LVCluster: Bounded Clustering using Laguerre Voronoi Diagram

Abu Quwsar Ohi
Department of Computer Science
University of Calgary
Calgary, AB, Canada
mdabuquwsar.ohi@ucalgary.ca

Marina L. Gavrilova
Department of Computer Science
University of Calgary
Calgary, AB, Canada
mgavrilo@ucalgary.ca

ABSTRACT

Clustering, a fundamental technique in unsupervised learning, identifies similar groups within a dataset. However, clustering algorithms encounter limitations when requiring a predetermined number of clusters/centroids/labels. This paper proposes a novel approach of clustering by integrating concepts from Voronoi diagrams in Laguerre geometry, namely, **L**aguerre **V**oronoi **Cluster**ing (LVCluster). Laguerre geometry introduces circles by adding radius weight metric to centroids, enabling dynamic exclusion from clustering criteria. Consequently, this approach offers flexibility by necessitating only one hyperparameter, an upper-bound value for the number of circles. LVCluster can be optimized using gradient descent and can be jointly optimized with deep neural network architectures. The experimental results indicated that LVCluster outperforms clustering algorithms when trained individually and jointly with deep neural networks on increased cluster centroids.

Keywords

Laguerre Geometry, Voronoi Diagram, Clustering, KMeans, Gradient Descending

1. INTRODUCTION

Clustering, an essential technique in data mining and machine learning, involves grouping data points into meaningful clusters based on their similarities. It is a fundamental tool for exploratory data analysis, pattern recognition, and knowledge discovery in various domains. Over the years, clustering has witnessed extensive research and application across diverse fields, including image analysis, document clustering, bioinformatics, customer segmentation, and anomaly detection, etc.

Clustering algorithms require hyperparameters based on which they extract the similarity of data. Due to the advancement of deep learning [Yan15a] methods, clustering algorithms are being incorporated with deep learning architectures [Maz20a]. The integration of clustering algorithms with deep learning architectures can be performed through either joint or independent training. Joint training indicates training deep learning (DL) models by adding DL objective loss and clustering loss during training. Independent training involves training DL separately with loss function and applying unsupervised clustering algorithms in the learned DL representation.

The most common clustering algorithm used in deep learning is KMeans [Stu82a], due to its simple clustering objective. KMeans clustering objective can be optimized using backpropagation, which gives an advantage of implementation with deep learning architectures. However, the required hyperparameter of the clustering algorithm often limits the usability of clustering algorithms. KMeans algorithm requires the number of centroids/classes as hyperparameters. Pre-defining the number of classes can be critical in unsupervised learning as the number of classes is sometimes impossible to measure in large datasets. On the other hand, density-based clustering algorithms [Mar96a] do not require the number of classes pre-defined. However, it requires a distance value to connect two components into the same group. Selecting the proper density value can be critical based on the representation and deviation of data distribution.

Deep learning architectures have also been applied to clustering data [Jun16a]. These methods aim to project data into a latent space while ensuring the data points in the space are clusterable. Deep clustering techniques have gained popularity due to their ability to automatically extract features from data. However, the objectives of clustering algorithms and deep clustering algorithms are different. Deep clustering algorithms produce projections of data that are clusterable and implement clustering algorithms inside the neural network. In contrast, classic clustering algorithms directly cluster data in the latent space. In both cases, one important problem that has always been an open question is the pre-defined number of clusters to generate. This work introduces a clustering algorithm that gives flexibility in pre-defining the number of clusters using computational

geometry data structure with the property of modern deep learning architecture.

The objective of this paper is to propose a clustering algorithm that does not require a fixed number of centroids or classes during training. Therefore, the clustering algorithm would offer flexibility in varying data distributions. Additionally, the algorithm has to operate without requiring labeled data, enabling unsupervised learning scenarios. Moreover, the algorithm should be optimizable using gradient descent to be jointly trained with deep neural network architectures end-to-end. With all these advantages, the clustering algorithm will be very similar to a deep learning layer, such as softmax, yet unsupervised and loosely dependent on the number of classes and labels.

