The Shape of Learning Curves

A Review

Viering, Tom; Loog, Marco

DOI
10.1109/TPAMI.2022.3220744

Publication date
2023

Document Version
Final published version

Published in
IEEE Transactions on Pattern Analysis and Machine Intelligence

The Shape of Learning Curves: A Review

Tom Viering and Marco Loog

Abstract—Learning curves provide insight into the dependence of a learner's generalization performance on the training set size. This important tool can be used for model selection, to predict the effect of more training data, and to reduce the computational complexity of model training and hyperparameter tuning. This review recounts the origins of the term, provides a formal definition of the learning curve, and briefly covers basics such as its estimation. Our main contribution is a comprehensive overview of the literature regarding the shape of learning curves. We discuss empirical and theoretical evidence that supports well-behaved curves that often have the shape of a power law or an exponential. We consider the learning curves of Gaussian processes, the complex shapes they can display, and the factors influencing them. We draw specific attention to examples of learning curves that are ill-behaved, showing worse learning performance with more training data. To wrap up, we point out various open problems that warrant deeper empirical and theoretical investigation. All in all, our review underscores that learning curves are surprisingly diverse and no universal model can be identified.

Index Terms—Learning curve, training set size, supervised learning, classification, regression

1 Introduction

The more often we are confronted with a particular problem to solve, the better we typically get at it. The same goes for machines. A *learning curve* is an important, graphical representation that can provide insight into such learning behavior by plotting generalization performance against the number of training examples. Two example curves are shown in Fig. 1.

We review learning curves in the context of standard supervised learning problems such as classification and regression. The primary focus is on the shapes that learning curves can take on. We make a distinction between well-behaved learning curves that show improved performance with more data and ill-behaved learning curves that, perhaps surprisingly, do not. We discuss theoretical and empirical evidence in favor of different shapes, underlying assumptions made, how knowledge about those shapes can be exploited, and further results of interest. In addition, we provide the necessary background to interpret and use learning curves as well as a comprehensive overview of the important research directions.

1.1 Outline

The next section starts off with a definition of learning curves and discusses how to estimate them in practice. It also briefly considers so-called *feature curves*, which offer a complementary view. Section 3 covers the use of learning curves, such as 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. Section 4 considers evidence supporting well-behaved learning curves: curves that generally show improved performance with more training data. We review the parametric models that have been studied empirically and cover the theoretical findings in favor of some of these. Many of the more theoretical results in the literature have been derived particularly for Gaussian process regression as its learning curve is more readily analyzed analytically. Section 5 is primarily devoted to those specific results. Section 6 then follows with an overview of important cases of learning curves that do not behave well and considers possible causes. We believe that especially this section shows that our understanding of the behavior of learners is more limited than one might expect. Section 7 provides an extensive discussion. It also concludes our review. The remainder of the current section goes into the origins and meanings of the term "learning curve" and its synonyms.

1.2 Learning Curve Origins and Meanings

With his 1885 book *Über das Gedächtnis* [1], Ebbinghaus is generally considered to be the first to employ and qualitatively describe learning curves. These curves report on the number of repetitions it takes a human to perfectly memorize an increasing number of meaningless syllables. Such learning curves have found widespread application for predicting human productivity, and are the focus of previous surveys on learning curves [2], [3].

While Ebbinghaus is the originator of the learning curve concept, it should be noted that the type of curves from [1], [2], [3] are importantly different from the curves central to this review. In the machine learning setting, we typically care about the generalization performance, i.e., the learner's performance on *new and unseen* data. Instead, Ebbinghaus's subject goal is to recite exactly that string of syllables that has been provided to him. Similarly, studies of human productivity focus on the speed and cost of repetitions of the same task. This means in a way that the performance on the

- *Tom Viering is with the Delft University of Technology, 2628 CD Delft, The Netherlands. E-mail: t.j.viering@tudelft.nl.*
- *Marco Loog is with the Delft University of Technology, 2628 CD Delft, The Netherlands, and also with the University of Copenhagen, 1165 København, Denmark. E-mail: m.loog@tudelft.nl.*

Manuscript received 30 March 2021; revised 22 August 2022; accepted 24 October 2022. Date of publication 9 November 2022; date of current version 5 May 2023.
(Corresponding author: Tom Viering.)
Recommended for acceptance by S. Zilles.
Digital Object Identifier no. 10.1109/TPAMI.2022.3220744

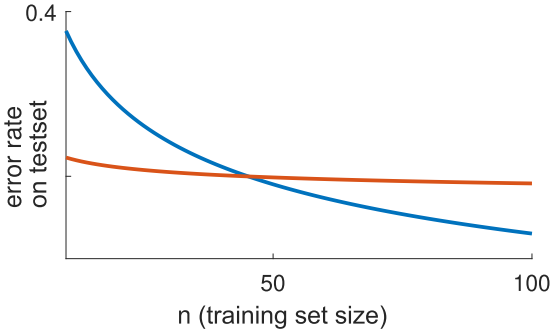

Fig. 1. Two crossing learning curves. Red starts at a lower error, while blue reaches a lower error rate given enough data. One number summaries cannot characterize such behaviors. For more details see Sections 2.4 and 3.1.

training set is considered. This measure is also called the resubstitution or apparent error in classification [4], [5], [6]. Indeed, as most often is the case for the training error as well, memorization performance gets worse as the amount of training data increases, i.e., it is more difficult to memorize an increasing number of syllables.

The term learning curve considered in this review is different from the curve that displays the training error—or the value of any objective function—as a function of the number of epochs or iterations used for optimization. Especially in the neural network literature, this is what the learning curve often signifies[1] [7], [8], [9]. What it has in common with those of Ebbinghaus is that the performance is plotted against the number of times that (part of) the data has been revisited, which corresponds directly to the number of repetitions in [1]. These curves, used to monitor the optimality of a learner in the training phase, are also referred to as *training curves* and this terminology can be traced back to [10]. We use training curve exclusively to refer to these curves that visualize performance during training. Many researchers and practitioners, however, use the term learning curve instead of training curve [8], which, at times, may lead to confusion.

In the machine learning literature synonyms for learning curve are *error curve*, *experience curve*, *improvement curve* and *generalization curve* [8], [11], [12]. Improvement curve can be traced back to a 1897 study, on learning the telegraphic language [13]. Generalization curve was first used in machine learning in 1990 [11], [12]. Decades earlier, the term was already used to indicate the plot of the intensity of an animal's response against stimulus size [14]. Learning curve or, rather, its German equivalent was not used as an actual term in the original work of Ebbinghaus [1]. The English variant seems to appear 18 years later, in 1903 [15]. *Lehrkurve* follows a year after [16].

We traced back the first mention of learning curve in connection to learning machines to a discussion in an 1957 issue of *Philosophy* [17]. A year later, Rosenblatt, in his famous 1958 work [18], uses learning curves in the analysis of his perceptron. Following this review's terminology, he uses this term to refer to a training curve. Foley [19] was possibly the first to use a learning curve, as it is defined in this review, in an experimental setting such as is common nowadays. The theoretical study of learning curves for supervised learners dates back at least to 1965 [20].

2 Definition, Estimation, Feature Curves

This section makes the notion of a learning curve more precise and describes how learning curves can be estimated from data. We give some recommendations when it comes to plotting learning curves and summarizing them. Finally, feature curves offer a view on learners that is complementary to that of learning curves. These and combined learning-feature curves are covered at the end of this section.

2.1 Definition of Learning Curves

Let S_n indicate a training set of size n, which acts as input to some learning algorithm A. 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 class label or regression target). $\mathcal{X}$ denotes the input space and $\mathcal{Y}$ the output space. The (x, y) pairs of the training set are i.i.d. samples of an unknown probability distribution P_{XY} over $\mathcal{X} \times \mathcal{Y}$. Predictors h come from the hypothesis class $\mathcal{H}$, which contains all models that can be returned by the learner A. An example of a hypothesis class is the set of all linear models $\{h : x \mapsto a^T x + b | a \in \mathbb{R}^d, b \in \mathbb{R}\}$.

When h is evaluated on a sample x, its prediction for the corresponding y is given by $\hat{y} = h(x)$. 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 $L_{\text{sq}}(y, \hat{y}) = (y - \hat{y})^2$ and the zero-one loss for (binary) classification $L_{01}(y, \hat{y}) = \frac{1}{2}(1 - y\hat{y})$ when $\mathcal{Y} = \{-1, +1\}$.

The typical goal is that our predictor 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}

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

Here, as in most that follows, we omit the subscript XY.

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. Note that training sets can be partially ordered, meaning $S_{n-1} \subset S_n$. However, a single S_n may deviate significantly from the expected behavior. Therefore, we are often interested in an averaging over many different sets S_n, and ideally 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 talk about *the* learning curve, this is what is meant.

The preceding learning curve is defined for a single problem P. Sometimes we wish to study how a model performs over a range of problems or, more generally, a full distribution $\mathcal{P}$ over problems. The learning curve that considers

1. It should be noted that such plots have to be interpreted with care, as the number of optimization steps or similar quantities can dependent on the training set size, the batch size, etc. Precise specifications are key therefore.

such averaged performance is referred to as the problem-average (PA) learning curve

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

The general term problem-average was coined in [21]. PA learning curves make sense for Bayesian approaches in particular, where an assumed prior over possible problems often arises naturally. As GP's are Bayesian, their PA curve is frequently studied, see Section 5. The risk integrated over the prior, in the Bayesian literature, is also called the Bayes risk, integrated risk, or preposterior risk [22, page 195]. The term preposterior signifies that, in principle, we can determine this quantity without observing any data.

In semi-supervised learning [23] and active learning [24], it can be of additional interest to study the learning behavior as a function of the number of *unlabeled* and *actively selected* samples, respectively.

2.2 Estimating Learning Curves

In practice, we merely have a finite sample from P and we cannot measure $R(h)$ or consider all possible training sets sampled from P. We can only get an estimate of the learning curve. Popular approaches are to use a hold-out dataset or k-fold cross validation for this [25], [26], [27], [28] as also apparent from the Weka documentation [29] and Scikit-learn documentation [30]. Using cross validation, k folds are generated from the dataset. For each split, a training set and a test set are formed. The size of the training set S_n is varied over a range of values by removing samples from the originally formed training set. For each size the learning algorithm is trained and the performance is measured on the test fold. The process is repeated for all k folds, leading to k individual learning curves. The final estimate is their average. The variance of the estimated curve can be reduced by carrying out the k-fold cross validation multiple times [31]. This is done, for instance, in [25], [26], [27], [32].

Using cross validation to estimate the learning curve has some drawbacks. For one, when making the training set smaller, not using the discarded samples for testing seems wasteful. Also note that the training fold size limits the range of the estimated learning curve, especially if k is small. Directly taking a random training set of the preferred size and leaving the remainder as a test set can be a good alternative. This can be repeated to come to an averaged learning curve. This recipe—employed, for instance in [32], [33]—allows for easy use of any range of n, leaving no sample unused. Note that the test risks are not independent for this approach. Alternatively, drawing samples from the data with replacement can also be used to come to to variable sized training sets (e.g., similar to bootstrapping [34], [35]). For classification, often a learning curve is made with stratified training sets to reduce the variance further.

A learner A often depends on hyperparameters. Tuning once and fixing them to create the learning curve will result in suboptimal curves, since the optimal hyperparameter values can strongly depend on n. Therefore, ideally, the learner should internally use cross validation (or the like) to tune hyperparameters for each training set it receives.

An altogether different way to learning curve estimation is to assume an underlying parametric model for the learning curve and fit this to the learning curves estimates obtained via approaches described previously. The approach is not widespread under practitioners, but is largely confined to the research work that studies and exploits the general shape of learning curves (see Section 4.1).

Finally, note that all of the foregoing pertains to PA learning curves as well. In that setting, we may occasionally be able to exploit the special structure of the assumed problem prior. This is the case, for instance, with Gaussian process regression, where problem averages can sometimes be computed with no additional cost (Section 5).

2.3 Plotting Considerations

When plotting the learning curve, it can be useful to consider logarithmic axes. Plotting n linearly may mask small but non-trivial gains [36]. Also from a computational standpoint it often makes sense to have n traverse a logarithmic scale [37] (see also Section 3.2). A log-log or semi-log plot can be useful if we expect the learning curve to display power-law or exponential behavior (Section 4), as the curve becomes straight in that case. In such a plot, it can also be easier to discern small deviation from such behavior. Finally, it is common to use error bars to indicate the standard deviation over the folds to give an estimate of the variability of the individual curves.

2.4 Summarizing Learning Curves

It may be useful at times, to summarize learning curves into a single number. A popular metric to that end is the area under the learning curve (AULC) [38], [39], [40] (see [41] for early use). To compute this metric, one needs to settle at a number of sample sizes. One then averages the performance at all those sample sizes to get to the area of the learning curve. The AULC thus makes the curious assumption that all sample sizes are equally likely.

Important information can get lost when summarizing. The measure is, for instance, not able to distinguish between two methods whose learning curves cross (Fig. 1), i.e., where the one method is better in the small sample regime, while the other is in the large sample setting. Others have proposed to report the asymptotic value of the learning curve and the number of samples to reach it [42] or the exponent of the power-law fit [43].

Depending on the application at hand, particularly in view of the large diversity in learning curve shapes, these summaries are likely inadequate. Recently, [44] suggested to summarize using *all* fit parameters, which should suffice for most applications. However, we want to emphasize one should try multiple parametric models and report the parameters of the best fit *and* the fit quality (e.g., MSE).

2.5 Fitting Considerations

Popular parametric forms for fitting learning curves are given in Table 1 and their performance is discussed in Section 4.1. Here, we make a few general remarks.

