Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

Hub based K-Means Subspace Clustering for Improved Efficiency

Anuradha Sanapala[1], B. Jaya Lakshmi[2], K. B. Madhuri[3]

[1]Research Scholar, Duke Training Center, Abu Dhabi, United Arab Emirates

[2]Associate Professor, Department of IT, Gayatri Vidya Parishad College of Engineering (A), Andhra Pradesh, India

[3]Professor, Department of IT, Gayatri Vidya Parishad College of Engineering (A), Andhra Pradesh, India

[2]meet_jaya200@gvpce.ac.in

Article Info

Page Number: 1679 - 1691

Publication Issue:

Vol 72 No. 1 (2023)

Abstract

Clustering is the primary data mining functionality that groups the data points based on their similarities. As the dimensionality of the dataset increases, each data point appears to be equidistant to each other, thus making distance metrics less significant. Clustering in subspaces attempts to resolve the issue of the curse of dimensionality to some extent. However, determining clusters relevant to a subspace is a challenging task. Hubs are the data points which appear to be neighbours for most of the data points. Hence, the clusters are usually surrounded by such hubs, and it is efficient to consider these hubs as seed points while performing partitional clustering. In this paper, Hub based K-means Subspace Clustering (HKSC) is proposed, where K refers to the number of clusters to be identified. The initial seed points are selected using Hubness Scores on each subspace, and clusters are found using the partitional method. The proposed algorithm is evaluated and compared with state-of-the-art subspace clustering algorithms such as SUBCLU, SCHISM, and PCoC in terms of cluster quality metrics, namely purity and silhouette coefficient. It is proved that the proposed algorithm outperforms the existing algorithms. With regard to purity, on average, HKSC has shown an improvement of 71%, 18%, and 15% over SUBCLU, SCHISM and PCoC respectively. With respect to silhouette coefficient, the clustering result was 300% better when compared to SUBCLU result and 54% better than SCHISM. Concerning the execution time, HKSC showed 56% less than that of SUBCLU. The proposed approach uses the concept of hubs in order to efficiently mine the subspace clusters in partitional subspace clustering.

Article History

Article Received: *15 October 2022*

Revised: *24 November 2022*

Accepted: *18 December 2022*

Keywords: cluster quality, Hubness Score, partitional clustering, purity, subspace clustering.

1. Introduction

Nowadays, data are abundant in nature, both in terms of size as well as dimensionality. Most real-world applications accumulate voluminous data that provide ample opportunity for data analysts to extract knowledge in support of decision making. Data mining is the study of extracting interesting patterns from huge volumes of data. Depending on the context of decision making, different types of patterns are to be extracted by applying appropriate data mining techniques. However, the combinations of techniques are often employed for decision support, which is referred to as Data Science in recent times. Clustering, which is one of the important data pre-processing techniques, has become quite challenging due to the enormous size of the data.

Drawbacks of Traditional Clustering Methods:

(i) For high dimensional data, the true clusters are masked in the subspaces.
(ii) Due to inherent sparsity of the data objects in high dimensional data spaces, the distance measures become insignificant with the increase in dimensionality of the dataset. This is referred to as curse of dimensionality. Due to this, the traditional methods [1] are incapable of mining meaningful clusters.
(iii) Each attribute in the dataset contributes differently for different clusters and some of the attributes may be irrelevant for a given cluster.

Need for Subspace Clustering

The real-world data often consists of descriptions of complex data objects each of which is described in terms of a large set of attributes or variables [2]. The Availability of data in abundance calls for efficient algorithms to analyse the data for pattern extraction. This is a challenging task. Data mining functionalities such as cluster analysis become more complex as the number of dimensions increases [3]. The distance between data points may not be properly discriminated in a high-dimensional space. This is referred to as the curse of dimensionality.

The most common approaches to deal with the curse of dimensionality are dimensionality reduction, feature selection and feature creation. The dimensionality reduction methods such as Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) [4] make use of linear algebra techniques and transform the original high dimensional feature space to lower dimensional feature space. These methods may not be well suited for the subspace clustering process since the clusters identifiable in the transformed feature space coincide with the clusters of the original space, still hiding the clusters of significance in subspaces.

