ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol.12 (2022) No.1, pp.363-382
DOI:10.33168/JSMS.2022.0125

Applying Spectral Clustering Algorithm to Group Users by Interest

Chaitra H. K.[1], Suneetha K. R.[2]

[1] Department of Computer Science and Engineering, SJB Institute of Technology, India

[2] Department of Computer Science and Engineering, Bangalore Institute of Technology, India

hkchaitra82@gmail.com

Abstract. The use of the Internet as a medium of communication between businesses and consumers has increased in popularity. The website of a company must be designed to fulfill the specific needs of its clients. Users appreciate returning to websites that are simple to navigate. The usability of a website can be improved by observing and analyzing the habits of its visitors. The recommendation of the website to the users based on their interest is an important thing for growing the business. The grouping of users based on their interests is the initial stage of website recommendation. The user groping is taken for this research. The weblog is collected from Kannada University. The weblog is preprocessed. To achieve success, adequate data pre-processing techniques need to be devised. Data cleaning, user identification, and session identification are the pre-processing activities that were addressed in this research. The preprocessed weblog is used for identifying users with similar interests. The workflow is broken into three important features such as matrix calculation, graph building, and clustering. The three matrices are produced notably page, distance, and adjacency. By employing the user ID and page URL the page matrix is formed. The distance matrix is calculated using the Euclidean distance. The adjacency matrix is populated with the help of threshold values. Based on the adjacency matrix the graph is constructed. Then the clustering is accomplished using the spectral clustering algorithm. The Silhouette and Davies-Bouldin Index are applied to validate the clustering.

Keywords: Weblog, Coverage, Session, Matrix, Graph, Euclidean

1. Introduction

The growing interest in data mining studies is a direct result of the available vast data that are being generated in today's technologically driven society. In data mining, information may be extracted from enormous amounts of data (M. Spiliopoulou, 2001) through the use of algorithms. Currently, data mining programs are concentrating their efforts on web mining. It is possible to extract knowledge from web data in a variety of ways, including through papers, links between documents, and usage records from websites (Jaideep Srivastava et al 2005). Because of the enormity of the Internet's collection, it may be difficult for users to locate web data such as multimedia, and text documents when they are required by the website. Weblog files, which are text files that contain all of a user's interactions with online pages, are stored on a computer's hard drive. A plethora of information can be found on weblogs, including data that is incomplete or useless. Using web mining technology, which is a sort of data mining, you may eliminate repeated patterns from online sites while simultaneously looking for intriguing patterns. Web content mining, web structure mining, and web use mining are some of the approaches used in this procedure, to mention a few examples. The term "text mining" refers to the process of searching through millions of online pages to find information that is relevant to a search request. Web structure mining, when used in conjunction with database approaches for websites, can aid in the detection of associations between web pages and their structural modules of hyperlink connections, which is structure data, as well as the detection of associations between web pages and their structural modules of hyperlink connections (Magali et al 2014). When the web structure schema is provided by database approaches for websites, the use of web structure mining can aid in the detection of associations between web pages and their structural modules of hyperlink connections, which is structure data. When a user accesses a particular web resource, web servers produce and keep log files on the server's hard drive regularly. For example, in internet use mining, which is a sort of data mining, log data is investigated and mined for key trends that may be identified. A possible method is to employ principles like these in a variety of applications, ranging from website redesign to prefetching and caching to business intelligence (Facca, F. M.et al 2005). The detection of patterns in web usage and the assessment of such patterns are the two most important components of web usage mining, in my opinion. Before converting raw log data into a suitable format for use with a particular data mining method, it is important to prepare the data for use with the method in question. Data preparation in a server log covers a variety of tasks such as data cleansing, user identification, and session identification, to name a few. Web usage mining refers to the process of removing superfluous entries and attributes from a database, and the term "data cleaning" refers to this process. Users' activities are organized and stored in a user activity file during the user identification phase, which is the first step of the user identification

process. Because a user can return to a website multiple times, resulting in multiple sessions for the same user (Liu, H., et al 2007), users' activities are divided down into various sessions during the session identification process. Traditionally used data preparation procedures fail at scaling points as the volume of logs from well-known websites grows to terabyte and petabyte sizes every day. In addition to the Hadoop MapReduce and Spark frameworks, this article's evaluation of server log analytics approaches includes a look at other frameworks.

