A Comparative Analysis of Learning Curve Models and their Applicability in Different Scenarios

Finding datasets patterns which lead to certain parametric curve model

Anna Kalandadze

Supervisors: Dr. Jesse Krijthe, Dr. Tom Viering

[1]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
June 25, 2023

Name of the student: Anna Kalandadze
Final project course: CSE3000 Research Project
Thesis committee: Dr. Jesse Krijthe, Dr. Tom Viering, Dr. Zhengjun Yue

Abstract

Learning curves display predictions of the chosen model's performance for different training set sizes. They can help estimate the amount of data required to achieve a minimal error rate, thus aiding in reducing the cost of data collection. However, our understanding and knowledge of the various shapes of learning curves and their applicability are still insufficient. Despite the presence of a curve that demonstrates a high level of accuracy on average, this parametric model can still exhibit inadequate performance in certain scenarios. Therefore, the objective of this research is to identify specific patterns in the datasets that influence the selection of a particular parametric curve model. To accomplish this, I conduct experiments to assess the performance of different parametric learning curves including *power*, *exponential* and *Morgan-Mercer-Flodin* (mmf) based on the number of features, classes, outliers, and machine learning models. I find that mmf and exponential curves outperform power law for all machine learning models. All curves work best with Logistic Regression, Bernoulli Naive Bayers and Multinomial Naive Bayers models. Exponential and mmf curves provide better results than power law for a small number of classes. Mmf also outperforms power law for the majority of numbers of features and outlier percentages.

1 Introduction

In the area of supervised machine learning, it is challenging to get enough data to optimize the performance of the learning model. A learning curve serves as a valuable graphical tool that allows us to gain insight into the learning behavior by illustrating the relationship between generalisation performance and the number of training samples [1] [2]. A typical example of a learning curve can be seen in Figure 1. This figure represents the error rate on the y-axis versus the training data size on the x-axis for the machine learning algorithm called Gradient Boosting Classifier.

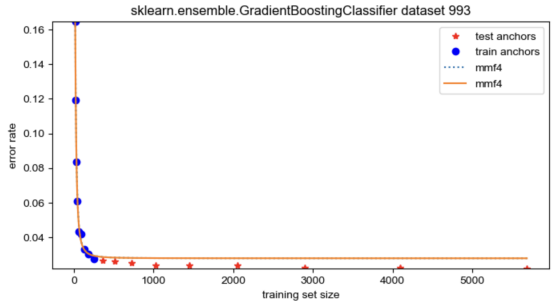

Figure 1: Example of a learning curve for Gradient Boosting Classifier

Learning curves provide an approximation of how much data is needed to achieve an expected error rate. As data collection is a costly process, using a minimal amount of data is often desired. With the help of learning curves, the optimal amount of data can be found, and thus we can reduce the cost of data collection [3]. We also often want to find accurate and fast machine learning algorithms, and learning curves provide insight into the performance of different algorithms [4]. Thus, learning curves lead to efficient solutions and effective data usage.

Currently, many shapes of the learning curves have been found and experimented with [2], leading to one of the parametric curves outperforming the others on average. However, average best performance does not lead to optimal performance for many datasets. This means that some datasets have poor performance using averagely the most accurate learning curve. Therefore, a deeper analysis is necessary to determine more precise signals for using a specific parametric curve model.

The research question can be formulated as:

Which learning curve model provides the best fit in what case?

To answer the research question, I use fitting results of learning curves from the learning curve database (LCDB) [2]. An analysis should be conducted on whether there are characteristics of datasets such as a number of dimensions or learning models which lead to choosing the same parametric curve model.

There are multiple procedures available to identify whether the learning curve performs well. In this research, I will analyse mean squared error (MSE) and mean absolute error (MAE/L1) using LCDB datasets. I assume that the lower MSE or MAE are, the better the learning curve fits. Then, using chosen metrics, an analysis will be conducted on whether datasets with certain characteristics require the same learning curve. If such patterns are identified, I will analyse them.

This paper consists of multiple chapters. Firstly, I will relate to the existing literature in Chapter 2, and explain current limitations of existing work. Secondly, I will introduce the methodology of this research in Chapter 3. Then, in Chapter 4 experimental setup will be explained. The results of the experiments will be discussed in Chapter 5, followed by a discussion of the results in Chapter 6 and a conclusion in Chapter 7. Lastly, I will argue about the reproducibility and responsibility of this research in Chapter 8.

2 Related literature

Multiple other kinds of research were made to find the shape of the curve that leads to the best predictions. For example, in [5], Frey and Fisher conclude that power law fits best for decision trees. The same is concluded by Gu et al. [6] using large datasets. Research by Mark Last [7] also concludes that power law provides the best fit, although mentions that for one of the training cases, exponential shape gives more precise predictions. In [8] exponential curve provides better predictions on average. At the same time, Singh [9] states that in certain cases logarithmic learning curves outperform

power law.

Examined literature provides a lot of proof that different parametric learning curve models should be used. However, there is no clear indication of which curve to use given datasets characteristics. Another limitation of [2] is that fitting was measured using mean squared error (MSE), which can be not representative for some datasets [10]. According to [10], R-squared, which is used in [5], may be more suitable. However, it is not valid for non-linear learning curves [11] as R-squared assumes that total variance of data equals the sum of error variance and variance explained by the model. That is not the case for non-linear data. Spiess & Neumeyer [12] point out a poor performance of R-squared in their experiment for non-linear data.

In [13], an analysis of MSE and MAE is also made. However, this analysis is limited, as the research considers only classification and regression factors to choose whether to use MSE or MAE. It also evaluates the most accurate parametric curve model on average for classification and regression and does not consider the characteristics of the datasets.

Even with the existence of mentioned literature, many works contradict each other by stating that different parametric models work best. Thus, there is no consensus yet reached. Also, all mentioned work concentrates on finding overall average patterns and does not analyse specific dataset characteristics which may influence behaviour of learning curves. In this research, I will provide an analysis of the patterns that lead to a certain parametric curve model, instead of looking at average performance. I will also analyse which types of accuracy measures should be used, as current related work does not provide reasoning for choosing one measure over the other.

3 Methodology

This research paper examines whether there are patterns to the best fit for a learning curve. This means that datasets with certain characteristics, such as the number of features, classes, the percentage of outliers, or the use of a machine learning model, may lead to the choice of an identical learning curve parametric model. To identify such patterns, I collected fitting results from LCDB and determined which metric to use to measure performance. Then, using this metric, I determined if there were patterns when a certain learning curve provided the best performance.