Considering the above-mentioned constraints, this paper proposes a Laguerre Voronoi clustering algorithm, namely, LVCluster. The proposed algorithm requires one hyperparameter: an upper bound of the centroids/labels of the given input data. LVCluster algorithm dynamically groups clusters and excludes extra cluster regions based on the input data distribution. Therefore, LVCluster does not require knowledge of the exact number of classes in the given data. As a result, it is well suited for unsupervised learning on large datasets using deep neural networks (DNN) with an unknown number of classes.

The overall contributions of the paper are:

- The paper proposes a Laguerre Voronoi Diagram based clustering strategy for unsupervised clustering.
- The approach leverages the properties of Laguerre geometry to introduce adaptability to the identification of necessary classes during the training process.
- The proposed clustering algorithm is trained using backpropagation and is jointly trainable with deep learning methods.

The performance of the proposed clustering algorithm is evaluated in three spatial datasets: Congress voting dataset [Con87a], Iris dataset [Raf88a], and Breast cancer dataset [Wol95a]. Moreover, the performance of the proposed clustering algorithm is evaluated when it is jointly trained with deep neural networks to cluster speakers in an open-set scenario.

2. LITERATURE REVIEW

Clustering algorithms have been effectively studied and applied across various domains of machine learning. Clustering algorithms have been one of the fundamental approaches in unsupervised learning algorithms. Early clustering algorithms such as KMeans [Stu82a] is the most widely used partitioning-based algorithms due to its simplicity and efficiency. KMeans algorithm has different optimization techniques, among which Lloyd's [Stu82a] optimization is widely utilized. Lloyd's algorithm is also known as Voronoi iteration as it tries to partition a set of data into Voronoi cells. The center of each Voronoi cell is referred to as the site, which is learned in Lloyd's algorithm through iteration. Any data point in a Voronoi cell has a minimum distance to the Voronoi site. Lloyd's algorithm is specifically formulated to work on Euclidean spaces. In Euclidean space, Lloyd's algorithm cannot penalize unnecessary Voronoi sites on demand. As a result, the number of Voronoi cells must be correctly pre-defined. Apart from Lloyd's iterative algorithm, other approaches use gradients to learn optimal sites [Leo94a].

The popularity of deep learning algorithms has contributed to the increased importance of the KMeans algorithm, primarily due to its adaptability to be trained with gradient descending approach. As a result, various unsupervised deep learning algorithms utilized KMeans clustering's loss function jointly in the training process [Maz20a]. However, the only limitation that the KMeans algorithm faces is the pre-defined number of centroids or Voronoi cells. The pre-definition limits its usability as it is challenging to know the number of classes beforehand in self-supervised learning.

Density-based clustering algorithms, such as DBSCAN [Mar96a] and OPTICS [Mih99a], focus on identifying regions with high-density data space. However, these algorithms are not optimizable using a gradient descending approach. Density-based algorithms are prone to density parameters. As a result, it is challenging to integrate density-based algorithms with deep learning as the density distribution produced by the deep learning models can change over time.

Deep learning architectures can excellently produce clusterable data by learning the inherent representation of the given data. Further, they can generate clusterable embedding vectors based on appropriate loss functions. Deep embedded clustering (DEC) [Jun16a] is one of the fundamental approaches that produces clusterable embedding from image data using KMeans clustering loss. Although it performs excellently on image representation, it is still pruned to the fixed number of clusters/classes to produce. If the number of classes is approximated, DEC would fail to produce projection form data.

