Acta Scientiarum

http://periodicos.uem.br/ojs
ISSN on-line: 1807-8664
Doi: 10.4025/actascitechnol.v44i1.58653

An empirical research and comparative analysis of clustering performance for processing categorical and numerical data extracts from social media

Shini Renjith[1,2*], A. Sreekumar [1] and M. Jathavedan [1]

[1]Department of Computer Applications, Cochin University of Science and Technology, Kalamassery, Ernakulam, Kerala, 682022, India. [2]Department of Computer Science and Engineering, Mar Baselios College of Engineering and Technology, Mar Ivanios College Rd, Nalanchira, Thattinakam, Thiruvananthapuram, Kerala, 695015, India. *Author for correspondence. E-mail: shinirenjith@gmail.com

ABSTRACT. Social media has significantly influenced modern lifestyle and the way in which most of the industries operate their business. Social media data refers to the contents created by users during their social interactions in the form of text, sound, visuals, etc. It has now evolved as the major source of information for different industry verticals like retail, marketing, advertising, tourism, hospitality, education, etc. The huge volume of data resulted in the necessity for better and efficient procedures for personalized information retrieval. Traditional data mining and information retrieval techniques based on content-based and/or collaborative filtering proved computationally costly and less scalable against the volume it must deal with. Adoption of clustering techniques is a potential solution for this problem as it can minimize the amount of data required to be managed in industrial applications like recommender systems. This empirical research focuses on evaluating multiple clustering algorithms with the goal of finding an ideal solution for clustering numerical data extracted from social media sources. Three different publicly available datasets with varying number of attributes and records from tourism domain are used for the experiments conducted as part of this work.

Keywords: Collaborative filtering; clustering algorithm; data mining; recommender systems; social media.

Received on April 13, 2021.
Accepted on August 9, 2021.

Introduction

Be it any industry, social media offers quite a lot of opportunities for data scientists to perform their research. Social media interactions can be treated as a true reflection of the societal thought process on various matters (Kaplan & Haenlein, 2010). It contains huge volume of data about users and their information exchanges captured for a considerable period. Various personalized and contextualized recommender systems available across different industries leverages the predictive power of social media at large (Schoen et al., 2013; Renjith, Sreekumar, & Jathavedan, 2019; Renjith, Biju, & Mathew, 2020; Renjith, Sreekumar, & Jathavedan, 2021a, b). One of the key challenges in the area is the huge amount of data required to be processed.

Collaborative and hybrid filtering algorithms (Renjith & Anjali, 2013a; Renjith & Anjali, 2014) are extensively used by conventional recommender systems in societal contexts for forecasting probable user actions and thereby generating contextualized recommendations. The challenge associated with this approach is the huge amount of data to be processed when social media data is considered as the source of information (Jiang, Qian, Mei, & Fu, 2016; Coelho, Nitu, & Madiraju, 2018). This leads to the concept of clustering input data and process only the relevant cluster. It is important to identify the best clustering approach to be adopted considering the datasets in consideration (Estivill-Castro, 2002). This empirical study compares the quality of output and performance of multiple clustering algorithms by applying on three real time datasets collected from travel and tourism industry.

Section 2 describes the clustering concept with a quick review of various clustering algorithms and cluster evaluation techniques that are examined in this empirical study. Sections 3 reviews existing literature in this area and Section 4 describes on the methodology adopted, tools used, and datasets considered. Section 5 captures the quantitative and statistical data collated through the experiments. Section 6 discusses our observations and inferences, and Section 7 summarizes the paper along with a brief on our future research plans.

Antecedents

Clustering

In the world of machine learning, clustering is the process of creating data segments within a dataset with similar elements. The aim of clustering algorithms is to form clusters with the highest intra-cluster similarity and the lowest inter-cluster similarity. Similarity is calculated in terms of a distance measure with less distance indicating more similarity. Typical distance measures in consideration include Euclidean, Cosine, Manhattan, Jaccard and Minkowski distances (Renjith & Anjali, 2013b).

K-means algorithm

K-means (Hartigan & Wong, 1979; MacQueen, 1967) is the most commonly used partitioning clustering technique. It groups a given dataset into k dissimilar segments via an iterative process. The mean value of elements present in a cluster is labeled as the centroid. The aim of the algorithm is to ensure minimum within cluster variation for each cluster being formed. The within cluster variation at cluster level and total within cluster variation is calculated as Equation (1) and (2) respectively.

$$WCV(C_k) = \sum_{E_i \in C_k} (E_i - \mu_k)^2 \quad (1)$$

$$Total\ WCV = \sum_{k=1}^{K} W\ CV(C_k) = \sum_{k=1}^{K} \sum_{E_i \in C_k} (E_i - \mu_k)^2 \quad (2)$$

E_i is an element in cluster, C_k with μ_k being the centroid and K being the total count of clusters formed.

K-means algorithm is considered as the simplest, less complex, easy to implement, and efficient clustering approach and thereby is the most popular one. The major challenge with this algorithm is its low tolerance towards the existence of noise or outliers. Other limitations include the prerequisite to specify the cluster count in advance, sensitiveness for initialization, and its inability to deal with non-convex cluster shapes.

K-medoids algorithm

K-medoids algorithm (Kaufman & Rousseeuw, 1987; Kaufman & Rousseeuw, 1990) works like k-means algorithm but differs in the logic used for determining the centroids. While k-means algorithm does not require an actual element from the dataset to mark as centroid, k-medoids algorithm always assign an element from the population as the medoid whose average dissimilarity with other cluster elements is the lowest. Partitioning around Medoids (PAM) is the most used k-medoids procedure. PAM algorithm iteratively identifies k medoids from the elements of dataset using an objective function and assign other elements to the nearest medoids for form clusters.

Compared to k-means algorithm, k-medoids algorithm is less susceptible to the presence of outliers. While k-means algorithm focuses on minimizing the total squared error, k-medoids attempts to minimize the sum of dissimilarities among entities within a cluster and its centroid. The main constraint with the algorithm is its high time complexity while comparing with k-means algorithm.

Clustering large applications (CLARA)

CLARA (Clustering for Large Applications) algorithm (Kaufman & Rousseeuw, 1987; Park & Jun, 2009) is an expansion of k-medoids algorithm to handle datasets having large volume. The CLARA algorithm selects a sample from the population to apply PAM procedure to determine best possible set of medoids. The goodness of these medoids are then verified against the complete population. The sampling and clustering process are repeated to minimize the sampling bias. This approach helps in reducing the limitations of PAM like lengthy processing time and high memory needs.

The major advantage of CLARA algorithm is its ability to deal with larger datasets while preserving the benefits of k-Medoids or PAM. The key drawback of the algorithm is its dependency on the sample size being chosen. Also, any probable bias in selecting the sample may influence the overall quality of the clustering process.

Fuzzy C-means algorithm

Fuzzy C-means algorithm (FCM) (Bezdek, Ehrlich, & Full, 1984) is a soft clustering technique with every element in the population can belong to each cluster formed to a certain magnitude. Based on the similarity or distance of an element with the centroid of a cluster, the extent of belonging of it to the cluster is

calculated. If the element is nearer to a centroid, its belonging to the corresponding cluster will be high, of course with the cumulative membership value for an element at any point of time is kept as 100% or 1. Mathematically, the belongingness of an element in clusters can be denoted as in Equation (3).

$$\sum_{k=1}^{K} \mu_k (E_i) = 1 \quad (3)$$

K is cluster count and μ_k is the extent of belongingness of an element, E_i in cluster, C_k.

Fuzzy C-means algorithm is the best option to select while the dataset contains overlapped clusters. As it allows partial belongings of entities in multiple clusters, the algorithm always converges. However, this results in high computational time requirements. As like k-means algorithm, FCM also has the constraint of specifying cluster count in advance and has high sensitiveness for initialization.

Agglomerative nesting

Agglomerative nesting aka hierarchical agglomerative clustering (AGNES or HAC) (Kaufman & Rousseeuw, 1990; Zepeda-Mendoza & Resendis-Antonio, 2013) is the hierarchical clustering strategy which adopts a bottom-up approach. This algorithm commences with every element in the dataset as an individual cluster. At each succeeding phase, the algorithm successively merges the closest pair of clusters till one cluster remains. The algorithm has to take irreversible clustering decisions based on local patterns at each stage as it lacks global distribution details of the dataset. Agglomerative nesting is the widely used hierarchical clustering algorithm in practice.

Hierarchical clustering algorithm does not require upfront information on the cluster count. Agglomerative nesting is easy to implement and could yield best results in most of the scenarios. Major challenges include relatively high time complexity ($O(n^2 \log n)$) and difficulty in identifying correct number of clusters from dendrogram. Also

Divisive analysis

Divisive Analysis aka hierarchical divisive clustering (DIANA or HDC) (Kaufman & Rousseeuw, 1990; Jayaprada, Amarapini, & Gayathri, 2014) is the hierarchical clustering strategy which adopts a top-down approach. The clustering process begins from the top with all elements considered as members of one group. This cluster is then sliced using a flat clustering algorithm like k-means. These steps are recursively performed until each element become a member of a singleton cluster.

The implementation of top-down approach is complicated in comparison with bottom-up approach as it require a separate algorithm to slice the clusters. However, this clustering model possess the advantage of having complete knowledge of the global distribution of dataset while taking clustering decisions.

Expectation-maximization

The expectation-maximization (EM) clustering technique (Bouveyron, Girard, & Schmid, 2007) works like k-means algorithm, but with the key difference of not performing hard-assignment of elements to clusters and rather do a soft assignment. EM clustering algorithm determines the probability of cluster belongingness (expectation) based on probability distributions. The aim of EM clustering algorithm is to ensure maximum overall probability for the final set of clusters. The algorithm assumes that dataset is always a subset of Gaussian distribution mixture.

The EM clustering is widely used for determining the missing data in a sample. In EM algorithm the likelihood always increases with the number of iterations. It leverages both forward and backward probabilities and suffers with slow convergence to the local optima.

Related works

There are some good review articles available which mainly talk about the challenges and approaches to address those in the big data context. (Parker, 2012) explained the differences between large scale machine learning and standard supervised classification scenario. (Jagadish et al., 2014) articulated various challenges in leveraging the full potential of big data like inconsistency, heterogeneity, incompleteness, privacy, timeliness, visualizations, etc. (Grolinger et al., 2014) described various issues with MapReduce while handling big data. (Najafabadi et al., 2015) narrated on how data analytics problems can be tackled with the help of deep learning and the improvements required in specific areas of deep learning to perform better. (L'Heureux, Grolinger, Elyamany, & Capretz, 2017) shared a good composition of the restraints in performing

machine learning approaches with big data and their cause-effect relationship with four dimensions of big data - i.e., volume, variety, velocity, and veracity.

