ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Performance Evaluation of Extreme Learning Machine with k-Means Techniques for Clustering

Gopinathan.S
Department of Computer Science
University of Madras
Chepauk, Chennai, India

Pandiyan.M
Department of Computer Science
University of Madras
Chepauk, Chennai, India

Thangavel.P
Department of Computer Science
University of Madras
Chepauk, Chennai, India

Abstract: This paper aims to analysis the performance of different activation functions and various number of hidden neurons applied to an integrated Extreme Learning Machine (ELM) k-Means algorithm for clustering. ELM is an emerging and effective learning algorithm which is used to project the data into higher dimensional feature space, then k-means techniques used for clustering in this feature space. In this work we applying three different activation functions such as sigmoidal, radial basis and triangular basis function in ELM feature space were examined. ELM k means with different activation functions and various number of hidden neurons where considered for this analysis and the performance were compared with traditional k means algorithm. The experimental results proves that ELM k-Means gives better accuracy then k-Means algorithms especially the sigmoidal activation function along with ELM k means returns high accuracy with best number of hidden neurons for all four data sets.

Keywords: Extreme Learning Machine (ELM), k-Means algorithm, radial basis activation function, triangular basis activation function, sigmoidal activation function.

I. INTRODUCTION

Data clustering is an inevitable task in most of the scientific and engineering stream. Clustering techniques are widely used in many application domains such as machine learning, data compression, image analysis, computer vision and so on. The goal of clustering is to assign a set of data objects on the basis of a similarity between the objects. In the past decades many clustering techniques were introduced by the researchers. Most widely used techniques are the k-Means algorithm [1]. k-Means fit effectively well in case of ellipsoidal or spherical data distribution that is linearly separable. The inabilities to cluster data that are subjective to arbitrary shapes make k-Means algorithm fail because clustering is nonlinear. To solve such problems, a nonlinear transformation is applied to the original data thus project to high dimensional feature space and then performing clustering. This lead to the introductions of new kernel based clustering. Scholkopf et al. [2] proposed a kernel principal component analysis that requires eigenvalues but problems include local minima and scalability Local minima was further solved by Tzortzis and Likas [3] which does not depend on cluster initialization and is based on a global kernel k-Means algorithm. The problem of scaling a large kernel matrix was studied by Chitta et al. [4]. To provide high efficiency Zhang et al. [5] integrates the goodness of kernel principal component analysis along with ant based clustering. Recently, ELM based clustering techniques were studied by the researchers because it approximates any continuous target function and also has the capability to classify any disjoint region. The benefits of the ELM and their enhanced performance to solve the classification and regression problems inferred that clustering in elm high dimensional space would also produce excellent results. ELM has the Universal approximation and classification capability, this proficiency results in feature space that is more convenient for clustering after mapping into high dimensional feature space. Qing He et al. [6] proposed a method to solve a clustering problem in elm feature space using k-Means algorithm which is convenient and simpler then the kernel based feature mapping technique because it requires less human intervention and has no necessity to adjust any parameters. Another approach for clustering in extreme learning machine was studied by Alshamiri et al. [7], which integrate Extreme learning machine along with k-Means algorithm. In this paper, we analyses the performance of extreme learning machine with k-Means algorithm using different activation functions for various hidden neurons. The ELM method is used to project the data into high dimensional feature space and k-Means algorithm uses the Euclidean distance to find similarity between the data objects and perform the clustering within the feature space.

The rest of this paper is organized as follows: Section 2 describes the basics of the Extreme Learning Machine (ELM). Section 3 introduces k-Means algorithm. In Sect. 4, we evaluate the different activation function ELM k-Means algorithm. In Section 5 shows the performance evaluation result. Finally, Section 6 concludes this paper.

