Empirical Investigation of Learning Curves
Assessing Convexity Characteristics

Kristián Gogora[1]

Supervisor(s): Jesse Krijthe[1]**, Tom Viering**[1]

[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: Kristián Gogora
Final project course: CSE3000 Research Project
Thesis committee: Jesse Krijthe, Tom Viering, Zhengjun Yue

Abstract

Nonconvexity in learning curves is almost always undesirable. A machine learning model with a nonconvex learning curve either requires a larger quantity of data to observe progress in its accuracy or experiences an exponential decrease of accuracy at low sample sizes, with no improvement in accuracy even when more data points are added. This paper proposes a novel approach to determine the convexity of a learning curve, which relies on calculating the second derivative of the learning curve to estimate its convexity. Along the way, we have confirmed the correctness of the proposed method from multiple perspectives, such as testing it with baselines or establishing confidence intervals for the convexity of the learning curve. Lastly, we compare our method to an alternative method and highlight some of its shortcomings.

1 Introduction

A learning curve is a function that gives us insight into the impact of the machine learning model's error rate with respect to the number of training points used to train our model [1]. One possible application of learning curves is determining the number of training data points our machine learning model should have. In this field of research, we usually call the data points on the learning curve anchors to differentiate them from the data used to train a classifier.

Generally speaking, most literature researching this field assumes that learning curves are universally convex and decreasing unit cost functions [2]. We think that one of the main driving factors for this assumption is the Convex Gaussian Minimax Theorem [3]. The Convex Gaussian Minimax Theorem is a statistical tool, and one of its many use cases is estimating the true mean of a Gaussian distribution [4], which could potentially show us the true shape of the learning curves. This motivates us to ask the questions *"How many learning curves are nonconvex?"* and *How can we estimate nonconvexity?*. To help answer this question, we first address, *"When is a discrete function nonconvex?"*. Next, *"How does the convexity of the individual learning functions affect the averaged learning function?"* and *How certain are we in the convexity of the learning curve?* might also explain why convex learning curves were assumed as standards. Finally, with the question *Are high convexity violation learning curves nonconvex?* we will compare this paper's proposed method of checking convexity to the convexity violation implemented inside of the Learning Curve Database (LCDB) [5]. In this paper, we will address the issue of convexity violation using an alternative approach while also determining the overall convexity of the learning curve.

The paper will adopt the following structural organization. In Section 2 some related works will be analyzed. Section 3 will introduce an algorithm for calculating the second derivative and define theoretical concepts that will be later used in the paper. Subsequently, Section 4 will propose methods for evaluating convexity on a discrete function. The principal outcomes of the research question will be presented in Section 5. Next, Section 6 will report on the experimental results derived from Section 5. Section 7 will offer conclusions, suggest a few approaches for future improvements, and identify potential gaps in the research. Lastly, in Section 8, ethical considerations surrounding the study will be discussed.

2 Related Work

There has been minimal research conducted on verifying the convexity of learning curves. One example can be seen in [5], however, the method used is relatively crude. It relies on two adjacent anchors to determine the convexity of the middle anchor. The main shortcoming of this technique is sensitivity to inaccuracies due to noise in the anchor points and that the method assumes that the neighbouring anchor points are an equal distance from the middle one, which is not the case. Lastly, the proposed measure only describes the largest convexity violation, which is correct locally but not on the entire learning curve. There is another useful tool introduced in [5], which is the LCDB, which allows us access to learning curves with different dataset-classifier combinations. A different work that also explores the convexity of the learning curve can be found in [6], which tries to "repair" the convexity of the learning curve.

3 Methodology

The following section will define some tools needed to construct the algorithm for estimating the Second Derivative of the learning curve, after which we will introduce the algorithm itself. Next, we will propose a metric to evaluate the convexity of the learning curve. Finally, we will define the confidence interval of an anchor's convexity.