3.1 Fitting results collection

Firstly, I collected fitting results from LCDB. Since LCDB contains 246 datasets, 16 learning curves models, and 20 machine learning models, I decided to proceed with the existing data instead of gathering new fitting results.

The data I collected includes the characteristics of the datasets, such as the number of features and the number of classes. Additionally, I required fitting results from each experiment. I collected the results for each dataset and included the performance of each parametric model. To preprocess the data, I grouped collected results by dataset, learning curve, and machine learning model. LCDB provided results for multiple anchor points used for training and testing to indicate the accuracy of the machine learning model when trained on a specific sample size. I averaged both the MSE and MAE of each learning curve for all numbers of training anchors. I also extracted the fields that I will use in the future experiment setup, including prediction on the learning curve and real value of the data point.

3.2 Determining which metric to use to measure performance

There were two metrics available directly from fitting results, which I decided to analyse. To do this, I examined both formulas:

$$MSE = \sum_{i=1}^{D} (x_i - y_i)^2 \tag{1}$$

$$MAE = \sum_{i=1}^{D} |x_i - y_i| \tag{2}$$

where:
D - number of observations
x_i - actual value of observation
y_i - predicted value of observation

From the formulas, I saw that both metrics avoid cancellation of negative errors by squaring in 1 and taking absolute value in 2. The biggest difference is that MSE penalizes large errors disproportionately and more severely than smaller errors, as it takes squared difference and sums the result, while MAE does not square difference. Thus, if we do not want to penalize outliers, we can use the MAE metric.

While depending on the situation, outliers can be penalized or tolerated, I analysed possible explanations for outliers' appearance.

3.3 Analysis on outlier appearance

To explain possible causes of a high number of outliers, I stated a hypothesis:

If there are a lot of outliers in the features of the dataset, then there are a lot of outliers in predictions of the learning curve.

To test this hypothesis, I first calculated the percentage of outliers in features. To do this, I used the interquartile range rule (IQR) [14]. This method calculates the range between the first quartile (Q1) and the third quartile (Q3) of the data. Then it assumes that every point of data that is less than $Q1 - (1.5 \cdot range)$ or more than $Q3 + (1.5 \cdot range)$ is an outlier. Some features were categorical, so I could not directly find outliers. For them, I used one-hot encoding to represent categorical data as numerical. One-hot encoding represents categorical variables as binary vectors and each unique category is represented by a separate binary column. A value of 1 in a column indicates the presence of that category, while 0 indicates its absence. To better understand how it works, Popov [15] provides a detailed example of a one-hot encoding technique. According to [16], one-hot encoding is a leading technique to deal with categorical data due to its consistent and accurate performance.

Then, I calculated the percentage of outliers in predictions using the IQR method. I found a correlation between two

Table 1: Chosen parametric curve models for experiments

Model Name	Formula
exp4	$c - exp(-ax^d + b)$
pow4	$a - b(d + x)^{-c}$
mmf4	$(ab + cx^d)/(b + x^d)$

a, b, c, d - are hyperparameters of the curve

groups of outliers. As I wanted to test that increase in feature outliers leads to an increase in the prediction outliers, I used the linear Pearson correlation coefficient. It measures the strength of the linear relationship between two variables by calculating their standard deviations and covariance [17].

3.4 Find patterns for best learning curve model

For all the characteristics a similar procedure was applied. I chose four characteristics: machine learning algorithm, number of features, classes, and outliers. Mohr & van Rijn [4] state that a learning curve can help choose an algorithm to use. However, in case of the limited time available, we may want to know which learning curve works best for a specific algorithm to evaluate it. Huang & Guan [18] mention that classification in a large number of classes can be challenging. It may require more time and data than datasets with a few number of classes. Thus, I would like to find out if there is a difference in the performance of multiple learning curves for a number of classes. In [19], Bui introduces a hypothesis that dimensionality may influence the shape of a learning curve. Although the research was inconclusive, I decided to check if certain dimensionality leads to the high performance of specific parametric models. Lastly, as I already analysed outliers, I wanted to check if a certain parametric model is robust to feature outliers or, vice versa, produces specifically bad results.

I decided to explore in detail three learning curves: *Morgan-Mercer-Flodin* (mmf4), *power* (pow4), and *exponential* (exp4). The formulas can be found in the Table 1. I chose specifically these curves, as mmf4 produces the best results on average [2], while power and exponential curves often appear in related literature.

I started by analysing learning curves individually based on one of the chosen characteristics. To do this, I calculated the median MSE/MAE per each characteristic type. I chose to calculate median over a mean as I could not assume normal distribution. Visual comparison of precision measures would not show whether the difference was indeed significant, so I used statistical tests to compare medians. As I dealt with multiple groups, I needed to use a test suitable for multiple groups. Such tests show if there is a significant difference between groups. To determine which groups differ, I proceeded with a post-hoc test pairwise. If the test showed significant difference, I compared medians of the error measures, and determined when a learning curve works best.

I also wanted to compare curves mentioned in Table 1. For this, I plotted the medians of the curves for each characteristic and conducted statistical tests to determine significant differences. That showed me which curve out of three behaves best depending on a characteristic.

4 Experimental Setup

In this section, I will explain the setup of experiments that are used for achieving tasks discussed in the methodology.

4.1 Experiment 1: Correlation between number of outliers in features and predictions

To understand whether a high number of outliers in features of datasets leads to many outliers in learning curve predictions, I used a pre-calculated percentage of feature outliers. I calculated a number of prediction outliers by grouping LCDB data by dataset id and combining all differences between actual and predicted values in one array. On that array, I ran the IQR method to detect outliers. As a distribution of differences had long tails, I increased a constant from 1.5 to 10 and considered a data point an outlier if it laid outside $[Q1 - (10 \cdot IQR); Q3 + (10 \cdot IQR)]$ range. Then, I ran Pearson correlation from *pandas* in Python. It assessed whether large percentage of feature outliers leads to large percentage of prediction outliers. As Pearson correlation assesses only linear relationships, I also created a plot of feature outliers percentages and prediction outliers percentages. Using it, I assessed whether there is a visible non-linear correlation between the two variables.

4.2 General setup for finding patterns that lead to the best performance of certain parametric learning curve model

To experiment based on a characteristic, I grouped processed fitting results by that characteristic and collected MSE and MAE metrics. The data I used included infinity MSE values for some of the observations. To perform chosen statistical test, I needed the same number of observations, so I could not delete a row. Instead, I replaced the infinity value with 100000, which was appropriate, as such MSE is considered very high. To achieve precise results, I also eliminated outlier MSE/MAE observations and instead imputed values using linear interpolation from *pandas*. As I dealt with a large number of observations, such interpolation worked fast and provided smooth and continuous estimates between data points. Then, depending on a characteristic, different statistical analysis was made which will be described in the next paragraphs.

