A web tool for k-means clustering

Konstantinos Gratsos[1], Stefanos Ougiaroglou[1], and Dionisis Margaris[2]

[1] Department of Information and Electronic Engineering, School of Engineering, International Hellenic University, 57400 Sindos, Thessaloniki, Greece
kostisgratsos12@gmail.com, stoug@ihu.gr
[2] Department of Digital Systems, School of Economics and Technology, University of the Peloponnese, 23100, Sparta, Greece
margaris@uop.gr

Abstract. The k-means clustering finds many applications in different domains. Researchers and practitioners utilize k-means through specialized software or libraries of programming languages. This implies knowledge on these tools. This paper presents Web-k-means, a user-friendly web application that simplifies the process of running the k-means clustering and identifying the optimal number of clusters via the elbow method. Web-k-means allows users to upload datasets, select relevant features, and finally execute the algorithm. The application displays the graph generated by elbow method, suggests a value for the number of clusters and presents the cluster assignments in a data table allong with an exploratory data plot. Thus, the users can easily perform clustering analyses, without specialized software or programming skills. Additionally, Web-k-means includes a REST API that allows users to run clustering tasks and retrieve the results programmatically. The usability of Web-k-means was evaluated by the System Usability Scale (SUS) and experiments were conducted for the CPU times evaluation. The results indicate that Web-k-means is an simple, efficient and user friendly tool for cluster analysis.

Keywords: Clustering, K-means, Elbow method, Web application, Web service

1 Introduction

Clustering or unsupervised learning[1] is a common data analysis task that involves grouping instances with similar characteristics into clusters. A cluster is a set of instances in which each instance is closer (or most similar) to every instance in the cluster, rather than every instance outside the cluster. Nowadays, Clustering finds many applications in various domains, including market research, search engines, psychology and medicine, biology, etc.

Clustering was originated by Zubin and Tryon in psychology domain and Driver and Kroeber in anthropology domain in the 1930s. However, the development of clustering techniques was delayed due to computational difficulties. From the late 1950s on-wards, the rise of computing power led to the development of various clustering techniques widely used today. Therefore, computer

science plays a crucial role in clustering as it is used to perform complex calculations and processing that are necessary to identify clusters.

K-means [8] may be characterized as the most popular and widely-used clustering algorithm. It aims to group instances into k clusters based on their similarity. The algorithm is based on a iterative procedure that assigns instances to the nearest cluster centroid. A major issue that must by addressed by the user is the k parameter determination. The elbow method [7] is a popular technique for determining the optimal number of clusters. The K-means clustering with or without the elbow method has been incorporated in many specialized standalone software (e.g. Matlab, Weka [5], SPSS, Orange [4] etc) and in libraries of programming languages (e.g. scikit-learn [10]). The use of k-means and elbow method through software and programming environments may imply software licence, download and installation, specialized knowledge on the specialised software and programming skills. To the best of the authors' knowledge, there is no free web application for clustering purposes. This observation constitutes the motivation of the present work.

The contribution of the paper is Web-k-means, a user-friendly web application that enables researchers and practitioners to easily perform k-means clustering with the well-known k-means++ initialization technique [3] via the web. The elbow method has been integrated in Web-k-means and allows the user to identify the optimal number of clusters. Moreover, Web-k-means provides an exploratory plot of the cluster assignments that enables users to understand the characteristics of each discovered cluster.

The paper is organized as follows: Section 1 provides an overview of the k-means clustering. Section 3 reviews the elbow method. Section 4 presents Web-k-means in detail. System evaluation through CPU time measurements and usability testing using the System Usability Scale (SUS) questionaire are presented in Section 5, and finally Section 6 concludes the paper and outlines future work.

2 K-means clustering

K-means clustering groups instances based on similarities. It involves iteratively assigning of each instance to a cluster, based on its distance from the centroid (mean of each cluster) and then computing the updated centroid, until the centroids no longer change. The algorithm aims to minimize the sum of squared distances between data instances and their assigned cluster centroids, known as the within-cluster sum of squares (WCSS). More formally, given a set of instances, $X = \{x_1, x_2, \ldots, x_n\}$, k-means clustering aims to assign the n instances to k clusters ($k \leq n$) $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the following function (within-cluster sum of squares):

$$E = \sum_{j=1}^{k} \sum_{x_i \in S_j} ||x_i - \mu_j||^2$$

where μ_j is the mean of cluster S_j and $||x_i - \mu_j||$ is the chosen distance metric between the data point x_i and the corresponding mean.

The algorithm starts by randomly initializing k centroids, which are used as the initial cluster centers. Then, each instance is assigned to the nearest centroid, and the centroid is moved to the center of the assigned instances. This process is repeated iteratively until the centroids no longer move.