Some works [45], [46], [47], [48] seem to perform simple least squares fitting on log-values in order to fit power laws or exponentials. If, however, one would like to find the optimal parameters in the original space in terms of the mean squared error, non-linear curve fitting is necessary (for

TABLE 1
Parametric Learning Curve Models

Reference	Formula	Used in
POW2	an^{-b}	**[46]*** [47], [48], [49]
POW3	$an^{-b}+c$	**[49]*[50]*** [45], [102]
LOG2	$-a\log(n)+c$	**[47]*** [46], [48], [49], [102]
EXP3	$a\exp(-bn)+c$	**[102]*** [50]
EXP2	$a\exp(-bn)$	[46], [47], [48]
LIN2	$-an+b$	[46], [47], [48], [102]
VAP3	$\exp(a+b/n+c\log(n))$	[49]
MMF4	$(ab+cn^d)/(b+n^d)$	[49]
WBL4	$c-b\exp(-an^d)$	[49]
EXP4	$c-\exp(-an^\alpha+b)$	[50]
EXPP3	$c-\exp((n-b)^\alpha)$	[50]
POW4	$c-(-an+b)^{-\alpha}$	[50]
ILOG2	$c-(a/\log(n))$	[50]
EXPD3	$c-(c-a)\exp(-bn)$	[103]

Note that some curves model performance increase rather than loss decease. The first column gives the abbreviation used and the number of parameters. Bold and asterisk marks the paper this model came out as the best fit.

example using Levenberg–Marquardt [49]). Then again, assuming Gaussian errors in the original space may be questionable, since the loss is typically non-negative. Therefore, confidence intervals, hypothesis tests, and p-values should be interpreted with care.

For many problems, one should consider a model that allows for nonzero asymptotic error (like POW3 and EXP3 in Table 1). Also, often the goal is to interpolate or extrapolate to previously *unseen* training set sizes. This is a generalization task and we have to deal with the problems that this may entail, such as overfitting to learning curve data [49], [50]. Thus, learning curve data should also be split in train and test sets for a fair evaluation.

2.6 Feature Curves and Complexity

The word feature refers to the d measurements that constitutes an input vector x. A feature curve is obtained by plotting the performance of a machine learning algorithm A against the varying number of measurements it is trained on [51], [52]. To be a bit more specific, let $\varsigma_{d'}$ be a procedure that selects d' of the original d features, hence reducing the dimensionality of the data to d'. A feature curve is then obtained by plotting $\bar{R}(A(\varsigma_{d'}(S_n)))$ versus d', while n is now the quantity that is fixed. As such, it gives a view complementary to the learning curve.

The selection of d' features as carried out by means of $\varsigma_{d'}$, can be performed in various ways. Sometimes features have some sort of inherent ordering. In other cases PCA or feature selection can provide such ordering. When no ordering can be assumed, $\varsigma_{d'}$ samples d' random features from the data—possibly even with replacement. In this scenario, it is sensible to construct a large number of different feature curves, based on different random subsets, and report their average as the final curve.

Typically, an increase in the number of input dimensions means that the complexity of the learner also increases. As such it can, more generally, be of interest to plot performance against any actual, approximate, or substitute measure of the complexity. Instead of changing the dimensionality, changing parameters of the learner, such as the smoothness of a kernel or the amount of filters in a CNN, can also be used to vary the complexity to obtain similar curves [51], [53], [54], [55], [56]. Such curves are sometimes called *complexity curves* [55], *parameter curves* [57] or *generalization curves* [58].

One of the better known phenomena of feature curves is the so-called peaking phenomenon (also the peak effect, peaking or Hughes phenomenon [4], [59], [60], [61]). The peaking phenomenon of feature curves is related to the curse of dimensionality and illustrates that adding features may actually degrade the performance of a classifier, leading to the classical U-shaped feature curve. Behavior more complex than the simple U-shape has been observed as well [62], [63], [64] and has recently been referred to as double descent [65] (see Fig. 2a). This is closely related to peaking of (ill-behaving) learning curves (Section 6.2).

2.7 Combined Feature and Learning Curves

Generally, the performance of a learner A is not influenced independently by the the number of training samples n and the number of features d. In fact, several theoretical works suggest that the fraction $\alpha=\frac{d}{n}$ is essential (see, e.g., [66], [67], [68]). Because of the feature-sample interaction, it can be insightful to plot multiple learning curves for a variable number of input dimensions or multiple feature curves for different training set sizes. Another option is to make a 3D plot—e.g., a surface plot—or a 2D image of the performance against both n and d directly. Instead of the number of features any other complexity measure can be used.

Fig. 2a shows such a plot for pseudo-Fisher's linear discriminant (PFLD; see Section 6.2) when varying both n and d'. Taking a section of this surface, we obtain either a learning curve in Fig. 2b (horizontal, fixed d) or a feature curve in Fig. 2c (vertical, fixed n). Fig. 2a gives a more complete view of the interaction between n and d. In Fig. 2d we can see that the optimal d' depends on n. Likewise, for this model, there is an optimal n for each d', i.e., the largest possible value of n is not necessarily the best.

Duin [63] is possibly the first to include such 3D plot, though already since the work of Hughes [51], figures that combine multiple learning or feature curves have been used, see for example [69], [70], and for combinations with complexity curves see [53], [54], [71], [72]. More recently, [56], [73] gives 2D images of the performance of deep neural networks as a function of both model and training set size.

3 General Practical Usage

The study of learning curves has both practical and research/theoretical value. While we do not necessarily aim to make a very strict separation between the two, more emphasis is put on the latter further on. This section focuses on part of the former and covers the current, most important uses of learning curves when it comes to applications, i.e., model selection and extrapolation to reduce data collection and computational costs. For a valuable, complementary review that delves specifically into how learning curves are deployed for decision making, we refer to [74].

3.1 Better Model Selection and Crossing Curves

Machine learning as a field has shifted more and more to benchmarking learning algorithms, e.g., in the last 20 years,

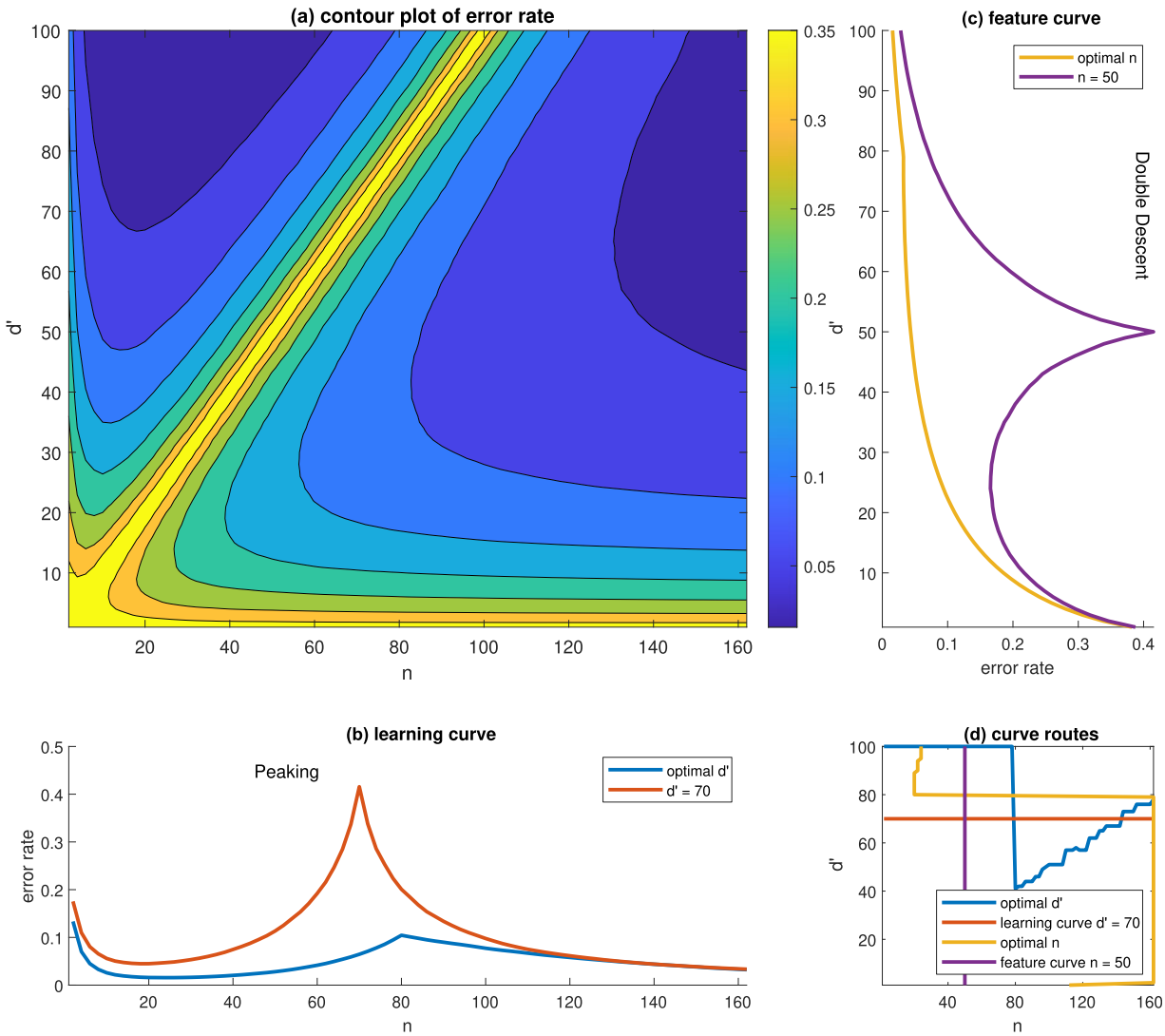

Fig. 2. (a) Image of the error for varying sample size n and dimensionality d', for the pseudo-Fisher learning algorithm (without intercept) on a toy dataset with two Gaussian classes having identity covariance matrices. Their means are a distance of 6 apart in 100 dimensions, with every dimension adding a same amount to the overall distance. (b) By fixing d' and varying n, i.e., taking a horizontal section, we obtain a learning curve (red). We also show the curve where d' is chosen optimally for each n (blue). (c) This plot is rotated by 90 degrees. By fixing n and varying d', i.e., a vertical section, we obtain a feature curve (purple). We also show the curve where the optimal n is chosen for each d' (yellow). (d) Here we show the paths taken through the image to obtain the curves. The learning curve and feature curves are the straight lines, while the curves that optimize n or d' take other paths. The largest n and d are not always optimal. We note that Section 6 covers learning curves that do not necessarily improve with more data. Section 6.2 discusses peaking (displayed here).

more than 2,000 benchmark datasets have been created (see [75] for an overview). These benchmarks are often set up as competitions [76] and investigate which algorithms are better or which novel procedure outperforms existing ones [36]. 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 [77]. 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 learning curves this would be visible as a crossing of the different curves (see Fig. 1). 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.

Crossing curves have also been referred to as the scissor effect and have been investigated since the 1970s [70], [78], [79] (see also [80]). Contrary to such evidence, there are papers that suggest that learning curves do not cross [81], [82]. [36] calls into question the claim of [81] as the datasets may be too small to find the crossing point. The latter claim by [82] is specific to deep learning, where, perhaps, exceptions may occur that are currently not understood.

Perhaps the most convincing evidence for crossing curves is given in [36]. The paper compares logistic regression and decision trees on 36 datasets. In 15 of the 36 cases the learning curves cross. This may not always be apparent, however, as large sample sizes may be needed to find the crossing point. In the paper, the complex model (decision tree) is better for large sample sizes, while the simple model (logistic regression) is better for small ones. Similarly, Strang et al. [83] performed a large-scale meta-learning study on 294 datasets,

comparing linear versus nonlinear models, and found evidence that non-linear methods are better when datasets are large. Ng and Jordan [28] found, when comparing naive Bayes to logistic regression, that in 7 out of 15 datasets considered the learning curves crossed. [45], [84], [85], [86] provide further evidence.

Also using learning curves, [36] 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 [4], [5], [57], [87]. Besides sample size, dimensionality seems also an important factor to determine whether linear or non-linear methods will dominate [83]. To that end, learning curves combined with feature curves may offer further insights.

3.2 Extrapolation to Reduce Data Collection Costs

When collecting data is time-consuming, difficult, or otherwise expensive the possibility to accurately extrapolate a learner's learning curve can be useful. Extrapolations (typically base on some parametric learning curve model, see Section 4.1) give an impression beforehand of how many examples to collect to come to a specific performance and allows one to judge when data collection can be stopped [46]. Examples of such practice can, for instance, be found in machine translation [50] and medical applications [25], [26], [32]. Last [48] 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 data can be determined and data collection stopped.

Determining a minimal sample size is called sample size determination. For usual statistical procedures this is done through what is called a power calculation [88]. For classifiers, sample size determination using a power calculation is unfeasible according to [25], [26]. John and Langley [89] illustrate that a power calculations that ignores the machine learning model indeed fails to accurately predict the minimal sample size.

Sample size determination can be combined with meta-learning, which uses experience on previous datasets to inform decisions on new datasets. To that end, [90] builds a small learning curve on a new and unseen dataset and compares it to a database of previously collected learning curves to determine the minimum sample size.

3.3 Speeding Up Training and Tuning

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

To speed up training, so-called progressive sampling [37] uses 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. It is recommended to use a geometric series for n to reduce computational complexity.

Several variations on progressive sampling exist. John and Langley [89] 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. [91] 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 [92]. [90] combines meta-learning with progressive sampling to obtain a further speedup.

To speed up model selection, [93] compares 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. Leite and Brazdil [94] propose an iterative process that predicts the required sample sizes, builds learning curves, and updates the performance estimates in order to compare two classifiers. Rijn et al. [95] extend 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. [96] does not resort to meta-learning, and instead uses partial learning curves to speed up model selection. This simpler approach can also already lead to significantly improved run times.

With regards to hyperparameter tuning, already in 1994 Cortes et al. [45] 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. [97] extrapolates the learning curve to optimize hyperparameters. [44] takes this a step further and actually optimize several design choices, such as data augmentation. One obstacle for such applications is that it remains unclear when the learning curve has which shape.

4 Well-Behaved Learning Curves

We deem a learning curve well-behaved if it shows improved performance with increased training sample sizes, i.e., $\bar{R}_n(A) \geq \bar{R}_{n+1}(A)$ for all n. In slightly different settings, learners that satisfy this property are called smart [99, page 106] and monotone [99].

There is both experimental and theoretical evidence for well-behaved curves. In the case of large deep learning models, empirical evidence often seems to point to power-law behavior specifically. For problems with binary features and decision trees, exponential curves cannot be ruled out, however. There generally is no definitive empirical evidence that power laws models are essentially better. Theory does often point to power-law behavior, but showcases exponentials as well. The most promising theoretical works characterizing the shape are [100] and [101]. Theoretical PA curves are provably monotone, given the problem is well-specified and a Bayesian approaches is used. They favor exponential and power-law shapes as well.

