Learning Curve Theory

Marcus Hutter

DeepMind

http://www.hutter1.net/

February 14, 2021

Abstract

Recently a number of empirical "universal" scaling law papers have been published, most notably by OpenAI. 'Scaling laws' refers to power-law decreases of training or test error w.r.t. more data, larger neural networks, and/or more compute. In this work we focus on scaling w.r.t. data size n. Theoretical understanding of this phenomenon is largely lacking, except in finite-dimensional models for which error typically decreases with $n^{-1/2}$ or n^{-1}, where n is the sample size. We develop and theoretically analyse the simplest possible (toy) model that can exhibit $n^{-\beta}$ learning curves for arbitrary power $\beta > 0$, and determine to which extent power laws are universal or depend on the data distribution or loss function: Roughly, learning curves exhibit a power law with $\beta = \frac{\alpha}{1+\alpha}$ for α-Zipf-distributed data independent of the choice of loss. Furthermore, noise rapidly deteriorates/improves in instantaneous/time-averaged learning curves for increasing n, suggesting that model selection should be based on cumulative (AUC) or time-averaged error, not final test error.

Contents

Keywords

Power Law, Scaling, Learning Curve, Theory, Data Size, Error, Loss, Zipf, ...

1 Introduction

Power laws in large-scale machine learning. The 'mantra' of modern machine learning is 'bigger is better'. The larger and deeper Neural Networks (NNs) are, the more data they are fed, the longer they are trained, the better they perform. Apart from the problem of overfitting [BHM18] and the associated recent phenomenon of double-descent [BHMM19], this in itself is rather unsurprising. But recently 'bigger is better' has been experimentally quantified [JN17], most notably by Baidu [HNA$^+$17] and OpenAI [HKK$^+$20, KMH$^+$20, HKHM21].

They observe that the *error* or *test loss* decreases as a power law, with the *data size*, with the *model size* (number of NN parameters), as well as with the *compute budget* used for training, assuming one factor is not "bottlenecked" by the other two factors. If all three factors are increased appropriately in tandem, the (reducible) loss has power-law scaling over a very wide range of data/model size and compute budget.

If there is intrinsic noise in the data (or a non-vanishing model mis-specification), the loss can never reach zero, but at best can converge to the intrinsic entropy of the data (or the intrinsic representation=approximation error). When we talk about *error*, sometimes also called excess error or reducible error, we mean test loss with this potential offset subtracted, similar to average regret in online learning.

Ubiquity/universality of power laws. Power laws have been observed for many problem types (supervised, unsupervised, transfer learning) and data types (images, video, text, even math) and many NN architectures (Transformers, ConvNets, ...) [HNA$^+$17, RRBS19, HGLS20, HKK$^+$20, KMH$^+$20]. This has led some to the belief that power laws might be universal: Whatever the problem, data, model, or learning algorithm, learning curves follow power laws. To which extent this conjecture is true, we do not know, since theoretical understanding of this phenomenon is largely lacking.

This work. In this work we focus on scaling with data size n. As explained below, any reasonable finitely-parameterized model and reasonable loss function leads to a scaling law $n^{-\beta}$ with $\beta = \frac{1}{2}$ or $\beta = 1$, but not the observed $\beta \approx 0.05...0.35 < \frac{1}{2}$. We therefore conjecture that any theoretical explanation of power laws for a variety of β (beyond 0-1 and absolute error implying $\beta = \frac{1}{2}$ and locally-quadratic loss implying $\beta = 1$) requires real-world data and models of unbounded complexity, that is, no finite-dimensional model can "explain" all information in the data.

Possible modelling choices are (a) scaling up the model with data, as done in the experiments, or (b) consider non-parametric models (e.g. Gaussian processes), or (c) a model with (countably-)infinitely-many parameters. We choose (c) for mathematical simplicity compared to (b), and because (c) clearly separates scaling with data from scaling with model size, unlike (a). In future, (a) and (b) should definitely also be pursued, in particular, since we have no indication that our findings transfer.

Within our toy model, we show that for domains of unbounded complexity, a large variety of learning curves are possible, including *non*-power-laws. It is plausible that this remains true for most infinite models. Real data is often Zipf distributed (e.g. the frequency of words in text), which is itself a power law. We show that this, in our toy model, implies power law learning curves with "interesting" β, though most (even non-Zipf) distributions *also* lead to power laws but with "uninteresting" β.

Key findings. Valid within our toy model, and potentially beyond,

(I) our primary finding is that learning curves consist of 3 terms, (1) a theoretically well-understood loss-dependent power law (usually $n^{-1/2}$ or n^{-1}) or exponential decay in the deterministic instantaneous case, (2) a data-dependent loss-independent power law $n^{-\beta}$ for $0 < \beta \leq 1$, with (typically small) $\beta = \frac{\alpha}{1+\alpha}$ for α-Zip-distributed data, and (3) an irreducible term due to noise (and model approximation error in more general models).

