Non-Monotonicity in Empirical Learning Curves

Identifying non-monotonicity through slope approximations on discrete points

Codrin Socol[1]

Supervisors: Dr. Jesse Krijthe[1] **, Dr. 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 28, 2023

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

Abstract

Learning curves are used to shape the performance of a Machine Learning (ML) model with respect to the size of the set used for training it. It was commonly thought that adding more training samples would increase the model's accuracy (i.e., they are monotone), but recent works show that may not always be the case. In other words, some learners on some problems show non-monotonic behaviour. To this extent, we introduce a new method to identify non-monotonicity in empirical learning curves by approximating the curve's slope through regression around the discrete points it is defined on. This paper formalises this metric and then evaluates its accuracy through different experiments. Finally, we run the proposed metric on a subset of the extensive Learning Curve Database (LCDB) by Mohr et al. to gain better insights into the problem of non-monotonicity of learning. We found that the metric can identify non-monotonicity in learning curves well *(98% experimental accuracy)* and does not consider small increases due to measurement error as non-monotonicity in the curve. Finally, we have identified that non-monotonicity may be a property of some classifiers, such as Linear Discriminant Analysis. Moreover, we identified that non-monotonicity is frequently observed in datasets with faster training times.

Keywords: learning curve · non-monotonicity · meta-learning · LCDB · Machine Learning

1 Introduction

Learning curves outline the evolution of model performance with respect to increasing the training set size. They can point out how large the sample set needs to be through extrapolation to achieve the desired predictor accuracy. By estimating the input data size, organisations can forecast training costs for a model [1].

An interesting property to discuss on the topic of learning curves is *monotonicity*. Intuition would point out that learning curves are monotone: by increasing the number of samples in the dataset, we lower the error rate of the prediction algorithm. In other words, we strive for "well-behaved learning curves"[2]. Monotonicity ensures that we can expect a lower error rate in prediction by enlarging the training set, which can better predict the training costs for ML classifiers. In their work [3], Marco Long et al. showed that even standard learners, such as least squares regression or linear models trained with the hinge loss, show non-monotonic behaviour, which counters this initial belief. These findings set the foundation for a new possible research direction, i.e., understanding how many learning curves are non-monotone and what factors could influence the monotonicity of curves. This research paper aims to answer the following question:

How many learning curves are non-monotone, and what influences this?

This paper will propose a heuristic for identifying non-monotone learning curves that needs to also account for and handle noise in the data. Then, a qualitative analysis of this heuristic will be conducted to showcase its effectiveness. Thirdly, using this method to identify non-monotone learning curves, an analysis of the LCDB database[1] will be conducted to find how large the subset of curves that show non-monotone behaviour is. Finally, we will draw some conclusions on why some learning curves show non-monotonic behaviour, as well as discuss the limitations of the proposed heuristic.

The paper will follow this structure. Section 2 discusses related work from existing literature. Section 3 introduces the methodology of the research, followed by Sections 4 and 5, which go into detail about the experimental setup and results, respectively. Section 6 discusses interpretations of the experimental results, while Section 7 outlines the responsible aspect of the research. The final section draws the conclusions of this work and points to future possible research directions.

2 Related Work

Viering et al. mentioned in [4] that the monotonicity of learning is an open problem yet to be explored and tackled. Other works, such as [5], concluded that studying the monotonicity of meta-learning is not a trivial task. Even though it is desired, the property of monotonicity may not always be present in learning curves (LCs). In their review [2], authors pointed out that many works assume that the curves will be *well-behaved*, i.e., smooth and having monotonically non-decreasing performance [6]. In other words, it is commonly considered that the model's error rate does not increase with adding more training samples.

Previous works also proposed metrics to make learners more monotone [5], but none try to identify non-monotonicity in LCs.

The Learning Curve Database (LCDB) paper proposes a metric to study the non-monotonicity of learning curves by looking at the maximum function increase of an overall-descending learning curve (maximum violation) [1]. This metric evaluates how large the monotonicity violation is, which can be effectively used to rank learning curves based on how large the violation is. Using this metric, authors discovered that it is useful to study monotonicity past a certain minimum training set size due to an increased measurement imbalance. However, it is unclear where the line is drawn between what is considered measurement noise and what is considered actual non-monotonicity in the curve using this metric. Thus, using this metric to judge the monotonicity of curves as a classification problem is challenging.