While conventional clustering refers to the process of grouping the data objects that exhibit similar behaviour, all the features of data may not be relevant to characterize the members of the given cluster. Clusters in full-dimensional space might not be interesting for all purposes since different features contribute differently to form clusters of objects for varied purposes. So, subspace clustering focuses on identifying a specific subset of attributes that describe a cluster. In other words, subspace clustering

aims to find meaningful clusters in subspaces formed by different combinations of attributes [5].

To deal with voluminous data, data reduction/feature extraction techniques may be applied. However, these techniques involve loss of information leading to poor results. Subspace clustering addresses this problem as it mines the clusters in all possible subspaces. So, subspace clustering techniques are more useful in finding clusters hidden in subspaces. There are two important purposes involved in subspace clusters. Firstly, once the clusters are found, depending on the context, the clusters from the relevant subspace may be considered for decision making. Secondly, the subset of attributes that contribute to forming the clusters would be considered significant for that cluster. Subspace clustering is a complex task since the number of possible subspaces increases exponentially with the dimensionality of a dataset, which in turn results in enormously large numbers of subspace clusters affecting the interpretability of results negatively. For n-dimensional data, the possible number of subspaces is 2n-1. There is a trade-off between number of subspace clusters and subspace cluster quality. The existing subspace clustering algorithms which explore exponential number of subspace clusters take more time with considerably low cluster quality.

Research on subspace clustering approaches aims to provide better solutions to the challenge of minimizing the complexity of exploring the subspace clusters in partitional methods. The authors intend to carry this research work to reduce the execution time of the subspace clustering algorithm and at the same time improving the quality of the subspace clusters. So, to reduce the complexity of the subspace clustering process while applying the partitional methods, the concept of hubs and Hubness Score is used. Hubs are used to interpret the data distribution, and it is found that they are closely located near the centre of clusters; therefore, it is appropriate to consider hubs as the initial seed points while expanding the clusters. Unlike the existing partitional subspace clustering algorithms, this reduces the number of iterations, and the algorithm converges soon resulting in better quality clusters thus minimising the research gap.

2. Related Work

In recent decades, subspace clustering has been a major focus area among researchers. The algorithms in this area can be categorized as bottom-up and top-down approaches [6]. One of the oldest algorithms for finding subspace clusters is SUBCLU [7], which follows a bottom-up approach to finding subspace clusters. First, it applies DBSCAN [7] algorithm to generate all one-dimensional clusters. Then, it checks in the higher dimensions if this clustering structure still exists. The monotonicity property of clustering is used to conclude that there can be no other clusters found in higher dimensions. Once all the lowest dimensional clusters are found, it generates candidates for the next level. For doing so, it takes all possible combinations of attributes. To minimize the number of subspaces generated, it prunes the combination that includes a subspace for which no clusters existed in a lower-dimensional space. The next stage is to evaluate the relevant subspaces as well as the one-dimensional clusters further. In this step, it iteratively applies DBSCAN for each cluster in one-dimensional space. For finding clusters in k-dimensional space, it takes the best subspace of (k-1)

dimension, the best subspace being the one with minimum coverage of data points.

Unlike SUBCLU, SCHISM [6] is a top-down procedure that uses the concept of support and Chernoff-Hoeffding bounds. It mines for the maximal interesting subspaces by using a depth-first search with backtracking. As a first step, it divides the given dataset into a user-specified number of discrete bins. Further, it performs a data transformation by converting the horizontal format to vertical for optimal computation and memory utilization. Once it finds all the interesting maximal subspaces while pruning the search tree by merging similar subspaces, it finally either assigns a data point to the subspaces by estimating probability distribution functions or marks it as an anomaly.

A subspace clustering algorithm that focuses on summarizing the results is PCoC [6]. All the one-dimensional subspace clusters are first computed using the DBSCAN algorithm. Then, the set of all low-dimensional subspace clusters up to two or three dimensions can be generated by taking the intersection of objects and the union of attributes of two subspace clusters. The resultant cluster is said to be valid if it contains at least two elements. This set of clusters is given as input to the PCoC algorithm, which performs a k-medoids style grouping on clusters that results in a set of subspace clusters with their centres.

