Journal of Network and Innovative Computing.
ISSN 2160-2174 Volume 1 (2013) pp. 154 - 162

Efficient Mining of Web Access Patterns using Constrained Self-Organizing Map Clustering

Visakh R[1]

[1]Department of Computer Science and Engineering,
Rajagiri School of Engineering and Technology, Kochi-39, Kerala, India
visakhr@rajagiritech.ac.in

Abstract: **Web usage mining attempts to reveal interesting patterns of web access from a large number of web users. The main source of web usage data is web server logs. From an organizational point of view, web usage mining is very important as it assists in server management. The administrators of web servers can analyze web server log data to understand user behavior, allocate resources accordingly and provide customized service to similar groups of users. Clustering is a predominant data mining that partitions a group of unlabelled data instances into distinct groups or clusters. Several clustering techniques have been proposed in literature, which includes stand-alone as well as ensemble clustering techniques. Most of them lack robustness and cannot effectively visualize clustering results to help knowledge discovery and constructive learning. This paper explains the use of Self Organizing Maps (SOM) in a cluster ensemble framework based on some prior input constraints. Cluster ensemble is a set of clustering solutions obtained as a result of individual clustering on subsets of the original high-dimensional data. The final consensus matrix is fed to a neural network which transforms the input data to a lower-dimensional output map. The map clearly depicts the distribution of input data instances into clusters. The proposed method is tested on real web log data. Evaluation of clusters obtained justifies the superiority of the approach over conventional clustering techniques.**

Keywords: Data mining, Neural Networks, Self-organizing map, Web usage mining, CCEF-SOM, spectral clustering, constraint-based cluster ensemble.

I Introduction

Web usage mining refers to the discovery of interesting patterns of web access from a history of web usage data. Usually, web servers of an organization records the web activity of its users in a file called log file. Such log files contain immense volume of web access data over a typical period of say, one month. The website administrators perform a regular cleaning of such log files in order to limit the volume of data. Such log files will not be made available to the public in view of security concerns. Analyzing web server logs can reveal interesting patterns of usage from web users. The website management can have a study of such interestingness statistics to provide more efficient and customized web services to its users. Thus, web usage mining is a part of the organizational policies towards developing a very efficient resource management system.

However, most often, it is difficult to provide personalized web services to individual users. One reason is obviously the surprisingly large number of web users. An organization cannot afford to spend resources on a particular user only; it needs to mobilize its resource utilization on a variety of grounds. Another reason is the difference in interests of users. Each and every user is generating a unique request in one or the other aspect. As such, providing a customized service model is seemingly impossible. Also, as part of resource management, a web server may cache frequently requested web pages to reduce access latency. This again is based on certain intuitions that a particular page is likely to be accessed soon in future. But it may prove wrong. Another technique is prefetching [2], which tries to cache web pages based on access behavior of users. The choice is that of the web administrators. In the latter case, measures have to be taken to ensure that web page delivered to the user is the latest one.

Web usage mining from logs is not at all an easy task owing to the huge quantity of data available at the input level. For efficient mining of access patterns, two considerations are essential. One is the method chosen for knowledge extraction. A good idea is to track the navigation pattern of a variety of web users and then prepare a similarity matrix based on user interests [1]. The second and most important one is to ensure that good quality data are available at the input. Since the log files are voluminous, they may contain junk data which are not suitable for knowledge extraction. Therefore, web log files must be preprocessed thoroughly before applying any knowledge extraction technique.

The panacea to the resource allocation and cache coherence problem is to identify distinct clusters of users of the web. This has been an active area of research for the past decade. Many research works have been published that clusters web users based on dynamic behavior of usage. However, few of them considered the challenge of effectively clustering web log data which is multidimensional as well as heterogeneous. Clustering such data without knowing class labels can be really annoying, since relying on statistical measures blindly can yield no results. What is needed is a feature-based approach that incorporates semi-supervision into the clustering

task.
A Self-Organizing Map (SOM) is a neural network model to transform a set of high-dimensional continuous input data to a lower-dimensional discrete data representation. The most important feature of SOM is that while performing this dimensionality reduction, the topological ordering of data is preserved. A Kohonen SOM is a special kind of SOM in which there are a set of input neurons and output neurons. There is no hidden layer. The Kohonen SOM is an unsupervised training technique whereby, the network of neurons learns themselves without any supervision. The structure of a Kohonen SOM is depicted in Figure 1.

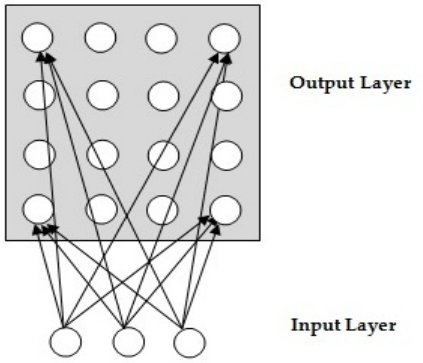


Figure. 1: A Kohonen SOM showing input and output neuron layers and their interconnections.

