Effect of Recursive Cluster Elimination with Different Clustering Algorithms Applied to Gene Expression Data

Cihan Kuzudisli
Department of Computer Engineering
Hasan Kalyoncu University
Gaziantep, Turkey
cihan.kuzudisli@hku.edu.tr

Burcu Bakir-Gungor
Department of Computer Engineering
Abdullah Gul University
Kayseri, Turkey
burcu.gungor@agu.edu.tr

Bahjat F. Qaqish
Department of Biostatistics
University of North Carolina at Chapel Hill
North Carolina, USA
bahjat_qaqish@unc.edu

Malik. Yousef
Department of Information Systems
Zefat Academic College
Zefat, Israel
malik.yousef@gmail.com

2023 Innovations in Intelligent Systems and Applications Conference (ASYU) | 979-8-3503-0659-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ASYU58738.2023.10296734

***Abstract*— Feature selection (FS) is an effective tool in dealing with high dimensionality and reducing computational cost. Support Vector Machines – Recursive Cluster Elimination (SVM-RCE) is one of several algorithms that have been developed for FS in high dimensional data. SVM-RCE involves a clustering step which originally is k-means. Using various performance metrics, three alternative algorithms are evaluated in this context; k-medoids, Hierarchical Clustering (HC), and Gaussian Mixture Model (GMM). Comparisons will be carried out on five publicly available gene expression datasets. The results show that k-means in SVM-RCE obtains higher performance than other tested algorithms in terms of classification performance. Additionally, HC shows a similar performance to k-means. Our findings show superiority of using k-means. This study can contribute to the development of SVM-RCE with different variations, leading to decrease in the number of selected genes, and an increase in prediction performance.**

Keywords—Recursive Cluster Elimination, Feature Selection, Clustering, Gene Expression Data Analysis

I. INTRODUCTION

Dimensionality in datasets has increased substantially through advances in technology. This high dimensionality makes it hard for data analysts to draw meaningful knowledge. Feature selection (FS) has arisen as an effective dimensionality reduction technique since FS is capable of removing irrelevant, redundant and noisy features, which can in turn decrease computational time, facilitate model generalization and provide interpretability of data [1]. FS aims at reaching a subset of original feature space without transformation of features. Various FS approaches are available in the literature and generally applied on datasets consisting of thousands of features, and small numbers of samples [2]. This is also the case for gene expression data where classification of sample classes by significant genes (features) poses a challenging problem [3]. Identification of significant genes are important as the data usually includes redundancy, irrelevancy or noise and obtaining a reduced set of genes can lead to early disease detection, stable disease diagnosis or prognosis and effective clinical treatment [4].

Traditionally, FS approaches can be categorized as filter, wrapper and embedded techniques. Filter type FS has no interaction with the learning algorithm and utilizes intrinsic properties of the data. Wrapper models use a search strategy and various subsets are evaluated through a classifier. Embedded techniques integrate FS during the training of the model. Due to their interaction with a classifier, wrapper and embedded approaches are tailored to a specific learning algorithm and have more computational complexity than filter methods. Later on, hybrid and ensemble techniques have emerged by employing these traditional methods in different variations.

Yousef et al. introduced the term "recursive cluster elimination" with their methodology called Support Vector Machines – Recursive Cluster Elimination (SVM-RCE) into the community [5-7]. Their approach has received widespread attention and adopted extensively in the field. In SVM-RCE, after separation of samples for training and test, initially genes (features) are subject to a t-test to select top 1000 features and then these features are grouped into clusters using K-means algorithm. Next, these clusters are scored by SVM for their classification capability and those with low scores are eliminated at a predefined rate. Remaining genes from surviving clusters are combined; and all the steps are repeated recursively until a predetermined number of clusters remains. Weis et al. [8] presented a SVM-RCE like approach in which following the determination of feature clusters, accuracy is measured including all clusters at first and then each cluster is dropped sequentially to maximize accuracy. Luo et al. [9] applied infinite norm of weight coefficient vector from the SVM model in order to give a score for each cluster rather than scoring by cross-validation. Their algorithm demonstrated significant reduction in running time while performing comparatively as in SVM-RCE. In [10], SVM-RCE is employed for classification between individuals with posttraumatic stress disorder (PTSD), post-concussion syndrome (PCS) + PTSD, and controls. Connectivity paths obtained from 125 brain regions are taken as features and they achieve higher classification performance via imaging-based grouping compared to conventional grouping. Jin et al.[11] proposed a similar study and included SVM-RCE with minor modifications in their experiments to show that dynamic functional and effective connectivity provide better accuracy compared to static connectivity. Yet another study [12] exploits SVM-RCE to evaluate the performance for various

feature sets with the intention of elucidating biomarkers of autism spectrum disorder (ASD).