(Xu & Wunsch II, 2005) did an extensive survey of clustering algorithms, but lacked on covering the big data or social media context in it. (Shirkhorshidi, Aghabozorgi, Wah, & Herawan, 2014) provided an academic review of various clustering algorithms to handle big data concerns. Other theoretical works regarding the usage of clustering algorithms in the context of big data include the literatures from (Sajana, Rani, & Narayana, 2016), (Ajin & Kumar, 2016), and (Dave & Gianey, 2016). The significant works in empirical analysis of clustering algorithms are limited. (Wei, Lee, & Hsu, 2003) conducted an experimental study of the data characteristics of CLARA, CLARANS, GACR, and GAC-RARw clustering algorithms. (Fahad et al., 2014) attempted a comparison of five candidate clustering algorithms using ten different datasets - eight of which are simulated and two are publicly available datasets used in multiple researches.

Latest studies (Shin, 2021a, b, c, d, e, f) in the area of algorithmic journalism give high focus to algorithmic trust, which can be considered as a measure of the digital affordance to algorithm based offerings. It is important to establish sufficient level of algorithmic trust before finalizing an approach in artificial intelligence based systems. It is observed that there are only a limited number of empirical analysis are conducted on clustering algorithms in the past focusing on a particular industry segment. Few attempts of this type include experimental works performed by S. Renjith et al. by using datasets from tourism domain (Renjith, Sreekumar, & Jathavedan, 2018; Renjith, Sreekumar, & Jathavedan, 2020a, b, c). Our attempt in this work is to perform an empirical analysis of the core clustering algorithms explicitly focusing on real datasets from travel and tourism domain.

Methodology

Approach

This research is conducted using a three-stage approach as depicted in Figure 1. The same process is repeated for all datasets in consideration to arrive at the inferences.

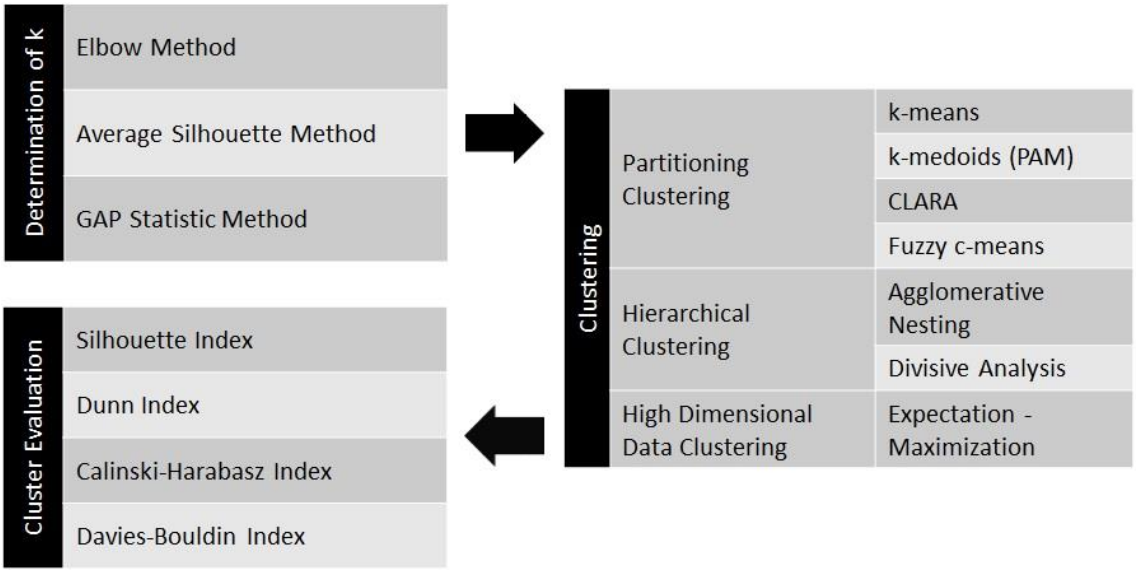

Figure 1. Three-stage research methodology adopted in this work.

Determination of k

Partitioning, hierarchical and expectation-maximization clustering techniques require to determine the optimal cluster count (k). However, determination of k is subjective and highly dependent on the similarity measures chosen and parameters considered for clustering. We chose to use the three most common algorithms to determine k - two direct methods (Elbow Method (Thorndike, 1953), Average Silhouette Method (Rousseeuw, 1987)) and one statistical testing method (GAP Statistic Method (Tibshirani, Walther, & Hastie, 2001)). Direct methods work by optimizing a criterion like intra-cluster sum of squares in Elbow Method and statistical testing method compares evidences against a null hypothesis.

Clustering

We have experimented with four partitioning clustering algorithms (k-means (Hartigan & Wong 1979; MacQueen, 1967), k-medoids (Kaufman & Rousseeuw, 1987; Kaufman & Rousseeuw, 1990), CLARA (Kaufman & Rousseeuw, 1987; Park & Jun, 2009), and fuzzy c-means (Bezdek et al., 1984)), two hierarchical clustering algorithms (agglomerative nesting (Kaufman & Rousseeuw, 1990; Zepeda-Mendoza & Resendis-Antonio, 2013), and divisive analysis (Kaufman & Rousseeuw, 1990; Jayaprada et al., 2014)), and one model-based high dimensional data clustering algorithm (expectation - maximization (Bouveyron et al., 2007)).

Cluster evaluation

We have validated the clustering outputs using three popular internal evaluation criteria namely Silhouette Index (Rousseeuw, 1987), Dunn Index (Dunn, 1973), Calinski-Harabasz Index (Calinski & Harabasz, 1974) and Davies-Bouldin Index (Davies & Bouldin, 1979).

Tools used

R programming language (R Core Team, 2009; Tierney, 2012), the free open-source programming language for statistical computing and RStudio (Racine, 2011), its integrated development environment are extensively leveraged in this experimental analysis. The specific packages that are used in this research are captured in Table 1.

Table 1. Purpose of R packages used in the analysis.

Package	Purpose
stats	k-means algorithm
cluster	Partition around medoids (k-medoids or PAM) and Clustering for Large Applications (CLARA) algorithms
ppclust	Fuzzy c-means (FCM) algorithm
factoextra	Agglomerative nesting (AGNES) and Divisive analysis (DIANA) algorithms
HDclassif	Expectation - Maximization algorithm (Bergé, Bouveyron, & Girard, 2012)
NbClust	Get optimal count of clusters (Charrad, Ghazzali, Boiteau, & Niknafs, 2014)
clusterCrit	Internal evaluation of clusters
stats	k-means algorithm

Datasets used

We leveraged three real-world datasets from travel and tourism domain that are publicly available on The UCI Machine Learning Repository (Renjith & Anjali, 2014; Renjith, Sreekumar, & Jathavedan, 2018) for our experiments and recorded the results. These datasets correspond to the user interest details collated from destination reviews, ratings on attractions visited, and feedbacks provided on different point of interests from three different geographies. Subsequent sections depict more specifics of the datasets in consideration.

Dataset 1

High level dataset description for dataset 1 is provided in Table 2. This dataset was used by the research team to evaluate collaborative filtering technique and was derived from reviews on points of interest published by 249 top contributing members of holidayiq.com in 2014 (Renjith & Anjali, 2014). Reviews spanning across 6 categories mentioned in Table 2 about the point of interests located in South India were collated and the number of reviews per category per reviewer is captured. Personally identifiable information (PII) is masked in the dataset to ensure anonymity.

Dataset 2

High level dataset description for dataset 2 is provided in Table 3. Dataset 2 is collated by crawling TripAdvisor.com (Renjith, Sreekumar, & Jathavedan, 2018). Destination reviews falling under 10 categories specified in Table 3 across East Asian countries are captured. Each user rating is recorded in a scale of 0 to 4 and the average rating is calculated per category per user. Personally identifiable information (PII) is not captured in the dataset.

Table 2. Dataset description for dataset 1.

Description	User interest information derived from traveler reviews on tourist destinations from South India
Record Count	249
Count of Attributes	1 user attribute and 6 user interest attributes
Details of Attributes	Attrib01 : Unique user identification number Attrib02 : Count of reviews published on sports facilities. Attrib03 : Count of reviews published on religious destinations. Attrib04 : Count of reviews published on natural bodies like beach, lake, etc. Attrib05 : Count of reviews published on cinemas, exhibitions, etc. Attrib06 : Count of reviews published on shopping destinations. Attrib07 : Count of reviews published on parks or picnic destinations.

Table 3. Dataset description for dataset 2.

Description	User interest information derived from destination reviews on tourist destinations across East Asian countries
Record Count	980
Count of Attributes	1 user attribute and 10 feedback attributes
Details of Attributes	Attrib01 : Unique user identification number Attrib02 : Avg. user rating on art galleries Attrib03 : Avg. user rating on dance clubs Attrib04 : Avg. user rating on juice bars Attrib05 : Avg. user rating on restaurants Attrib06 : Avg. user rating on museums Attrib07 : Avg. user rating on resorts Attrib08 : Avg. user rating on parks/picnic spots Attrib09 : Avg. user rating on beaches Attrib10 : Avg. user rating on theaters Attrib11 : Avg. user rating on religious institutions

Dataset 3

High level dataset description for dataset 3 is provided in Table 4. Dataset 3 is collated by taking user ratings from Google reviews (Renjith, Sreekumar, & Jathavedan, 2018). Ratings on points of interest from 24 categories mentioned in Table 4 throughout Europe are captured. Average user rating per category is calculated and recorded (Google ratings range from 1 to 5). No personally identifiable information (PII) is stored in our systems.

Determination of k

Three indices considered by us in this paper to determine optimal cluster count (K) are Elbow Method, Average Silhouette Method, and one GAP Statistic Method. We used the functions available in the R package factoextra for capturing and plotting the criteria considered.

Empirical research

Elbow method

Elbow method (Thorndike, 1953) is proposed based on the clustering goal of achieving the lowest total within-cluster sum of square (Total WSS). This approach calculates the 'Total WSS' as a function of cluster count for the dataset in consideration. The optimal cluster count, K' is determined through iteration when there is not much improvement to 'Total WSS' by choosing (K'+1) clusters. Mathematically, the 'Total WSS' is represented as Equation (4).

$$Total\ WSS = f(k) = \sum_{k=1}^{K} \sum_{E_i \in C_k} (\mu_k - E_i)^2 \quad (4)$$

K is the number of clusters considered in each iteration and E_i is an element of the cluster, C_k having centroid, μ_k.