2. Literature Review

Bin Shen and colleagues conducted an analysis of preferred navigation patterns mining, in which they looked at both the route traveled and the time it took to traverse that route. The authors' results on relative path selection and preference for time-related path sequences have led them to develop the concepts of user interest and time preference, which they have named after themselves. There are two ways to describe a forest: PNP-Forest and Prefix. PNP-Forest is the more formal term. To identify patterns of travel, techniques such as span forest mining are employed. Identification of PNP is accomplished by the use of a two-step technique. The PNP-forest technique, which is now available, is used to insert and process frequently occurring navigation patterns generated by efficient frequent sequential pattern mining algorithms, which are generated by frequent sequential pattern mining algorithms. The proposed technique by the authors involves a great deal of unneeded and uncomfortable navigation (Shen B, et al 2016). An online and real-time classifier, suggested by Adeniyi et al., can be used to recognize clients /visitors clickstream data, match it with a certain user group, and recommend a customized browsing option based on this data at any given time to satisfy the unique user's requirement at that time. The K-Nearest Neighbor technique, which uses a combination of server logs and web content, is capable of classifying a user's browsing tendencies and anticipating their future requests Adeniyi D.A.,et al 2016). The fact that RuiliGeng developed a methodology that only evaluates user interaction levels means that it is unable to expose user experiences at the elementary level of the web page. Patterns in real-world usage paths are discovered through the use of a process known as use mining. The Ideal User Interactive Path (IUIP) technique can be used by website designers to determine both the paths and the amount of time required to execute user-oriented tasks on their sites. This tool bundle of modeling and analysis tools is designed to assist in the measurement, analysis, and general quality improvement of Web-based applications and services. Based on the deviation data provided by this comparison, researchers can discover usability issues and make recommendations for (Geng R.,et al 2015). According to Mobasher et al., the individuality of each user was discovered. All weblog entries are first processed into a treated log, which is then kept for further examination. Sequence patterns are subdivided into sub-sequences by employing a forward route

created by the traversal pattern to break them down. To form a cluster, a clustering algorithm reduces the number of pathways that are sufficiently similar to one another. The classification divides users' habits into those who are interested and those who are not.

It is feasible to overcome the problem of sparse data by utilizing Bias SVD to calculate user similarity in matrix decomposition, which may be used to determine user similarity (Da Sun,et al 2020). The author proposes a recommendation system based on the Bias SVD function. The program makes use of the FunkSVD algorithm to deconstruct the similarity matrix while also considering the user's preference. Kasap and Tunga et al (2017) proposed an HDMR-based method for webpage recommendation, which they call the HDMR-based method. This particular concept was developed to facilitate the operation of online stores. It was discovered that using the newly created approach in conjunction with a customer's purchase history matrix could improve the search process. The model created multivariate data by combining some different datasets. Individuals may find themselves unable to operate properly in a variety of situations as a result of this. The model had to deal with a lot of data as well. Zhang et al. (2017) provided a method for recommending a webpage in microblog applications, which was designed to recommend a webpage in microblog applications. To determine the relationship between the log file data and the user graph, the data from both were compared. The recommendation was influenced by the user graph. When it came to advertising the site on microblogs, the method proved to be effective. As an alternative to the link prediction scheme proposed by Sharif and Raghavan (2017), some hybrid feature sets created from the web database and used to recommend webpages were also investigated. The prediction technique used a graph structure-based feature set extracted from the pages to construct a hybrid recommendation scheme, which was then used to develop a hybrid recommendation scheme. We were able to propose web pages by utilizing supervised learning algorithms. By selecting the most appropriate weight values for integrating the feature set, the overall performance can be enhanced. To recommend appropriate URLs to the user, Li et al. (2017) created an entity similarity-based semantic network based on entity similarities. It was decided to develop a recommendation system that was primarily reliant on the depth and breadth of the content. The method was created to address the problem of unbalance that is prevalent in website recommendation systems. Class drift, on the other hand, is considered a flaw in the plan itself. Nguyen et al. (2014) advocated that a recommendation system includes multiple variables, such as online usage and domain expertise, to provide more accurate results. Generally speaking, domain names for websites can be represented in one of two ways. It was necessary to use the conceptual prediction model to construct the semantic network that would be used in the recommendation process, and the semantic network was automatically generated by applying the conceptual prediction model. This solution

avoids the occurrence of a new suggestion page in the existing schemes by working around the problem. There are three types of web mining techniques (Rajesh K Shukla et al, 2020), which are as follows: web usage mining, web structure mining, and web content mining. An investigation into online usage mining procedures, which includes pre-processing techniques, pattern finding tools, and a pattern analysis tool, is conducted in this research study. In addition, this study discusses some of the applications of internet usage mining. The datasets from MSNBC and CTI are utilized to test the web page prediction methods in this study. When compared to the results of the existing algorithms, the maximum frequency pattern algorithm provided by the authors beats current methods such as the K-means and the FCM (Om Prakash P.G, et al 2021).

3. Methodology

3.1. Proposed Work

In this research, the weblog was collected from Kannada University. The grouping of users based on their interests is the first stage of website recommendation. The user groping is taken for this research. The workflow is split into three main divisions such as matrix calculation, graph creation, and clustering. By using the user ID and page URL the page matrix is created. The distance matrix is calculated using the Euclidean distance. The adjacency matrix is filled with the help of threshold values. Based on the adjacency matrix the graph is created. Then the clustering is made using the spectral clustering algorithm (Nascimento MCV et al., 2011). The Silhouette (Peter J. Rousseeuw1987), and Davies-Bouldin Index (Davies et al., 1979) are employed to validate the clustering.Spectrum clustering can group "points" that are not necessarily vectors and employ for this a "similarity," which is less restrictive than a distance.