The idea of identifying the number of centroids in the KMeans algorithm is not new. XMeans [Pel20a] is an early improvement of the KMeans algorithm that finds appropriate cluster regions by repeated subdivision. Unsupervised KMeans [Sin20a] introduced a version of the KMeans algorithm that does not require the number of centroids as a hyperparameter. Even though the improved KMeans algorithms optimally approximate the centroids dynamically, they lack the ability to be trained using backpropagation. Hence,

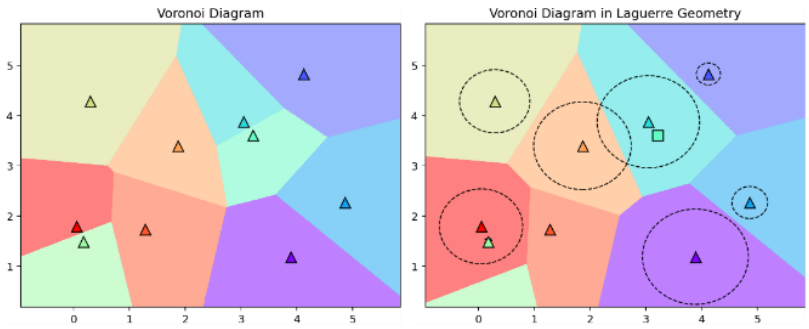

Figure 1 Difference of Voronoi diagram in Euclidean and Laguerre geometry.

recently enhanced KMeans algorithms cannot be implemented in a deep learning setup.

This paper introduces a novel clustering algorithm that inherits principles from principles of Laguerre geometry to dynamically determine the number of centroids. Instead of fixing the number of clusters, the algorithm requires an upper bound value of the number of clusters. Moreover, the clustering algorithm is trainable using backpropagation. As a result, it can be jointly trained with deep learning architectures simultaneously. The proposed clustering algorithm alleviates the need to fix the number of clusters, making the clustering algorithm truly unsupervised.

3. METHODOLOGY

The clustering algorithm inherits the basic properties of KMeans while incorporating the properties of Laguerre geometry. The algorithm can cluster spatial data. The algorithm requires an upper bound of the number of clusters. The clustering algorithm has two stages: a) initialization and b) optimization. Similar to classic KMeans, the proposed clustering method is sensitive to initial region selection. After the initialization, the circles are adjusted using backpropagation. First, the concept of the Voronoi Diagram in Laguerre geometry is discussed. Thereafter, the initialization and optimization steps are discussed in the following section.

Voronoi Diagram in Laguerre Geometry

In Laguerre geometry cluster circles are formed instead of cluster centroids. Circles in Laguerre Voronoi Diagram are set of points $C = \{c_1, c_2, \ldots, c_k\}$, $c_i \in \mathbb{R}^d$ with a corresponding set of radius $R = \{r_1, r_2, \ldots, r_k\}$, $r_i \in \mathbb{R}$. The coverage of circle c_i is the region that is of the minimum distance to the circle c_i. In Laguerre geometry, the coverage of a circle c_i can be controlled by increasing the value of the corresponding radius r_i and vise versa. The coverage of a circle is defined as a Voronoi polygon (convex polygon), which can be mathematically interpreted as,

$$V(c_i) = \cap_j \{P \in \mathbb{R}^d | D_L(c_i, P) \leq D_L(c_j, P)\} \quad (1)$$

Here, $P \in \mathbb{R}^d$ is a set of points in the plane. $D_L(\cdot,\cdot)$ is a distance function in terms of Laguerre geometry derived as follows,

$$D_L(c_i, P) = d(c_i, P) - r_i^2 \quad (2)$$

Intuitively, the equation measures the tangent line from point P to the circle $(c_i; r_i)$. $d(.,.)$ could be any distance metric function. Whenever a Voronoi diagram is drawn using the above formula of Laguerre geometry, the formed partition of the whole frame is called the Laguerre Voronoi Diagram (LVD). A gamut of applications in material sciences [Ale04a], biometrics [Che06a], and networks [Kef09a] benefitted from advanced algorithms of the Voronoi diagram and Laguerre Voronoi diagrams. One of the interesting properties of the Voronoi diagram is a Voronoi polygon can have zero area if it can be formed using one or more Voronoi polygons. Due to this property, LVD is different from Voronoi diagrams in Euclidean space, formed by KMeans and any other clustering algorithms. A circle can exist in LVD without covering any region, resulting in having no cluster assignment. Figure 1 depicts an example of the classic Voronoi Diagram and Voronoi Diagram in Laguerre Geometry. The triangles indicate centroids/sites, and the color background indicates the region/Voronoi cell of the centroids. Dashed circles demonstrate the radius of circles in Laguerre geometry. Squares indicate that the circle was not assigned any region/Voronoi cell, which is only possible in the Voronoi diagram in Laguerre geometry.