4.3 Experiment 2: Analysis of curves based on machine learning model

While the datasets used for all models were identical, I needed to use a paired statistical test to find out if there is a significant difference. I had a group of observations, and I could not assume a normal distribution, hence I used the Friedman test. This is a non-parametric statistical test that is used on a group of data of 3 or more measurements to determine if there is a significant difference across a group. It is a paired test, which means there is a dependent variable: the same dataset. The test assigns ranks to each of the observations, and calculates the test-statistic and p-value. Detailed examples of using the Friedman test can be found in [20]. If the p-value is less than chosen significance level α (0.05),

then we found a significant difference within a group of observations. If the p-value is greater, then no difference was found. For this experiment, I used existing Friedman test implementation in Python from *spicy.stats*.

Then, if there was a difference, I needed to compare observations pairwise to find which groups differ. For this I used Wilcoxon signed rank test. It calculates the differences between the pairs of observations of compared groups and assigns a rank to each of the differences. Then it calculates the test statistic and p-value [21]. If the p-value is less than the significance level, I conclude that the pair is significantly different. I used an existing method from *scipy.stats*. As the test makes multiple comparisons within the group, I corrected the significance level using Holm's method. It adjusts α to minimize the probability of falsely identifying significant differences. To apply Holm's correction, I used *statsmodels.stats.multitest.multipletests*.

If a pairwise significant difference was found, I assigned ranks to each of the models. The pairwise test did not give me a result of which algorithm worked best, so I collected all MSE/MAE medians and compared pairs that had p-value $<$ 0.05. If the median of one model was less than the median of another, I added an edge from the worse performer to the better one and vice versa. This way I created a directed graph. Lastly, I assigned the ranks based on the number of outgoing edges: models with the least number of outgoing edges received the best rank of 1.

This allowed me to conduct individual analysis of learning curves to understand whether they work better with certain machine learning algorithms.

Then, I also compared 3 chosen learning curves to each other depending on a model. I used the same setup as described above, but this time I compared not models but 3 curves based on the model type. This gave me a result of which of the three learning curves works best for which learner.

4.4 Experiment 3: Analysis of curves based on the number of features

As the number of samples presented for each number of features differed, I divided the number of features using buckets. I created 8 buckets of length 10, 9 buckets of length 200, and 1 final bucket for datasets from 1080 to 100001 features. Examining the number of features without buckets would be inefficient since there were a lot of unique values, as well as it is challenging to interpret the difference between individual numbers of features. Instead, I created groups to see how learning curves behave on datasets with a small number of features versus very large dimensionality.

Since each observation had different datasets, I used the independent Kruskal-Willis test. It is a non-parametric test, thus it can be used on skewed data. It is used to find whether there is a significant difference for multiple groups. It sorts observations of all groups and assigns ranks to them. Then, it adds up the received ranks per group and calculates the test-statistic. From this statistic, it calculates the p-value. For a detailed explanation of this test, you can refer to [22]. I used Kruskal-Wallis test from *scipy.stats* library.

If the p-value was less than 0.05, I set up post-hoc Dunn's test with Holm's alpha correction. I chose this test as it is non-parametric. It is also a rank-based test, so per each group it calculates a mean rank. Then test-statistic and p-value are calculated. In the end, it produces a list of pairs that show a significant difference. A detailed example of Dunn's test can be seen in [23]. I used already existing implementation of Dunn's test from *scikit_posthocs*.

After doing this, I received an indication of individual curve performance. Then, I compared the three curves with each other using the same buckets of features. I followed the procedure discussed in 4.3 to determine whether one of the curves performs better than others for a small or large number of features.

4.5 Experiment 4: Analysis of curves based on the number of classes

To set up an experiment, I first needed to analyse the distribution of a number of classes for the datasets I had. From LCDB datasets, I had only 21 unique values for the number of classes, so I started to assess them individually. I repeated a procedure discussed in 4.4: I used Kruskal-Wallis and Dunn's test for individual comparison. Then, I did a group comparison using Friedman and Wilcoxon signed-rank tests.

However, the distribution of classes was not equal, for example, I had 145 datasets with 2 classes and only 1 dataset with 100 classes. Thus, I decided to create buckets to assess a general pattern: the behaviour of the learning curves on the datasets with relatively small and large number of classes. I created the following buckets: [0, 5), [5, 10), [10, 20), [20, 30), [30, 50), [100, 355]. I compared buckets using the same setup from 4.4.

4.6 Experiment 5: Analysis of curves based on the percentage of outliers

As each dataset had a different number of outliers, assessing them individually would not be an option. To get an impression of a general pattern, I first plotted all three curves with their performance on the graph. However, some of the curves were unpredictable in behaviour and I could not assess them properly. Thus, I divided percentages of outliers into buckets: [0, 0.5), [0.5, 1), [1, 3.5), [3.5, 6), [6, 18.61]. I processed them the same way as in 4.4 both individually and groupwise. This allowed me to say whether one curve behaves better on datasets with a small percentage of outliers versus a large percentage of outliers.

5 Experimental results

This section describes results for each of the experiments conducted.

5.1 Experiment 1: Correlation between number of outliers in features and in predictions

The examination was done using the percentages of outliers in features and predictions. Pearson correlation coefficient was $r = 0.1011639$, which indicated a weak linear relationship. Using the coefficient, I could not state that high percentages of outliers in features led to outliers in predictions.

