Learning Curves in Machine Learning

Can patterns be identified amongst learning curves after the application of the K-Means algorithm using point and statistical vectors?

Pravesha S.P. Ramsundersingh

Supervisor(s): Tom Viering[1], Taylan Turan[2]

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
January 28, 2024

Name: Pravesha S.P. Ramsundersingh
Final project course: CSE3000 Research Project
Thesis committee: Tom Viering[1], Taylan Turan[2], Hayley Hung

[1]https://www.tudelft.nl/ewi/over-de-faculteit/afdelingen/software-technology/computer-science-engineering-teaching-team/people/tom-viering

[2]https://www.tudelft.nl/ewi/over-de-faculteit/afdelingen/intelligent-systems/pattern-recognition-bioinformatics/pattern-recognition-bioinformatics/people/taylan-turan

Abstract

A learning curve can serve as an indicator of the "performance of trained models versus the training set size" [1]. Recent research on learning curve analysis has been carried out within the Learning Curve Database (LCDB) [2]. This paper will investigate if there are similarities amongst these curves by clustering those provided by the LCDB. The experiment employs two distinct input parameters: point vectors and statistical vectors. By conducting individual learner analysis, individual dataset analysis, principal component analysis, and other experiments, patterns are isolated for both input sets. Upon further exploration of shapes and distributions, the concluding remark is that the point vector clustering produced one key concrete pattern amongst certain learning techniques. In contrast, the statistical vector findings are more inconclusive and do not exhibit a clear distinction that could be linked to any dominant patterns.

1 Introduction

One of the general expectations within the Machine Learning (ML) field is the idea that more training data results in a better model performance [1]. Nevertheless, as the number of data points increases, the time, complexity, and expense of the process also increases. By indicating an estimate of the number of samples required for a certain model, such challenges could potentially be mitigated. Learning curves, which display the "performance of trained models versus the training set size" [1], are commonly used to gauge this. The current state of research on learning curves is that it is not yet well-understood. This refers to the absence of conclusive findings that indicate a direct correlation between the performance of a model and the shape, parameters, and other characteristics of a learning curve.

This paper will investigate if any significant patterns can be identified after the application of a K-Means clustering algorithm on all learning curves within the given Learning Curve Database (LCDB) [2], using the statistical features of the dataset and raw curves. In search for concrete equivalence relations, the unsupervised learning technique of distance-based clustering was chosen to establish groups within the LCDB data. The features considered include statistical and point vectors extracted from the learning curves. By incorporating both sets of input data in this experiment, the goal is to enhance the likelihood of pattern discovery and broaden the research scope.

The Methodology outlined in Section 3 will indicate the detailed research process, forms of measurements, and pre-processing mechanisms. The Results & Discussion in Section 5 will showcase the outcomes of the clustering algorithm through both data and graphical visualisations of the formed clusters and address the research question to discern patterns. The Conclusions in Section 6 will assess the primary findings and explore potential future additions to the experiment.

2 Relevant Work

Previous studies have delved into clustering concerning learning curves. In 2002, Meek [3] examined the application of learning curve sampling method to model-based clustering. The main objective was to investigate the impact of utilising finite mixture models to maintain accuracy and reduce the runtime of learning curves. In spite of the fact that both model-based clustering and distance-based clustering are commonly used in unsupervised machine learning, the research did not directly group learning curves. However, in 2005, Navarro and Lee [4] employed an alternative form of model-based clustering to address the challenge of partitioning a set of learning curves. The application of minimum description length-based clustering technique yielded an optimal solution, presenting six candidate solutions. Expanding the scope to other curve variants such as principal curves, research by Moraes et al. [5] applied a K-segment algorithm on curves to identify clusters within a dataset. Additionally, in 2012, Tarpey [6] also used the K-means algorithm with an input data of estimated regression coefficients from a curve. Endorsing the application of model-based clustering experiments, similar to the aforementioned studies, Tarpey [6] acknowledged that finite mixture models addressed the limitations of the K-means algorithm that had impacted the results.

Nevertheless, research on clustering large sets of curves based on point or statistical vectors remains an open research area. There is also a gap in experiments with distance-based clustering approaches, such as the K-means algorithm.

3 Methodology

The methodology consists of the tools used throughout the project, pre-processing of data, and algorithmic approach.

3.1 Tools, Software, and Data

To explore learning curves, a Learning Curve Database[3] (LCDB) is provided for experimentation. The system comprises of 20 working learners that can be applied to around 250 datasets. Python is the chosen programming language, selected not only for its extensive package availability but also because previous student projects within this research group have consistently utilised Python and the LCDB. Adhering to this standard facilitates seamless comparison and information exchange with both prior research and the ongoing work of the current team.

