A Survey of Learning Curves with Bad Behavior: or How More Data Need Not Lead to Better Performance

Marco Loog[1,2] and Tom J. Viering[1]

[1] Delft University of Technology, The Netherlands
[2] University of Copenhagen, Denmark
{m.loog,t.j.viering}@tudelft.nl

Abstract. Plotting a learner's generalization performance against the training set size results in a so-called learning curve. This tool, providing insight in the behavior of the learner, is also practically valuable for model selection, predicting the effect of more training data, and reducing the computational complexity of training. We set out to make the (ideal) learning curve concept precise and briefly discuss the aforementioned usages of such curves. The larger part of this survey's focus, however, is on learning curves that show that more data does not necessarily leads to better generalization performance. A result that seems surprising to many researchers in the field of artificial intelligence. We point out the significance of these findings and conclude our survey with an overview and discussion of open problems in this area that warrant further theoretical and empirical investigation.

Keywords: Learning curve · Sample complexity curve · Monotonicity · Learning theory.

1 Introduction

A curve that shows the dependence of a learner's generalization performance as a function of the training set size is known as a learning curve.[3] No matter the problem under consideration, it seems reasonable to require from a learner that its corresponding learning curve behaves monotonically, i.e., the learner delivers improved generalization performance with the availability of increasing amounts of training data. As any single draw of additional data from the problem at hand could be arbitrary bad by chance, this improved performance should be considered in expectation, or at least averaged over a large number of independent instantiations. In that respect, the requirement of performance improvement with more data may appear a mere theoretical one: in a real-world scenario, only a finite sample is available to train, tune, and validate a learner and we may not be able to effectively estimate its expected performance. Nevertheless, also from a practical point of view it is arguably of use to know that, upon collecting more data, one's learner is at least not expected to become worse.

In our experience, partly anecdotal of course, most researchers in AI indeed expect improved performance of their learner with more data. Actual evidence of this can be found in the literature. [71], for instance, states that the learning curve must start

[3] Arguably, the term sample complexity curve could be used as well (cf. [92]).

decreasing once the training set size becomes larger than the VC-dimension, while [14] claims that for many real-world problems its decay is monotonically. [79] and [27] state that it is expected that performance improves with more data, [80] takes it as conventional wisdom that the learning curve acts monotonically, and [7] considers this behavior to be widely accepted. [65] assumes well-behaved curves, which, as a rule, means they are smooth and monotonic [91]. Moreover, [3] explicitly states that the generalization error decreases as training set size increases. In the meta-review of our 2019 paper [46]—covered later on, this monotonic behavior is tellingly qualified as folk wisdom [50]. Indeed, it may come as a surprise to many researchers that a learner's performance can actually deteriorate with more training data. Even in expectation.

This review goes through different settings in which such unexpected phenomenon can occur and highlights various of its facets. We refer to these learners as badly behaved or ill-behaved and, by association, we refer to their learning curves in a likewise manner. Next to providing an in-depth review of the literature related to this particular subject, it also discusses our current understanding of this behavior and provides intriguing open questions and interesting directions for further research. To keep the survey self-contained, we spend some paragraphs on properly defining (idealized) learning curves. Subsequently, to also underline the practical relevance of the study of learning curves, we briefly cover its main usages.

A comprehensive review on learning curves can be found in [85]. The current work is inspired by Section 6 from this review and offers a fully revised extension and update of this important section. For a review complementary to [85], one that delves specifically into how such curves can provide insight into the learning phase and how they are important for meta-learning, we refer to [52]. Finally, please also check https://github.com/tomviering/ill-behaved-learning-curves.

1.1 Outline

In this work, we review learning curves in the context of standard supervised learning problems such as classification and regression. Section 2 makes precise what we mean by a learner, a learning problem, and the associated learning curve. To underline the practical use of learning curves, Section 3 sketches the insight into model selection they can give us, and how they are employed, for instance, in meta-learning and reducing the cost of labeling or computation. Sections 4 and 5 then follow with an overview of important cases of learning curves that do not behave well. The former section focuses on standard learning curves, while the latter considers a particular kind of learning curves that is often used in Bayesian analysis. These two section shows that our understanding of the behavior of learners is more limited than one might expect. Section 6 turns to some specific and general approaches to make learners (more) well-behaved, while Section 7 provides a discussion and concludes our review.

2 Problems, Learners, and Expected Curves

To formalize learning problems, learners, and learning curves, let u start by introducing $\mathcal{X}$ to denote a generic input space and $\mathcal{Y}$ an output space. Let S_N indicate a training set

of size N, which takes input-output pairs from $\mathcal{X} \times \mathcal{Y}$ and acts as input to our learning algorithm A, or learner for short. The N pairs (x_i, y_i) of the training set are i.i.d. samples from an unknown probability distribution P_{XY} over $\mathcal{X} \times \mathcal{Y}$, which defines our learning problem. A trained learner $A(S_N)$ delivers a hypothesis h from some hypothesis class $\mathcal{H}$, which contains all models that can, a priori, be returned by the learner A.

An example of a hypothesis class is the set of all linear models $\{h : x \mapsto a^T x + b \mid a \in \mathbb{R}^d, b \in \mathbb{R}\}$. In standard classification and regression, S_N consists of (x, y) pairs, where $x \in \mathbb{R}^d$ is the d-dimensional input vector (i.e., the features, measurements, or covariates) and y is the corresponding output (e.g. a discrete class label or continuous regression target).

When evaluating h on an input x, its prediction for the corresponding y is given by $\hat{y} = h(x) \in \mathcal{Y}$. The performance of a particular hypothesis h is measured by a loss function ℓ that compares y to $\hat{y}$. Examples are the squared loss for regression, where $\mathcal{Y} \subset \mathbb{R}$ and $\ell_{\text{squared}}(y, \hat{y}) = \ell(y, \hat{y}) = (y - \hat{y})^2$, and the zero-one loss for (binary) classification, where $\ell_{\text{0-1}}(y, \hat{y}) = \ell(y, \hat{y}) = \frac{1}{2}(1 - y\hat{y})$ when $\mathcal{Y} = \{-1, +1\}$.