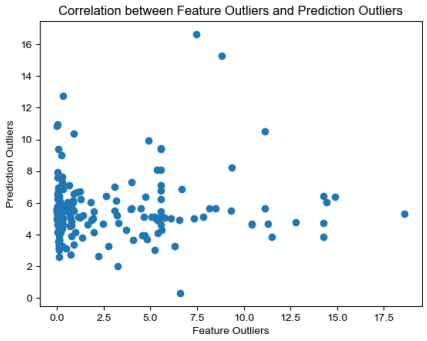

Figure 2: Correlation between outliers in features and outliers in predictions

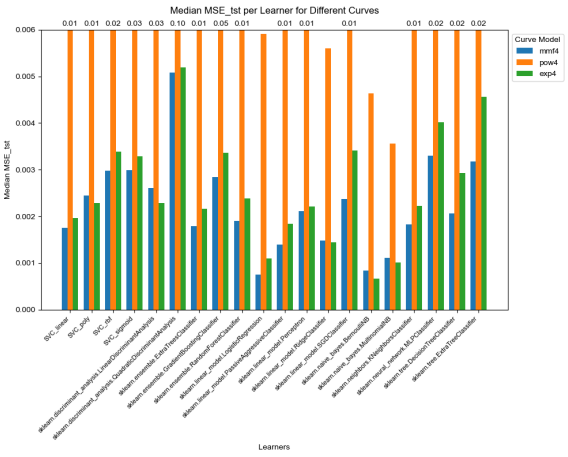

Figure 3: Comparison of mmf4, pow4 and exp4 learning curves using MSE based on machine learning models

Figure 2 also does not show any visible correlation between two variables, so I rejected a hypothesis that outliers in features lead to prediction outliers.

5.2 Experiment 2: Analysis of learning curves based on machine learning model

The analysis started with representation of medians of MSE for all three curves in Figure 3. It allowed me to see average performance of learning curves, assess individual accuracy as well as compare learning curves to each other. Similar graph for MAE can be found in Figure 7 in Appendix A.

Individual assessment

Friedman test concluded that for every learning curve there's significant difference between learners. Table 2 represents assigned ranks to every learner for a particular curve. The smaller number of rank, the better result this learner produces using MSE or MAE.

MMF4 Results

The mmf4 parametric curve shows the best results using Logistic Regression, Bernoulli Naive Bayes and Multinomial Naive Bayes algorithms. The worst results are reported using Quadratic Discriminant Analysis. MSE and MAE are generally consistent, and produce similar ranking with a fluctuation

Table 2: Ranking of learners for mmf4 using MSE and MAE (smaller rank corresponds to better performance)

	mmf4		pow4		exp4	
Learner	Rank MSE	Rank MAE	Rank MSE	Rank MAE	Rank MSE	Rank MAE
Logistic Regression	1	1	1	1	1	1
Bernoulli Naive Bayes	1	1	1	1	1	1
Multinomial Naive Bayes	1	1	1	1	1	1
Passive Aggressive Classifier	1	2	2	2	2	2
Ridge Classifier	1	2	1	1	2	2
Linear Support Vector Classification (SVC)	1	3	1	1	2	2
Polynomial SVC	1	3	2	2	2	2
Extra Trees Classifier	2	2	1	1	3	2
Perceptron	2	3	2	2	3	3
Linear Discriminant Analysis	2	4	2	5	2	3
KNeighbors Classifier	3	2	2	2	3	2
Random Forest Classifier	3	2	2	1	3	3
Decision Tree Classifier	3	3	2	2	3	5
Stochastic Gradient Descent Classifier	4	3	2	2	4	6
Radial basis function SVC	4	3	2	2	3	5
Sigmoid SVC	4	3	4	4	3	3
Gradient Boosting Classifier	4	3	3	4	3	4
Extra Tree Classifier	5	5	3	3	5	7
Multi-layer Perceptron Classifier	5	6	2	3	3	7
Quadratic Discriminant Analysis	5	7	5	6	6	8

in Linear Discriminant Analysis.

POW4 Results

The pow4 parametric curve works best on Logistic Regression, Bernoulli Naive Bayes, Multinomial Naive Bayes, Ridge Classifier, Linear Support Vector Classification and Extra Trees Classifier algorithms. The worst accuracy appears using Quadratic Discriminant Analysis. Findings of MSE and MAE are similar, except of Linear Discriminant Analysis, where MSE assigns rank 2 to the algorithm, while MAE lowers it to 5.

EXP4 Results

The learning curve exp4 shows the best precision using Logistic Regression, Bernoulli Naive Bayes and Multinomial Naive Bayes, while Quadratic Discriminant Analysis produces the worst results. MAE and MSE results slightly differ, especially using Multi-layer Perceptron Classifier.

Group assessment

Friedman test confirmed that there is a difference between parametric models for every learner. Then I proceeded with pairwise comparison and interpreted it using Figure 3. I summarised the comparison in Table 3.

Exponential and Morgan-Mercer-Flodin shapes outperform power law in all the scenarios. So, even though pow4 ranks some of the learners, such as Extra Trees Classifier, higher than exp4 and mmf4, it shows poorest results compared to other two learning curves. Figure 3 also confirms that power law produces poor results for all the learners. In most cases, mmf4 produces better on equally precise results as exp4 with an exception of Linear Discriminant Analysis using MAE, and Ridge Classifier using MSE.

5.3 Experiment 3: Analysis of learning curves based on the number of features

Figure 4 represents MAE medians for each bucket of features. A similar graph for MSE can be found in Figure 8 in Appendix A.

Table 3: Pairwise comparisons of learning curves using MSE and MAE based on machine learning model. X denotes that no significant difference was found between parametric models

	MSE			MAE		
Learner	mmf4 vs pow4	mmf4 vs exp4	exp4 vs pow4	mmf4 vs pow4	mmf4 vs exp4	exp4 vs pow4
Linear SVC	mmf4	x	exp4	mmf4	x	exp4
Polynomial SVC	mmf4	x	exp4	mmf4	mmf4	exp4
Radial basis function SVC	mmf4	mmf4	exp4	mmf4	mmf4	exp4
Sigmoid SVC	mmf4	x	exp4	mmf4	mmf4	exp4
Linear Discriminant Analysis	mmf4	x	exp4	mmf4	exp4	exp4
Quadratic Discriminant Analysis	mmf4	mmf4	exp4	mmf4	mmf4	exp4
Extra Trees Classifier	mmf4	mmf4	exp4	mmf4	mmf4	exp4
Gradient Boosting Classifier	mmf4	mmf4	exp4	mmf4	mmf4	exp4
Random Forest Classifier	mmf4	mmf4	exp4	mmf4	mmf4	exp4
Logistic Regression	mmf4	mmf4	exp4	mmf4	mmf4	exp4
Passive Aggressive Classifier	mmf4	mmf4	exp4	mmf4	mmf4	exp4
Perceptron	mmf4	x	exp4	mmf4	x	exp4
Ridge Classifier	mmf4	exp4	exp4	mmf4	mmf4	exp4
Stochastic Gradient Descent Classifier	mmf4	mmf4	exp4	mmf4	mmf4	exp4
Bernoulli Naive Bayes	mmf4	x	exp4	mmf4	mmf4	exp4
Multinomial Naive Bayes	mmf4	x	exp4	mmf4	mmf4	exp4
KNeighbors Classifier	mmf4	mmf4	exp4	mmf4	mmf4	exp4
Multi-layer Perceptron Classifier	mmf4	mmf4	exp4	mmf4	mmf4	exp4
Decision Tree Classifier	mmf4	mmf4	exp4	mmf4	mmf4	exp4
Extra Tree Classifier	mmf4	mmf4	exp4	mmf4	mmf4	exp4