3.2 Pre-processing of Data

The experiment is the clustering of curves based on different input parameters. The two input parameters chosen for this research question are point vectors and statistical vectors. Each curve is initially extracted from the LCDB. However, not all combinations of classifiers and datasets were available, leading to fewer curves in the final input sets.

To prevent the shortcoming of distance-based unsupervised learning strategies, the data undergoes min max scaling, within the custom range [-1, 1].

The following pre-processing techniques were applied to form the final input vectors for the algorithm:

[3] https://github.com/fmohr/lcdb/blob/main/README.md

Point Vector

A point vector consists of each point of the learning curve of each dataset. The vector is composed of n points. Given the varying lengths of the learning curves, linear interpolation is used to ensure a consistent number of points n for each curve. The interpolation process involves extending shorter curves to match the maximum curve length within the LCDB.

In the LCDB, the curves comprise of observation at "powers of, i.e. 16, 24, 32, 45, 64, 91, 128, ... until 90% of the dataset size" as indicated by Mohr et al. [2]. This interpolation process does not consider anchor point of the curves. The current interpolation calculation utilises evenly spaced numbers within a [0, 1] interval. This scaling technique essentially represents a form of linearisation as well. Figure 1 illustrates the linear interpolation process applied to a shorter curve as an example.

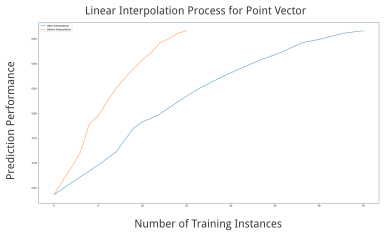


Figure 1: A comparison of a random curve extracted from the Extra Tree classifier before (orange line) and after (blue line) undergoing the linear interpolation procedure

Statistical Vector

A statistical vector consists of mean, standard deviation, skew, and kurtosis. The selection of these features were a personal design choice, influenced by the widespread popularity and familiarity with these mathematical indicators. The vector size is set at 4, corresponding to 4 features that form the statistical basis for each curve. The mean validation scores of each curve in the LCDB are used, and the np functions for each feature is applied to calculate each element in the vector.

3.3 Clustering & Dimensionality Reduction

The chosen algorithm for clustering, categorised as a centroid model based procedure, is the K-means algorithm. The core objective of K-means is to assign a set of data points to one of the K clusters where there is a minimisation of the sum of distances between the data point and assigned cluster centroid [7].

The K-means algorithm was chosen based on its simplicity, scalability, and computational efficiency. The scalability of K-means allows it to operate efficiently within high-dimensional spaces. Lastly, coupled with its simplicity of interpretation and scalability, it has a linear time complexity, making it suitable for handling large sets of data [8].

However, it is crucial to take into account the drawbacks of K-means. Potential disadvantages include distortion in clustering due to outliers, the assumption of spherical shapes, and the need for a predetermined number of K clusters [8]. To guarantee a consistent result and prevent irregular clusters influenced by outliers, the algorithm was executed 100 times for each input vector dataset. The average of the 100 clusters was then considered the final result. The value K was calculated using the Silhouette Score, as discussed in the following section.

Hyper-parameter Tuning

To implement this algorithm on a dataset, a value of K must be assigned. The optimal K value can be defined as the ideal splitting of the given dataset to create well-defined clusters. Determining the 'best' K value differs for each dataset and can be accomplished through two prevalent modeling approaches: the Elbow Method and the Silhouette Score. Since the Elbow Method is somewhat outdated and has shown to not consistently indicate the best value of K [9], the Silhouette Score has been used in this case.

The Silhouette Score is a quantitative measure of the optimal splitting of a given dataset [10]. It takes a range of K values and indicates how well-defined the clusters are. Within each cluster, the Silhouette Score quantifies how well the data fits and how distinct it is in relation to the other created clusters [10]. However, this method has certain drawbacks such as its sensitivity to outliers and irregular shapes, leading to varying results [10]. To ensure consistency and overcome these issues, the Silhouette Score was executed 10 times, and the average was computed to establish the final K value for both input vector datasets.

A graph for each dataset can be generated to indicate the optimal value of K. The graph displays the Silhouette Scores across varying K values, with the highest Silhouette Score indicating the 'best' clustering value. This approach was tested between $K = 2$ and $K = 20$, corresponding to the 20 learners.

Principal Component Analysis

The LCDB contains over 4300 curves, resulting in a cluster array of approximately the same size, making it challenging to analyse and visualise effectively. To examine the generated clusters, Principal Component Analysis (PCA) can be applied. PCA creates a graphical representation of the cluster distribution by reducing dimensionality. Each component explains the maximum variance in data by projecting it onto the eigenvectors of the covariance matrix of the data. The terms 2D and 3D PCA refer to the use of 2 and 3 principal components, respectively. Both 2D and 3D PCA provide a better insight of shape and data distribution.