The standard objective is that our hypothesis performs well on average on all new and unseen observations. Ideally, this is measured by the expected loss or risk R over the true distribution P_{XY}:

$$R(h) = \int \ell(y, h(x)) dP(x, y). \tag{1}$$

Here, as in most that follows, we omit the subscript XY to P_{XY}. Let us already note, in addition, that the evaluation loss employed in Equation (1) does not have to match the loss that is considered by the learner A at training time. In fact, in classification, the loss actually optimized is typically different from the 0-1 loss or accuracy that is considered at evaluation and test time (cf. [45]), an issue we return to in Section 4.2.

Now, an *individual* learning curve considers a single training set S_N for every N and calculates its corresponding risk $R(A(S_N))$ as a function of N. However, as noted already, a single S_N may deviate significantly from the expected behavior. Therefore, we are often interested in an averaging over many different random draws S_N. Ideally, we would like to evaluate the expectation

$$\bar{R}_N(A) = \mathop{\mathbb{E}}_{S_N \sim P^N} R(A(S_N)). \tag{2}$$

The plot of $\bar{R}_N(A)$ against the training set size N gives us the expected learning curve. From this point onward, when we mention the term "learning curve" without any further specifications, this is what is referred to.

The preceding learning curve is defined for a single learning problem P. Sometimes we wish to study how a model performs over a set or range of problems or, more generally, a full distribution $\mathcal{P}$ over such problems. The learning curve that considers such averaged performance is referred to as the problem-average (PA) learning curve:

$$\bar{R}_N^{\text{PA}}(A) = \mathop{\mathbb{E}}_{P \sim \mathcal{P}} \bar{R}_N(A). \tag{3}$$

The general term problem-average was coined in [14]. PA learning curves are particularly useful in the analysis of Bayesian approaches, where an assumed prior over possible

problems often arises naturally. This particular notation of learning curve is primarily employed in Section 5. In the Bayesian literature, the risk integrated over the prior is also called the Bayes risk, integrated risk, or preposterior risk [55, page 195]. The term preposterior signifies that, in principle, we can determine this quantity without observing any data, as the prior $\mathcal{P}$ prespecifies what data we expect to see.

3 General Practical Usage

Studies of learning curves have both practical and theoretical value. Here, we do not necessarily make a very strict separation between the two, though the primary emphasis in this survey is on the latter. This section, however, focuses in part of the former and briefly covers the current, most important uses of learning curves when it comes to applications. A more extensive and complete overview can be found in [52]. We return to the practical value of the theoretical findings presented in the discussion.

3.1 Model Evaluation

Machine learning as a field has shifted more and more to benchmarking learning algorithms. In the last 20 years, for instance, more than 5000 benchmark datasets have been created (see https://paperswithcode.com/ for an overview). These benchmarks are often set up as competitions [70] and investigate which algorithms are better or which novel procedure outperforms existing ones [63]. Typically, a single number, summarizing performance, is used as evaluation measure.

A recent meta-analysis indicates that the most popular measures are accuracy, the F-measure, and precision [6]. An essential issue these metrics ignore is that sample size can have a large influence on the relative ranking of different learning algorithms. In a plot of the learning curves of the different learners this would be visible as a crossing of their curves. In that light, it is beneficial if benchmarks consider multiple sample sizes, to get a better picture of the strengths and weaknesses of the approaches. The learning curve provides a concise picture of this sample-size dependent behavior.

Also using learning curves, [63] finds that, besides sample size, separability of the problem can be an indicator of which algorithm will dominate the other in terms of the learning curve. Beyond that, the learning curve, when plotted together with the training error of the algorithm can be used to detect whether a learner is overfitting [34,17,16,43].

3.2 Reduction of Data Collection Costs

When collecting data is difficult, time-consuming, or otherwise expensive, the possibility to accurately extrapolate a learner's learning curve can be useful. Extrapolations, which are typically base on some parametric learning curve model, give an impression beforehand of how many examples to collect to come to a specific performance [23]. As such, they also allows one to judge when data collection can be stopped. Examples of such practice can, for instance, be found in machine translation [36] and medical applications [54,29,21]. [39] quantifies potential savings assuming a fixed cost per collected sample and per generalization error. Extrapolating the learning curve using some labeled data,

the point at which it is not worth anymore to label more can be determined and data collection can be stopped.

3.3 Speeding Up Training, Tuning, and Selecting

Learning curves can be used to reduce computation time and memory with regards to training models, model selection and hyperparameter tuning.

Reference [65] speeds up training by so-called progressive sampling, using a learning curve to determine if less training data can reach adequate performance. If the slope of the curve becomes too flat, learning is stopped, making training potentially much faster. They recommended to use a geometric series for N to reduce computational complexity.

Several variations on progressive sampling exist. [35] proposes the notion of probably close enough where a power-law fit is used to determine if the learner is so-called epsilon-close to the asymptotic performance. [49] gives a rigorous decision theoretic treatment of the topic. By assigning costs to computation times and performances, they estimate what should be done to minimize the expected costs. Progressive sampling also has been adapted to the setting of active learning [81]. [40] combines meta-learning with progressive sampling to obtain a further speedup in the training phase.

Model selection can also be sped up. [41] compares the learners' initial learning curves to a database of learning curves to predict which of two classifiers will perform best on a new dataset. This can be used to avoid costly evaluations using cross validation. Moreover, [42] proposes an iterative process that predicts the required sample sizes, builds learning curves, and updates the performance estimates in order to compare two classifiers. [67] extends the technique to rank many machine learning models according to their predicted performance, tuning their approach to come to an acceptable answer in as little time as possible.