(II) a secondary finding is that the signal-to-noise ratio rapidly deteriorates/improves with n in instantaneous/time-averaged learning curves. Though we do not study model selection per se in this work, augmented by arguments laid out by [Hut06] for log-loss, this suggests that model selection should be based on cumulative (AUC) or time-averaged error, rather than final test error.

Contents. In Section 2 we review some scaling theory (with model size, compute, and data size) we are aware of, and discuss their relations. In Section 3 we introduce our setup: classification with countable "feature" space and a memorizing algorithm, the simplest model and algorithm we could come up with that exhibits interesting/relevant scaling behavior. In Section 4 we derive and discuss general expressions for expected learning curves and for various specific data distributions: finite, Zipf, exponential, and beyond, many but not all lead to power laws. In Section 5 we estimate the uncertainty in empirical learning curves. We show that the signal-to-noise ratio deteriorates with n, which implies that many (costly) runs need to be averaged in practice to get a smooth learning curve. On the other hand, the signal-to-noise ratio of the time-averaged learning curves tends to infinity, hence even a single run suffices for large n. In Section 6 we perform some simple control experiments to confirm and illustrate the theory and claims, and the accuracy of the theoretical expressions. In Section 7 we discuss (potential) extensions of our toy model towards a more comprehensive and realistic theory of scaling laws: noisy labels, other loss functions, continuous features, models that generalize, and deep learning. Section 8 concludes with limitations and potential applications. Appendix A discusses losses beyond 0-1 loss. Appendix B contains derivations and approximations of the expected error. Appendix C derives exact and approximate expressions for the time-averaged variance. Appendix D proves the n^{-1} upper bound on the learning curve for sub-exponential distributions. Appendix E considers noisy labels. Appendix F derives an approximation of sums by integrals, tailored to our purpose. Appendix G lists notation. Appendix H contains some mores plots.

2 Scaling Theory

Scaling with model size. Consider a function $f : [0; 1]^d \to \mathbb{R}$ which we wish to approximate. A naive approximation is to discretize the hyper-cube to an ε-grid. This constitutes a model with $m = (1/\varepsilon)^d$ parameters, and if f is 1-Lipschitz, can approximate f to accuracy $\varepsilon = m^{-1/d}$, i.e. the (absolute) error scales with model size m as a power law with exponent $-1/d$. More generally, there exist (actually linear basis function) models with m parameters that can approximate all functions f whose first k derivatives are bounded to accuracy $O(m^{-k/d})$ [Mha96], again a power law, and without further assumptions, no reasonable model can do better [DHM89]; see [Pin99] for reformulations and discussions of these results in the context of NNs. Not being aware of this early

theoretical work, this scaling law has very recently been empirically verified and extended by [SK20]. Instead of naively using the input dimension d, they determine and use the (fractal) dimension of the data distribution in the penultimate layer of the NN.

Scaling with compute. Most NNs are trained by some form of stochastic gradient descent, efficiently implemented in the form of back-propagation. Hence compute is proportional to number of iterations i times batch-size times model size. So studying the scaling of error with the number of iterations tells us how error scales with compute. The loss landscape of NNs is highly irregular, which makes theoretical analyses cumbersome at best. At least asymptotically, the loss is locally convex, hence the well-understood stochastic (and online) convex optimization could be a first (but possibly misleading) path to search for theoretical understanding of scaling with compute. The error of most stochastic/online optimization algorithms scales as a power law $i^{-1/2}$ or i^{-1} for convex functions [Bub15, Haz16].

Scaling with data size. This is the traditional domain of Statistical Learning Theory (SLT) [SB14], online learning [GPS18], and online convex optimization [Haz16]. The fundamental (PAC) theorem of SLT states that the empirical error converges to the generalization error at a rate of $n^{-1/2}$ for models of finite VC-dimension, and n i.i.d. samples. Most results in SLT are applications, adaptations, or generalizations of this result, all keeping the $n^{-1/2}$ rate. Most models have finite VC-dimension (SVMs, regression, NNs, finite decision trees, ...). The core theorem applies to Empirical Risk Minimization (ERM) algorithms, but extends to e.g. (stochastic) gradient descent approximations. It generalizes to convex-Lipschitz-bounded and convex-smooth-bounded losses. Rates improve to n^{-1} for β-smooth ("locally quadratic") losses. The average regret considered in online learning theory and online convex optimization has similar requirements on the model (e.g. finite-dimensional) and exhibits the same rates $n^{-1/2}$ or n^{-1} (or $\frac{1}{n}\ln n$ due to the time-average), under similar conditions. An example of a *non*-parametric model whose sample complexity has been analysed with "interesting" rate, is k-nearest neighbors (kNN). For d-dimensional Lipschitz functions, the error of kNN is bounded by $n^{-1/(d+1)}$ [SB14, Thm.19.3&19.5]. This power law and the intuition for the power $-1/(d+1) \approx -1/d$ is similar to [Pin99, SK20] discussed above but in terms of model size. Again, the origin of the power law in our model is very different.