Subspace clustering is widely used in the application domains like biology [2], computer vision, astronomy, the discovery of conducive living environments for animals, community identification in social networking sites and social media mining. Recent research focused on multi-view subspace clustering [8-10] as real-time data can sometimes be obtained from multiple sources with distinct sets of attributes. It has also been applied to address the problem of image segmentation by using the feature vectors of images [5, 11]. In recent years, the concept of hubness to find clusters in subspaces has also been gaining popularity. While some research focused on the impact of hubness on K-nearest neighbour graphs [12], others studied hub-based clustering for high-dimensional data by focusing on the association between hubs and subspace clusters [13].

Section 3 describes in detail the methodology used in the proposed Hub based K-means Subspace Clustering (HKSC). The experimental results are provided in Section 4. Section 5 concludes the paper with its possible future scope.

3. Methodology

The existing algorithms for subspace clustering extend the idea of density connectedness to each subspace; however, the proposed algorithm introduces a novel approach of Hub based K-means Subspace Clustering to search for the clusters in each feature space. The algorithms are explained in detail with an example by first defining the introductory concepts.

3.1. Preliminary concepts

Let DB be the set of data points. Let A be the set of attributes. A subspace is a feature space formed by a subset of attributes; For example, $S \subseteq A$ be the subset of attributes and hence a subspace. For a set of attributes, A, the possible number of subspaces are $2|A| - 1$.

Epsilon (ε): Epsilon is defined as the radius around an object that defines the neighbourhood of it.

Epsilon-Neighbourhood: It is defined as the area covered within the epsilon radius of an object.

MinPoints: It is a user-specified threshold value of the number of objects that lie in the epsilon-neighbourhood of a data point for it to be dense.

Hubness Score*:* The Hubness Score of a data point $<d_i, a_i>$, where d_i represents the i^{th} data point in DB and a_i represents the i^{th} attribute in A, is defined as the number of times $<d_i, a_i>$ has contributed as a neighbour for other data points.

K-Hubs*:* Once the data points are arranged in decreasing order of their Hubness Scores, the top K points, $<d_1, a_1>$ to $<d_k, a_k>$, are assigned as K-Hubs.

3.2. Hub based K-means Subspace Clustering (HKSC)

The proposed methodology for finding subspace clusters in each dimension is a two-step process in which the algorithm starts with identifying K-Hubs, where K is the number of clusters to be found in each subspace, and iteratively applies the k-means [14] algorithm to find the clusters in each feature space.

The FindKHubs algorithm begins by computing the Hubness Scores of all the data points. The data points that have crossed the threshold would be treated as hubs. Then the K-Hubs would be chosen as the initial cluster centres, unlike the traditional k-means method where the initial cluster centres are chosen randomly. Once the initial centres are decided, then it iteratively assigns all the other data points to one of the cluster centres to which it is nearer. After the first iteration is completed, it updates the cluster centres by computing the average of all the data points in a cluster. It continues the process until there is no change in clustering or it reaches the max iterations. Thus, convergence is reached with a minimal number of iterations. This whole process is applied to all possible subspaces of the dataset.

Algorithm: HKSC
Input: Dataset, K, MinPoints.
Output: subspace clusters from all combinations of subspaces.
Method:

(1) call FindKHubs method and assign the returned K data points as initial seeds of K clusters C_1, C_2, …., C_k
(2) for each subspace
(3) repeat
(4) for each data point $<d_i, a_i>$
(5) find distance from each seed point
(6) assign data point $<d_i, a_i>$ to nearest cluster
(7) end for

(8) calculate new cluster centres by updating cluster means

(9) until two successive iterations yield the same clustering patterns

(10) end for

Algorithm: FindKHubs
Input: Dataset, K, epsilon.
Output: K-Hubs.
Method:
(1) Hubness Score←0
(2) for each data point $<d_i, a_i>$
(3) for each data point $<d_j, a_j>$ where $i \neq j$
(4) if $<d_i, a_i>$ lies within epsilon neighbourhood of $<d_j, a_j>$
(5) increment Hubness Score of $<d_i, a_i>$
(6) end for
(7) update the final score of $<d_i, a_i>$
(8) reset Hubness Score
(9) end for
(10) Sort the data points in decreasing order of their Hubness Scores
(11) return top K data points as K-Hubs

3.2.1. Example

Consider a dataset D with 12 objects, namely o1, o2, …., and o12, which are described over three attributes scilicet A1, A2, and A3. The input parameters, namely the dataset, epsilon, and K, are fed to FindKHubs. For this example, epsilon is fixed to 0.3, and K is taken as 3.