With regards to hyperparameter tuning, already in 1994, [11] devised an extrapolation scheme for learning curves, based on the fitting of power laws, to determine if it is worth to fully train a neural network. In the deep learning era, this has received renewed attention. [30] uses extrapolation to optimize hyperparameters. [31] takes this a step further and actually optimize several design choices, such as data augmentation. For such applications, a good learning curve model is essential.

4 Badly Behaving Learning Curves

It is important to understand that learning curves do not always behave well and that this is not necessarily an artifact of the finite sample or the way an experiment is designed. Deterioration with more training data can obviously occur when considering the individual curve $R(A(S_N))$ for a particular training set, because for every N, we can be unlucky with our draw S_N. That ill-behavior can also occur in expectation, i.e., for $\bar{R}_N(A)$, however, is less obvious.

The three subsections cover the three main settings in which bad behavior for standard learning curves can arise. Next to that, we point out what we currently understand of what are the essential differences between these settings. Subsequently, Section 5 turns to PA learning curves. An overview of the qualitative shapes of these learning curves, including the PA ones, is given in Figure 1.

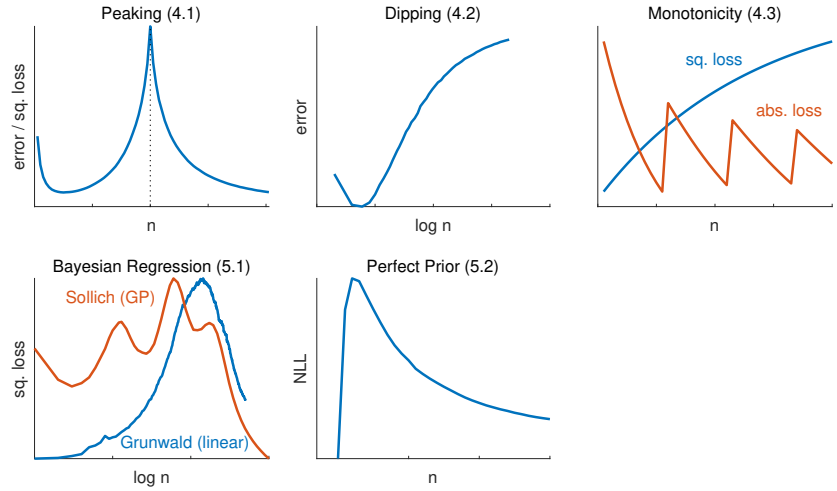

Fig. 1. Qualitative overview of various learning curve shapes placed in different categories with references to their corresponding subsections in the subtitle. All have the sample size n or $\log n$ on the horizontal axis. Dotted lines indicate the transition from under to overparametrized models. Abbreviations; error: classification error; sq. loss: squared loss; NLL: negative log likelihood; abs. loss: absolute loss.

4.1 Peaking and Double Descent

Probably the earliest abnormal behavior identified and studied among learning curves is so-called peaking. The term indicates that the learning curve takes on a maximum, typically in the form of a cusp. See Figure 1 (top left).

Unlike many other bad behaviors, peaking can occur in the realizable setting, i.e., where the learning model is actually well-specified, in the sense that the overall best-performing model is in the model class considered. Its cause seems related to instability of the model, which is maximal around the point where N hits the capacity of the learner. This phenomenon should not be confused with peaking for feature curves[4] (there are, however, direct connections between these viewpoints [18,47]). The feature-curve phenomenon has gained quite some renewed attention in recent years under the name double descent [4]. By now, the term (sample-wise) double descent has become a term for the peak in the learning curve for deep neural networks as well [57,56].

Peaking was first observed for pseudo-Fisher's linear discriminant (PFLD) in 1989 [83] and has been (re)considered at numerous occasions [47]. The PFLD is the classifier minimizing the squared loss, using minimum-norm or ridgeless linear regression based on the pseudo-inverse. PFLD often peaks at $d \approx N$, both for the squared loss and classification error. A first theoretical model explaining this behavior in the so-called thermodynamical limit is given in [59]. In such works, often originating from statistical

[4] A feature curve plots the performance of a learner against the varying number d of measurements it is trained on [32,33].

physics, the usual quantity of interest is $\alpha = \frac{d}{N}$ that controls the relative sizes for d and N going to infinity [61,90,20].

Reference [66] investigates peaking in the finite sample setting where each class is a Gaussian. They approximately decompose the generalization error in three terms. The first term measures the quality of the estimated means and the second the effect of reducing the dimensionality due to the pseudo-inverse. These terms reduce the error when N increases. The third term measures the quality of the estimated eigenvalues of the covariance matrix. This term increases the error when N increases, because more eigenvalues need to be estimated at the same time if N grows, reducing the quality of their overall estimation. These eigenvalues are often small and as the model depends on their inverse, small estimation errors can have a large effect, leading to a large instability [72] and peak in the learning curve around $N \approx d$. Using an analysis similar to [66], [38] study the peaking phenomenon in semi-supervised learning and show that unlabeled data can both mitigate or worsen it.

The work in [19] illustrates experimentally that SVMs may not suffer from peaking and Opper [60] presents a similar conclusion. For specific learning problems, [59] and [89] already give a theoretical underpinning for the absence of double descent for the perceptron of optimal (or maximal) stability, which is a classifier closely related to the SVM. [62] studies the behavior of the SVM in the thermodynamic limit, which does not show peaking either. [76] demonstrates, however, that double descent for feature curves can occur using the (squared) hinge loss, where the peak is typically located at a training sample size N that is larger than the dimensionality d.

Further insight into when peaking can occur may be obtained from [1] and [28]. These perform a rigorous analysis of the PFLD and standard least-squares regression using random matrix theory. Results should, however, be interpreted with care as these are typically derived in an asymptotic setting where both N and d (or some more appropriate measure of complexity) go to infinity, i.e., a setting similar to the earlier mentioned thermodynamic limit. [12] shows that a peak can occur both where the training set size N equals the input dimensionality d and when N matches the number of parameters of the learner. This depends on the learner's degree of nonlinearity. Multiple peaks are also possible for $N < d$ [58].