3.2. Weblog Collection and Preprocessing

Website owners are always interested in knowing more about the behavior of their site's users while they are using it. Web mining can be used to discover the patterns of end-user behavior. Because of the enormous volume of web documents, which may include text, photographs, and other multimedia files, it may be difficult to identify the precise information you are looking for on the World Wide Web due to the sheer volume of information available. When it comes to gathering information from the internet, web mining is important. There are numerous issues on web mining that must be addressed before any useful information can be extracted from the data. The data from Kannada University's online logs are being used in this investigation. A total of 442867 samples have been entered into the online log, which covers the period from October 31st, 2019 to November 30th, 2019. A log is depicted in Figure 1 for reference.

Fig. 1: Sample of weblog data

Reduce the number of entries in weblog data by removing and filtering out extraneous information to segregate only useful/vital data that may be employed to identify the user's navigational habits. In this research, the error status codes, bot, and supportive files are removed. The raw weblogs contain 442867 samples and 87.52% of the data is reduced. Finally, the total number of data ready for further processing is 55284. The statistics of the raw data and cleansed data are shown in figure 2.

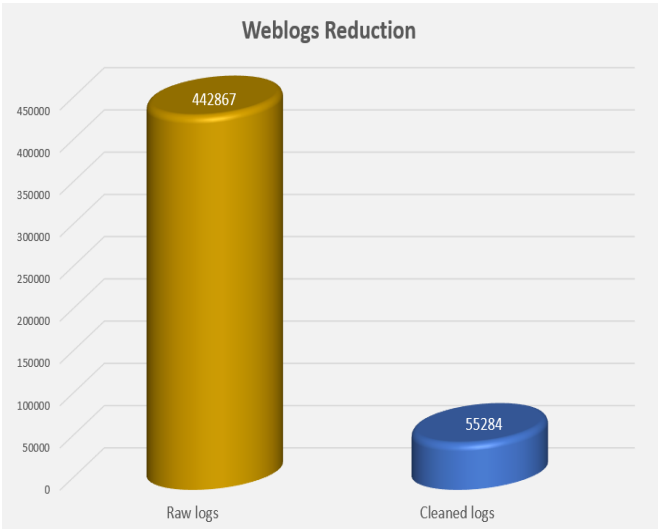

Fig. 2: Weblog reduction.

The IP address of the user, the user’s name, the user request time, URL, IP, the

bytes communicated and received, and so on are used to identify the user. In this work, the user is identified by their IP address, which is unique to them. The algorithm behind the user identification is detailed below. By using this technique, the 8074 unique ID is found.

Algorithm 1: User Identification

```
Start
For each record in the log table:
        User ID=0
        If (IP address in IP List ≠ User agent):
                User ID=User ID+1
        Else
                User ID= User ID
        End
End
```

It is not recognized that a user has visited the same website more than once with a considerable lag between each visit because the same session has not occurred in this case. Session identification is primarily concerned with the segmentation of data. It was the usual practice in web applications to set a session timeout of 30 minutes. It was necessary to detect the start and finish timings of each user access and convert them into minutes. This time log is used to determine the duration of each session. The algorithm behind the session identification is detailed below:

Algorithm 2: Session Identification

```
Start
For each record in the log table:
        Session ID=0
        Find start time
          Find end time
        time (minutes)=end time – start time
        If (IP address in IP List && time >=30minutes):
                Session ID= Session ID+1
        Else
                Session ID= Session ID
        End
End
```

Figure 3 depicts the frequency distribution of the total number of sessions per user as represented by the total number of sessions per user. The x-axis represents the number of sessions per user, while the y-axis represents the frequency of sessions per user. It is obvious from the chart that when the number of sessions per user rises, the frequency will decrease to less than 10.

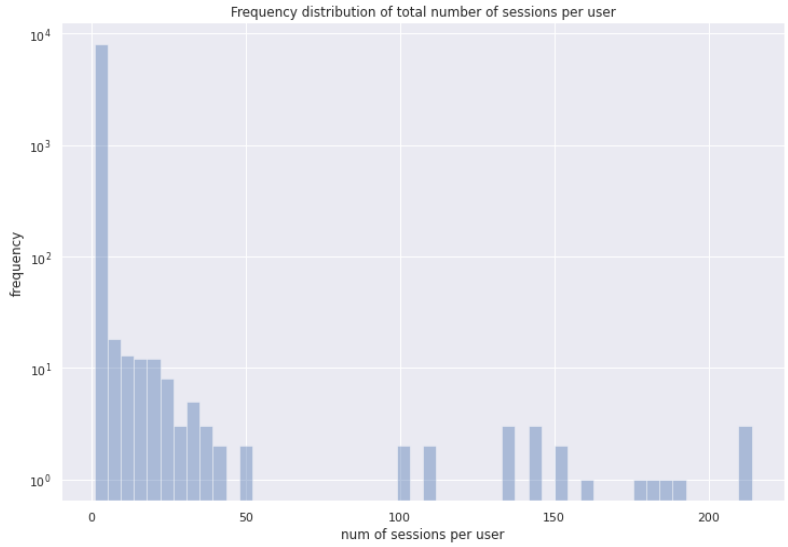

Fig. 3: Frequency distribution of the total number of sessions per user

Figure 4 depicts the frequency distribution of the length of a user's session time. The x-axis represents the duration of the session in seconds, while the y-axis represents the frequency of the session. Based on the graph, it can be deduced that the frequency of sessions and the duration of each session are inversely related. If the session duration is shorter than 5000 seconds, the frequency with which the web page is visited is higher than ten times.