Algorithm 1 illustrates the pseudo-code for the k-means clustering technique. The algorithm works in the following manner: Firstly, it takes a dataset $X = x_1, x_2, \ldots, x_n$ and forms k clusters (C). The algorithm initially assigns each instance x_i to the cluster with the closest centroid (mean) c_j, which is termed as the assignment step (lines 4–6). After all the instances have been assigned, the algorithm recalculates the new means, by averaging the corresponding instances of the clusters (lines 7-9), which can be expressed as

$$c_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

The algorithm then re-executes the assignment step by taking into account the new means. As a result, the k-means are adjusted in each step. The algorithm terminates when the means (cluster centroids) do not change in one iteration (cluster consolidation) (line 10). Since each instance is closer to the centroid of the cluster it belongs to, the within-cluster sum of squares function is minimized.

Algorithm 1 k-means clustering

Require: $X = x_1, x_2, \ldots, x_n$ - dataset, k - number of clusters
Ensure: $C = C_1, C_2, \ldots, C_k$ - set of clusters
1: Select k initial centroids $c_1, c_2, \ldots, c_k$ randomly from X
2: **repeat**
3: Create k empty clusters $C_1, C_2, \ldots, C_k$
4: **for** $i = 1$ to n **do**
5: Assign x_i to the cluster C_j with the closest centroid c_j
6: **end for**
7: **for** $j = 1$ to k **do**
8: Update centroid c_j as the mean of all points in cluster C_j
9: **end for**
10: **until** no more changes in the assignment of points to clusters

The resulted clusters depend on the randomly selected initial centroids. In effect, k-means will discover different clusters by examining the same data but utilizing different initial cluster centroids. K-means++ is a popular initialization technique. It improves the initial centroid selection by probabilistically choosing the initial centroids that are farthest away from each other, resulting in a more accurate and efficient clustering. The algorithm starts by selecting a random instance as the first centroid, and then selects each subsequent centroid based on the distance from the previous centroids, with a higher probability of choosing instances that are farther away. This helps to avoid the common issue of k-means clustering getting stuck in sub-optimal solutions due to poor initialization.

K-means is popular and therefore, many variants have been proposed. K-medians [12] and k-modes [6] are well-known variants. K-medians clustering replaces the mean calculation with the median calculation. The main advantage is that it is less sensitive to outliers, which can cause problems in k-means. K-modes is designed to handle categorical data. In k-modes, instead of computing the mean or median, the mode (most frequent value) of each attribute in the cluster is used as the cluster center.

3 Elbow method

The elbow method is a well-known technique for determining the optimal number of clusters in k-means clustering. It works by plotting the within-cluster sum of squares (WCSS) for different values of clusters (k). The WCSS is the sum of the squared distances between each point and its assigned cluster centroid. Considering a specific k value, it is computed by summing up the squared distances for all instances within each cluster and then by finding the sum of the individual k cluster sums.

As k increases, the WCSS generally decreases, since each instance is closer to its assigned centroid. However, at a certain point, the marginal decrease in WCSS diminishes, creating a noticeable bend in the plot. This bend, or "elbow", represents the point of diminishing returns in terms of clustering performance, beyond which adding more clusters does not provide much additional benefit. The optimal number of clusters is often chosen at the elbow point on the plot. By this way, k-means discovers compact and well-separated clusters.

In Figure 1, the x-axis represents the number of clusters (k), while the y-axis represents the WCSS values. As k increases, the WCSS decreases, but at $k = 3$, there is a noticeable bend in the plot. This bend represents the "elbow" point, which indicates that adding more clusters beyond this point does not provide significant improvements in clustering performance. In this case, the optimal number of clusters might be chosen as 3.

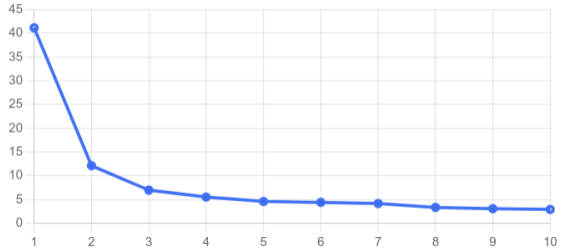


Fig. 1: Example of an Elbow Method Plot

4 The Web-k-means application

4.1 Description

Web-k-means is a user-friendly web application for cluster analysis. With Web-k-means, users can upload datasets in csv, xls or xlsx format, choose the attributes that will be used in clustering, and finally execute the k-means clustering by utilizing k-means++ initialization method. Web-k-means displays the "Elbow diagram", recommends the optimal cluster number, and presents the the instances and their assigned clusters in an exploratory plot and a tabular. The elbow graph, the exploratory plot and the dataset with the assigned clusters can be downloaded for further use and processing. Furthermore, Web-k-means provides a REST API that allows developers to submit their data, execute the elbow method and k-means clustering and obtain the cluster assignments from their own applications by incorporating simple API calls into them. The REST API is an advantageous feature of Web-k-means, expanding its usability beyond the web-based interface and allowing for more efficient and automated usage of the k-means clustering.