4.1 In-Depth Empirical Studies of Parametric Fits

Various works have studied the fitting of empirical learning curves and found that they typically can be modelled with function classes depending on few parameters. Table 1 provides a comprehensive overview of the parametric models studied in machine learning, models for human learning in [3] may offer further candidates. Two of the primary

objectives in studies of fitting learning curves are how well a model interpolates an empirical learning curve over an observed range of training set sizes and how well it can extrapolate beyond that range.

Studies investigating these parametric forms often find the power law with offset (POW3 in the table) to offer a good fit. The offset makes sure that a non-zero asymptotic error can be properly modeled, which seems a necessity in any challenging real-world setting. Surprisingly, even though Frey and Fisher [46] do not include this offset c and use POW2, they find for decision trees that on 12 out of 14 datasets they consider, the power law fits best. Gu et al. [49] extend this work to datasets of larger sizes and, next to decision trees, also uses logistic regression as a learner. They use an offset in their power law and consider other functional forms, notably, VAP, MMF, and WBL. For extrapolation, the power law with bias performed the best overall. Also Last [48] trains decision trees and finds the power law to perform best. Kolachina et al. [50] give learning curves for machine translation in terms of BLUE score for 30 different settings. Their study considers several parametric forms (see Table 1) but also they find that the power law is to be preferred.

Boonyanunta and Zeephongsekul [103] perform no quantitative comparison and instead postulate that a differential equation models learning curves, leading them to an exponential form, indicated by EXPD in the table. [104] empirically finds exponential behavior of the learning curve for a perceptron trained with backpropagation on a toy problem with binary inputs, but neither performs an in depth comparison. In addition, the experimental setup is not described precisely enough: for example, it is not clear how step sizes are tuned or if early stopping is used.

Three studies find more compelling evidence for deviations from the power law. The first, [105], can be seen as an in-depth extension of [104]. They train neural networks on four synthetic datasets and compare the learning curves using the r^2 goodness of fit. Two synthetic problems are linearly separable, the others require a hidden layer, and all can be modeled perfectly by the network. Whether a problem was linearly separable or not doesn't matter for the shape of the curve. For the two problems involving binary features exponential learning curves were found, whereas problems with real-valued features a power law gave the best fit. However, they also note, that it is not always clear that one fit is significantly better than the other.

The second study, [47], evaluates a diverse set of learners on four datasets and shows the logarithm (LOG2) provides the best fit. The author has some reservations about the results and mentions that the fit focuses on the first part of the curve as a reason that the power law may seem to perform worse, besides that POW3 was also not included. Given that many performance measures are bounded, parametric models that increase or decrease beyond any limit should eventually give an arbitrarily bad fit for increasing n. As such, LOG2 is anyway suspect.

The third study, [102], considers only the performance of the fit on training data, e.g., already observed learning curves points. They *do* only use learning curve models with a maximum of three parameters. They employs a total of 121 datasets and use C4.5 for learning. In 86 cases, the learning curve shape fits well with one of the functional forms, in 64 cases EXP3 gave the lowest overall MSE, in 13 it was POW3. A signed rank test shows that the exponential outperforms all others models mentioned for [102] in Table 1. Concluding, the first and last work provide strong evidence against the power law in certain settings.

Finally, in what is probably the most extensive learning curve study to date [106], the authors find power laws to often provide good fits though not necessarily preferable to some of the other models. In particular, for larger data sets, all four 4-parameter models from Table 1 often perform on par, but MMF4 and WBL4 outperform EXP4 and POW4 if enough curve data is used for fitting.

4.2 Power Laws and Eye-Balling Deep Net Results

Studies of learning curves of deep neural networks mostly claim to find power-law behavior. However, initial works offer no quantitative comparisons to other parametric forms and only consider plots of the results, calling into question the reliability of such claims. Later contributions find power-law behavior over many orders of magnitude of data, offering somewhat stronger empirical evidence.

Sun et al. [107] state that their mean average precision performance on a large-scale internal Google image dataset increases logarithmically in dataset size. This claim is called into question by [97] and we would agree: there seems to be little reason to believe this increase follows a logarithm. Like for [47], that also finds that the logarithm fit well, one should remark that the performance in terms of mean average precision is always bounded from above and therefore the log model should eventually break. As opposed to [107], [108] does observe diminishing returns in a similar large-scale setting. [109] also studies large-scale image classification and find learning curves that level off more clearly in terms of accuracy over orders of magnitudes. They presume that this is due to the maximum accuracy being reached, but note that this cannot explain the observations on all datasets. In the absence of any quantitative analysis, these results are possibly not more than suggestive of power-law behavior.

Hestness et al. [97] observe power laws over multiple orders of magnitude of training set sizes for a broad range of domains: machine translation (error rate), language modeling (cross entropy), image recognition (top-1 and top-5 error rate, cross entropy) and speech recognition (error rate). Its exponent was found to be between -0.07 and -0.35 and mostly depends on the domain. Architecture and optimizer primarily determine the multiplicative constant. For small sample sizes, however, the power law supposedly does not hold anymore, as the neural network converges to a random guessing solutions. In addition, one should note that that the work does not consider any competing models nor does it provide an analysis of the goodness of fit. Moreover, to uncover the power law, significant tuning of the hyperparameters and model size per sample size is necessary, otherwise deviations occur. Interestingly, [44] investigates robust curve fitting for the error rate using the power law with offset and find exponents of size -0.3 and -0.7, thus of larger magnitude than [97].

Kaplan et al. [110] and Rosenfeld et al. [73] also rely on power laws for image classification and natural language processing and remarks similar to those we made about [97]

apply. [110] finds that if the model size and computation time are increased together with the sample size, that the learning curve has this particular behavior. If either is too small this pattern disappears. They find that the test loss also behaves as a power law as function of model size and training time and that the training loss can also be modeled in this way. [73] reports that the test loss behaves as a power law in sample size when model size is fixed and vice versa. Both provide models of the generalization error that can be used to extrapolate performances to unseen sample and model sizes and that may reduce the amount of tuning required to get to optimal learning curves. Some recent additions to this quickly expanding field are [111] and [112].

On the whole, given the level of validation provided in these works (e.g., no statistical tests or alternative parametric models, etc.), power laws can currently not be considered more than a good candidate learning curve model.

4.3 What Determines the Parameters of the Fit?

Next to the parametric form as such, researchers have investigated what determines the parameters of the fits. [25], [45], and [44] provide evidence that the asymptotic value of the power law and its exponent could be related. Singh [47] investigates the relation between dataset and classifier but does not find any effect. They do find that the neural networks and the support vector machine (SVM) are more often well-described by a power law and that decision trees are best predicted by a logarithmic model. Only a limited number of datasets and models was tested however. Perlich et al. [36] find that the Bayes error is indicative of whether the curves of decision trees and logistic regression will cross or not. In case the Bayes error is small, decision trees will often be superior for large sample sizes. All in all, there are few results of this type and most are quite preliminary.

4.4 Shape Depends on Hypothesis Class

Turning to theoretical evidence, results from learning theory, especially in the form of Probably Approximately Correct (PAC) bounds [113], [114], have been referred to to justify power-law shapes in both the separable and non-separable case [44]. The PAC model is, however, pessimistic since it considers the performance on a worst-case distribution P, while the behavior on the actual fixed P can be much more favorable (see, for instance, [115], [116], [117], [118], [119]). Even more problematic, the worst-case distribution considered by PAC is *not* fixed, and can depend on the training size n, while for the learning curve P is fixed. Thus the actual curve can decrease much faster than the bound (e.g., exponential) [100]. The curve only has to be beneath the bound but can still also show strange behavior (wiggling).

A more fitting approach, pioneered by Schuurmans [120], [121] and inspired by the findings in [105], does characterize the shape of the learning curve for a fixed unknown P. This has also been investigated in [122], [123] for specific learners. Recently, Bousquet et al. [100] gave a full characterization of all learning curve shapes for the realizable case and optimal learners. This last work shows that optimal learners can have only three shapes: an exponential shape, a power-law shape, or a learning curve that converges arbitrarily slow. The optimal shape is determined by novel properties of the hypothesis class (not the VC dimension). The result of arbitrarily slow learning is, partially, a refinement of the no free lunch theorem of [99, Section 7.2] and concerns, for example, hypothesis classes that encompass all measurable functions. These results are a strengthening of a much earlier observation by Cover [124].

4.5 Shape Depends on the Problem

There are a number of papers that find evidence for the power-law shape under various assumptions. We also discuss work that finds exponential curves.

A first provably exponential learning curve can be traced back to the famous work on one-nearest neighbor (1NN) classifier [125] by Cover and Hart. They point out that, in a two-class problem, if the classes are far enough apart, 1NN only misclassifies samples from one class if all n training samples are from the other. Given equal priors, one can show $\bar{R}_n(A_{1\text{NN}}) = 2^{-n}$. This suggests that if a problem is well-separated, classifiers can converge exponentially fast. Peterson [126] studied 1NN for a two-class problem where P_X is uniform on [0,1] and $P_{Y|X}$ equals x for one of the two classes. In that case of class overlap the curve equals $\frac{1}{3} + \frac{3n+5}{2(n+1)(n+2)(n+3)}$, thus we have a slower power law.

Amari [127], [128] studies (PA) learning curves for a basic algorithm, the Gibbs learning algorithm, in terms of the cross entropy. The latter is equal to the logistic loss in the two-class setting and underlies logistic regression. [127] refers to it as the entropic error or entropy loss. The Gibbs algorithm A_G is a stochastic learner that assumes a prior over all models considered and, at test time, samples from the posterior defined through the training data, to come to a prediction [68], [129]. Separable data is assumed and the model samples are therefore taken from version space [68].

For A_G, the expected cross entropy, $\bar{R}^{\text{CE}}$, can be shown to decompose using a property of the conditional probability [127]. Let $p(S_n)$ be the probability of selecting a classifier from the prior that classifies all samples in S_n correctly. Assume, in addition, that $S_n \subset S_{n+1}$. Then, while for general losses we end up with an expectation that is generally hard to analyze [130], the expected cross entropy simplifies into the difference of two expectations [127]

$$\bar{R}_n^{\text{CE}}(A_G) = E_{S_n \sim P} \log p(S_n) - E_{S_{n+1} \sim P} \log p(S_{n+1}). \quad (4)$$

Under some additional assumptions, which ensure that the prior is not singular, the behavior asymptotic in n can be fully characterized. Amari [127] then demonstrates that

$$\bar{R}_n^{\text{CE}}(A_G) \approx \frac{d}{n} + o\left(\frac{1}{n}\right), \quad (5)$$

where d is the number of parameters.

Amari and Murata [131] extend this work and consider labels generated by a noisy process, allowing for class overlap. Besides Gibbs, they study algorithms based on maximum likelihood estimation and the Bayes posterior distribution. They find for Bayes and maximum likelihood that the entropic generalization error behaves as $H_0 + \frac{d}{2n}$, while the training error behaves as $H_0 - \frac{d}{2n}$, where H_0 is the best possible cross entropy loss. For Gibbs, the error behaves as $H_0 + \frac{d}{n}$, and the

training error as H_0. In case of model mismatch, the maximum likelihood solution can also be analyzed. In that setting, the number of parameters d changes to a quantity indicating the number of effective parameters and H_0 becomes the loss of the model closest to the groundtruth density in terms of KL-divergence.

In a similar vein, Amari et al. [130] analyze the 0-1 loss, i.e., the error rate, under a so-called annealed approximation [67], [68], which approximates the aforementioned hard-to-analyze risk. Four settings are considered, two of which are similar to those in [127], [128], [131]. The variation in them stems from differences in assumptions about how the labeling is realized, ranging from a unique, completely deterministic labeling function to multiple, stochastic labelings. Possibly the most interesting result is for the realizable case where multiple parameter settings give the correct outcome or, more precisely, where this set has nonzero measure. In that case, the asymptotic behavior is described as a power law with an exponent of -2. Note that this is essentially faster than what the typical PAC bounds can provide, which are exponents of -1 and $-\frac{1}{2}$ (a discussion of those results is postponed to Section 7.3). This possibility of a more rich analysis is sometimes mentioned as one of the reasons for studying learning curves [68], [118], [132].

For some settings, exact results can be obtained. If one considers a 2D input space where the marginal P_X is a Gaussian distribution with mean zero and identity covariance, and one assumes a uniform prior over the true linear labeling function, the PA curve for the zero one loss can exactly be computed to be of the form $\frac{2}{3n}$, while, the annealed approximation gives $\frac{1}{n}$ [130].

Schwartz et al. [133] use tools similar to Amari's to study the realizable case where all variables (features and labels) are binary and Bayes rule is used. Under their approximations, the PA curve can be completely determined from a histogram of generalization errors of models sampled from the prior. For large sample sizes, the theory predicts that the learning curve actually has an exponential shape. The exponent depends on the gap in generalization error between the best and second best model. In the limit of a gap of zero size, the shape reverts to a power law. The same technique is used to study learning curves that can plateau before dropping off a second time [134]. [135] proposes extensions dealing with label noise. [132] casts some doubt on the accuracy of these approximations and the predictions of the theory of Schwartz et al. indeed deviate quite a bit from their simulations on toy data. Recently, [101] presented a simple classification setup on a countably infinite input space for which the Bayes error equals zero. The work shows that the learning curve behavior crucially depends on the distribution assumed over the discrete feature space and most often shows power-law behavior. More importantly, it demonstrates how exponents other than the expected -1 or $-\frac{1}{2}$ can emerge, which could explain the diverse exponents mentioned in Section 4.2.

4.6 Monotone Shape if Well-Specified (PA)

The PA learning curve is monotone if the prior and likelihood model are correct and Bayesian inference is employed. This is a consequence of the total evidence theorem [136], [137], [138]. It states, informally, that one obtains the maximum expected utility by taking into account all observations. However, a monotone PA curve does not rule out that the learning curve for individual problems can go up, even if the problem is well-specified, as the work covered in Section 6.7 points out. Thus, if we only evaluate in terms of the learning curve of a single problem, using all data is not always the rational strategy.

It may be of interest to note here that, next to Bayes' rule, there are other ways of consistently updating one's belief—in particular, so-called probability kinematics—that allow for an alternative decision theoretic setting in which a total evidence theorem also applies [139].