The FindKHubs method first calculates the Hubness Scores of all the objects by checking for each object how many times it is acting as a neighbour for other data points. Table 1 shows the Hubness Scores of all the 12 objects. Then it arranges these points in the decreasing order of their Hubness Scores, and selects the top K data points, which is 3 in this case and returns as the output as shown in Table 2. These top-3 objects (o6, o7, and o1) are then considered the initial seeds for the next step in the process.

Once the initial seeds are returned by the FindKHubs algorithm, these points are given as input to the k-means algorithm along with K, the number of clusters. It starts generating clusters iteratively from lower-dimensional subspaces. The clusters generated after three iterations in the A1 subspace are shown in Tables 3.1 through 3.3. As the clustering structure in iterations 2 and 3 are the same, the algorithm converges here and gives the output clusters for the A1 subspace. In a similar manner, the algorithm is applied repeatedly for all the other subspaces.

Table 1. Dataset D

Object	A1	A2	A3
o1	0.441	0.502	0.571
o2	0.552	0.587	0.725
o3	0.617	0.649	0.736
o4	0.639	0.692	0.639
o5	0.141	0.169	0.529
o6	0.405	0.446	0.662
o7	0.349	0.347	0.879
o8	0.441	0.504	0.558
o9	0.201	0.240	0.549
o10	0.229	0.279	0.508
o11	0.613	0.614	0.906
o12	0.788	0.843	0.607

Table 2. Hubness Scores of the objects of D

Object	Hubness Score	Seed (Yes/No)
o1	9	Yes
o2	8	No
o3	8	No
o4	8	No
o5	4	No
o6	10	Yes
o7	10	Yes
o8	9	No
o9	6	No
o10	6	No
o11	8	No
o12	4	No

Table 3.1. Iteration 1

Cluster	Objects
C1	{o6}
C2	{o5, o7, o9, o10}
C3	{o11, o1, o12, o2, o3, o4, o8}

Table 3.2. Iteration 2

Cluster	Objects
C1	{o1, o6, o7, o8}
C2	{o5, o9, o10}
C3	{o11, o12, o2, o3, o4}

Table 3.3. Iteration 3

Cluster	Objects
C1	{o1, o6, o7, o8}
C2	{o5, o9, o10}
C3	{o11, o12, o2, o3, o4}

4. Results and Discussion

The proposed algorithm was developed in Java (JDK 1.8) on a PC with 8 GB RAM and a 2.11 GHz processor. Both existing and proposed algorithms are tested across various benchmark datasets taken from UCI Machine Learning Repository [15]. Table 4 gives a brief description of the datasets used for testing.

4.1.*Cluster quality measures*

Purity: It is an external evaluation criterion of cluster quality. It is the per cent of the total number of objects (data points) classified correctly. The range of purity is [0, 1].

The purity values of SUBCLU and HKSC are presented in Table 5 and depicted as a column chart in Figure 1. From the results, it is clear that the performance of HKSC is far more efficient when compared to the existing algorithms. HKSC outperformed individually with each of these methods. The purity has been improved by 71%, 18%, and 15% concerning SUBCLU, SCHISM, and PCoC respectively.

Silhouette Coefficient*:* It is the ratio of the difference between the inter-cluster distance and intra-cluster distance of an object to the maximum of those two distances.

In addition to the improvement of purity, HKSC has also shown a dramatic enhancement in terms of silhouette coefficient. It can be deduced from Table 6 and Figure 2 that HKSC outperformed SUBCLU with an improvement of 300%. The proposed algorithm showed a 54% increase in cluster quality with respect to SCHISM.

