A Novel Hybrid Neural Network for Data Clustering

Donghai Guan, Andrey Gavrilov

Department of Computer Engineering
Kyung Hee University, Korea
donghai@oslab.khu.ac.kr, Avg1952@rambler.ru

Abstract. Clustering plays an indispensable role for data analysis. Many clustering algorithms have been developed. However, most of them suffer either poor performance of unsupervised learning or lacking of mechanisms to utilize some prior knowledge about data (semi-supervised learning) for improving clustering result. In an effort to archive the ability of semi-supervised clustering and better unsupervised clustering performance, we develop a hybrid neural network model (HNN). It is the sequential combination of Multi-Layer Perceptron (MLP) and Adaptive Resonance Theory-2 (ART2). It inherits two distinct advantages of stability and plasticity from ART2. Meanwhile, by combining the merits of MLP, it not only improves the performance for unsupervised clustering, but also supports for semi-supervised clustering if partial knowledge about data is available. Experiment results show that our model can be used both for unsupervised clustering and semi-supervised clustering with promising performance.

1 Introduction

In general, data analysis methods consist of two categories: classification and clustering. Classification is supervised learning. In classification, we are provided with a collection of labeled data items and the problem is to label a newly encountered data item. Typically, the labeled patterns are used to learn the descriptions of classes which in turn are used to label a new pattern. In case of clustering, it is usually performed when no information is available concerning the membership of data items to predefined classes. For this reason, clustering is traditionally seen as part of unsupervised learning [1][2]. Recently, a kind of new data analysis methods is proposed, called semi-supervised clustering. It is different with traditional clustering by utilizing small amount of available knowledge concerning either pair-wise (must-link or cannot-link) constrains between data items or class labels for some items [3][4][5]. Semi-supervised clustering is especially suitable for those applications with partial but not much prior knowledge available. Although many unsupervised clustering methods have been developed, most of them are unable to support semi-supervised clustering. In other words, even some useful information about data is available, but we have no way to effectively utilize them through those methods. So developing a concrete clustering model that supports both unsupervised and semi-supervised learning is urgently needed.

In this paper, we develop a hybrid neural network (HNN) model. This model is originally proposed by us for invariant recognition of visual images [7]. In this work, we propose to use this model for unsupervised and semi-supervised clustering. This model is a sequential combination of Multi-Layer Perceptron (MLP) and Adaptive Resonance Theory-2 (ART2) [6]. HNN combines the advantages of MLP and ART2. On one hand, it inherits stability and plasticity from ART2 [7]. One the other hand, by combining the merits of MLP, semi-supervised cluster is supported. We have tested our method on two popular datasets: Iris and Balance Scale dataset, which are available at UCI Machine Learning Repository [8]. The experiments show the distinct merits of HNN which are also our contributions as follow:

- Its unsupervised clustering accuracy is better than most existing clustering methods.
- When it is used for semi-supervised clustering, small amount of prior information could greatly improve the clustering accuracy.

The structure of the paper is as follows. In section 2, we present the HNN's architecture and learning algorithm in detail. Section 3 is the experiments and comparisons with other clustering methods. We make the conclusion and describe future work in section 4.

2 Our Method

2.1 HNN Architecture

As shown in Fig.1, our proposed hybrid neural network is a combination of MLP and ART2 with MLP in front and ART2 back.

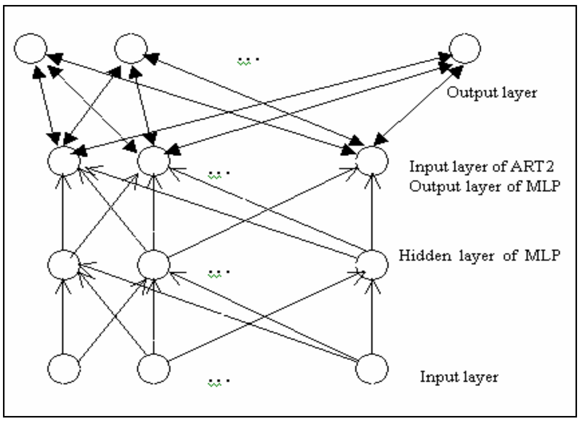

Fig. 1. Architecture of hybrid neural network

When it is used for unsupervised data clustering, the unlabeled data will be sent to the input layer of MLP firstly. Then the output of MLP will be the input of ART2. In