4.2 Dipping and Objective Mismatch

In peaking, the performance temporarily deteriorates, but recovers with the further increase of N. Optimal performance is typically achieved with an infinite sample size. In dipping, however, the learning curve may initially improve with more samples, but the performance eventually deteriorates and never recovers, even in the limit of infinite data [44]. Thus the best expected performance is reached at a finite training set size. See Figure 1 (top middle).

By constructing an explicit problem, [13] already showed that the nearest neighbor classifier is not always, what they refer to as, smart, meaning its learning curve can go up locally. A similar claim is made for kernel rules in Problems 6.14 and 6.15 from [13].

A one-dimensional toy problem for which many well-known linear classifiers dip is easily constructed [44]. In a different context, [5] provides an even stronger example where all linear classifiers optimizing a convex surrogate loss converge in the limit to the

worst possible classifier for which the error rate comes arbitrarily close to 1. Another example, Lemma 15.1 in [13], gives an very simple case of dipping for the likelihood where merely the estimation of the a prior class probability is considered.

What is essential for dipping to occur is that the the hypothesis class is misspecified and that the learner optimizes something else than the evaluation metric of the learning curve. Such objective misspecification is standard since many evaluation measures such as error rate, AUC, F-measure, and so on, are notoriously hard to optimize (see, e.g., [71]). In all of the above examples, the evaluation measure was the 0-1 loss, but the classifiers were optimized based on some standard surrogate measure like the hinge loss, the squared loss, or the likelihood.

Other works also show dipping of some sort. For example, [23] fits C4.5 to a synthetic dataset that has binary features for which the parity of all features determines the label. When fitting C4.5 the test error increases with the amount of training samples. They attribute this to the fact that the C4.5 is using a greedy approach to minimize the error and, as such, is closely related to objective misspecification. [9] shows a badly behaving curve for C4.5 that goes up. In addition, another 34 other curves were reported to not fit well using their parametric models, which may point to similar problems of curve increase. In [84], we find another potential example of dipping as, in Figure 6, the accuracy goes down with increasing sample sizes.

Anomaly or outlier detection using k-nearest neighbors (kNN) can shows dipping behavior as well [80]. Also here is a mismatch between the objective that is evaluated with, i.e., the AUC and kNN that does not optimize the AUC. [29] already shows kNN learning curves that deteriorate in terms of AUC in the standard supervised setting.

In active learning for classification, where the test error rate is often plotted against the size of the (actively sampled) training set, learning curves are regularly reported to dip [69,37]. That is, active learners provide optimal performance for a number of labeled samples that is smaller than the complete training set. Possibly, the active learner merely beats the inactive learner because it uses an objective that better matches the evaluation measure employed [48]. Finally, so-called negative transfer [88], as it occurs in transfer learning and domain adaptation, can be interpreted as dipping as well. In this case, more source data deteriorates performance on the target and the objective mismatch stems from the combined training from source and target data instead of the latter only.

4.3 Risk Monotonicity and Empirical Risk

Several novel examples of nonmonotonic behavior for density estimation, classification, and regression by means of standard empirical risk minimization (ERM) are shown in [46]. Similar to dipping, at some point, the squared loss increases with increasing N, but, in contrast with dipping, does eventually recover. See Figure 1 (top right).

These examples can neither be explained in terms of dipping nor in terms of peaking. Dipping is ruled out as, in ERM, the learner actually optimizes the loss that is used for evaluation. In addition, it is shown that learning problems can be constructed such that they act nonmonotonically at any sample size N. As nonmonotonicity can occur at small and large sample sizes, there is no link with the capacity of the learner and we can rule out an explanation in terms of peaking.

Reference [46] proofs nonmonotonicity for squared, absolute, and hinge loss. It demonstrates that likelihood estimators suffer the same deficiency. Two learners are reported that are provably monotonic: mean estimation based on the (negative) log-likelihood and the memorize algorithm from [71]. The latter algorithm does not really learn but outputs the majority voted classification label of each object if it has been seen before. Memorize is not PAC learnable [71,20], illustrating that monotonicity and PAC are, in that sense, different concepts.

The work in [86] and [46] shows experimentally that regularization can actually worsen the nonmonotonic behavior. This possibility had already been pointed out in [26] (see Subsection 5.2). Another experiment in [46] shows a surprisingly jagged learning curve for the absolute loss, Recent work [10] explains both behaviors and shows that the latter curves goes up and down perpetually.

5 PA Curves and Bad Behavior

Where standard learning curves deal with a single learning problem P, PA learning curves report an averaged learning curve, where the averaging is done according to a distribution $\mathcal{P}$ over learning problems. Using Bayesian inference, the PA learning curve is monotonic if the assumed prior over learning problems is correct, i.e., if the prior equals $\mathcal{P}$. This is a consequence of the total evidence theorem [68,24], which states, informally, that one obtains the maximum expected utility by taking into account all observations. As soon as there is any form of misspecification, also badly behaving PA curves can appear. See Figure 1 (bottom) for an overview of badly behaving PA curves. It is primarily the Bayesian regression setting that has been investigated.

5.1 Misspecified Bayesian Regression

Gaussian process regression, a particular instance of Bayesian regression, has been studied where the so-called teacher model provides the training data, while the student model learns, assuming a covariance or noise model different from the teacher. [74] analyzes the PA learning curve based on the eigenvalue decomposition of the covariances underlying the student and teacher model. The paper assumes both student and teacher use kernels with the same eigenfunctions but possibly differing eigenvalues. Subsequently, it considers various synthetic distributions for which the eigenfunctions and eigenvalues can be computed analytically and finds that for a uniform distribution on the vertices of a hypercube, multiple overfitting maxima and plateaus may be present in the learning curve. See Figure 1 (bottom left, red) for an example based on an actual experiment. For a uniform distribution in one dimension, the claim is that there may be arbitrarily many overfitting maxima .