4 Responsible Research

This section is of utmost importance to ensure that the entire research process adheres to the expectations set by both TU Delft and national research policies.

Throughout the process, prior research mentioned in Section 2 is used to familiarise with the content, serving as a solid foundation for the research question at hand. Prior research involves both scientific papers, and online resources. It is imperative to appropriately reference such information, at the precise location of its mention, throughout the entire paper. This practice upholds academic integrity, enabling readers to trace back to the cited research and understand the logical flow of this project.

To understand the requirements behind academic integrity, the TU Delft Vision on Integrity 2018-2024[4], and the Netherlands Code of Conduct for Research Integrity 2018[5] are used as the guidelines. The core principles articulated in both documents have been integrated into the research process.

Lastly, it is important to ensure the proper reproducibility of results. Implementations of the pre-processing, K-Means algorithm, PCA plots, and average cluster percentages, should be made accessible within a repository[6] upon the project's completion. In addition to code availability, clarity should be maintained throughout all sections of the report regarding the choices made. This transparency enables readers to comprehend the rationale behind each step, facilitating their ability to either understand, reproduce, or extend this work in the future.

5 Results & Discussion

This section is divided into the Silhouette Scores of the K-means algorithm, results for point vector, and results for statistical vector.

5.1 Silhouette Score

The Silhouette Score reached its peak at $K = 2$ for both point and statistical vector input data, as depicted in Figure 2. While there are additional peaks in both scores, none surpass $K = 2$.

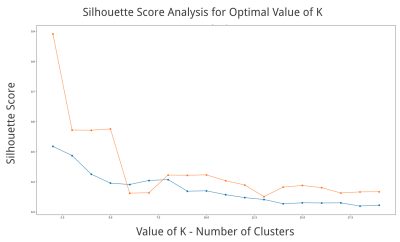


Figure 2: Silhouette Score plots for both point (blue line) and statistical vector (orange line), respectively, indicating the same optimal outcome of $K = 2$

When using the $sklearn$ K-Means function, varying results may be produced based on the input parameter configurations. The function provides an inertia variable, which is a parameter used to assess the similarity of clustering results. If there is minimal fluctuation in inertia, the clusters can be deemed as fairly stable. For the point vector dataset, the inertia value fluctuated within a 0.01 range, while for the statistical vector dataset, there was an even smaller deviation within a 0.0001 range. Both results indicate a high level of stability in the formed clusters using $K = 2$.

[4]https://www.tudelft.nl/over-tu-delft/strategie/integriteitsbeleid/tu-delft-vision-on-integrity-2018-2024

[5]https://www.nwo.nl/en/netherlands-code-conduct-research-integrity

[6]https://github.com/pravesha2000/CSE3000-Research-Project-Learning-Curves-in-Machine-Learning

5.2 Point Vector

Percentages

Seeing as there are many datasets per learner, the average cluster assignment per learner was calculated in an attempt to identify any learner-related patterns within the results.

All learners, with the exception of the Quadratic Discriminant (QDA) Learner, were predominantly averaged into Cluster 0. However, upon closer inspection of the percentages, many learners demonstrated a relatively even distribution, hovering around a 50-60% tipping point, with a slight preference for Cluster 0.

The methodology behind the QDA algorithm is most similar to that of the Linear Discriminant learner (LDA). The LDA also has a comparably low percentage at 54.6%. Nonetheless, it can be argued that the results are not similar enough.

An interesting group is formed at the 75-80% range with the following classifiers: Extra Tree (Ensemble), Random Forest, and Gradient Boosting. All three of these learners are categorised as ensemble learning techniques [11]. In terms of their underlying algorithm structure, this particular group of learners are considered the most similar when compared to the other 17 learners.

Regarding other possible significant findings, although Sigmoid and Logistic Regression learners might be considered similar in their use of the sigmoid function, they are separated by a margin of 20.3%.

And finally, both Linear & Polynomial and MLP & Perceptron pairs are within a 10% difference of one another.

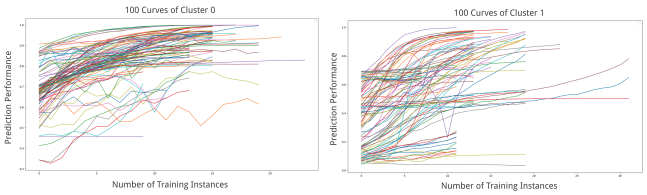
The average cluster assignment per dataset was also computed in an effort to discern any patterns related to the datasets within the clustering. If a classifier-dataset combination, often referred to as a single curve, is grouped within a cluster, the respective dataset is counted as a dataset occurrence.