The plot of 'Total WSS' against number of clusters form the shape of an elbow and hence the approach is named so. Figure 2 represents the elbow plots for the three datasets considered in this paper.

Table 4. Dataset description for dataset 3.

Description	Average user rating on different types of tourist destinations from Europe
Record Count	5456
Count of Attributes	1 user attribute and 24 rating attributes
Details of Attributes	Attrib01 : Unique user identification number Attrib02 : Avg. user feedback score for churches Attrib03 : Avg. user feedback score for resorts Attrib04 : Avg. user feedback score for beaches Attrib05 : Avg. user feedback score for parks Attrib06 : Avg. user feedback score for theatres Attrib07 : Avg. user feedback score for museums Attrib08 : Avg. user feedback score for malls Attrib09 : Avg. user feedback score for zoos Attrib10 : Avg. user feedback score for restaurants Attrib11 : Avg. user feedback score for pubs/bars Attrib12 : Avg. user feedback score for local services Attrib13 : Avg. user feedback score for burger/pizza shops Attrib14 : Avg. user feedback score for hotels/other lodgings Attrib15 : Avg. user feedback score for juice bars Attrib16 : Avg. user feedback score for art galleries Attrib17 : Avg. user feedback score for dance clubs Attrib18 : Avg. user feedback score for swimming pools Attrib19 : Avg. user feedback score for gyms Attrib20 : Avg. user feedback score for bakeries Attrib21 : Avg. user feedback score for beauty & spas Attrib22 : Avg. user feedback score for cafes Attrib23 : Avg. user feedback score for viewpoints Attrib24 : Avg. user feedback score for monuments Attrib25 : Avg. user feedback score for gardens

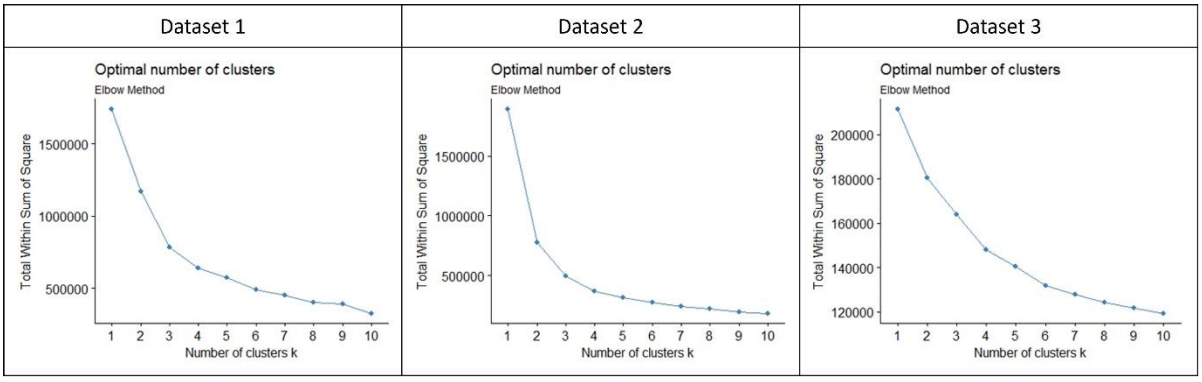

Figure 2. Determination of optimal cluster count using Elbow method.

Average silhouette method

Average silhouette method (Rousseeuw, 1987) is based on the clustering objective of having maximum Average Silhouette. This approach calculates the 'Average Silhouette' as a function of cluster count for the dataset in consideration. The optimal cluster count, K' is identified when the maximum value is achieved for 'Average Silhouette'. Mathematically, the 'Average Silhouette' is represented as Equation (5).

$$Average\ Silhouette\ =\ f(K) = \frac{1}{K}\sum_{k=1}^{K} S_k \quad (5)$$

K is the number of clusters considered in each iteration and S_k is the cluster mean Silhouette, which is calculated through steps Equation (6) to (10).

For each element, E_i determine the average dissimilarity, $\alpha(i)$ with all other elements of the cluster, C_k to which it belongs to

$$\alpha(i) = \frac{1}{n_k - 1}\sum_{i' \in I_k \text{ and } i' \neq i} d\,(E_i, E_{i'}) \quad (6)$$

For all other clusters, $C_{k\prime}$ to which element, E_i does not belong to, determine the average dissimilarity, $\delta(E_i, C_{k\prime})$ of the element, E_i to the elements of every other clusters, $C_{k\prime}$.

$$\delta(E_i, C_{k\prime}) = \frac{1}{n_{k\prime}} \sum_{i\prime \in I_{k\prime}} d\,(E_i, E_{i\prime}) \quad (7)$$

The dissimilarity between element E_i and the nearest cluster to which it does not belong to, $\beta(i)$ is computed as

$$\beta(i) = \min_{k\prime \neq k} \delta(E_i, C_{k\prime}) \quad (8)$$

The Silhouette width for the element E_i is calculated as

$$s(i) = \frac{\beta(i) - \alpha(i)}{\max(\alpha(i), \beta(i))} \quad (9)$$

The cluster mean Silhouette, S_k is computed as

$$S_k = \frac{1}{n_k} \sum_{i \in I_k} s\,(i) \quad (10)$$

Figure 3 represents the Average silhouette plots of the three datasets considered in this paper.

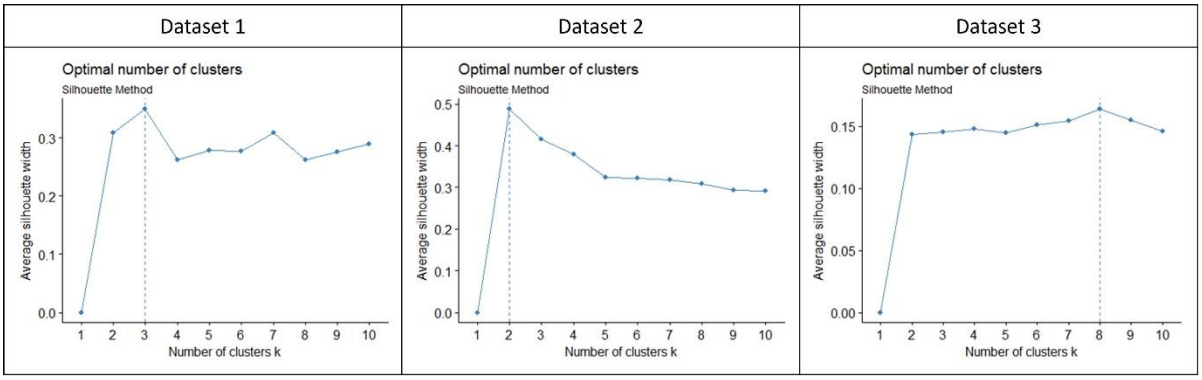

Figure 3. Determination of optimal cluster count using Average silhouette method.

Gap statistic method

Gap statistic method (Tibshirani, Walther, & Hastie, 2001) calculates 'total intra-cluster variation' as a function of cluster count for the dataset and compares it against anticipated values for null reference distribution of all the elements in the dataset. The optimal cluster count, K' is calculated as in Equation (11) to (13).

$$K' = \min(k),\ such\ that\ Gap(k)\ \geq Gap(k+1) - s_{k+1} \quad (11)$$

$$Gap_n(k) = E_n^*\{\log W_k\} - \log W_k \quad (12)$$

$$Simulation\ Error,\ s_k = \sqrt{1 + \frac{1}{B\ sd(k)}} \quad (13)$$

W_k is the variance quantity, B is the Monte Carlo and $sd(k)$ is the standard deviation.
Figure 4 represents the Gap Statistic plots for the three datasets considered in this paper.

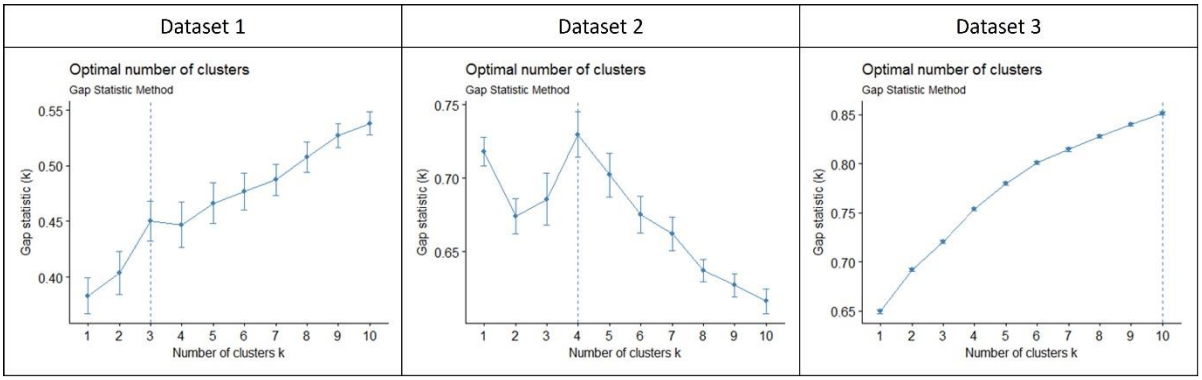

Figure 4. Determination of optimal cluster count using Gap Statistic method.

Clustering

Figures 5, 6, 7, 8, 9, 10 and 11 depicts the two dimensional plots of the resulting clusters on the three datasets in consideration.

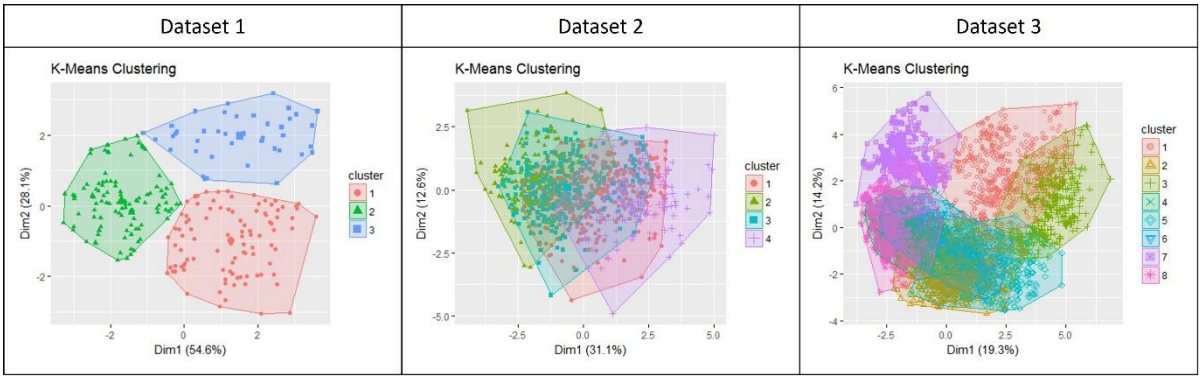

Figure 5. k-means clustering of three datasets in consideration.