II. EXTREME LEARNING MACHINE

Extreme Learning Machine aimed to speed the process of learning in a single layer feed forward network. Traditional algorithm requires training the parameters such as weights and bias of all the layers which was a heuristic task and conventional slow in gradient based learning algorithms such as back propagation algorithms (BP). Thus an ELM for single hidden layer feed forward neural networks (SLFNs) was

intended to randomly choose weights of the input and hidden layer, bias and also analytically determine the output weights. An extremely fast learning speed with minor training errors proves the performance of ELM over the conventional methods. It also obtains the smallest norm of weights and is widely used for solving problems based on classification and regression. For formal definition of ELM, we will follow the same notational convention as used in [9]. For instance, we are given a set of training examples.

$Z = \{(x_j, t_j) \mid x_j \in \Re^d, t_j \in \Re^m, j = 1,2,...,N\}$, Single Layer Feed Forward Networks (SLFNs) with K hidden neurons and activation function G(x), can be represented as [9]:

$$\sum_{i=1}^{K} \lambda_i G(w_i . x_j + b_j) = O_j,\ j = 1,2,\ldots, N. \quad (1)$$

Where, $\lambda_i = [\lambda_{i1},...,\lambda_{im}]^T$ is the weight vector connecting the i^{th} hidden node and the output nodes, $W_i = [w_{i1},\ldots,w_{id}]^T$ is the weight vector connecting the i^{th} hidden node and the input nodes, and b_i is the bias of the i^{th} hidden node. $w_i.x_j$ is the inner product of w_i and x_j. Parameters λ_i, i =1,...,K can be estimated such that,

$$\sum_{i=1}^{K} \lambda_i G(w_i . x_j + b_j) = t_j,\ j = 1,2,\ldots, N. \quad (2)$$

Equation (2) can be written as in [9]:

$$H\lambda = T \quad (3)$$

Where,

$$H = \begin{bmatrix} G(w_1.x_1 + b_1)...G(w_K.x_1 + b_K) \\ \vdots \quad \cdots \vdots \\ G(w_1.x_N + b_1)...G(w_K.x_N + b_K) \end{bmatrix}_{N\times K} \quad (4)$$

$$\lambda = \begin{bmatrix} \lambda_1^T \\ \vdots \\ \lambda_K^T \end{bmatrix}_{K\times m} \text{ and } T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N\times m} \quad (5)$$

H is called the hidden layer output matrix of the neural network [10].The ELM algorithm consists of three steps and can be summarized as follows [11]:

Algorithm 1. ELM Algorithm:

A training dataset with N tubles $Z = \{(x_j, t_j) \mid x_j \in \Re^d, t_j \in \Re^m, j = 1,2,...,N\}$, Activation function G(x), Hidden node number K.

Steps:

1. Randomly generate input weight W_i and bias b_i, i=1,…, K.
2. Compute the hidden layer output matrix H
3. Calculate the output weight $\lambda : \lambda = H^+ T$,where H^+ is the Moore-Penrose generalized inverse [12] Of the hidden layer output matrix H and T = $[t_i,\cdots,t_N]^T$.

Theoretically, the number of neurons in the hidden layer of the ELM should be large enough to achieve good generalization performance. A detailed discussion on hidden nodes selection in particular and ELM in general can be found in [9-11].

III. K-MEANS ALGORIM

k-Means is a traditional clustering unsupervised learning algorithm. k-Means performs clustering by partitions a group of data into k number of clusters. The k-Means algorithm consists of two phases. In the first step it calculates the k centroid and in the second step it takes each point of the cluster which has the most nearest centroid from the corresponding data point. There are different methods to define the distance of the nearest centroid and one of the most used methods is Euclidean distance. Once the grouping is done it recalculates the new centroid of each cluster and based on that centroid, a new Euclidean distance is calculated between each center and each data point and assigns the points in the cluster which has the minimum Euclidean distance. Each cluster in the partition is defined by its member objects and by its centroid. The centroid for each cluster is the point to which the sum of distances from all the objects in that cluster is minimized. k-Means starts its iteration by initializing the centers randomly. At every iteration, each pattern is assigned to its closest cluster, based on the distance between the pattern and the cluster center. The cluster centers in the next iteration are determined by computing the mean value of the patterns for each cluster. The algorithm terminates when there is no reassignment of any pattern from one cluster to another.