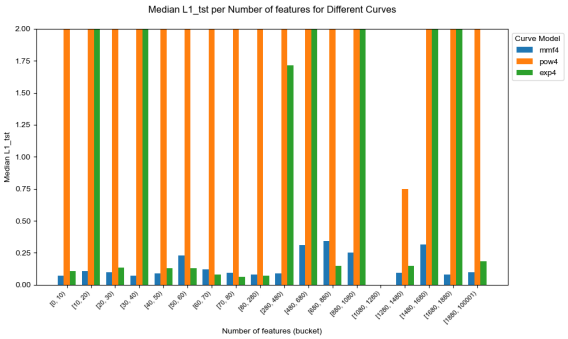

Figure 4: Comparison of mmf4, pow4 and exp4 learning curves using MAE based on the grouped number of features

5.4 Individual assessment

Even though visually it seems that there is a significant difference, Kruskal-Wallis test considers different sample sizes, and visual interpretation may not coincide with exact significance. For mmf4, p-values of 0.5494489 and 0.2746788 were obtained using MSE and MAE respectively. Power law reported p-value 0.0937335 for MSE and 0.0908338 for MAE. Lastly, p-value of exp4 was 0.4038479 for MSE and 0.4948854 for MAE. All p-values are greater than 0.05, thus the test did not reveal any difference individually.

5.5 Group assessment

Using Friedman test, learning curves were compared by bucket. I obtained the following results in Table 4. The table shows that Morgan-Mercer-Flodin always outperforms power law. On small number of dimensions $n < 20$ Morgan-Mercer-Flodin also outperforms exponential curve. Exponential curve produces better results than power law on small dimensions, and on buckets [40, 60) for both MAE and MSE. Visual difference can be seen in Figure 4 for buckets [280, 480), [480, 680), [680, 880), [880, 1880), but likely due to the small number of samples, the test does not prove statistical difference.

Table 4: Pairwise comparisons for MSE and MAE based on number of features. X denotes that no significant difference was found between parametric models

	MSE			MAE		
Bucket (Features)	mmf4 vs pow4	mmf4 vs exp4	exp4 vs pow4	mmf4 vs pow4	mmf4 vs exp4	exp4 vs pow4
[0, 10)	mmf4	mmf4	exp4	mmf4	mmf4	exp4
[10, 20)	mmf4	mmf4	exp4	mmf4	mmf4	exp4
[20, 30)	mmf4	x	x	mmf4	x	x
[30, 40)	mmf4	mmf4	x	mmf4	x	x
[40, 50)	mmf4	x	exp4	mmf4	x	exp4
[50, 60)	mmf4	x	exp4	mmf4	x	exp4
[60, 70)	mmf4	x	exp4	mmf4	x	x
[80, 280)	mmf4	x	x	mmf4	x	x
[1880, 100001)	mmf4	x	x	mmf4	x	x

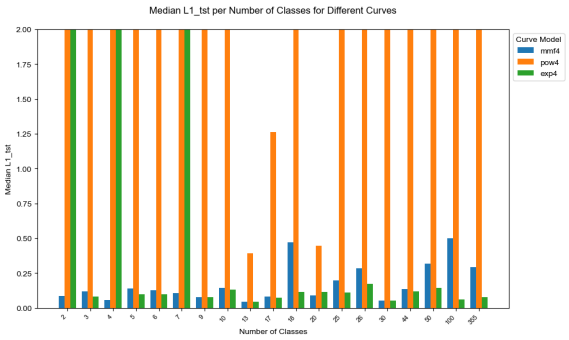

Figure 5: Comparison of mmf4, pow4 and exp4 learning curves using MAE based on the grouped number of classes

5.6 Experiment 4: Analysis of learning curves based on the number of classes

As there were not a lot of different classes, I assessed all of them in Figure 5. Similar graph for MSE can be found in Figure 9 in Appendix A.

Individual assessment

Kruskal-Wallis tests produced p-value of 0.6490089 and 0.4222196 for mmf4 using MSE and MAE respectively. For pow4, the result was 0.2531694 for MSE and 0.2423912 for MAE. Lastly, exp4 produced p-value of 0.5277016 using MSE and 0.9187818 using MAE. Thus, individually, I did not find motivation to use a certain parametric model for specific number of classes.

Group assessment

Treating each class individually, I obtained the following comparisons in Table 5. For the classes with reported significant difference, Morgan-Mercer-Flodin always outperforms power law. When grouping number of classes in Table 5, Wilcoxon test reported that exponential curve outperforms power law for number of classes < 10. Buckets [20, 30), [30, 50), [100, 355] did not report any significant difference. Mmf also outperforms exponential curve for number of classes < 5.

Table 5: Pairwise comparisons of learning curves using MSE and MAE based on the number of classes. X denotes that no significant difference was found between parametric models

	MSE			MAE		
Number of classes	mmf4 vs pow4	mmf4 vs exp4	exp4 vs pow4	mmf4 vs pow4	mmf4 vs exp4	exp4 vs pow4
2	mmf4	mmf4	exp4	mmf4	mmf4	exp4
3	mmf4	x	x	mmf4	x	x
7	mmf4	x	x	mmf4	x	x
10	mmf4	x	x	mmf4	x	x
4, 5, 6, 9, 13, 17, 18, 20, 25, 26, 28, 30, 44, 50, 100, 102, 355	x	x	x	x	x	x
Bucket of number of classes						
[0, 5)	mmf4	mmf4	exp4	mmf4	mmf4	exp4
[5, 10)	mmf4	x	exp4	mmf4	x	exp4
[10, 20)	mmf4	x	x	mmf4	x	x

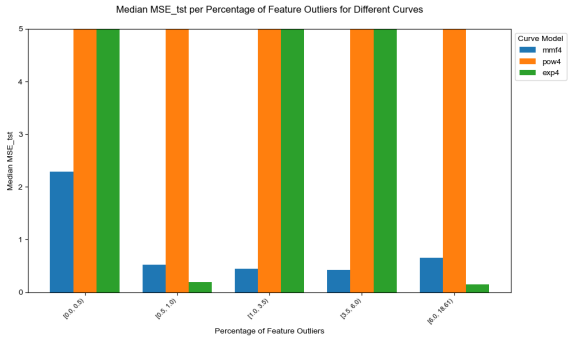

Figure 6: Comparison of mmf4, pow4 and exp4 learning curves using MSE based on the grouped percentage of outliers

5.7 Number of outliers

No general visual pattern was found for all the learning curves depending on the number of outliers. The corresponding graphs can be found in Figures 10 and 11 in Appendix B.