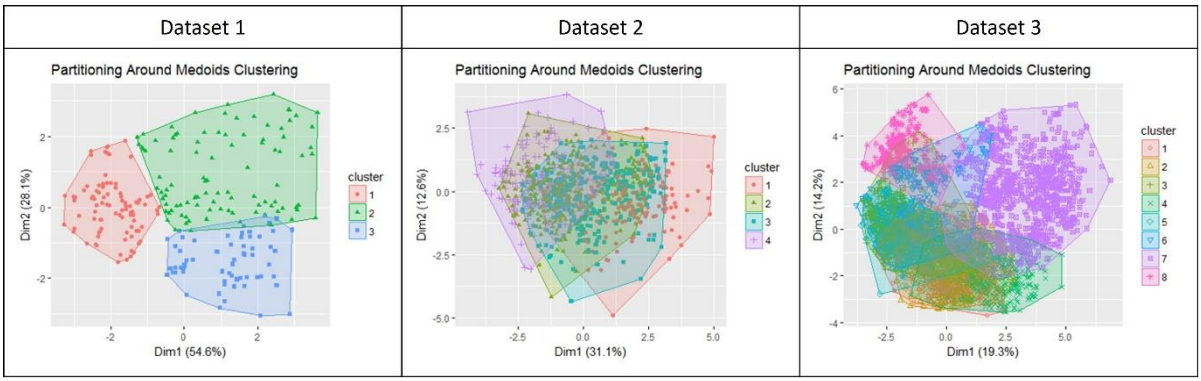

Figure 6. k-medoids (PAM) clustering of three datasets in consideration.

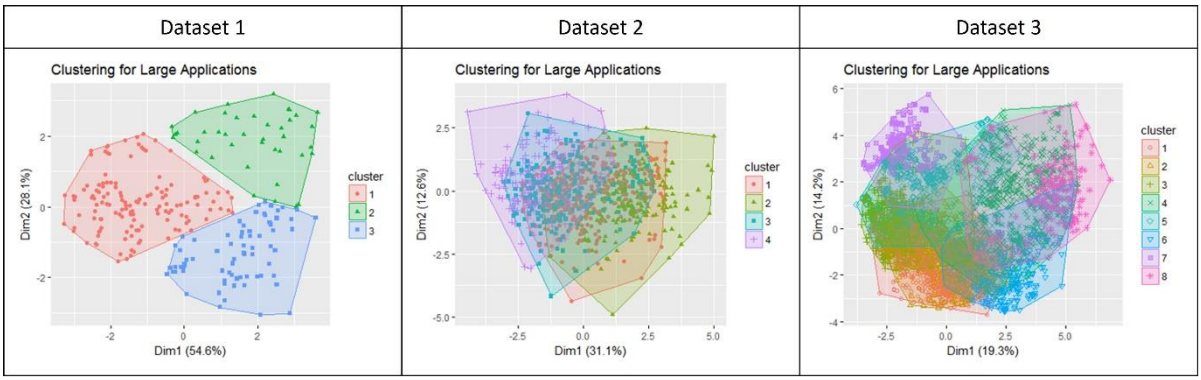

Figure 7. CLARA clustering of three datasets in consideration.

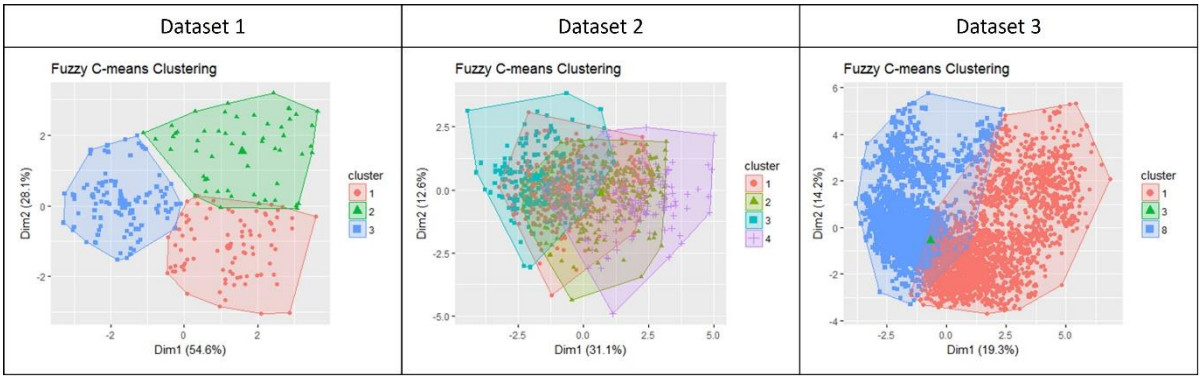

Figure 8. Fuzzy c-means clustering of three datasets in consideration.

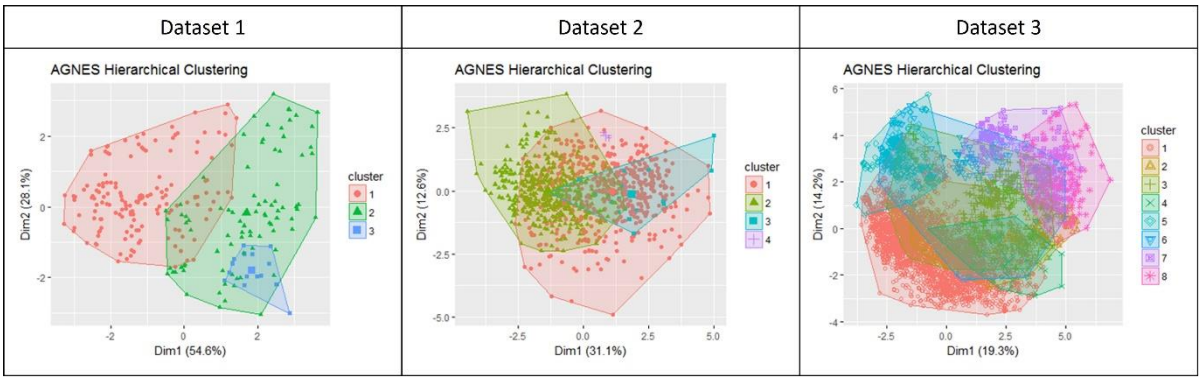

Figure 9. Agglomerative hierarchical clustering of three datasets in consideration.

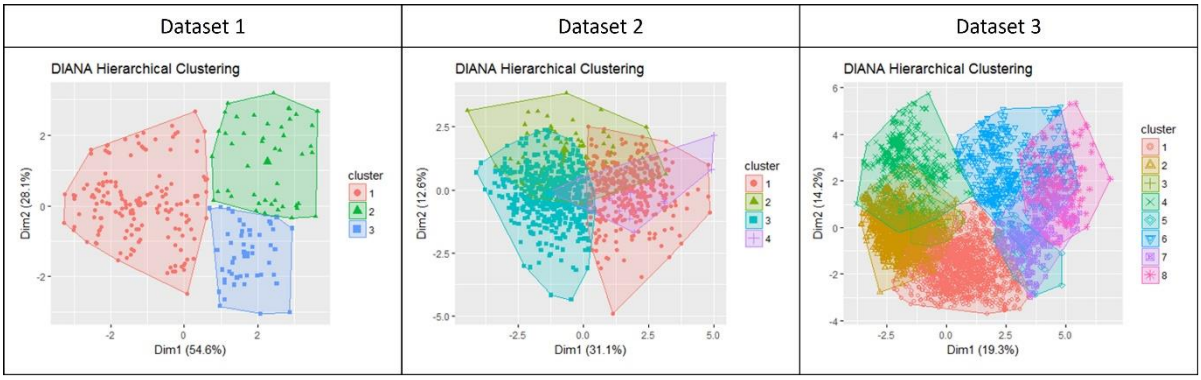

Figure 10. Divisive Hierarchical clustering of three datasets in consideration.

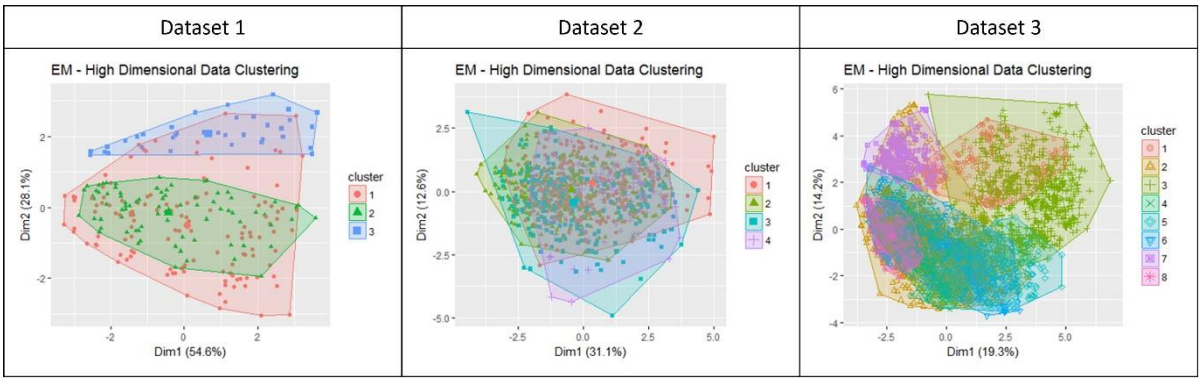

Figure 11. High Dimensional Data Clustering (using Expectation - maximization) of three datasets in consideration.

Cluster evaluation

Cluster evaluation is the process of measuring the quality or goodness of the clustering process. The key approach in evaluating clusters is internal evaluation, where the clustering accuracy is measured as a quality index. There are multiple internal evaluation measures proposed by various researchers and the key approaches include Silhouette index (Rousseeuw, 1987), Dunn index (Dunn, 1973), Calinski-Harabasz index (Calinski & Harabasz, 1974) and Davies-Bouldin index (Davies & Bouldin, 1979). Tables 5, 6 and 7 summarize complete set of observations captured using the R package clusterCrit for each clustering experiments on all three datasets.

Table 5. Cluster evaluation details for dataset 1.