The neurons are interconnected by connections or links. The connections are assigned weights. Initially, the weights are randomly assigned small values. (See section III-F for a discussion on weight initialization procedure in neural networks.) For each input pattern, a neuron is selected as the winning neuron. It is triggered and thus it is powerful enough to decide the topological location of a set of neighboring neurons. The neighboring neurons compete to become the next winner. They adjust their connection weights and try to minimize an error function. This is how the map re-organizes itself.
The performance of Kohonen SOM in terms of clustering accuracy has been a subject of serious concern for the last few years. It has been observed that SOM leads to poor quality clusters owing to the neighborhood factor. The winning neuron influences the position of neighboring neurons and so this may cause cluster centers to differ from the actual ones, thus leading to non-intuitive clusters. The inherently complicated structure of neural networks provides no clue as to what is happening inside the black box.
To improve clustering accuracy, a new cluster ensemble framework for clustering web users is proposed. Here, prior knowledge of input data is fed to the cluster ensemble framework. The prior knowledge of a dataset is in the form of constraints about pairs of input vectors.
The original dataset is first partitioned into multiple subsets. Then, a set of clustering solutions is obtained by performing spectral clustering on each of the subsets of data. Next, confidence factors for each of these clustering solutions are calculated. After that, a consensus matrix is constructed by considering all the clustering solutions and their corresponding confidence factors. The consensus matrix is the input to the SOM input neuron layer. The proposed methodology introduces some sort of supervision to the clustering task in the form of input constraints. Thus the semi-supervised ensemble clustering using SOM can lead to better clustering accuracy compared to original Kohonen SOM algorithm.
The effectiveness of the proposed algorithm is tested on real-time multivariate datasets. The proposed algorithm clearly outperforms state-of-the art constrained clustering algorithms.
The remainder of the paper is organized as follows. Section II gives the related research works in this context. Section III gives the detailed methodology and design of the proposed constrained cluster ensemble framework using SOM to cluster web users. Section IV describes the experimental setup, results obtained and a comparative discussion. Section V concludes the work with a note on future scope of this work.

II Related Work

There has been numerous research works on web usage data analysis [2], [5], [6], [14], [20] as well as on constrained clustering from data mining systems [3], [7], [8], [9], [15], [16] and [21].
K. Rangarajan et al. [2] proposed a technique for clustering users based on their web access patterns. They used the Adaptive Resonance Theory (ART1) to find clusters of users that show a similar access behavior. The suggested method could depict the dynamic behavior of web usage. They developed a prototype vector for each cluster that generalizes the URL access behavior of all users in that cluster. The suggested technique was evaluated against conventional K-means and found to produce better results for intra-cluster distances. They also proposed a prefetching scheme that predicted future web requests by users.
Fazia, et al. [3] proposed the idea of semi-supervised SOM based clustering to improve the quality of clustering. In this method, background knowledge about the input domain is supplied to the clusterer along with input dataset. In the so called CrTM technique, the prior knowledge about dataset is supplied in the form of pair-wise constraints. Experiments with synthetic datasets revealed that the quality of clusters improved significantly even using a small number of constraints.
Singh A [5] proposed a novel idea to improve web proxy servers performance by integrating web caching and prefetching using sequential data mining techniques. The method used Pre-order linked position coded Web Access pattern (PLWAP) algorithm to find frequently accessed web objects of each user by analyzing the browsing history from the access log files and then compared the results over conventional page replacement algorithms such as Least Recently Used (LRU) and Least Frequently Used (LFU). The experimental results revealed that pre-fetching enhanced the performance of web proxy server.
Bina Kotiyal, et al. [6] proposed an intelligent technique for providing personalized web service more efficiently and effectively. Their main contribution was a technique to determine which web pages are more likely to be accessed by the user in future. The paper uses two intelligent algorithms for analyzing web user access patterns namely Apriori and Eclat.

The performance comparison of the two algorithms in terms of time and space complexity was also done.

Yu, Wong and Wang [7] investigated cluster ensembles with the prior knowledge. Their major contribution was a new cluster ensemble approach called knowledge based cluster ensemble (KCE) which included the prior knowledge of the datasets into the cluster ensemble framework. The experiments in real datasets showed that the prior knowledge was not harmful for the cluster ensemble approaches and also clustering quality of the cluster ensemble approaches in most of datasets could be improved if they considered the prior knowledge.

Strehl and Ghosh [8] introduced the problem of combining multiple partitioning of a group of data instances into a single consolidated clustering. The researchers evaluated the effectiveness of cluster ensembles in three qualitatively different application scenarios. Promising results were obtained in all three situations for synthetic as well as real data-sets.

Fern and Brodley [9] gave a detailed study on random projection for clustering high-dimensional data. In this work, a single iteration of clustering consists of applying random projection to the high dimensional data and clustering the reduced data using the Expectation-Maximization (EM) algorithm. Experimental results on data sets showed that this ensemble approach achieved better clustering performance. The work also demonstrated that both the quality and the diversity of individual clustering solutions had strong impact on the resulting ensemble performance.

Chitraa et al. [14] presented a novel method for analyzing web usage logs based on an enhanced form of K-means clustering. Here, the number of clusters K and the initial set of centroids is chosen automatically and precisely. The algorithm effectively clustered web usage data and converged quickly compared to K-means. Clustering results obtained were stable and accurate.

Visakh [15] proposed a constraint based cluster ensemble approach called CCE-SOM using SOM neural network model. The proposed algorithm produced highly robust clusters of arbitrary shape using constraint based cluster ensemble. The CCE-SOM approach clearly outperformed existing techniques for constrained clustering in terms of purity of clusters obtained. Spectral clustering was used in forming clustering solutions for the random subsets of the dataset in the ensemble clustering framework. The SOM clustering produced a map of data instances revealing clusters of patterns present in the dataset. Evaluation on UCI datasets produced promising results.