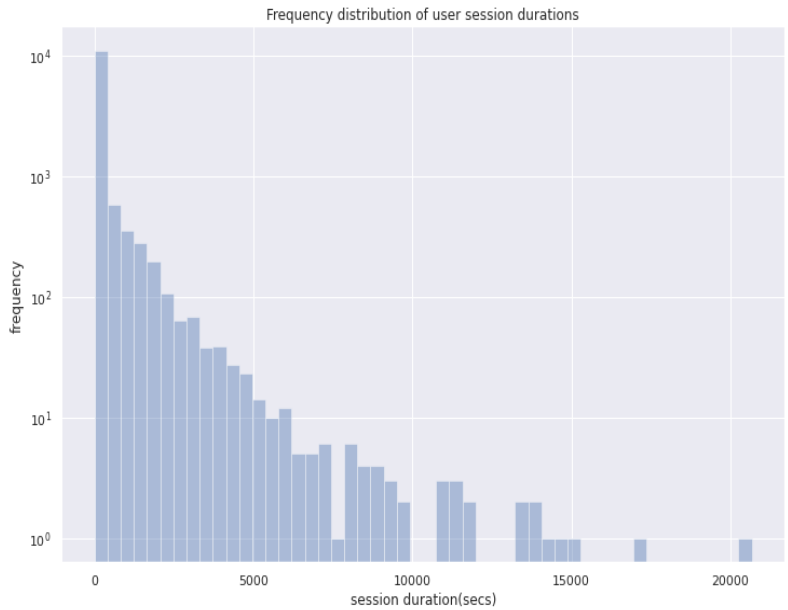

Fig. 4: Frequency distribution of user session durations

3.3. Grouping Similar Users

Pattern analysis and pattern discovery are done from pre-processing data.

3.3.1 Page Matrix

The first step is to create a page matrix. The page matrix contains users in rows and pages (URL) in the column. And the element of the matrix is time (t). The first element in the subscript of t represents the user and the second element represents the page (URL). Figure 5 shows the page matrix. the t_{11}, t_{12} element in the figure represents the total time spent by user 1 on page (URL) 1 and page (URL) 2. Similarly, t_{21}, t_{22} represents user 2 spent time on page (URL) 1 and page (URL) 2. Therefore, the row vector depicts the kind of individual while also drawing out their own particular set of capabilities. In addition, the column vector depicts the site's structure as well as the patterns of user access. To find groups of users who share a similar kind of user, row vectors can be used to identify groups of users. The algorithm used to create the page matrix is detailed below:

	Page 1	Page 2	Page 3	...	Page n
User 1	t_{11}	t_{12}	t_{13}	...	t_{1n}
User 2	t_{21}	t_{22}	t_{23}	...	t_{2n}
User 3	t_{31}	t_{32}	t_{33}	...	t_{3n}
⋮	⋮	⋮	⋮	⋮	⋮
User n	t_{n1}	t_{n2}	t_{n3}	...	t_{nn}

Fig. 5: Page matrix

Algorithm 3: Page Matrix

```
Start
Create rows for user
Create column for page (URL)
n=0
p=0
For rows >=n:
        For column >=p:
        Fill element in column
        p=p+1
       end
        n=n+1
end
```

3.3.2 Distance Matrix

Secondly, the distance matrix is created. The distance matrix contains the user in both rows and columns. And the element of the matrix is the similarity measure (d). The first element in the subscript of d represents the user in rows and the second element represents the user in columns. Figure 6 shows the distance matrix. The d12 element in the figure represents a similarity measure between user 1 and user 2. By using the Euclidean distance, the similarity measure is calculated. The formula used for calculating the similarity measure is given below:

$$Euclidean\ \ Distance\ \ (d_{ij}) = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}(User\ \ i - User\ \ j)^2}$$

Then, the Euclidean distance between vectors can be calculated. A smaller euclidean distance means higher similarity. The user accessing the page (URL) means the value is set to 1 else the value is set to 0. For simple understanding, two users are taken with 5 pages (URL). Case 1: User 1 access pages 1,2,4 and 5. Then the user 2 access pages 1,2,3,4 and 5. The Euclidean distance between these two users is minimum. Case 2: User 1 access pages 1,2 and 4. Then the user 2 access pages 3 and 5. The Euclidean distance between these two users is maximum. The algorithm for computing the distance matrix is described in more depth below:

	User 1	User 2	User 3	...	User n
User 1	d_{11}	d_{12}	d_{13}	...	d_{1n}
User 2	d_{21}	d_{22}	d_{23}	...	d_{2n}
User 3	d_{31}	d_{32}	d_{33}	...	d_{3n}
⋮	⋮	⋮	⋮	⋮	⋮
User n	d_{n1}	d_{n2}	d_{n3}	...	d_{nn}

Fig. 6: Distance matrix

Algorithm 4: Distance Matrix

Start
Create rows for user
Create a column for the user
n=0
p=0
For rows >=n:

For column >=p:
Calculate Euclidean distance
Fill Euclidean distance in column
p=p+1
end
n=n+1
end

3.3.3 Adjacency Matrix

Thirdly, the adjacency matrix is used for identifying the threshold value.

$$Threshold\ \ value\ \ T = 2 * \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}}{m(m-1)}$$

where, m→Number of users.

