Discovering Non-Redundant K-means Clusterings in Optimal Subspaces

Rahul Alapati
Department of Computer Science and Software Engineering
Auburn University
rza0037@auburn.edu

Sathish Akula
Department of Computer Science and Software Engineering
Auburn University
sza0096@auburn.edu

Aditya Mahajan
Department of Computer Science and Software Engineering
Auburn University
azm0136@auburn.edu

***Abstract*—Clustering on a huge object collection in high-dimensional space can often be done in more than one way, for instance, objects could be clustered by their shape or alternatively by their color. Non-redundant clustering addresses this class of problems. In this paper, we discuss the implementation of one such non-redundant clustering algorithm, NR-KMEANS. The idea is to find, non-redundant k-means like clusterings in different, orthogonally oriented subspaces of the high-dimensional space. The performance of the algorithm is reported in the form of two evaluation metrics namely, Pair Counting F-1 Measure and Average Variation of Information. The running time of the algorithm has also been reported.**

Keywords—clustering, k-means, high-dimensional data, subspace, non-redundant

I. INTRODUCTION

Data clustering is a popular approach of grouping a set of objects, in such a way that objects in the same group are more similar to each other than to those in other groups. K-Means is a popular clustering algorithm, given a set of n points, finds K centers such that the total squared error between each of the n points and its closest center is minimized. Each cluster center is defined by the empirical mean of its points.

However, in the case of high dimensional data, these classic clustering algorithms like K-means [1], suffer from the curse of dimensionality and also fail in identifying multiple non-redundant clusters in different subspaces. Data in high dimensional space is often represented in the form of sparse high-dimensional features, which doesn't contain any clustering structure.

The curse of dimensionality challenge can be handled by integrating the clustering and dimensionality reduction algorithms together. This integration helps in identifying the subspace for each cluster, in which it is best represented. The orthogonality between the subspaces ensures that the discovered clusterings are non-redundant and represent different views of the data. Thus, the basic idea of NR-KMEANS algorithm, is to find multiple interesting K-means clusterings in different mutually orthogonal subspaces.

The key contributions of NR-KMEANS algorithm are:

- **Multiple interesting k-means Clusterings in Optimal Subspaces**: NR-KMEANS discovers multiple, non-redundant k-means clusterings in orthogonal subspaces.
- **Efficiency**: NR-KMEANS is easy to implement and compatible with many proposed extensions of k-means. It is fast, even without sophisticated performance optimizations.
- **Lightweight Parameterization**: The only input parameter of NR-KMEANS is the number of clusters for each subspace.
- **Noise Handling:** Unlike the existing non-redundant clustering approaches, NR-KMEANS includes the idea of noise subspace, which helps it outperform all the existing methods.

The organization of the paper is as follows: In section II, the main ideas and steps involved in the algorithm are elaborated. In section III, the experiments and results are discussed in detail. Section IV ends the paper with discussion on the potential strong and weak points of the algorithm, with some possible directions and new ideas to solve the weak points.

II. METHODOLOGY

NR-KMEANS algorithm is a simple extension of the well-known Lloyd's algorithm, with its alternating assignment and update steps.

A. Cost Function

We extend the basic idea of classic k-means algorithm by making some assumptions and determine the cost function as follows:

$$F = \sum_{j=1}^{S} \sum_{i=1}^{k_j} \sum_{x \in C_{j,i}} ||P_j^T V^T x - P_j^T V^T \mu_{j,i}||^2$$

where S is the Number of Subspaces,
k_j is the Number of Clusters in the j^{th} subspace,
$C_{j,i}$ is the Objects of cluster i in subspace j,
P_j is the Projection onto the j^{th} subspace,
V is the orthogonal matrix of a rigid transformation
x is the object of the dataset and
$\mu_{j,i}$ is the original space mean of cluster i in subspace j.

Here, each clustering j contains kj clusters and resides in an arbitrarily oriented subspace that is orthogonal to the subspaces assigned the other S−1 clusterings. The orthogonal transformation matrix V rotates the data space such that the subspaces are all axis-parallel in the transformed space. The masking matrices Pj are used to project the data onto the respective axis-parallel subspace. Since the subspaces do not overlap, each dimension of the rotated data space is exclusively mapped onto a single subspace. A data point x can then be projected onto the j^{th} subspace by $P_j^T V^T x$.