Visakh and Lakshmipathi [16] proposed a constraint based cluster ensemble approach for analyzing outlier patterns with respect to the clusters. This approach produced high quality clusters of arbitrary shape using an ensemble of clustering approaches. Small clusters were determined and considered as outlier clusters. After detecting outliers, each outlier is given a membership value (ranges from 0 to 1) to a cluster. A mapping function was used to associate each outlier to a cluster (from which it outlies). The proposed method successfully revealed interesting outlier-cluster relationships.

Alzennyr et al. [20] proposed an evolutionary approach to cluster web usage data based on time sub-periods. They used the idea of identifying long navigations as a means for preprocessing web log data. Then, they used a divide and conquer strategy to split the usage data into several independent time frames. They adopted a dynamic clustering algorithm to cluster users into different classes. The changes in clusters of users over time were noted and analyzed.

Wagstaff, et al. [21] attempted to modify the traditional K-means algorithm by integrating apriori knowledge in the form of constraints. They proposed the COP-Kmeans algorithm which tried to create data partitions that satisfy all the constraints and at the same time, reduce the vector quantization error.

III Constrained Cluster Ensemble Using Self Organizing Map for Web Log Analysis

This work proposes a constrained cluster ensemble framework using SOM to effectively cluster web usage data (CCEF-SOM). The constrained cluster ensemble approach incorporates the prior domain knowledge of the datasets into the cluster ensemble framework. The prior knowledge of web usage dataset is fed in to the cluster ensemble framework in the form of pair-wise constraints. The original dataset is partitioned into a set of random subspaces. Then, a set of individual clustering solutions is obtained by performing spectral clustering on each of these subspaces. Next, confidence factors for each of these clustering solutions are calculated. After that, a consensus matrix is formed by considering all the clustering solutions and their corresponding confidence factors. The consensus matrix is then clustered using SOM technique to form the actual clustering results of the dataset. The overall framework of the proposed method is shown in Figure 2.

The CCEF-SOM technique proceeds in the following manner:

A. Collecting data from Web Server Log

B. Preprocessing

C. Generating Random Subspaces

D. Forming Initial Clusters

E. Consensus Matrix

F. Self Organizing Map Clustering

III-A Collecting data from Web Server Log

Web servers keep a log of the information about web usage history. Such log files can be analyzed to gain an understanding about the access behavior of diverse users and their interests. W3C has a defined a format for maintaining information in log files. For conducting experiments, NASAs Web server log dataset [13] has been used. The log data is of the format given below:

<hostname, timestamp, URL requested, HTTP code returned, bytes in the reply>

The usage history from 1st July, 1995 to the midnight of 6th July, 1995 has been selected for experimental analysis.

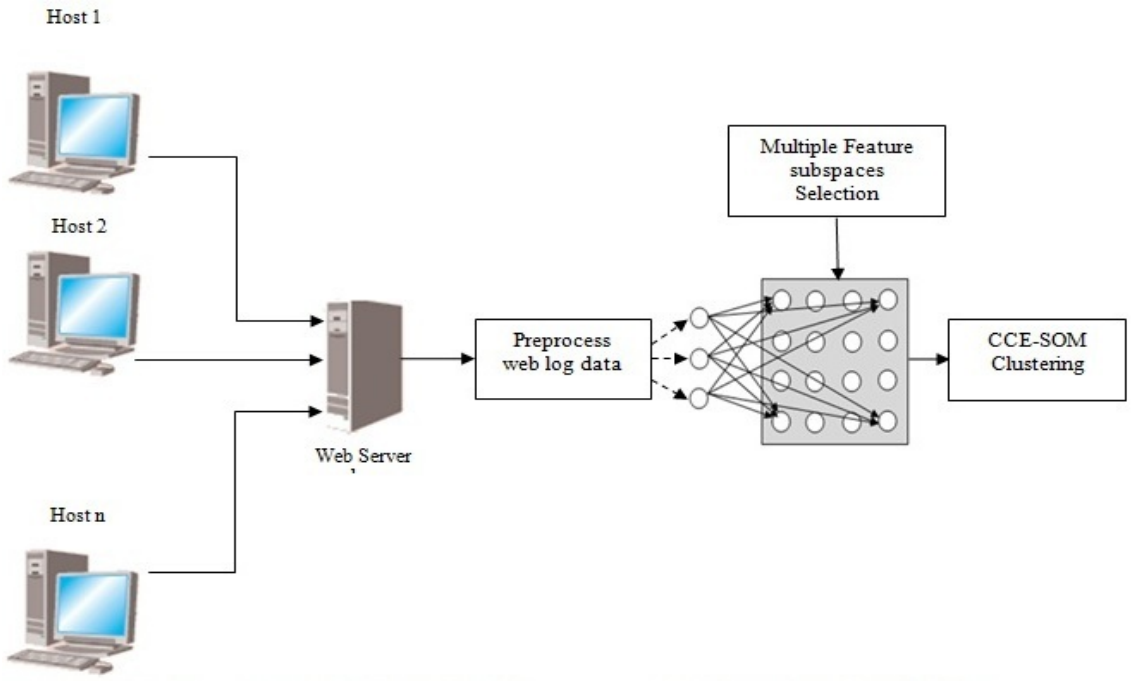


Figure. 2: General architectural framework for the proposed CCEF-SOM for clustering web users.