Convex function

Before exploring the algorithm first, we have to define what a convex function is. A function is convex if its Second Derivative is positive on its entire domain, however, the function's Second Derivative must exist.

$$\forall x \in \mathcal{D}, f''(x) \geq 0 \tag{1}$$

We can see this mathematical definition in Equation 1, where f is at least a twice differentiable function and $\mathcal{D}$ is the domain of the function. This definition holds for continuous functions, but in our case, we are working with discrete functions as a learning curve is constructed by many individual anchors. We extend the definition to be able to handle discrete functions by approximating differentiation by computing the finite difference at the desired points. In this new definition f becomes the discrete function of the learning curve and x will be considered as the sample size to which we calculate the error rate of the function.

Linear Regression

The next tool we need to construct our algorithm is Linear Regression. Linear regression is a statistical method used to find the best-fit line that represents the relationship between two or more variables [7]. The line of best-fit is the line which minimizes its squared difference with respect to the data points.

Algorithm for Estimating the Second Derivative

The Algorithm's main goal is to approximate the Second Derivative at every anchor point. We use a similar approach from [8], where the finite difference at a given point was approximated by using Linear Regression on the nearby data points. Additionally, we have borrowed the data loader and dataset of the learning curves from LCDB [5], also available on GitHub [1]. Now we present a high level description of the algorithm:

1. Takes as an input every anchor point of the learning curve
2. For each anchor point with sample size s_i, collects anchor points with sample size s_i, s_{i-1} and s_{i+1}, the set creation can be also described with the following equation: $A_i = \{(s, y) \in f | s \in s_{i-1}, s_i, s_{i+1}\}$
3. On each set of collected anchors perform Linear Regression. This way we can obtain the equation of the best-fit line: $\beta_0 + x\beta_1 \Leftarrow LinearRegression(A_i)$
4. Save the slopes of the fitted lines: $slope_i = \beta_1$
5. Repeat steps 1-4, but with the previously computed slopes as input

Steps 1-4 are used to calculate the finite difference of the input. Using these steps twice differentiates the function twice thus approximating its Second Derivative. We calculate the sample size in the same manner as it is calculated in LCDB [5]: $s_n = 16\sqrt{2}^n$. In step 2 it is important to note that there might be multiple anchor points with the same sample size.

Evaluating convexity

Now that we have created an algorithm for estimating the Second Derivative of the learning curve the only task we have left out is evaluating if a learning curve is convex or nonconvex. We already know the answer partially, based on our definition of convexity we only need to check if all calculated Second Derivatives are above zero. However, evaluating learning curves may not be this simple, since the learning curves are only approximated functions we must also take noise into consideration. Noise may cause fluctuation in value thus flipping the signs of some of the Second Derivative values, in which case it is not trivial to determine the convexity of the learning curve we run into the same issue if the learning curve would also contain both convex and nonconvex regions. Therefore we created a measure to calculate how convex the function is in its domain. The measure checks whether there are more convex than nonconvex anchors in the domain of the learning curve.

$$S_c = \{x \in f'' | s \geq 0\} \tag{2}$$

$$S_{nc} = \{x \in f'' | s < 0\} \tag{3}$$

$$\mathbf{M} = \frac{|S_c| - |S_{nc}|}{|f''|} \tag{4}$$

[1]https://github.com/fmohr/lcdb/

In Equations 2 and 3 f'' refers to a discrete set of second derivatives meaning that the set in Equation 2 only contains convex point and the set in Equation 3 only nonconvex. Lastly, by the Equation 4 my measure is defined as the difference between the number of convex and nonconvex anchors. If $\mathbf{M} \geq 0$ we define the learning curve as dominantly convex, otherwise we say it is dominantly nonconvex. The value of the measure ranges within the interval [-1, 1]. This range holds, since if all anchors are convex then $|S_c| = |f''|$, resulting in $\mathbf{M} = 1$, similar reasoning applies if all the anchors are nonconvex, where if $|S_{nc}| = |f''|$ holds, then $\mathbf{M} = -1$.