Laguerre Voronoi Clustering

The proposed algorithm leverages the property of an empty polygon in LVD to its advantage by setting an upper bound on the number of clusters. Therefore, the proposed algorithm can find the required number of cluster regions during training and shrink the remaining centroids. Figure 1 shows an example of the property. In Voronoi diagram based clustering (Figure 1-left), each centroid must have its Voronoi polygon on the space. On the contrary, LVD can dynamically shrink the area of Voronoi polygons to zero (Figure 1-

right). With proper optimization, LVCluster can learn the optimal polygon regions and disregard the excess centroids. This dynamic adaptation to the number of centroids is the novelty of the proposed LVCluster algorithm. In the following two sections, the initialization stage and optimization stage of the proposed algorithm are discussed.

3.1.1 Initialization

The initialization step defines circle centers and radii based on the input data. Initialization is an important step in clustering, as inaccurate initialization can cause the optimization to get trapped in local minima [Dav07a]. The initialization process of the proposed clustering algorithm follows the KMeans++ [Dav07a] initialization process, used in KMeans clustering. The circle centers c_i are first initialized using the KMeans++ initializer. Afterward, the radius r_i is derived from the minimum distance for each center to any other center. Using KMeans++ ensures that all the centers are placed around data points while maintaining distance among the other centers. Having different radii ensures coverage overlaps, which is to be optimized using a gradient descending algorithm.

3.1.2 Optimization

After proper initialization of the circles, their optimal placement is found using backpropagation. In the proposed algorithm the distance function to generate Voronoi polygon is stated as follows,

$$D_L(c_i, P) = d(c_i, P) - \sigma(r_i)^2 \quad (3)$$

Here $\sigma(\cdot)$ is sigmoid activation function used to give a non-linearity to the optimization problem. $d(.,.)$ is cosine distance function.

The objective of the clustering algorithm is to minimize the distance between the circle and the data points belonging to the Voronoi polygon of that circle. The objective can be mathematically interpreted as follows,

$$L(X, C) = \sum_{i=1}^{n} \min_{c_j \in C} D_L(c_j, x_i) \quad (4)$$

The function is used as a loss function and gradients of the center and radius are calculated based on the given loss function. Here, $x_i \in X$ is the set of data points given to the clustering algorithm as input. The circle center and radius are updated using the gradient descending algorithm with a fixed learning rate η.

3.1.3 Deep Learning Integration

The proposed LVCluster can be combined with deep learning architectures and can be trained jointly to achieve the clustering objective. The joint loss function can be derived as $L = L_{DNN} + L_C$. Here, L_{DNN} is the loss function of DNN that could be representation loss, reconstruction loss, or elbow loss. L_C is the clustering loss function derived in Eqn (4). Based on the architectural constraints, the clustering loss function can flow the gradients to the DNN. As the LVCluster algorithm adheres the same implementation strategy of KMeans clustering algorithm (disregarding the minor additional computation of $\sigma(r_i)^2$ in Eqn (3)), the computational complexity of both of the algorithms are identical.