The LCDB paper also discusses another phenomenon called *peaking* (or sample-wise double descent/ascent), usually encountered in neural networks, which describes a small region of monotonicity violation in an otherwise monotonic curve. The authors propose a different metric to identify peaking, which performs a binary classification of whether a curve observes this phenomenon. Peaking is just a special form of non-monotonic behaviour in learning curves.

Based on the review by Viering et al. [2], many academics

assume learning curves to be monotonic. Thus, little work has been performed to study the non-monotonicity of learning curves. Analysing non-monotonic behaviour in LCs could answer whether the monotonicity in learning curves is a property of the model learner, of the dataset, of both or neither. Understanding this behaviour could lead to better extrapolating the training costs for a model or aiding in the model selection process of a problem [7].

Authors of [8] defined the term *anchor point* to denote the performance of the ML model on a certain training sample size (i.e., the point on the learning curve), which we will also use henceforth.

3 Methodology

We propose the following setup to answer the question of the non-monotonicity of learning curves. Section 3.1 will introduce a new heuristic to identify non-monotonicity in empirical learning curves. Section 3.2 outlines the setup of two evaluation techniques to perform a qualitative analysis of the metric. Section 3.3 describes the setup of a quantitative study on a subset of the LCDB, an extensive collection of learning curves.

3.1 Identifying Non-monotonicity

The first step in studying the non-monotonicity property of learning curves is to define a metric that classifies the LCs as monotonic. For this purpose, we outline the following algorithm. For every anchor point on the learning curve, 20 points are chosen closely around it. The model is then retrained on training set sizes equal to these points to calculate the model's error rate on these, respectively. On these 20 points and the model errors associated with them, approximate the slope of the LC around the respective anchor point using a Linear Regression model. Repeat this process with different train set sampling seeds and average the slopes obtained from different measurements. If two consecutive anchor points have a positive slope (i.e., increasing LC), we classify the learning curve as non-monotonic. This metric is formally described in *Algorithm 1*.

The algorithm receives as input the list *N* of anchor points that define the discrete LC, two lists, $outer\_seeds$ and $inner\_seeds$ that are used to randomise the train/test/validation splits, the dataset features and target attribute, X and Y, respectively, as well as the $model$ to learn the dataset. The proposed metric approximates the slope of a learning curve at all anchor points by first retraining the model 20 times, with different training set sizes, in the proximity of the anchor point *(lines 7-10)*. The errors obtained from retraining the classifiers at these points are used to train a Linear Regression model (LR) to approximate the slope of the learning curve on the respective anchor points *(lines 11-12)*. This number of points (20) was chosen as a good tradeoff between runtime speed and accuracy for training the LR, as this number highly influences the time complexity of running the metric. This process is repeated on different randomisation $seeds$ for splitting the dataset into training, validation and testing sets. The slopes obtained are averaged to reduce the LR approximation error. If the slopes are positive at two consecutive anchor points on the LC, then the curve is considered to be increasing on the interval between these two anchors and thus classified as non-monotonic *(lines 17-22)*.

Algorithm 1 Identifying non-monotonicity through anchor point slope approximation

Input: N ▷ List of anchor points
Input: $outer\_seeds$ ▷ Random seeds for (train,val) and test split
Input: $inner\_seeds$ ▷ Random seeds for train validation split
Input: X ▷ Dataset features
Input: Y ▷ Target attributes
Input: $model$ ▷ The classifier the curve is modelling
Input: $final\_slopes \leftarrow []$
1: **for all** k in N **do**
2: $points \leftarrow$[k-10,k-9,...,k+10]
3: $slopes \leftarrow []$
4: **for all** s_1 in $outer\_seeds$ **do**
5: **for all** s_2 in $inner\_seeds$ **do**
6: $errors \leftarrow []$
7: **for all** p in $points$ **do**
8: $error \leftarrow Train(model, p, s_1, s_2, X, Y)$
9: $errors.append([p, error])$
10: **end for**
11: $lin\_reg \leftarrow LinearRegression(errors)$
12: $slopes.append(GetSlopeFromModel(lin\_reg))$
13: **end for**
14: **end for**
15: $final\_slopes.append(Mean(slopes))$
16: **end for**
17: **for all** $i = 0$ until $|N| - 1$ **do**
18: **if** $final\_slopes[i] > 0$ && $final\_slopes[i+1] > 0$ **then**
19: **return** $True$
20: **end if**
21: **end for**
22: **return** $False$

3.2 Metric Evaluation

This subsection describes the method for evaluating the algorithm introduced in Section 3.1 through two qualitative studies.