Web-k-means can be divided into three components: (i) modules for the elbow method and k-means clustering, (ii) a modern and user-friendly web interface, (iii) the back-end and the REST API service. The application was developed using open-source technologies and Git, with its source code available on GitHub[3]. Python was used to code the modules of the first component. Web-k-means uses the k-means implementation which is available in the scikit-learn library. Also, Pandas [9] library was utilized for datasets manipulation, and the kneed library [11] was used to obtain the number of clusters from elbow graph. PHP was used for the development the back-end as well as the REST API. The composer library was used for package management and PHPMailer for sending confirmation emails. A MySQL database was designed to manage users and their access levels (privileges). For the development of the front-end and the web interface, Javascript with the jQuery library, AJAX, and the Bootstrap framework were used. Web-k-means utilizes the server's file system to store the uploaded datasets. The REST API acts as the interface where all technologies can communicate with each other, as shown in Figure 2.

Each dataset can be either public or private. Private datasets are limited to the user who uploaded them, while public datasets are accessible to all registered users. However, only users with advanced privileges are able to upload public datasets. This feature is especially beneficial for educators aiming to share datasets with their students. The application offers three user roles: (i) simple user, (ii) public dataset creator, and (iii) administrators. Simple users are the users who can upload private dataset and use Web-k-means on them. They can also use public datasets uploaded by a public dataset creator. Public dataset creators have the same privileges with simple users but they can also upload public datasets. Administrators have the ability to upgrade a user account from simple user to public dataset creator.

[3] `https://github.com/KostisGrf/WebKmeans`

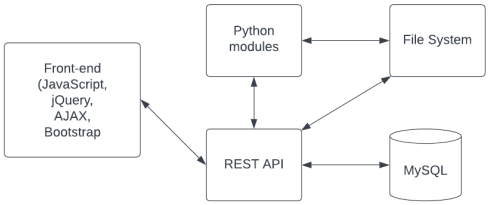


Fig. 2: Application architecture

It is worth mentioning that Web-k-means has been deployed in a web server at the department Information and Electroning Engineering of International Hellenic University[4]. It is free and open source. Therefore, users can access the source code on Github and deploy it on their own server.

4.2 Web interface

To use web-k-means, users must first register and confirm their email. Then, they can log in. The main page is divided into three distinct sections. The first section provides a web interface that enables users to either upload a new dataset on the web-k-means server or select an existing one (See Figure 3(c)). The supported datasets formats are csv, xls, and xlsx. Once a dataset has been selected, its data will be displayed in a tabular format, and the user can download or delete it (assuming that the user has the necessary permissions). Also, the names of the numerical attributes are displayed in check-boxes. Note that k-means can be applied to numerical data. The users can un-check the numerical attributes they want to be ignored in the k-means clustering.

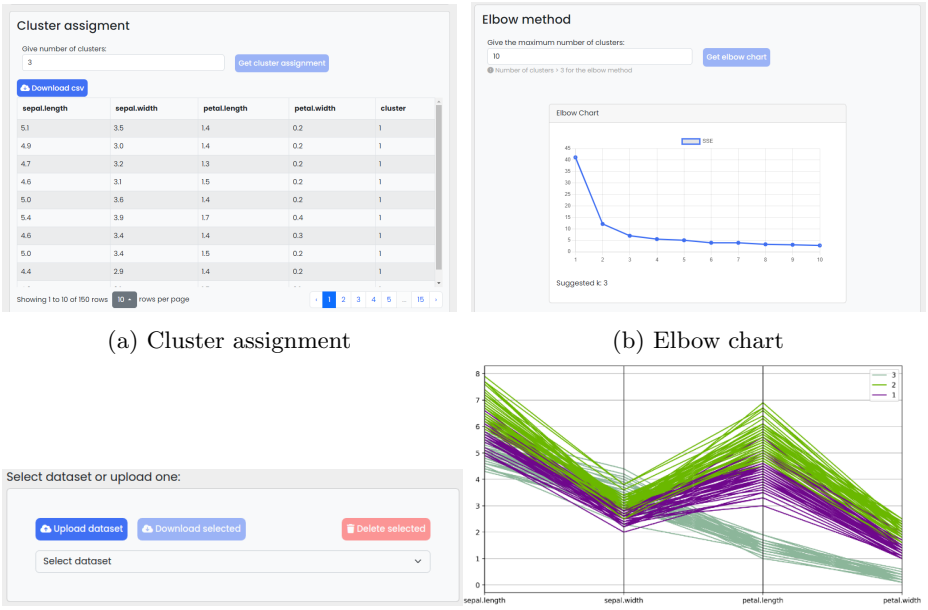
The second section concerns the elbow method. The users are able to enter the maximum number of clusters they want to examine and click "Get elbow chart". This generates the elbow graph and suggests an appropriate number of clusters in the range from 2 to the maximum number provided (See Figure 3(b)).