Our objective is to optimize this cost function with a modified version of Lloyd's algorithm.

B. Algorithm

The input to the algorithm is a dataset D and the number of clusters per subspace $k_1,\ldots,k_S$.

The output of the algorithm would be clusters per each subspace $C_{1,1},\ldots,C_{S,k_S}$, rotation matrix V, Projections $P_1,\ldots,P_S$ and Dimensionalities $m_1,\ldots,m_S$.

The following are the steps involved in the NR-KMEANS algorithm:

Step 1: **Initialize** V with a random orthogonal matrix.

Step 2: For each subspace j in [1,S]:

- Initialize the dimensionalities per subspace, m_j.
- Map the m_j features onto the masking matrix P_j.
- Initialize the cluster centers $\mu_{j,i}$ within the subspaces using k-means++ or with the random data points of dataset D.

Repeat the following steps until convergence.

Step 3: **Assignment Step**

For each subspace j in [1,S]:

Initialize each cluster i, $C_{j,i}$ in the j^{th} subspace

For each data point x in dataset D:

$$\forall \mathbf{x} \in \mathcal{D}:$$
$$i \leftarrow \arg\min_{i \in [1,k_j]} \left\| P_j^\top V^\top \mathbf{x} - P_j^\top V^\top \boldsymbol{\mu}_{j,i} \right\|^2$$
$$C_{j,i} \leftarrow C_{j,i} \cup \{\mathbf{x}\}$$

Compute the cost function and add the data point x to the respective cluster.

Step 4: **Update Cluster Centers**

For each subspace j in [1,S]:

For cluster i in $[1,k_j]$:

Update the cluster centers $\mu_{j,i}$ using the k-means++ algorithm.

Step 5: **Update the random orthogonal matrix**

Organize the subspaces in a pairwise manner such that they are orthogonal to each other.

For each pair(s,t) of subspaces:

Create a combined projection of the data on the s and t subspaces, $P_{s,t}$.

Perform eigen decomposition as described below:

$$V_{s,t}^{\langle c \rangle}, \mathcal{E} \leftarrow \text{eig}\left(P_{s,t}^\top V^\top (\Sigma_s - \Sigma_t) V P_{s,t}\right)$$

Update the dimensionalities:

$$m_s \leftarrow |\{e | e \in \mathcal{E} \wedge e < 0\}|$$
$$m_t \leftarrow |\mathcal{E}| - m_s$$

Update the random orthogonal matrix:

$$V \leftarrow V \times \text{toFull}(V_{s,t}^{\langle c \rangle})$$

Finally update the masking matrices P_s, P_t.

C. Implementation Details

In the initialization step, the dimensions of the features in dataset D are equally divided among the dimensionalities per subspace. For example, if the dimensions of the dataset D are 900 * 400, we equally divide them among the two subspaces as follows: 900 * 200 → Subspace 1 and 900 * 200 → Subspace 2. Also, the initial cluster centers can either be initialized using the k-means++ algorithm or using the random data points in the dataset D.

The assignment and update steps are repeated consecutively, until the cluster centers converge. In the assignment step, we project the data point x on to the subspace j using the masking matrix P_j and the random orthogonal matrix V. Then, like in k-means we calculate the distance of the projected data point x from every cluster center $\mu_{j,i}$ in the subspace j and assign it to the cluster i whose center $\mu_{j,i}$ is at minimum distance from x.

If we clearly observe, apart from the projection on to the respective subspace, the cost function is similar to that of the k-means algorithm. Hence, NR-KMEANS is an extension of k-means algorithm with a dimensionality reduction technique to its rescue.

In the update step, we update the cluster centers $\mu_{j,i}$ using the k-means++ algorithm [3]. Then we also update the random orthogonal matrix, by pairing the subspaces orthogonally and creating a combined projection of the data onto them. Based on the combined projection, we perform an eigen decomposition and then update the dimensionalities of the subspaces, random orthogonal matrix and the masking matrices in order to minimize the cost function.

The NR-KMEANS algorithm has been implemented in Python. The scikit learn python library [4] was used for importing the k-means++ algorithm and also for the calculating the evaluation metrics. The SciPy python library [5] was used for performing orthogonal transformations. All the operations on the data like matrix multiplication, matrix transpose, etc. were carried out using the NumPy python library [6].