After grouping percentage of outliers in buckets, visual difference can be seen in Figure 6. Similar Figure for MAE can be found in Figure 12 in Appendix B.

Individual assessment

Wilcoxon test produced the following results, when I assessed a curve based on a bucket: mmf4 had a p-value of 0.2085195 and 0.5852067 for MSE and MAE respectively, so no significant difference was found individually for mmf4. Power curve reported p-value 0.0009052 for MSE and 0.0008275 for MAE, meaning that there is a difference. Lastly, exp4 produced a p-value of 0.0966195 using MSE and 0.0426948 using MAE. Thus, exp4 reported statistical difference using MAE. After doing post-hoc Dunn's test on median value per bucket, I achieved significant difference before alpha correction. However, as alpha correction is obligatory for multiple group comparison, I could not rely on those findings. All found p-values with alpha correction were greater than 0.05, and thus no direct indication on learning curve performance per bucket of outliers could be made.

Group assessment

Comparing learning curves per bucket, I obtained results in Table 6. Morgan-Mercer-Flodin outperforms power law in all situations, and exponential curve also outperforms power law with small percentage of outliers < 1 or very big percentage > 6. If we mesure precision using MSE, exponential outperforms mmf4 for small percentage of outliers, while for MAE the result is the opposite.

Table 6: Pairwise comparisons of learning curves using MSE and MAE based on the number of outliers. X denotes that no significant difference was found between parametric models

	MSE			MAE		
Bucket (outlier percentage)	mmf4 vs pow4	mmf4 vs exp4	exp4 vs pow4	mmf4 vs pow4	mmf4 vs exp4	exp4 vs pow4
[0, 0.5)	mmf4	exp4	exp4	mmf4	mmf4	exp4
[0.5, 1)	mmf4	x	exp4	mmf4	x	exp4
[1, 3.5)	mmf4	x	x	mmf4	x	x
[3.5, 6)	mmf4	x	x	mmf4	mmf4	x
[6, 18.61)	mmf4	x	exp4	mmf4	x	exp4

6 Discussion and Limitations

In this section, I will analyse the results I got and discuss the limitations of this work.

6.1 General findings

The results section shows that the Morgan-Mercer-Flodin learning curve outperforms power law in all experimented scenarios where a significant difference was found, and often outperforms exponential curve. It confirms that mmf4 works generally best as stated in [2].

Exponential curve also outperforms power law for all machine learning algorithms, which coincides with [8]. Examined learning curves ranked algorithms similarly. This is expected since some machine learning algorithms work more accurately and faster than others, and all learning curves react to it by showing smaller MSE/MAE. We also saw that MSE and MAE assign rank similarly, which means that many datasets do not have a lot of large outliers. In some situations, MSE gives an algorithm a better rank than MAE, which means that MSE penalizes other datasets which have outliers. When MSE gives an algorithm a worse rank than MAE, it can mean that dataset has many outliers, which MAE tolerates.

For the number of features and classes, I did not find any difference in the individual behaviour of the learning curves. However, it doesn't mean that there is no difference, since the results in Figures 4 and 5 show that performance differs, especially for exponential curve. Statistical tests find no difference as there is a limited number of datasets available per bucket or class number. It also reflects a comparison between learning curves: the test can compare all 3 curves based on dimensionality < 20 and number of classes < 5, as the majority of datasets have those characteristics. For a lot of features and classes, no group difference was found, which can be also explained by the lack of observations.

When considering the percentage of feature outliers, I found out that for a very small percentage, exponential shape outperforms mmf4 using MSE, while mmf4 provides better results considering MAE. In this scenario, exponential shape may produce less large outliers than mmf4.

Kruskal test also showed a significant difference in the number of outliers, while Dunn's test did not. Dunn's test

makes multiple comparisons and adjusts p-values, which minimizes the risk of false positives. P-values become bigger after adjustment, and no statistical difference is shown.

I also did not confirm a hypothesis that feature outliers lead to prediction outliers. This means that when fitting learning curve, it does not see feature outliers as specific values and tries to fit outliers accurately. Thus, no prediction outliers are created.

Lastly, as I consider the performance of the learning curve based on MSE/MAE, I also need to consider that bigger error characteristics may be due to noise. To check it, a visual investigation was made for the learning curves. An example can be seen in Figures 13, 14, 15 in Appendix C. In some examples, I saw that mmf4 provides the best fit, followed by exp4 and pow4. However, it was not always the case. It can be explained by the fact that I averaged MSE and MAE for all the numbers of anchors, and they showed average behaviour for all training/test set sizes.

6.2 Limitations and Future Work

The research provides insight into when to use which learning curves. However, no precise patterns are yet found. I used LCDB data for this research, and while it contained 246 datasets, most of them had 2 classes and a small number of dimensions. That is why comparative analysis could not be fully completed. Even though I did not find precise patterns for large dimensionality or a large number of classes, further analysis should be conducted by comparing more datasets with those characteristics.

I also used averaged fitting results for all numbers of anchors. However, there can be an optimal number of training anchors that leads to the best accuracy, and then the optimal performance of the learning curve may be different from the average one. In the future, further analysis may be conducted about an optimal number of training anchors, and results should be also analysed using fitting data for optimal training-test set size proportion.

Lastly, in this research, I inspect all the characteristics individually. Even though it provides an analysis per characteristic, in the future, characteristics can be combined and analysed together. Then, additional more precise patterns may be found.

7 Conclusion

To summarise, this report aims to find insights into the patterns where specific learning curve works best. First, the examined literature showed that there is no consensus according to each learning curve that works best. Then I analysed the ways to measure the accuracy of the learning curve: MSE and MAE. I found out that MSE penalizes outliers, while MAE tolerates them. After this, I analysed outliers' behaviour and concluded that there is no evidence that outliers in features lead to prediction outliers.