Index	Rule	k-means	k-medoids	CLARA	FCM	AGNES	DIANA	EM
Ball-Hall	maximum difference	3108.694	3408.315	3286.101	3316.374	3855.931	3670.431	5023.637
Banfeld-Raftery	minimum	1997.632	2012.631	2015.598	2003.073	2083.153	2043.688	2136.191
C	minimum	0.109289	0.156458	0.106268	0.111697	0.202202	0.132742	0.360147
Calinski-Harabasz	maximum	149.7532	126.6908	137.2106	145.1667	66.9698	110.8282	35.9588
Davies-Bouldin	minimum	1.039654	1.169786	1.032207	1.071810	1.352548	1.104113	2.913632

Det Ratio	minimum difference	-9.2397	-7.3837	-8.4748	8.5860	-6.3090	7.9184	4.6422
Dunn	maximum	0.053582	0.047632	0.065719	0.044738	0.084775	0.056638	0.026759
Baker-Hubert Gamma	maximum	0.749760	0.639956	0.753406	0.743996	0.569685	0.700797	0.240393
G plus	minimum	0.058491	0.081130	0.059737	0.058632	0.107293	0.074145	0.182447
GDI	maximum	0.053582	0.047632	0.065719	0.044738	0.084775	0.056638	0.026759
GDI	maximum	0.312923	0.234861	0.380998	0.253149	0.413514	0.351454	0.152812
GDI	maximum	0.108711	0.081032	0.132948	0.088336	0.143536	0.122316	0.053064
GDI	maximum	1.010285	1.125364	1.056989	1.056989	1.014902	1.010285	0.806539
GDI	maximum	5.900093	5.548888	6.127773	5.980983	4.950473	6.269047	4.605828
GDI	maximum	2.049721	1.914494	2.138273	2.087042	1.718371	2.181810	1.599364
GDI	maximum	0.492003	0.548003	0.524025	0.517120	0.494528	0.512719	0.349893
GDI	maximum	2.873311	2.702065	3.037971	2.926124	2.412203	3.181536	1.998103
GDI	maximum	0.998202	0.932274	1.060093	1.021060	0.837306	1.107267	0.693837
GDI	maximum	0.399303	0.445758	0.419673	0.423407	0.356309	0.408797	0.124991
GDI	maximum	2.331941	2.197923	2.433007	2.395849	1.738001	2.536676	0.713777
GDI	maximum	0.810128	0.758334	0.848993	0.836023	0.603282	0.882837	0.247858
GDI	maximum	0.191100	0.215337	0.221162	0.210940	0.226125	0.229717	0.211111
GDI	maximum	1.116031	1.061772	1.282158	1.193606	1.102987	1.425447	1.205570
GDI	maximum	0.387715	0.366336	0.447406	0.416504	0.382860	0.496097	0.418632
Ksq DetW	maximum difference	5.58.E+28	6.98.E+28	6.09.E+28	-6.01.E+28	8.17.E+28	-6.51.E+28	-1.11.E+29
Log Det Ratio	minimum difference	NaN	NaN	NaN	535.384	NaN	515.229	382.264
Log SS Ratio	minimum difference	0.196804	0.029565	0.109333	0.165698	-0.607943	-0.104203	-1.229810
McClain-Rao	minimum	0.569550	0.612671	0.571827	0.569349	0.665627	0.600086	0.836456
PBM	maximum	3953.264	2874.049	3889.564	3352.529	3227.626	2942.661	1623.846
Point-Biserial	maximum	-27.09968	-23.18306	-27.97701	-26.60413	-21.69544	-26.58485	-9.37878
Ray-Turi	minimum	0.316503	0.344230	0.328744	0.313391	0.575935	0.352242	3.990921
Ratkowsky-Lance	maximum	0.431878	0.414224	0.419599	0.429530	0.370675	0.416439	0.244542
Scott-Symons	minimum	NaN	NaN	NaN	7472.943000	NaN	NaN	NaN
SD	minimum	0.444982	0.494001	0.463430	0.475443	0.595929	0.560087	0.745296
SD	minimum	0.016679	0.016177	0.016581	0.016061	0.021395	0.015918	0.055505
S Dbw	minimum	0.444982	0.827334	1.606287	Inf	1.462595	5.810087	1.678630
Silhouette	maximum	0.360524	0.305478	0.353079	0.323384	0.317772	0.311686	0.148054
Tau	maximum	0.512631	0.429613	0.524413	0.503537	0.402293	0.493362	0.166613
Trace W	maximum difference	783411.6	855770.7	821173.4	796810.6	1124800.0	913824.9	1344235.0
Trace WiB	maximum difference	4.160582	3.467845	3.866838	3.962628	3.601713	4.173832	2.394011
Wemmert-Gan carski	maximum	0.476168	0.420121	0.460678	0.457101	0.338303	0.407069	0.118968
Xie-Beni	minimum	17.576710	30.14763	13.40604	28.07055	10.17403	18.34990	87.07315

Table 6. Cluster evaluation details for dataset 2.

Index	Rule	k-means	k-medoids	CLARA	FCM	AGNES	DIANA	EM
Ball-Hall	maximum difference	412.059	408.698	419.786	413.067	1251.769	1620.499	1687.000
Banfeld-Raftery	minimum	5754.738	5746.353	5762.442	5753.021	7146.294	7034.551	7310.028
C	minimum	0.062866	0.065830	0.070027	0.063380	0.390822	0.313280	0.387854
Calinski-Harabasz	maximum	1338.7110	1322.6970	1290.2630	1333.2070	91.7943	147.9322	32.5353
Davies-Bouldin	minimum	0.829626	0.850951	0.872814	0.844661	4.241983	5.656803	5.532857
Det Ratio	minimum difference	-9.6247	9.3921	9.0682	-9.5368	-9.4520	-25.2786	4.4471
Dunn	maximum	0.010938	0.016630	0.009694	0.009391	0.004871	0.004938	0.004503
Baker-Hubert Gamma	maximum	0.816273	0.808407	0.795785	0.814896	0.165908	0.291998	0.107332
G plus	minimum	0.036715	0.037223	0.039204	0.036508	0.208229	0.174128	0.173479
GDI	maximum	0.010938	0.016630	0.009694	0.009391	0.004871	0.004938	0.004503
GDI	maximum	0.083379	0.123971	0.075915	0.070381	0.038030	0.038888	0.044819
GDI	maximum	0.029635	0.044156	0.027170	0.025084	0.014129	0.014456	0.015599
GDI	maximum	0.865136	0.841024	0.684735	0.865136	0.550273	0.896060	0.914911
GDI	maximum	6.595021	6.269512	5.362181	6.483784	4.296002	7.057368	9.106571
GDI	maximum	2.344042	2.233076	1.919110	2.310841	1.596043	2.623395	3.169596
GDI	maximum	0.348413	0.337470	0.307879	0.349650	0.202104	0.216721	0.188625
GDI	maximum	2.655986	2.515712	2.411012	2.620460	1.577830	1.706897	1.877479
GDI	maximum	0.944007	0.896047	0.862895	0.933940	0.586193	0.634495	0.653468
GDI	maximum	0.319263	0.295566	0.268033	0.308509	0.049881	0.036296	0.043258
GDI	maximum	2.433774	2.203327	2.098970	2.312131	0.389426	0.285867	0.430568
GDI	maximum	0.865027	0.784781	0.751216	0.824051	0.144679	0.106264	0.149862
GDI	maximum	0.123927	0.123076	0.117667	0.125257	0.130287	0.132107	0.128966
GDI	maximum	0.944711	0.917482	0.921449	0.938741	1.017159	1.040471	1.283662
GDI	maximum	0.335775	0.326789	0.329784	0.334570	0.377893	0.386769	0.446786

Ksq DetW	maximum difference	8.52.E+19	-8.73.E+19	-9.04.E+19	8.60.E+19	8.67.E+19	3.24.E+19	-1.84.E+20
Log Det Ratio	minimum difference	NaN	2195.069	2160.676	NaN	NaN	NaN	1462.398
Log SS Ratio	minimum difference	1.414612	1.402578	1.377751	1.410492	-1.265300	-0.788096	-2.302524
McClain-Rao	minimum	0.367882	0.370062	0.375365	0.367628	0.813124	0.708248	0.914219
PBM	maximum	4654.983	4288.195	4032.787	4472.888	374.272	239.935	80.346
Point-Biserial	maximum	-17.95063	-17.46502	-17.12382	-17.76397	-5.38039	-8.69440	-2.02907
Ray-Turi	minimum	0.253663	0.298844	0.323877	0.272556	12.228300	21.023680	14.771200
Ratkowsky-Lance	maximum	0.218594	0.217163	0.216372	0.217995	0.243366	0.292046	0.141017
Scott-Symons	minimum	NaN	NaN	NaN	NaN	NaN	NaN	NaN
SD	minimum	0.173550	0.173544	0.178539	0.174266	0.629842	0.835184	0.898221
SD	minimum	0.065546	0.071095	0.073397	0.067935	0.239807	0.286920	0.201936
S Dbw	minimum	2.160563	2.334583	1.959573	2.304092	4.564995	6.526231	7.487439
Silhouette	maximum	0.385259	0.371411	0.353295	0.377094	0.076218	-0.058792	0.029971
Tau	maximum	0.516042	0.503921	0.493095	0.511805	0.117232	0.204792	0.066915
Trace W	maximum difference	369788.7	373381.8	380877.8	371015.8	1475195.0	1300210.0	1719471.0
Trace WiB	maximum difference	8.475324	8.189494	7.884873	8.338135	4.168003	6.162842	2.878671
Wemmert-Gan carski	maximum	0.520069	0.514221	0.497763	0.516158	0.001709	0.000000	0.008162
Xie-Beni	minimum	216.125400	94.39833	247.59490	294.15460	1282.19600	1136.05800	1363.25800

Table 7. Cluster evaluation details for dataset 3.