If the similarity measure between users is less than the threshold, conclude that there is a strong correlation between the users. Else, conclude that no correlation between the users. The adjacency matrix is created using the distance matrix. The value of d_{ii} is set to 1 in the distance matrix if the d_{ii} is less than the threshold else set to 0.

Algorithm 5: Adjacency Matrix

Start
Take distance matrix
n=0
p=0
For rows >=n:
For column >=p:
If distance matrix <= Threshold:
Fill 1 in column
Else:
Fill 0 in column
p=p+1
end
n=n+1
end

3.3.4 Create graph

The graph is created using the adjacency matrix. The sample graph is shown in figure 7. The node in the graph represents the users. The users are connected based on the element in the adjacency matrix. if there is a 1 in adjacency matrix then the respective users are connected. Else the users are not connected. The algorithm that was used to generate the graph is described in greater below:

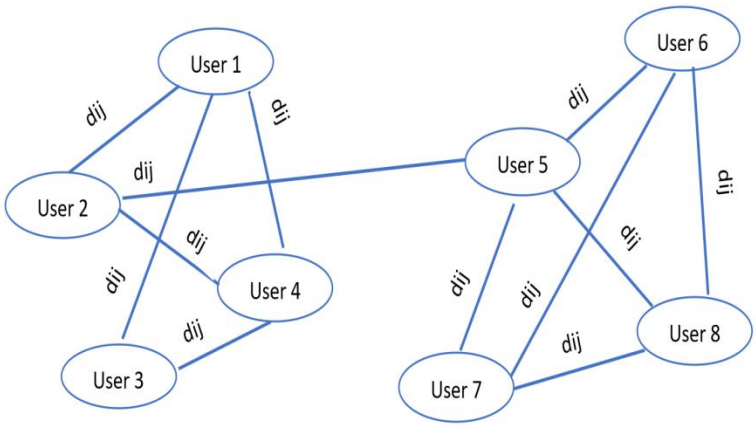

Fig. 7: Graph Creation

Algorithm 4: Create a graph

Start
Create a node for users
If 1 in adjacency matrix:
    Join nodes
Else:
    Don't join nodes.

3.3.5 Clustering

The clustering branch of data mining is quite important. The purpose of clustering is to divide a dataset into naturally occurring groups in such a way that data points within the same group are similar while data points within other groups are different. Although K-means algorithms, FCM algorithms, and EM algorithms are fundamental, they do not have the capability of dealing with complex data structures such as multidimensional arrays and matrices. When the sample space is nonconvex, it is trivial for these techniques to discover a local optimum. A growing number of academics are becoming interested in spectral clustering, owing to its high clustering performance and strong theoretical background, among other reasons. When using spectral clustering, there are no assumptions made regarding the general structure of the data to be considered. If the dataset is non-convex, it is capable of finding the global optimum and performing well in sample spaces of any shape by Nascimento MCV, de Carvalho ACPLF (2011). The notion of spectral clustering was developed in part as a result of the theory of spectral graphs. An undirected weighted graph is constructed, with each point in the dataset serving as a vertex and the similarity value between any two points connected by an edge serving as an edge weight. This graph considers data clustering as a graph partitioning issue. Then, using a certain graph cutting method, we may break down the graph into components that are connected and refer to them together as clusters.

The clustering is done based on the k value given. The sample clustering is shown in figure 8.

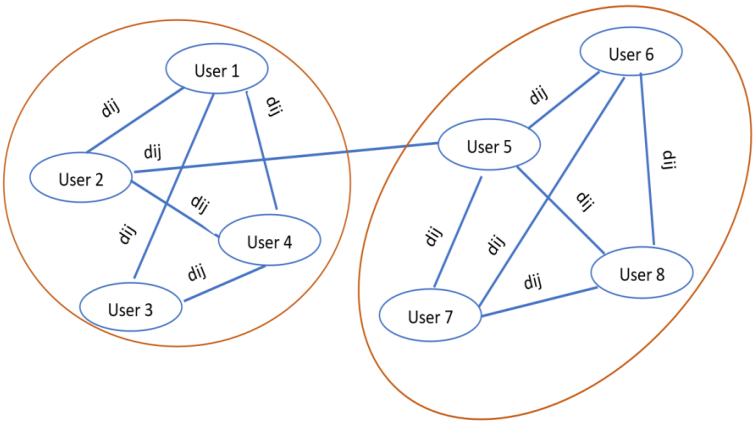

Fig. 8: Clustered Graph based on asynchronous fluid communities

3.4. Evaluation

Despite the development of numerous distance-based clustering approaches, the identification of an optimal cluster count in data has remained a recurring and, for the most part, unsolved difficulty for researchers. In this work, two scores are employed namely the Davies-Bouldin index and Silhouette.