4. EXPERIMENTAL RESULTS

Datasets

Three spatial datasets were used for the evaluation. The datasets are described below:

- Congress voting [Con87a] is a classification dataset containing two classes. The dataset has a total of 16 features explaining different conditions of voting. Some of the features were missing from the dataset, which were removed before training.
- Iris [Raf88a] is a classification dataset containing three classes of 50 instances each. It is a spatial dataset with four features. The dataset is widely used in statistics and machine learning.
- Breast cancer [Wol95a] dataset is a classification dataset containing two classes (malignant and benign). The dataset has 30 features extracted from a digitized image of a fine needle aspirate of a breast mass. The cancer dataset holds 569 instances in total. The features were standardized by removing the mean and scaling to unit variance.

Apart from the real-world datasets, synthetic datasets were used for the visualization of clusters. The synthetic datasets were generated using the sklearn [Fab11a] library with a standard deviation of 3.5.

Implementation

Three metrics were used for evaluation. The metrics are elaborated below:

- Accuracy: Due to an unsupervised learning strategy, the clustering algorithm generated pseudo labels for each data point. Therefore, the accuracy metric indicates the maximum match by mapping the pseudo labels with the ground truth labels.
- Adjusted Rand Index: The adjusted rand index is extracted and calculated from the contingency table determining the quality of the generated pseudo label depending on the ground truth.
- Normalized Mutual Information: Normalized mutual information extracts normalization of the mutual score by comparing the clustering algorithm-generated pseudo labels with the ground truth.

The three algorithms generate a real value score [0, 1]. A higher score from the metrics would indicate the quality of the pseudo labels compared to the ground truth. However, the number of centroids defined in the

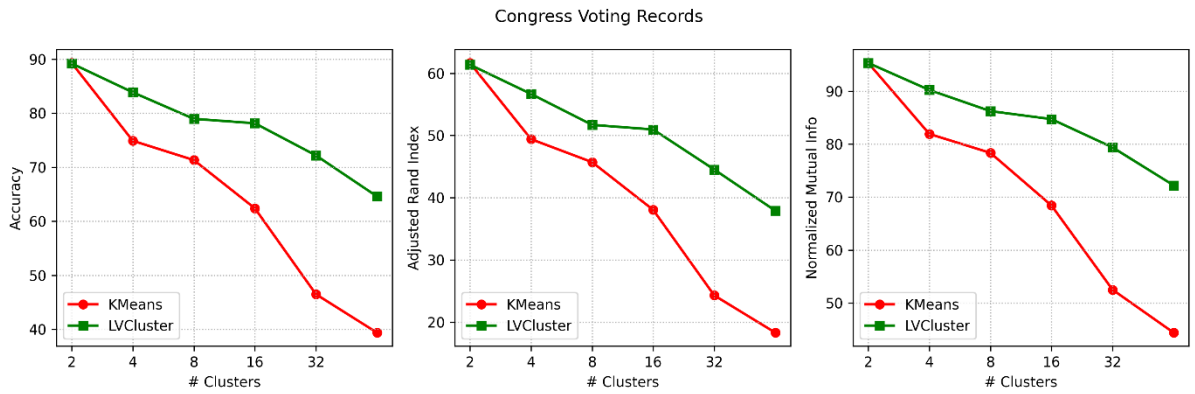

Figure 2 Performance comparison on congress voting records dataset.

Figure 3 Performance comparison on Iris dataset.

clustering algorithm would not be the same as the number of ground truth labels. Therefore, the metrics' higher score would evaluate how well the clustering algorithm identifies the number of centroids along with the clustering problem relative to the ground truth.

PyTorch [Ada19a] was used to build the clustering algorithms that support automatic differentiation. The clustering algorithms were additionally trained for open-set speaker recognition, aiming to investigate their performance and behavior when applied to the outputs of DNNs. The training criteria were implemented from [Abu23a]. The models were trained on the VoxCeleb1 [Ars17a] dataset in an unsupervised strategy and tested on different speakers on the VoxCeleb1 [Ars17a] dataset.