HNN, MLP could be treated as a data preprocessing layer, because it can provide data (features) conversion through its hidden layers. Appropriate data conversion depends on the connection weights of MLP. In our model, MLP utilizes error back propagation (EBP) to adjust its connection weights. We should note that the goal of training here is different with training of traditional MLP for classification. The goal here is to provide some additional help to ART2 through data transformation, so the training is secondary. Long training time for traditional MLP is avoided here. In section 2.2 and 2.3, the detailed algorithm for unsupervised and semi-supervised clustering will be presented.

2.2 HNN for Unsupervised Learning

The notations used in our algorithm are shown in Table 1.

Table 1. Notations

Notation	Descriptions
S_i , $S_i \in R^d$	S_i : input pattern . d : dimension of S_i . i : index of input pattern
O_i , $O_i \in R^m$	O_i : output of MLP given S_i . m : dimension of O_i . i : index of output pattern
NI	Number of neurons in the input-layer of MLP, $NI = d$
NK	Number of neurons in the output-layer of MLP
HLN	Number of hidden layers in MLP
NH	Number of neurons in the hidden layer of MLP (supposed $HLN = 1$)
N_{out}	Number of clusters (number of neurons in output-layer of ART2)
NS_j	Number of samples in cluster j
ρ	Vigilance value of ART2
w_{ij} , $(1 \le i \le NI , 1 \le j \le NH)$	In MLP, connection weight between i^{th} neuron of input-layer and j^{th} neuron of hidden layer (supposed $HLN = 1$)
w_{jk} ,$(1 \le j \le NH , 1 \le k \le NK)$	In MLP, connection weight between j^{th} neuron of hidden layer and k^{th} neuron of output layer (supposed $HLN = 1$)
W_i	The prototype (centroid) of cluster i
D_{ij}	The Euclidean distance between O_i and W_j

When HNN works as unsupervised clustering, its learning process is:

- Unlabeled data S_i is inputted into MLP, O_i is the output of MLP.

- O_i is inputted into ART2 for clustering. If O_i is recognized belonging to class j, then W_j (the prototype of class j) will be treated as the target output of MLP for S_i.
- MLP training will be adjusted based on error back propagation (EBP) algorithm.

The detailed C-like algorithm proceeds as follows:

Algorithm 1: **HNN used for unsupervised learning**

Input: multiple S_i (supposed totally n input patterns)

Output: Cluster number that each input pattern belongs to

Stage 1: HNN initialization

1) MLP initialization: $w_{ij} = 1/NI$, $w_{jk} = 1/NH$

2) ART2 initialization: $N_{out} = 0$

Stage 2: Clustering

3) The i^{th} sample S_i is inputted into MLP

4) If ($i == 1$), ($N_{out} = 1$, and $W_1 = O_1$); else, goto step 5

5) For ($j = 1 : N_{out}$), calculating D_{ij} . Then, select the minimal one D_{ij^*}

6) Vigilance test:

If ($D_{ij^*} < \rho$), successful, S_i is recognized belonging to cluster j^*

W_{j^*} updating. $W_{j^*} = W_{j^*} + D_{ij^*} / (1 + NS_{j^*})$, $NS_{j^*} = NS_{j^*} + 1$,

Goto step 8;
else, Goto step 7

7) $N_{out} = N_{out} + 1$, $W_{j^*} = W_{N_{out}} = O_i$, S_i is recognized belonging to this new cluster

8) MLP training by EBP with a small number of iterations (O_i is actual output, W_{j^*} is target output)

In this algorithm, we should note that training of MLP here is totally different with traditional MLP training. In traditional MLP training, EBP need to reduce the error-function of MLP to a very small value. While in HNN, EBP is used only for decreasing the distance between actual output and target output of MLP. So long time training is not needed.