The last section requires users to input their desired number of clusters. This can be either the number suggested from the previous step or chosen independently by the user. By clicking "Get cluster assignment", users will receive a table displaying the data along with the cluster assignments (See Figure 3(a)). The cluster assignments can also be presented in an exploratory data plot which summarises the key characteristics of each cluster (See Figure 3(d)). The "Download CSV" button allows users to download the table in CSV format. The elbow graph and the exploratory data plot can be also downloaded.

4.3 Web service

The web service is designed to be a REST API with eleven endpoints which make the aforementioned functionalities available to other applications through

[4] `https://webkmeans.iee.ihu.gr`



(a) Cluster assignment

(b) Elbow chart

(c) Dataset selection

(d) Exploratory plot

Fig. 3: Web-k-means interface

HTTP requests. The users/programmers must sign up and acquire an API key to access the web service. Each HTTP requests must be accompanied by the user's API key. Four endpoints are dedicated to user account functions, such as sign-up, login, editing of account, and deletion. Another four endpoints are exclusively reserved for managing datasets: upload, accessing, deletion and retrieval of datasets name and characteristics. The remaining three endpoints are for executing k-means and visualizing its results. More specifically, one of these triggers the elbow method and returns WCSS values along with the recommended number of clusters. Another endpoint triggers k-means and returns cluster assignments. The last endpoint generates the exploratory plot of the clustered data. All endpoints return their results in JSON format. For example, below is a JSON request and a JSON response of the endpoint that triggers the elbow method:

```
JSON request:
{"dataset":"sample.csv","dataset-type":"personal","clusters":"10","columns":["Age","Salary"],
"apikey":"0a8366a07c0d8fccx48bab2e657f12d0"}
JSON response:
{"sse": ["41.166","12.128","6.982","5.5258","4.589","4.674","3.757","3.193","3.299","2.695"],
"suggested-k":"3"}
```

The web interface includes a web page with instructions for utilizing the eleven endpoints along with the user's API key. The web page presents examples of possible HTTP requests with the corresponding responses.

5 System evaluation

5.1 CPU time measurements

Running the k-means clustering can be intensive in terms of CPU and RAM usage. To tackle this issue, the Pandas library was employed to effectively manage the large datasets. Additionally, the scikit-learn library was utilized to enhance the algorithm's execution efficiency.

In order to measure the performance of Web-k-means, we conducted an experimental study by using six datasets distributed by the keel dataset repository [2][5]. The experimental measurements are obtained by executing the elbow method for 20 clusters and the k-means clustering using the suggested number of clusters through the web interface. The experimental results are presented in Table 1. As we expected, the results indicate that the execution times are directly affected by the size and the number of rows and columns of each dataset. It is worth mentioning that the execution of the elbow method is more time consuming than the execution of the k-means clustering. This happens because the elbow method executes clustering several times (from k=1 to k=20 in the case of our experimentation) and compute a WCSS value for each clustering task. Poker is a quite large dataset. It has over a million rows. The elbow method took more than a minute to run, while the k-means clustering took only 24 seconds. The measurements obtained by using such large datasets can be improved by hosting Web-k-means to a more powerful computer.

Table 1: CPU time measurements

Dataset	**Size (Kb)**	**Number of rows**	**Number of columns**	**Time for elbow (k=20)**	**suggested k**	**Time for k-means**
penbased	538	10,992	16	1.06s	6	0.24s
letter	716	20,000	16	2.49s	7	0.35s
magic	1,462	19,020	10	1.67s	5	0.23s
texture	1,495	5,500	40	1.02s	4	0.45s
shuttle	1,559	57,999	9	3.08s	5	0.31s
poker	24,563	1,025,009	10	72s	6	23.25s

5.2 Usability testing

The usability of Web-k-means was evaluated using the System Usability Scale (SUS) questionnaire[6]. More specifically, SUS was used to measure the users overall satisfaction with Web-k-means, as well as their perception of its effectiveness, efficiency, and ease of use. SUS is a widely used tool for measuring the usability

[5] `https://sci2s.ugr.es/keel/datasets.php`
[6] `https://forms.gle/dUSecKe1pgES661z8`

of web applications. It consists of ten questions that participants complete to rate their experience. The questions of the questionnaire are presented bellow.