Although majority of individual datasets were classified within Cluster 0, there was also an even distribution among the dataset percentages. There were 211 dataset occurrences in Cluster 0, and 172 in Cluster 1. In Cluster 0, 11 datasets were perfectly clustered, and 21 datasets had less than 5 instances of Cluster 0, indicating that majority fell within the range of 10-20 instances of Cluster 0. In Cluster 1, 20 datasets were perfectly clustered, and 62 datasets had less than 5 instances of Cluster 1.

The dataset statistics align with the trends observed in the learner statistics, such as a fairly uniform distribution. There are 100 datasets with fewer than 60% of learners in Cluster 0 and 108 datasets with over 60% of learners in Cluster 0. A similar pattern is found in Cluster 1. In addition to this, the difference in dataset occurrences of Cluster 0 and Cluster 1 is 39. Combining the findings from the learner analysis and the difference in dataset occurrences, it can be inferred that the two clusters are relatively similar, resulting in tipping ranges (50-60%).

In Figure 3, 100 random curves from Cluster 0 and Cluster 1 are plotted. The curves in Cluster 1 indicate a steady rise in performance, and eventually plateauing around 5-10 instances of training. On the other hand, several curves in Cluster 0 fluctuate heavily and begin with a higher average performance than Cluster 1. Despite the more non-monotonous

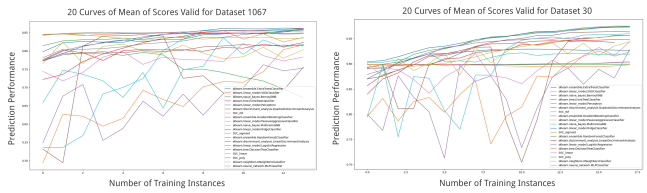
behavior observed in the curves of Cluster 0, there are still instances of similar curves between the two clusters. These matching curves typically initiate at around 0.6 prediction performance and slowly increase.



(a) (b)

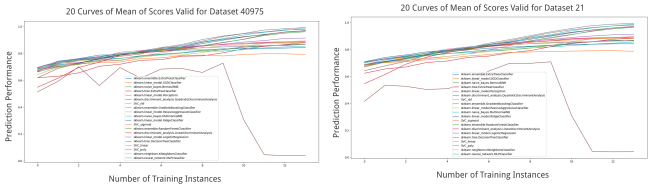
Figure 3: 100 random curves from Cluster 0 and Cluster 1, respectively, for point vector

The datasets found with 100% of learners in Cluster 0 show the most non-monotonous shapes. For example, dataset 1067 and 30, are shown in Figure 4. In contrast, the curves found with 95% of learners in Cluster 1 are quite flat, with a slow increase in performance across training instances. For instance, dataset 40975 and 21, are shown in Figure 5. Curves found with 25% of learners in both clusters showcase a more rapid and pronounced increase in performance. This suggests that the K-means algorithm created clusters based on monotonicity of the curves. Cluster 0 consisting of fluctuating curves, and Cluster 1 containing slowly progressing curves.



(a) (b)

Figure 4: Datasets 1067 and 30, respectively, that are perfectly classified in Cluster 0 for point vector, and demonstrate notably non-monotonous behaviour



(a) (b)

Figure 5: Datasets 40975 and 21, respectively, that are 95% classified in Cluster 1 for point vector, and show predominant linear shapes

PCA Plots

The output of K-Means algorithm for the input point vector dataset is indicated in Figure 6. The two clusters are denoted as Cluster 0 and Cluster 1, represented by blue and orange respectively. The cluster centroids are indicated by the red crosses. Out of the total of 4367 points, 2702 points were allocated to Cluster 0, and 1165 point were assigned to Cluster 1.

The clusters formed are widely distributed across Principal Component 1, indicating a fairly even distribution of data along the x-axis. Cluster 0 has a smaller area compared to Cluster 1 but includes 1537 more data points. This suggests a more concentrated distribution of points within Cluster 0. Within the range of [-0.5, -1] of Principal Component 2 on the graph, there is a line formed across both Cluster 0 and Cluster 1. This line along the y-axis of the Principal Component 2 suggests an alignment in variability in the data. Both clusters are not spread apart from another, converging in the middle of the graph. The cluster centroids are relatively close to each other, both falling within the [-1, 1] range of Principal Component 1. The close proximity of cluster centroids could support the observation that average cluster percentages fall within a tipping range (50-60%), favouring either Cluster 0 or Cluster 1. Therefore, despite the optimum being $K = 2$, according to the Silhouette Score, which implies the optimum formation of well-defined clusters, the clusters remain relatively similar and closely located.

However, due to the nature of PCA as a dimensionality reduction technique, there is a potential loss in information during the process. As a result, clusters may seem to overlap in a lower dimensional space while retaining distinct separations in the original dimensional space. Similarly, the way cluster centroids appear on the PCA plot may change in a higher dimensional space.