Index	Rule	k-means	k-medoids	CLARA	FCM	AGNES	DIANA	EM
Ball-Hall	maximum difference	23.301	22.290	24.320	22.069	25.389	24.444	23.157
Banfeld-Raftery	minimum	16980.860	17080.530	17380.240	-Inf	18250.960	17684.940	17418.150
C	minimum	0.151689	0.210255	0.201569	0.263825	0.215935	0.181886	0.287781
Calinski-Harabasz	maximum	550.3073	507.4880	450.7393	470.4533	261.3554	393.6296	386.4883
Davies-Bouldin	minimum	2.105948	2.134367	2.500716	1.963873	2.104312	2.020829	2.834875
Det Ratio	minimum difference	1486.8010	618.5754	336.1966	4.6127	676.4823	288.4327	184.1171
Dunn	maximum	0.005693	0.010780	0.007623	0.001034	0.080858	0.012336	0.034581
Baker-Hubert Gamma	maximum	0.723710	0.602866	0.625110	0.424875	0.523657	0.624652	0.437096
G plus	minimum	0.033017	0.046522	0.047721	0.143781	0.118866	0.072171	0.072640
GDI	maximum	0.005693	0.010780	0.007623	0.001034	0.080858	0.012336	0.034581
GDI	maximum	0.022423	0.038339	0.028310	0.003905	0.295241	0.044018	0.133730
GDI	maximum	0.007825	0.013596	0.009957	0.001374	0.103715	0.015526	0.047017
GDI	maximum	0.886049	0.965920	0.911923	0.697276	0.917979	0.978892	0.681791
GDI	maximum	3.490176	3.435305	3.386773	2.632919	3.351862	3.493016	2.636612
GDI	maximum	1.218022	1.218263	1.191174	0.926316	1.177476	1.232025	0.926984
GDI	maximum	0.521696	0.509647	0.546404	0.476318	0.579940	0.597573	0.445269
GDI	maximum	2.054980	1.812563	2.029278	1.798582	2.117565	2.132342	1.721935
GDI	maximum	0.717159	0.642790	0.713725	0.632779	0.743880	0.752101	0.605401
GDI	maximum	0.278428	0.261215	0.206479	0.308091	0.276217	0.326578	0.153811
GDI	maximum	1.096736	0.929015	0.766836	1.163354	1.008565	1.165340	0.594814
GDI	maximum	0.382745	0.329457	0.269707	0.409292	0.354299	0.411028	0.209126
GDI	maximum	0.284736	0.278561	0.293977	0.352395	0.260646	0.325292	0.217224
GDI	maximum	1.121586	0.990706	1.091796	1.330648	0.951711	1.160751	0.840044
GDI	maximum	0.391417	0.351334	0.384000	0.468150	0.334326	0.409410	0.295344
Ksq DetW	maximum difference	2.67.E+89	6.42.E+89	1.18.E+90	1.21.E+91	5.87.E+89	1.38.E+90	2.16.E+90
Log Det Ratio	minimum difference	39852.710	35068.000	31741.350	8341.192	35556.240	30905.300	28456.160
Log SS Ratio	minimum difference	-0.346617	-0.427621	-0.546205	-1.757077	-1.091213	-0.681683	-0.699992
McClain-Rao	minimum	0.717476	0.759115	0.758793	0.852714	0.824172	0.777429	0.822768
PBM	maximum	2.263	2.342	2.612	3.301	2.423	2.528	2.211
Point-Biserial	maximum	-0.87507	-0.73370	-0.76860	-0.68436	-0.82559	-0.89172	-0.56328
Ray-Turi	minimum	1.389454	1.830232	2.960701	1.448698	1.870234	1.346409	4.359012
Ratkowsky-Lance	maximum	0.223888	0.213439	0.208174	0.213276	0.192858	0.208925	0.201579
Scott-Symons	minimum	NaN	NaN	NaN	-Inf	NaN	NaN	NaN
SD	minimum	0.768706	0.685568	0.737399	0.608274	0.839776	0.826652	0.811169
SD	minimum	0.418255	0.505083	0.652722	0.320192	0.482594	0.398025	0.786303
S Dbw	minimum	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Silhouette	maximum	0.149896	0.138825	0.129434	NaN	0.158646	0.163184	0.132553
Tau	maximum	0.353806	0.291808	0.315408	0.300432	0.369940	0.387363	0.222056
Trace W	maximum difference	123845.9	127970.2	133879.1	180303.3	158266.9	140403.3	141264.1
Trace WiB	maximum difference	16.287450	12.412110	10.943120	3.603373	13.819290	12.616240	9.776459
Wemmert-Gan carski	maximum	0.238891	0.224468	0.180752	0.190689	0.204696	0.211226	0.138728
Xie-Beni	minimum	3323.916000	1074.63400	2172.26900	128586.70000	21.82478	943.66480	86.23692

Silhouette index

The Silhouette Index (SI) is a cluster evaluation criterion which represents the quality of clustering by indicating how well the elements in the dataset is clustered (Rousseeuw, 1987). Alternatively, it indicates how similar an element is to the cluster it belongs to. A high value for SI is considered as a result of good clustering. Mathematically, Silhouette Index is calculated as the mean of the mean Silhouette (10) of all the clusters as in Equation (14).

$$S.I. = \frac{1}{K}\sum_{k=1}^{K} S_k \quad (14)$$

Figure 12 represents evaluation of multiple clustering algorithms on all three datasets considered in this paper using Silhouette Index.

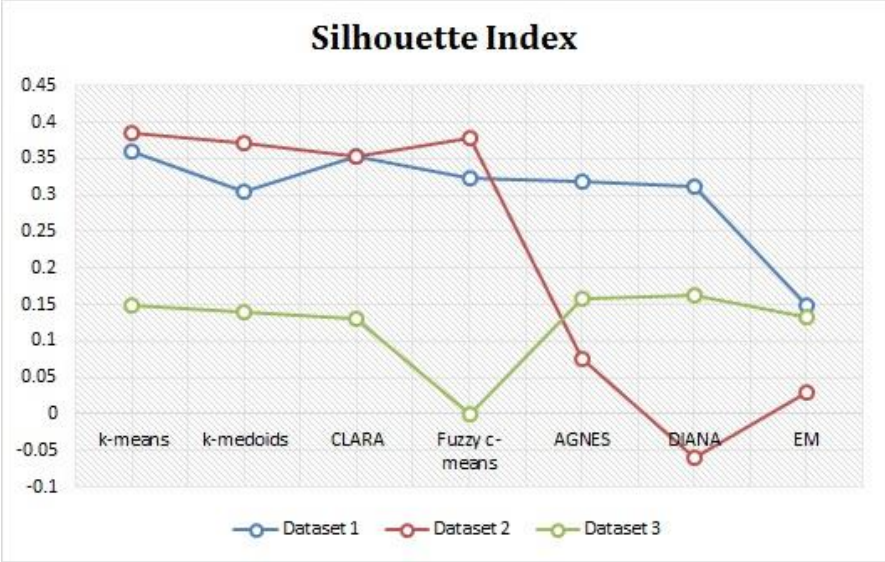

Figure 12. Evaluation of clustering algorithms using Silhouette Index.

Dunn index

Dunn Index (DI) is an internal cluster evaluation metric described as the ratio of minimum inter-cluster separation to maximum intra-cluster distance (Dunn, 1973). Mathematically, it can be arrived through steps Equation (15) to (19). The distance between two clusters C_k and $C_{k'}$ can be represented by the distance between their closest points.

$$d_{kk'} = \min_{i \in I_k \text{ and } j \in I_{k'}} \left|\left| E_i^{\{k\}} - E_j^{\{k'\}} \right|\right| \quad (15)$$

The inter-cluster separation can be computed as the minimum of the pairwise distance.

$$d_{\min} = \min_{k \neq k'} d_{kk'} \quad (16)$$

The largest distance separating two points within a cluster (also referred as diameter of the cluster) can be calculates as below:

$$D_k = \max_{i,j \in I_k \text{ and } i \neq j} \left|\left| E_i^{\{k\}} - E_j^{\{k\}} \right|\right| \quad (17)$$

The intra-cluster compactness is computed as the maximum value of intra-cluster distances.

$$d_{\max} = \max_{1 \leq k \leq K} D_k \quad (18)$$

Dunn index can be represented as the ratio between inter-cluster separation and intra-cluster compactness.

$$D.I. = \frac{d_{\min}}{d_{\max}} \quad (19)$$

Figure 13 represents evaluation of multiple clustering algorithms on all three datasets considered in this paper using Dunn Index.

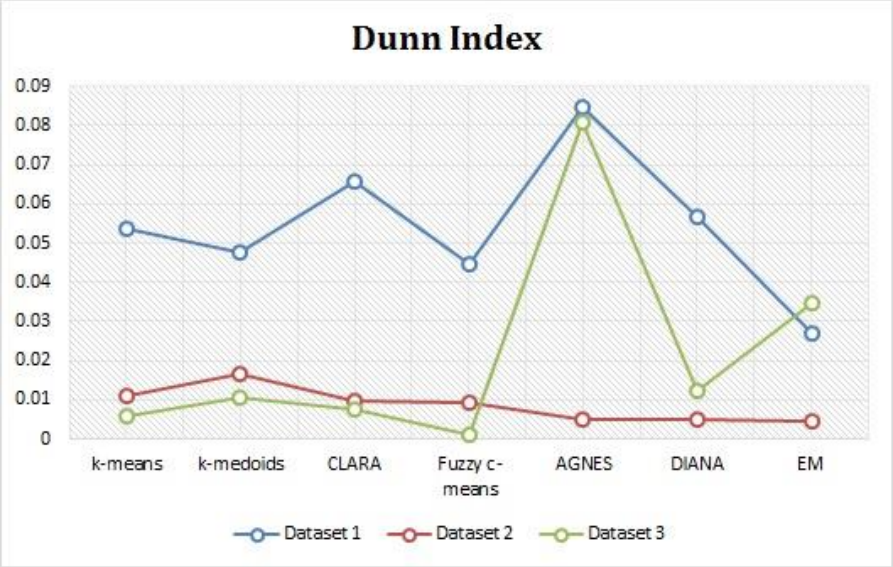

Figure 13. Evaluation of clustering algorithms using Dunn Index.

Calinski-Harabasz index

The Calinski-Harabasz Index (CHI) is an internal cluster evaluation measure expressed as the ratio of inter-cluster dispersion (variance between clusters) and intra-cluster dispersion (variance within each cluster) (Calinski & Harabasz, 1974). A larger value for Calinski-Harabasz Index is considered as the indication of good clustering. Mathematically, Calinski-Harabasz Index is calculated through steps Equation (20) to (23).