2.3 HNN for Semi-supervised Learning

Semi-supervised clustering can be used in case of a small amount of prior knowledge available. The knowledge here means partial samples' labels are known before clustering and they will be the "teacher" of MLP. The algorithm works as follows:

Algorithm 2: **HNN used for semi-supervised learning**

Input: multiple S_i (some samples' labels available, S_y)

Output: Cluster number that each input pattern belongs to

Stage 1: HNN initialization

1) MLP initialization: $w_{ij} = 1/NI$, $w_{jk} = 1/NH$

2) ART2 initialization: $N_{out} = 0$

Stage 2: Learning from the samples with labels known

3) Cluster prototype calculation: $W_i = W_i + \left\| S_y - W_i \right\| / (1 + N_i)$

4) MLP training by EBP.

Stage 3: Clustering

The clustering here is same with stage 2 in Alg. 1

From this algorithm, we can see this semi-supervised learning could achieve better result since those labeled data adjust weights of MLP to more appropriate values. In other words, based on these labeled data, the output conversion is more suitable for ART2 to get a better result.

3 Experiments and Comparisons

We test our model on two popular datasets, Iris and Balance Scale. Both of them are available at UCI Machine Learning Repository.

3.1 Experiment for Unsupervised Learning

The dataset in this part is iris, which is one of the most popular data sets to examine the performance of novel methods in pattern recognition and machine learning. There are three categories in the data set (i.e., iris setosa, iris versicolor and iris virginical), each having 50 patterns with four features. Iris setosa can be linearly separated from iris versicolor and iris virginical, while iris versicolor and iris virginical are not linearly separable. Table 2 summarizes some of the clustering results reported in the literature. From the table, we can see that our approach provides better result than most existing methods (except Mercer Kernel Based Clustering). The parameters used for this experiment are shown in Table 3 and existing methods we use in our experiments are as follow:

- GLVQ: general learning vector quantization;
- GFMM: general fuzzy min-max neural network;
- SVC: support vector clustering;
- FCM; fuzzy c-means;
- CDL: cluster detection and labeling network;
- HC: hierarchical clustering: RHC: relative hierarchical clustering;
- FA: fuzzy adaptive resonance theory.

Table 2. Experiment results on Iris

Algorithms	Number of errors	Percentage of errors
GLVQ[9]	17	11.3%
FCM [10]	16	10.6%
GFMM [11]	0~7	0~4.7%
Mercer Kernel Based Clustering [12]	3	2%
SVC[13]	4	2.7%
CDL[14]	6	4%
HC [15]	13~17	8.7~11.3%
RHC[15]	5~6	3.3~4%
FA [16]	6.77~46.4	4.5~30.9%
K-Means	16	10.67%
******HNN (our approach)******	**4**	**2.7%**

Table 3. Parameters in HNN for clustering Iris

MLP	1 hidden layer; 4 neurons in hidden layer
	4 neurons in output layer
	Exponential Sigmoid activation function, a=1
	Learning rate=0.1
	Iterations=1
ART2	Vigilance value R=0.08

In addition to HNN, we also use k-means to cluster Iris. For both k-means and HNN, we find that almost all the mis-clustered samples are in versiclor or virginical. It is not surprised since versicolor and viginical are not linearly separable. In fact, both of k-means and HNN exploits Euclidean distance as similarity measure, however, HNN can greatly improve the clustering performance compared with k-means. The reason is that the MLP part provides feature conversion (or mapping). As a result, most samples in versicolor and virginical are linearly separable after feature conversion (or mapping).

3.2 Experiment for Semi-supervised Learning

We test the performance of HNN for semi-supervised clustering on two datasets. One dataset is Iris, which has been used in last experiment. The other one is extracted from Balance Scale dataset. We randomly select 60 samples, 20 samples for each class. For Balance Scale dataset, there are three categories with four features. The result of clustering is shown in Table 4. For both of the two datasets, 10% of total samples (15 samples in Iris, 6 samples in Balance Scale Dataset) are used for training. We can see that clustering errors can be greatly reduced. The parameters we used in this experiment are shown in Table 5.