3.4.1 Clustering

David L. Davies and Donald W. Bouldin developed the Davies–Bouldin index (DBI) in 1979, they intended it to be used as a tool for evaluating different clustering methods. Using the dataset's numbers and properties, the clustering is evaluated internally to determine whether or not it was a successful clustering effort. There is a problem with this strategy in that it does not necessarily suggest the best information retrieval as a result of a good value offered by it. Let $R_{i.i}$ be a measure of how effectively the clustering method performs in practice. For this measure to be effective, there must be a significant enough gap between two clusters namely i and j called $M_{i.i}$, as well as a small internal dispersion inside each cluster S_i and S_j for cluster$i$ and j for the measure to be effective. To ensure that these characteristics are not lost, the Davies–Bouldin index is calculated as a ratio of S_i and $M_{i.i}$.

$$R_{i,j} \geq 0$$
$$R_{i,j} = R_{j,i}$$
$$S_j \geq S_k \&\& M_{i,j} = M_{i,k} \rightarrow R_{i,j} > R_{i,k}$$
$$S_j = S_k \&\& M_{i,j} \leq M_{i,k} \rightarrow R_{i,j} > R_{i,k}$$

In proportion to the drop in value, cluster dispersion and 'tightness' within individual clusters become increasingly noticeable. A solution that satisfies all of these criteria may be found in the following:

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}}$$

DB is the abbreviation for the Davies–Bouldin index. In this case, both the data and the algorithm play a role. Di chooses the worst-case scenario to prevent overfitting into the cluster that is most similar to cluster I. It is possible to utilize the average cluster similarity in this formulation, or it is possible to use it in conjunction with a weighted average, and so on.

$$DB = \frac{1}{N}\sum_{i=1}^{N} D_i$$

Where, N→ Number of clusters

$$D_i = \max_{j \neq i} R_{i,j}$$

Under these conditions, the index that has been defined must be symmetric and non-negative. Because it is defined in terms of how much scatter there is inside a cluster, a lower value suggests that clustering is more effective than a higher number. It is the average of the similarity indicated as S_i above between the most similar clusters in each of the clusters taken over all of them. Because the Davies–Bouldin index (DBI) measures how similar a cluster is to another, it supports the assumption that no cluster has to be identical to another in every way. Using this index, it is possible to identify how many clusters are present in the data by plotting it against the number of clusters across which it was computed across the data. If i is the smallest possible number, it is a good estimate of how many clusters the data might be sorted into under ideal circumstances. It is easier to compute Davies-Bouldin scores than it would otherwise be. Because it is calculated completely using point-wise distances, the index is entirely reliant on the dataset's numbers and qualities to function properly.

3.4.2 Silhouette

It is feasible to assess and evaluate data clusters for consistency through the use of a technique known as "silhouette." A visual depiction of the effectiveness of each object's classification using the classification algorithm is generated using the technique Peter J. Rousseeuw. (1987), When compared to the items in other groups, how similar is an object to the objects in its cluster when compared to the objects in other groups? This is determined by looking at the silhouette value (separation). Small shapes surrounded by thin lines depict items that are well-matched with other items inside its cluster but are not well-matched with other objects in the near vicinity of the form. When a big number of goods have a high monetary worth, a clustering arrangement is lucrative for the organization, as is the case in most cases. If a large number of points have low or negative values, the clustering setup probably has either an excessive number of clusters or a limited number of clusters. To locate the silhouette, any distance metric, such as the Euclidean or Manhattan distance, may be used to compute the position of the silhouette and pinpoint its location. The Silhouette value can be calculated using the below equation:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, if \ \ |C_i| > 1$$

$$s(i) = 0, if \ \ |C_i| = 1$$

Where,

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j)$$

C_i→ith data in C cluster

It is possible to quantify the tightness of a cluster by looking at the mean $s(i)$over all of the points in the cluster. As a result, it is critical to determine how successfully the data has been clustered by examining their total means. The presence of a big or small number of clusters is feasible; nevertheless, it is more typical for certain clusters to have silhouettes that are significantly thinner than those of others. As a consequence, using silhouette plots and mean values, it is feasible to determine the number of clusters present in a dataset in question. It is possible to increase the likelihood that the silhouette will be maximized by using

feature weights that are unique to each cluster using feature weights that are specific to each cluster R.C. de Amorim, C. Hennig (2015).

4. Results and Discussion

The dataset was collected from Kannada University. From making the data clean done some preprocessing steps. The distance matrix created using the Euclidean distance is shown in figure 9. The rows and columns in the matrix represent the user id. The element in each matrix denotes the distance between the users.