III-B Preprocessing

Web server data log consists of enormously large volume of data in it. Using raw data for analyzing patterns is extremely cumbersome and error-prone. Therefore, data cleaning is performed on log data to make it suitable for analysis.
The usage history from 1st July, 1995 to the midnight of 6th July, 1995 has been selected for experimental analysis. Requests are analyzed with respect to base URL. Requests for individual objects such as image files, video clips etc. are counted as base URLs only. Finally, a certain percentage of users that show significant usage frequency are selected for analysis. Similarly, a threshold is set for URLs also. URLs that are most accessed are preferred. From the preprocessed log data, it is possible to represent each user by means of an M-dimensional feature vector:

$$H_1 = \{URL_1, URL_2, ..., URL_M\}$$
$$H_2 = \{URL_1, URL_2, ..., URL_M\}$$
$$\vdots$$
$$H_N = \{URL_1, URL_2, ..., URL_M\}$$

where each dimension represents a URL accessed by that user.

III-C Generating Random Subspaces

The main idea of the proposed technique is to first generate multiple partitions by projecting the data on to a random space. The projected data is clustered using efficient spectral clustering algorithm [18]. Assume that the dataset H consists of a set of data instances $\{H_1, H_2, ..., H_n\}$ where n is the number of data instances), the i-th subspace is generated by randomly selecting d^i dimensions from d dimensions of the data. The above process is repeated K times and K datasets $\{H^1, H^2, ..., H^K\}$ are generated.

III-D Forming Initial Clusters

Spectral clustering (SC) is applied to generate clustering solutions for the multiple partitions. Given a dataset $H^k (k \in \{1, 2, ..., K\})$ with n data instances, spectral clustering clusters these data instances into K classes. SC constructs an affinity matrix F whose individual entry is defined as follows:

$$f_{ij} = E(x_i, x_j) \tag{1}$$

where $E(x_i, x_j)$ is the Euclidean distance between the samples x_i and x_j.
Next, it constructs a diagonal matrix R whose diagonal entry $r_{ii} (i \in \{1, 2, 3, ..., n\})$ is defined as follows:

$$r_{ii} = \sum_{j=1}^{n} f_{ij} \tag{2}$$

SC then normalizes the affinity matrix F by applying the following transformation:

$$T = F^{-\frac{1}{2}} R F^{-\frac{1}{2}} \tag{3}$$

Then, it selects the first l largest eigenvectors of T, and obtains an $n \times l$ matrix Z. It further re-normalizes the rows of Z and obtains a normalized matrix X:

$$X_{ij} = \frac{Z_{ij}}{(\sum_j Z_{ij}^2)^{\frac{1}{2}}} \tag{4}$$

Each row of X is treated as a new data instance $x_i^{'} (i \in \{1, 2, 3, ..., n\})$, and applies K-means to cluster these new data instances into K classes. If the instance $x_i^{'}$ is assigned to the class C, the corresponding original data instance x_i is assigned to the class C as well. SC is repeatedly applied to the K datasets to obtain K clustering solutions $\{I^1, I^2, ..., I^K\}$. Each clustering solution partitions the dataset H^k into K disjoint classes.

III-E Consensus Matrix

For each clustering solution, an $n \times K$ membership matrix M^k is constructed whose entry is defined as follows:

$$M_{ik}^k = 1 \text{ if } x_i \in I^k,\ 0 \text{ otherwise} \tag{5}$$

where n is the number of data instances, and k is the number of classes.
The constrained cluster ensemble algorithm represents domain knowledge of input datasets in the form of constraints. There are two sets of constraints [21]:

- **Must-link constraints:** Each pair of instances is considered similar and should be clustered into the same cluster.
- **Cannot-link constraints:** Each pair of instances is considered dissimilar and they cannot be clustered into the same clusters.

After clusters are obtained by constrained cluster ensemble, a consensus matrix is formed by considering the clustering solutions as well as their confidence factors. This consensus matrix is fed to the input set of neurons that forms a part of the SOM clustering. Here, the confidence factors along with the input data provide a form of supervision to cluster the data. Specifically, each clustering solution provides a cluster label that acts as a new feature describing each data instance, which helps to obtain the final clustering solution.

III-F Self Organizing Map Clustering

The Self Organizing Map (SOM) consists of a set of input neurons and a set of output neurons. All the input nodes will be connected to all the output nodes. Each connection has a weight associated with it. Usually, weights are initialized with randomly selected small values in the range $(0, 1)$. However, a thumb rule is to consider the individual fan-in of neurons in weight initialization [17], where fan-in refers to the number of connections incoming to a particular neuron. The weights are initialized in the range as given in (6).

$$(-\frac{2.4}{f_i}, \frac{2.4}{f_i}) \tag{6}$$

where f_i is the fan-in of the i-th neuron.
The output nodes form a two-dimensional fully connected network. The number of output nodes in the map determines the robustness and accuracy of SOM. During training, the output nodes in the map reorganize itself such that the data instances in the input that are closer are mapped onto nearby locations. This property of SOM serves it to be used as a very efficient tool for data clustering.
On each learning epoch, a training input sample x is selected randomly and the corresponding winning output neuron win is selected by using (7).

$$win = min\{|w_n - x|\} \tag{7}$$

where w_n represents the weight matrix of the n-th node.
Next, the neighboring nodes of the winning neuron try to win in the next training epoch. This leads to a form of competitive learning. This is shown in Figure 3. The weight matrices of the neighboring nodes are updated by using (8).