We are not aware of any successful attempts to theoretically explain the power laws observed by OpenAI with $\beta < 1/2$. [Cho20] and [HNA+17] consider a very simple Bernoulli model: Essentially they observe that the Bernoulli parameter can be estimated to accuracy $1/\sqrt{n}$ from n i.i.d samples, i.e. the absolute loss (also) scales with $1/\sqrt{n}$ [HNA+17] and the log-loss or KL-divergence scales with $1/n$ [Cho20]. Indeed, the latter holds for any loss, locally quadratic at the minimum, so is not at all due to special properties of KL as [Cho20] suggests. These observations are simple instantiations of the above statistical learning theory results, and also trivially follow from the central limit theorem for virtually any finitely-parameterized model in the under-parameterized regime of more-data-than-parameters. This is of course always the case for their Bernoulli model, which only has one parameter, but *not* for the over-parameterized regime some modern NNs work in. Anyway, the scaling laws identified by OpenAI et al. are $n^{-\beta}$, for various $\beta < 1/2$, which neither the Bernoulli nor any finite-dimensional model can explain.

Data size vs iterations vs compute. Above we have used the fact that compute is (usually in deep learning) proportional to number of learning iterations, provided batch and model size are kept fixed. In addition,

(i) in *online learning*, every data item is used only once, hence the size of data used up to iteration n is proportional to n.

(ii) This is also true for *stochastic learning* algorithms for some recent networks, such as GPT-3, trained on massive data sets, where every data item is used at most once (with high probability).

(iii) When generating *artificial data*, it is natural to generate a new data item for each iteration.

Hence in all of these 3 settings, the *learning curves*, error-with-data-size, error-with-iterations, and error-with-compute, are scaled versions of each other. For this reason, scaling of error with iterations, also tells us how error scales with data size and even with compute, but scaling with model size is different.

3   Setup

We formally introduce our setup, model, algorithm, and loss function in this section: We consider classification problems with 0-1 loss and countable feature space. A natural practical example application would be classifying words w.r.t. some criterion. Our toy model is a deterministic classifier for features/words sampled i.i.d. w.r.t. to some distribution. Our toy algorithm predicts/recalls the *class* for a new *feature* from a previously observed (*feature,class*) pair, or acts randomly on a novel *feature*. The probability of an erroneous prediction is hence proportional to the probability of observing a new feature, which formally is equivalent to the model in [Cha81]. The usage and analyses of the model and resulting expressions are totally different though. While [Cha81]'s aim is to develop estimators for the probability of discovering a new species from data whatever the unknown true underlying probabilities, we are interested in the relationship between the true probability distribution of the data and the resulting learning curves, i.e. the scaling of expected (averaged) error with sample size. In Appendix A we show that, within a for our purpose irrelevant multiplicative constant, the results also apply to most other loss functions.

The toy model.   The goal of this work is to identify and study the simplest model that is able to exhibit power-law learning curves as empirically observed by [HNA$^+$17, HKK$^+$20, KMH$^+$20] and others. Consider a classification problem $h \in \mathcal{H} := \mathcal{X} \to \mathcal{Y}$, e.g. $\mathcal{Y} = \{0, 1\}$ for binary classification, where classifier h is to be learnt from data $\mathcal{D}_n := \{(x_1, y_1), ..., (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$. For finite $\mathcal{X}$ and $\mathcal{Y}$, this is a finite model class ($|\mathcal{H}| < \infty$), which, as discussed above, can only exhibit a restrictive range of learning curves, typically $n^{-1}/n^{-1/2}/e^{-O(n)}$ for locally-quadratic/absolute/0-1 error. In practice, $\mathcal{X}$ is often a (feature) vector space $\mathbb{R}^d$, which can support an infinite model class ($|\mathcal{H}| = \infty$) (e.g. NNs) rich enough to exhibit (at least empirically) $n^{-\beta}$ scaling for many different $\beta \notin \{\frac{1}{2}, 1\}$, typically $\beta \ll 1$. The smallest potentially suitable $\mathcal{X}$ would be countable, e.g. $\mathbb{N}$, which we henceforth assume. The model class $\mathcal{H} := \mathbb{N} \to \mathcal{Y}$ is uncountable and has infinite VC-dimension, hence is not PAC learnable, but nevertheless can be learnt consistently. Furthermore, for simplicity we assume that data $\mathcal{D}_n := \{(i_1, y_1), ..., (i_n, y_n)\} \equiv (i_{1:n}, y_{1:n})$ with "feature" $i_t \in \mathbb{N}$ "labelled" y_t is noise-free = deterministic, i.e. $y_t = y_{t'}$ if $i_t = i_{t'}$. Let $h_0 \in \mathcal{H}$ be the unknown true labelling function. We discuss relaxations of some of these assumptions later in Section 7, in particular extension to other loss function in Appendix A and noisy labels in Appendix E. Let features i_t be drawn i.i.d. with $\mathbb{P}[i_t = i] =: \theta_i \geq 0$ and (obviously) $\sum_{i=1}^{\infty} \theta_i = 1$. The infinite vector

$\boldsymbol{\theta} \equiv (\theta_1, \theta_2, ...)$ characterizes the feature distribution. The labels are then determined by $y_t = h_0(i_t)$.