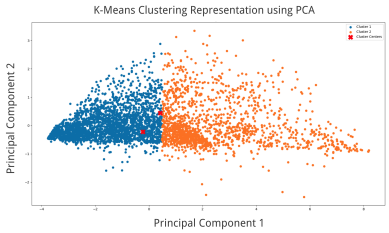
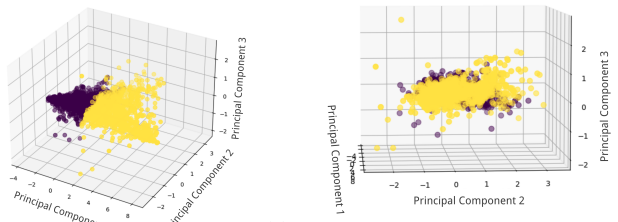


Figure 6: 2D PCA plot for point vector

For a more in-depth assessment of the cluster shapes, 3D PCA plots were generated. As seen in Figure 7b, the line identified in the 2D PCA plot is densely populated with points in the direction of the newly introduced Principal Component 3. Both Cluster 0 and Cluster 1 exhibit similar lengths along the third component. This may imply that using point vectors as input does not lead to distinctly separable clusters, particularly given the similarity of points and when considering scaling.



(a) (b)

Figure 7: 3D PCA plots for point vector from two different perspectives, indicating similar distribution in the newly introduced axis

5.3 Statistical Vector

Pre-processing of Data

In the initial phase of constructing the statistical feature vector, 88 precision errors occurred. These errors were found during the calculations of either skew or kurtosis, leading to the automatic assignment of NaN values. Since the K-Means algorithm cannot handle NaN values, it required manual intervention. The first approach was to replace the NaN value with a default float value within the custom range [-1, 1], initially set to 1.0. However, due to K-Means sensitivity to such changes, the second option was to exclude all vectors containing a NaN value. Given that only 88 out of the 4367 vectors had NaN values, the final K-Means clustering for the statistical vector was conducted without these points.

Percentages

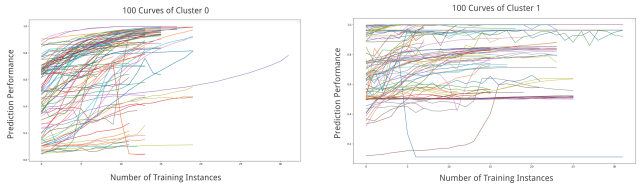
All learners are classified within Cluster 0, which is logical considering around 97% of all data points are within Cluster 0. Two learners, ExtraTreeClassifier (Tree), and MLPClassifier, are perfectly classified with 100% of the datasets grouped in Cluster 0. The remaining learners all fall within the 90-99% range, signifying an overwhelming majority for Cluster 0.

In the point vector analysis, ensemble learning technique learners showed similar behaviour. Extra Tree (Ensemble) and Gradient Boosting differed by 0.2%, which suggests another potential likeness. However, Random Forest is located at the 96% range. Although this is an incredibly small difference - relative to percentages seen in the point vector results - of around 2% from the other two learners, it is still not as close as Extra Tree (Ensemble) and Gradient Boosting.

Relating to the occurrences of NaN values, the learners with the most occurrences NaN values (Polynomial, BernoulliNB, RBF, and Sigmoid) happen to be the learners with the lowest percentages within the 92-94% range. It could be argued that if the precision error was eliminated and these values were included, that the range could be higher. Although there is no data to support that notion, it is interesting that those 4 learners are also the learners with the lowest average clustering percentages.

Within a difference of 178 dataset occurrences, there is overwhelming majority for Cluster 0. The distribution of datasets matches that of the learners. Having 70 of the datasets with 100% of learners in Cluster 0 and no datasets with less than 25% of learners in Cluster 0 makes it clear that there is a strong correlation among the points of Cluster 0. This could lead to the notion that Cluster 1 is compiled of outliers. There are no datasets with 100% of learners in Cluster 1. The overall distribution of datasets within Cluster 1 is bottom-heavy, suggesting no clear preference for this cluster by any dataset.

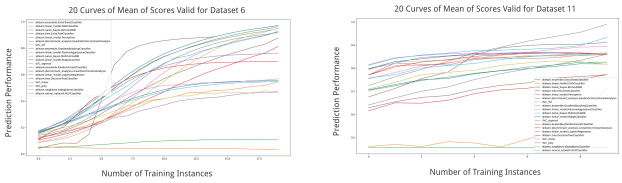
In Figure 8, 100 random curves from Cluster 0 and Cluster 1 are depicted. The curves in Cluster 0 are primarily concave, showing a gradual and steady performance increase followed by an extended flatted stretch. The curves in Cluster 1 are less concave, with majority being either flat or frequently fluctuating. There is minimal resemblance between the curves found in two clusters.