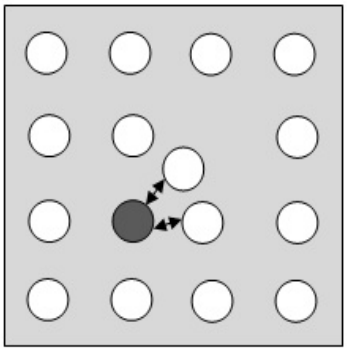


Figure. 3: The winning neuron (shown darkened) tries to pull its neighbors towards itself as a result of updating weights according to (8).

$$w_k(new) = w_k(old) + \alpha\Delta(win, k)(w_n - x) \tag{8}$$

where $w_k(new)$ is the new weight of the kth node, $\Delta(win, k)$ is a convergence function that determines the degree of convergence of neighbors of the winning neuron towards the latter and α is the learning rate that can be decreased during the subsequent training phases.
Learning rate during the beginning of training phase usually varies from 0.3 to 0.5 which can be decayed subsequently. This will improve performance of SOM training. The convergence function $\Delta(win, k)$ can be chosen as given in (9).

$$\Delta(win, k) = e^{-\frac{(|r^k - r^{win}|)^2}{2\sigma^2}} \tag{9}$$

where r^k and r^{win} are the position vectors of the neighboring and winning neuron respectively, and σ is a width parameter that decreases over time.

IV Experiment and Results

IV-A Evaluation Metrics

For evaluating the performance of the proposed CCE-SOM approach, four external criteria, namely, Rand Index, Precision, Recall and F-measure [11] are used. The different metrics used are introduced in this section. Rand index shows the pair-wise agreement between sets of data instances in the cluster set K and the label set C. Rand index is calculated using (10).

$$R = \frac{a + d}{a + b + c + d} \tag{10}$$

where a is the number of pairs which have same label in C and same cluster in K, b is the number of pairs having same label in C but are clustered differently, c is the number of pairs that have different class labels but are clustered into same K, d is the number of pairs that have different labels and are clustered separately.
Rand index takes the values from $0 \leq R \leq 1$.
Values close to 1 indicate excellent clustering while those close to 0 shows very poor correlation between clusters found and the class labels.

Purity is the simplest way of expressing clustering quality. To compute purity, each cluster is assigned to a class that is predominant in that cluster. Then this assignment is validated against (11).

$$Purity(K,C) = \frac{1}{N}(\sum_{k} max_j|k_i \cup c_j|) \qquad (11)$$

where $K = k_1, k_2, k_3, , k_n$ is the set of clusters obtained, $C = c_1, c_2, c_3, , c_m$ is the set of class labels.
Precision is the percentage of correct cluster assignments.

$$P = \frac{TP}{TP + FP} \qquad (12)$$

where TP and FP represents the true positive rate and false positive rate respectively. True positive (TP) decision assigns two similar data instances to the same cluster. False positive (FP) decision assigns two dissimilar instances to the same cluster.
Recall is the percentage of correct cluster assignments which were actually obtained by the clusterer.

$$R = \frac{TP}{TP + FN} \qquad (13)$$

where FN represents the false negative rate. A false negative (FN) decision assigns two similar data instances to different clusters.
Finally, the $F\text{-}measure$ combines both precision and recall to a single measure. It is used to give additional weight to recall.

$$F = \frac{(\alpha^2 + 1)P.R}{\alpha^2.P + R} \qquad (14)$$

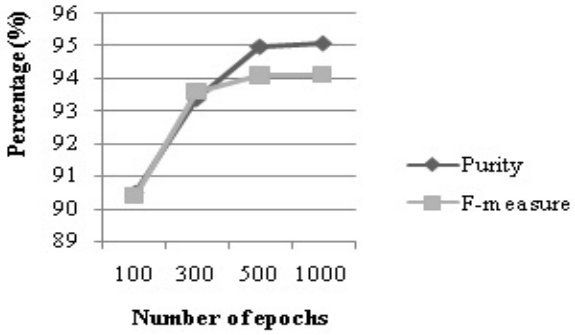
where α is the parameter used to give weight to recall. If $\alpha > 1$, then false negatives will be penalized more than false positives.
For evaluation of the clusters obtained, a technique known as Generic Access Pattern Vector (GAPV) is introduced. Corresponding to each cluster of users, a generalized vector is found out that clearly depicts the access pattern of all users belonging to that cluster. When a test dataset of user access log is given for clustering, the GAPV is found out for each cluster and then the GAPV to which the test vector best corresponds to is selected as its label. This is compared against the actual cluster assignment obtained as a result of SOM clustering. For finding the most similar GAPV corresponding to a test vector, the Manhattan distance or City Block measure is used. This is most appropriate for web log data. Manhattan distance finds the closest GAPV corresponding to a test vector by computing sum of absolute differences of the attribute values as given in (15).

$$CB(H_i, G_n) = \sum_{i=1}^{M} |URL_{i,k} - URL_{G_n,k}| \qquad (15)$$

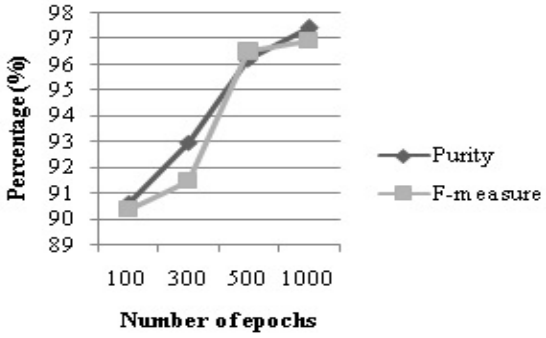
where $CB(H_i, G_n)$ is the City Block distance between host H_i and $GAPV$ for each cluster, $URL_{i,k}$ and $URL_{G_n,k}$ are the k-th attribute values for the i-th host and $GAPV$ for that cluster respectively. For calculating Rand index, pairs of users are chosen and their actual cluster assignments are compared against their labels. Number of matches and mismatches are noted and evaluated against (10).