Accuracy Analysis

This experiment will look into the classification power for learning curves. In other words, the results shall yield the accuracy rate of the proposed metric for the non-monotonic learning curves. The experiment will use the following structure. Firstly, artificial learning curves will be generated from parameters known to yield either a monotonic or non-monotonic curve. These will act as the ground truth of the experiment. Then, the proposed metric is run on these curves, and predictions will be compared to the ground truth label.

Ablation Study

The second experiment will answer whether it is necessary to consider a learning curve as non-monotonic if the slopes in two consecutive anchor points are positive or if it is enough to consider only one anchor point at each step. It is also important to evaluate the performance of the algorithm proposed to effects such as $peaking$. As described in Section 2, $peaking$ represents a small region where the learning curve exhibits

a slight increase while decreasing everywhere else on its domain.

This study will be performed in two steps. **Step 1** will compare the algorithm's accuracy under the two conditions described above, by running the metric in both scenarios. **Step 2** will measure which of the two setups better handles the $peaking$ phenomena in learning curves by introducing peaking at an anchor point in a monotonic learning curve and then running the algorithm under both scenarios.

3.3 Evaluating non-monotonicity in the LCDB

The LCDB is an extensive collection of learning curves containing 20 classifiers and over 240 datasets. Thus, it offers a wide variety of learning curves to be analysed, the perfect setting for understanding how large the set of learning curves that exhibit non-monotonic behaviour is and the causes that may influence this. This experiment aims to assess the non-monotonicity of empirical learning curves in the LCDB by running Algorithm 1 on a subset of it. We aim to identify the ratio of non-monotonic learning curves to monotonic ones. We also strive to understand whether monotonicity is a property of the learner, of the dataset, of both or none.

4 Experimental Setup

This section details the setup of the three experiments conducted on the introduced metric and the general environment in which the studies are conducted.

4.1 Environment

The studies described in this paper, as well as the metric to evaluate non-monotonicity, are written in Python[1] and run through Jupyter Notebooks[2]. All experiments interface with the LCDB API[3] to retrieve meta-data about the learning curves, such as the list of anchor points, training times, retrieving the datasets, as well as some plotting utility functions. In turn, the LCDB uses Scikit Learn [9] to implement the classifiers and OpenML [10] to source the datasets.

All source code and results produced are available publicly in a GitHub[4] repository at [11].

4.2 Experiment 1: How accurate is the introduced metric?

The first qualitative study performed on the proposed metric is an accuracy test. The experiment generates artificial learning curves based on the $exp3$ [2] parametric model, outlined in Equation 1. Parameters a, b and c are variables that are tuned when the model is fitted to a learning curve.

$$exp3(x) = a \cdot e^{-bx} + c + \varepsilon \tag{1}$$

$$\frac{d}{dx} exp3(x) = -ab \cdot e^{-bx} \tag{2}$$

The $exp3$ parametric model has been chosen for this experiment as it was found to be a good modeller for LCs [2]. The ε parameter represents noise added to the learning curve function to simulate the measurement error, as in *real* learning curves. This parameter is sampled from a standard normal distribution with a mean of 0 and a standard deviation of 0.002. This standard deviation value was chosen to keep the noise small enough not to introduce drastic changes in the function but large enough to simulate real measurement noise.

Equation 2 gives the first derivative (slope) of the parametric model used for this experiment. The sign of the first derivative indicates the monotonicity of the learning curve. If the slope is negative, the LC decreases, while a positive first derivative means the LC will be ascending. By adding more training samples to the classifier, we aim to minimise the learning error. Thus, the learning curve should ideally be decreasing. This is the definition we used for monotonic learning curves. Thus, a monotonicity violation would mean that the curve has a region on its domain where it is increasing.

Equation 2 highlights that the value combinations of parameters a and b determine the sign of the first derivative of the $exp3$ function. Parameter c is not present in the first derivative formula, and thus its values do not influence the slope.