D. Challenges in Implementation

The following are the challenges faced by us in implementation of the NR-KMEANS:

- The authors don't provide a detailed explanation of the NR-KMEANS algorithm. Especially, the update step in which the random orthogonal matrix is updated, leaves the readers in confusion.
- The procedure to create a combined projection onto a pair of orthogonal subspaces is not explained.
- The details about eigen value decomposition, updating the dimensionalities, orthogonal matrix and projections are not clearly laid out, which makes the implementation a lot difficult.
- Also, the interpretation of the features in the datasets is not at all mentioned, which leaves the developers in dark about the dimensionalities of different matrices used in the algorithm.

After taking into consideration, all the challenges mentioned above, and numerous attempts to implement the step 5 where the random orthogonal matrix is updated, unfortunately we had to limit ourselves up to step 4, where we update the cluster centers until convergence.

III. Experiments and Results

A. Datasets

Six synthetic and real-world datasets were considered for performing our experiments as described in the paper. The following four datasets are special non-redundant datasets and contain more than one set of class labels:

- ALOI-2Sub Dataset [7]

  Amsterdam Library of Object Images (ALOI) dataset. It consists of 288 data points and 611 dimensions.

- Stickfigures Dataset [8]

  Consists of 900 data points and 400 dimensions.

- Fruits Dataset [9]

  Consists of 105 data points and 6 dimensions.

- Syn3Sub Dataset [2]

  Consists of 2000 data points and 15 dimensions.

The following two datasets are taken from the UCI repository [10] and contain only a single set of labels each:

- Spam Dataset

  Consists of 4601 data points and 56 dimensions.

- Shuttle Dataset

  Consists of 43500 data points and 9 dimensions.

We performed an effectiveness test and an efficiency test of our algorithm on the ALOI dataset and tested its scalability on the Stickfigures dataset. The ALOI dataset has 2 subspaces with 2 clusters per each subspace. Whereas, the Stickfigures dataset consists of 2 subspaces with 3 clusters per each subspace.

We were not able to perform the effectiveness and efficiency tests on the Fruits, Syn3Sub, Spam and Shuttle dataset, because the number of subspaces and the number of clusters per subspace were not at all described, either in the manuscript or in the source of the dataset. For example, in the case of Syn3Sub, it was mentioned in the manuscript that the dataset consisted of three clustered subspaces with 3, 4 and 5 Gaussian clusters, respectively. However, the real Syn3Sub dataset, was not in line with this description. All these challenges, unfortunately, limited us to run our experiments on the ALOI and Stickfigures datasets.

B. Evaluation Measures

The following two metrics were used to measure the effectiveness/quantitative ability of NR-KMEANS algorithm:

- Pair Counting F-1 Measure
  It is used in order to account for multiple sets of class labels on the ground-truth side, as well as for multiple sets of clusterings as a result of algorithms. Like the traditional F1-score the best achievable score is 1.

- Average Variation of Information
  It is used to measure the measure the redundancy among the subspaces. It measures the distance between two different clusterings, where higher values are better.

C. Quantitative Experiment

In this experiment, our implementations of NR-KMEANS algorithm is compared with several state of the art algorithms namely, STATPC [11], INSCY [12], RESCU [13], ISAAC [14], mSC [15] and Ortho 1&2 [16] as described in the paper.

STATPC, INSCY and RESCU assign each cluster its individual axis-parallel subspace. Whereas, ISAAC, mSC and Ortho 1&2 aim to find multiple clusterings in multiple, arbitrarily oriented subspaces. These algorithms were selected because of their common goal to reduce the redundancy between the clusters.

We have performed this experiment on both full dimensional datasets as well as the PCA applied datasets. The principal component analysis (PCA) has been used to reduce the dimensionality of the ALOI dataset from 611 to 8 and Stickfigures dataset from 400 to 5, while keeping 90% of the total variance.