In this study, we seek to analyze the effects of four different clustering algorithms on SVM-RCE. In the original framework, K-means [13] is used for clustering. Here, this algorithm is replaced with three clustering algorithms (i.e., k-medoids, hierarchical and gaussian mixture model (GMM) clustering). The performances of four distinct clustering algorithms were comparatively evaluated on five gene expression datasets. Section 2 of this paper presents an overview of the considered clustering algorithms. In Section 3, we present our experimental results via comparing the effects of different clustering algorithms. Section IV concludes the study.

II. Theoretical Background

A. K-means Clustering

K-means clustering [13] is one of the widely used unsupervised learning algorithms that is mainly employed to discover data structure and generate clusters. With the number of clusters (k) known in advance, the initial step is to determine cluster centers by selecting k features randomly as initial cluster centers. Then, features are assigned to these k clusters according to their proximities. When all features are assigned, cluster centers are recalculated and features are partitioned into clusters again by their distance to new cluster centers. This phenomenon continues until there is no assignment of any feature to a new cluster. Euclidean distance is generally used to determine the nearest distance between the data point and cluster center, and defined for two vectors x= (x_1, x_2, …, x_k) and y = (y_1, y_2, …, y_k) as

$$d_{xy} = \left[\sum_{i=1}^{k}(x_i - y_i)^2\right]^{1/2} \quad (1)$$

Ultimately, this algorithm aims at minimizing the objective function

$$E = \sum_{i=1}^{m}\sum_{j=1}^{n} I(x_i \in C_j)\|x_i - m_j\|^2 \quad (2)$$

where m and n represent total number of samples and clusters, respectively along with each sample x_i and cluster center m_j. $I(F) = 1$ if F is true and 0 otherwise.

B. K-medoids Clustering

K-means clustering is sensitive to outliers since distribution of data may be deformed to a great extent by an object with a very large value. This requires another approach that is more robust to dirty and noisy data, which are likely to occur in real environments. K-medoids clustering is more robust than k-means since the k-medoids algorithm selects data points as cluster centers instead of taking the mean of object values [14]. Hence, a medoid refers to an object of a dataset. A medoid can be interpreted as the object of a cluster, whose average dissimilarity to all other objects in the cluster is the least. In other words, it's the most centrally located data point in the dataset.

K-medoids clustering is similar to k-means clustering in that it starts with randomly selecting k representative objects or medoids and each remaining object is assigned to its nearest medoid. However, k-medoids clustering employs medoids as reference points rather than averaging object values in each cluster. After the selection of initial medoids, the method updates medoids iteratively in every assignment of a data point to a cluster. This cycle continues until there is no change in any medoids. This method operates based on minimizing the sum of the dissimilarities between each data point and its corresponding medoid. An absolute error criterion is considered and defined as:

$$E = \sum_{j=1}^{k}\sum_{x\in C_j} |x - m_j| \quad (3)$$

where E is the sum of absolute error for all data points in the dataset; x is the data point representing an object in cluster C_j; and m_j is the medoid of cluster C_j.

C. Hierarchical Clustering

Hierarchical Clustering (HC) is another type of widely used clustering algorithm in data exploratory analysis. In HC, each instance of data is initially considered as a single cluster and then two clusters with the closest distance or the highest similarity are merged according to a given metric. This is repeated until all the instances belong to a single hierarchically connected cluster. The result of HC is represented as a cluster tree called dendrogram that is a graphical representation showing agglomerated clusters at each step [15]. Dendrograms allow visualization of formation of clusters consecutively and understanding the underlying structure of the data. In addition, specification for the initial number of clusters is not needed and the desired number of clusters can be obtained by selecting a cutoff point in the hierarchy.

HC approaches can be separated into two types: agglomerative (bottom-up) and divisive (top-down) [16]. Agglomerative approach represents each instance as a cluster at first and combines them iteratively until the formation of the single (root) cluster. In the divisive approach, all instances are in the same cluster at the beginning and then the cluster is divided recursively into clusters until a single instance remains in them. Agglomerative Hierarchical Clustering (AHC) is more widely used than Divisive Hierarchical Clustering (DHC) due to the sophistication of DHC, and easy implementation of AHC.