k-Means algorithm is as follows [7]:

1. Initialize the number of k cluster centers.
2. Assign each pattern to its nearest cluster.
3. Compute the new cluster centers (6).
4. Repeat steps 2 and 3 until there is no change for each cluster.

k-Means algorithm designed to solve clustering is as follows, Given N patterns that is intended to find M clusters based on a measure of similarity such that the patterns within a cluster have a high level of similarity with each other than patterns belonging to different clusters. The mean of cluster c_m is defined as μ_m ,

$$\mu_m = \frac{1}{N_m} \sum_{p_i \in c_m} P_i \quad (6)$$

N_m is the number of patterns in cluster c_m

The squared error between μ_m and the patterns in cluster c_m is defined as in [8]:

$$E(c_m) = \sum_{p_i \in C_m} \|P_i - \mu_m\|^2 \quad (7)$$

The main objective of k-means algorithm is to minimize the sum of the squared error over M clusters,

$$E(c) = \sum_{m=1}^{M} \sum_{p_i \in c_m} \|P_i - \mu_m\|^2 \quad (8)$$

IV. ELM K-MEANS ALGORITHM

This section describes an approach that combines both the ELM and k-Means algorithms [15]. ELM performs a nonlinear transformation of the input data into a high-dimensional feature space. That is more advantages over the nonlinear data transformation thus increases the possibility of linear separability among the patterns. The integration of ELM with k-means algorithm helps to formulate inherent data structure in the new space. In ELM, the hidden layer maps the data from the input space R^J to the high-dimensional feature space $R^Q(Q >> J)$ where clustering to data is performed. After transforming the data into the ELM feature space, the traditional clustering method that is k-Means algorithms can be used directly in ELM feature space as we called ELM k-Means algorithm.

The ELM k-Means algorithm consists of seven steps and can be summarized follows [6],

Algorithm 1. ELM k-Means algorithm.

Input:

k: The number of clusters,

L: The number of the hidden-layer nodes,

D: A data set containing m objects.

Output:

A set of k clusters.

Method:

1: Mapping the original data objects in D into the ELM feature space H using $h(X) = [h_1(x),\ldots,h_i(x),\ldots,h_L(x)]^T$.

2: Arbitrarily choose k objects from H as the initial cluster center.

3: repeat

4: Assign / Reassign each object to the cluster to which the object that is the most similar, based on the mean value of the objects in the cluster.

5: Update the cluster means, i.e., calculate the mean of the objects for each cluster.

6: until no change in the cluster centers or reached the maximal iteration number limit.

7: return k set of clusters.

In this study we use the three different activation functions like sigmoidal, radial basis and Triangular basis activation function in the elm feature mapping of the hidden nodes (H).

V. EXPRIMENTAL RESULT

In this paper, four benchmark data sets are used to analyze the performance of the ELM k-means algorithm. These data sets can be downloaded from the UCI Machine Learning Repository. Table I include the following information, data sets, number of patterns, number of attribute features and number of classes.

Table I. Data set Details

Sr. No.	*Dataset*	*patterns*	*Attributes*	*classes*
1	Iris	150	4	3
2	Wine	178	13	3
3	Glass	214	9	6
4	Dermatology	358	34	6