The inter-cluster dispersion can be defined as the dispersion of cluster centroids, $G^{\{k\}}$ with respect to the global centroid, G of the dataset in consideration. If n_k is the number of elements in the cluster, C_k,.

$$BCD = \sum_{k=1}^{K} n_k \left|\left| G^{\{k\}} - G \right|\right|^2 \quad (20)$$

The intra-cluster dispersion of cluster C_k is the sum of the squared distances between its elements, $E_i^{\{k\}}$ and the cluster centrod, $G^{\{k\}}$.

$$WCD^{\{k\}} = \sum_{i \in I_k} \left|\left| E_i^{\{k\}} - G^{\{k\}} \right|\right|^2 \quad (21)$$

Hence, the pooled intra-cluster dispersion can be explained as the sum of the intra-cluster dispersions of all the clusters

$$WCD = \sum_{k=0}^{K} W\,CD^{\{k\}} \quad (22)$$

Calinski-Harabasz Index is calculated as the ratio of BCD and WCD.

$$C.H.I. = \frac{\frac{BCD}{(K-1)}}{\frac{WCD}{(N-K)}} = \frac{(N-K)}{(K-1)} \frac{BCD}{WCD} \quad (23)$$

Figure 14 represents evaluation of multiple clustering algorithms on all three datasets considered in this paper using Calinski-Harabasz Index.

Davies-Bouldin index

The Davies-Bouldin Index (DBI) is a measure to evaluate cluster quality as it attempts to detect group of clusters that are well separated and compact by calculating the ratio of intra-cluster distances to inter-cluster distances (Davies & Bouldin, 1979). A smaller value for DBI represents better quality of clustering. Mathematically, Davies-Bouldin Index is calculated through steps Equation (24) to (26).

Calculate the mean distance, δ_k of the elements in cluster C_k to its centroid, $H^{\{k\}}$.

$$\delta_k = \frac{1}{n_k} \sum_{i \in I_k} \left|\left| M_i^{\{k\}} - H^{\{k\}} \right|\right| \quad (24)$$

The distance between centroids $H^{\{k\}}$ and $H^{\{k'\}}$ of clusters C_k and $C_{k'}$ is computed as

$$\Delta_{kk'} = d\left(H^{\{k\}}, H^{\{k'\}}\right) = \left|\left|H^{\{k'\}} - H^{\{k\}}\right|\right| \quad (25)$$

For each cluster, C_k identify the maximum, M_k of $\frac{\delta_k+\delta_{k'}}{\Delta_{kk'}}$ for all $k' \neq k$.

Davies-Bouldin Index is the mean value of M_k aong all the clusters in the dataset in consideration.

$$D.B.I. = \frac{1}{K}\sum_{k=1}^{K} M_k = \frac{1}{K}\sum_{k=1}^{K} \max_{k'\neq k}\left(\frac{\delta_k+\delta_{k'}}{\Delta_{kk'}}\right) \quad (26)$$

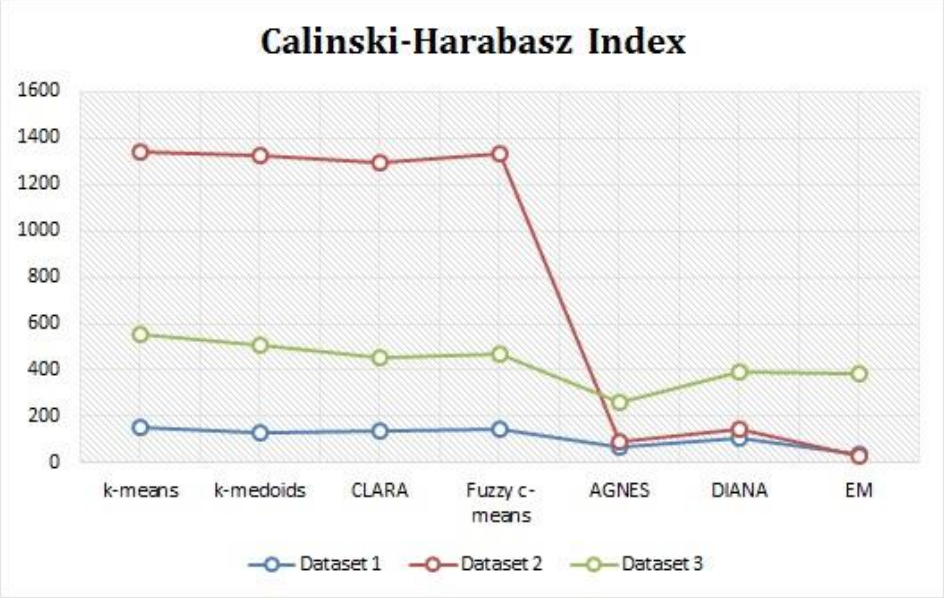

Figure 14. Evaluation of clustering algorithms using Calinski-Harabasz Index.

Figure 15 represents evaluation of multiple clustering algorithms on all three datasets considered in this paper using Davies-Bouldin Index.

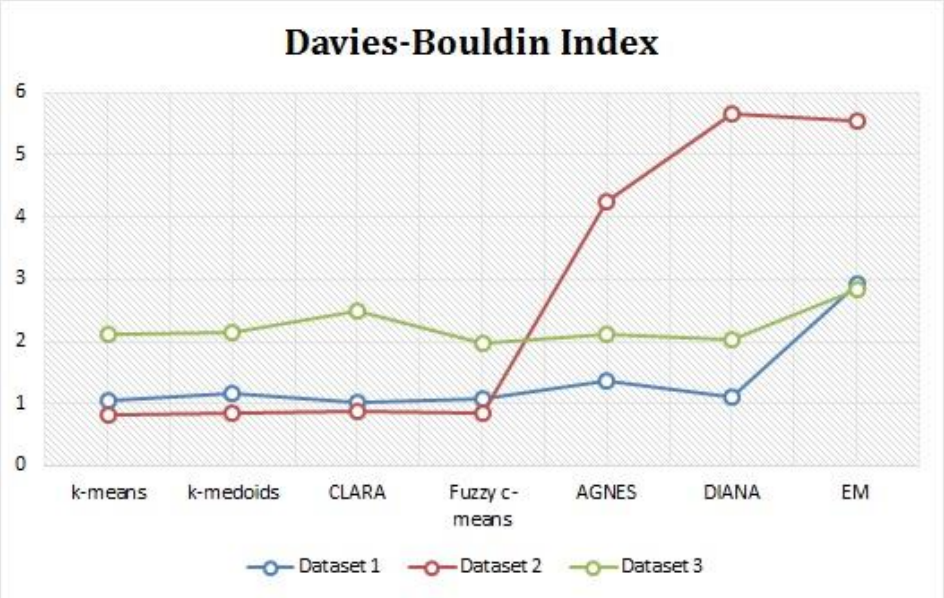

Figure 15. Evaluation of clustering algorithms using Davies-Bouldin Index.

Discussion

In this experiment, we referred to three real datasets covering numerical information extracted from reviews, feedbacks, and ratings from travelers that are collated from holidayiq.com, tripadvisor.com, and Google destination reviews, respectively. As part of the empirical analysis, we evaluated seven core clustering algorithms using internal evaluation strategies. Our consideration included four partitioning clustering

algorithms (k-means, k-medoids, CLARA, and Fuzzy c-means), two hierarchical clustering algorithms (AGNES, and DIANA) and one model-based clustering algorithm (EM).

We could observe that partitioning algorithms showed an edge with lower volume and a smaller number of attributes. However, no pattern is observed with larger volume of data and high number of attributes. So, we recommend evaluating and choosing an algorithm for the dataset to be processed rather than selecting an algorithm upfront. We introduced a structured and data driven approach in our analysis using defined and repeatable processes and is explained in section 4.1 which can be extended for evaluating clustering algorithms in similar contexts.

Another attribute which requires attention is the execution time required for various clustering algorithms. Though the execution time is highly dependent on the hardware configuration and environment variables used, it is also important to check the time complexity for various clustering algorithms. The cardinality and dimensionality of the dataset being processed have major influence on the overall processing time. We have conducted an explicit research on the same and the results are already published (Renjith, Sreekumar, & Jathavedan, 2020a).

Conclusion

Clustering is considered as a key strategy for handling large volume of data generated in social media platforms. It helps to reduce the volume of data to be processed and thereby reducing the computational cost and processing time required. Any type of social media content that can be treated as the reflection of traveler traits and/or feedbacks can be considered as a tourism social media data.

In this experiment, we evaluated seven core clustering algorithms against three real datasets covering numerical information extracted from reviews, feedbacks, and ratings from travelers on holidayiq.com, tripadvisor.com, and Google destination reviews. Based on the results obtained, no algorithm could outperform in all tourism scenarios as performance varied against evaluation criteria chosen and dataset being considered. So, it is critical to evaluate and select appropriate clustering algorithms for each of the dataset to be processed.

Another cause for the high computational intensiveness of social media data is the curse of dimensionality. As a subsequent step of this work, we are planning to leverage dimensionality reduction techniques prior to clustering process. We aim to evaluate various dimensionality reduction techniques ranging from principal component analysis (PCA) to deep learning techniques like autoencoder against real data from tourism domain.

Acknowledgements

We acknowledge support from the Department of Computer Applications, Cochin University of Science and Technology and Department of Computer Science and Engineering, Mar Baselios College of Engineering and Technology for all guidance, reviews, valuable suggestions, and very useful discussions.

References

Ajin, V. W., & Kumar, L. D. (2016). Big data and clustering algorithms. In *Proceedings of the 2016 International Conference on Research Advances in Integrated Navigation Systems* [RAINS], (p. 1-5). DOI: http://doi.org/10.1109/rains.2016.7764405

Bergé, L., Bouveyron, C., & Girard, S. (2012). HDclassif: an R Package for model-based clustering and discriminant analysis of high-dimensional data. *Journal of Statistical Software, 46*(6), 1-29. DOI: http://doi.org/10.18637/jss.v046.i06

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences, 10*(2-3), 191–203. DOI: http://doi.org/10.1016/0098-3004(84)90020-7

Bouveyron, C., Girard, S., & Schmid, C. (2007). High-dimensional data clustering. *Computational Statistics & Data Analysis 52*(1), 502–519. DOI: http://doi.org/10.1016/j.csda.2007.02.009

Calinski, T., & J. Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics, 3*(1), 1-27. DOI: http://doi.org/10.1080/03610927408827101

Charrad, M., Ghazzali, N., Boiteau, V., & Niknafs, A. (2014). NbClust: an R package for determining the relevant number of clusters in a data set. *Journal of Statistical Software, 61*(6), 1-36. DOI: http://doi.org/10.18637/jss.v061.i06

Coelho, J., Nitu, P., & Madiraju, P. (2018). A personalized travel recommendation system using social media analysis. In *Proceedings of the 2018 IEEE International Congress on Big Data (BigData Congress)*, (p. 260-263). DOI: http://doi.org/10.1109/bigdatacongress.2018.00046