Confidence intervals

$$y = \beta_0 + \beta_1 x \tag{5}$$

$$\beta_1 = \beta_1 \pm SE(\beta_1) t_{n-2,\alpha/2} \tag{6}$$

$$SE(\beta_1) = \frac{\sigma}{\sum (x_i - \overline{x})^2} \tag{7}$$

We establish a simple linear regression with two dimensions in which case we can define a regression line as shown in Equation 5, where we are mostly interested in the value of β_1 corresponding to the slope of the line while ignoring the value of β_0. We assume that the anchor points at a given sample size have normal distribution around a mean [9]. Next, we can establish the confidence interval of the slope according to [10], which is demonstrated in Equation 6, where t is a value from t-distribution, and SE is the standard error, what we further define in Equation 7.

4 Experimental Setup

The experimental setup for evaluating the convexity of learning curves must be done systematically to assess the behavioural patterns of the learning curves correctly. In this section, the algorithm proposed in the Methodology section will be examined first. Next, a comparison between an individual and a whole set of learning curves of the same dataset and classifier will be evaluated. After this, the confidence interval of each anchor point per learning curve will be calculated. Lastly, a general evaluation of the convexity violation measure will be made for the LCDB dataset of learning curves.

Experiment 1: Is the Second Derivative Estimating Algorithm working correctly?

Firstly, it is crucial to make sure that the devised algorithm performs what it was designed for. To test this algorithm we will have to also combine it with the measure that will be used to evaluate learning curves. The experiment will include a test for a convex and a nonconvex function. In this case, a quadratic function, $f(x) = x^2$, will be chosen to assess the correctness of the algorithm. Afterwards, we will try out a more complex learning curve, we will create a decreasing exponential with a local maximum, $f(x) = e^{-0.05x} + 0.5e^{-\frac{(x-128)^2}{16}}$, which will serve as a more practical test. All of the above-mentioned baseline learning curves will be generated in the following way:

1. Take samples of the function with inputs corresponding to s_n according to Equation **??** up until $n = 20$,

which corresponds to a dataset with the size of 16384 data points.

2. Add random noise with 5% of the difference between the largest and smallest anchor as standard deviation.
3. Map the values between 0 and 1.
4. Run the algorithm on the modified values
5. Repeat steps 1-4 one hundred times and average the result

Experiment 2: How many learning curves in the LCDB dataset are nonconvex?

In this experiment, the learning curves provided by the LCDB dataset will be evaluated. The dataset itself has 20 different classifiers and 248 datasets. Overall, 4366 different classification-dataset combinations will be used to rank the learning curves, with each learning curve having 25 anchor points per sample size. It is important to note that some anchors of the learning curve are NaN valued, to deal with this we have decided to ignore such anchors. Next, the convexity measure will be computed for all the existing combinations and ranked in decreasing order to view the most and least convex learning curves. Lastly, the percentage of learning curves with a negative measure will be calculated.

Experiment 3: Are there any differences in convexity, if we only take individual learning curves instead of measuring all of them at once?

The next experiment will provide insight into the difference between taking a single learning curve from a sample of 25 learning curves. The convexity measure will be determined for each individual learning curve, and the overall convexity will be decided on a majority vote. We will propose that a violation occurs if the convexity of the learning curve changes compared to Experiment 2.

Experiment 4: After calculating the confidence interval, how many anchors exhibit a mismatch in convexity measure?

Setting up this experiment became more challenging due to the way the Algorithm Determining Convexity was constructed, after using the first linear regression most of the variance is reduced to zero on each sample size since the first linear regression creates only one anchor point for every sample size. To eliminate this issue and retain the variance, Experiment 3's method was used to calculate the first derivative. It is acceptable to use this method because individual learning curves are not interacting between themselves, thus retaining the noise that each of them carries. After this, the second derivative is calculated with Experiment 2's method, additionally while calculating the slope from a set of anchors the confidence interval is also calculated here. A $\alpha = 0.95$ is used to calculate the critical value. Finally, it is checked if the anchor's slope and one of the confidence interval values have different signs, which we count as a violation in the convexity interval. More precisely a violation happens if the slope is greater than zero, but the lower bound of the confidence interval is less than zero, and vice versa.