Of course, in reality, our model probably has some misspecification, which is a situation that has been considered for Gaussian process models and Bayesian linear regression. Some of the unexpected behavior this can lead to is covered in Section 6.5 for PA curves and Section 6.6 for the regular learning curve. Next, however, we cover further results in the well-behaved setting. We do this, in fact, specifically for Gaussian processes.

5 Gaussian Process Learning Curves

In Gaussian process (GP) regression [140], it is especially the PA learning curve (Section 2.1) for the squared loss, under the assumption of a Gaussian likelihood, that has been studied extensively. A reason for this is that many calculations simplify in this setting. We cover various approximations, bounds, and ways to compute the PA curve. We also discuss assumptions and their corresponding learning curve shapes and cover the factors that affect the shape. It appears difficult to say something universally about the shape, besides that it cannot decrease faster than $O(n^{-1})$ asymptotically. The fact that these PA learning curves are monotone in the correctly specified setting can, however, be exploited to derive generalization bounds. We briefly cover those as well. This section is limited to correctly specified, well-behaved curves, Section 6.5 is devoted to ill-behaved learning curves for misspecified GPs.

The main quantity that is visualized in the PA learning curve of GP regression is the Bayes risk or, equivalently, the problem averaged squared error. In the well-specified case, this is equal to the posterior variance σ_*^2 [141, Equation (2.26)] [141],

$$\sigma_*^2 = k(x_*, x_*) - k_*^T (K(X_n, X_n) + \sigma^2 I)^{-1} k_*. \tag{6}$$

Here k is the covariance function or kernel of the GP, $K(X_n, X_n)$ is the $n \times n$ kernel matrix of the input training set X_n, where $K^{lm}(X_n, X_n) = k(x_l, x_m)$. Similarly, k_* is the vector where the lth component is $k(x_l, x_*)$. Finally, σ is the noise level assumed in the Gaussian likelihood.

The foregoing basically states that the averaging over all possible problems $\mathcal{P}$ (as defined by the GP prior) and all possible output training samples is already taken care of by Equation (6). Exploiting this equality, what is then left to do to get to the PA learning curve is an averaging over all test points x_* according to their marginal P_X and an averaging over all possible input training samples X_n.

Finally, for this section, it turns out to be convenient to introduce a notion of PA learning curves in which only the

averaging over different input training sets $X_n \in \mathcal{X}^n$ has *not* been carried out yet. We denote this by $R^{\mathrm{PA}}(X_n)$ and we will not mention the learning algorithm A (GP regression).

5.1 Fixed Training Set and Asymptotic Value

The asymptotic value of the learning curve and the value of the PA curve for a fixed training set can be expressed in terms of the eigendecomposition of the covariance function. This decomposition is often used when approximating or bounding GPs' learning curves [141], [142], [143].

With P_X be the marginal density and $k(x, x')$ the covariance function. The eigendecomposition constitutes all eigenfunctions ϕ_i and eigenvalues λ_i that solve for

$$\int k(x, x')\phi(x)dP_X(x) = \lambda\phi(x'). \tag{7}$$

Usually, the eigenfunctions are chosen so that

$$\int \phi_i(x)\phi_j(x)dP_X(x) = \delta_{ij}, \tag{8}$$

where δ_{ij} is Kronecker's delta [140]. The eigenvalues are non-negative and assumed sorted from large to small ($\lambda_1 \geq \lambda_2 \geq \ldots$). Depending on P_X and the covariance function k, the spectrum of eigenvalues may be either degenerate, meaning there may be finite nonzero eigenvalues, or nondegenerate in which case there are infinite nonzero eigenvalues. For some, analytical solutions to the eigenvalues and functions are known, e.g., for the squared exponential covariance function and Gaussian P_X. For other cases the eigenfunctions and eigenvalues can be approximated [140].

Now, take Λ the diagonal matrix with λ_i on the diagonal and let $\Phi_{li} = \phi_i(x_l)$, where each x_l comes from the training matrix X_n. The dependence on the training set is indicated by $\Phi = \Phi(X_n)$. For a fixed training set, the squared loss over all problems can then be written as

$$R^{\mathrm{PA}}(X_n) = \mathrm{Tr}(\Lambda^{-1} + \sigma^{-2}\Phi(X_n)^T\Phi(X_n))^{-1}, \tag{9}$$

which is exact [142]. The only remaining average to compute to come to a PA learning curve is with respect to X_n. This last average is typically impossible to calculate analytically (see [141, p. 168] for an exception). This leads one to consider the approximations and bounds covered in Section 5.3.

The asymptotic value of the PA learning curve is [144]

$$\bar{R}^{\mathrm{PA}}_{\infty} = \sum_{i=1}^{\infty} \frac{\lambda_i\tau}{\tau + \lambda_i}, \tag{10}$$

where convergence is in probability and almost sure for degenerate kernels. Moreover, it is assumed that $\sigma^2 = n\tau$ for a constant τ, which means that the noise level grows with the sample size. The assumption seems largely a technical one and Le Gratiet and Garnier [144] claim this assumption can still be reasonable in specific settings.

5.2 Two Regimes, Effect of Length Scale on Shape

For many covariance functions (or kernels), there is a characteristic length scale l that determines the distance in feature space one after which the regression function can change significantly. Williams and Vivarelli [141] make the qualitative observation that this lead the PA curve to often have two regimes: initial and asymptotic. If the length scale of the GP is not too large, the initial decrease of the learning curve is approximately linear in n (initial regime). They explain that, initially, the training points are far apart, and thus they can almost be considered as being observed in isolation. Therefore, each training point reduces the posterior variance by the same amount, and thus the decrease is initially linear. However, when n is larger, training points get closer together and their reductions in posterior variance interact. Then reduction is not linear anymore because an additional point will add less information than the previous points. Thus there is an effect of diminishing returns and the decrease gets slower: the asymptotic regime [141], [142].

The smaller the length scale, the longer the initial linear trend, because points are comparatively further apart [141], [142]. Williams and Vivarelli further find that changing the length scale effectively rescales the amount of training data in the learning curve for a uniform marginal P_X. Furthermore, they find that a higher noise level and smoother covariance functions results in earlier reaching of the assymptotic regime for the uniform distribution. It remains unclear how these results generalize to other marginal distributions.

Sollich and Halees [142] note that in the asymptotic regime, the noise level has a large influence on the shape of the curve since here the error is reduced by averaging out noise, while in the non-asymptotic regime the noise level hardly plays a role. They always assume that the noise level is much smaller than the prior variance (the expected fluctuations of the function before observing any samples). Under that assumption, they compare the error of the GP with the noise level to determine the regime: if the error is smaller than the noise, it indicates that one is reaching the asymptotic regime.

5.3 Approximations and Bounds

A simple approximation to evaluate the expectation with respect to the training set from Equation (9) is to replace the matrix $\Phi(X_n)^T\Phi(X_n)$ by its expected value nI (this is the expected value due to Equation (8)). This results in

$$\bar{R}^{\mathrm{PA}}_n \approx \sum_{i=1}^{\infty} \frac{\lambda_i\sigma^2}{\sigma^2 + n\lambda_i}. \tag{11}$$

The approximation, which should be compared to Equation (10), is shown to be an upper bound on the training loss and a lower bound on the PA curve [145]. From asymptotic arguments, it can be concluded that the PA curve cannot decrease faster than $\frac{1}{n}$ for large n [141, p. 160]. Since asymptotically the training and test error coincide, Opper and Vivarelli [145] expect this approximation to give the correct asymptotic value, which, indeed, is the case [144].

The previous approximation works well for the asymptotic regime but for non-asymptotic cases it is not accurate. Sollich [146] aims to approximate the learning curve in such a way that it can characterizes both regimes well. To that end, he, also in collaboration with Halees [142], introduces three approximations based on a study of how the matrix inverse in Equation (9) changes when n is increased. He derives recurrent relations that can be solved and lead to

upper and lower approximations that typically enclose the learning curve. The approximations have a form similar to those in Equation (11). While [146] initially hypothesized these approximations could be actual bounds on the learning curve, Sollich and Halees [142] gave several counter examples and disproved this claim.

For the noiseless case, in which $\sigma = 0$, Michelli and Wahba [147] give a lower bound for $R^{\mathrm{PA}}(X_n)$, which in turn provides a lower bound on the learning curve

$$\bar{R}_n^{\mathrm{PA}} \geq \sum_{i=n+1}^{\infty} \lambda_i. \tag{12}$$

Plaskota [148] extends this result to take into account noise in the GP and Sollich and Halees [142] provide further extensions that hold under less stringent assumptions. Using a finite-dimensional basis, [149] develops a method to approximate GPs that scales better in n and finds an upper bound. The latter immediately implies a bound on the learning curve as well.

Several alternative approximations exist. For example, Särkkä [150] uses numerical integration to approximate the learning curve, for which the eigenvalues do not need to be known at all. Opper [151] gives an upper bound for the entropic loss using techniques from statistical physics and derives the asymptotic shape of this bound for Wiener processes for which $k(x, x') = \min(x, x')$ and the squared exponential covariance function. [142] notes that in case the entropic loss is small it approximates the squared error of the GP, thus for large n this also implies a bound on the learning curve.

5.4 Limits of the Eigenvalue Spectrum

Sollich and Halees [142] also explore what the limits are of bounds and approximations based on the eigenvalue spectrum. They create problems that have equal spectra but different learning curves, indicating that learning curves cannot be predicted reliably based on eigenvalues alone. Some works rely on more information than just the spectrum, such as [141], [149] whose bounds also depend on integrals involving the weighted and squared versions of the covariance function. Sollich and Halees also shows several examples where the lower bounds from [145] and [148] can be arbitrarily loose, but at the same time cannot be significantly tightened. Also their own approximation, which empirically is found to be the best, cannot be further refined. These impossibility results leads them to question the use of the eigenvalue spectrum to approximate the learning curve and what further information may be valuable.

Malzahn and Opper [152] re-derive the approximation from [142] using a different approach. Their variational framework may provide more accurate approximations to the learning curve, presumably also for GP classification [140], [142]. It has also been employed, among others, to estimate the variance of PA learning curves [153], [154].

5.5 Smoothness and Asymptotic Decay

The smoothness of a GP can, in 1D, be characterized by the Sacks-Ylvisaker conditions [155]. These conditions capture an aspect of the smoothness of a process in terms of its derivatives. The order s used in these regularity conditions indicates, roughly, that the sth derivative of the stochastic process exists, while the $(s+1)$th does not. Under these conditions, Ritter [156] showed that the asymptotic decay rate of the PA learning curve is of order $O(n^{-(2s+1)/(2s+2)})$. Thus the smoother the process the faster the decay rate.

As an illustration: the squared exponential covariance induces a process that is very smooth as all orders of derivatives exist. The smoothness of the so-called modified Bessel covariance function [140] is determined by its order k, which relates to the order in the Sacks-Ylvisaker conditions as $s = k - 1$. If $k = 1$, the covariance function leads to a process that is not even once differentiable. In the limit of $k \to \infty$, it converges to the squared exponential covariance function [140], [141]. For the roughest case $k = 1$ the learning curve behaves asymptotically as $O(n^{-1/2})$. For the smoother SE covariance function the rate is $O(n^{-1}\log(n))$ [151]. For other rates see [141, Chapter 7] and [141].

In particular cases, the approximations in [142] can have stronger implications for the rate of decay. If the eigenvalues decrease as a power law $\lambda_i \sim i^{-r}$, in the asymptotic regime ($\bar{R}_n^{\mathrm{PA}} \ll \sigma^2$), the upper and lower approximations coincide and predict that the learning curve shape is given by $(n/\sigma^2)^{-(r-1)/r}$. For example, for covariance function of the classical Ornstein–Uhlenbeck process, i.e., $k(x, x') = \exp - \frac{|x-x'|}{\ell}$ with ℓ the length scale, we have $\lambda_i \sim i^{-2}$. The eigenvalues of the squared exponential covariance function decay faster than any power law. Taking $r \to \infty$, this implies a shape of the form $\frac{\sigma^2}{n}$. These approximation are indeed in agreement with known exact results [142]. In the initial regime ($\bar{R}_n^{\mathrm{PA}} \gg \sigma^2$), if one takes $\sigma \to 0$ the lower approximation gives $n^{-(r-1)}$. In this case, the suggested curve for the Ornstein–Uhlenbeck process takes on the form n^{-1}, which agrees with exact calculations as well. In the initial regime for the squared exponential, no direct shape can be computed without additional assumptions. Assuming $d = 1$, a uniform input distribution, and large n, the approximation suggests a shape of the form ne^{-cn^2} for some constant c [142], which is faster even than exponential.

5.6 Bounds Through Monotonicity

Section 4.6 indicated that the PA learning curve is always monotone if the problem is well specified. Therefore, under the assumption of a well-specified GP, its learning curve is monotone when its average is considered over all possible training sets (as defined by the correctly specified prior). As it turns out, however, this curve is already monotone *before* averaging over all possible training sets, as long as the smaller set is contained in the larger, i.e., $R^{\mathrm{PA}}(X_n) \geq R^{\mathrm{PA}}(X_{n+1})$ for all $X_n \subset X_{n+1}$. This is because the GP's posterior variance decreases with every addition of a training object. This can be proven from standard results on the conditioning a multivariate Gaussians [141]. Such sample-based monotonicity of the posterior variance can generally be obtained if the likelihood function is modeled by an exponential family and a corresponding conjugate prior is used [157].

Williams and Vivarelli [141] use this result to construct bounds on the learning curve. The key idea is to take the n training points and treat them as n training sets of just a

single point. Each one gives an estimate of generalization error for the test points (Equation (6)). Because the error always decreases when increasing the training set, the minimum over all train points creates a bound on the error for the whole training set for the test point. The minimizer is the closest training point.

To compute the a bound on the generalization error, one needs to consider the expectation w.r.t. P_X. In order to keep the analysis tractable, Williams and Vivarelli [141] limit themselves to 1D input spaces where P_X is uniform and perform the integration numerically. They further refine the technique to training sets of two points as well, and, find that as expected the bounds become tighter, since larger training sets imply better generalization. Sollich and Halees [142] extend their technique to non-uniform 1D P_X. By considering training points in a ball of radius ρ around a test point, Lederer et al. [143] derive a similar bound that converges to the correct asymptotic value. In contrast, the original bound from [141] becomes looser as n grows. Experimentally, [143] shows that under a wide variety of conditions their bound is relatively tight.