The following tables 1 and 2, show the Pair Counting F1 Measure of the ALOI and Stickfigures datasets:

Table 1: Pair-Counting F1 (pc-F1) measures of ALOI and Stickfigures datasets with full dimensions

Algorithm	**ALOI**	**Stickfigures**
Our NR-KMEANS	0.93	0.97
Original NR-KMEANS	0.77	1.00
ORTH1	0.77	0.71
ORTH2	0.67	0.79

Table 2: Pair-Counting F1 measures of PCA applied ALOI and Stickfigures datasets

Algorithm	**ALOI (PCA = 8)**	**Stickfigures (PCA = 5)**
Our NR-KMEANS	0.64	0.68
Original NR-KMEANS	1.00	1.00
ISAAC	0.46	0.86
mSC	0.81	0.71
ORTH1	0.82	0.89
ORTH2	0.82	0.71
STATPC	0.52	0.60
INSCY	NA	0.62
RESCU	0.33	0.58

All the data shown in tables 1 and 2, except for our NR-KMEANS has been extracted from the paper for comparison. In table 1, only 3 algorithms namely original NR-KMEANS, ORTH1 and ORTH2 have been reported, because they are the only 3 which performed the experiment on full dimensional data. Also, the INSCY in the case of ALOI dataset, failed to produce any result either due to memory constraints or runtime demands.

Higher pc-F1 values indicate higher effectiveness of the algorithm. From the tables we can observe that the original NR-KMEANS algorithm achieves the best achievable pc-F1 score of 1 and it is the most effective algorithm for clustering high dimensional data. The perfect score of 1, shows the importance of the orthogonality of subspaces in finding non-redundant clusters in high dimensional space.

Our NR-KMEANS algorithm is comparable with the other state of the art algorithms in case of the full dimensional datasets, but is only better than RESCU, STATPC and ISAAC when PCA is applied.

The following tables 3 and 4, show the Average Variation of Information measure of ALOI and Stickfigures datasets:

Table 3: Average Variation of Information measures of ALOI and Stickfigures datasets with full dimensions

Algorithm	ALOI	Stickfigures
Our NR-KMEANS	0.33	0.52
Original NR-KMEANS	0.95	2.20
ORTH1	0.95	1.73
ORTH2	0.82	1.31

Table 4: Average Variation of Information measures of PCA applied ALOI and Stickfigures datasets

Algorithm	ALOI (PCA = 8)	Stickfigures (PCA = 5)
Our NR-KMEANS	0.63	0.85
Original NR-KMEANS	1.39	2.20
ISAAC	NA	2.06
mSC	1.38	1.26
ORTH1	1.37	1.96
ORTH2	0.60	1.52
STATPC	NA	NA
INSCY	NA	NA
RESCU	NA	NA

All the data shown in tables 3 and 4, except for our NR-KMEANS has been extracted from the paper for comparison. In table 3, only 3 algorithms namely original NR-KMEANS, ORTH1 and ORTH2 have been reported, because they are the only 3 which performed the experiment on full dimensional data. All the measures marked NA were not reported due to the failure of the corresponding algorithms due to memory or runtime constraints.

Even though, the performance of our NR-KMEANS in terms of pc-F1 on full dimensional datasets is comparable to other algorithms, it is outperformed by an extent in case of average variation of information. Higher values of variance show higher non-redundancy the clusters.

Lower variance values in case of our algorithm indicate redundancy of data objects in our clusters. This can be attributed to the fact that our NR-KMEANS doesn't optimize the random orthogonal transformation matrix V by combining the subspaces as described in the paper.

D. Runtime Experiment

In this experiment, we track and report the runtime (in seconds) of our NR-KMEANS algorithm for ALOI and Stickfigures datasets with full and reduced dimensions.

The following tables 5 and 6, report the runtime of our NR-KMEANS in seconds:

Table 5: Runtime (in seconds) of our NR-KMEANS on ALOI and Stickfigures datasets with full dimensions

Algorithm	ALOI	Stickfigures
Our NR-KMEANS	3480	1800

Table 6: Runtime (in seconds) of our NR-KMEANS on ALOI and Stickfigures datasets with reduced dimensions

Algorithm	ALOI (PCA = 8)	Stickfigures (PCA = 5)
Our NR-KMEANS	30	30

The original NR-KMEANS has better time complexity when compared to our implementation, because the original was implemented in SCALA, which is by default a Scalable Language.

The runtime of the original NR-KMEANS was measured by increasing the number of data objects, while keeping the number of features fixed and also by increasing the number of features, while keeping the number of data objects fixed. We couldn't perform this experiment on our NR-KMEANS due to limited understanding about the structure of the ALOI and Stickfigures datasets.

E. Challenges in repeating the Qualitative Experiment

We couldn't repeat the qualitative experiment as described in the paper, on the Stickfigures dataset, because the paper doesn't mention any details about the interpretation of the data into figures.