(a) (b)

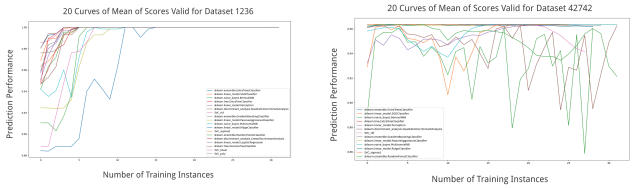
Figure 8: 100 random curves from Cluster 0 and Cluster 1, respectively, for statistical vector

The datasets with 100% of their learners in Cluster 0 show the most consistent concave shapes. For example, dataset 6 and 11 are shown in Figure 9. However, datasets such as 1236 and 42742 in Figure 10 within Cluster 1 show the highly fluctuating curves, accompanied by some instances of nearly horizontal curves.



(a) (b)

Figure 9: Datasets 6 and 11, respectively, that are perfectly classified in Cluster 0 for statistical vector, and present quite smooth curves



(a) (b)

Figure 10: Datasets 1236 and 42742, respectively, that are classified in Cluster 1 for statistical vector, revealing fairly irregular shapes when compared to Figure 9

As anticipated, the curves produced by both perfectly clustered learners in Cluster 0, Extra Tree (Tree) and MLP, reveals

that they consistently showcase the overall concave shapes recognised in Figure 8a, Figure 9a, and Figure 9b.

PCA Plots

The result of K-means algorithm for the input statistical vector dataset is indicated in Figure 11. With a total of 4279 points, 4169 points were assigned to Cluster 0 and 110 points were assigned to Cluster 1.

The 2D PCA plot indicates that most of the points grouped in Cluster 0 are densely concentrated within a linear strip, within the [-0.2, 1.0] range of Principal Component 2. This implies a strong correlation among all of the learner's statistical features, given that over 4000 points are situated in that small area in a similar direction.

In Cluster 1, there is a much more circular and evenly distributed arrangement of points, resembling the classic shape that K-Means aims to generate. However, the spread of points across both clusters is highly disproportionate as there are only 110 points in Cluster 1.

Considering most average cluster percentages are within the 90-100% range, no single learner is predominantly found in Cluster 1. This reiterates that all points in Cluster 1 may simply be outliers, and that majority of the statistical features of the curves are very similar to one another, making them indistinguishable through the K-means algorithm.

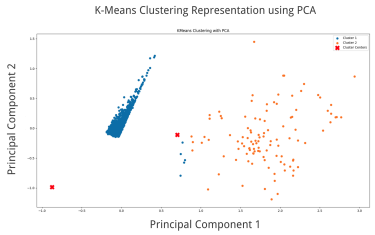


Figure 11: 2D PCA plot for statistical vector

Seeing the presence of a distinct, highly concentrated linear strip identified in the 2D PCA plot, a 3D PCA plot was generated to observe how the data is distributed along the z-axis direction. As illustrated in Figure 12, Cluster 0 forms a parabolic shape that does not stretch far across the third Principal Component, indicating once again a highly concentrated segment within all three dimensions. No additional information can be discerned that is not already evident in the 2D PCA plot.

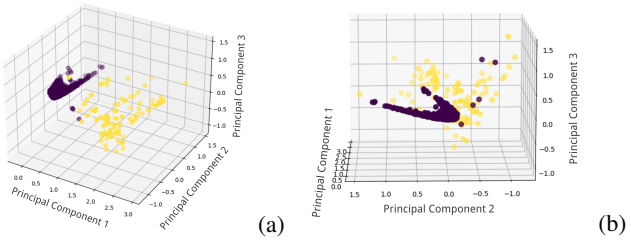


Figure 12: 3D PCA plots for statistical vector from two different perspectives, illustrating the dense population within Cluster 0

	Mean	Std	Skew	Kurtosis
Dataset	-0.835	-0.965	-0.067	-0.857
Cluster 0	-0.841	-0.971	-0.073	-0.876
Cluster 1	-0.774	-0.942	-0.095	-0.823

Table 1: Average of each statistical feature for the entire dataset, Cluster 0, and Cluster 1, respectively

Statistical Features

As shown in Table 1, the average of each statistical feature in the vector was computed. The first row represents the average for all the statistical vectors, allowing for comparison with the created clusters. Across all features, it is evident that Cluster 1 deviates the most from the overall dataset average. Particularly with the average mean element, Cluster 1 deviates by 0.061, whereas Cluster 0 deviates by 0.006. This is mainly attributed to the majority of points in the entire dataset being located within Cluster 0. Moreover, the relatively high deviation between the dataset averages and Cluster 1 reinforces the notion that only the significantly outlying curves are placed in Cluster 1.

Scaling led to alterations in the statistical feature averages. The average numbers without scaling reveal a comparable pattern to Table 1, with Cluster 0 showing a stronger deviation from the overall dataset average.

