Subtractive Mountain Algorithm for Data Processing

Daniel-Calin POPEANGA, Alexandru BOICEA, Yasser AL HADAD

Computer Science Department, Faculty of Automatic Control and Computers, National University for Science and Technology Politehica Bucharest (e-mail: daniel.popeanga@upb.ro, alexandru.boicea@upb.ro, yasser.al@stud.acs.upb.ro).

Abstract: The data clustering method is used in many data processing applications. This paper presents the Subtractive Mountain clustering algorithm which is based on adaptive Jaccard Index as a distance function. A case study is presented to test and analyze the performance of the algorithm compared to other algorithms. Ground truth method is used for evaluation.

Keywords: data clustering, density-based algorithm, ground truth method, Subtractive Mountain clustering algorithm

1. INTRODUCTION

Many clustering algorithms that belong to different libraries were tested using Python scripting language but not all algorithms can perform text clustering task. Most of them fail to perform the task in a reasonable time, and others generate errors. Only K-means and bisecting K-means clustering algorithms succeed to generate satisfying results. Both algorithms belong to "sklearn" library. The documentation doesn't mention the distance function that is used, so mostly it is Euclidean distance.

Subtractive Mountain algorithm is a density-based clustering algorithm that demonstrates its performance in wide range of applications. It is used to estimate the number of clusters (Ansari et al., 2015; Hsieh et al., 2021) and also to select initial cluster centers (Kurukuru et al., 2020). This initialization optimizes the results of K-means in discovering web usage clusters (Singh et al., 2015) and segmentation of image (Tongbram et al., 2021). Subtractive Mountain Algorithm is used also as a standalone method for natural languages processing (Clar et al., 2021) and for optimizing the performance of MapReduce Model for Big Data Analytics (Zanoon et al., 2020). This research investigates the potential of Subtractive Mountain clustering algorithm in text clustering.

The evaluation metrics of clustering are generally based on compactness within clusters and separation between clusters, but this approach are not useful for clusters with non-convex shape. Text clustering serves useful intelligent applications. These applications require clustering algorithms with the ability to recognize different classes within a dataset. Therefore ground truth method is considered in this research to provide a real evaluation regarding the classification quality of clustering algorithms.

The most popular clustering algorithm is K-means with Euclidean distance. Jaccard Distance is used as a dissimilarity measure between records to provide the initial position of K centroids to optimize the final result of K-means (Shameem et al., 2009). Depends on applied n-gram for tokenization, text datasets can have more dimensions (tokens) than records, therefore, in this research, a short version of Jaccard index is considered as an intuitive distance function for measure the similarity between the records of text type.

2. SUBTRACTIVE MOUNTAIN CLUSTERING ALGORITHM

Subtractive Mountain is a density-based clustering algorithm that simulates the human perspective in recognizing clusters (Yang and Wu, 2005).

Fig. 1 illustrates the main concept of the Mountain function (Al Hadad, 2018). The effect of mountain value in recognizing the clusters is demonstrated using mesh diagram and contour diagram. It uses the Iris dataset to show the effect of mountain function in two and three dimensions. Fig. 1-A and Fig. 1-B show the distribution of Iris data with two big obvious clusters. Fig. 1-C and Fig. 1-D show the result of computing the Mountain function using contour and mesh diagram.The equations below reflect the original algorithm. The algorithm consists of several steps as follows:

1- Computing Mountain value for every point, more points around the target point at a smaller distance, the higher the mountain value is for this point. Mountain function (*Moun*) is described in equation 1 below.

$$Moun_{k=1}(p_i) = \sum_{j=1}^{n} e^{-\alpha d(x_j,\ p_i)} \quad (1)$$

The index of 'k' attached to the function ($Moun_{k=1}$) with value one refers to the first iteration. p_i represents the targeted point and x_j is a point of data set. Euclidean distance $d(x_j, p_i)$ is the distance between a point and the targeted point of which mountain value is computed. α is a positive constant that should be provided which acts like a radius that assigns a high weight to points located within this radius.

2- The point with maximum mountain value is assigned as a centroid (center of cluster) of the respective 'k' iteration as it is shown in equation 2 below. 'p_k^*' refers to selected point to be a centroid at 'k' iteration.

$$Moun_k(p_k^*) = max_i\{\ Moun_k(p_i)\ \} \quad (2)$$

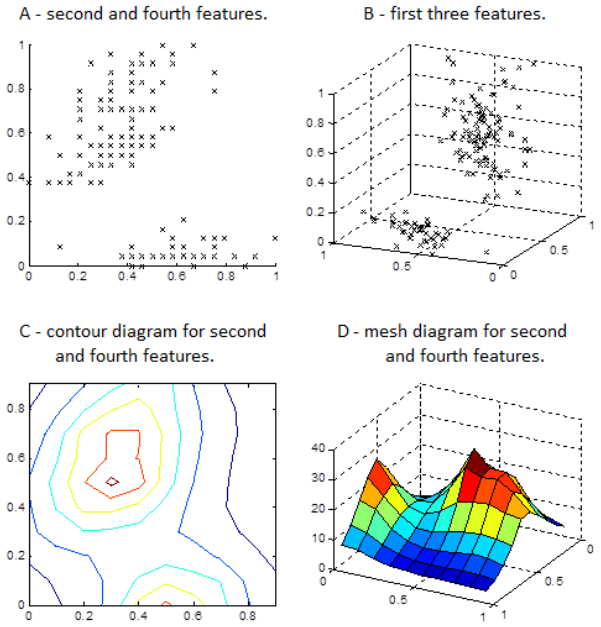

Fig. 1. Representation of mountain value in clusters recognition.