In HC, the distance between pairs of instances is determined by different distance metrics such as Euclidean, Minkowski, or Manhattan, where Euclidean distance is typically preferred. With determination of pairwise distances, we need some procedure called linkage in order to quantify the pairwise distances between clusters and merge them. Some of the most widely used linkage types are single linkage, average linkage, complete linkage, centroid linkage and Ward's method [17].

D. Gaussian Mixture Model Clustering

Gaussian Mixture Model (GMM) clustering is a model-based algorithm and employs a mixture of Gaussian distributions to train the data and implement a soft partition [18]. A GMM can be viewed as a linear association of distinct Gaussian components with density function

$$p(\mathcal{X}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathcal{X}|\mu_k, \sum_k) \quad (4)$$

where K is the number of components in the model, π_k is the mixing proportion of the kth component and

$$\mathcal{N}(\mathcal{X}|\mu_k, \sum_k) = \frac{1}{(2\pi)^{D/2}\sqrt{|\sum_k|}} exp\left(-\frac{1}{2}(\mathcal{X}-\mu_k)^T \sum_k{}^{-1}(\mathcal{X}-\mu_k)\right) \quad (5)$$

is the kth Gaussian density function with mean μ_k, covariance $\sum_k$. The parameters of the model are represented by $\theta = \{\pi_1, \mu_1, \sum_1, \ldots, \pi_k\}$.

Given a set of N observations $\{x_1, x_2, \ldots, x_N\}$, θ is approximated by maximum likelihood estimation (MLE) method and computed as

$$\mathcal{L}(\theta) = \log p(\mathcal{X}|\theta) = \log \prod_{i=1}^{N} p(x_i|\theta) \quad (6)$$

$$= \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu, \sum)\right) \quad (7)$$

III. Results and Discussion

A. Experimental Setup

We carried out our experiments on 5 gene expression datasets that are associated with different kinds of complex diseases, and these datasets were downloaded from the GEO [19] database. Each dataset has two classes, i.e., positive and negative, with a specific number of samples. Table I lists the details of the employed datasets.

We employed Knime [20] platform for execution of SVM-RCE with different clustering algorithms. We have run SVM-RCE with each clustering algorithm separately, and collected results for 10 iterations. Performance metrics using Monte Carlo cross-validation were collected for 5 repetitions. We used a fixed number of cluster pattern as 90, 80, 70, 60, 50, 40, 30, 20, 10, 5, 2, 1 during cluster elimination at each iteration. For each cluster level, we obtained gene number and other performance metrics and got averaged values for 10 iterations . In order to obtain a balanced data, if the number of one class label is more than twice as large as another, then the size of the bigger class is reduced to twice the smaller label. If not, then the class with the bigger size is reduced, and it is equalized to the smaller one.

B. Results

In our experiments, we examined the average number of features (genes) and Area Under Curve (AUC) metrics obtained for four clustering algorithms. In order to make a fair comparison, we have selected a similar amount of average number of genes along with their corresponding average AUC values. Table II shows the average gene number and AUC for 10 iterations. Columns A, B, C and D represent use of K-means, K-medoids, HC and GMM, respectively.

K-medoids clustering obtains the highest accuracy only for GDS2547 and HC provides almost the same accuracy with more genes. In other datasets, k-medoids is not so effective and gives the lowest classification performance. For GDS3268, HC and GMM perform similarly but GMM achieves it with less number of features. K-means presents a similar performance with a similar number of genes as GMM.

TABLE I. Basic Information on Datasets

Dataset	# of Genes	# of Samples	Classes
GDS2547	12647	164	positive: 89 negative: 75
GDS3268	44290	200	positive: 129 negative: 71
GDS3646	22186	132	positive: 110 negative: 22
GDS3875	22646	117	positive: 93 negative: 24
GDS5037	41001	108	positive: 88 negative: 20

In GDS3646, k-means slightly outperforms HC with a few more genes. For GDS3875, k-means provides the highest classification accuracy with almost the same number of genes as GMM. HC and GMM have the same performance and they reach an accuracy similar to k-means but HC does it with around less than one and a half times as many genes as k-means and GMM. For GDS5037, k-means surpasses all other clustering algorithms significantly with almost the same number of genes as GMM. Overall, SVM-RCE with k-means, which is the original case, is superior to integration of other clustering algorithms. Note that HC provides a comparable accuracy with k-means with similar number of genes. It is also noteworthy that the deviation in accuracies is minimal for k-means compared to others and this shows stability of it among datasets. Although k-medoids attains smallest number of genes on average, its effect on performance is the lowest.