6 Ill-Behaved 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 set up. Deterioration with more training data can obviously occur when considering the curve $R(A(S_n))$ for a particular training set, because for every n, we can be unlucky with our draw of S_n. That ill-behavior can also occur in expectation, i.e., for $\bar{R}_n(A)$, may be less obvious.

In the authors' experience, most researchers expect improved performance of their learner with more data. Less anecdotal evidence can be found in literature. [115, page 153] states that when n surpasses the VC-dimension, the curve must start decreasing. [21, Subsection 9.6.7] claims that for many real-world problems they decay monotonically. [158] calls it expected that performance improves with more data and [49] makes a similar claim, [159] and [160] consider it conventional wisdom, and [103] considers it widely accepted. Others assume well-behaved curves [37], which means that curves are smooth and monotone [161]. Note that a first example of nonmonotonic behavior had actually already been given in 1989 [62], which we further cover in Section 6.2.

Before this section addresses actual bad behavior, we cover phase transitions, which are at the brink of becoming ill-behaved. Possible solutions to nonmonotonic behavior are discussed at the end. Fig. 3 provides an overview of types of ill-behaved learning curve shapes with subsection references. The code reproducing these curves (all based on actual experiments) can be retrieved from https://github.com/tomviering/ill-behaved-learning-curves.

6.1 Phase Transitions

As for physical systems, in a phase transition, particular learning curve properties change relatively abruptly, (almost) discontinuously. Fig. 3 under (6.1) gives an example of how this can manifest itself. In learning, techniques from statistical physics can be employed to model and analyze these transitions, where it typically is studied in the limit of large samples and high input dimensionality [68]. Most theoretical insights are limited to relatively simple learners, like the perceptron, and often apply to PA curves.

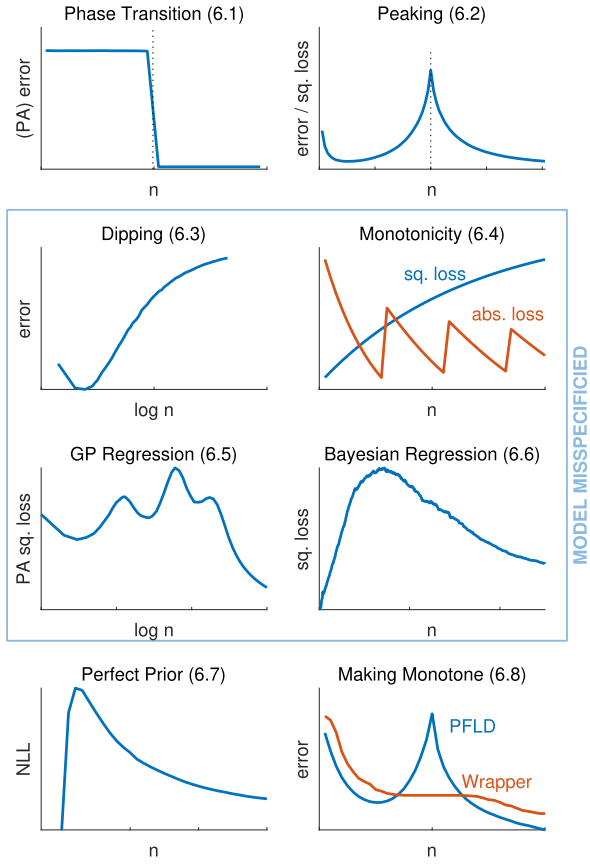

Fig. 3. Qualitative overview of various learning curve shapes placed in different categories with references to their corresponding subsections. All have the sample size n on the horizontal axis. Dotted lines indicate the transition from under to overparametrized models. Abbreviations; error: classification error; sq. loss: squared loss; NLL: negative log likelihood; abs. loss: absolute loss; PA indicates the problem-average learning curve is shown.

Let us point out that abrupt changes also seem to occur in human learning curves [13], [162], in particular when the task is complex and has a hierarchical structure [163]. A first mention of the occurrence of phase transitions, explicitly in the context of learning curves, can be found in [164]. It indicates the transition from memorization to generalization, which occurs, roughly, around the time that the full capacity of the learner has been used. Györgyi [165] provides a first, more rigorous demonstration within the framework of statistical physics—notably, the so-called thermodynamic limit [68]. In this setting, actual transitions happen for single-layer perceptrons where weights take on binary values only.

The perceptron and its thermodynamic limit are considered in many later studies as well. The general finding is that, when using discrete parameter values—most often binary weights, phase transitions can occur [132], [166], [167]. The behavior is often characterized by long plateaus where the perceptron cannot learn at all (usually in the overparametrized, memorization phase, where $n < d$) and has random guessing performance, until a point where the perceptron starts to learn (at $n > d$, the underparametrized regime) at which a jump occurs to non-trivial performance.

Phase transitions are also found in two-layer networks with binary weights and activations [167], [168], [169]. This happens for the so-called parity problem where the aim is to detect the parity of a binary string [170] for which Opper [171] found phase transitions in approximations of the learning curve. Learning curve bounds may display phase transitions as well [118], [172], though [132], [172] question whether these will also occur in the actual learning curve. Both Sompolinsky [173] and Opper [171] note that the sharp phase transitions predicted by theory, will be more gradual in real-world settings. Indeed, when studying this literature, one should be careful in interpreting theoretical results, as the transitions may occur only under particular assumptions or in limiting cases.

For unsupervised learning, phase transitions have been shown to occur as well [174], [175] (see the latter for additional references). Ipsen and Hansen [176] extend these analyses to PCA with missing data. They also show phase transitions in experiments on real-world data sets. [177] provides one of the few real application papers where a distinct, intermediate plateau is visible in the learning curve.

For Fig. 3 under (6.1), we constructed a simple phase transition based on a two-class classification problem, $y \in \{+1, -1\}$, with the first 99 features standard normal and the 100th feature set to $\frac{y}{100}$. PFLD's performance shows a transition at $n = 100$ for the error rate.

6.2 Peaking and Double Descent

The term peaking indicates that the learning curve takes on a maximum, typically in the form of a cusp, see Fig. 3 under (6.2). Unlike many other ill behaviors, peaking can occur in the realizable setting. Its cause seems related to instability of the model. This peaking should not be confused with peaking for feature curves as covered in Section 2.6, which is related to the curse of dimensionality. Nevertheless, the same instability that causes peaking in learning curves can also lead to a peak in feature curves, see Fig. 2. The latter phenomenon has gained quite some renewed attention under the name double descent [65].

By now, the term (sample-wise) double descent has become a term for the peak in the learning curve for deep neural networks [56], [178]. Related terminologies are model-wise double descent, that describe a peak in the plot of performance versus model size, and epoch-wise double descent, that shows a peak in the training curve [56].

Peaking was first observed for the pseudo-Fisher's linear discriminant (PFLD) [62] and has been studied already for quite some time [179]. 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 thermodynamical limit is given in [66]. In such works, originating from statistical physics, the usual quantity of interest is $\alpha = \frac{d}{n}$ that controls the relative sizes for d and n going to infinity [67], [68], [129].

Raudys and Duin [80] investigate this behavior 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 [71] and peak in the learning curve around $n \approx d$. Using an analysis similar to [80], [180] studies the peaking phenomenon in semi-supervised learning (see [23]) and shows that unlabeled data can both mitigate or worsen it.

Peaking of the PFLD can be avoided through regularization, e.g., by adding λI to the covariance matrix [71], [80]. The performance of the model is, however, very sensitive to the correct tuning of the ridge parameter λ [71], [181]. Assuming the data is isotropic, [182] shows that peaking disappears for the optimal setting of the regularization parameter. Other, more heuristic solutions change the training procedure altogether, e.g., [183] uses an iterative procedure that decides which objects PFLD should be trained on, as such reducing n and removing the peak. [184] adds copies of objects with noise, increasing n, or increases the dimensionality by adding noise features, increasing d. Experiments show this can remove the peak and improve performance.

Duin [63] illustrates experimentally that the SVM may not suffer from peaking in the first place. Opper [171] suggests a similar conclusion based on a simplistic thought experiment. For specific learning problems, both [66] and [166] 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. Opper [185] studies the behavior of the SVM in the thermodynamic limit which does not show peaking either. Spigler et al. [186] show, however, that double descent for feature curves can occur using the (squared) hinge loss, where the peak is typically located at an $n > d$.

Further insight of when peaking can occur may be gleaned from recent works like [187] and [188], which perform a rigorous analysis of the case of Fourier Features with PFLD 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. Furthermore, [189] shows that a peak can occur where the training set size n equals the input dimensionality d, but also when n matches the number of parameters of the learner, depending on the latter's degree of nonlinearity. Multiple peaks are also possible for $n < d$ [182].

6.3 Dipping and Objective Mismatch

In dipping, the learning curve may initially improve with more samples, but the performance eventually deteriorates and never recovers, even in the limit of infinite data [190], see Fig. 3 below (6.3). Thus the best expected performance is reached at a finite training set size. By constructing an explicit problem [99, page 106], Devroye et al. already

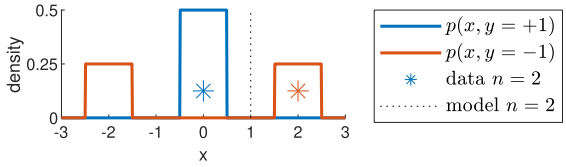

Fig. 4. A two-class problem that dips for various linear classifiers (cf. [190]). The sample data (the two s) shows that, with small samples, the linear model with optimal error rate can be obtained. However, due to the surrogate loss typically optimized, the decision boundary obtained in the infinite sample limit is around the suboptimal $x = 0$.

showed that the nearest neighbor classifier is not always smart, meaning its learning curve can go up locally. A similar claim is made for kernel rules [99, Problems 6.14 and 6.15]. A 1D toy problem for which many well-known linear classifiers (e.g., SVM, logistic regression, LDA, PFLD) dip is given in Fig. 4. In a different context, Ben-David et al. [191] provide 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 approaches 1.

Another example, Lemma 15.1 in [98], gives an insightful case of dipping for likelihood estimation.

What is essential for dipping to occur is that the classification problem at hand 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 (cf. e.g., [115, page 119] [192]). If classification-calibrated loss functions are used and the hypothesis class is rich enough to contain the true model, then minimizing the surrogate loss will also minimize the error rate [191], [193]. As such, consistent learners, that deliver asymptotically optimal performance by definition, cannot have learning curves that keep increasing monotonically and, therefore, cannot dip. Other works also show dipping of some sort. For example, Frey and Fisher [46] fit 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 thus is closely related to objective misspecification. Brumen et al. [102] also shows an ill-behaving curve of C4.5 that seems to go up. They note that 34 more curves could not be fitted well using their parametric models, where possibly something similar is going on. In [194], we find another potential example of dipping as, in Fig. 6, the accuracy goes down with increasing sample sizes.

Anomaly or outlier detection using k-nearest neighbors (kNN) can also shows dipping behavior [160] (referred to as gravity defying learning curves). Also here is a mismatch between the objective that is evaluated with, i.e., the AUC, and kNN that does not optimize the AUC. Hess and Wei [32] also show kNN learning curves that deteriorate in terms of AUC in the standard supervised setting.

Also in active learning [24] 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 [195], [196]. In that case, active learners provide optimal performance for a number of labeled samples that is smaller than the complete training set. This could be interpreted as a great success for active learning. It implies that even in regular supervised learning, one should maybe use an active learner to pick a subset from one's complete training set, as this can improve performance. It cannot be ruled out, therefore, that the active learner uses an objective that matches better with the evaluation measure [197].

Meng and Xie [159] construct a dipping curve in the context of time series modeling with ordinary least squares. In their setting, they use an adequate parametric model, but the distribution of the noise changes every time step, which leads least squares to dipping. In this case, using the likelihood to fit the model resolves the non-monotonicity.

Finally, so-called negative transfer [198], as it occurs in transfer learning and domain adaptation [199], [200], 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.

6.4 Risk Monotonicity and ERM

Several novel examples of non-monotonic behavior for density estimation, classification, and regression by means of standard empirical risk minimization (ERM) are shown in [201]. Similar to dipping, the squared loss increases with n, but in contrast does eventually recover, see Fig. 3 under (6.4). However, these examples cannot be explained either in terms of dipping or peaking. Dipping is ruled out as, in ERM, the learner optimizes the loss that is used for evaluation. In addition, non-monotonicity can be demonstrated for any n and so there is no direct link with the capacity of the learner, ruling out an explanation in terms of peaking.

Proofs of non-monotonicity are given for squared, absolute, and hinge loss. It is demonstrated that likelihood estimators suffer the same deficiency. Two learners are reported that are provably monotonic: mean estimation based on the L_2 loss and the memorize algorithm. 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 [68], [114], illustrating that monotonicity and PAC are essentially different concepts. Here, we like to mention [202], which shows generalization even beyond memorization of the finite training data.

It is shown experimentally that regularization can actually worsen the non-monotonic behavior. In contrast, Nakkiran [182] shows that optimal tuning of the regularization parameter can guarantee monotonicity in certain settings. A final experiment from [201] shows a surprisingly jagged learning curve for the absolute loss, see Fig. 3 under (6.4).

6.5 Misspecified Gaussian Processes

Gaussian process misspecification has been studied in the regression setting where the so-called teacher model provides the data, while the student model learns, assuming a covariance or noise model different from the teacher. If they are equal, the PA curve is monotone (Section 4.6).

Sollich [203] analyzes the PA learning curve using the eigenvalue decomposition earlier covered. He assumes both student and teacher use kernels with the same eigenfunctions but possibly differing eigenvalues. Subsequently, he

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 Fig. 3 under (6.5)), even if the student uses the teacher noise level. For a uniform distribution in one dimension, there may be arbitrarily many overfitting maxima if the student has a small enough noise level. In addition the convergence rates change and may become logarithmically slow.

The above analysis is extended by Sollich in [204], where hyperparameters such as length scale and noise level, are now optimized during learning based on evidence maximization. Among others, he finds that for the hypercube the arbitrary many overfitting maxima do not arise anymore and the learning curve becomes monotone. All in all, Sollich concludes that optimizing the hyperparameters using evidence maximization can alleviate non-monotonicity.