The toy algorithm. We consider a simple tabulation learning algorithm $A : \mathbb{N} \times (\mathbb{N} \times \mathcal{Y})^* \to \mathcal{Y}$ that stores all past labelled features $\mathcal{D}_n$ and on next feature $i_{n+1} = i$ recalls y_t if $i_t = i$, i.e. feature i has appeared in the past, or outputs, in its simplest instantiation, *undefined* if $i \notin i_{1:n}$ i.e. is new. Formally:

$$A(i, \mathcal{D}_n) := \begin{cases} y_t & \text{if} \quad i = i_t \text{ for some } t \leq n \\ \perp & \text{else} \quad \text{i.e. if } i \notin i_{1:n} \end{cases} \tag{1}$$

Error. Algorithm A only makes an error predicting label y_{n+1} if $i \notin i_{1:n}$. We say A makes 1 unit of error in this case. Formally, the *(instantaneous) error* E_n of algorithm A when predicting y_{n+1} from $\mathcal{D}_n$ is defined as

$$\mathsf{E}_n := [\![i_{n+1} \notin i_{1:n}]\!]$$

The expectation of this w.r.t. to the random choice of $\mathcal{D}_n$ and i_{n+1} gives the *expected (instantaneous) error*

$$\mathbb{E}_n := \mathbb{E}[\mathsf{E}_n] = \mathbb{P}[i_{n+1} \notin i_{1:n}] = \sum_{i=1}^{\infty} \theta_i (1 - \theta_i)^n \tag{2}$$

A formal derivation is given in Appendix B, but the result is also intuitive: If feature i has not been observed so far (which happens with probability $(1-\theta_i)^n$), and then feature i is observed (which happens with probability θ_i), the algorithm makes an error. $\mathbb{E}_n$ as a function of n constitutes an (expected) learning curve, which we will henceforth study. In Appendix A we show that expression (2) remains valid within an irrelevant multiplicative constant for most other loss functions.

4 Expected Learning Curves

We now derive theoretical expected learning curves for various underlying data distributions. We derive exact and approximate, general and specific expressions for the scaling of *expected* error with sample size. Specifically we consider finite models, which lead to exponential error decay, and infinite Zipf distributions, which lead to interesting power laws with power $\beta < 1$. Interestingly even highly skewed data distributions lead to power laws, albeit with "uninteresting" power $\beta = 1$.

Exponential decay. In the simplest case of m of the θ_i being equal and the rest being 0, the error $\mathbb{E}_n = (1 - \frac{1}{m})^n \leq e^{-n/m}$ decays exponentially with n. This is not too interesting to us, since

   (a) this case corresponds to a finite model (see below),
   (b) exponential decay is an "artifact" of the deterministic label and discontinuous 0-1 error (it can also happen for stochastic labels with a majority predictor, but not the more common averaging or stochastic predictor),
   (c) will become a power law $1/n$ after time-averaging (see Section 5).
   (d) most importantly for us, does not explain the Deep Learning power law learning curves.

Superposition of exponentials. Since (2) is invariant under bijective renumbering of features $i \in \mathbb{N}$, we can w.l.g. assume $\theta_1 \geq \theta_2 \geq \theta_3 \geq ....$ Some θs may be equal. If we group equal θs together into $\bar{\bar{\theta}}_j$ with multiplicity $m_j > 0$ and define $\bar{\bar{\vartheta}}_j := -\ln(1 - \bar{\bar{\theta}}_j)$, then

$$\mathbb{E}_n = \sum_{j=1}^{M} m_j \bar{\bar{\theta}}_j e^{-n\bar{\bar{\vartheta}}_j} \qquad (3)$$
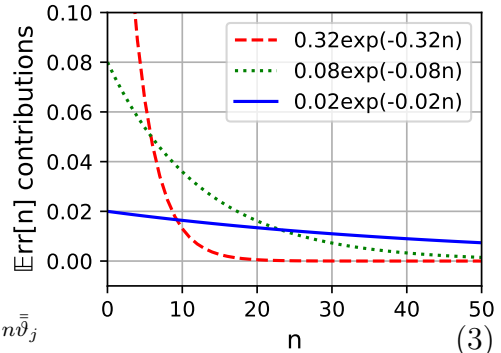
where $M \in \mathbb{N} \cup \{\infty\}$ is the number of *different* $\theta_i > 0$. This is a superposition of exponentials in n (note that $\sum_{j=1}^{M} m_j \bar{\bar{\theta}}_j = 1$) with different decay rates $\bar{\bar{\vartheta}}_j$. If different $\bar{\bar{\theta}}_j$ have widely different magnitudes and/or for suitable multiplicities m_j, the sum will be dominated by different terms at different "times" n. So there will be different phases of exponential decay, starting with fast decay $e^{-n\bar{\bar{\vartheta}}_1}$ for small n, taken over by slower decay $e^{-n\bar{\bar{\vartheta}}_2}$ for larger n, and $e^{-n\bar{\bar{\vartheta}}_3}$ for even larger n, etc. though some terms may never (exclusively) dominate, or phases may be unidentifiably muddled together (see figure above). In any case, if $M = \infty$, the dominant terms shift indefinitely to ever smaller θ for ever larger n. For $M < \infty$ eventually $e^{-n\bar{\bar{\vartheta}}_M}$ for the smallest $\bar{\bar{\vartheta}}$ will dominate $\mathbb{E}_n$. The same "caveats" (a)-(d) apply as for $M = 1$ in the previous paragraph.