Comparison Analysis

The analysis explores the influence of the initial number of clusters, which serves as a hyperparameter provided to the clustering algorithm during the initialization stage. Figure 2 reports a performance comparison of the KMeans and LVCluster algorithms on congress voting records dataset. Both clustering algorithms show optimal performance when given the accurate number of centroids. KMeans and LVCluster show similar performance when the centroid was set to 2. Increasing the number of centroids causes a reduction in performance for the KMeans algorithm. In contrast, the LVCluster shows less performance degradation with an increasing number of clusters.

Figure 3 depicts a performance comparison on the Iris dataset with three classes. Both clustering algorithms show degraded performance when the number of centroids was set to two, which is less than the actual number of classes. Initializing the clustering algorithms with three shows the best performance for both algorithms. Further increasing the number of centroids causes a reduction of performance for the KMeans algorithm. LVCluster also shows performance degradation while increasing centroids. However, compared to LVCluster degradation is slighter than KMeans performance degradation.

Figure 4 reports a performance comparison on the breast cancer dataset with two classes. Both clustering algorithms show a similar trajectory of degradation. Each class of the dataset contains smaller and independent sub-clusters. Therefore increasing the initial number of clusters in the clustering algorithm causes the clustering algorithm leads it to treat the sub-clusters as separate classes. As a result, the LVCluster struggles to effectively reduce the number of cluster

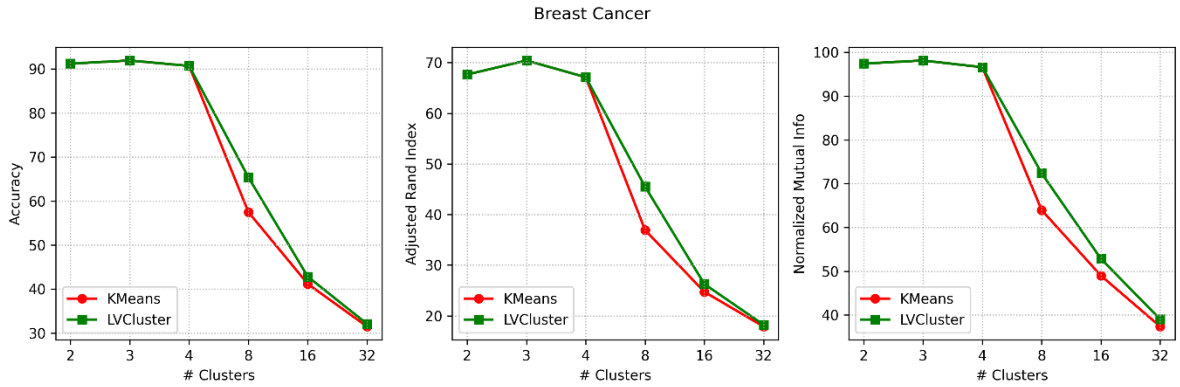

Figure 4 Clustering performance comparison on breast cancer classification dataset.

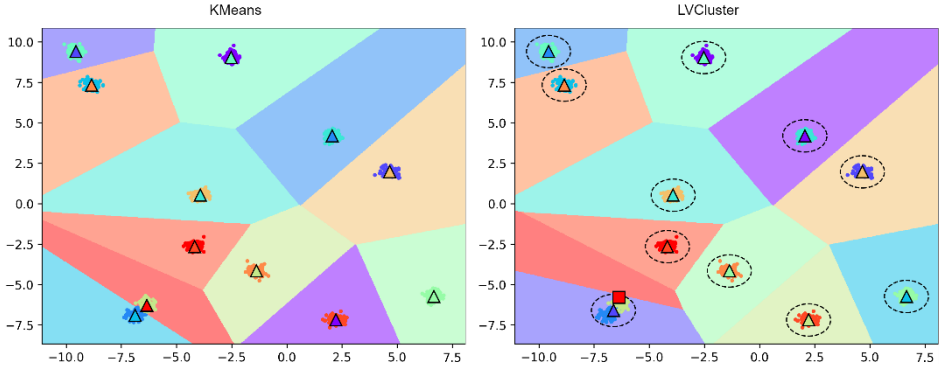

Figure 5 Voronoi diagram comparison of KMeans and LVCluster.