6.6 Misspecified Bayesian Regression

Grünwald and Van Ommen [205] show that a (hierarchical) Bayesian linear regression model can give a broad peak in the learning curve of the squared risk, see Fig. 3 under (6.6). This can happen if 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 according to Grünwald and Van Ommen. The peak in the learning curve is fairly broad and occurs in various experiments. As also no approximations are to blame, the authors conclude that Bayes' rule is really at fault as it cannot handle the misspecification. The non-monotonicity can happen if the probabilistic model class is not convex.

Following their analysis, a modified Bayes rule is introduced, in which the likelihood is raised to some power η. The parameter η cannot be learned in a Bayesian way, leading to their SafeBayes approach. Their technique alleviates the broad peak in the learning curve and is empirically shown to make the curves generally more well-behaved.

6.7 The Perfect Prior

As we have seen in Section 4.6, the PA learning curve is always monotone if the problem is well specified and a Bayesian decision theoretical approach is followed. Nonetheless, the fact that the PA curve is monotone does not mean that the curve for every individual problem is. [206] offers an insightful example (see also Fig. 3 beneath (6.7)): 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 $n \to \infty$, the optimal loss is recovered, forming a bump in the learning curve. Note that this construction is rather versatile and can create non-monotonic behavior for practically any Bayesian estimation task. In a similar way, any type of regularization can lead to comparable learning curve shapes (see also [99], [201] and Section 6.4).

An related example can be found in [157]. 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. GPs fall in this last class.

6.8 Monotonicity: A General Fix?

This section has noted a few particular approaches to restore monotonicity of a learning curve. One may wonder, however, whether generally applicable approaches exist that can turn any learner into a monotone one. A first attempt is made in [207] which proposes a wrapper that, with high probability, makes any classifier monotone 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—following a hypothesis test, 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 monotone version may learn more slowly, giving rise to the question whether there always will be a trade-off between monotonicity and speed (refer to the learning curve in Fig. 3 under (6.8)).

[208] extended this idea, proposing 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 them to determine when to switch to a model trained with more data. In contrast to [207], they argue that their second algorithm does not learn slower, as its generalization bound coincides with a known lower bound of regular supervised learning.

Recently, [209] proved that, in terms of the 0-1 loss, any learner can be turned into a monotonic one. It extends a result obtained in [210], disproving a conjecture by Devroye et al. [99, page 109] that universally consistent monotone classifiers do not exist.

7 Discussion and Conclusion

Though there is some empirical evidence that for large deep learners, especially in the Big Data regime, learning curves behave like power laws, such conclusions seem premature. For other learners results are mixed. Exponential shapes cannot be ruled out and there are various models that can perform on par with power laws. Theory, on the other hand, often points in the direction of power laws and also supports exponential curves. GP learning curves have been analyzed for several special cases, but their general shape remains hard to characterize and offers rich behavior such as different regimes. Ill-behaved learning curves illustrate that various shapes are possible that are hard to characterize. What is perhaps most surprising is that these behaviors can even

occur in the well-specified case and realizable setting. It should be clear that, currently, there is no theory that covers all these different aspects.

Roughly, our review sketches the following picture. Much of the work on learning curves seems scattered and incidental. Starting with Hughes, Foley, and Raudys, some initial contributions appeared around 1970. The most organized efforts, starting around 1990, come from the field of statistical physics, with important contributions from Amari, Tishby, Opper, and others. These efforts have found their continuation within GP regression, to which Opper again contributed significantly. For GPs, Sollich probably offered some of the most complete work.

The usage of the learning curve as an tool for the analysis of learning algorithms has varied throughout the past decades. In line with Langley, Perlich, Hoiem, et al. [36], [42], [44], we would like to suggest a more consistent use. We specifically agree with Perlich et al. [36] that without a study of the learning curves, claims of superiority of one approach over the other are perhaps only valid for very particular sample sizes. Reporting learning curves in empirical studies can also help the field to move away from its fixation on bold numbers, besides accelerating learning curve research.

In the years to come, we expect investigations of parametric models and their performance in terms of extrapolation. Insights into these problems become more and more important—particularly within the context of deep learning—to enable the intelligent use of computational resources. In the remainder, we highlight some specific aspects that we see as important.

7.1 Averaging Curves and The Ideal Parametric Model

Especially for extrapolation, a learning curve should be predictable, which, in turn, asks for good parametric model. It seems impossible to find a generally applicable, parametric model that covers all aspects of the shape, in particular ill-behaving curves. Nevertheless, we can try to find a model class that would give us sufficient flexibility and extrapolative power. Power laws and exponentials should probably be part of that class, but does that suffice?

To get close to the true learning curve, some studies average hundreds or thousands of individual learning curves [26], [28]. Averaging can mask interesting characteristics of individual curves [6], [211]. This has been extensively debated in psychonomics, where cases have been made for exponentially shaped individual curves, but power-law-like average curves [212], [213]. In applications, we may need to be able to model individual, single-training-set curves or curves that are formed on the basis of relatively small samples. As such, we see potential in studying and fitting individual curves to better understanding their behavior.

7.2 How to Robustly Fit Learning Curves

A technical problem that has received little attention (except in [44], [106]) is how to properly fit a learning curve model. As far as current studies at all mention how the fitting is carried out, they often seem to rely on simple least squares fitting of log values, assuming independent Gaussian noise at every n. Given that this noise model is not bounded—while a typical loss cannot become negative, this choice seems disputable. At the very least, derived confidence intervals and p-values should be taken with a grain of salt.

In fact, fitting can often fail, but this is not always reported in literature. In one of the few studies, [106], where this is at all mentioned, 2% of the fits were discarded. Therefore, further investigations into how to robustly fit learning curve models from small amounts of curve data seems promising to investigate. In addition, a probabilistic model with assumptions that more closely match the intended use of the learning curve seems also worthwhile.

7.3 Bounds and Alternative Statistics

One should be careful in interpreting theoretical results when it comes to the shape of learning curves. Generalization bounds, such as those provided by PAC, that hold uniformly over all P may not correctly characterize the curve shape. Similarly, a strictly decreasing bounds does not imply monotone learning curves, and thus does not rule out ill-behavior. We merely know that the curve lies under the bound, but this leaves room for strange behavior. Furthermore, PA learning curves, which require an additional average over problems, can show behavior substantially different from those for a single problem, because here averaging can also mask characteristics.

Another incompatibility between typical generalization bounds and learning curves is that the former are constructed to hold with *high probability* with respect to the sampling of the training set, while the latter look at the *mean* performance over all training sets. Though bounds of one type can be transformed into the other [214], this conversion can change the actually shape of the bound, thus such high probability bounds may also not correctly characterize learning curve shape for this reason.

The preceding can also be a motivation to actually study learning curves for statistics other than the average. For example, in Equation (2), instead of the expectation we could look at the curves of the median or other percentiles. These quantities are closer related to high probability learning bounds. Of course, we would not have to choose the one learning curve over the other. They offer different types of information and, depending on our goal, may be worthwhile to study next to each other. Along the same line of thought, we could include quartiles in our plots, rather than the common error bars based on the standard deviation. Ultimately, we could even try to visualize the full loss distribution at every sample size n and, potentially, uncover behavior much more rich and unexpected.

A final estimate that we think should be investigates more extensively is the training loss. Not only can this quantity aid in identifying overfitting and underfitting issues [4], [211], but it is a quantity that is interesting to study as such or, say, in combination with the true risk. Their difference, a simple measure of overfitting, could, for example, turn out to behave more regular than the two individual measures.

7.4 Research Into Ill-Behavior and Meta-Learning

We believe better understanding is needed regarding the occurrence of peaking, dipping, and otherwise non-monotonic or phase-transition-like behavior: when and why does

this happen? Certainly, a sufficiently valid reason to investigate these phenomena is to quench one's scientific curiosity. We should also be careful, however, not to mindlessly dismiss such behavior as mere oddity. Granted, these uncommon and unexpected learning curves have often been demonstrated in artificial or unrealistically simple settings, but this is done to make at all insightful that there is a problem.

The simple fact is that, at this point, we do not know what role these phenomena play in real-world problems. [106] shared a database of learning curves for 20 learners on 246 datasets. These questions concerning curious curves can now be investigated in more detail, possibly even in automated fashion. Given the success of meta-learning using learning curves 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.

7.5 Open Theoretical Questions

There are two specific theoretical questions that we would like to ask. Both are concerned with the monotonicity.

One interesting question that remains is whether monotonic learners can be created for losses other than 0-1, unbounded ones in particular. Specifically, we wonder whether maximum likelihood estimators for well-specified models behave monotonically. Likelihood estimation, being a century-old, classical technique [215], 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 because this estimator has been extensively studied already and still plays a central role in statistics and abutting fields.

Second, as ERM remains one of the primary inductive principles in machine learning, another question that remains interesting—and which was raised in [201], [209] as well, is when ERM *will* result in monotonic curves?

7.6 Concluding

More than a century of learning curve research has brought us quite some insightful and surprising results. What is more striking however, at least to us, is that there is still so much that we actually do not understand about learning curve behavior. Most theoretical results are restricted to relatively basic learners, while much of the empirical research that has been carried out is quite limited in scope. In the foregoing, we identified some specific challenges already, but we are convinced that many more open and interesting problems can be discovered. In this, the current review shouls

Acknowledgments

We would like to thank Peter Grünwald, Alexander Mey, Jesse Krijthe, Robert-Jan Bruintjes, Elmar Eisemann and three anonymous reviewers for their valueable input.

References

[1] H. Ebbinghaus, *Über Das Gedächtnis: Untersuchungen Zur Experimentellen Psychologie*. Leipzig, Germany: Duncker & Humblot, 1885.
[2] L. E. Yelle, "The learning curve: Historical review and comprehensive survey," *Decis. Sci.*, vol. 10, no. 2, pp. 302–328, 1979.
[3] M. J. Anzanello and F. S. Fogliatto, "Learning curve models and applications: Literature review and research directions," *Int. J. Ind. Ergonom.*, vol. 41, no. 5, pp. 573–583, 2011.
[4] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.
[5] M. Loog, "Supervised classification: Quite a brief overview," in *Machine Learning Techniques for Space Weather*. Amsterdam, The Netherlands: Elsevier, 2018, pp. 113–145.
[6] R. A. Schiavo and D. J. Hand, "Ten more years of error rate research," *Int. Statist. Rev.*, vol. 68, no. 3, pp. 295–310, 2000.
[7] T. Domhan et al., "Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 3460–3468.
[8] C. Perlich, "Learning curves in machine learning," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds., Berlin, Germany: Springer, 2010, pp. 577–488.
[9] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning*. Berlin, Germany: Springer Science & Business Media, 2011.
[10] M. Osborne, "A modification of veto logic for a committee of threshold logic units and the use of 2-class classifiers for function estimation," Ph.D. dissertation, Dept. Comput. Sci., Oregon State Univ., Corvallis, OR, 1975.
[11] L. E. Atlas et al., "Training connectionist networks with queries and selective sampling," in *Proc. 2nd Int. Conf. Neural Inf. Process. Syst.*, 1990, pp. 566–573.
[12] H. Sompolinsky et al., "Learning from examples in large neural networks," *Phys. Rev. Lett.*, vol. 65, no. 13, 1990, Art. no. 1683.
[13] W. L. Bryan and N. Harter, "Studies in the physiology and psychology of the telegraphic language," *Psychol. Rev.*, vol. 4, no. 1, 1897, Art. no. 27.
[14] K. W. Spence, "The differential response in animals to stimuli varying within a single dimension," *Psychol. Rev.*, vol. 44, no. 5, 1937, Art. no. 430.
[15] E. J. Swift, "Studies in the psychology and physiology of learning," *Amer. J. Psychol.*, vol. 14, no. 2, pp. 201–251, 1903.
[16] O. Lipmann, "Der einfluss der einzelnen wiederholungen auf verschieden starke und verschieden alte associationen," *Z. Psychol. Physiol. Si.*, vol. 35, pp. 195–233, 1904.
[17] A. Ritchie, "Thinking and machines," *Philosophical*, vol. 32, no. 122, 1957, Art. no. 258.
[18] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, 1958, Art. no. 386.
[19] D. H. Foley, "Considerations of sample and feature size," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 5, pp. 618–626, Sep. 1972.
[20] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Trans. Electron.*, vol. EC-14, no. 3, pp. 326–334, Jun. 1965.
[21] R. O. Duda et al., *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2012.
[22] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
[23] O. Chapelle et al., *Semi-Supervised Learning*. Cambridge, MA, USA: The MIT Press, 2010.
[24] B. Settles, "Active learning literature survey," University of Wisconsin-Madison, Madison, WI, Tech. Rep. 1648, 2009.
[25] S. Mukherjee et al., "Estimating dataset size requirements for classifying DNA microarray data," *J. Comput. Biol.*, vol. 10, no. 2, pp. 119–142, 2003.
[26] R. L. Figueroa et al., "Predicting sample size required for classification performance," *BMC Med. Inform. Decis*, vol. 12, no. 1, 2012, Art. no. 8.
[27] A. N. Richter and T. M. Khoshgoftaar, "Learning curve estimation with large imbalanced datasets," in *Proc. IEEE 18th Int. Conf. Mach. Learn. Appl.*, 2019, pp. 763–768.
[28] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst.*, 2002, pp. 841–848.
[29] 2022. [Online]. Available: https://waikato.github.io/weka-wiki/experimenter/learning_curves/

[30] 2022. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.learning_curve.html

[31] J.-H. Kim, "Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap," *Comput. Statist. Data Anal.*, vol. 53, no. 11, pp. 3735–3745, 2009.

[32] K. R. Hess and C. Wei, "Learning curves in classification with microarray data," *Seminars Oncol.*, vol. 37, no. 1, pp. 65–68, 2010.

[33] 2022. [Online]. Available: http://prtools.tudelft.nl/

[34] B. Efron, "Estimating the error rate of a prediction rule: Improvement on cross-validation," *J. Amer. Statist. Assoc.*, vol. 78, no. 382, 1983, Art. no. 316.

[35] A. K. Jain, R. C. Dubes, and C. -C. Chen, "Bootstrap techniques for error estimation," *IEEE Trans Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 628–633, Sep. 1987.