An additional experiment involved modifying the custom range from [-1,1] to [0,1]. While this adjustment shifted the results along the x-axis, it did not alter the fundamental patterns and shapes identified.

6 Conclusions

Conclusions is divided into potential future work and final findings derived from the Results & Discussion in Section 5.

6.1 Final Findings

Clear isolated patterns could not be determined after clustering the point vectors. An apparent relation among ensemble learning techniques surfaced in the learner analysis. With regards to outliers, the Quadratic Discriminant classifier was the sole learner favouring Cluster 1, while all other learners had a tipping average clustering percentage. The dataset analysis further confirmed the wide distribution and lack of correlation within both clusters. Ultimately, the 2D PCA graph visualised the clusters, showcasing the extensive dispersion of both groupings. The 3D PCA graph indicated a similar shape and distribution in the z-axis.

In the statistical vectors, two clearly defined non-overlapping clusters emerged. With a significant 97% of the vectors in Cluster 0, it was clear that Cluster 1 consisted of outlier curves that did not follow the typical concave shapes of standard learning curves. The highly concentrated Cluster 0 proved challenging to further analyse, as both the learner and dataset analysis indicated that all percentages were closely aligned, unlike the results for the point vector. More definitive patterns could not be uncovered through a secondary clustering analysis, performed exclusively on Cluster 0.

In conclusion, exact equivalence relations could not be established through K-means clustering.

6.2 Future Work

To further investigate potential patterns, there are several extensions that can be made within the existing scope of research.

In contradiction to Tarpey [6], the clusters formed within the PCA plots for the point vector seemed to overlap. However, as mentioned earlier, this could be a result of information loss during dimensionality reduction. Further research into the reasons behind K-means possibly failing to generate non-overlapping clusters in this particular instance could be explored.

For specifically the point vector analysis, the current interpolation does not consider anchor point of the curves. This may influence the clustering outcome, which is why this could be incorporated and compared with the original results to identify if there are any noteworthy changes. Another aspect regarding interpolation is that the shorter curves are lengthened to align with the maximum curve length within the LCDB. It would be intriguing to explore whether reversing this process, specifically shortening the longer curves, could result in a different clustering.

Focusing on the statistical vector analysis, the largest cluster consisted of over 4000 data points. To understand whether patterns can be determined within that group, the K-means algorithm was applied once again on itself. Nevertheless, this led to the formation of another densely populated cluster without any clear dominant patterns. Similarly, for the point vector, employing the second most optimal value of $K = 3$ yielded identical PCA plots regarding shape and distribution.

Concerning both statistical and point vectors, one could argue that K-means did not adequately take into account their distinctive features. As highlighted in Tarpey's 2012 paper [6], which investigates distance-based clustering, model-based clustering leverages valid parametric assumptions - a crucial aspect overlooked by K-means. A possible extension could be exploring the application of a model-based clustering algorithm on both point and statistical vectors.

In summary, the investigation of a model-based clustering algorithm could enhance the research by assessing parametric assumptions within the curves. It is also suggested that statistical and point vectors may not be the most suitable input data for identifying patterns in curve database.

A Appendix

A.1 Percentages of Clustering

Within Section 5 Results & Discussion, the statistical summaries are presented. Specific percentages for each learner are detailed in Figure 13 below.

Learner	Cluster	Percentage
Linear	0	68.1
Poly	0	64.9
RBF	0	65.7
Sigmoid	0	51.0
Decision Tree	0	63.8
Extra Tree (tree)	0	51.7
Logistic Regression	0	71.3
Passive Aggressive	0	60.4
Perceptron	0	56.0
Ridge Classifier	0	63.9
SGD Classifier	0	58.4
MLP Classifier	0	61.9
BernoulliNB	0	52.9
Multinomial	0	62.7
K-Neighbours	0	60.2
Extra Tree (ensemble)	0	75.8
Random Forest	0	75.2
Quadratic Discriminant	1	38.3
Linear Discriminant	0	54.6
Gradient Boosting	0	76.0

(a)

Learner	Cluster	Percentage
Linear	0	97.9
Poly	0	92.7
RBF	0	93.3
Sigmoid	0	93.8
Decision Tree	0	98.5
Extra Tree (tree)	0	100
Logistic Regression	0	95.4
Passive Aggressive	0	97.5
Perceptron	0	99.1
Ridge Classifier	0	98.0
SGD Classifier	0	98.7
MLP Classifier	0	100
BernoulliNB	0	94.5
Multinomial	0	97.7
K-Neighbours	0	99.5
Extra Tree (ensemble)	0	98.5
Random Forest	0	96.5
Quadratic Discriminant	0	99.5
Linear Discriminant	0	99.4
Gradient Boosting	0	98.3

(b)

Figure 13: Exact percentage of datasets in Cluster 0 per learner for point and statistical vector, respectively

A.2 Other Experiments

K=3 for Point Vector

When running the K-means algorithm with $K = 3$, three clusters were formed, aligning perfectly with the shape and distribution of the original two clusters. Although there is an extra segmentation, the overall shape remained unchanged. This meant that neither of the optimal values $K = 2$ and $K = 3$ resulted in clearly separated clusters. Figure 14 shows the PCA plot produced.