Table 7 summarises the running times of the algorithms. The same is shown pictorially in Figure 3. It can be concluded that the execution time of HKSC when compared to the existing SUBCLU method has been reduced by 56% on average.

Hub based K-means Subspace Clustering (HKSC), the proposed algorithm, is tested on different benchmark datasets from the UCI machine learning repository by comparing its performance with existing subspace clustering algorithms: SUBCLU, SCHISM, and PCoC. It is observed that HKSC has produced better quality clusters consistently in all the datasets. The proposed algorithm is proved to be a better and optimal method for forming subspace clusters with maximum purity, minimum execution time, and a good silhouette coefficient.

Table 4. Dataset description

Dataset	Tuples	Attributes	Classes
Seeds	210	7	3
User Knowledge Modeling	258	5	4
Wholesale Customer	440	7	2

Data			
Pima Indian Diabetes	768	8	2
Bank Note	1372	4	2
Yeast	1484	8	9
Steel Plates	1772	9	7
Image Segmentation	2100	9	7
Wine Quality	3000	9	7

Table 5. Purity comparison among various algorithms for different benchmark datasets

Dataset	SUBCLU	SCHISM	PCoC	HKSC
Seeds	0.342	0.736	0.739	0.878
User Knowledge Modeling	0.642	0.649	0.636	0.659
Wholesale Customers	0.681	0.712	0.735	0.835
Pima Indian Diabetes	0.659	0.674	0.684	0.706
Bank Note	0.555	0.726	0.555	0.797
Yeast	0.317	0.548	0.558	0.608
Steel Plates	0.365	0.372	0.539	0.553
Image Segmentation	0.198	0.544	0.428	0.584
Wine Quality	0.417	0.502	0.656	0.694