I examined the behaviour of three learning curves depending on the machine learning model, number of features, classes and outliers. I made both individual learning curve assessments as well as compared learning curves to each other. The result was that mmf4 and exp4 outperform pow4 for all machine learning models. All curves rank models similarly, meaning that they all work best with almost identical models. For all dimensions where significant difference was found, mmf4 outperformed pow4. There were no dimensions for which exp4 worked better than mmf4. For a lot of classes, no difference was found, likely due to data absence. For the small number of classes < 10, mmf4 outperformed pow4. For the number of classes < 5 mmf4 also produced better results than exp4, while exp4 outperformed pow4. Lastly, for all the percentages of feature outliers, mmf4 outperformed pow4.

Overall, this work confirms that mmf4 works best on average. However, no precise concrete patterns in the learning curve comparison analysis were found. This can be explained by insufficient amount of data, especially for a number of classes and dimensions. I also used averaged MSE/MAE between all anchors, which may not be as representative of finding an optimal combination of training/test set sizes. Deep future research is still needed to address these limitations, as well as combine multiple characteristics to uncover more specific patterns.

8 Responsible Research

In the section, I will elaborate to what extent this work is reproducible and discuss possible risks.

Firstly, the data that I used comes from an open platform OpenML. The data there is stored and existing data should not updated. However, this is still an unlikely risk: OpenML could delete a dataset or it can be adjusted. Still, according to today's policies of the website, stored datasets would be left unchanged, meaning that there would not be added or deleted data to them.

Secondly, the code that I used is stored on GitHub and is open for viewing using the following link: GitHub Learning Curves. While it may be updated, GitHub supports a feature of viewing state of the repository on certain time, thus the code that I used for this report can be viewed and used. All the code that I wrote additionally is also stored on GitHub , and can be viewed and cloned freely.

Furthermore, all methods that I used are implemented in the built libraries in Python. This way, I decrease the probability of making a mistake in the code, and make the code more readable and easy to replicate. However, there is also a risk that library's developers discover bugs, and will change their implementation. At the same time, the libraries that I use are documented, and each update to them have descriptions. Thus in case of reproducibility problems, documentation of libraries can be addressed. All the methods and names of the algorithms are also described in this paper.

Lastly, all the setup and experiments are conducted using MacBook Pro (14-inch, 2023). This paper is also available in the TU Delft education repository, which is open and free.

As for the other ethical considerations, this work is concerned about Machine Learning optimisation. This can be used maliciously, for example to develop deepfake technology [24]. In Machine Learning field such risks are one the primary focuses of current researches.

References

[1] Tom Viering and Marco Loog. The shape of learning curves: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2022.

[2] Felix Mohr, Tom J. Viering, Marco Loog, and Jan N. van Rijn. Lcdb 1.0: An extensive learning curves database for classification tasks. In Massih-Reza Amini, Stéphane Canu, Asja Fischer, Tias Guns, Petra Kralj Novak, and Grigorios Tsoumakas, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 3–19, Cham, 2023. Springer Nature Switzerland.

[3] Felix Mohr and Jan N. van Rijn. Learning curves for decision making in supervised machine learning – a survey, 2022.

[4] Felix Mohr and Jan N. van Rijn. Towards model selection using learning curve cross-validation. In *8th ICML Workshop on Automated Machine Learning (AutoML)*, 2021.

[5] Lewis J. Frey and Douglas H. Fisher. Modeling decision tree performance with the power law. In David Heckerman and Joe Whittaker, editors, *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, volume R2 of *Proceedings of Machine Learning Research*. PMLR, 03–06 Jan 1999. Reissued by PMLR on 20 August 2020.

[6] Baohua Gu, Feifang Hu, and Huan Liu. Modelling classification performance for large data sets. In X. Sean Wang, Ge Yu, and Hongjun Lu, editors, *Advances in Web-Age Information Management*, pages 317–328, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[7] Mark Last. Predicting and optimizing classifier utility with the power law. In *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, pages 219–224, 2007.

[8] Botjan Brumen, Ivan Rozman, Marjan Heriko, Ales Cernezel, and Marko Hölbl. Best-fit learning curve model for the c4.5 algorithm. *Informatica*, 25:385–399, 2014.

[9] Sameer Singh. Modeling performance of different classification methods : Deviation from the power law. 2005.

[10] Davide Chicco, Matthijs J Warrens, and Giuseppe Jurman. The coefficient of determination r-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Comput Sci*, 7:e623, July 2021.

[11] Felix "Mohr, Tom J. Viering, Marco Loog, and Jan N." van Rijn. Supplementary material lcdb 1.0: An extensive learning curves database for classification tasks. 2023.

[12] Andrej-Nikolai Spiess and Natalie Neumeyer. An evaluation of r2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: A monte carlo approach. *BMC pharmacology*, 10:6, 06 2010.

[13] Dean Nguyen. In search of best learning curve model. 2022.

[14] H. P. Vinutha, B. Poornima, and B. M. Sagar. Detection of outliers using interquartile range technique from intrusion dataset. In Suresh Chandra Satapathy, Joao Manuel R.S. Tavares, Vikrant Bhateja, and J. R. Mohanty, editors, *Information and Decision Sciences*, pages 511–518, Singapore, 2018. Springer Singapore.

[15] Anton Popov. 1 - feature engineering methods. In Kunal Pal, Samit Ari, Arindam Bit, and Saugat Bhattacharyya, editors, *Advanced Methods in Biomedical Signal Processing and Analysis*, pages 1–29. Academic Press, 2023.

[16] Jia-Chen Zhao. Enhancement encoding: A novel imbalanced classification approach via encoding the training labels, 2023.

[17] Vijay Kotu and Bala Deshpande. Chapter 4 - classification. In Vijay Kotu and Bala Deshpande, editors, *Data Science (Second Edition)*, pages 65–163. Morgan Kaufmann, second edition edition, 2019.

[18] Yizhen Huang and Yepeng Guan. On the linear discriminant analysis for large number of classes. *Engineering Applications of Artificial Intelligence*, 43:15–26, 2015.

[19] NAM THANG Bui. Factors related to dataset that influence the shape of learning curves. 2022.

[20] Michael R. Sheldon, Michael J. Fillyaw, and W. Douglas Thompson. The use and interpretation of the friedman test in the analysis of ordinal-scale data in repeated measures designs. *Physiotherapy Research International*, 1(4):221–228, 1996.

[21] Denise Rey and Markus Neuhäuser. *Wilcoxon-Signed-Rank Test*, pages 1658–1659. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[22] Patrick E. McKight and Julius Najab. *Kruskal-Wallis Test*, pages 1–1. John Wiley & Sons, Ltd, 2010.

[23] Alexis Dinno. Nonparametric pairwise multiple comparisons in independent groups using dunn's test. *Stata Journal*, 15:292–300, 04 2015.