The work in [25] shows that a (hierarchical) Bayesian linear regression model can give a broad peak in the learning curve of the squared risk. See Figure 1 (bottom left, blue) for one of their curves. One way this can happen is when the homogeneous noise assumption is violated, while the estimator is otherwise consistent. Specifically, let data be generated as follows. For each sample, a fair coin is flipped. Heads means the sample is generated according to the ground truth probabilistic model contained in the

hypothesis class. Misspecification happens when the coin comes up tails and a sample is generated in a fixed location without noise. The peak in the learning curve cannot be explained by dipping, peaking or known sensitivity of the squared loss to outliers. The peak in the learning curve is fairly broad and occurs in various experiments. As also no approximations are to blame, [25] concludes that Bayes' rule is really at fault as it cannot handle the misspecification.

5.2 The Perfect Prior

A monotonic PA curve does not rule out that the learning curve for individual problems can go up, even if the problem is well-specified. [26] offer an insightful example: consider a fair coin and let us estimate its probability p of heads using Bayes' rule. We measure performance using the negative log-likelihood on an unseen coin flip and adopt a uniform Beta(1,1) prior on p. This prior, i.e., without any training samples, already achieves the optimal loss since it assigns the same probability to heads and tails. After a single flip, $N = 1$, the posterior is updated and leads to a probabilities of $\frac{1}{3}$ or $\frac{2}{3}$ and the loss must increase. Eventually, with N going to infinity, the optimal loss is recovered, forming a bump in the learning curve. See Figure 1 (bottom middle) for the resulting curve. Note that this construction is rather versatile and can create nonmonotonic behavior for practically any Bayesian estimation task. In a similar way, any type of regularization can lead to comparable learning curve shapes, as was already mentioned in Subsection 4.3 (see also [86,46]).

A related example can be found in [2]. It shows that the posterior variance can also increase for a single problem, unless the likelihood belongs to the exponential family and a conjugate prior is used. Gaussian processes fall in this last class and, as such, their PA curve is monotone if there is no model misspecification.

6 Fixing Monotonicity

Some works set out to restore monotonicity in rather specific settings. Peaking of the PFLD, as defined and discussed in Subsection 4.1, can be avoided through regularization, though the tuning has to be done carefully [66,72,82,72]. Assuming the data is isotropic, [58] shows that peaking disappears for the optimal setting of the regularization parameter. Other, more heuristic solutions change the training procedure altogether, e.g., [15] uses an iterative procedure that decides which objects PFLD should be trained on. [73] adds copies of objects with noise or increases the dimensionality by adding noise features.

Reference [75] extends the study of Gaussian process regression from Subsection 5.1 and suggests to optimize hyperparameters such as length scale and noise level during learning based on evidence maximization. Among others, the paper finds that the earlier considered hypercube does not lead to arbitrary many overfitting maxima anymore. In fact, the learning curve becomes monotonic. Following the Bayesian regression analysis of [25], as covered in Subsection 5.1, the same work introduces a modified Bayes rule, where the likelihood is raised to some power. This parameter, however, cannot be learned in a Bayesian way, leading to their safe-Bayes approach. This technique alleviates the

broad peak in the learning curve and is empirically shown to make the curves more well-behaved.

A first attempt at a more generally applicable approach to restore monotonicity, though focusing on 0-1 loss, is made by [87]. They propose a wrapper that, with high probability, makes any classifier monotonic in terms of the the error rate. The main idea is to consider N as a variable over which model selection is performed. When N is increased, a model trained with more data is compared to the previously best model on validation data. Only if the new model is judged to be significantly better, the older model is discarded. If the original learning algorithm is consistent and if the size of the validation data grows, the resulting algorithm is consistent as well. It is empirically observed that the monotonic version may learn slower.

This idea is extended in [51], which proposes two algorithms that do not need to set aside validation data while guaranteeing monotonicity. To this end they assume that the Rademacher complexity of the hypothesis class composited with the loss is finite. This allows it to determine when to switch to a model trained with more data. In contrast to [87], it argues that the second algorithm does not learn slower, as its generalization bound coincides with a known lower bound of regular supervised learning.

Finally, [8] offers an analysis of the intricacies that can play a role in the potentially bad behavior of learning curves under the 0-1 loss. Based on these insights they formulate a general, sophisticated transformation applicable to a base learner that leads to monotonic behavior. The uniform error bounds they provide for their approach readily implies that every PAC learnable class admits a monotonic learner. Notably, their work took inspiration from [64] in which it is shown that universally consistent (binary) classifiers can be monotonic (notably, it gives an explicit method to construct a monotonic histogram classifiers). This last, actually rather recent result disproves a conjecture from [13] that posits that such universally consistent classifiers do not exist.

7 Discussion and Conclusions

In the past few decades, it are especially the more theoretical papers that have studied badly behaving learning curves. In addition, it seems that the monotonicity problem is currently gaining some traction from that side of the research spectrum as well. In our personal experience, many practitioners dismiss concerns over nonmonotonic behavior with the remark that such is only found in rather artificial settings. Often, the claim is made that such behavior does not occur in real-world applications or on real-world data.

But how do we know? Even if it is the case that this behavior does not happen in practice, this is still left to be demonstrated. Besides, the relatively artificial settings that some proofs may rely on, at the very least, show that there is something we do not understand. Therefore, these examples should be taken as a starting point to improve our understanding of the learning process. We should try to understand, both in practice and theory, how bad this behavior can get, how it can be mitigated, and what we gain or loose by this. Such understanding may certainly be of direct relevance if we want to exploit the usage of learning curves, as covered in Section 3, to its full extent. In all, empirical and theoretical research challenges abound (see also [8,46,53] for instance).

7.1 Nonmonotonicity in Other AI Disciplines

This survey's focus is on standard supervised learning, but what can we say about more complicated learning settings? What to expect, for instance, when dealing with adversaries, when we act in an environment that is only partially observable, or if we are dealing with multiple agents? Arguably, we can expect even more exotic learning behavior in general and, therefore, also new settings in which learning can uniquely fail.