[36] C. Perlich et al., "Tree Induction vs. Logistic regression: A learning-curve analysis," *J. Mach. Learn. Res.*, vol. 4, no. 1, pp. 211–255, 2003.

[37] F. Provost et al., "Efficient progressive sampling," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 1999, pp. 23–32.

[38] E. Perez and L. A. Rendell, "Using multidimensional projection to find relations," in *Proc. 12th Int. Conf. Mach. Learn.*, 1995, pp. 447–455.

[39] D. Mazzoni and K. Wagstaff, "Active learning in the presence of unlabelable examples," in *Proc. Eur. Conf. Mach. Learn.*, 2004.

[40] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2008, pp. 1070–1079.

[41] E. E. Ghiselli, "A comparison of methods of scoring maze and discrimination learning," *J. Gen. Psychol.*, vol. 17, no. 1, 1937, Art. no. 15.

[42] P. Langley, "Machine learning as an experimental science," *Mach. Learn.*, vol. 3, no. 1, pp. 5–8, 1988.

[43] N. Bertoldi et al., "Evaluating the learning curve of domain adaptive statistical machine translation systems," in *Proc. Workshop Statist. Mach. Transl.*, 2012, pp. 433–441.

[44] D. Hoiem et al., "Learning curves for analysis of deep networks," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 4287–4296.

[45] C. Cortes et al., "Learning curves: Asymptotic values and rate of convergence," in *Proc. 6th Int. Conf. Neural Inf. Process. Syst.*, 1994, pp. 327–334.

[46] L. J. Frey and D. H. Fisher, "Modeling decision tree performance with the power law," in *Proc. 7th Int. Workshop Artif. Intell. Statist.*, 1999, pp. 59–65.

[47] S. Singh, "Modeling performance of different classification methods: Deviation from the power law," Dept. Comput. Sci., Vanderbilt University, Nashville, TN, USA, Tech. Rep., 2005.

[48] M. Last, "Predicting and optimizing classifier utility with the power law," in *Proc. IEEE 7th Int. Conf. Data Mining Workshops*, 2007, pp. 219–224.

[49] B. Gu et al., "Modelling classification performance for large data sets," in *Proc. Int. Conf. Web-Age Inf. Manage*, 2001, Art. no. 317.

[50] P. Kolachina et al., "Prediction of learning curves in machine translation," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, 2012, pp. 22–30.

[51] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 1, pp. 55–63, Jan. 1968.

[52] A. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," *Handbook Statist.*, vol. 2, pp. 835–855, 1982.

[53] S. Raudys and A. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 3, pp. 252–264, Mar. 1991.

[54] R. P. Duin et al., "Experiments with a featureless approach to pattern recognition," *Pattern Recognit. Lett.*, vol. 18, no. 11/13, pp. 1159–1166, 1997.

[55] L. Torgo, "Kernel regression trees," in *Proc. Eur. Conf. Mach. Learn.*, 1997, pp. 118–127.

[56] P. Nakkiran et al., "Deep double descent: Where bigger models and more data hurt," in *Proc. Int. Conf. Learn. Representations*, 2019.

[57] R. Duin and D. Tax, "Statistical pattern recognition," in *Handbook of Pattern Recognition and Computer Vision*, C. H. Chen and P. S. P. Wang, Eds., Singapore: World Scientific, 2005, pp. 3–24.

[58] L. Chen et al., "Multiple descent: Design your own generalization curve," 2020, *arXiv:2008.01036*.

[59] J. M. Van Campenhout, "On the peaking of the hughes mean recognition accuracy: The resolution of an apparent paradox," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-8, no. 5, pp. 390–395, May 1978.

[60] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," in *Handbook of Statistics*, P. Krishnaiah and L. Kanal, Eds., Amsterdam, The Netherlands: Elsevier, 1982, vol. 2, ch. 39, pp. 835–855.

[61] S. Raudys and V. Pikelis, "On dimensionality, sample size, classification error, and complexity of classification algorithm in pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-2, no. 3, pp. 242–252, May 1980.

[62] F. Vallet et al., "Linear and nonlinear extension of the pseudo-inverse solution for learning boolean functions," *Europhys. Lett.*, vol. 9, no. 4, 1989, Art. no. 315.

[63] R. P. Duin, "Classifiers in almost empty spaces," in *Proc. IEEE 15th Int. Conf. Pattern Recognit.*, 2000, pp. 1–7.

[64] A. Zollanvari et al., "A theoretical analysis of the peaking phenomenon in classification," *J. Classification*, vol. 37, pp. 421–434, 2020.

[65] M. Belkin et al., "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 32, pp. 15 849–15 854, 2019.

[66] M. Opper et al., "On the ability of the optimal perceptron to generalise," *J. Phys. A*, vol. 23, no. 11, 1990, Art. no. L581.

[67] T. L. Watkin et al., "The statistical mechanics of learning a rule," *Rev. Mod. Phys.*, vol. 65, no. 2, 1993, Art. no. 499.

[68] A. Engel and C. Van den Broeck, *Statistical Mechanics of Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2001.

[69] A. K. Jain and W. G. Waller, "On the optimal number of features in the classification of multivariate gaussian data," *Pattern Recognit.*, vol. 10, no. 5/6, pp. 365–374, 1978.

[70] R. P. W. Duin, "On the accuracy of statistical pattern recognizers," Ph.D. dissertation, Dept. Appl. Phys., Technische Hogeschool Delft, Delft, Netherlands, 1978.

[71] M. Skurichina and R. P. Duin, "Stabilizing classifiers for very small sample sizes," in *Proc. IEEE 13th Int. Conf. Pattern Recognit.*, 1996, pp. 891–896.

[72] R. Duin, "On the choice of smoothing parameters for parzen estimators of probability density functions," *IEEE Trans. Comput.*, vol. 25, no. 11, pp. 1175–1179, Nov. 1976.

[73] J. S. Rosenfeld et al., "A constructive prediction of the generalization error across scales," 2019, *arXiv:1909.12673*.

[74] F. Mohr and J. N. van Rijn, "Learning curves for decision making in supervised machine learning–A survey," 2022, *arXiv:2201.12150*.

[75] 2022. [Online]. Available: https://paperswithcode.com/sota

[76] D. Sculley et al., "Winner's curse? On pace, progress, and empirical rigor," in *Proc. Int. Conf. Learn. Representations*, 2018.

[77] K. Blagec et al., "A critical analysis of metrics used for measuring progress in artificial intelligence," 2020, *arXiv:2008.02577*.

[78] S. Raudys, "On the problems of sample size in pattern recognition (in Russian)," in *Proc. 2nd All-Union Conf. Statist. Meth. Contr. Theory*, 1970, pp. 64–67.

[79] L. Kanal and B. Chandrasekaran, "On dimensionality and sample size in statistical pattern classification," *Pattern Recognit.*, vol. 3, no. 3, pp. 225–234, 1971.

[80] S. Raudys and R. Duin, "Expected classification error of the fisher linear classifier with pseudo-inverse covariance matrix," *Pattern Recognit. Lett.*, vol. 19, no. 5/6, pp. 385–392, 1998.

[81] R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Mining*, 1996, pp. 202–207.

[82] J. Bornschein et al., "Small data, big decisions: Model selection in the small-data regime," 2020, *arXiv:2009.12583*.

[83] B. Strang et al., "Don't rule out simple models prematurely: A large scale benchmark comparing linear and non-linear classifiers in OpenML," in *Proc. Int. Symp. Intell. Data Anal.*, 2018, pp. 303–315.

[84] N. Mørch et al., "Nonlinear versus linear models in functional neuroimaging: Learning curves and generalization crossover," in *Proc. Biennial Int. Conf. Inf. Process. Med. Imag.*, 1997, pp. 259–270.

[85] J. W. Shavlik et al., "Symbolic and neural learning algorithms: An experimental comparison," *Mach. Learn.*, vol. 6, no. 2, pp. 111–143, 1991.

[86] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Mach. Learn.*, vol. 29, no. 2/3, pp. 103–130, 1997.
[87] R. P. D. Duin and E. M. Pekalska, *Pattern Recognition: Introduction Terminology*, 37 Steps, 2016.
[88] S. Jones et al., "An introduction to power and sample size estimation," *Emerg. Med. J.*, vol. 20, no. 5, 2003, Art. no. 453.
[89] G. H. John and P. Langley, "Static versus dynamic sampling for data mining," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Mining*, 1996, pp. 367–370.
[90] R. Leite and P. Brazdil, "Improving progressive sampling via meta-learning on learning curves," in *Proc. Eur. Conf. Mach. Learn.*, 2004, pp. 250–261.
[91] C. Meek et al., "The learning-curve sampling method applied to model-based clustering," *J. Mach. Learn. Res.*, vol. 2, no. Feb., pp. 397–418, 2002.
[92] K. Tomanek and U. Hahn, "Approximating learning curves for active-learning-driven annotation," in *Proc. Int. Conf. Lang. Resour. Eval.*, 2008, pp. 1319–1324.
[93] R. Leite and P. Brazdil, "Predicting relative performance of classifiers from samples," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 497–503.
[94] R. Leite and P. Brazdil, "An iterative process for building learning curves and predicting relative performance of classifiers," in *Proc. Portuguese Conf. Artif. Intell.*, 2007, pp. 87–98.
[95] J. N. van Rijn et al., "Fast algorithm selection using learning curves," in *Proc. Int. Symp. Intell. Data Anal.*, 2015, pp. 298–309.
[96] F. Mohr and J. N. van Rijn, "Fast and informative model selection using learning curve cross-validation," 2021, *arXiv:2111.13914*.
[97] J. Hestness et al., "Deep learning scaling is predictable, empirically," 2017, *arXiv:1712.00409*.
[98] L. Devroye et al., *A Probabilistic Theory of Pattern Recognition*, vol. 31. New York, NY, USA: Springer, 1996.
[99] T. Viering et al., "Open problem: Monotonicity of learning," in *Proc. Conf. Learn. Theory*, 2019, pp. 3198–3201.
[100] O. Bousquet et al., "A theory of universal learning," 2020, *arXiv:2011.04483*.
[101] M. Hutter, "Learning curve theory," 2021, *arXiv:2102.04074*.
[102] B. Brumen et al., "Best-fit learning curve model for the C4.5 algorithm," *Informatica*, vol. 25, no. 3, pp. 385–399, 2014.
[103] N. Boonyanunta and P. Zeephongsekul, "Predicting the relationship between the size of training sample and the predictive power of classifiers," in *Proc. Int. Conf. Knowl.-Based Intell. Inf. Eng. Syst.*, 2004, pp. 529–535.
[104] S. Ahmad and G. Tesauro, "Study of scaling and generalization in neutral networks," *Neural Netw.*, vol. 1, no. 1, 1988, Art. no. 3.
[105] D. Cohn and G. Tesauro, "Can neural networks do better than the vapnik-chervonenkis bounds?," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1991, pp. 911–917.
[106] F. Mohr et al., "LCDB 1.0: An extensive learning curves database for classification tasks," in *Proc. Eur. Conf. Mach. Learn.*, 2022.
[107] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 843–852.
[108] A. Joulin et al., "Learning visual features from large weakly supervised data," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 67–84.
[109] D. Mahajan et al., "Exploring the limits of weakly supervised pretraining," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 181–196.
[110] J. Kaplan et al., "Scaling laws for neural language models," 2020, *arXiv:2001.08361*.
[111] M. A. Gordon et al., "Data and parameter scaling laws for neural machine translation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2021, pp. 5915–5922.
[112] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12 104–12 113.
[113] V. Vapnik, *Estimation of Dependences Based on Empirical Data Berlin*. Berlin, Germany: Springer, 1982.
[114] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2014.
[115] W. L. Buntine, "A critique of the valiant model," in *Proc. 11th Int. Conf. Artif. Intell.*, 1989, pp. 837–842.
[116] W. E. Sarrett and M. J. Pazzani, "Average case analysis of empirical and explanation-based learning algorithms," Univ. California Irvine, Irvine, CA, Tech. Rep., pp.89–35, 1989.
[117] D. Haussler and M. Warmuth, "The probably approximately correct (PAC) and other learning models," in *Foundations of Knowledge Acquisition*. Berlin, Germany: Springer, 1993, pp. 291–312.
[118] D. Haussler et al., "Rigorous learning curve bounds from statistical mechanics [longer version]," *Mach. Learn.*, vol. 25, no. 2/3, pp. 195–236, 1996.
[119] J. Hestness et al., "Deep learning scaling is predictable, empirically," 2017, *arXiv:1712.00409*.
[120] D. Schuurmans, "Characterizing rational versus exponential learning curves," in *Proc. Eur. Conf. Comput. Learn. Theory*, 1995, pp. 272–286.
[121] D. Schuurmans, "Characterizing rational versus exponential learning curves," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 140–160, 1997.
[122] H. Gu and H. Takahashi, "Exponential or polynomial learning curves? Case-based studies," *Neural Comput.*, vol. 12, no. 4, pp. 795–809, 2000.
[123] H. Gu and H. Takahashi, "How bad may learning curves be?," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1155–1167, Oct. 2000.
[124] T. M. Cover, "Rates of convergence for nearest neighbor procedures," in *Proc. Hawaii Int. Conf. Syst. Sci.*, 1968, pp. 413–415.
[125] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.
[126] D. Peterson, "Some convergence properties of a nearest neighbor decision rule," *IEEE Trans. Inf. Theory*, vol. IT-16, no. 1, pp. 26–31, Jan. 1970.
[127] S.-I. Amari, "A universal theorem on learning curves," *Neural Netw.*, vol. 6, no. 2, pp. 161–166, Jan. 1993.
[128] S. Amari, "Universal property of learning curves under entropy loss," in *Proc. Int. Joint Conf. Neural Netw.*, 1992, pp. 368–373.
[129] M. Opper and D. Haussler, "Calculation of the learning curve of bayes optimal classification algorithm for learning a perceptron with noise," in *Proc. 4th Annu. Workshop Comput. Learn. Theory*, 1991, pp. 75–87.
[130] S.-I. Amari et al., "Four types of learning curves," *Neural Comput.*, vol. 4, no. 4, pp. 605–618, 1992.
[131] S.-I. Amari and N. Murata, "Statistical theory of learning curves under entropic loss criterion," *Neural Comput.*, vol. 5, no. 1, pp. 140–153, 1993.
[132] H. S. Seung et al., "Statistical mechanics of learning from examples," *Phys. Rev. A*, vol. 45, no. 8, 1992, Art. no. 6056.
[133] D. B. Schwartz et al., "Exhaustive learning," *Neural Comput.*, vol. 2, no. 3, pp. 374–385, 1990.
[134] N. Tishby et al., "Consistent inference of probabilities in layered networks: Predictions and generalization," in *Proc. Int. Joint Conf. Neural Netw.*, 1989, pp. 403–409.
[135] E. Levin et al., "A statistical approach to learning and generalization in layered neural networks," in *Proc. 2nd Annu. Workshop Comput. Learn. Theory*, 1989, pp. 245–260.
[136] L. J. Savage, *The Foundations of Statistics*. Hoboken, NJ, USA: Wiley, 1954.
[137] I. J. Good, "On the principle of total evidence," *Brit. J. Philos. Sci.*, vol. 17, no. 4, pp. 319–321, 1967.
[138] P. D. Grünwald and J. Y. Halpern, "When ignorance is bliss," in *Proc. 20th Conf. Uncertainty Artif. Intell.*, 2004, pp. 226–234.
[139] P. R. Graves, "The total evidence theorem for probability kinematics," *Philos. Sci.*, vol. 56, no. 2, pp. 317–324, 1989.
[140] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[141] C. K. I. Williams and F. Vivarelli, "Upper and lower bounds on the learning curve for gaussian processes," *Mach. Learn.*, vol. 40, no. 1, pp. 77–102, 2000.
[142] P. Sollich and A. Halees, "Learning curves for Gaussian process regression: Approximations and bounds," *Neural Comput.*, vol. 14, no. 6, pp. 1393–1428, 2002.
[143] A. Lederer et al., "Posterior variance analysis of gaussian processes with application to average learning curves," 2019, *arXiv:1906.01404*.
[144] L. L. Gratiet and J. Garnier, "Asymptotic analysis of the learning curve for gaussian process regression," *Mach. Learn.*, vol. 98, no. 3, pp. 407–433, 2015.
[145] M. Opper and F. Vivarelli, "General bounds on bayes errors for regression with gaussian processes," in *Proc. 11th Int. Conf. Neural Inf. Process. Syst.*, 1998, pp. 302–308.
[146] P. Sollich, "Learning Curves for Gaussian Processes," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1998, pp. 344–350.