[24] Mika Westerlund. The emergence of deepfake technology: A review. *Technology Innovation Management Review*, 9:40–53, 11/2019 2019.

A Representation of median performance of the learning curves on machine learning models, number of features, classes, and outliers

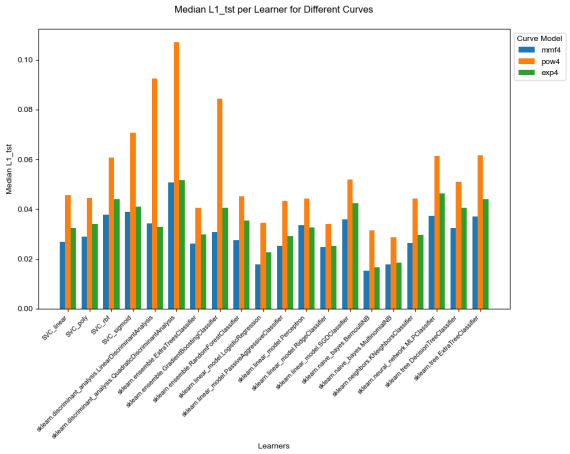

Figure 7: Comparison of mmf4, pow4 and exp4 learning curves using MAE based on machine learning models

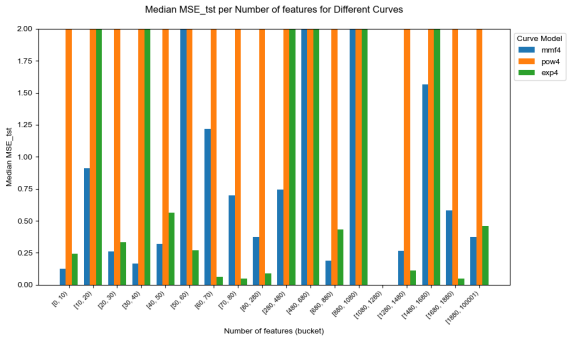

Figure 8: Comparison of mmf4, pow4 and exp4 learning curves using MSE based on number of features (divided in buckets)

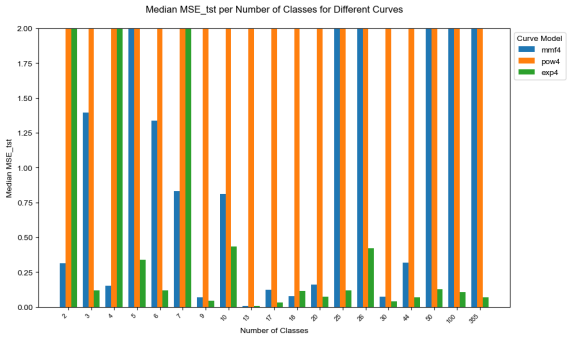

Figure 9: Comparison of mmf4, pow4 and exp4 learning curves using MSE based on number of classes

B Representation of median performance of the learning curves on the number of outliers

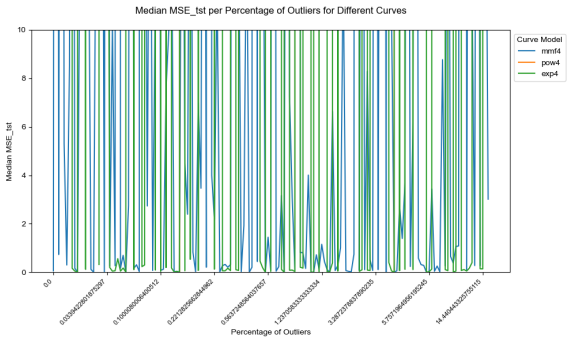

Figure 10: Behaviour of learning curves depending on the percentage of outliers using MSE

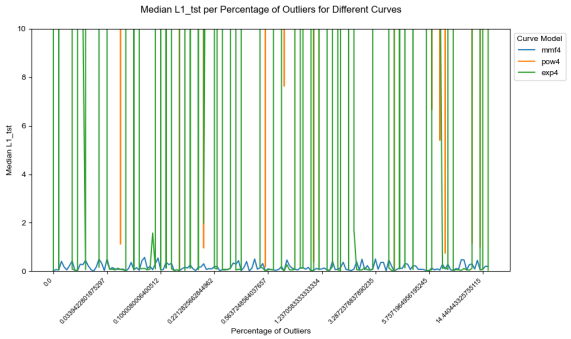

Figure 11: Behaviour of learning curves depending on the percentage of outliers using MAE

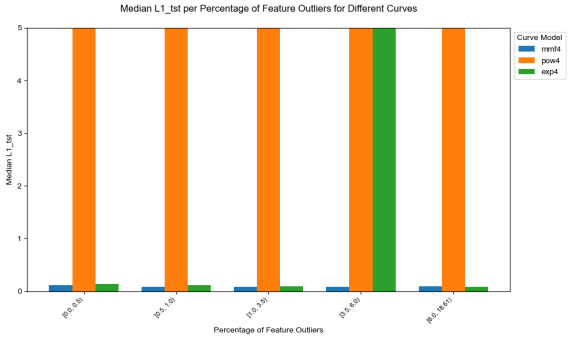

Figure 12: Comparison of mmf4, pow4 and exp4 learning curves using MAE based on the percentage of outliers

C Representation of mmf4, pow4 and exp4 for training and test anchors

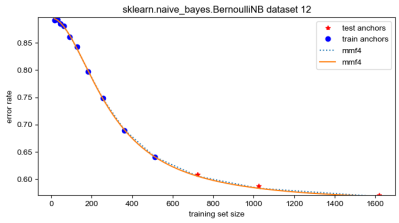

Figure 13: Learning curve mmf4 for dataset 12 using Bernoulli Naive Bayes

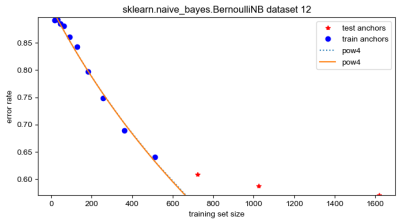

Figure 14: Learning curve pow4 for dataset 12 using Bernoulli Naive Bayes

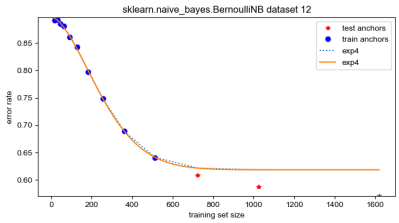

Figure 15: Learning curve exp4 for dataset 12 using Bernoulli Naive Bayes