3- The Reverse_Mountain_Function is called after selecting a centroid. Its role is to decrease the mountain value for points that are close to the previous selected centroid in step 2 to avoid selecting another centroid that belong to the same cluster. Revised mountain function is described in equation 3 as follows.

$$Moun_K(x_i) = Moun_{k-1}(x_i) - \left(Moun_{k-1}(p^*_{k-1}) \ . \ e^{-\beta d(x_i, P^*_{k-1})} \right) \quad (3)$$

The mountain value ($Moun$) of a certain point 'x_i' at iteration k is computed by subtracting the mountain value of the selected centroid at the previous iteration ($Moun_{k-1}(p^*_{k-1})$), multiplied by ($e^{-\alpha d(x_i, P^*_{k-1})}$), from the mountain value of this point at the previous iteration $Moun_{k-1}(x_i)$. '$d(x_i, P^*_{k-1})$' is the distance between 'x_i' point and last selected centroid 'P^*_{k-1}'. β is a positive constant that should be provided by the user. In this study, value of zero is assigned to both α and β. It eliminates the need of manual setting of these parameters. It also simplifies the equations and optimizes the processing time.

4- Repeat step 3 until a predefined stop criterion is fulfilled. Stop's criterion (δ) is a constant positive parameter. Equation 4 describes stop criterion below. In this study, the number of required clusters is provided by user, so Equation 4 is not used.

$$\frac{Moun_k(p^*_k)}{Moun_1(p^*_1)} < \delta \quad (4)$$

3. CLUSTERING PERFORMANCE EVALUATION

Measuring the quality of a clustering method is important for evaluation of the clustering results. It can be performed in two ways: internal and external quality measurements.

For internal quality measurements the cluster formation assessment is compared only to the results themselves, ie the structure of the clusters detected and their interrelationships. The two main concepts are compactness and isolation. Compactness measures how tightly the data points are grouped within a cluster. Separation measures are how different the found clusters are from each other. More formally, compactness is the within-cluster variance and separation is the distance between clusters.

The main internal quality measures are based on Silhouette score. It uses intra-cluster distance (compactness) and inter-cluster distance (separation) to measure the overall score.

Silhouette score is an evaluation metric that is used to calculate the quality of a clustering method. It uses intra-cluster distance (compactness) and inter-cluster distance (separation) to measure the overall score (Tushar, 2021). Silhouette score of a single data-point is computed for every data point as it is shown in equation 5, then, the mean value of all samples is returned as a Silhouette score.

$$S_i = b_i - a_i / max(b_i , a_i) \quad (5)$$

Where b_i is the average distances between point 'i' and the points that belong to the nearest cluster that 'i' is not a part of and a_i is the average distances between point 'i' and the points that belong to the same cluster that 'i' is a part of.

The overall silhouette score is equal to the mean silhouette score of all points. The number of clusters should be 2 at least and smaller than the number of data-points.

There are many metrics that are used to evaluate clustering results based on compactness within clusters and separation between clusters.

External quality measurements requires external knowledge and it can used more methods.

In this paper the ground truth method is used to provide a better evaluation regarding the clustering quality.

Every tested dataset has records classified in two classes (labels) ex. Positive and negative comments, or good and bad products. Unigram is used for tokenization. A good clustering algorithm should recognize the difference between classes. A good cluster should include records from a single class. The worst result is in case that a cluster has records from both classes equally. This means that this particular cluster fails to recognize the difference between classes. The records of certain class that appear the most in a cluster are considered the well-classified records of the cluster. The implemented ground truth method is linear in which well-classified records in all clusters are summed, and then divided by the tot number of records belonging to the dataset (both classes).

4. ADAPTIVE IMPLEMENTATION OF SUBTRACTIVE MOUNTAIN ALGORITHM FOR TEXT CLUSTERING

The algorithm was tailored to fit the task of text clustering. The pre-processing phase of text datasets involves text filtering (stop words, removing URL links and numbers, etc.), stemming, and SMOTE (Synthetic Minority Over-sampling Technique). It was implemented using Python scripting language as follows:

1- Feature extraction: simple tokenization method is applied. It ignores the frequency of words, and generates a matrix in which every row represents a text record (comment, article, etc.) and every column represents a word. Every cell has value one or zero to indicate whether the word appears in the record or not as it is seen in Figure 2.

2- A short version of Jaccard index is used as a distance function. More specifically, the numerator part that represents the number of common words between two records as a distance value. The distances between every two records are calculated by multiplying the tokenized matrix by its transpose matrix to generate the distance matrix (Y), see Figure 2. Zero distance value means that there are no common words between two records (no similarity) whereas a high integer value means that the two records are close to each other (similar to each other).