	164.100.133.228	106.193.47.149	64.233.172.41	64.233.173.169	64.233.173.167
164.100.133.228	0.000000	465.068740	2958.202657	3794.840881	4334.728243
106.193.47.149	465.068740	0.000000	2946.273583	3788.342000	4325.983396
64.233.172.41	2958.202657	2946.273583	0.000000	4386.121287	4973.150839
64.233.173.169	3794.840881	3788.342000	4386.121287	0.000000	4407.925775
64.233.173.167	4334.728243	4325.983396	4973.150839	4407.925775	0.000000
...	...	...	...	...	...
106.193.15.85	459.268133	67.193750	2947.039367	3788.937596	4326.505442
157.49.128.162	1052.102353	966.556258	3100.767168	3900.873870	4427.104497
157.45.209.69	464.577155	7.071068	2946.281729	3788.348335	4325.989406
202.83.56.245	464.456292	9.055385	2946.287160	3788.352558	4325.993105
106.193.44.46	467.311772	50.744064	2946.710197	3788.310007	4325.408920

3497 rows × 3497 columns

Fig. 9: Distance matrix

By using the distance matrix obtained in the previous step, used for calculating the adjacency matrix. The threshold value is used for this work. The threshold value obtained is 660.1820309871049. The adjacency matrix is shown in figure 10.

	164.100.133.228	106.193.47.149	64.233.172.41	64.233.173.169	64.233.173.167	180.151.118.160	14.139.184.38
164.100.133.228	1	1	0	0	0	1	1
106.193.47.149	1	1	0	0	0	1	1
64.233.172.41	0	0	1	0	0	0	0
64.233.173.169	0	0	0	1	0	0	0
64.233.173.167	0	0	0	0	1	0	0
...	...	...	...	...	...	...	...
106.193.15.85	1	1	0	0	0	1	1
157.49.128.162	0	0	0	0	0	0	0
157.45.209.69	1	1	0	0	0	1	1
202.83.56.245	1	1	0	0	0	1	1
106.193.44.46	1	1	0	0	0	1	1

3497 rows × 3497 columns

Fig. 10: Adjacency matrix

Next, similar users are identified by applying the Clustered Graph based on

asynchronous fluid communities' algorithm. For evaluating the metrics are used. based on the metrics the k value is chosen. The list of k values such as 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 30, and 50 are used. Using the above-mentioned score, the best k value is chosen. The score for different k values is shown in Table 1.

Table 1. Score for various k values

Number of clusters	Silhouette	Davies-Bouldin Index
2	0.84	0.11
3	0.84	5.74
4	0.82	1.51
5	0.82	2.52
6	0.82	0.60
7	0.86	2.47
8	0.86	2.01
9	0.86	1.91
10	0.81	1.79
15	0.79	4.82
20	0.78	2.60
30	0.33	5.40
50	0.02	6.93

With the help of the previously provided values, figure 11 is done. It helps to identify the k value.

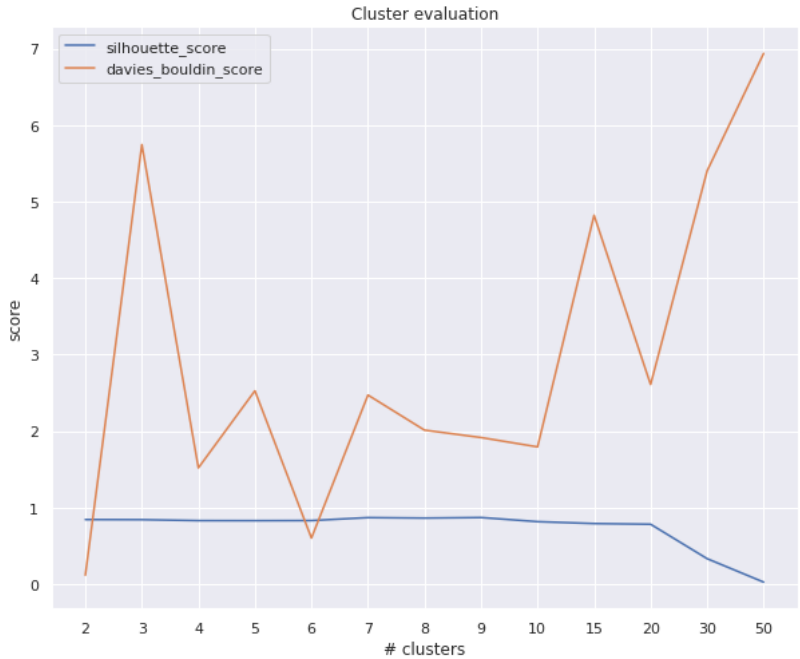

Fig. 11: Cluster evaluation

The x-axis in the graph shows the k-values and the y-axis represents the score. For indicating the Davies-Bouldin index the red color is used and for Silhouette the blue color is used. A measure of average "similarity" between clusters, the Davies-Bouldin index is derived from a comparison of the distance between clusters and the size of a cluster when comparing them. The lowest possible score is zero. The value nearer to zero represents the division is better. The high-scoring value of Silhouette represents, the clusters are more likely to be found in dense, well-separated groups than are sparsely distributed groups. By using the above-mentioned criteria, the best k is chosen. The Davies-Bouldin index should be below from the graph it is found that at the value of k=2. And the second-lowest value is obtained at the value of k=6. Then the Silhouette value will be the same and nearer to 0.8 for k value from 2 to 10. Both scores are good at the k value of 2.