- $a < 0, b > 0$ - Positive derivative (ascending $exp3$);
- $a > 0, b < 0$ - Positive derivative (ascending $exp3$);
- $otherwise$ - Negative derivative (descending $exp3$).

The experiment was run on 3769 different learning curves with different values for parameters a,b and c. The parameter values were sampled from curve fits on the $exp3$ model, previously computed by authors of the LCDB. On top of this, the ε parameter was added for modelling data noise and to make the experiment more realistic. In the LCDB, these curves represent fits on *real* problems that try to approximate the real, unknown learning curve, which is impossible to calculate. However, not all these fits are representative, as some have poor fit accuracy. In order to ensure the reliability of this study, only half of the fits were considered, i.e., those with the minimum Mean Squared Error on the last anchor point. Finally, the experiment learning curves were split as:

- 3179 Non-monotonic LCs;
- 590 Monotonic LCs.

4.3 Experiment 2: Ablation study - *Is it necessary to consider two consecutive anchor points?*

This experiment aims to identify whether using two consecutive anchor points to judge non-monotonicity is necessary or if one is enough *(step 1)*. It also studies whether the metric can handle curve $peaking$ *(step 2)*.

Step 1. Experiment 1 is rerun with the slope sign condition changed to one anchor point instead of two consecutive ones to judge non-monotonicity. We hypothesise that using only one anchor point is insufficient to clearly indicate whether the empirical LC is non-monotonic due to noise in data and/or measurements. Thus, we perform this experiment to validate this hypothesis.

[1] https://www.python.org/

[2] https://jupyter.org/

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

[4] https://github.com/

Step 2. The 590 monotonic LCs from Experiment 1 were slightly modified by introducing an artificial peaking around the anchor at index 5, such that the model error at anchor 5 is higher than the error at anchor index 4. This peaking is only observed around anchor index 5. The added peaking is randomly sampled from the same normal distribution as the ε parameter from Equation 1. The rest of the learning curve remained unchanged. The algorithm was executed on these modified LCs under both conditions, the initial method and the ablation of the second positive slope condition, to identify which scenario better identifies these $peakings$. We expect that the algorithm will manage to correctly identify it since *peaking* is a monotonicity violation phenomenon.

4.4 Experiment 3: How many learning curves are non-monotonic in the LCDB?

The LCDB is an extensive collection of learning curves and, thus, is perfect for evaluating non-monotonicity. Due to its size and the long time it takes to run the metric from Algorithm 1, only a subset of the LCDB will be used for this experiment. In [1], authors have identified that some learners, such as Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), or Stochastic Gradient Descent (SGD) show peaking *(sample-wise double descent/ascent)* [12], a phenomenon encountered in neural networks, where model performance decreases after an initial increase, followed by another increase afterwards.

This experiment aims to identify the set size of LCs that show non-monotonic behaviour by running the metric proposed in this research on a subset of the LCDB. Of the 20 learners currently supported by LCDB, only 11 will be considered for this experiment. Based on previous findings in [1], we sampled classifiers for this experiment that showed either highly increased or decreased chance of being non-monotonic or to contain *peaking*. The time necessary to train a classifier varies from model to model while also depending on the dataset. Some learners take very long to train on certain problems. For example, the SGD classifier took almost two hours to learn the dataset with $openmlid$ 1567 on a personal machine. Thus, running the experiment on the entire LCDB is unfeasible within the timeframe of this research project. To this extent, the datasets were chosen based on how fast the 11 chosen classifiers learned them. Given the relatively short time to perform this study, we have decided to sample the 20% fastest datasets by training time, to include as many learning curves as possible in this evaluation.

Not all classifiers can learn all datasets. For example, we identified experimentally that LDA could not be used to learn a target attribute that does not contain more than one sample for a certain value, as the covariance is ill-defined. Thus, the datasets that were unsuited for use were filtered out. Moreover, some datasets chosen for this study have a small number of samples, some having less than 256 or even less than 64. For this reason, the number of LCs that will be analysed in this experiment is smaller at starting anchors 64 and 256, respectively, compared to 16.

5 Experimental Results

This section describes in-depth the results of the three experiments described in Section 3.