3- Every row from the tokenized matrix represents a record from dataset. The sum of the every row of the distance matrix represents the mountain value of the record represented by that row, see Figure 2.

$$X = \begin{bmatrix} 1 & 1 & 0 & & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & 1 \\ & \vdots & & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

The resulted tokenized matrix: (X)
Distance matrix: ($Y=X^TX$) multiplying the tokenized matrix by its transpose matrix to count the number of common words that appear between every two records.
Mountain value: $y_i = \sum_{j=0}^{n} yij$) by computing the sum of every row (record) of the distance matrix (Y)

Fig. 2. Adaptive implementation of Subtractive Mountain algorithm for text clustering.

4- The Reverse_Mountain_Function removes all words (columns from the distance matrix) that appear in the previously selected centroid rather than applying a function or percentage.

The research aims to demonstrate the ability of Subtractive Mountain clustering algorithm to perform text clustering tasks. A tailored implementation was written in Python to use it further in this research.

The code includes three methods:

- Mountain_Function is used to compute the mountain value for all points. The numerator part of Jaccard index is considered as distance function. It represents the sum of all common words between every two records. The distance matrix (Y) that includes all distances is computed by multiplying the tokenized matrix by its transpose matrix. The sum of the distances between a record and all other records constitutes the mountain value. It is computed by taking the sum of every row of the distance matrix.
- Fit_Function selects the point with highest mountain to be the first centroid. Then remaining iterations are applied using reverse function until the stop criterion is met. Finally, the remaining records are assigned to the selected centroids using the distance function.
- Reverse_Mountain_Function removes the words that appear in the previously selected centroids to assure that the next selected centroid does not belong to a cluster that was already selected by a previous iteration. Then it recomputes mountain value to choose the following centroid.

5. CASE STUDY

This section aims to demonstrate the potential of the adaptive implementation of Subtractive Mountain clustering algorithm that is described in previous section in text clustering tasks. Three datasets are used in this study, range between 5800 – 32000 records without SMOTE. Every dataset has a text column as input, and a labeled column as output that is used to assess the clustering result using ground truth method. Datasets are listed as follows:

Table 1. A few records as examples from dataset 1 - Twitter Sentiment Analysis.

	label	tweet
13	1	@user #cnn calls #michigan middle school 'buil...
14	1	no comment! in #australia #opkillingbay #se...
17	1	retweet if you agree!
23	1	@user @user lumpy says i am a . prove it lumpy.
34	1	it's unbelievable that in the 21st century we'...
...	...	...
2404	0	found my future husband on the plane to marbs ...
2405	0	, excited, clapping, unbreakable kimmy schmid...
2406	0	not long until the england v wales match! the ...
2407	0	nude rear naughty naked school girls
2408	0	"uk now has the highest average level of stude...

4484 rows × 2 columns

- **Dataset 1**: Twitter Sentiment Analysis (32000 records) – 'Sentiment140.com' is the source of the dataset (Al Hadad, 2018). Data has two columns. There are two values in the "label" column: 0 for positive tweets and 1 for negative. The second column is for text-based comments. There are 2242 records in this dataset with value of 1, and another 29720 records are with value of 0. Table 1 shows some records of dataset 1.
- **Dataset 2**: 20 Newsgroups dataset consists of about 20,000 newsgroup documents that have been distributed (almost) evenly among 20 different newsgroups. It is provided by Ken Lang in (Lang, 1995). It becomes a common dataset for experimentation in text applications of machine learning techniques, such as text classification and text clustering. In this research, 6 news groups are considered that include 5800 records or articles. Table 2 shows the chosen topics for every label.

Table 2. Total records per label and their relation topics for dataset 2 - 20 Newsgroups.

Label 0 (Computer Components) - 2588 records
1- comp.sys.ibm.pc.hardware
2- comp.sys.mac.hardware
3- comp.os.ms-windows.misc
Label 1 (Religion) – 2766 records
1- soc.religion.christian
2- talk.religion.misc
3- alt.atheism

- **Dataset 3**: Amazon Fine Food Reviews (10000 records), available in Kaggle.ro web site (Amazon, 2023). It is also published and documented (McAuley et al., 2013). The dataset have three columns, see Table 3. Two columns for comments (Summary and text) are used in text clustering task, and 'Score' column with values that range from one to five, one for negative review, and five for positive review.

Table 3. A few records as examples from dataset 3 - Amazon Fine Food Review.

	Score	Summary	Text
1	1	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
12	1	My Cats Are Not Fans	My cats have been happily eating Felidae Plati...
26	1	Nasty No flavor	The candy is just red , No flavor . Just plan...
50	1	Don't like it	This oatmeal is not good. Its mushy, soft, I d...
62	1	stale product.	Arrived in 6 days and were so stale i could no...
...	...	...	...
1419	5	the family loved it	if u have not tried this u r missing the best ...
1420	5	Best for on the road	Cooked in healthy coconut oil that's why it st...
1424	5	Vintage Tasting Coffee	Forget Starbucks, Peets or anything else. This...
1425	5	Coffee from Amazon.com	This coffee is the smoothest dark roast coffe...
1426	5	Best Coffee Ever	I love this coffee! Like the other reviewers, ...