Approximations. First, in our subsequent analysis we (can) approximate $(1 - \theta_i)^n =: e^{-n\vartheta_i} \approx e^{-n\theta_i}$, justified as follows: (i) For $n\theta_i \ll 1$ this is an excellent approximation. (ii) For $\theta_i \ll 1$, $\vartheta_i \approx \theta_i$, while numerically $e^{-n\vartheta_i}/e^{-n\theta_i} \not\approx 1$ for $n\theta_i \gg 1$, but the exponential scaling of $e^{-n\vartheta_i}$ and $e^{-n\theta_i}$ we care about is sufficiently similar. (iii) There can only be a finite number of $\theta_i \not\ll 1$, say, θ_i for $i \leq i_0$ are not small, then already for moderately large $n \gg 1/\theta_{i_0}$, all features $i \leq i_0$ are observed with high probability and hence do not contribute (much) to the expected error (formally $e^{-n\theta_i} \ll 1$ for $i \leq i_0$) hence can safely be ignored in any asymptotic analysis.

Second, let $f : \mathbb{R} \to \mathbb{R}$ be a smooth and monotone decreasing interpolation of $\boldsymbol{\theta} : \mathbb{N} \to \mathbb{R}$, i.e. $f(i) := \theta_i$ and $f'(x) \leq 0$. We can then approximate the error as follows:

$$\mathbb{E}_n = \sum_{i=1}^{\infty} f(i)(1 - f(i))^n \approx \int_1^{\infty} f(x)e^{-nf(x)}dx$$
$$= \int_0^{\theta_1} \frac{ue^{-nu}du}{|f'(f^{-1}(u))|} \stackrel{\times}{\approx} \frac{1}{n^2|f'(f^{-1}(\frac{1}{n}))|} \qquad (4)$$

The first $\approx$ uses the two approximations introduced above. The equality follows from a reparametrization $u = f(x)$ and $f(1) = \theta_1$ and $f(\infty) = 0$ and $dx = du/f'(x)$ and $f' < 0$. The numerator ue^{-nu} is maximal and (strongly) concentrated around $u = 1/n$, hence $u \approx 1/n$ gives most of the integral's contribution. Therefore replacing u by $1/n$ in the denominator can be a reasonable approximation. The last $\approx$ follows from this and $\int_0^{\theta_1} ue^{-nu}du \approx \int_0^{\infty} ue^{-nu}du = 1/n^2$ for $n\theta_1 \gg 1$.

Intuitively, the expected error (2) is dominated by samples i_0 for which $\theta_{i_0} \approx \frac{1}{n}$. Estimating the number of such i_0 multiplied by $\theta_{i_0}(1 - \theta_{i_0})^n \approx \theta_{i_0}e^{-1}$ leads to approximation (4). In Appendix B we show that the approximation error of the integral representation is bounded by $1/en + o(1/n)$.

Zipf-distributed data. Empirically many data have been observed to have a power-law distribution, called Zipf distribution in this context, that is, for a countable domain the

7

frequency of the ith most frequent item is approximately proportional to $i^{-(\alpha+1)}$ for some $\alpha > 0$. In our model this will be the case if $\theta_i \propto i^{-(\alpha+1)}$, so let $u = f(x) = \alpha \cdot x^{-(\alpha+1)}$. This implies $x = f^{-1}(u) = (u/\alpha)^{-1/(1+\alpha)}$ and $f'(x) = -\alpha(\alpha+1)x^{-(\alpha+2)} = -\alpha(\alpha+1)(u/\alpha)^{(\alpha+2)/(\alpha+1)}$, hence

$$\mathbb{E}_n \;\approx\; \frac{1}{n^2|f'(f^{-1}(\frac{1}{n}))|} \;\overset{\times}{=}\; n^{-\beta}, \qquad\qquad \text{where} \quad \beta := \frac{\alpha}{1+\alpha}$$

That is, Zipf-distributed data (with power $\alpha+1$) lead to a power-law learning curve (with power $\beta = \frac{\alpha}{1+\alpha} < 1$). The more accurate integral representation leads to the same power law but with correct coefficient $\mathbb{E}_n \approx c_\alpha n^{-\beta}$ with $c_\alpha = \alpha^{1/(\alpha+1)}\Gamma(\frac{\alpha}{1+\alpha})/(\alpha+1)$. $c_1 = \frac{1}{2}\sqrt{\pi} \doteq 0.886$ and $c_{0.1} = 1.177$ in excellent agreement with the fit curves in Figure 2.