The Stickfigures dataset consists of features in the form of numbers and in the qualitative experiment they are considered as 9 different poses of Stickfigures. Due to the limitation of knowledge about interpretation of the numbers into poses, unfortunately we couldn't complete the qualitative experiment.

IV. Future Work

The following are **Strong Points** of the paper:

- NR-KMEANS discovers multiple, non-redundant k-means like clusterings in orthogonal subspaces. The orthogonality between the subspaces ensures that the discovered clusterings are non-redundant and represent different views of the data.
- It also handles the curse of dimensionality, by projecting sparse high dimensional features onto a low dimensional subspace and clusters it into one of the clusters in that particular subspace.
- It is efficient and it is implemented in SCALA, hence scalable to a larger extent.
- It requires only a single parameter as input, unlike many state of the art algorithms, which require the number of the subspaces, dimensionality of subspace, etc. as input.
- Its use of noise subspace helps it outperform all the existing non-redundant clustering methods.

The following are the **Weak Points** of the paper:

- The main objective of the assignment step of the NR-KMEANS algorithm is to minimize the cost function. However, with each step as the cost function decreases, the NR-KMEANS tends to converge towards a local minima.
- The use of an old and exhausted clustering algorithm like k-means to solve a complex non-redundant clustering problem may not be scalable to a large and complex dataset.
- The selection of number of subspaces, clusters per subspace, cluster centers and optimization steps are complex mathematical operations and in case of a large dataset, can affect the runtime of the algorithm.

Future efforts may be directed towards the following:

- Rectifying the NR-KMEANS cost function to converge at a global minima, instead of a local minima.
- Incorporating a fully automated selection procedure for the number of subspaces and clusters within them.
- Exploring the possibility of the use of other clustering algorithms like hierarchical clustering, etc. or even a combination of effective clustering algorithms to cluster high dimensional data.

References

[1] Lloyd, S., 1982. Least squares quantization in pcm. Information Theory,IEEE Transactions on 28, 129 - 137.

[2] Mautz, Dominik, et al. "Discovering Non-Redundant K-means Clusterings in Optimal Subspaces." Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2018.

[3] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 1027–1035.

[4] https://scikit-learn.org/stable/

[5] https://www.scipy.org/

[6] http://www.numpy.org/

[7] http://aloi.science.uva.nl/

[8] Stephan Günnemann, Ines Färber, Matthias Rüdiger, and Thomas Seidl. 2014. SMVC: semi-supervised multi-view clustering in subspace projections. In The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014. 253–262.

[9] Juhua Hu, Qi Qian, Jian Pei, Rong Jin, and Shenghuo Zhu. 2015. Finding Multiple Stable Clusterings. In 2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015. 171–180.

[10] https://archive.ics.uci.edu/ml/datasets.html

[11] Gabriela Moise and Jörg Sander. 2008. Finding non-redundant, statistically significant regions in high dimensional data: a novel approach to projected and subspace clustering. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008. 533–541.

[12] Ira Assent, Ralph Krieger, Emmanuel Müller, and Thomas Seidl. 2008. INSCY: Indexing Subspace Clusters with In-Process-Removal of Redundancy. In Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), Research Track Paper KDD 2018, August 19–23, 2018, London, United Kingdom 1981 December 15-19, 2008, Pisa, Italy. 719–724.

[13] Emmanuel Müller, Ira Assent, Stephan Günnemann, Ralph Krieger, and Thomas Seidl. 2009. Relevant Subspace Clustering: Mining the Most Interesting Nonredundant Concepts in High Dimensional Data. In ICDM 2009, The Ninth IEEE International Conference on Data Mining, Miami, Florida, USA, 6-9 December 2009. 377–386.

[14] Wei Ye, Samuel Maurus, Nina Hubig, and Claudia Plant. 2016. Generalized Independent Subspace Clustering. In Data Mining (ICDM), 2016 IEEE 16th International Conference on. IEEE, 569–578.

[15] Donglin Niu, Jennifer G. Dy, and Michael I. Jordan. 2010. Multiple Non-Redundant Spectral Clustering Views. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel. 831–838.

[16] Ying Cui, Xiaoli Z. Fern, and Jennifer G. Dy. 2007. Non-redundant Multi-view Clustering via Orthogonalization. In Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA. 133–142.