5.1 Experiment 1: Accuracy of introduced metric

Table 1: Accuracy Test Results. The brackets describe the percentage of correctly classified curves from the total number of monotonic or non-monotonic LCs, respectively.

	Actual Non-monotonic	Actual Monotonic
Predicted Non-monotonic	3140 ***(98.77%)***	36 *(6.10%)*
Predicted Monotonic	39 *(1.23%)*	554 ***(93.90%)***

Table 1 describes the results from the accuracy test experiment. The introduced method has correctly identified non-monotonic LCs with a high accuracy of 98.77% and monotonic LCs with 93.9% accuracy. The experiment outlines the performance of the metric and represents a good indication that it has the potential of correctly classifying $real$ learning curves as well.

5.2 Experiment 2: Ablation Study

Table 2 describes the results from **Step 1** of the ablation study. Compared to the accuracy study described in Experiment 1, we can see a slight increase $(+0.82\%)$ in the non-monotonicity prediction accuracy. On the other hand, there is a large decrease (-31.87%) in the monotonicity prediction performance. During this experiment, the metric failed to correctly classify around 4 out of every 10 monotonic learning curves. This means that when considering only one anchor point to decide on monotonicity, the metric tends to classify a learning curve as non-monotone at the slightest increase. This could happen due to noisy measurements in a monotonic learning curve, thus leading to misclassification. The results of this ablation study reaffirm the initial hypothesis and confirm the need to consider the LC slope in at least two consecutive anchor points to judge whether a learning curve is non-monotonic or not to ensure the reliability of the classification.

Table 2: Ablation Study results. The brackets describe the increase/decrease compared to Experiment 1 Results from **Table 1**.

	Actual Non-monotonic	Actual Monotonic
Predicted Non-monotonic	3166 ***(+0.82%)***	224
Predicted Monotonic	13	366 ***(-31.87%)***

Results from **Step 2** are highlighted in Table 3, which describes how well the algorithm handles $peaking$, both in the

initial and in the ablation setup. Considering two consecutive anchor point slopes to judge monotonicity proved unsuited for identifying peaking around one single anchor. Under this setting, only 19 out of 590 LCs with $peaking$ were correctly classified. This means the initial assumption was false, and the heuristic cannot identify peaking if it occurs only at one anchor. On the other hand, the ablated metric performed significantly better, with a 45.25% success rate, but it still failed to determine more than half of the experimental learning curves as non-monotonic.

Table 3: Metric evaluation on 590 artificial monotonic LCs with peaking at anchor index 5.

	Correctly classified as non-monotonic	% of total curves (590)
Ablation Metric	267	**45.25%**
Initial Metric	19	**3.22%**

5.3 Experiment 3: How many learning curves are non-monotonic in the LCDB?

Table 4 describes the results of the experiment. Figure 1 shows an aggregation of results across all learners. It seems that for the datasets studied, analysing non-monotonicity from the beginning of the curve (anchor point with 16 training samples) considers more LCs as non-monotonic. Thus, it may be useful to consider studying monotonicity from a certain point onwards, such as the anchor point with 256 training samples, as learning is more prone to errors when a minimal amount of training data is used[1]. This outcome is consistent with the findings in the LCDB paper. In the case where the analysis starts at 256 training samples, the experiment identified that 256 learning curves contained non-monotonic behaviour out of 413 total curves ($\sim 62\%$).

Figure 2 brings forth the experiment's results per classifier. The figure also suggests that non-monotonicity could be a property of the learner. For example, the *Extra Trees Classifier* and the *Random Forest Classifier*, which both use decision trees to classify the target attributes, show significantly less non-monotonic behaviour than the other learners studied in this experiment. On the other hand, LDA and the SVC variations showed a consistent increase in number of curves that exhibit non-monotonicity.

6 Discussion

Section 3.1 introduced an algorithm that can be used to identify non-monotonicity in empirical learning curves. It does so by approximating the LC's slope at key points by training a Linear Regression (LR) model on 20 points around the anchors. The choice of the number of points heavily impacts the result of the algorithm. If too few points are used, the LR will be prone to a higher error. If too many points are used, the algorithm's time complexity will drastically increase, as LR is trained at every anchor point on the curve multiple times. For each point used to train, the regressor needs the classifier to be retrained to retrieve its training error. As a trade-off between speed and performance, 20 points have been used for slope approximation at each anchor point. However, this amount is not necessarily ideal. One limitation of using Linear Regression for slope estimation is its error. Suppose the LC is almost constant on the points used to train the LR (slope is approximately zero). In that case, the slope resulting from LR may become positive (or negative) due to LR training error and, in turn, lead to misclassification.