IV-B Experimental Setup

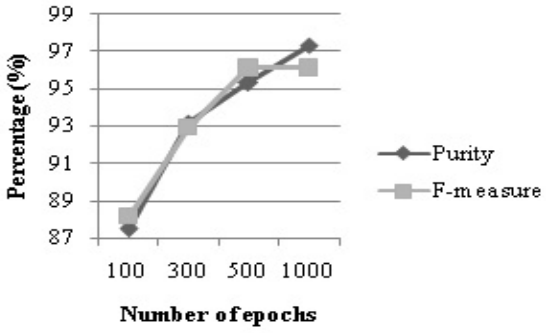
The performance of the proposed algorithm is first tested using three popular UCI datasets [12]: Tic-tac-toe, Mushroom and Iris. The Tic-tac-toe dataset consists of 958 instances. Total number of attributes is 9. For conducting the experiment, 100 randomly selected instances were used. The class attribute is nominal and it represents two values: positive or negative. The CCE-SOM algorithm gave 97.4% purity in clustering this data. The precision and recall rates are 97.5% and 97.4% respectively. The F-measure is 96.9%. The evaluation on Tic-tac-toe dataset is illustrated in Figure 4(a).



(a) Tic-tac-toe



(b) Mushroom



(c) Iris

Figure. 4: Evaluation of CCEF-SOM on various datasets.

The next set of experiments was conducted on Mushroom dataset. The Mushroom dataset consists of 8124 instances out of which 100 instances were randomly chosen for the experiment. The number of attributes is 22. All are nominally valued. The whole data belongs to two classes: edible or

Table 1: Results of evaluation of CCEF-SOM on Tic-tac-toe, Mushroom and Iris datasets.

Evaluation Criteria	Tic-tac-toe	Mushroom	Iris
Purity	97.4%	95.1%	97.3%
Precision	97.5%	93.6%	94.9%
Recall	97.4%	95.1%	97.3%
F-measure	96.9%	94.1%	96.1%

Table 2: Preprocessed NASA web server log dataset details.

Item	Count
Total number of log entries	64716
Log entries after preprocessing	64000
Significant users identified	100
Relevant URLs identified	35

poisonous. The CCEF-SOM algorithm gave 95.1% purity in clustering the mushroom data. The precision and recall rates are 93.6% and 95.1% respectively. The F-measure obtained is 94.1%. The evaluation on Mushroom dataset is illustrated in Figure 4(b).

Next, the performance of CCEF-SOM is evaluated on the well-known Iris dataset. Iris is a multivariate dataset that consists of 150 instances and 4 predictive, numeric attributes. The class attribute is nominal and may be any of the following: Iris-setosa, Iris versicolour or Iris-virginica. The CCEF-SOM algorithm gave 97.3% purity in clustering this data. The precision and recall rates are 94.9% and 97.3% respectively. The F-measure is 96%. The results are shown in Figure 4(c).

The results of evaluation of the CCEF-SOM on all three UCI datasets are summarized in Table 1. In order to cluster web users based on their access behavior, experiments were conducted on real web server log data collected from the NASA Kennedy Space Center web server [13] located in Florida.

The NASA-HTTP log consisted of web access details of various hosts from 1st July, 1995 to 31st July, 1995. Taking into account the huge volume of web log, experiment was restricted to analyze log of users during a time period of one week, starting from 1st July to 6th July midnight.

The preprocessed data consisted of 64000 entries, each corresponding to HTTP requests from web users. The details of log entries after preprocessing are listed in Table 2.

The overall clustering accuracy obtained was 96.2%. Precision and Recall rates obtained were 97.28% and 96.88% respectively. The Rand index value 0.96 indicates excellent clustering of web users.

V Comparison with existing approaches

The performance of the proposed CCEF-SOM algorithm is compared with existing constrained as well as fully unsupervised clustering algorithms.

The accuracy of clusters obtained for CCEF-SOM on the Tic-tac-toe dataset is 97.4%. The improvement obtained in clustering accuracy is significant when compared with CrTM [3], COP-Kmeans [21] and simple SOM clustering [4]. The clustering accuracies obtained are 96.7%, 92% and 84.2% for CrTM, COP-Kmeans and SOM respectively. The prior knowledge of datasets provided a sort of supervision to the clustering process and thereby, helped to minimize the error while making cluster assignments. The number of instances

Table 3: Performance comparison of CCEF-SOM with other approaches on the Tic-tac-toe dataset.