The number of records increases after applying SMOTE process. Multiple n-gram tokenization types are using starting from unigram up to 7-gram. The Subtractive Mountain algorithm based on adaptive Jaccard Index as a distance function, K-means and Bisecting K-means algorithms based on Euclidean distance were tested on these three Datasets.

6. TEXT PRE-PROCESSING

Data pre-processing task involves several steps as it was mentioned in section 4, these steps were performs in the following order:

1- Removing HTML tags.

2- Removing punctuation and special characters ('.', '?', '!', '#', etc.).

3- Removing stop words. It usually refers to group of words that do not have an impact of classification task such as "the", "is", "at", etc. The list of stop words was provided by "nltk" library.

4- Stemming: It transforms text into canonical (stem) form, For example "drives" and "driving" can be represented by the stem word "drive".

5- SMOTE function: it duplicates the number of instances of certain label in order to equalize the number of records that belongs to every output label.

Figure 3 shows text pre-processing result of dataset 3 before applying the SMOTE. Class "WordCloud" from Python is used to show the words within every label. The most frequent words within every label are highlighted with big font. Label 1 includes positive tweets, and label 2 includes negative tweets.

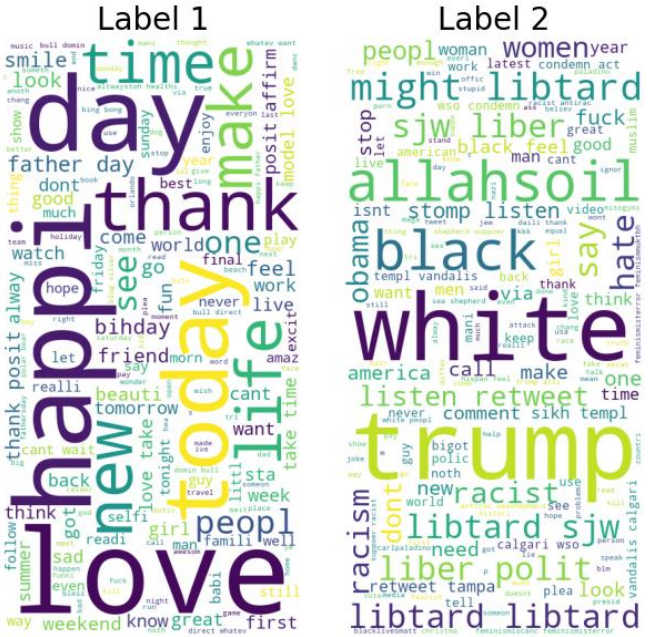

Fig. 3. Result of text pre-processing grouped by labels using Word-Cloud graph.

7. INITIAL ANALYSIS OF DATASETS

All three selected datasets in this study contains labels with two classes. Tokenization process generates a matrix in which every token is considered a new dimension. The number of tokens increases as n-gram increases too, see Table 4. During this research many n-grams were tested. The results of clustering using uni-gram tokenization are listed in detail in the following sections. Also some results from the combined uni-gram and bi-gram (or all to bi-gram) tokenization are listed and compared to the results of simple uni-gram tokenization. Finally, all to 7-gram tokenization that combined all n-grams up to 7-gram is used in order to test the algorithm processing efficiency in the most extreme situations.

Silhouette score is computed using the labels that are provided by every dataset. The results shows at Table 5 that Silhouette score is closer to zero than one, which indicates that there is no correlation between input and output, or indicates overlapping cluster. In case of "Twitter Sentiment Analysis" and "Amazon Fine Food Reviews" datasets, the silhouette score is a little higher due to the effect of SMOTE function that duplicate the number of instances in order to equalize the number of records that belongs to every output label.

Table 4. Initial analysis of datasets (Tokens count).

N-gram	Tokens count		
	Dataset 1: Twitter Sentiment Analysis [Records: 59440]	Dataset 2: 20 Newsgroups dataset [Records: 5532]	Dataset 3: Amazon Fine Food Reviews [Records: 12366]
Uni-gram	29,309	26,756	10,498
Bi-gram	136,533	365,445	153,030
All to bi-gram (Uni-gram & Bi-gram)	165,842	392,201	163,528
All to 7-gram (uni-gram, …, 7-gram	640,019	3,136,111	1,302,870

Table 5. Initial analysis of datasets (Silhouette score).

N-gram	Silhouette score		
	Dataset 1: Twitter Sentiment Analysis [Records: 59440]	Dataset 2: 20 Newsgroups dataset [Records: 5532]	Dataset 3: Amazon Fine Food Reviews [Records: 12366]
Uni-gram	0.137208	0.004858	0.06001
Bi-gram	0.192844	0.000856	0.170175
Tri-gram	0.185982	0.000851	0.210053
4-gram	0.176098	0.000863	0.218475
5-gram	0.16092	0.000901	0.221376
6-gram	0.151899	0.000884	0.220465
7-gram	0.150956	0.000914	0.218531
All to bi-gram	0.171491	0.003358	0.090338
All to tri-gram	0.174838	0.002769	0.108162
All to 4-gram	0.175964	0.002383	0.121263
All to 5-gram	0.178294	0.002048	0.128437
All to 6-gram	0.175769	0.001874	0.135652
All to 7-gram	0.178382	0.001731	0.139922

8. RESULTS AND ANALYSIS (UNI-GRAM)

The ground truth method reveals that subtractive mountain algorithm with adaptive Jaccard index as a distance function is better than K-means and Bisecting K-means with Euclidean distance.