7.2 Meta-Learning

We think better understanding is needed with respect to the occurrence of peaking, dipping, and otherwise nonmonotonic behavior. As indicated already, the simple fact is that, at this point, we do not know what role these phenomena play in real-world problems. Now that many benchmark datasets are readily available, this issue can be studied more rigorously. Properly summarizing and openly sharing learning curve data can further support this research. Automated techniques may then be developed to find curious learning curve phenomena and possibly predict them.

Given the success of meta-learning for curve extrapolation and model selection [40,78] this seems a promising possibility. Such meta-learning studies on large amounts of datasets could, in addition, shed more light on what determines the parameters of learning curve models, a topic that has been investigated relatively little up to now. Predicting these parameters robustly from very few points along the learning curve will prove valuable for virtually all applications.

An initial study into the behavior of learning curves on a newly developed database can be found in [53]. The database makes learning curves for 246 datasets and 20 classifiers publicly available and presents some basic and preliminary findings. Among them, the most interesting from a practical point of view may be that peaking does not seem to occur very often. In all, this database offers ample opportunity to investigate learning curves extensively in an empirical way.

7.3 Two Theoretical Questions

There are two rather specific theoretical questions that we like readers to consider.

The first one asks whether maximum likelihood estimators for well-specified models behave monotonically. Likelihood estimation, being a century-old, classical technique [22,77], has been heavily studied, both theoretically and empirically. In much of the theory developed, the assumption that one is dealing with a correctly specified model is common, but we are not aware of any results that demonstrate that better models are obtained with more data. The question is interesting for the likelihood exactly because this estimator has been extensively studied already and still plays a central role in statistics and abutting fields as well.

The second question is broader: for standard regression problems, among the consistent learners are there monotonic ones? [8] demonstrates that for the classification setting monotonic learners can be designed and they ask the same question for more general loss function. They indicate, however, that one should probably restrict oneself to bounded losses. A derivative question could therefore be: for unbounded losses, can we show that for every (consistent) learners a problem exists on which this learner behaves badly.

7.4 Concluding

It is striking that there is still so much that we actually do not understand about learning curve behavior and, as such, learning itself. Even some of the most simple settings elude us. Most theoretical results are restricted to relatively basic learners, while the empirical research that has been carried out is rather limited in scope. We identified some specific challenges in the foregoing, but we are convinced that many more interesting problems can be discovered. We are convinced that a deeper understanding of badly behaving learning curves will also turn out to be practically beneficial. To us, however, a sufficiently valid reason to investigate it should be to quench one's scientific curiosity.

References

1. Advani, M.S., Saxe, A.M.: High-dimensional dynamics of generalization error in neural networks. arXiv:1710.03667 (2017)
2. Al-Saleh, M.F., Masoud, F.A.: A note on the posterior expected loss as a measure of accuracy in bayesian methods. Applied mathematics and computation **134**(2-3), 507–514 (2003)
3. Amari, S.i., Fujita, N., Shinomoto, S.: Four types of learning curves. Neural Computation **4**(4), 605–618 (1992)
4. Belkin, M., Hsu, D., Ma, S., Mandal, S.: Reconciling modern machine-learning practice and the classical bias–variance trade-off. PNAS **116**(32), 15849–15854 (2019)
5. Ben-david, S., Loker, D., Srebro, N., Sridharan, K.: Minimizing the misclassification error rate using a surrogate convex loss. ICML pp. 1863–1870 (2012)
6. Blagec, K., Dorffner, G., Moradi, M., Samwald, M.: A critical analysis of metrics used for measuring progress in artificial intelligence. arXiv:2008.02577 (2020)
7. Boonyanunta, N., Zeephongsekul, P.: Predicting the relationship between the size of training sample and the predictive power of classifiers. In: KES. pp. 529–535. Springer (2004)
8. Bousquet, O.J., Daniely, A., Kaplan, H., Mansour, Y., Moran, S., Stemmer, U.: Monotone learning. In: Conference on Learning Theory. pp. 842–866 (2022)
9. Brumen, B., Rozman, I., Heričko, M., Černezel, A., Hölbl, M.: Best-fit learning curve model for the c4. 5 algorithm. Informatica **25**(3), 385–399 (2014)
10. Chen, Z., Loog, M., Krijthe, J.H.: Explaining two strange learning curves. In: BNAIC/BeNeLearn. p. accepted (2022)
11. Cortes, C., Jackel, L.D., Solla, S.A., Vapnik, V., Denker, J.S.: Learning curves: Asymptotic values and rate of convergence. In: NeurIPS. pp. 327–334 (1994)
12. d'Ascoli, S., Sagun, L., Biroli, G.: Triple descent and the two kinds of overfitting: Where & why do they appear? arXiv:2006.03509 (2020)
13. Devroye, L., Györfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition, vol. 31. Springer, New York, NY, USA (1996)
14. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. John Wiley & Sons (2012)
15. Duin, R.P.W.: Small sample size generalization. 9th Scandinavian Conference on Image Analysis pp. 1–8 (1995)
16. Duin, R.P.W., Pekalska, E.M.: Pattern Recognition: Introduction and Terminology. 37 Steps (2016)
17. Duin, R.P.W., Tax, D.M.J.: Statistical pattern recognition. In: Chen, C.H., Wang, P.S.P. (eds.) Handbook Of Pattern Recognition And Computer Vision, pp. 3–24. World Scientific (2005)
18. Duin, R.P.W.: Classifiers in almost empty spaces. In: ICPR. vol. 2, pp. 1–7. IEEE (2000)
19. Duin, R.P.W.: Classifiers in almost empty spaces. In: ICPR. pp. 1–7 (2000)