Table 4: Monotonicity evaluation of different classifiers. The table shows the number of classified non-monotonic curves (bolded) and the total number of curves analysed (between brackets)

Classifiers	Non-monotonic LCs at first anchor 16	Non-monotonic LCs at first anchor 64	Non-monotonic LCs at first anchor 256
Linear Discriminant Analysis (LDA)	**123** *(125)*	**116** *(121)*	**84** *(105)*
Quadratic Discriminant Analysis (QDA)	**33** *(33)*	**29** *(31)*	**21** *(29)*
SGD Classifier (SGD)	**69** *(69)*	**57** *(67)*	**36** *(57)*
Gradient Boosting Classifier	**13** *(17)*	**11** *(17)*	**11** *(14)*
Logistic Regression (Log.Reg.)	**41** *(42)*	**40** *(41)*	**30** *(37)*
SVC (linear)	**28** *(30)*	**22** *(29)*	**13** *(24)*
SVC (poly)	**38** *(38)*	**26** *(37)*	**19** *(32)*
SVC (rbf)	**34** *(34)*	**22** *(33)*	**16** *(29)*
SVC (sigmoid)	**35** *(35)*	**23** *(34)*	**16** *(29)*
Extra Trees Classifier	**9** *(51)*	**7** *(49)*	**4** *(38)*
Random Forest Classifier	**14** *(26)*	**11** *(24)*	**6** *(19)*

Experiment 1 showed the effectiveness of the metric proposed in Algorithm 1, correctly classifying most of the curves used in the experiment. The learning curves used also contain random noise in order to mimic measurement errors in *real* LCs. The study concluded that the metric accounts for small increases in the curve that are caused by noise in the data and does not classify this as non-monotonic behaviour, which is intended, thus making the algorithm a good candidate for judging non-monotonicity of curves in practice.

Experiment 2 proved the necessity of considering the LC slope approximations in two consecutive anchor points to handle data noise better and decrease the misclassification of

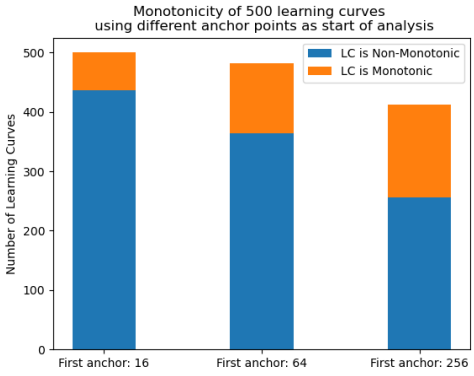

Figure 1: Monotonicity evaluation of 500 learning curves, with different anchors as the first point of monotonicity analysis: 16, 64 and 256 training samples, respectively.

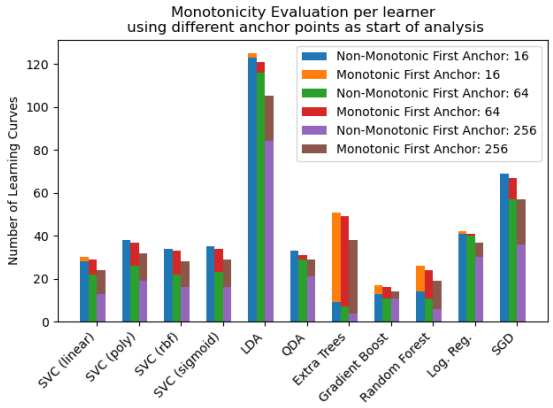

Figure 2: Monotonicity evaluation of 500 learning curves, with different anchors as the first point of monotonicity analysis. Results are displayed per classifier. Learner abbreviations are consistent with those in Table 4.

monotonicity errors. One limitation of this experiment is that it only uses the $exp3$ parametric model to generate artificial learning curves. Some fits of the $exp3$ model may not be very accurate. This is the reason why the half of the fits from the LCDB with highest MSE errors on last anchor were removed for the purpose of this experiment. Moreover, $exp3$ tries to best model the *true* learning curve, but it is still an estimation. The *true* function that models the learning curve is impossible to determine.