K-means method is primarily based at the enter parameter k and the records are split into k clusters. The algorithm uses an iterative method to determine the k centroids positions, records are grouped in k clusters, and every cluster centroid might be used as a reference for the followed iteration. Iteration makes the chosen reference nearer to the proper cluster centroid, so the clustering result is getting better (Liu, 2011). K-means approach has suitable impact at the convex cluster; can run in parallel. On the other hand, there are some disadvantages that can listed as follows: require the number of clusters as an input; Cannot find out the non-sphere cluster; not able to acquire most appropriate answer; be at risk of the interference of unusual factors; be loss of scalability; clustering consequences are on occasion unbalanced.

Ground truth method is used to evaluate clustering quality. Figure 4 shows that the minimum quality result occurs in case of grouping the records in two clusters (the start of the graph). Then, the performance of clustering increases as the number of clusters increases as well. Figure 5 shows that processing time increases as the number of clusters increases as well. Every dataset has records that are classified in two labels. An example of detailed clustering result is shown by Figure 5. It is clarify the way that records are distributed in clusters, and the number of records that belongs to every label (output). The results reveal the following observations:

1- The Subtractive Mountain clustering algorithm is stable (without fluctuation) because there are no parameters that are initialized randomly, so it is generated the same result at every run as it is seen in Figure 4, Figure 8 and Figure 12.

2- The Subtractive Mountain clustering algorithm achieved the minimum processing time at all datasets as it is seen in Figure 5, Figure 9 and Figure 13. This can be due to the following reasons:

   - The simplicity of distance function.
   - Using the native matrix multiplication function in Python.
   - The number of iterations doesn't exceed the number of clusters.

3- Overall, Figure 4, Figure 8 and Figure 12 show a better performance for Subtractive Mountain clustering algorithm than the others algorithms. However, it should be a further investigation to check whether it is the effect of distant function that is used for Subtractive Mountain algorithm or the effect of the algorithm itself.

4- Figure 7, Figure 11 and Figure 15 show that Subtractive Mountain algorithm tends to distribute the records in more clusters than the other algorithms.

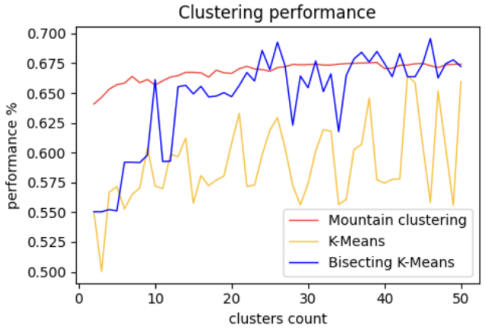

Fig. 4. Comparison of clustering quality of dataset1 (uni-gram).

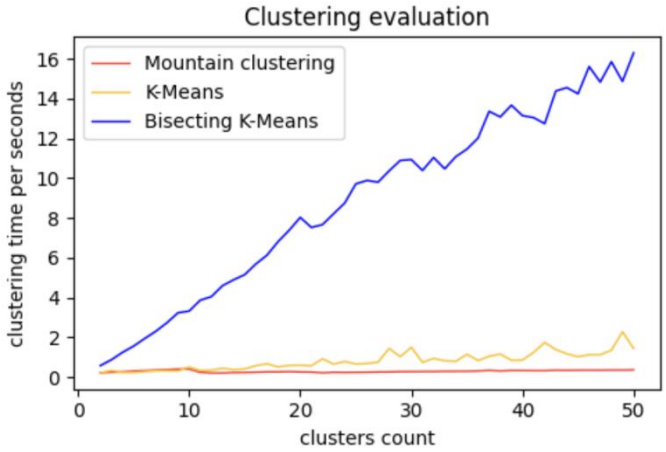

Fig. 5. Processing time comparison of clustering dataset1 (uni-gram).

cluster nr.	label 1	label 2	records in cluster
6	23352	28160.0	
8	2513	140.0	
2	2447	20.0	
5	721	288.0	
0	554	357.0	
9	12	746.0	
3	110	0.0	
4	8	9.0	
1	2	0.0	
7	1	0.0	
cluster nr.	label 1	label 2	records in cluster

Fig. 6. Detailed clustering result (dataset 1) of 10 clusters using bisecting K-Means (uni-gram) {performance 59.30% - silhouette score: 0.163}.

cluster nr.	label 1	label 2	records in cluster
0	10294	23809	
3	1699	2137	
7	3356	361	
5	2771	671	
9	2780	321	
1	2740	178	
6	2272	558	
2	2082	344	
8	1625	355	
4	101	986	
cluster nr.	label 1	label 2	records in cluster

Fig. 7. Detailed clustering result (dataset 1) of 10 clusters using Subtractive Mountain algorithm (uni-gram) {performance 74.96% - silhouette score: 0.130}.