A. Dataset

The data sets considered in this work can be described briefly as follows. Fisher Iris data is the most popular data sets to test the performance of novel methods in pattern recognition and machine learning. There are three classes in the data set (Setosa, Versicolor and Virginica), each having 50 patterns with four features (sepal length, sepal width, petal length and petal width). Wine data set consist of 178 patterns of 13 attributes and 3 classes. The glass data type consists of 214 patterns of 9 attributes and 6 glass types. The glass types include float processed building windows, non-float processed building windows, vehicle windows, containers, tableware and head lamps. Dermatology data set aims to determine the type of Eryhemato-Squamous Disease. It contains 366 patterns. After the removal of missing values, the data set consists of 358 patterns and 34 features belonging to 6 different classes

B. Results and Discussion

For each data set, we report the Accuracy of clustering as follows,

$$Accuracy = \frac{\sum_{i=1}^{N} \eta(o_i, t_i)}{N} \times 100$$

Where N is the number of instances of a dataset, o_i and t_i are real category labels and predicted label respectively. o_i and t_i were compared. If the prediction is true t_i is set to 1 otherwise set to 0. In this work the number of hidden neuron is selected based on the highest mean accuracy using trial and error method. For experimental comparison of k- Means and ELM k-Means, both the algorithm iterates for an average of 20 times. The following tables (II-V) depicts the performance of these algorithms with different cluster numbers for each data set and their accuracy values for different activation function like Sigmoidal Function(SF), Radial Basis Function(RBF) and Triangular Basis Function(TBF). For the four data set considered, different number of hidden neuron s were chosen. A high accuracy is reached when the number of hidden neuron is 280 for iris data set, 460 for Dermatology data set, 290 for Glass dataset and Wine data set having the highest accuracy at 220. Fig [1] and Fig [2] depicts the accuracy of Wine and Iris dataset with various hidden neurons.

C. Figures and Tables

Table II. Iris data set

No.of.Cluster	*k-Means*	*ELM k-Means with SF*	*ELM k-Means with RBF*	*ELM k-Means with TBF*
3	80.7000	**92.9067**	79.5333	83.9333
4	71.6000	79.9000	76.5333	`76.0331
5	61.6333	70.1667	71.6667	68.6000
6	53.2667	64.2333	63.0667	62.3000

Table III. Dermatology data set

No.of.Cluster	*k-Means*	*ELM k-Means with SF*	*ELM k-Means with RBF*	*ELM k-Means with TBF*
3	49.4382	**64.1210**	51.7697	46.4045
4	49.0169	55.9381	47.9775	42.3034
5	43.6798	50.4494	41.7416	39.1292
6	40.7584	47.4438	37.8933	37.6685

Table IV. Wine data set

No.of.Cluster	*k-Means*	*ELM k-Means with SF*	*ELM k-Means with RBF*	*ELM k-Means with TBF*
3	26.2430	47.6397	42.6816	36.6899
4	28.2682	58.2263	49.8603	38.1564
5	29.0223	57.7933	50.6704	41.6201
6	29.9860	**60.0698**	56.3966	40.3771
7	29.9302	59.2039	53.0587	40.8380

Table V. Glass data set

No.of.Cluster	*k-Means*	*ELM k-Means with SF*	*ELM k-Means with RBF*	*ELM k-Means with TBF*
3	48.9252	49.7196	49.9766	49.0654
4	51.7290	52.2196	50.7944	49.5794
5	51.7991	52.6168	50.0234	48.0841
6	51.6355	**52.7103**	49.9766	47.8505
7	50.0701	52.5701	47.3131	46.2617