Experiment 5: How well does the proposed metric work compare to the one proposed in the LCDB?

In this experiment, learning curves with the top 10 highest convexity violations according to the metric proposed in [5] are compared to this paper's measure. It is important to note that the ranking in LCDB does not explicitly say that a high convexity violation should automatically correspond to a nonconvex curve. These selected learning curves' convexity will be eyeballed and then compared to this paper's measure.

5 Results

In the Result section we will present the outcomes of the experiments we defined in the previous section.

Experiment 1: Is the Second Derivative Estimating Algorithm working correctly?

We have evaluated the Second Derivative Estimator Algorithm's performance with baseline functions 100 times. In Table 1 we summarize our result. The Second Derivative Estimator Algorithm estimated the convex baseline with an average score of 0.404 and the worst score was 0.1, meaning that the algorithm correctly assumes that the baseline is convex in all 100 runs. Testing the baseline for a nonconvex curve also produced similar results. More specifically the average score was -0.388, while its worst score was -0.1. The next curve we explored shown in Figure 3, was a decreasing exponential with a hill, we considered this curve to be convex since the nonconvex part represented only a small portion of the whole learning curve, where the results of the baseline were 0.327 average score and 0.1 worst score.

Baseline	Average case	Worst case	Best case
Convex	0.404	0.1	0.8
Nonconvex	-0.388	-0.1	-0.8
Convex with hill	0.327	0.1	0.6

Table 1: Baseline testing results of the Algorithm

Experiment 2: How many learning curves in the LCDB dataset are nonconvex?

Among the 4366 learning curves, 8.86% were identified as nonconvex. In Figure 4 we can see that the most learning curves are produced by the *sigmoid SVC* classifier with 28.34% of its learning curves being nonconvex, while the least nonconvex learning curves were created by the *Linear Discriminant Analysis* learner with 1.16%. Figure 5 shows us some dataset ids (OpenmlId) which produced the largest quantity of nonconvex learning curves.

Experiment 3: Are there any differences in convexity, if we only take individual learning curves instead of measuring all of them at once?

From Table 2, we can observe that when a learning curve is nonconvex according to Experiment 2 in 81.40% of the cases indicate that if we investigate the individual learning curves the majority of them turn out to be convex, while on the other hand if Experiment 2 predicts a convex learning curve only in 0.53% of the cases the individual learning curves would be nonconvex in the majority.

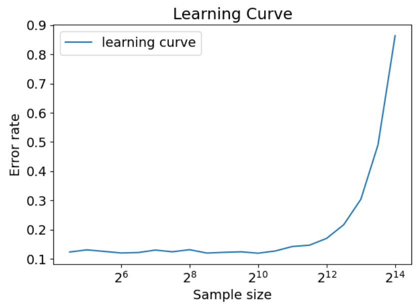

Figure 1: Example learning curve, for the convex baseline with an average score of 0.404

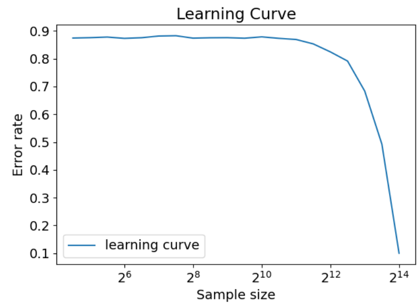

Figure 2: Example learning curve, for the nonconvex baseline with an average score of -0.388

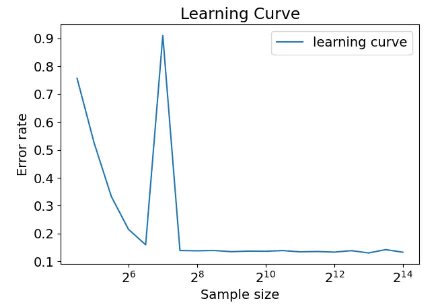

Figure 3: Convex baseline with nonconvex violation with an average score of 0.327