20. Engel, A., Van den Broeck, C.: Statistical mechanics of learning. Cambridge University Press (2001)
21. Figueroa, R.L., Zeng-Treitler, Q., Kandula, S., Ngo, L.H.: Predicting sample size required for classification performance. BMC med. inf. and decision making **12**(1), 8 (2012)
22. Fisher, R.A.: An absolute criterion for fitting frequency curves. Messenger of Mathematics **41**, 155–160 (1912)
23. Frey, L.J., Fisher, D.H.: Modeling decision tree performance with the power law. In: AISTATS (1999)
24. Good, I.J.: On the principle of total evidence. The British Journal for the Philosophy of Science **17**(4), 319–321 (1967)
25. Grünwald, P., van Ommen, T.: Inconsistency of Bayesian inference for misspecified linear models, and a proposal for repairing it. Bayesian Analysis **12**(4), 1069–1103 (2017)
26. Grünwald, P.D., Kotłowski, W.: Bounds on individual risk for log-loss predictors. JMLR **19**, 813–816 (2011)
27. Gu, B., Hu, F., Liu, H.: Modelling classification performance for large data sets. In: International Conference on Web-Age Information Management. pp. 317–328. Springer (2001)
28. Hastie, T., Montanari, A., Rosset, S., Tibshirani, R.J.: Surprises in high-dimensional ridgeless least squares interpolation. arXiv:1903.08560 (2019)
29. Hess, K.R., Wei, C.: Learning curves in classification with microarray data. Seminars in Oncology **37**(1), 65–68 (2010)
30. Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M.M.A., Yang, Y., Zhou, Y.: Deep Learning Scaling is Predictable, Empirically. arXiv:1712.00409 (2017)
31. Hoiem, D., Gupta, T., Li, Z., Shlapentokh-Rothman, M.: Learning curves for analysis of deep networks. In: International Conference on Machine Learning. pp. 4287–4296. PMLR (2021)
32. Hughes, G.: On the mean accuracy of statistical pattern recognizers. IEEE Trans. IT **14**(1), 55–63 (1968)
33. Jain, A., Chandrasekaran, B.: Dimensionality and Sample Size Considerations in Pattern Recognition Practice. Handbook of Statistics **2**, 835–855 (1982)
34. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. TPAMI **22**(1), 4–37 (2000)
35. John, G.H., Langley, P.: Static versus dynamic sampling for data mining. In: KDD. vol. 96, pp. 367–370 (1996)
36. Kolachina, P., Cancedda, N., Dymetman, M., Venkatapathy, S.: Prediction of Learning Curves in Machine Translation. In: ACL. pp. 22–30. Jeju Island, Korea (2012)
37. Konyushkova, K., Sznitman, R., Fua, P.: Introducing geometry in active learning for image segmentation. In: CVPR. pp. 2974–2982 (2015)
38. Krijthe, J.H., Loog, M.: The peaking phenomenon in semi-supervised learning. In: S+SSPR). pp. 299–309. Springer (2016)
39. Last, M.: Predicting and optimizing classifier utility with the power law. In: ICDMW. pp. 219–224. IEEE (2007)
40. Leite, R., Brazdil, P.: Improving progressive sampling via meta-learning on learning curves. In: ECML. pp. 250–261. Springer (2004)
41. Leite, R., Brazdil, P.: Predicting Relative Performance of Classifiers from Samples. In: ICML. pp. 497—-503. Bonn, Germany (2005)
42. Leite, R., Brazdil, P.: An iterative process for building learning curves and predicting relative performance of classifiers. In: Portuguese Conference on Artificial Intelligence. pp. 87–98. Springer (2007)
43. Loog, M.: Supervised classification: Quite a brief overview. In: Machine Learning Techniques for Space Weather, pp. 113–145. Elsevier (2018)

44. Loog, M., Duin, R.P.W.: The dipping phenomenon. In: S+SSPR. pp. 310–317. Hiroshima, Japan (2012)
45. Loog, M., Krijthe, J.H., Jensen, A.C.: On measuring and quantifying performance: error rates, surrogate loss, and an example in semi-supervised learning. In: Handbook of Pattern Recognition and Computer Vision, pp. 53–68. World Scientific (2016)
46. Loog, M., Viering, T., Mey, A.: Minimizers of the empirical risk and risk monotonicity. In: NeurISP. pp. 7478–7487 (2019)
47. Loog, M., Viering, T., Mey, A., Krijthe, J.H., Tax, D.M.J.: A brief prehistory of double descent. PNAS **117**(20), 10625–10626 (2020)
48. Loog, M., Yang, Y.: An empirical investigation into the inconsistency of sequential active learning. In: ICPR. pp. 210–215. IEEE (2016)
49. Meek, C., Thiesson, B., Heckerman, D.: The learning-curve sampling method applied to model-based clustering. JMLR **2**(Feb), 397–418 (2002)
50. MetaReview: Reviews: Minimizers of the empirical risk and risk monotonicity. https://papers.nips.cc/paper/2019/file/0f9cafd014db7a619ddb4276af0d692c-MetaReview.html (2019)
51. Mhammedi, Z.: Risk monotonicity in statistical learning. NeurIPS **34** (2021)
52. Mohr, F., van Rijn, J.N.: Learning curves for decision making in supervised machine learning–a survey. arXiv:2201.12150 (2022)
53. Mohr, F., Viering, T.J., Loog, M., van Rijn, J.N.: LCDB 1.0: An extensive learning curves database for classification tasks. In: Machine Learning and Knowledge Discovery in Databases, ECMLPKDD. p. accepted. Lecture Notes in Computer Science, Springer (2022)
54. Mukherjee, S., Tamayo, P., Rogers, S., Rifkin, R., Engle, A., Campbell, C., Golub, T.R., Mesirov, J.P.: Estimating dataset size requirements for classifying dna microarray data. Journal of computational biology **10**(2), 119–142 (2003)
55. Murphy, K.P.: Machine learning: a probabilistic perspective. MIT press (2012)
56. Nakkiran, P.: More data can hurt for linear regression: Sample-wise double descent. arXiv:1912.07242 (2019)
57. Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., Sutskever, I.: Deep double descent: Where bigger models and more data hurt. In: ICLR (2019)
58. Nakkiran, P., Venkat, P., Kakade, S., Ma, T.: Optimal regularization can mitigate double descent. arXiv:2003.01897 (2020)
59. Opper, M., Kinzel, W., Kleinz, J., Nehl, R.: On the ability of the optimal perceptron to generalise. Journal of Physics A: Mathematical and General **23**(11), L581 (1990)
60. Opper, M.: Learning to generalize. Frontiers of Life **3**(part 2), 763–775 (2001)
61. Opper, M., Haussler, D.: Calculation of the learning curve of bayes optimal classification algorithm for learning a perceptron with noise. In: COLT. vol. 91, pp. 75–87 (1991)
62. Opper, M., Urbanczik, R.: Universal learning curves of support vector machines. Physical Review Letters **86**(19), 4410 (2001)
63. Perlich, C., Provost, F., Simonoff, J.S.: Tree Induction vs. Logistic Regression: A Learning-Curve Analysis. JMLR **4**(1), 211–255 (2003)
64. Pestov, V.: A universally consistent learning rule with a universally monotone error. arXiv: 2108.09733 (2021)
65. Provost, F., Jensen, D., Oates, T.: Efficient progressive sampling. In: ACM SIGKDD. pp. 23–32 (1999)
66. Raudys, S., Duin, R.P.W.: Expected classification error of the Fisher linear classifier with pseudo-inverse covariance matrix. Pattern Recognition Letters **19**(5-6), 385–392 (1998)
67. van Rijn, J.N., Abdulrahman, S.M., Brazdil, P., Vanschoren, J.: Fast algorithm selection using learning curves. In: LNCS. vol. 9385, pp. 298–309. Springer Verlag (oct 2015)
68. Savage, L.J.: The foundations of statistics. John Wiley & Sons, Inc. (1954)
69. Schohn, G., Cohn, D.: Less is more: Active learning with support vector machines. In: ICML. pp. 839–846 (2000)