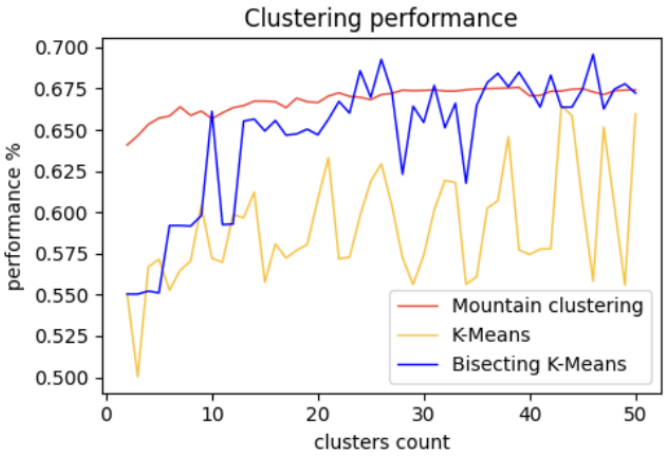

Fig. 8. Comparison of clustering quality of dataset 2 (uni-gram).

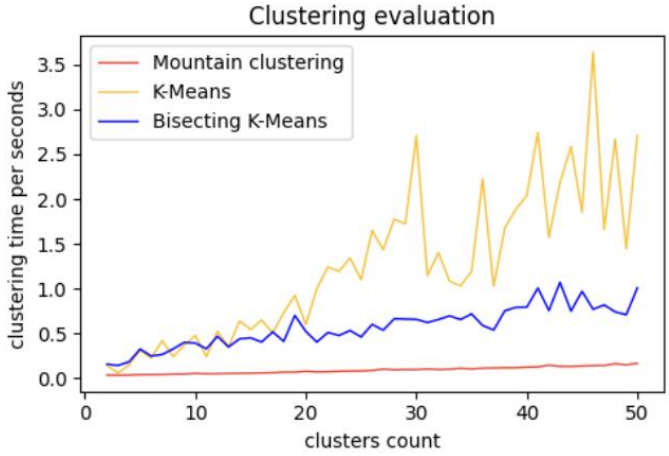

Fig. 9. Processing time comparison of clustering dataset 2 (uni-gram).

cluster nr.	label 1	label 2	records in cluster
7	1007.0	647.0	
8	427.0	1003.0	
4	591.0	344.0	
5	259.0	543.0	
1	409.0	162.0	
0	71.0	34.0	
3	0.0	27.0	
2	1.0	4.0	
6	0.0	2.0	
9	1.0	0.0	
cluster nr.	label 1	label 2	records in cluster

Fig. 10. Detailed clustering result (dataset 2) of 10 clusters using bisecting K-Means (uni-gram) {performance 66.12% - silhouette score: 0.039}.

cluster nr.	label 1	label 2	records in cluster
7	734	1371	
9	979	299	
8	336	540	
2	254	272	
6	131	130	
5	152	42	
3	99	27	
0	53	53	
1	11	19	
4	17	13	
cluster nr.	label 1	label 2	records in cluster

Fig. 11. Detailed clustering result (dataset 2) of 10 clusters using Subtractive Mountain algorithm (uni-gram) {performance 65.67% - silhouette score: 0.060}.

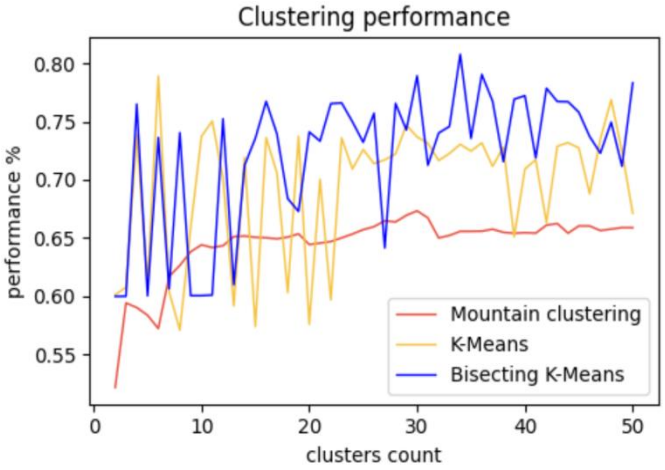

Fig. 12. Comparison of clustering quality of dataset 3 (uni-gram).