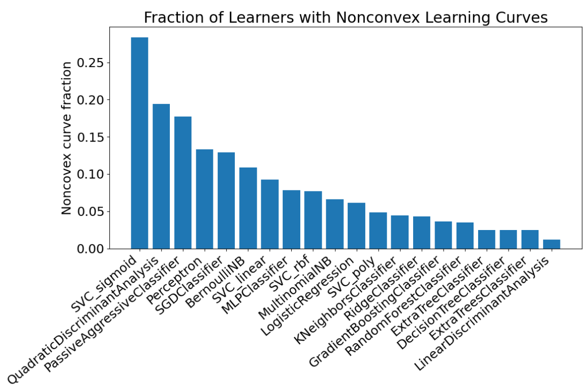

Figure 4: Learners and the fraction of their learning curves that are nonconvex

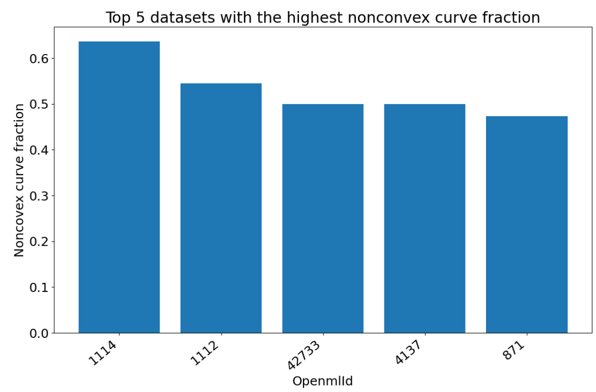

Figure 5: Top 5 datasets that produce the biggest amount of nonconvex learning curves

Curve type	Percentage of the mismatching evaluation
Convex	0.53%
Nonconvex	81.40%

Table 2: Percentage where conducting this experiment did not match the result of Experiment 2

Curve type	Mean Percentage of confidence interval violation
Convex	56.84%
Nonconvex	74.03%
Overall	58.36%

Table 3: Average percentage of anchors per learning curve that had confidence interval violation

Experiment 4: After calculating the confidence interval, how many anchors exhibit a mismatch in convexity measure?

A learning curve's anchors percentage that does not have the same sign on the second derivative on the upper or lower bounds of its confidence interval is demonstrated in Table 3. On average, 58.36% of the learning curves' anchors have such violations. While we only observe nonconvex learning curves, this percentage increases to 74.03% and on convex learning curves we see this average violation decrease to 56.84%.

Experiment 5: How well does the proposed metric work compare to the one proposed in the LCDB?

This experiment's results are summarized in Table 4. Out of the 10 learning curves with the highest violation according to LCDB, we have observed that only 1 learning curves are nonconvex according to our measure. However, while eyeballing we discovered that additional 3 learning curves were misclassified as convex by our measure, while 1 learning curve had undecidable shapes. Lastly, 5 learning curves had high violation in LCDB, but were convex learning curves.

6 Discussion

From the results of Experiment 1, we can see that the measure we proposed correctly estimated the convexity of the baselines in all three cases. We give an example of all three types of baselines in Figures 1, 2 and 3. We must also note that even though all the curve scores had the correct sign, some of the scores were very close to zero, which in this case would mean that the curve was neither convex nor nonconvex.