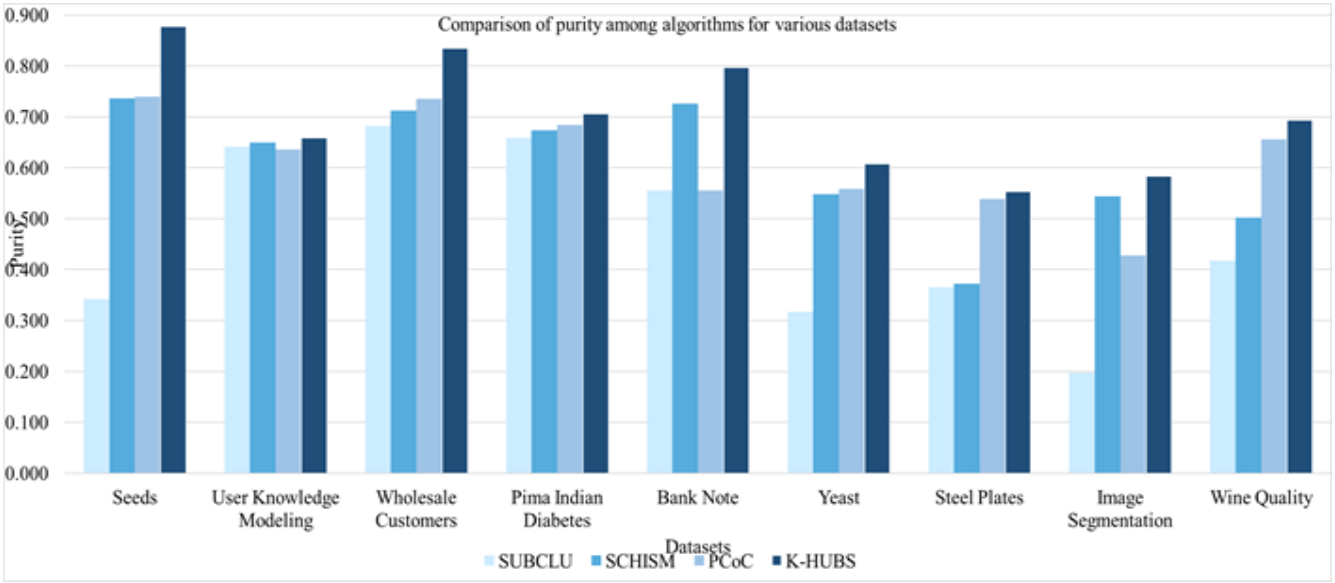


Figure 1. Comparison of purity various algorithms

Table 6. Silhouette Coefficient comparison among various algorithms

Dataset	SUBCLU	SCHISM	HKSC
Seeds	-0.567	0.591	0.608
User Knowledge Modeling	-0.516	0.174	0.283
Wholesale Customer Data	0.091	0.301	0.346
Pima Indian Diabetes	0.146	0.216	0.305
Bank Note	-1.000	0.422	0.430
Yeast	0.482	0.531	0.590
Steel Plates	-0.094	0.237	0.469
Image Segmentation	0.167	0.142	0.435
Wine Quality	-0.030	0.192	0.280