	CCEF-SOM	SOM	CrTM	COP-Kmeans
Clustering Accuracy	97.4%	84.2%	96.7%	92%

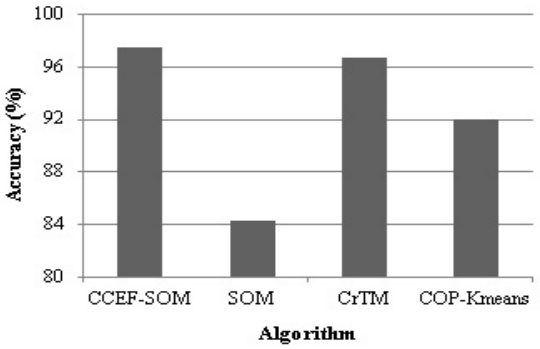
Table 4: Performance comparison of CCEF-SOM with other approaches on real NASA web log dataset.

CCEF-SOM	SOM	ART-1	EM	
Clustering Accuracy	96.2%	92.12%	92.3%	74.5%

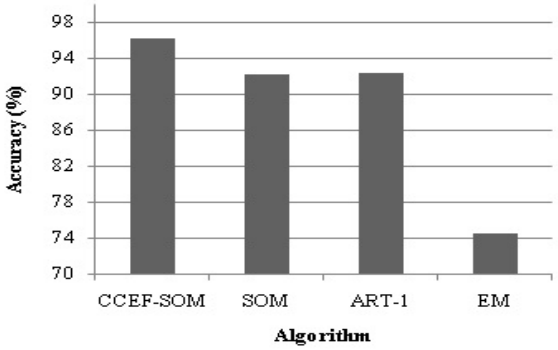
in the dataset and the class distribution affects the clustering to a significant effect. For example, the evaluation of CCEF-SOM clustering on Mushroom dataset yielded only 95.1% accuracy, while COP-Kmeans achieved a slightly higher accuracy of 96%.

The performance comparison of CCEF-SOM with other existing approaches on Tic-Tac-Toe dataset is illustrated in Figure 5(a) as well as summarized in Table 3.

For the real NASA web log dataset, a clustering accuracy of 96.2% was achieved. For SOM [4], ART-1 [3] and Expectation-Maximization (EM) algorithms, the accuracy achieved were 92.12%, 92.3% and 74.5% respectively. Figure 5(b) illustrates the obtained results on NASA web log dataset for CCEF-SOM, SOM, ART-1 and EM algorithms. The same has been tabulated in Table 4.



(a) Tic-tac-toe



(b) NASA web server log

Figure. 5: Performance comparison of CCEF-SOM with other approaches on various datasets.

The efficiency of CCEF-SOM clustering on NASA web log dataset is compared with plain SOM and the results obtained are summarized in Table 5.

Table 5: Results of evaluation of CCEF-SOM and SOM on NASA web log dataset.

Evaluation Criteria	CCEF-SOM	SOM
Rand Index	0.96	0.89
Precision	97.28%	94.66%
Recall	96.88%	95.5%
F-measure	97.06%	95.11%

Table 6: Results of evaluation of CCEF-SOM and SOM on UCI Syskill Webert dataset.

Evaluation Criteria	CCEF-SOM	SOM
Rand Index	0.98	0.98
Precision	97.6%	95.7%
Recall	96.85%	97.2%
F-measure	97.18%	96.52%

Figure 6 shows the Receiver Operating Characteristic curve (ROC) for NASA web dataset for CCEF-SOM and plain SOM. ROC is a plot of the true positive rate against false positive rate. The ROC curve shows a trade-off between sensitivity (or True positive rate) and specificity (or True negative rate). The more the curve is closer to the left border and the top border of the graph, the more accurate the test is. The area under ROC curve indicates the accuracy of clustering.

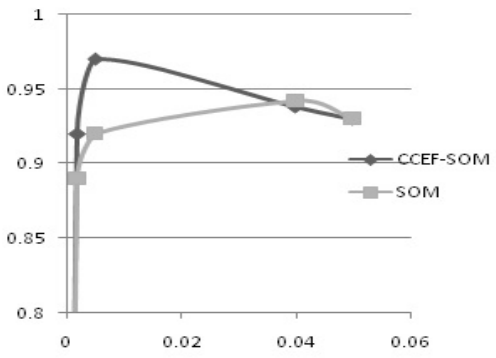


Figure. 6: ROC curves for CCEF-SOM and SOM algorithms for NASA web server log dataset.

Finally, a set of experiments was also conducted on the Syskill Webert dataset from UCI repository [19] to evaluate the precision, recall, F-measure and rand index for the proposed CCEF-SOM as well as conventional SOM clustering techniques. Syskill Webert is a multivariate, text dataset that consists of 332 instances. Each instance is characterized by a set of 5 categorical attributes.

CCEF-SOM gave a precision and recall of 97.6% and 96.85% respectively while, SOM gave 95.7% and 97.2% respectively. The reduction is recall for CCEF-SOM compared to SOM is justified owing to the categorical nature of the attributes of Syskill Webert dataset. F-measure obtained was 97.18% and 96.52% respectively for CCEF-SOM and SOM. The Rand index values were close to 1 for both CCEF-SOM and plain SOM. This indicates high correlation between clustered results and actual labels. The evaluation results are summarized in Table 6.