5. Conclusion

For website owners, it's always a good idea to understand more about how their visitors use their site. End-user activity patterns can be discovered with the use of web mining. It may be difficult to find the specific information you're looking for online, given the sheer volume of pages available. Finding people who share similar interests is an important initial step in establishing a website recommendation system for users. Data from Kannada University's web server records are used in this study. Until the algorithmic pre-processing, there are 442867 samples. The data has been cleaned up and reduced to 55284 samples to identify specific users and sessions. Page, distance, and adjacency matrices are all computed in this step. To create a graph, the final matrix value of the adjacency is used. Spectral clustering is employed to group persons who are interested in the same topic. The Silhouette and Davies-Bouldin Index are used to verify the accuracy of the user's clustering. Using this clustered strategy, websites will be recommended to customers in the future.

References

M. Spiliopoulou. (2001). Web usage mining for web site evaluation, *Communications of the ACM*, 43(8), 127-134.

Jaideep Srivastava, Prasanna Desikan, Vipin Kumar, (2005). Web Mining Concepts, Applications, and Research Directions, *Studies in Fuzziness and Soft Computing*, 180, 275-307. https://doi.org/10.1007/11362197_10

Magali, Chaitra L., AyeshaAzeema Maniyar, and Padma Dandannavar, (2015). Pre-Processing and Analysis of Web Server Logs, *International Journal of Innovative Research in Advanced Engineering*, 8(2), 46-55.

Erritali, Mohammed, and Hanane Ezzikouri. (2015). Pretreatment of weblog files. *Journal of Information Sciences and Computing Technologies*, 2(1), 108-121.

Pabarskaite, Z., & Raudys, A. (2007). A process of knowledge discovery from weblog data: Systematization and critical review. *Journal of Intelligent Information Systems*, 28(1), 79-104.

Facca, F. M., & Lanzi, P. L. (2005). Mining interesting knowledge from weblogs: a survey. *Data & Knowledge Engineering*, 53(3), 225-241.

Liu, H., & Kešelj, V. (2007). Combined mining of Web server logs and web contents for classifying user navigation patterns and predicting users' future requests. *Data & Knowledge Engineering*, 61(2), 304-330.

Nascimento MCV, de Carvalho ACPLF, (2011). Spectral methods for graph clustering: a survey. *Eur J Oper Res*, 211(2), 221–231.

Da Sun, Tong Nie, (2020). A web service recommendation algorithm based on BaisSVD, *IEEE 5th Information Technology and Mechatronics Engineering Conference*. https://doi.org/10.1109/ITOEC49072.2020.9141664

Shen B., Cao L., Yao M., Gao Y. (2016). Mining preferred navigation patterns by consolidating both selection and time preferences, *World Wide Web*, 19(5), 979-1007.

Adeniyi D.A., Wei Z., Yongquan Y., (2016). Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method, *Applied Computing and Informatics*, 12(1), 90-108.

Geng R., Tian J. (2015). Improving web navigation usability by comparing actual and anticipated usage, *IEEE Transactions on Human-Machine Systems*, 45(1), 84-94.

D. Kerana Hanirex et al, (2011). Efficient Algorithm for Mining Frequent Itemsets Using Clustering Technique, *International Journal on Computer Science and Engineering*, 3(3), 1028-1032.

Ö. Y. Kasap and M. A. Tunga, (2017). A polynomial modeling based algorithm in the top-N recommendation, Exp. *Syst. Appl.* 79, 313–321.

S. Zhang, S. Zhang, N. Y. Yen, and G. Zhu (2017). The recommendation system of micro-blog topic based on user clustering, Mobile *Netw. Appl.*, 22(2), 228–239.

M. A. Sharif and V. V. Raghavan, (2017). Link prediction based hybrid recommendation system using user-page preference graphs, *Int. Joint Conf. Neural Networks (IJCNN)*, 1147–1154.

H. Li, Z. Xu, T. Li, G. Sun and K. K. R. Choo, (2017). An optimized approach for massive web page classi¯cation using entity similarity based on semantic network, *Future Gen. Computer Syst.* Vol. 76, 510–518.

T. T. S. Nguyen, H. Y. Lu, and J. Lu, (2014). Web-page recommendation based on web usage and domain knowledge, *IEEE Trans. Knowl. Data Eng*., 26(10), 2574–2587.

Davies, David L.; Bouldin, Donald W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2), 224–227.

Peter J. Rousseeuw. (1987). Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Computational and Applied Mathematics*, 20, 53–65.

R.C. de Amorim, C. Hennig. (2015). Recovering the number of clusters in data sets with noise features using feature rescaling factors. *Information Sciences*, 324, 126–145

Rajesh K Shukla; Prachi Sharma; Noopur Samaiya; Monika Kherajani, (2020). WEB USAGE MINING-A Study of Web data pattern detecting methodologies and its applications in Data Mining, *International Conference on Data, Engineering and Applications (IDEA)*, IEEE Xplore.

Om Prakash P.G.; Ananthakumaran S; Sathishkumar M; Ganeshan R, (2021). Analyzing the User Navigation Pattern from Web Logs Using Maximum Frequent Pattern Approach, *6th International Conference on Inventive Computation Technologies*, https://doi.org/10.1109/ICICT50816.2021.9358751