Exponentially-distributed data. An exponential data distribution $\theta_i \propto e^{-\gamma i}$ is more skewed than any power law. For $u = f(x) = \gamma \cdot e^{-\gamma x}$ we have $x = f^{-1}(u) = \frac{1}{\gamma}\ln\frac{\gamma}{u}$ and $f'(x) = -\gamma^2 e^{-\gamma x} = -\gamma u$, hence both approximations in (4) give $1/\gamma n$. A rigorous upper bound $\mathbb{E}_n \overset{\times}{\le} (\frac{1}{e} + \frac{1}{\gamma})\frac{1}{n} + o(\frac{1}{n})$ follows from (11) in Appendix B, and a rigorous lower bound $\mathbb{E}_n \overset{\times}{\ge} \frac{1}{n}$ from the next paragraph. So even an exponential data distribution leads to a power law learning curve, though the exponent 1 is much larger than observed in (most) experiments, which hints at that data are not exponentially distributed, assuming this toy model has any real-world relevance.

Beyond exponentially-distributed data. For (quite unrealistic) decay faster than exponential, e.g. $\theta_i \propto e^{-\gamma i^2}$, the approximations (4) are too crude, but somewhat surprisingly we *always* get a (sort of) power law as long as $\theta_i > 0$ for infinitely many i. First, the previous paragraph implies that $\mathbb{E}_n \overset{\times}{\le} n^{-1}$ for any $\theta_i \overset{\times}{\le} e^{-\gamma i}$ for any $\gamma > 0$, i.e. the error decreases at least with n^{-1} if the ith item has at most exponentially small probability in i. For a (partial) converse, define $n_i := \lceil 1/\vartheta_i \rceil \le 1/\vartheta_i + 1$. Plugging $\vartheta_i := -\ln(1-\theta_i) \ge \theta_i$ and $2\theta_i \ge \vartheta_i \ge 1/n_i$ for $\theta_i < 0.79$ into $\mathbb{E}_n \ge \theta_i(1-\theta_i)^n = \theta_i e^{-n\vartheta_i}$ we get

$$\mathbb{E}_{n_i} \;\ge\; \theta_i e^{-n_i\vartheta_i} \;\ge\; \frac{1}{2n_i}e^{-(1/\vartheta_i+1)\vartheta_i} \;\ge\; \frac{1}{2n_i e^2} \;\overset{\times}{=}\; n_i^{-1}$$

Hence, if $\theta_i > 0$ for infinitely many i, then there are infinitely many n for which $\mathbb{E}_n \overset{\times}{\ge} n^{-1}$. For θ_i going to zero exponentially or slower, the spacing between n_{i+1} and n_i has bounded ratio $n_{i+1}/n_i \overset{\times}{\le} e^\gamma$, which implies $\mathbb{E}_n \overset{\times}{\ge} n^{-1}$ for all n. For faster decaying θ_i, e.g. $\theta_i = e^{-\gamma i^2}$ this is no longer the case. So in some weak sense, power law learning curves are universal, but it's mostly $1/n$, so not useful to explain observed power laws.

5 Learning Curve Variance

So far we have considered *expected* learning curves. This corresponds to averaging infinitely many experimental runs. In practice, only finitely many runs are possible, sometimes as few as 5 or even 1. In the following we consider the variance $\mathbb{V}_n$ of the instantaneous error $\mathsf{E}_n = [\![i_{n+1} \notin i_{1:n}]\!]$ as a function of n. The standard error when averaging k runs is then $\sqrt{\mathbb{V}_n/k}$. The question we care most about here is whether (for large n) this is small or large compared to $\mathbb{E}_n$, because this determines whether learning curves (for small k) are smooth or look random, and how large k suffices for a good signal-to-noise ratio. We also consider time-average expected learning curves and their variance, which are much smoother. Note that cumulative errors are like attenuated drifting random walks, a property time-average learning curves qualitatively inherit (red curve in Figure 1 right).

Instantaneous Variance. $\mathsf{E}_n \in \{0, 1\}$, hence $\mathsf{E}_n^2 = \mathsf{E}_n$, hence

$$\mathbb{E}[\mathsf{E}_n^2] = \mathbb{E}[\mathsf{E}_n] = \sum_{i=1}^{\infty} \theta_i (1 - \theta_i)^n =: \mu_n, \quad \text{hence}$$

$$\mathbb{V}[\mathsf{E}_n] = \mathbb{E}[\mathsf{E}_n^2] - \mathbb{E}[\mathsf{E}_n]^2 = \mu_n(1 - \mu_n)$$

Since $\mu_n \to 0$ for $n \to \infty$, the standard deviation

$$\sigma_n := \sqrt{\mathbb{V}[\mathsf{E}_n]} = \sqrt{\mu_n(1 - \mu_n)} \approx \sqrt{\mu_n} \gg \mu_n = \mathbb{E}_n$$

That is, the standard deviation is much larger than then mean for large n. Indeed, for a single run, there is no proper learning *curve* at all, since $\mathsf{E}_n \in \{0, 1\}$ (see Figures 4&5 top left). In order to get a good signal-to-noise ratio one would need to average a large (and indeed increasing with n) number $k \gg 1/\mu_n$ of runs (see Figures 1&4&5).