We have also measured the execution time of SVM-RCE with different clustering configurations for 10 iterations. Average time consumed by each configuration per iteration is shown in Fig. I. As shown in Fig. I, k-medoids algorithm has the highest execution time per iteration besides its low performance. Actually, it is known that k-medoids algorithm can take high execution time due to its exhaustive search in ascertaining the cost for swapping medoids [21]. HC provides the smallest execution time, and the execution times of other clustering algorithms are very close to HC.

IV. Conclusion

SVM-RCE is an impressive technique in FS because it eliminates weakly scoring clusters and combines features from surviving clusters iteratively so that highly scoring clusters including strongly relevant features remain in the final outcome. After separation of data, SVM-RCE starts with clustering whose result plays a crucial role before eliminating clusters. The clustering quality of SVM-RCE affects the eliminated clusters, thereby altering measured accuracies,

TABLE II. Average Gene and Auc Values for Four Clustering Algorithms

Dataset	Avg Gene #				Avg AUC #			
	A	*B*	*C*	*D*	*A*	*B*	*C*	*D*
GDS2547	72.1	60.2	76.7	72.8	0.80	0.85	0.83	0.74
GDS3268	41.5	51.5	63.3	43.8	0.70	0.61	0.73	0.72
GDS3646	27.8	22.5	21.8	24.7	0.76	0.56	0.73	0.66
GDS3875	42.1	39.8	28.1	45.4	0.79	0.73	0.77	0.77
GDS5037	38.3	27.0	31.2	37.5	0.72	0.56	0.56	0.63
Average:	44.4	40.2	44.2	44.8	0.75	0.66	0.72	0.70

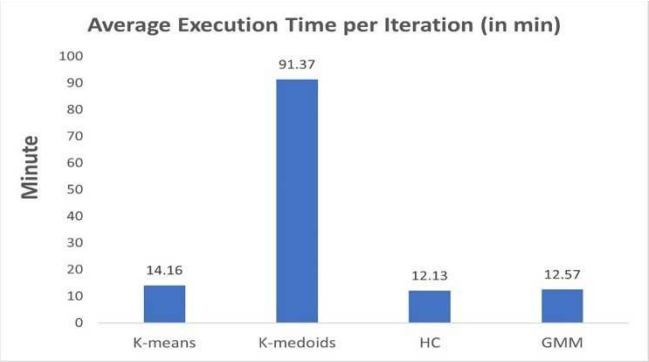

Fig. I. The effects of clustering algorithms on execution time per iteration

average number of genes, and even running time. The aim of this study is to analyze the effects of different clustering algorithms on SVM-RCE and make comparisons between them. We considered four different clustering methods independently: k-means, k-medoids, HC, and GMM. K-means is the default algorithm used in SVM-RCE Hence, we actually compared the original SVM-RCE with three different clustering algorithms. The results of the experiments showed the dominance of k-means algorithm in classification performance, and competitive performance of HC. K-medoids presented the lowest performance in general and it achieved the best accuracy just for one dataset. The performance of GMM was close to HC, but with slightly less accuracy. It has also been observed that all clustering algorithms, except k-medoids, have similar effects on the running time.

To conclude, impacts of k-means and HC are noteworthy with respect to average accuracies. K-means is identified as the most stable algorithm across different datasets. The gap between k-means and HC in terms of accuracy is not large, making HC a promising clustering algorithm for SVM-RCE. Our future work will employ more datasets and increase the number of iterations to maintain stability of the results. We will also deepen our analyses on the selected genes by different configurations.

REFERENCES

[1] C. Jie, L. Jiawei, W. Shulin, Y. Sheng, "Feature selection in machine learning: A new perspective," *Neurocomputing*, *300*, pp. 70-79, July 2018.

[2] P. Dhal, C. Azad, "A comprehensive survey on feature selection in the various fields of machine learning," *Applied Intelligence*, pp. 1-39, March 2022.