70. Sculley, D., Snoek, J., Wiltschko, A., Rahimi, A.: Winner's curse? on pace, progress, and empirical rigor. In: ICLR (2018)
71. Shalev-Shwartz, S., Ben-David, S.: Understanding machine learning: From theory to algorithms. Cambridge university press (2014)
72. Skurichina, M., Duin, R.P.W.: Stabilizing classifiers for very small sample sizes. In: ICPR. vol. 2, pp. 891–896. IEEE (1996)
73. Skurichina, M., Duin, R.P.W.: Regularisation of Linear Classifiers by Adding Redundant Features. Pattern Anal. Appl. **2**(1), 44–52 (1999)
74. Sollich, P.: Gaussian Process Regression with Mismatched Models. In: NeurIPS. pp. 519–526 (2002)
75. Sollich, P.: Can Gaussian Process Regression Be Made Robust Against Model Mismatch? In: Deterministic and Statistical Methods in Machine Learning. pp. 199–210 (2004)
76. Spigler, S., Geiger, M., d'Ascoli, S., Sagun, L., Biroli, G., Wyart, M.: A jamming transition from under-to over-parametrization affects generalization in deep learning. Journal of Physics A **52**(47), 474001 (2019)
77. Stigler, S.M.: The epic story of maximum likelihood. Statistical Science **22**(4), 598–620 (2007)
78. Strang, B., van der Putten, P., van Rijn, J.N., Hutter, F.: Don't rule out simple models prematurely: a large scale benchmark comparing linear and non-linear classifiers in openml. In: IDA. pp. 303–315. Springer (2018)
79. Tax, D.M.J., Duin, R.P.W.: Learning curves for the analysis of multiple instance classifiers. In: S+SSPR. pp. 724–733. Springer (2008)
80. Ting, K.M., Washio, T., Wells, J.R., Aryal, S.: Defying the gravity of learning curve: a characteristic of nearest neighbour anomaly detectors. Machine Learning **106**(1), 55–91 (2017)
81. Tomanek, K., Hahn, U.: Approximating learning curves for active-learning-driven annotation. In: LREC. vol. 8, pp. 1319–1324 (2008)
82. Tresp, V.: The equivalence between row and column linear regression. Tech. rep., Siemens (2002)
83. Vallet, F., Cailton, J.G., Refregier, P.: Linear and nonlinear extension of the pseudo-inverse solution for learning boolean functions. EPL (Europhysics Letters) **9**(4), 315 (1989)
84. Vanschoren, J., Pfahringer, B., Holmes, G.: Learning from the past with experiment databases. In: Pacific Rim International Conference on Artificial Intelligence. pp. 485–496. Springer (2008)
85. Viering, T., Loog, M.: The shape of learning curves: a review. arXiv: 2103.10948 (2021)
86. Viering, T., Mey, A., Loog, M.: Open problem: Monotonicity of learning. In: Conference on Learning Theory. pp. 3198–3201 (2019)
87. Viering, T.J., Mey, A., Loog, M.: Making learners (more) monotone. In: IDA. pp. 535–547. Springer (2020)
88. Wang, Z., Dai, Z., Póczos, B., Carbonell, J.: Characterizing and avoiding negative transfer. In: CVPR. pp. 11293–11302 (2019)
89. Watkin, T.L.H., Raut, A., Biehl, M.: The Statistical Mechanics of Learning a Rule. Reviews of Modern Physics **65**(2), 499–556 (1993). https://doi.org/10.1103/RevModPhys.65.499
90. Watkin, T.L., Rau, A., Biehl, M.: The statistical mechanics of learning a rule. Rev. of Modern Physics **65**(2), 499 (1993)
91. Weiss, G.M., Battistin, A.: Generating well-behaved learning curves: An empirical study. In: ICDATA (2014)
92. Zubek, J., Plewczynski, D.M.: Complexity curve: a graphical measure of data complexity and classifier performance. PeerJ Computer Science **2**, e76 (2016)