Time-averaged Mean and Variance. In practice, beyond averaging over runs, other averages are performed to reduce noise. One alternative is to report the time-averaged error $\overline{\mathsf{E}}$, rather than the instantaneous error E. We can calculate its mean and variance as follows

$$\overline{\mathsf{E}}_N := \frac{1}{N} \sum_{n=0}^{N-1} \mathsf{E}_n \tag{5}$$

$$\mathbb{E}[\overline{\mathsf{E}}_N] = \frac{1}{N} \sum_{n=0}^{N-1} \mathbb{E}[\mathsf{E}_n] = \frac{1}{N} \sum_{i=1}^{\infty} \theta_i \sum_{n=0}^{N-1} (1 - \theta_i)^n \overset{(a)}{=} \frac{1}{N} \sum_{i=1}^{\infty} 1 - (1 - \theta_i)^N \tag{6}$$

$$\mathbb{E}[\overline{\mathsf{E}}_N^2] \overset{(b)}{=} \frac{1}{N^2} \sum_{i=1}^{\infty} [1 - (1 - \theta_i)^N] + \frac{1}{N^2} \sum_{i \neq j} [1 - (1 - \theta_i)^N - (1 - \theta_j)^N + (1 - \theta_i - \theta_j)^N]$$

$$\mathbb{V}[\overline{\mathsf{E}}_N] \overset{(c)}{=} \frac{1}{N^2} \sum_{i=1}^{\infty} (1 - \theta_i)^N [1 - (1 - \theta_i)^N] - \frac{1}{N^2} \sum_{i \neq j} [(1 - \theta_i)^N (1 - \theta_j)^N - (1 - \theta_i - \theta_j)^N]$$

$$\tag{7}$$

where (a) is simple algebra, (b) follows from inserting the definition of $\overline{\mathsf{E}}_N$ and some rather tedious algebraic manipulations (see Appendix C), and (c) from inserting (a) and (b) into the definition of variance and simple algebraic manipulation. We now revisit the exponential and Zipf case studied earlier, after a trivial but note-worthy observation.

Case $\theta_i = [\![i = 1]\!]$. In this case $i_n = 1 \; \forall n$, hence $\mathsf{E}_0 = 1$ and $\mathsf{E}_n = 0 \; \forall n \geq 1$ and $\mathbb{V}[\mathsf{E}_n] = 0 \; \forall n$. This is the fastest any error can decay, 0 after 1 observation, hence the fastest any time-averaged error can decay is $\overline{\mathsf{E}}_N = 1/N$. This means, for any learning problem that has instantaneous error decaying faster than $1/N$, one should report the instantaneous error rather than the slower decaying and hence much larger time-averaged error. Most problems of interest in Deep Learning have much slower learning curves though, and for those, the time-averaged and the instantaneous error have the *same* decay rate, but the time-averaged has lower variance, so is the preferred one to plot or report.

Case $\theta_i = \frac{1}{m}[\![i \leq m]\!]$. In this case, while $\mathbb{E}_n = (1 - \frac{1}{m})^n \approx e^{-n/m}$ decays exponentially,

9

the average quantities decay with $1/N$ (or $1/N^2$):

$$
\begin{aligned}
\mathbb{E}[\overline{\mathsf{E}}_N] &= \tfrac{m}{N}[1 - (1 - \tfrac{1}{m})^N] \longrightarrow \tfrac{m}{N} \quad \text{for} \quad N \to \infty \\
\mathbb{V}[\overline{\mathsf{E}}_N] &= \tfrac{m^2}{N^2}[\tfrac{1}{m}(1 - \tfrac{1}{m})^N + (1 - \tfrac{1}{m})(1 - \tfrac{2}{m})^N - (1 - \tfrac{1}{m})^{2N}] \\
&\to \tfrac{m}{N^2}[e^{-N/m} - e^{-2N/m}] \to \tfrac{m}{N^2}e^{-N/m} \quad \text{for} \quad N \to \infty \\
\sigma[\overline{\mathsf{E}}_N] &\approx \tfrac{\sqrt{m}}{N}e^{-N/2m} \quad \ll \quad \tfrac{m}{N} \approx \mathbb{E}[\overline{\mathsf{E}}_N] \quad \text{for} \quad N \gg m
\end{aligned}
$$