1. I think that I would like to use this website frequently
2. I found the website unnecessarily complex
3. I thought the website was easy to use
4. I think that I would need the support of a technical person to be able to use this website
5. I found the various functions in this website were well integrated
6. I thought there was too much inconsistency in this website
7. I would imagine that most people would learn to use this website very quickly
8. I found the website very cumbersome to use.
9. I felt very confident using the website
10. I needed to learn a lot of things before I could get going with this website

Each question is rated on a 5-point Likert scale ranging from "strongly disagree" (1) to "strongly agree" (5). To calculate the SUS score, for each of the odd numbered questions, 1 is subtracted from the score. For each of the even numbered questions, their value is subtracted from 5. The resulting scores are then added up and multiplied by 2.5 to give a final score between 0 and 100. A SUS score of 80 or above is considered excellent.

We requested people to complete the questionnaire, and 22 of them did so. Most of them were computer science undergraduate students who attend a data mining course. Table 2 presents the response count for each range. The SUS score is 83.4. Therefore the results of SUS illustrate that the users are satisfied with the experience of using Web-k-means.

Table 2: Results of System Usability Scale (SUS) Questionnaire

Question	**1**	**2**	**3**	**4**	**5**
1.	0	2	3	9	8
2.	14	7	0	0	1
3.	0	0	1	4	17
4.	14	6	2	0	0
5.	0	0	3	7	12
6.	18	4	0	0	0
7.	1	0	4	4	13
8.	6	8	8	0	0
9.	0	0	1	7	14
10.	6	4	9	2	1

6 Conclusions and future work

The paper presented Web-k-means, a user-friendly web application that allows researchers and practitioners to easily perform k-means cluster analysis with

the kmeans++ centroid initialization method. The elbow method has been integrated in order to give the user the ability to identify the optimal number of clusters. With Web-k-means, users can upload datasets, choose the attributes that will be used in clustering, and conduct cluster analysis with k-means. The application displays the elbow graph, recommends the optimal cluster number, and presents the clustering results in a tabular form and an exploratory data plot. Additionally, Web-k-means provides a REST API that allows developers to submit their data, execute the elbow method and the k-means clustering, and obtain the clustering results from their own applications by incorporating simple API calls into them.

In our future work, we plan to extend Web-k-means by integrating k-medians and k-modes. Also, we plan to integrate a mechanism for data pre-processing tasks, such as missing values imputation and normalization.

References

1. Aggarwal, C.C., Reddy, C.K.: Data Clustering: Algorithms and Applications. Chapman Hall/CRC, 1st edn. (2013)
2. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic & Soft Computing **17**(2-3), 255–287 (2011)
3. Arthur, D., Vassilvitskii, S.: K-means++: The advantages of careful seeding (2007)
4. Demšar, J., Curk, T., Erjavec, A., Črt Gorup, Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., Štajdohar, M., Umek, L., Žagar, L., Žbontar, J., Žitnik, M., Zupan, B.: Orange: Data mining toolbox in python. Journal of Machine Learning Research **14**, 2349–2353 (2013)
5. Frank, E., Hall, M.A., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I.H.: Weka: A machine learning workbench for data mining., pp. 1305–1314. Springer, Berlin (2005), `http://researchcommons.waikato.ac.nz/handle/10289/1497`
6. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery **2**(3), 283–304 (1998)
7. Kodinariya, T.M., Makwana, P.R.: A review on the Elbow method in clustering. International Journal of Computer Applications **1**(6), 97–100 (2013)
8. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability **1**(14), 281–297 (1967)
9. Wes McKinney: Data Structures for Statistical Computing in Python. In: Stéfan van der Walt, Jarrod Millman (eds.) Proceedings of the 9th Python in Science Conference. pp. 56 – 61 (2010). https://doi.org/10.25080/Majora-92bf1922-00a
10. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
11. Satopaa, V., Albrecht, J., Irwin, D., Raghavan, B.: Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In: 2011 31st International Conference on Distributed Computing Systems Workshops. pp. 166–171 (2011)
12. Sengupta, J.S., Auchter, R.F.: A k-medians clustering algorithm. Applied Statistics **39**(1), 67–79 (1990). https://doi.org/10.2307/2347822