[147] C. A. Michelli and G. Wahba, "Design problems for optimal surface interpolation," Dept. Stat., Wisconsin University Madison, Madison, WI, USA, Tech. Rep., 1979.

[148] L. Plaskota, *Noisy Information and Computational Complexity*, vol. 95. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[149] G. F. Trecate et al., "Finite-dimensional approximation of gaussian processes," in *Proc. 11th Int. Conf. Neural Inf. Process. Syst.*, pp. 218–224, 1999.

[150] S. Särkkä, "Learning curves for gaussian processes via numerical cubature integration," in *Proc. Int. Conf. Artif. Neural Netw.*, 2011, pp. 201–208.

[151] M. Opper, "Regression with Gaussian processes: Average case performance," in *Proc. Theor. Aspects Neural Comput.: A Multidisciplinary Perspective*, 1997, pp. 17–23.

[152] D. Malzahn and M. Opper, "Learning curves for gaussian processes regression: A framework for good approximations," in *Proc. 13th Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 255–261.

[153] D. Malzahn and M. Opper, "Learning curves for Gaussian processes models: Fluctuations and Universality," in *Proc. Int. Conf. Artif. Neural Netw.*, 2001, pp. 271–276.

[154] D. Malzahn and M. Opper, "A variational approach to learning curves," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 463–469.

[155] K. Ritter et al., "Multivariate integration and approximation for random fields satisfying sacks-ylvisaker conditions," *Ann. Appl. Prob.*, vol. 5, no. 2, pp. 518–540, 1995.

[156] K. Ritter, "Almost optimal differentiation using noisy data," *J. Approx. Theory*, vol. 86, no. 3, pp. 293–309, 1996.

[157] M. F. Al-Saleh and F. A. Masoud, "A note on the posterior expected loss as a measure of accuracy in Bayesian methods," *Appl. Math. Comput.*, vol. 134, no. 2/3, pp. 507–514, 2003.

[158] D. M. Tax and R. P. Duin, "Learning curves for the analysis of multiple instance classifiers," in *Proc. Joint IAPR Int. Workshops Statist. Techn. Pattern Recognit. Struct. Syntactic Pattern Recognit.*, 2008, pp. 724–733.

[159] X. L. Meng and X. Xie, "I got more data, my model is more refined, but my estimator is getting worse! am i just dumb?," *Econometric Rev.*, vol. 33, no. 1/4, pp. 218–250, 2014.

[160] K. M. Ting et al., "Defying the gravity of learning curve: A characteristic of nearest neighbour anomaly detectors," *Mach. Learn.*, vol. 106, no. 1, pp. 55–91, 2017.

[161] G. M. Weiss and A. Battistin, "Generating well-behaved learning curves: An empirical study," in *Proc. Int. Conf. Data Sci.*, 2014, pp. 1–4.

[162] W. L. Bryan and N. Harter, "Studies on the telegraphic language: The acquisition of a hierarchy of habits," *Psychol. Rev.*, vol. 6, no. 4, 1899, Art. no. 345.

[163] G. Vetter et al., "Phase transitions in learning," *J. Mind. Behav.*, vol. 18, pp. 335–350, 1997.

[164] S. Patarnello and P. Carnevali, "Learning networks of neurons with boolean logic," *Europhys. Lett.*, vol. 4, no. 4, 1987, Art. no. 503.

[165] G. Györgyi, "First-order transition to perfect generalization in a neural network with binary synapses," *Phys. Rev. A*, vol. 41, no. 12, pp. 7097–7100, 1990.

[166] T. L. H. Watkin et al., "The statistical mechanics of learning a rule," *Rev. Mod. Phys.*, vol. 65, no. 2, pp. 499–556, 1993.

[167] K. Kang et al., "Generalization in a two-layer neural network," *Phys. Rev. E*, vol. 48, no. 6, 1993, Art. no. 4805.

[168] M. Opper, "Statistical Mechanics of Learning: Generalization," in *The Handbook of Brain Theory and Neural Networks*, Cambridge, MA, USA: MIT Press, 1995, Art. no. 20.

[169] H. Schwarze and J. A. Hertz, "Statistical mechanics of learning in a large committee machine," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1993, pp. 523–530.

[170] D. Hansel et al., "Memorization without generalization in a multilayered neural network," *Europhys. Lett.*, vol. 20, no. 5, pp. 471–476, 1992.

[171] M. Opper, "Learning to generalize," *Front. Life*, vol. 3, no. Part 2, pp. 763–775, 2001.

[172] H. Seung, "Annealed theories of learning," in *Proc. CTP-PRSRI Joint Workshop Theor. Phys.*, 1995, pp. 32–41.

[173] H. Sompolinsky, "Theoretical issues in learning from examples," in *Proc. NEC Res. Symp.*, 1993, pp. 217–237.

[174] M. Biehl and A. Mietzner, "Statistical mechanics of unsupervised learning," *Europhys. Lett.*, vol. 24, no. 5, 1993, Art. no. 421.

[175] D. C. Hoyle and M. Rattray, "Statistical mechanics of learning multiple orthogonal signals: Asymptotic theory and fluctuation effects," *Phys. Rev. E*, vol. 75, no. 1, 2007, Art. no. 016101.

[176] N. Ipsen and L. K. Hansen, "Phase transition in PCA with missing data: Reduced signal-to-noise ratio, not sample size!," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 2951–2960.

[177] R. A. Bhat et al., "Adapting predicate frames for Urdu propbanking," in *Proc. Workshop Lang. Technol. Closely Related Lang. Lang. Variants*, 2014, pp. 47–55.

[178] P. Nakkiran, "More data can hurt for linear regression: Samplewise double descent," 2019, *arXiv:1912.07242*.

[179] M. Loog et al., "A brief prehistory of double descent," *Proc. Nat. Acad. Sci. USA*, vol. 117, no. 20, pp. 10 625–10 626, 2020.

[180] J. H. Krijthe and M. Loog, "The peaking phenomenon in semi-supervised learning," in *Proc. Joint IAPR Int. Workshops Statist. Techn. Pattern Recognit. Struct. Syntactic Pattern Recognit.*, 2016, pp. 299–309.

[181] V. Tresp, "The equivalence between row and column linear regression," Siemens, Tech. Rep., 2002.

[182] P. Nakkiran et al., "Optimal regularization can mitigate double descent," 2020, *arXiv:2003.01897*.

[183] R. P. W. Duin, "Small sample size generalization," in *Proc. Scand. Conf. Image Anal.*, 1995, pp. 1–8.

[184] M. Skurichina and R. P. W. Duin, "Regularisation of linear classifiers by adding redundant features," *Pattern Anal. Appl.*, vol. 2, no. 1, pp. 44–52, 1999.

[185] M. Opper and R. Urbanczik, "Universal learning curves of support vector machines," *Phys. Rev. Lett.*, vol. 86, no. 19, 2001, Art. no. 4410.

[186] S. Spigler et al., "A jamming transition from under-to over-parametrization affects generalization in deep learning," *J. Phys. A*, vol. 52, no. 47, 2019, Art. no. 474001.

[187] M. S. Advani and A. M. Saxe, "High-dimensional dynamics of generalization error in neural networks," 2017, *arXiv:1710.03667*.

[188] T. Hastie et al., "Surprises in high-dimensional ridgeless least squares interpolation," 2019, *arXiv:1903.08560*.

[189] S. d'Ascoli et al., "Triple descent and the two kinds of overfitting: Where & why do they appear?," 2020, *arXiv:2006.03509*.

[190] M. Loog and R. P. W. Duin, "The dipping phenomenon," in *Proc. Joint IAPR Int. Workshops Statist. Techn. Pattern Recognit. Struct. Syntactic Pattern Recognit.*, 2012, pp. 310–317.

[191] S. Ben-david et al., "Minimizing the misclassification error rate using a surrogate convex loss," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1863–1870.

[192] M. Loog et al., "On measuring and quantifying performance: Error rates, surrogate loss, and an example in semi-supervised learning," in *Handbook of PR and CV*. Singapore: World Scientific, 2016.

[193] P. L. Bartlett et al., "Large margin classifiers: Convex loss, low noise, and convergence rates," in *Proc. 16th Int. Conf. Neural Inf. Process. Syst.*, 2004, pp. 1173–1180.

[194] J. Vanschoren et al., "Learning from the past with experiment databases," in *Pacific Rim Int. Conf. AI*, 2008, pp. 485–496.

[195] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 839–846.

[196] K. Konyushkova et al., "Introducing geometry in active learning for image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2974–2982.

[197] M. Loog and Y. Yang, "An empirical investigation into the inconsistency of sequential active learning," in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 210–215.

[198] Z. Wang, Z. Dai, B. Póczos, and J. Carbonell, "Characterizing and avoiding negative transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11 293–11 302.

[199] K. Weiss et al., "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, 2016, Art. no. 9.

[200] W. M. Kouw and M. Loog, "A review of domain adaptation without target labels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 766–785, Mar. 2021.

[201] M. Loog et al., "Minimizers of the empirical risk and risk monotonicity," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 7478–7487.

[202] A. Power et al., "Grokking: Generalization beyond overfitting on small algorithmic datasets," 2022, *arXiv:2201.02177*.

[203] P. Sollich, "Gaussian process regression with mismatched models," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst.*, 2002, pp. 519–526.

[204] P. Sollich, "Can gaussian process regression be made robust against model mismatch?," in *Proc. Deterministic Statist. Methods Mach. Learn.*, 2004, pp. 199–210.
[205] P. Grünwald and T. van Ommen, "Inconsistency of Bayesian inference for misspecified linear models, and a proposal for repairing it," *Bayesian Anal.*, vol. 12, no. 4, pp. 1069–1103, 2017.
[206] P. D. Grünwald and W. Kotłowski, "Bounds on individual risk for log-loss predictors," *J. Mach. Learn. Res.*, vol. 19, pp. 813–816, 2011.
[207] T. J. Viering et al., "Making learners (more) monotone," in *Proc. Int. Symp. Intell. Data Anal.*, 2020, pp. 535–547.
[208] Z. Mhammedi and H. Husain, "Risk-monotonicity in statistical learning," 2020, *arXiv:2011.14126*.
[209] O. J. Bousquet et al., "Monotone learning," in *Proc. 35th Annu. Workshop Comput. Learn. Theory*, 2022, pp. 842–866.
[210] V. Pestov, "A universally consistent learning rule with a universally monotone error," *J. Mach. Learn. Res.*, vol. 23, no. 157, pp. 1–27, 2022.
[211] D. J. Hand, *Construction and Assessment of Classification Rules*. Hoboken, NJ, USA: Wiley, 1997.
[212] R. B. Anderson and R. D. Tweney, "Artifactual power curves in forgetting," *Memory Cogn.*, vol. 25, no. 5, pp. 724–730, 1997.
[213] A. Heathcote et al., "The power law repealed: The case for an exponential law of practice," *Psychon. Bull. Rev.*, vol. 7, no. 2, pp. 185–207, 2000.
[214] A. Mey, "A note on high-probability versus in-expectation guarantees of generalization bounds in machine learning," 2020, *arXiv:2010.02576*.
[215] S. M. Stigler, "The epic story of maximum likelihood," *Statist. Sci.*, vol. 22, no. 4, pp. 598–620, 2007.

Tom Viering received the MSc degree in computer science from the Delft University of Technology, in 2016 and is currently working toward the PhD degree with the Pattern Recognition Laboratory there. His principal research interests are statistical learning theory and its application to problems such as active learning, domain adaptation, and semi-supervised learning. Other topics that interest him are non-monotonicity of learning, interpretable machine learning, generative models, online learning, and deep learning.

Marco Loog received the MSc degree in mathematics from Utrecht University, and the PhD degree from the Image Sciences Institute. He worked as a scientist with the IT University of Copenhagen, the University of Copenhagen, and Nordic Bioscience. He now is with the Delft University of Technology to research and to teach. He is an honorary professor with the University of Copenhagen. His principal research interest is with supervised learning in all sorts of shapes and sizes.