Dave, M., & Gianey, H. (2016). Different clustering algorithms for big data analytics: a review. In *Proceedings of the 2016 International Conference System Modeling & Advancement in Research Trends* (SMART), (p. 328-333). DOI: http://doi.org/10.1109/sysmart.2016.7894544

Davies, D. L., & Bouldin, D. W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1*(2), 224-227. DOI: http://doi.org/10.1109/tpami.1979.4766909

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics, 3*(3), 32–57. DOI: http://doi.org/10.1080/01969727308546046

Estivill-Castro, V. (2002). Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter 4*(1), 65–75. DOI: http://doi.org/10.1145/568574.568575

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. I., ... Bouras, A. (2014). A survey of clustering algorithms for big data: taxonomy and empirical analysis. *IEEE Transactions on Emerging Topics in Computing, 2*(3), 267-279. DOI: http://doi.org/10.1109/tetc.2014.2330519

Grolinger, K., Hayes, M., Higashino, W. A., L'Heureux, A., Allison, D. S., & Capretz, M. A. M. (2014). Challenges for mapreduce in big data. In *Proceedings of the 2014 IEEE World Congress on Services* [IEEE], (p. 182-189). DOI: http://doi.org/10.1109/services.2014.41

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics, 28*(1), 100-108. DOI: http://doi.org/10.2307/2346830

Jagadish, H. V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., & Shahabi, C. (2014). Big data and its technical challenges. *Communications of the ACM, 57*(7), 86–94. DOI: http://doi.org/10.1145/2611567

Jayaprada, S., Amarapini, A., & G. Gayathri. (2014). Hierarchical divisive clustering with multi view-point based similarity measure. In *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications* (FICTA) 2013, (p. 483–91). DOI: http://doi.org/10.1007/978-3-319-02931-3_55

Jiang, S., Qian, X., Mei, T., & Fu, Y. (2016). Personalized travel sequence recommendation on multi-source big social media. *IEEE Transactions on Big Data, 2*(1), 43–56. DOI: http://doi.org/10.1109/tbdata.2016.2541160

Kaplan, A. M., & Haenlein, M. (2010). Users of the world unite! The challenges and opportunities of social media. *Business Horizons 53*, 59–68. DOI: http://doi.org/10.1016/j.bushor.2009.09.003

Kaufman, L., & Rousseeuw, P. J. (1987). Clustering by means of medoids. In Y. Dodge (Ed.), *Statistical Data Analysis Based on the L_1–Norm and Related Methods* (p. 405-416). North-Holland, NL: [s.n.]

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: an introduction to cluster analysis*. Hoboken, NJ: John Wiley & Sons. DOI: http://doi.org/10.1108/intr-06-2013-0115

L'Heureux, A., Grolinger, K., Elyamany, H. F., & Capretz, M. A. M. (2017). Machine learning with big data: challenges and approaches. *IEEE Access 5*, 7776–97. DOI: http://doi.org/10.1109/access.2017.2696365

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, (Vol. 1, p. 281-297). Recovered from: https://bitlybr.com/tLMBe

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data 2*(1). DOI: http://doi.org/10.1186/s40537-014-0007-7

Park, H.-S., & Jun, C.-H. (2009). A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications, 36*(2). 3336–3341. DOI: http://doi.org/10.1016/j.eswa.2008.01.039

Parker, C. (2012). Unexpected challenges in large scale machine learning. In *Proceedings of the 1st International Workshop on Big Data Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications - BigMine 12*, (p. 1-6). DOI: http://doi.org/10.1145/2351316.2351317

Racine, J. S. (2011). RStudio: a platform-independent IDE for R and sweave. *Journal of Applied Econometrics, 27*(1), 167–172. DOI: http://doi.org/10.1002/jae.1278

R Core Team. (2009). *R: a language and environment for statistical computing*. Vienna, AU: R Foundation for Statistical Computing.

Renjith, S., & Anjali, C. (2013a). A personalized travel recommender model based on content-based prediction and collaborative recommendation. *International Journal of Computer Science and Mobile Computing*, 66–73.

Renjith, S., & Anjali, C. (2013b). Fitness function in genetic algorithm based information filtering-a survey. *International Journal of Computer Science and Mobile Computing*, 80–86.

Renjith, S., & Anjali, C. (2014). A personalized mobile travel recommender system using hybrid algorithm. In *Proceedings of the 2014 First International Conference on Computational Systems and Communications* (ICCSC), (p. 12-17). DOI: http://doi.org/10.1109/compsc.2014.7032612

Renjith, S., Biju, M., & Mathew, M. M. (2020). A sentiment-based recommender system framework for social media big data using open-source tech stack. In V. K. Gunjan & J. M. Zurada (Eds.), *Proceedings of the International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications. Advances in Intelligent Systems and Computing,* (Vol. 1245, p. 407-414). DOI: http://doi.org/10.1007/978-981-15-7234-0_36

Renjith, S., Sreekumar, A., & Jathavedan, M. (2018). Evaluation of partitioning clustering algorithms for processing social media data in tourism domain. *In Proceedings of the 2018 IEEE Recent Advances in Intelligent Computational Systems [RAICS*] (p. 127–131). DOI: http://doi.org/10.1109/RAICS.2018.8635080

Renjith, S., Sreekumar, A., & Jathavedan, M. (2019). An extensive study on the evolution of context-aware personalized travel recommender systems. *Information Processing and Management, 57*(1), 102078. DOI: http://doi.org/10.1016/j.ipm.2019.102078

Renjith, S., Sreekumar, A., & Jathavedan, M. (2020a). Performance evaluation of clustering algorithms for varying cardinality and dimensionality of data sets. *Materials Today*: *Proceedings, 27*(1), 627-633. DOI: http://doi.org/10.1016/j.matpr.2020.01.110

Renjith, S., Sreekumar, A., & Jathavedan, M.(2020b). A comparative analysis of clustering quality based on internal validation indices for dimensionally reduced social media data. In N. Chiplunkar & T. Fukao (Eds.), *Advances in artificial intelligence and data engineering*. *Advances in intelligent systems and computing* (Vol. 1133, p. 1047-1065). Singapore, SI: Springer.

Renjith, S., Sreekumar, A., & Jathavedan, M. (2020c). Pragmatic evaluation of the impact of dimensionality reduction in the performance of clustering algorithms. In T. Sendogan, M. Murugappan & S. Misra (Eds.), *Advances in electrical and computer technologies. Lecture notes in electrical engineering (*Vol. 672, p. 499-512). Singapore, SI: Springer. DOI: http://doi.org/10.1007/978-981-15-5558-9_45

Renjith, S., Sreekumar, A., & Jathavedan, M. (2021a). SMaRT: a framework for social media based recommender for tourism. In M. Palesi, L. Trajkovic, J. Jayakumari & J. Jose (Eds.), *Second International Conference on Networksand Advances in Computational Technologies. Transactions on Computational Science and Computational Intelligence*, (p. 297-307). Cham, SW: Springer. DOI: http://doi.org/10.1007/978-3-030-49500-8_26

Renjith, S., Sreekumar, A., & Jathavedan, M. (2021b). SemRec - an efficient ensemble recommender with sentiment based clustering for social media text corpus. *Concurrency and Computation: Practice and Experience, 33*(20), e6359. DOI: http://doi.org/10.1002/cpe.6359

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics, 20*, 53–65. DOI: http://doi.org/10.1016/0377-0427(87)90125-7

Sajana, T., Rani, C. M. S., & Narayana, K. V. (2016). A survey on clustering techniques for big data mining. *Indian Journal of Science and Technology, 9*(3), 1-12. DOI: http://doi.org/10.17485/ijst/2016/v9i3/75971

Schoen, H., Gayo-Avello, D., Metaxas, P. T., Mustafaraj, E., Strohmaier, M., & Gloor, P. (2013). The power of prediction with social media. *Internet Research, 23*(5), 528–543.

Shin, D. (2021a). How do people judge the credibility of algorithmic sources?. *AI & Society, 2021*. DOI: http://doi.org/10.1007/s00146-021-01158-4

Shin, D. (2021b). The perception of humanness in conversational journalism: an algorithmic information-processing perspective. *New Media & Society, 2021*. DOI: http://doi.org/10.1177/1461444821993801

Shin, D. (2021c). A cross-national study on the perception of algorithm news in the East and the West. *Journal of Global Information Management (JGIM), 29*(2), 77-101. DOI: http://doi.org/10.4018/JGIM.2021030105

Shin, D. (2021d). Embodying algorithms, enactive artificial intelligence and the extended cognition: you can see as much as you know about algorithm. *Journal of Information Science , 2021*. DOI: http://doi.org/10.1177/0165551520985495

Shin, D. (2021e). Expanding the role of trust in the experience of algorithmic journalism: user sensemaking of algorithmic heuristics in korean users. *Journalism Practice, 2020*, 1-24. DOI: http://doi.org/10.1080/17512786.2020.1841018

Shin, D. (2021f). The effects of explainability and causability on perception, trust, and acceptance: implications for explainable AI. *International Journal of Human-Computer Studies, 146*, 102551. DOI: http://doi.org/10.1016/j.ijhcs.2020.102551

Shirkhorshidi, A. S., Aghabozorgi, S., Wah, T. Y., & Herawan, T. (2014). Big data clustering: a review. In *Proceedings of the International Conference on Computational Science and Its Applications* [ICCSA], (p. 707–720). DOI: http://doi.org/10.1007/978-3-319-09156-3_49

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Statistical Methodology. Series B, 63*(2), 411–423. DOI: http://doi.org/10.1111/1467-9868.00293

Tierney, L. (2012). The R statistical computing environment. *Lecture Notes in Statistics*, 435–447. DOI: http://doi.org/10.1007/978-1-4614-3520-4_41

Thorndike, R. (1953). Who Belongs in the Family?. *Psychometrika, 18*(4), 267–276. DOI: http://doi.org/10.1007/bf02289263

Wei, C.-P., Lee, Y. H., & Hsu, C.-M. (2003). Empirical comparison of fast partitioning-based clustering algorithms for large data sets. *Expert Systems with Applications, 24*(4), 351–363. DOI: http://doi.org/10.1016/s0957-4174(02)00185-9

Xu, R., & Wunsch II, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks, 16*(3), 645–678. DOI: http://doi.org/10.1109/tnn.2005.845141

Zepeda-Mendoza, M. L., & Resendis-Antonio, O. (2013). Hierarchical Agglomerative Clustering. In W. Dubitzky, O. Wolkenhauer, K. H. Cho & H. Yokota (Eds.), *Encyclopedia of Systems Biology* (p. 886–887). New York, NY: Springer.