimensions of cluster. In addition, the subspace cluster is represented by $\{C_1, \ldots, C_k\}$. All values are based on ground truth = $\{C_a, C_b, \ldots, C_f\}$. To the measure of subspace precision after the calculation, this below formula is conducted.



(a)



(b)

Fig 2: (a)Subspace clustering in 2 Dimension, (b) Subspace clustering in 3 Dimension

$$SC\ (P,Q) = \frac{1}{P}\sum_{Ci\in P}\max_{Cj\,\in\,Q}\{SC\,(Ci,Cj)\} + \sum_{Ci\in P1}\max_{Cj\,\in\,Q}\{SC\,(Ci,Cj)\} \qquad (13)$$

Where: P= set of subspace calculated

P1=P-(max term of calculated subspace)

To the value [15, 16, 18] selection of subspace cluster, P1 is inferior value of maximum value. The quantity of data close to the maximum value, but not over 15%, will be computed for the same cluster as the maximum cluster.

4.2 Results before and after Weight Optimization

Table 1: Comparison of evaluate measures for different clustering algorithm on real world dataset

		Number of cluster	Accuracy
Iris	not weight optimized	11	68.0%
	weight optimized	5	99.3%
glass	not weight optimized	11	67.46%
	weight optimized	9	83.62%
Thyroid	not weight optimized	29	35.26%
	weight optimized	26	47.0%

According to the experiment, the dataset of iris flower, glass and thyroid are tested by conducting the automatic unsupervised clustering computational intelligent system to compare the results before and after the weight optimization.

4.3 Results Compared to Other Algorithms Processed by Weka 3.6 for Clustering Tasks

Table 2: Comparison of evaluate measures for different clustering algorithm on real world dataset

%Accuracy of Clustering	Iris	Glass	Thyroid	Initial K
AUCCI	99.3	83.6	47.0	Auto
Expectation maximisation	84.6	57.1	39.0	manual
FarthestFirst	80.6	77.2	44.5	manual
FilteredClusterer	92.0	65.9	39.1	manual
MakeDensityBasedClusterer	92.6	70.5	38.2	manual
SimpleKmean	92.0	65.9	39.1	manual

Regarding to the table, it demonstrates the measure of precision between AUCCI algorithm and other algorithm clustering (Expectation maximization, FarthestFirst, FilteredClusterer, MakeDensityBasedClusterer and SimpleKmean) of WEKA Version 3.6 including the clustering algorithm. From the results, AUCCI gains more precision in all type of data and finding automatically the initial number of cluster k.

5. CONCLUSION

The weight optimization under the automatic unsupervised clustering using computation intelligent (AUCCI) is useful to adjust the incremental weight learning by applying the first layer of radial basis function and membership function regarding AUCCI's type. The processed results represent the optimized weight value used for further calculation of the first layer of any system. Therefore, it is considered as very significant value. As stated above, it can be seen that the outcome seems better when the weight is automatically and suitably optimized. The weight value gained from this system is also beneficial to other computation intelligent systems requiring the weight value to process with the input or output data. Normally, general algorithms automatically randomize the weight value and the processed results may be good or bad or unexpectable. However, another method of determining the weight value can be manually conducted by only specific experts.From the results, AUCCI gains more precision in all type of data and finding automatically the initial number of cluster k.

6. REFERENCES

[1] A.Abraham, Meta learning evolutionary artificial neural networks, Neurocomputing, 2004, Vol. 56, pp.1–38.

[2] A. J. Al-Shareef and M. F. Abbod, Neural networks initial weights optimisation, in Proceedings of the 12th International Conference on Modelling and Simulation (UKSim '10), 2010, pp. 57–61.

[3] A. Patrikainen and M. Meila, Comparing subspace clusterings, IEEE Transactions on Knowledge and Data Engineering, 2006, 18(7),pp.902–916.

[4] Center for Machine Learning and Intelligent Systems, UCI Machine Learning Repository(2011),http://archive.ics.uci.edu/ml/.

[5] C.zhang, H.Shao, Y.Li,Particle swarm optimization for evolving artificial neural network, IEEE Intl. Conf. on Systems2000, Vol. 4,pp.2487 – 2490.

[6] E.Atashpaz Gargari, C. Lucas,Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, IEEE Congress on Evolutionary Computation, 2007,pp.4661-4667.

[7] S.-J.Han, S.-B. Cho,Evolutionary neural networks for anomaly detection based on the behavior of a program, IEEE Trans. Systems,2006,Vol.36,pp.559-570.

[8] E. Muller, S. Gunnemann, I. Assent, and T. Seidl. Evaluating clustering in subspace projections of high dimensional data. PVLDB, 2009, 2(1),pp.1270–1281.

[9] G. Yen and P. Meesad,Pattern classification by an incremental learning fuzzy neural network, *in Proc.IJCNN***,** 1999, pp. 3230-3235.

[10]G. Yen and P. Meesad, Constructing a fuzzy expert system using the ILFN network and the genetic algorithm, in *Proc. IEEE Inter. Con. Syst. ManCybern.***,** 2000, pp. 1917-1922.

[11]A. Kizilay. and Makal.S,A neural network solution for identification and classification of cylindrical targets above perfectly conducting flat surfaces, J. of Electromagn. Waves and Appl. **21**(2007)2147–2156, doi:10.1163/156939307783152759

[12]T. Kohonen and P. Somervuo ,Self-Organizing Maps of Symbol Strings with Application to Speech Recognition, Neurocomputing, 1998,Vol. 21, No. 1-3, pp.19-30.

[13]K.Miao, F. Chen and Z. G. Zhao, Stock price forecast based on bacterial colony RBF neural network, J.QingDao University. **20** (2007)50–54,doi: CNKI:SUN:QDDD.0.2007-02-011.

[14]M.C.P.de Souto, A.Yamazaki and T.B.Ludernir, Optimization of neural network weights and architecture for odor recognition using simulated

annealing, in Proc. 2002 Intl. Joint Conf. on Neural Networks, 2002,Vol.1, pp.547–552.

[15]M.A. Mohamed, E. A. Soliman, and M. A. El-Gamal, Optimization and characterization of electromagnetically coupled patch an-tennas using RBF neural networks, J. of Electromagn. Waves and Appl.**20** (2006)1101–1114.

[16]M. Yang, D. Kriegman, and N. Ahuja,Detecting Faces in Images: A Survey,IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002,vo1.24, no.1, pp.34-58.

[17]R. Poli, J. Kennedy, and T. Blackwell, Particle swarm optimization, Swarm Intelligence, 2007,vol. 1, pp. 33–37.

[18]T. Kohonen, Self-Organizing Maps, Springer Verlag(Berlin, 2001).

[19]W. M. Jenkins,Neural network weight training by mutation, J.Computers and Structures. **84** (2006) 2107-2112,doi: 10.1016/j.compstruc.2006.08.066

[20]X.He, J.Zeng, J.Jie, Artificial neural network weights optimization design based on MEC algorithm,Conf. on Machine Learning and Cybernetics, 2004,Vol. 6,pp. 3361 – 3364.

[21]Y. Lee, S. H. Oh, and M. W. Kim,The effect of initial weights on premature saturation in back-propagation learning, in Proceedings of the International Joint Conference on Neural Networks,1991, pp. 765–770.