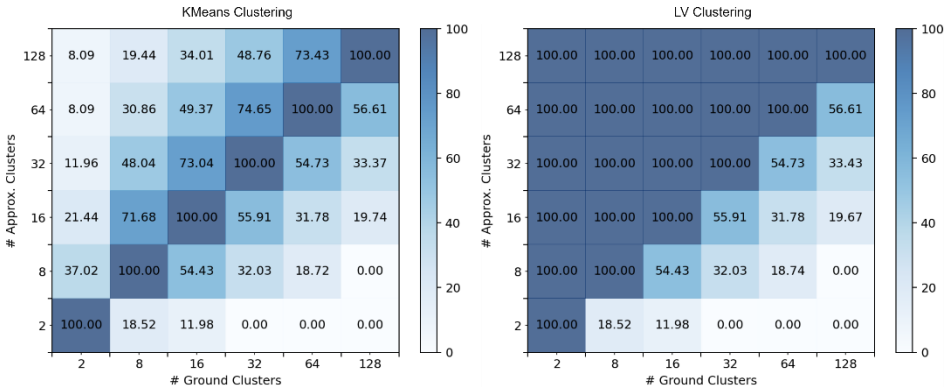

Figure 6 Adjusted rand index on different cluster choices.

regions relative to the original number of clusters defined in the dataset.

In the above comparisons, LVCluster shows an upper-bound performance compared to the KMeans algorithm. Moreover, LVCluster stabilizes the performance when the number of clusters were increased. As LVCluster can remove unnecessary centroids from the dataset, it can detect sub-optimal cluster representation. However, the performance of LVCluster on increasing the number of centroids depends on how well the clusters are distributed.

Figure 5 shows an example of the Voronoi diagram of the centroids. The example has 12 cluster regions where 10 of the cluster regions are well-separated. Regardless, two of the cluster regions are tightly bound together. From the perspective of the distribution of the data, there are 11 centroids as two of the regions are so close that they can be considered as one cluster region. KMeans clustering with 12 centroids would generate 12 regions. However, as the data is not well-separated, the LVCluster would generate 11 regions considering the two tightly bounded regions as one. From the perspective of data

Table 1 Comparison of clustering algorithms when trained with DNN.

Algorithm	#Clusters = 40			#Cluster = 128		
	Accuracy	Adjusted Rand Index	Normalized Mutual Information	Accuracy	Adjusted Rand Index	Normalized Mutual Information
KMeans	98.9	96.26	99.4	14.64	20.47	32.73
LVCluster	99.9	96.26	99.9	99.2	95.94	99.8

distribution, performance is a relative factor. Excessive noise and merged regions can cause LVCluster to generate fault assumptions based on the actual classification task.

Figure 6 exhibits a comparison of the adjusted rand index varied by the number of clusters on a well-separated data distribution. LVCluster shows the best result whenever the input number of clusters is greater than the number of actual clusters in the dataset. The overall comparisons highlight that LVCluster can dynamically reduce the required number of clusters based on the data distribution to gain better performance than KMeans clustering.

Table 1 shows the performance comparison of the KMeans and LVCluster algorithm when trained jointly with DNN architecture [Abu23a] to recognize speakers in unsupervised open-set conditions. In an open-set problem, a model is trained and tested on two different datasets with two different speaker sets. Therefore, the number of speakers in the testing set (40 speakers) is unknown to the model. When using the KMeans algorithm to perform clustering on the speaker embeddings, the clustering algorithm generates the same performance as LVCluster when the number of clusters is set to the same as the number of speakers in the test dataset. However, increasing the number of clusters to 128 shows that the performance of the KMeans clustering algorithm degrades drastically. In contrast, the performance of LVCluster slightly reduces. LVCluster dynamically adjusts the number of clusters and finds the best possible clustering representation from the speaker embedding space. Therefore, it can be validated that LVCluster can be useful when the number of clusters of a distribution is unknown, yet the distribution contains a clusterable representation.