Table 4. Performance comparison between unsupervised and semi-supervised clustering

	IRIS	Balance Scale (60 samples)
HNN: Unsupervised Clustering	4 errors	16 errors
HNN: Semi-supervised clustering (10%)	1 error	7 errors

Table 5. Parameters in HNN for semi-supervised clustering

MLP	1 hidden layer; 4 neurons in hidden layer
	4 neurons in output layer
	Exponential Sigmoid activation function, a=1
	Learning rate=0.1
	Iterations=10
ART2	Vigilance value R=0.1

4 Conclusions and Future Work

In this paper, we propose a new data clustering method. It is a combination of Multi-Layer Perceptron and Adaptive Resonance Theory 2. To testify the performance of our method, we have done a set of experiments on two known dataset: Iris and Balance Scale. Experiment results show that our proposed method surpasses most existing methods in the following two aspects:

- It provides better unsupervised clustering accuracy.
- It also supports semi-supervised clustering, which is crucial for those applications with a small amount of information available. Most existing clustering methods cannot be used for semi-supervised clustering.

Although we have developed this method and tested it one some known datasets, in the future, many issues should be considered. Two main issues are:

- In fact, no universal clustering methods exist. We should explore that which kind of data and applications our algorithm is more suitable for.
- Most parameters used in our method are fixed, such as learning rate, iterations number and vigilance value. We will consider how to make them dynamic and adaptive for different tasks.

Reference

[1] Rui Xu, Wunsch, D.: Survey of Clustering Algorithms. In IEEE Transaction on Neural Networks, Vol. 16, (2005) 645-678

[2] A.K.Murty M.N. and Flynn P.J.: Data Clustering: A Review, In ACM Computing Surveys, Vol. 21, (1999) 264-323

[3] Sugato Basu.: Semi-supervised Clustering with Limited Background Knowledge, In Proc. of the Ninth AAAI/SIGART Doctoral Consortium, (2004) 979-980

[4] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney: Semi-supervised Clustering by Seeding, In Proc. of the Nineteenth International Conference on Machine Learning (ICML), (2002) 19-26

[5] Nizar Grira, Michel Crucianu and Nozha Boujemma, Unsupervised and Semi-supervised Clustering: a Brief Survey, A Review of Machine Learning Techniques for Processing Multimedia Content, 2004, http://www-rocq.inria.fr/~crucianu/src/BriefSurveyClustering.pdf

[6] G.A. Carpenter and S.Crossberg: ART2: Self-organization of stable category recognition codes for analog input patters, In Appl. Opt., Vol. 26, (1987) 4919-4930

[7] Andrey Gavrilov, Young-Koo Lee and Sungyoung Lee.: Hybrid Neural Network Model Based on Multi-layer Perceptron and Adaptive Resonance Theory, In Proc. of International Symposium on Neural Networks 2006, (2006) 707-713

[8] http://www.ics.uci.edu/~mlearn/MLRepository.html.

[9] N. Pal, J. Bezdek, and E. Tsao.: Generalized Clustering Networks and Kohonen's Self-organizing Scheme, In IEEE Transaction on Neural Networks, Vol. 4, (1993) 549-557

[10] R. Hathaway and J. Bezdek.: Fuzzy C-Means Clustering of Incomplete Data, In IEEE Transaction on Systems, Man, Cybern, Vol. 31, (2001) 735-744

[11] B. Gabrys and A. Bargiela.: General Fuzzy Min-Max Neural Network for Clustering and Classification, In IEEE Transaction on Neural Networks, Vol. 11, (2000) 769-783

[12] M. Girolami.: Mercer Kernel Based Clustering in Feature Space, In IEEE Transaction on Neural Networks, Vol. 13, (2002) 780-784

[13] A. Ben-Hur, D. Hom, H. Siegelmann, and V. Vapnik.: Support Vector Clustering, In J. Of Machine Learning Research, Vol. 2, (2001) 125-137

[14] T. Eltoft and R. deFigueiredo.: A New Neural Network for Cluster Detection and Labeling, In IEEE Transaction on Neural Networks, Vol. 9, (1998) 1021-1035

[15] R. Mollineda and E. Vidal: A Relative Approach to Hierarchical Clustering, In Pattern Recognition and Applications, Frontiers in Artificial Intelligence and Applications, The Netherlands: IOS Press, (2000) 19-28

[16] A. Baraldi and E. Alpaydin: Constructive feedforward ART clustering Networks—Part I and II, In IEEE Transaction on Neural Networks, Vol. 13, (2002) 645-677