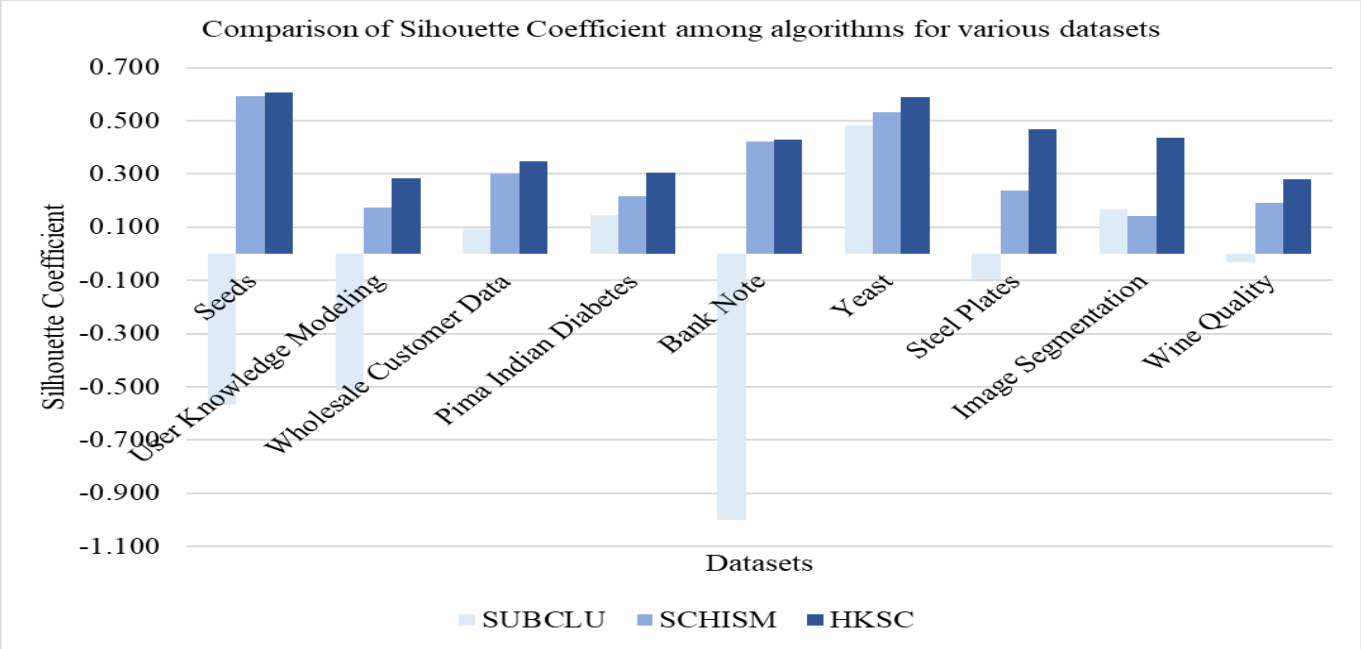


Figure 2. Comparison of silhouette coefficient of various algorithms

Table 7. Comparison of execution time (in minutes) between SUBCLU & HKSC

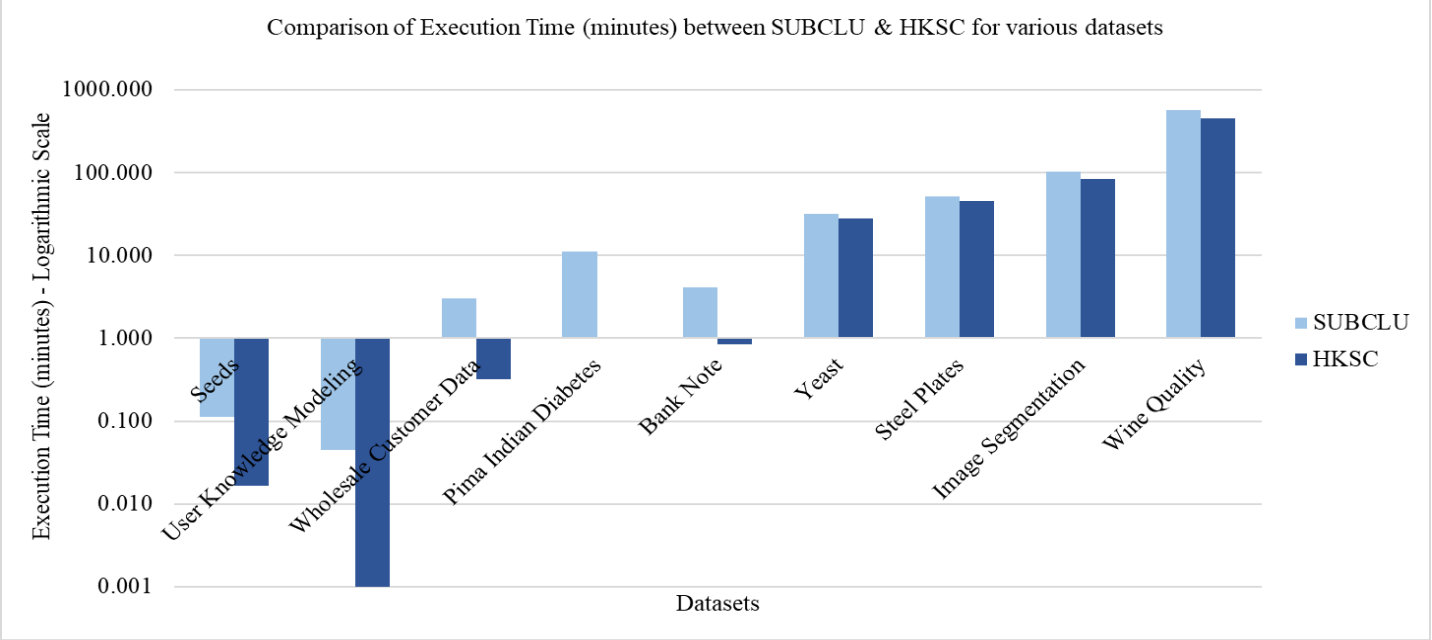
Dataset	SUBCLU	HKSC
Seeds	0.111	0.017
User Knowledge Modeling	0.045	0.001

Wholesale Customer Data	3.007	0.317
Pima Indian Diabetes	11.230	1.000
Bank Note	4.155	0.850
Yeast	32.000	28.000
Steel Plates	52.000	45.500
Image Segmentation	104.000	84.483
Wine Quality	569.000	455.533

Figure 3. Comparison of execution time between SUBCLU & HKSC

5. Conclusion

Clustering in high-dimensional data poses the problem of generating meaningless clusters because



of the exponential increase in the distance measures as we move towards the higher dimensions. Existing subspace clustering algorithms address this problem up to some extent by searching for meaningful clusters in different subspaces. This paper proposed a novel approach to subspace clustering, which uses the concept of hubness to select the initial seeds for the k-means clustering algorithm that iteratively finds clusters in each possible feature space. Since the traditional k-means method chooses initial cluster centres randomly, the output differs each time a different set is considered, sometimes compromising the cluster quality. Interestingly, the proposed method evaluates the data points before choosing the initial seeds, which helps in better convergence with improved

cluster quality. Hub based K-means Subspace Clustering (HKSC), the proposed algorithm was compared with state-of-the-art algorithms like SUBCLU, SCHISM and PCoC, and with regard to purity, it has shown improvement of 71%, 18% and 15% over SUBCLU, SCHISM and PCoC algorithms respectively. With respect to silhouette coefficient, the clustering result was 300% better when compared to SUBCLU result and 54% better than that of SCHISM. The runtime of HKSC was 56% less when compared to that of SUBCLU.