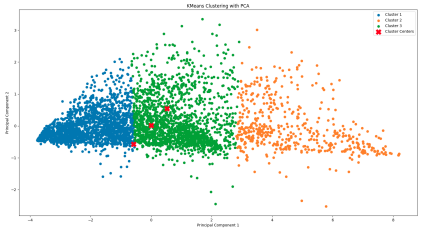


Figure 14: 2D PCA plot for point vector using $K = 3$

Statistical Vector Clustering

Since a densely populated region was observed within the statistical vector clustering, an additional K-means clustering was performed exclusively on that particular cluster. Nevertheless, no significant results were obtained, as depicted in Figure 15.

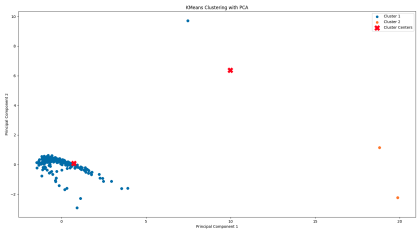


Figure 15: 2D PCA plot for statistical vector on only Cluster 0 using $K = 2$

Range of Scaling

An extra PCA graph was produced to show the impact of changing the scales from [-1,1] to [0,1] in the statistical vector clustering. Figure 16 displays the change in PCA graph.

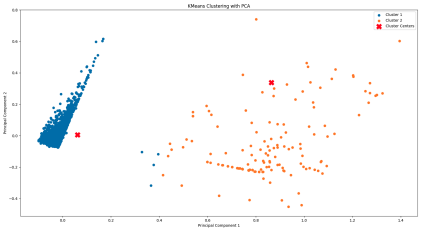


Figure 16: 2D PCA plot for statistical vector using [0,1] custom range instead of [-1,1]

A.3 Large Language Model

Reference: OpenAI. (2024). ChatGPT (Jan 2024 version) [Large Language Model]. https://chat.openai.com/chat. Prompt: "Rephrase: ... [insert sentence]".

References

[1] Viering, T. (2023). *"How Much Data is Enough?" Learning Curves for Machine Learning.* Project Forum. https://projectforum.tudelft.nl/course_editions/74/generic_projects/4899

[2] Mohr, Felix and Viering, Tom J and Loog, Marco and van Rijn, Jan N. (2022). *LCDB 1.0: An Extensive Learning Curves Database for Classification Tasks*. Machine Learning and Knowledge Discovery in Databases. https://github.com/fmohr/lcdb/blob/main/README.md

[3] Meek, C. (2002). *The Learning-Curve Sampling Method Applied to Model-Based Clustering.* Journal of Machine Learning Research 2 (page 397-418). https://www.jmlr.org/papers/volume2/meek02a/meek02a.pdf

[4] Navarro, D. & Lee, M. (2005). *An Application of Minimum Description Length Clustering to Partitioning Learning Curves.* https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1523403

[5] Moraes, E. (2019). *Data clustering based on principal curves*. Advances in Data Analysis and Classification (2020).

[6] Tarpey, T. (2012). *Linear Transformations and the k-Means Clustering Algorithm: Applications to Clustering Curves.* https://www.tandfonline.com/doi/epdf/10.1198/000313007X171016?needAccess=true

[7] Sharma, P. (2023). *The Ultimate Guide to K-Means Clustering: Definition, Methods and Applications.* Analytics Vidhya. https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/#:~:text=The%20k%2Dmeans20algorithm%20uses,and%20their%20assigned%20cluster%20centroid.

[8] J, E. (2023). *K-Means Clustering: 7 Pros and Cons uncovered.* Datarundown. https://datarundown.com/k-means-clustering-pros-cons/

[9] Henrique, A. (2022). *Stop using the Elbow Method - Geek Culture - Medium.* Medium. https://medium.com/geekculture/stop-using-the-elbow-method-96bcfbbbe9fd

[10] Educative. (2024). *What is Silhouette Score?*. Educative Answers - trusted answers to developer questions. https://www.educative.io/answers/what-is-silhouette-score

[11] Wohlwend, B. (2023). *Decision Tree, Random Forest, and XGBoost: An Exploration into the Heart of Machine Learning.* Medium. https://medium.com/@brandon93.w/decision-tree-random-forest-and-xgboost-an-exploration-into-the-heart-of-machine-learning-90dc212f4948