VI Conclusion

A constraint based cluster ensemble approach using Self-Organizing Map neural network model for analyzing web access patterns is proposed. The main contribution of this paper is a cluster ensemble framework that incorporates prior user-specific constraints into the SOM algorithm. The proposed algorithm clusters web users based on their diverse web access behavior. Even though there exists several web usage analysis tools that provide statistical data and textual information about web usage, they rarely impart any meaningful idea about the inherent dynamic behavior of web users. The suggested CCEF-SOM approach clearly outperforms existing techniques for constrained clustering in terms of precision and accuracy of clusters obtained. Spectral clustering is used in forming initial clustering solutions for the random subsets of the preprocessed web log dataset. Spectral clustering is selected because of its inherent ability to produce highly robust clusters for multidimensional data. The SOM clustering produces a map of web users revealing clusters of interesting patterns present in the dataset. The suggested model can be applied to find a solution to the cache coherence problem in web servers. In future, complex constraints revealing stochastic behavior of user access log information may be incorporated to extend the proposed approach.

References

[1] Zhen Rong, Yan Tang, and Su Liu, Research on Web Log Mining, *Proceedings of the International Conference on Information Engineering and Applications (IEA)*, pp. 849-856, Vol. 217, Springer, 2013.

[2] Santosh K. Rangarajan, Vir V. Phoha, Kiran S. Balagani, Rastko R. Selmic, S.S. Iyengar, *Adaptive Neural Network Clustering of Web Users*, IEEE Computer Society, 2004.

[3] Fazia Bellal, Khalid B and Alexander A.,SOM based clustering with instance-level constraints, *European Symposium on Artificial Neural Networks*, Advances in Computational Intelligence and Learning, Bruges (Belgium), April 2008, ISBN 2-930307-08-0.

[4] T. Kohonen, Self-Organizing Maps, Springer, Berlin, 1994.

[5] Singh A, Singh A.K., Web Pre-fetching at Proxy Server Using Sequential Data Mining, *Third International Conference on Computer and Communication Technology (ICCCT)* at Allahabad, India, Pages: 20 25, IEEE, Nov 2012.

[6] Bina Kotiyal, Ankit Kumar, Bhaskar Pant, R.H. Goudar, Shivali Chauhan, Sonam Junee, User behavior analysis in web log through comparative study of Eclat and Apriori, *7th International Conference on Intelligent Systems and Control (ISCO)*, Tamil Nadu, India, Pages: 421-426, IEEE, Jan 2013.

[7] Z. Yu, H. S. Wong, J. You, Q. Yang, H. Liao, Knowledge based Cluster Ensemble for Cancer Discovery from Bio-Molecular Data, *IEEE Transactions on Nanobioscience*, 2011.

[8] A. Strehl and J. Ghosh, Cluster Ensembles A Knowledge Reuse Framework for Combining Multiple Partitions, *Journal of Machine Learning Research 3*, pp.583-617, 2002.

[9] X.Z. Fern and C.E. Brodley, Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach, *Proc. 20th Intl Conf. Machine Learning*, pp.186-193, 2003.

[10] D. Greene, A. Tsymbal, N. Bolshakova, and P. Cunningham, Ensemble Clustering in Medical Diagnostics, *Technical Report TCD-CS- 2004-12*, Dept. of Computer Science, Trinity College, Dublin, Ireland, 2004.

[11] Han, J. and Kamber, M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann, Second edition, 2006.

[12] C. Blake, C. Merz, UCI repository of machine learning databases. *Technical Report, University of California*, 1998.

[13] Jim Dumoulin, NASA Kennedy Space Center, http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html.

[14] V Chitraa and Dr. Antony S.Thanamani, An Enhanced Clustering Technique for Web Usage Mining, *International Journal of Engineering Research and Technology*, Vol.1 Issue 4, June 2012.

[15] Visakh R., Using Self-Organizing Maps in Constrained Ensemble Clustering Framework, *Proceedings of the 12th International Conference on Intelligent Systems Design and Applications*, Pages 224-229, IEEE, Nov. 2012.

[16] Visakh R and Lakshmipathi B, Constraint based Cluster Ensemble to Detect Outliers in Medical Datasets, *International Journal of Computer Applications*, Volume-45, Number-15, May 2012. Foundation of Computer Science, New York, USA.

[17] Satish Kumar, *Neural Networks: A Classroom Approach*, Tata McGraw-Hill Publishing Company, Third reprint, 2007.

[18] A. Ng, M. Jordan, and Y. Weiss, *On spectral clustering: Analysis and an algorithm*, In Advances in Neural Information Processing Systems,14, pp. 849-856, 2001.

[19] Michael Pazzani, UCI repository of machine learning databases. *Technical Report, University of California*, 1998.

[20] Alzennyr Da Silva, Yves Lechevallier, Fabrice Rossi, Francisco De Carvahlo, *Clustering Dynamic Web Usage Data*, Innovative Applications in Data Mining, arXiv:1201.0963[stat.ML], Jan 2012.

[21] K. Wagstaff, C. Cardie, S. Rojers and S. Schroedl, Constrained K-means Clustering with Background Knowledge, *Eighteenth International Conference on Machine Learning*, 2001.

Author Biography

Visakh R received the B.Tech degree in Information Technology from Govt. Engineering College, Trivandrum, Kerala, India in the year 2005. He received the M.Tech degree in Computer Science and Engineering discipline from Anna University of Technology, Chennai, India in 2012. He is currently working as Assistant Professor in the Dept. of Computer Science and Engineering, Rajagiri School of Engineering and Technology, Kochi, India.

His current research interests include neural networks, data mining and knowledge discovery, pattern classification and machine intelligence.