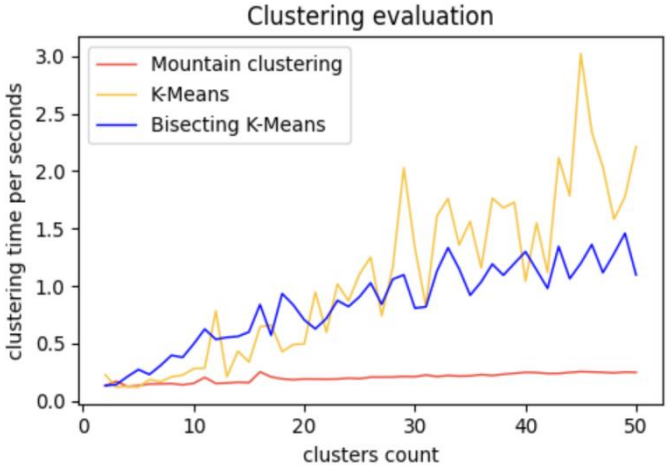

Fig. 13. Processing time comparison of clustering dataset 3 (uni-gram).

cluster nr.	label 1	label 2	records in cluster
6	1279.0	3720.0	
1	2853.0	410.0	
0	1223.0	1689.0	
3	826.0	337.0	
2	0.0	11.0	
5	0.0	10.0	
7	0.0	5.0	
4	1.0	0.0	
9	1.0	0.0	
8	0.0	1.0	
cluster nr.	label 1	label 2	records in cluster

Fig. 14. Detailed clustering result (dataset 3) of 10 clusters using K-Means (uni-gram) {performance 73.73% - silhouette score: -0.013}.

cluster nr.	label 1	label 2	records in cluster
4	1183	874	
9	1177	601	
2	784	992	
8	1098	517	
3	485	813	
0	170	1011	
6	534	347	
1	230	636	
5	297	168	
7	225	224	
cluster nr.	label 1	label 2	records in cluster

Fig. 15. Detailed clustering result (dataset 3) of 10 clusters using mountain algorithm (uni-gram) {performance 64.42% - silhouette score: -0.164}.

9. RESULTS AND ANALYSIS (MULTIPLE-GRAMS)

This section uses all to bi-gram (uni-gram & bi-gram) tokenization in clustering as well as all to 7-gram as stress test. The results reveal that for every n-gram added to tokenization, the performance of subtractive mountain algorithm increases slightly. On the other hand, K-means struggles significantly (Fig. 16 and Fig. 17), Even in case of dataset 3 (Fig. 12) in which the algorithm shows superiority in uni-gram, this superiority was lost due to the decrease of overall performance. Moreover, the result was much worse for k-means with “all to tri-gram” test as it is seen in Fig. 18. Finally, the most extreme case in this research was considered as a stress test using the configuration that is detailed in table 6. The results reveal the following observations:

1- Subtractive Mountain has minimum processing time.

2- Multiple n-grams increases the performance of Subtractive Mountain slightly.

3- Multiple n-grams decrease the performance of K-means & bisecting K-means significantly.

Table 6. Processing time comparison (time in seconds).

All to 7-gram Clusters number : 50			
dataset	**1**	**2**	**3**
Records count **Tokens count**	**59440** **640,019**	**5532** **3,136,111**	**12366** **1,302,870**
Subtractive Mountain	1.96	8.96	4.06
K-means	7.15	26.27	9.33
Bisecting K-means	32.16	31.47	14.78

Fig. 16. Comparison of clustering quality of dataset 1 (uni-gram & bi-gram).

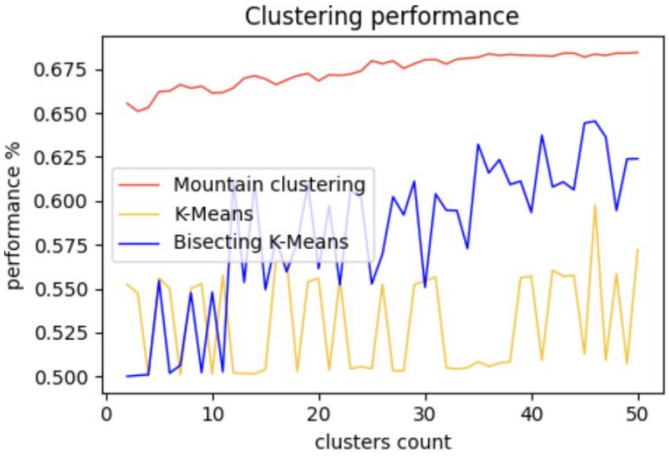

Fig. 17. Comparison of clustering quality of dataset 2 (uni-gram & bi-gram).

Bisecting K-means seems relatively better performance than the K-means, but it fluctuated. On the other hand, the processing time of Bisecting K-means was lower than the K-means. Based on the results, Bisecting K-means is the second-best algorithm. Both the K-means and Bisecting K-means algorithms get multithread processing and succeeded in performing the clustering task of the given datasets in a reasonable time.

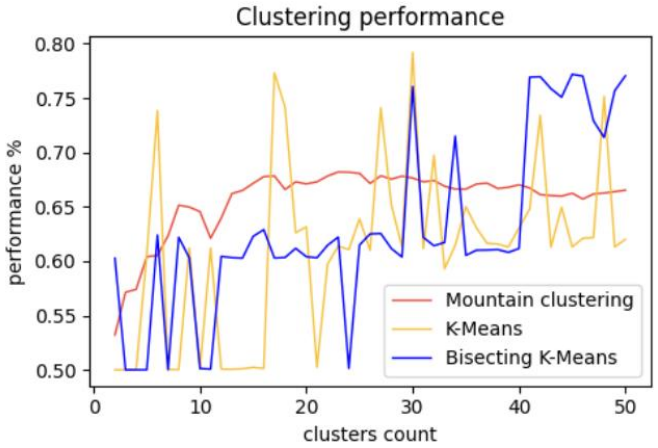

Fig. 18. Comparison of clustering quality of dataset 3 with all to tri-gram tokenization (combined uni-gram & bi-gram).