The proposed algorithm is applied to numeric datasets. This research work could be extended to apply to other complex data types and to deal with datasets containing missing values. In certain domains, data objects contribute partially to more than one cluster of a subspace, and the research work could be expanded to extract soft clusters based on the participation count of the cluster members. A data object could be a member of different clusters in a given subspace based on its characteristics. Fuzzy membership of the data object could be considered as future work while forming subspace clusters.

References

[1] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka, C. I. Eke, A. A. Akinyelu, "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," Engineering Applications of Artificial Intelligence, vol. 110, 2022.

[2] S. Ehsani, C. K. Reddy, B. Foreman, J. Ratcliff, V. Subbian, "Subspace Clustering of Physiological Data From Acute Traumatic Brain Injury Patients: Retrospective Analysis Based on the PROTECT III Trial," JMIR Biomed Engineering, vol. 6, no. 1, 2022.

[3] W. Li, J. Hannig, S. Mukherjee, "Subspace Clustering through Sub-Clusters," The Journal of Machine Learning Research, vol. 22, no. 1, pp. 2413–2449, 2021.

[4] H. Peng, N.G. Pavlidis, "Weighted sparse simplex representation: a unified framework for subspace clustering, constrained clustering, and active learning," Data Mining and Knowledge Discovery, vol. 36, no. 3, pp. 958–986, 2022.

[5] B. Sun, P. Zhou, L. Du, X. Li, "Active deep image clustering," Knowledge-Based Systems-Elsevier, vol. 252, 2022.

[6] B. J. Lakshmi, M. Shashi, K. B. Madhuri, "A rough set based subspace clustering technique for high dimensional data," Journal of King Saud University - Computer and Information Sciences, vol. 32, no. 3, pp. 329-334, 2020.

[7] J. R. Jørgensen, K. Scheel, I. Assent, "GPU-INSCY: A GPU-Parallel Algorithm and Tree Structure for Efficient Density-based Subspace Clustering," in Proc. 24th International Conference on Extending Database Technology (EDBT), pp. 25-36, 2021.

[8] R. -k. Lu, J. -w. Liu, W, X. Zuo, W. -m. Li, "Multi-view subspace clustering with consistent and view-specific latent factors and coefficient matrices," in Proc. International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2021.

[9] Q. Zheng, J. Zhu, "Multi-view Subspace Clustering with View Correlations via low-rank tensor learning," Computers and Electrical Engineering, vol. 100, 2022.

[10] Y. Duan, H. Yuan, C. S. Lai, L. L. Lai, "Fusing Local and Global Information for One-Step Multi-View Subspace Clustering," Applied Science, vol. 12, no. 10, 2022.

[11] J. Francis, A. Johnson, B. Madathil, S. N. George, "A Joint Sparse and Correlation Induced Subspace Clustering Method for Segmentation of Natural Images", in Proc. IEEE 17th India Council International Conference (INDICON), pp. 1-7, 2020.

[12] B. Bratic, M. E. Houle, V. Kurbalija, V. Oria, M. Radovanovic, "The Influence of Hubness on NN-Descent," International Journal on Artificial Intelligence Tools, vol. 28, no. 6, 2019.

[13] P. Mani, C. Domeniconi, "Hub-based subspace clustering," Neurocomputing, vol. 413, pp. 193-209, 2020.

[14] I. Ali, A. U. Rehman, D. M. Khan, Z. Khan, M. Shafiq, J. -g. Choi, "Model Selection Using K-Means Clustering Algorithm for the Symmetrical Segmentation of Remote Sensing Datasets," Symmetry, vol. 14, no. 6, 2022.

[15] The UCI Machine Learning Repository website, 2021. [Online]. Available: http://archive.ics.uci.edu/ml