5. CONCLUSION

This paper proposes LVCluster, a novel clustering algorithm that inherits the property of Laguerre geometry to dynamically determine the number of centroids. The algorithm requires an upper bound number of clusters as hyperparameters, which often reduces the necessity to identify the number of classes in an unknown dataset. LVCluster shows performance gain than KMeans whenever the number of cluster centroids is set higher than the actual number of cluster centroids. The algorithm is optimized using backpropagation, thereby opening up opportunities for its application in deep learning architectures.

6. ACKNOWLEDGMENTS

The authors acknowledge the Natural Sciences and Engineering Research Council (NSERC) Discovery Grant funding, as well as the NSERC Strategic Partnership Grant (SPG) and the University of Calgary Transdisciplinary Connector Funding for the partial funding of this project.

7. REFERENCES

[Yan15a] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. nature 521, 7553 (2015), 436–444.

[Maz20a] Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. 2020. Deep k-means: Jointly clustering with k-means and learning representations. Pattern Recognition Letters 138, (2020), 185–192.

[Stu82a] Stuart Lloyd. 1982. Least squares quantization in PCM. IEEE transactions on information theory 28, 2 (1982), 129–137.

[Mar96a] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, and others. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd, 226–231.

[Jun16a] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In International conference on machine learning, PMLR, 478–487.

[Pel20a] Pelleg, Dan, and Andrew W. Moore. 2000. X-means: Extending k-means with efficient estimation of the number of clusters. In ICML, vol. 1, pp. 727-734.

[Sin20a] Sinaga, K.P. and Yang, M.S., 2020. Unsupervised K-means clustering algorithm. IEEE access, 8, pp. 80716-80727.

[Con87a] Congressional Voting Records. 1987. UCI Machine Learning Repository.

[Raf88a] R. A. Fisher. 1988. Iris. UCI Machine Learning Repository.

[Wol95a] Wolberg and W. Street. 1995. Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository.

[Leo94a] Leon Bottou and Yoshua Bengio. 1994. Convergence properties of the k-means algorithms. Advances in neural information processing systems 7, (1994).

[Mih99a] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering points to identify the clustering structure. ACM Sigmod record 28, 2 (1999), 49–60.

[Ale04a] Alexey V Anikeenko, MG Alinchenko, VP Voloshin, Nikolai N Medvedev, Marina L Gavrilova, and P Jedlovszky. 2004. Implementation of the Voronoi-Delaunay method for analysis of intermolecular voids. In Computational Science and Its Applications–ICCSA 2004: International Conference, Assisi, Italy, May 14-17, 2004, Proceedings, Part III 4, Springer, 217–226.

[Che06a] Chengfeng Wang, Marina Gavrilova, Yuan Luo, and Jon Rokne. 2006. An efficient algorithm for fingerprint matching. In 18th International Conference on Pattern Recognition (ICPR'06), IEEE, 1034–1037.

[Kef09a] Kefeng Xuan, Geng Zhao, David Taniar, Bala Srinivasan, Maytham Safar, and Marina Gavrilova. 2009. Network Voronoi diagram based range search. In 2009 International conference on advanced information networking and applications, IEEE, 741–748.

[Dav07a] David Arthur, Sergei Vassilvitskii, and others. 2007. k-means++: The advantages of careful seeding. In Soda, 1027–1035.

[Fab11a] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and others. 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research 12, (2011), 2825–2830.

[Ada19a] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and others. 2019. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems 32, (2019).

[Abu23a] Abu Quwsar Ohi and Marina L Gavrilova. 2023. A Novel Self-supervised Representation Learning Model for an Open-Set Speaker Recognition. In International Conference on Computer Information Systems and Industrial Management, Springer, 270–282.

[Ars17a] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2017. Voxceleb: a large-scale speaker identification dataset. arXiv preprint arXiv:1706.08612, (2017).