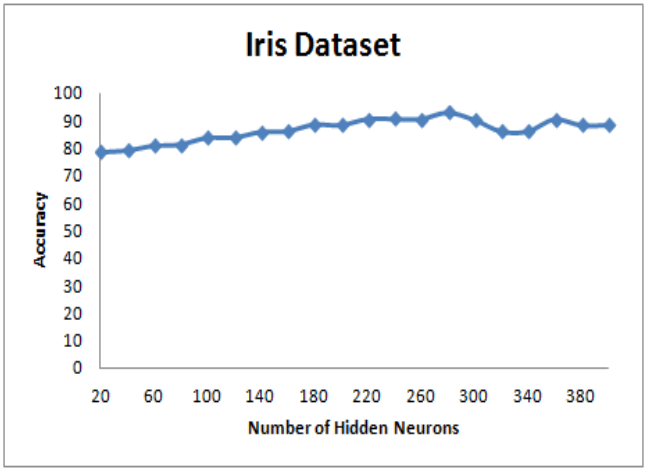

Fig I. Iris dataset clustering accuracy for various hidden neurons

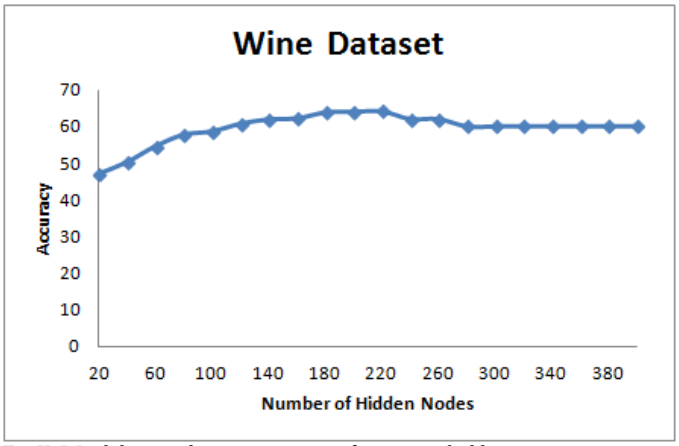

Fig II. Wind dataset clustering accuracy for various hidden neurons

VI. CONCLUSION

In the research paper we study the problem of clustering using the integration of ELM and k-Means algorithm with three different activation functions like sigmoidal, radial basis and triangular basis and various hidden neurons are applied to four different datasets. It is found after testing the activation functions on various datasets, sigmoidal function aims to produce better accuracy when compared with the other activation functions. Further, we intend to combine ELM Clustering feature with new clustering techniques yielding higher accuracy in the clustering problem.

VII. REFERENCES

[1] Ng, M.K.: A note on constrained K-means algorithms. Pattern Recognit. 33, 515– 519 , 2000

[2] Scholkopf, B., Smola, A., Muller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. 10(5), 1299–1319 , 1998

[3] Tzortzis, G.F., Likas, A.C.: The global kernel K-means algorithm for clustering in feature space. IEEE Trans. Neural Networks 20(7), 1181–1194 , 2009

[4] Chitta, R.,Jin, R.,Havens, T.C.,Jain, A.K.: Approximate kernel k-Means: solution to large scale kernel clustering. In: Proceedings of 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), New York, USA, PP.895-903 , 2011

[5] Zhang, L., Cao, Q.: A novel ant-based clustering algorithm using the kernel method. Inf. Sci. 181, 4658–4672 , 2011

[6] Qing He., Xin Jin.,Changyingu Du.,Fuzhen Zhuang.,Zhongzhi Shi.: Clustering in extreme learning machine feature space Neurocomputing 128, 88-95 , 2014

[7] A.K.Alsharmiri., B.R. Surampudi., Alok Sing.:A Novel ELM K-Means Algorithm for Clustering. Springer SEMCCO 2014, LNCS 8947, 212-222 , 2015

[8] Jain, A.K.: Data clustering: 50 years beyond K-means. Pattern Recognit. 31, 651– 666 , 2010

[9] Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing 70, 489–501 , 2006

[10] Huang, G.B.: Learning capability and storage capacity of two-hidden-layer feedforward networks. IEEE Trans. Neural Networks 14(2), 274–281 , 2003

[11] Lan, Y., Soh, Y.C., Huang, G.B.: Constructive hidden nodes selection of extreme learning machine for regression. Neurocomputing 73, 3191–3199 , 2010

[12] Serre, D.: Matrices: Theory and Applications. Springer-Verlag Inc., New York , 2002