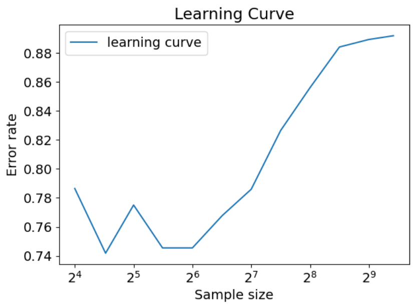

Figure 6: Nonconvex learning curve with OpenmlId 54, *sigmoid SVC* learner and score of -0.166

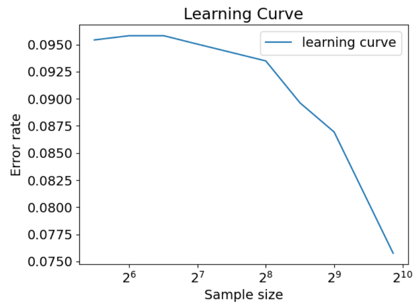

Figure 7: Nonconvex learning curve with OpenmlId 4137, *sigmoid SVC* learner and score of -0.333

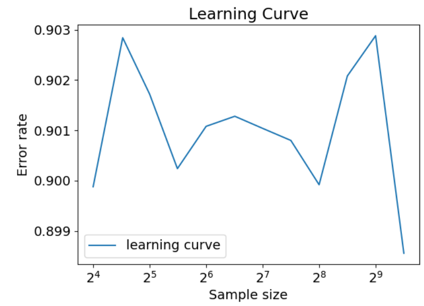

Figure 8: Nonconvex learning curve with OpenmlId 1503, *sigmoid SVC* learner and score of -0.333

OpenmlId	learner	LCDB violation	Measure
28	LDA	0.315	0.778
41163	LDA	0.308	0.556
14	QDA	0.276	0.231
40996	Ridge	0.272	0.75
1041	Ridge	0.255	0.625
554	Ridge	0.242	0.75
962	QDA	0.239	0.2857
728	Perceptron	0.234	0.0
40910	QDA	0.227	0.125
1441	QDA	0.225	-0.333

Table 4: Top 10 highest violations of convexity in LCDB compared to our measure. QDA - *Quadratic Discriminant Analysis*, LDA - *Linear Discriminant Analysis*

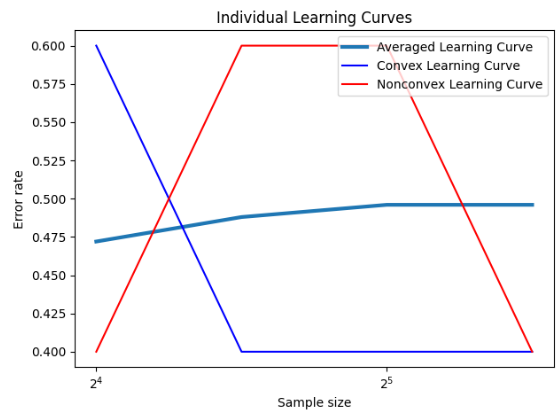

Figure 9: Individual learning curves of OpenmlId 346, learner *Passive Aggressive Classifier* and a score of -1.0, there are 15 convex and 10 nonconvex learning curves present, however,r we have plotted only one convex and one nonconvex curve for the sake clarity

Experiment 2 revealed that based on this paper's measure 8.86% of learning curves are nonconvex. Figure 4 proposes that nonconvexity might also be more common in some learners than others, meanwhile, Figure 5 points toward that nonconvexity might be a property of a specific dataset. From a total of 4366 dataset-learner combinations, we chose to manually explore all 206 learning curves with *sigmoid SVC* as the learner. Out of these learning curves we most commonly saw 3 types of nonconvex learning curves. The first type, had an increase in the error rate in the early stages of the curve and then its error rate stagnated on the rest of the curve, sometimes the increase in error rate came at a higher sample size, which bears resemblance to what we would see in overfitting, we can also observe this phenomenon in Figure 6. The second type, see an example in Figure 7, had the opposite shape as described in the previous type, the error rate did not change until a larger sample size, where the error rate suddenly dropped. Lastly, there were some curves that seemed to be neither convex nor nonconvex, however, they were still labelled as nonconvex learning curves, we show one such curve in Figure 8.