[3] L. Abdallah, W. Khalifa, L. C. Showe, M. Yousef, "Selection of significant clusters of genes based on ensemble clustering and recursive cluster elimination (RCE),". J Proteomics Bioinform, 10(8), 2017, pp. 186-192.

[4] B. Bakir-Gungor., H. Hacilar, A. Jabeer., O. U. Nalbantoglu, O. Aran, M. Yousef, "Inflammatory bowel disease biomarkers of human gut microbiota selected via different feature selection methods," PeerJ, 10, e13205, April 2022.

[5] M. Yousef, S. Jung, L. C. Showe, M. K. Showe, "Recursive Cluster Elimination (RCE) for classification and feature selection from gene expression data". BMC bioinformatics, 8(1), pp. 1-12, May 2007.

[6] M. Yousef, B. Bakir-Gungor, A. Jabeer, G. Goy, R. Qureshi, L. C. Showe, "Recursive Cluster Elimination based Rank Function (SVM-RCE-R) implemented in KNIME," F1000Research, 2020, 9.

[7] M. Yousef, A. Jabeer, B. Bakir-Gungor, "SVM-RCE-R-OPT: Optimization of Scoring Function for SVM-RCE-R," in Database and Expert Systems Applications - DEXA 2021 Workshops. Editors G. Kotsis, A. M. Tjoa, I. Khalil, B. Moser, A. Mashkoor, J. Sametinger, et al. (Springer International Publishing), Salmon Tower Building, 1479, pp. 215–224, September 2021.

[8] D. C. Weis, D. P. Visco Jr, J. L. Faulon, "Data mining PubChem using a support vector machine with the Signature molecular descriptor: classification of factor XIa inhibitors," Journal of Molecular Graphics and Modelling, 27(4), pp. 466-475, November 2008.

[9] L. K. Luo, D. F. Huang, L. J. Ye, Q. F. Zhou, G. F. Shao, H. Peng. "Improving the computational efficiency of recursive cluster elimination for gene selection," IEEE/ACM transactions on computational biology and bioinformatics, 8(1), 2010, pp. 122-129.

[10] D. Rangaprakash, et al., "Compromised hippocampus-striatum pathway as a potential imaging biomarker of mild-traumatic brain injury and posttraumatic stress disorder," Human brain mapping, 38(6), 2017, 2843-2864.

[11] C. Jin, et al., "Dynamic brain connectivity is a better predictor of PTSD than static connectivity," Human Brain Mapping 38, 4479–4496, June 2017.

[12] N. Chaitra, P. A. Vijaya, G. Deshpande, "Diagnostic prediction of autism spectrum disorder using complex network measures in a machine learning framework," Biomedical Signal Processing and Control, 62, 102099, September 2020.

[13] S. Na, L. Xumin, G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," In 2010 Third International Symposium on intelligent information technology and security informatics, Ieee, 2010, pp. 63-67.

[14] A. Bhat, "K-medoids clustering using partitioning around medoids for performing face recognition," International Journal of Soft Computing, Mathematics and Control, 3(3), pp. 1-12, August 2014.

[15] C. F. Olson, "Parallel algorithms for hierarchical clustering," Parallel computing, 21(8), 1995, pp. 1313-1325.

[16] T. Li, A. Rezaeipanah, E. M. T. El Din, "An ensemble agglomerative hierarchical clustering algorithm based on clusters clustering technique and the novel similarity measurement," Journal of King Saud University-Computer and Information Sciences, 34(6), pp. 3828-3842, June 2022.

[17] A. Habib, M. Akram, C. Kahraman, "Minimum spanning tree hierarchical clustering algorithm: a new Pythagorean fuzzy similarity measure for the analysis of functional brain networks," Expert Systems with Applications, 201, 117016, September 2022.

[18] C. M. Bishop, N. M. Nasrabadi, Pattern recognition and machine learning, 1st edn., Springer New York, NY, 2006, p. 738.

[19] E. Clough, T. Barrett, "The gene expression omnibus database," Statistical Genomics: Methods and Protocols, 2016, pp. 93-110.

[20] M. R. Berthold, et al., "KNIME: the konstanz information miner," Studies in Classification, Data Analysis, and Knowledge Organization, 2007, pp. 319-326.

[21] H. S. Park, C. H. Jun, "A simple and fast algorithm for K-medoids clustering," Expert systems with applications, 36(2), 2009, pp. 3336-3341.