Experiment 2 showed that the algorithm performance drastically decreased for classifying monotonic LCs, by more than 30%. Upon careful inspection, a few of these incorrectly classified LCs were almost *constant* (i.e., neither increasing nor decreasing) on the interval defined by the anchor points. Adding noise in the data to test how the algorithm handles the noise to a constant function led to the LC being misclassified. Here we can see the limitations of approximating the slope using Linear Regression. However, using the algorithm and checking the condition for two consecutive anchor point slopes, we were able to mitigate this misclassification and, thus, again proving the need to consider two anchor points instead of just one.

Another limitation of the introduced metric is that it fails to identify peaking in the curve if it is observed around one single anchor point. This was observed during **Step 2** of Experiment 2. This happens as a consequence of using slopes at two anchor points to judge non-monotonicity. The algorithm will identify curve peaking if it occurs at more than one consecutive anchor point, as it will judge it the same as any other monotonicity violation.

An important aspect to consider when introducing Algorithm 1 is the time complexity it takes to execute it. To reiterate, the algorithm retrains the model 20 times and uses the training errors from these runs to approximate the LC's slope through Linear Regression. This process is repeated on 25 different randomisation seeds for each anchor point of the learning curve. As a direct result, the metric runs slowly, in some cases taking more than one hour to judge non-monotonicity on large datasets. The worst performance encountered during Experiment 3 was the *Gradient Boosting Classifier* on the dataset with OpenML id 18, which ran for 2.5 hours on a personal machine. From this point of view, the metric is unfeasible for analysing an extensive collection of learning curves, such as the LCDB, unless run on more powerful machines and with parallelisation.

Experiment 3 entailed the execution of the algorithm proposed on a subset of the LCDB. Datasets were considered based on how fast the chosen classifiers learned them. This limits the amount of information that can be extracted from the experiment. For example, some datasets are trained fast because they contain fewer samples (less than 300). Learning these datasets may be difficult, as high variance can happen at small training set sizes [1]. Experiment 3 outlined that a high number of learning curves are non-monotonic on datasets that take the least amount of time to train. Around 60% of the curves used for the experiment were non-monotonic. There is no way to tell whether the same ratio is obtained on the other datasets and/or learners that were not included in this study. It may be the case that non-monotonicity is a property of these datasets that are learned fast, and datasets that take longer to train are more monotonic overall. From this perspective, Experiment 3 is inconclusive and more research is needed to identify the *true* ratio of non-monotonic curves.

7 Responsible Research

An important aspect that surfaces when conducting research is the reproducibility of the results. The researcher is responsible for showing that their research is conducted ethically and that they are fully transparent with their work.

We have ensured that the results of this study are reliable through careful planning and conducting the experiments. This work has been peer-reviewed by fellow students and the supervising team at different project stages. Their feedback has been incorporated into this final work. Moreover, the entire research has been conducted with reproducibility in mind, and results have been transparently reported in this paper.

To ensure the experiments are reproducible, we have publicly made the experiments and related data, source code and

plots available. These can be found in a GitHub repository, here [11]. The repository contains one Jupyter Notebook for each experiment conducted. The $ReadME.md$ file contains information on how to install all necessary dependencies and how to run the experiments. Moreover, the experiments will yield the same results outlined in Section 5 when run just by following the steps in the Methodology and Experimental Setup sections or through running the code in the repository mentioned above.

Finally, the training times discussed in this paper are highly dependent on the hardware used for running the studies. The experiments were conducted on a laptop running Intel i5-1235U (up to 4.4GHz), with 24GB of main memory. Running the experiments on different hardware specifications may result in different training times.

8 Conclusions and Future Work

To conclude, this paper introduced the need to identify non-monotonicity in learning curves (LC). In order to establish this property, we proposed an algorithm that could be used to judge whether an LC is monotonic by approximating the slopes of the LC in the points it is defined on. The pseudocode is in Algorithm 1. We then evaluated its accuracy in Experiment 1. This showed overwhelmingly good results and indicated that the algorithm might be proper to judge whether curves are non-monotonic. Experiment 2 proved our hypothesis that using slopes at two neighbouring anchor points to judge non-monotonicity is necessary. Moreover, it indicated that the metric could not identify $peaking$ in the curve if it is only observable around one anchor. Experiment 3 showed that most learners experience non-monotonic behaviour at the beginning of the learning curve, but this decreases if the monotonicity analysis starts at a later anchor point, such as 256. Moreover, the experiment shows that non-monotonicity may be a property of some classifiers, such as *Linear Discriminant Analysis*, for the datasets it learns the fastest.