Analyzing Experiment 3 an interesting pattern emerges from Table 2, we observe that whenever a curve is nonconvex according to Experiment 2, Experiment 3 will predict that the same curve is convex 81.40% of the time. There are two possible explanations for this phenomenon, one is that the learning curve should be nonconvex, however since we take only 25 learning curve samples, there are not enough learning curves for Experiment 3 to study. The second possible explanation is that these learning curves should be convex as suggested by Experiment 3 and that the nonconvexity arises from the noise in the data. An example of mismatching results between Experiment 2 and 3 can be seen in Figure 9, where the learning curves' score was -0.666, however, there were 15 convex and 10 nonconvex individual learning curves.

We can see a trend of nonconvex learning curves having more violations according to both Experiments 3 and 4, however, in Experiment 4 the violations between convex and nonconvex learning curves were not as big as in Experiment 3. Since we perceived such a large average violation in Experiment 4, we can identify the main cause of the problem, which is that we simply did not have enough data points, thus making the variance large. This result also helps us explain why were there so many nonconvex learning curve violations in Experiment 3. Therefore by sampling more learning curves for each dataset-learner combination, we should potentially reduce the number of violations in both Experiments 3 and 4 leading us to better accuracy in our measure.

Experiment 5 gives us insight into how the convexity violation measure in LCDB works. Ranking the learning curves based on the highest violation has both advantages and disadvantages compared to our measure. The biggest drawback in LCDB's measure is if the learning curve has double descending property, see Figure 10, it incorrectly assumes violation, when in reality there are two discontinuous convex curves. The main advantage is that the method rewards a learning curve for having a single large convexity violation which determines the overall shape of the curve, however, this does not work if the learning curve is double descending or it is noisy.

Returning to our measure Experiment 5 also helped us identify some of our shortcomings. In Figure 11, we can eyeball that the curve should be nonconvex, however, we identified it as convex. The reason for this particular misclassification is nonconvexity is only visible on the global scale, while due to noise our algorithm mistakes some of the anchor points being convex. One solution, which was also suggested in Experiments 3 and 4 is to simply increase the number of individual learning curves per learning curve. Another possible fix would be to not consider 0 valued second derivatives as convex anchors, since in these types of curves the error rate seems to remain constant after the initial increase in error.

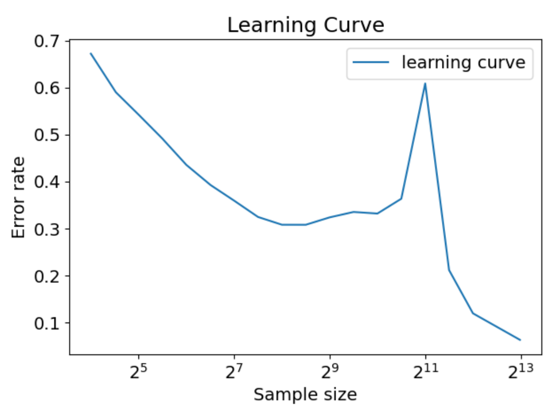

Figure 10: Double descent learning curves of OpenmlId 41163, learner Linear Discriminant Analysis and a score of 0.556

7 Conclusions and Future Work

In summary, the main goal of this paper was to estimate how many learning curves are nonconvex in LCDB and to propose an algorithm to find them. Starting with Experiment 1 we concluded that our proposed measure worked for the baseline learning curves, furthermore, in Experiment 2 we analyzed the nonconvex learning curves inside LCDB and we observed that the nonconvex learning curves are undesirable, because either we do not get continuous performance error

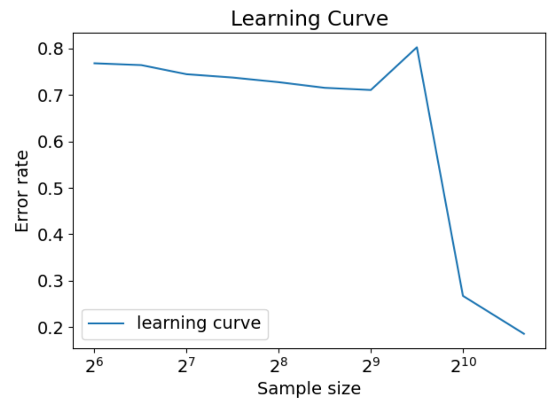

Figure 11: Misclassified learning curve as convex, OpenmlId 14, learner Quadratic Discriminant Analysis and a score of 0.231