Algorithms performance depends on memory and CPU resources. The optimization of the distance function greatly reduces the processing time, even without using the multi-threads feature.

Moreover, finding a way to increase the quality of clustering is not easy. Only some of the results are labelled. In order to get a good evaluation of the clustering quality, we used three datasets with labels and a linear method based on the ground truth approach to get a clear insight.

10. CONCLUSIONS

The implementation of Subtractive mountain algorithm, presented in this research, has the best processing time. Even without the implementation of multi-threading feature, while other algorithms have the advantage of multi-threading feature. Using multiple n-grams improves the performance of clustering quality slightly, while for other algorithms the performances decrease. Subtractive Mountain clustering algorithm is a density-based clustering algorithm, so it has the ability to recognize non-convex clusters forms whereas K-means can't. Therefore ground truth method was used for evaluation instead of the metrics that are based on cohesion and separation, such as Silhouette that are not useful in this situation. The ground truth method reveals that subtractive mountain algorithm with adaptive Jaccard index as a distance function is better than K-means and Bisecting K-means with Euclidean distance.

REFERENCES

Al Haddad, Y., (2018), A Modified mountain algorithm for clusters centers estimations, The International Conference on Informatics in Economy IE 2018, Iasi, Romania, pp. 521-426, Vol. 17, ISSN 2284-7472.

Amazon Fine Food Reviews, (2023). Url: https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews?select=Reviews.csv, [Cited: 2023 May 22].

Ansari, Z. A., Sattar, S. A., Babu, V, Azeem M. F., (2015). Mountain density-based fuzzy approach for discovering web usage clusters from web log data. Fuzzy Sets and Systems, vol. 279, pp. 40-63, DOI: 10.1016/j.fss.2015.01.021.

Liu, B., (2011), Web Data Mining, Springer-Verlag Berlin Heidelberg. Second Edition, ISBN 978-3-642-19459-7. DOI 10.1007/978-3-642-19460-3

Clar, N., Salgado, P. A., Perdicoúlis, T-P A., (2021), Subtractive mountain clustering algorithm applied to a chatbot to assist elderly people in medication intake, Conference MLNLP2021, pp. 33-48, DOI: 10.48550/arXiv.2110.00933.

Goyal, G., (2021), Twitter Sentiment Analysis Using Python | Introduction & Techniques. URL: https://www.analyticsvidhya.com/blog/2021/06/twitter-sentiment-analysis-a-nlp-use-case-for-beginners/, [Cited: 24 May 2023].

Hsieh, J.-N., Ali, M. and Yang, M.-S., (2021) Subtractive Clustering for Categorical Data with a Novel Separation Difference Validity Index. Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery, vol. 88, pp. 1695-1703, DOI: 10.1007/978-3-030-70665-4_184.

Kurukuru, V. S. B., Blaabjerg, F., Khan, M. A., (2020), Ahteshamul Haque, A Novel Fault Classification Approach for Photovoltaic Systems. Energies, vol. 13, issue 2, DOI: 10.3390/en13020308.

Lang, K., (1995), NewsWeeder: learning to filter netnews, International Conference on Machine Learning, pp. 331-339, DOI:10.1016/b978-1-55860-377-6.50048-7.

McAuley, J., Leskovec, J., (2013), From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews, Proceedings of the 22nd international conference on World Wide Web, pp. 897–908, May 2013, DOI:10.1145/2488388.2488466.

Shameem, M.-U.-S., Ferdous, R,, (2009) An efficient k-means algorithm integrated with Jaccard distance measure for document clustering. First Asian Himalayas International Conference on Internet, pp. 1-6, 2009, DOI: 10.1109/AHICI.2009.5340335.

Singh, G., Sundaram, S., (2015), Subtractive Clustering Scheme for Text-Independent Online Writer Identification. 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, pp. 311-315, DOI: 10.1109/ICDAR.2015.7333774.

Tongbram, S., Shimray, B. A, Singh, L. S., (2021), Segmentation of image based on k-means and modified subtractive clustering,. Indonesian Journal of Electrical Engineering and Computer Science, vol. 22, No. 3, 2021, DOI: 10.11591/ijeecs.v22.i3.pp1396-1403.

Tushar J., (2021), Evaluating Clustering Algorithm — Silhouette Score. *medium.com.* [Online] 21 05 2021. [Cited: 31 08 2023.] https://tushar-joshi-89.medium.com/silhouette-score-a9f7d8d78f29.

Yang, M.-S., Wu, K.-L., (2005), A modified mountain clustering algorithm. Pattern Analysis and Applications, vol. 8, no. 1-2, pp. 125–138, June 2005, DOI:10.1007/s10044-005-0250-9.

Zanoon, N., Alkharabsheh, K., Ryalat, M. H., (2020), Optimizing MapReduce Model for Big Data Analytics Using Subtractive Clustering Algorithm. International Journal of Advanced Science and Technology, vol. 29, No. 4, pp. 4106-4119.