Due to time and processing power limitations, only a fraction of the LCDB was analysed during Experiment 3. It may prove interesting to run the algorithm proposed in this paper in its entirety. One purpose of doing this could be to determine which classifiers inherently show non-monotonic behaviour. The results of Experiment 3 were inconclusive in determining whether certain datasets also show repeated non-monotonic behaviour, which may come to light if the metric is run on the entire LCDB.

Another research direction could be to slightly adapt the metric described in this paper to rank learning curves based on, for example, in how many anchor points we identified positive slopes. This will output an ordered list of LCs based on the *degree of non-monotonicity*, which can then be used to create a comparative analysis with the maximum violation metric described in [1].

Another improvement to be done could be optimising the time complexity of the algorithm proposed. A possibility could be to approximate the slope by calculating the gradient of two points around the anchor point instead of using Linear Regression. This will reduce the overhead complexity of retraining the model 20 times and executing Linear Regression on every anchor point. However, we hypothesise this would, in turn, degrade the metric's performance. Another improvement that could be done on the metric is to run the regressor only once per anchor point, to improve the time complexity at the expense of handling the approximation error of LR. A third option to improve the runtime may be to add parallelisation to the algorithm by running it on multiple threads simultaneously or even using GPU computing.

The research outlined in this paper builds on top of the LCDB paper; thus, the experimental setup partially replicates the environment described there. An example is using error rate to explore the shapes of learning curves, the same metric the authors use in the LCDB. However, the LCDB supports other performance metrics, such as *logloss* or *f1*. The learning curve shape may differ from metric to metric. To this extent, it may prove useful to analyse how monotonicity is affected by changing the performance metric.

All in all, this paper aimed to shine more light on the non-monotonicity of meta-learning of ML classifiers by looking at how we can properly identify it, and which learners and datasets exhibit this behaviour. However, this work is not a definitive answer to the issue of non-monotonicity. The monotonicity of learning curves still remains an open question that requires more research to fully answer.

References

[1] Felix Mohr, Tom J. Viering, Marco Loog, and Jan N. van Rijn. Lcdb 1.0: An extensive learning curves database for classification tasks. *Machine Learning and Knowledge Discovery in Databases*, pages 3 –19, 2023.

[2] Tom Viering and Marco Loog. The shape of learning curves: a review, 2022.

[3] Marco Loog, Tom Viering, and Alexander Mey. Minimizers of the empirical risk and risk monotonicity. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[4] Tom Viering, Alexander Mey, and Marco Loog. Open problem: Monotonicity of learning. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 3198–3201. PMLR, 25–28 Jun 2019.

[5] Tom J. Viering, Alexander Mey, and Marco Loog. Making learners (more) monotone, 2019.

[6] Gary M Weiss and Alexander Battistin. Generating well-behaved learning curves: An empirical study. In *Proceedings of the International Conference on Data Science (ICDATA)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2014.

[7] P. Brazdil, J.N. van Rijn, C. Soares, and J. Vanschoren. *Metalearning: Applications to Automated Machine Learning and Data Mining*. Cognitive Technologies. Springer International Publishing, 2022.

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

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] Matthias Feurer, Jan N. van Rijn, Arlind Kadra, Pieter Gijsbers, Neeratyoy Mallik, Sahithya Ravi, Andreas Mueller, Joaquin Vanschoren, and Frank Hutter. Openml-python: an extensible python api for openml. *arXiv*, 1911.02490, 2020.

[11] CodrinSocol. Cse3000-research-project repository, https://github.com/CodrinSocol/cse3000-research-project.

[12] Marco Loog, Tom Viering, Alexander Mey, Jesse H. Krijthe, and David M. J. Tax. A brief prehistory of double descent. *Proceedings of the National Academy of Sciences*, 117(20):10625–10626, may 2020.