rate decrease only a sudden drop, or with increasing sample size we find out that the error rate is not decreasing, instead the error rate was the lowest during low sample sizes. In Experiments 3 and 4 we shifted our attention to see how likely we have correctly identified a nonconvex learning curve and we concluded that to acquire more precise results we would need to increase the number of learning curves generated per dataset-learner combination. Finally, in Experiment 5 we have pointed out some of the shortcomings of the method proposed in LCDB, while also discovering edge cases where our algorithm does not perform correctly.

This research is not yet complete, first, we want to address the Second Derivative Estimator Algorithm, which finds the second derivative by taking linear regression times. While it works in practice, there should be alternative methods to do the same more efficiently. One of my suggestions for this is to use quadratic regression instead of linear regression, mainly because we can estimate the second derivative with one regression instead of two. Additionally, I would propose stricter criteria for a curve to be considered nonconvex, such as defining a convex curve through a convex set or a convex hull. Lastly, our final suggestion would be to increase the number of learning curves generated for each dataset-learner combination from 25 to at least 250. This would certainly reduce the number of violations discovered in Experiments 3 and 4.

8 Responsible Research

While this paper has little to no direct ethical implications we should still put importance on other aspects of responsible research such as reproducibility. We publicize the codebase used to generate the results of this paper on GitHub [2]. The dataset used in this paper was originally produced in LCDB [5], which contained the accuracies of the dataset-learner combinations. From the LCDB library, we have also used the tool to load the learning curve accuracies. Additionally, *sklearn* library was used for the *Linear Regression* implementation, *scipy* was used to calculate the *Critical Value*,

[2] https://github.com/kikigogo9/Research-project

lastly we used *matplotlib* for creating graphs.

Aside from general statistics done on evaluating the learning curves, we also had to cherry pick some of them to provide us with examples. The issue here becomes that these few learning curves are not representing the entire dataset, due to time constraints we did not have the necessary time to analyze all 4366 learning curves individually.

We have also employed useful tools such as *Grammarly* and *ChatGPT* to help us check the grammatical correctness and consistency of the paper. The most commonly used prompt in *ChatGPT* was 'Please check for grammar mistakes: <section>', after which we double-checked the output to make sure the corrected sentences would still retain their original meaning.

References

[1] Derek Hoiem, Tanmay Gupta, Zhizhong Li, and Michal Shlapentokh-Rothman. Learning curves for analysis of deep networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4287–4296. PMLR, 18–24 Jul 2021.

[2] Georgi Li and S. Rajagopalan. A learning curve model with knowledge depreciation. *European Journal of Operational Research*, 105(1):143–154, 1998.

[3] David Bosch, Ashkan Panahi, Ayca Ozcelikkale, and Devdatt Dubhashi. Random features model with general convex regularization: A fine grained analysis with precise asymptotic learning curves. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 11371–11414. PMLR, 25–27 Apr 2023.

[4] Christos Thrampoulidis, Samet Oymak, and Babak Hassibi. A tight version of the gaussian min-max theorem in the presence of convexity. *CoRR*, abs/1408.4837, 2014.

[5] 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.

[6] 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.

[7] Dastan Maulud and Adnan M Abdulazeez. A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, 1(4):140–147, 2020.

[8] Karsten Ahnert and Markus Abel. Numerical differentiation of experimental data: local versus global methods. *Computer Physics Communications*, 177(10):764–774, 2007.

[9] Felix Mohr and Jan N van Rijn. Learning curves for decision making in supervised machine learning–a survey. *arXiv preprint arXiv:2201.12150*, 2022.

[10] Samprit Chatterjee and Ali S Hadi. *Regression analysis by example*, page 37. John Wiley & Sons, 2006.