The expressions for the mean and variance follow from the general expression (6) and (7) above, by inserting $\theta_i = \tfrac{1}{m}$ for $i \leq m$ and noting that $i > m$, for which $\theta_i = 0$, give no contribution, hence $\sum_i 1 = m$ and $\sum_{i \neq j} 1 = m(m-1)$, and rearranging terms. The conclusion most interesting to us is that the standard deviation is (much) smaller than the mean for N (much) larger than m, so the time-averaged learning curves have a much better signal-to-noise ratio; see Figure 1. Also $\sigma[\overline{\mathsf{E}}_N] \approx N^{-1/2} \ll 1 \approx \mathbb{E}[\overline{\mathsf{E}}_N]$ for $m \gg N$. The intuition is easy: For $m \gg N$, at ever $n < N$, a new i_{n+1} is observed, i.e. $\mathsf{E}_n = 1$ w.h.p. For $N \gg m$, all m errors have been made, i.e. $\overline{\mathsf{E}}_N = \tfrac{m}{N}$ w.h.p, i.e. in both cases the variance is small. Only for $m \approx N$ is there sizeable uncertainty in E_N. The situation is similar for the most interesting Zipf case:

Case $\theta_i \propto i^{-(\alpha+1)}$. Recall that for Zipf-distribution $\theta_i \propto i^{-(\alpha+1)}$, the expected error followed power law $\mathbb{E}_n \approx c_\alpha n^{-\beta}$, where $0 < \beta = \tfrac{\alpha}{1+\alpha} < 1$. The time-averaged error

$$
\mathbb{E}[\overline{\mathsf{E}}_N] \approx \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}[\mathsf{E}_n] \approx \frac{c_\alpha}{N}\int_0^N n^{-\beta}dn = c'_\alpha N^{-\beta} \quad \text{with} \quad c'_\alpha := \frac{c_\alpha}{1-\beta} = \alpha^{\frac{1}{1+\alpha}}\Gamma(\tfrac{\alpha}{1+\alpha})
$$

follows the same power law with the *same* exponent β, which is a generic property as foreshadowed earlier. As for the variance, we show in Appendix C that

$$
\begin{aligned}
\mathbb{E}[\overline{\mathsf{E}}_N^2] &\overset{\times}{\approx} \mathbb{E}[\overline{\mathsf{E}}_N]^2, \qquad \mathbb{V}[\overline{\mathsf{E}}_N] \overset{\times}{\approx} N^{-\frac{1+2\alpha}{1+\alpha}}, \quad \text{hence} \\
\sigma[\overline{\mathsf{E}}_N] &\overset{\times}{\approx} N^{-\frac{1/2+\alpha}{1+\alpha}} \quad \ll \quad N^{-\frac{\alpha}{1+\alpha}} \overset{\times}{\approx} \mathbb{E}[\overline{\mathsf{E}}_N]
\end{aligned}
$$

In particular, the signal-to-noise ratio is $\qquad \dfrac{\sigma[\overline{\mathsf{E}}_N]}{\mathbb{E}[\overline{\mathsf{E}}_N]} \overset{\times}{\approx} N^{-1/(2+2\alpha)}$

That is, the standard deviation is much smaller than then mean. A single run suffices to get a good (and excellent for $n \gtrsim 500$) signal-to-noise ratio for the averaged and cumulative error (see Figures 2&3&6 (right) and Figures 4&5 (top left)). Still, the infinite Zipf model leads to a more noisy learning curve than the finite uniform model. Intuitively, for every N, new $i_{N+1} \notin i_{1:N}$ have small but sufficient chance, contributing to the error and variance, decreasing exponentially in the uniform model, but only as a power law in the Zipf model.

General θ Case. One can show that the signal-to-noise ratio for the time-averaged error improves with N in general for any choice of θ. First note that the argument of $\sum_{i \neq j}[...]$ in (7) is non-negative, hence the variance is upper-bounded by the first $\sum_i$. Using $1 - (1 - \theta_i)^N \leq \theta_i N$, we get $\mathbb{V}[\overline{\mathsf{E}}_N] \leq \tfrac{1}{N}\mathbb{E}_N$, so the

signal-to-noise ratio is $\dfrac{\sigma[\overline{\mathsf{E}}_N]}{\mathbb{E}[\overline{\mathsf{E}}_N]} \leq \dfrac{\sqrt{\tfrac{1}{N}\mathbb{E}_N}}{\mathbb{E}[\overline{\mathsf{E}}_N]} = \dfrac{\sqrt{N\mathbb{E}_N}}{\sum_{n=0}^{N-1}\mathbb{E}_n} \to 0 \quad \text{for} \quad N \to \infty$

10

To prove the limit $\to 0$ we have to distinguish two cases: First note that $\mathbb{E}_n$ is monotone decreasing ($\mathbb{E}_n \searrow$). (i) For bounded total error $\sum_{n=0}^{\infty} \mathbb{E}_n \leq c$ (e.g. exponential error decay in finite models), $\mathbb{E}_n \searrow$ implies $\mathbb{E}_N = o(1/N)$, which implies that the numerator tends to 0; the denominator is lower-bound by $\mathbb{E}_0 = 1$. (ii) For unbounded total error $\sum_{n=0}^{N-1} \mathbb{E}_n \to \infty$ (most infinite models, e.g. Zipf and even exponential θ_i), we factor the denominator as $\sum_{n=0}^{N-1} \mathbb{E}_n \equiv \sqrt{\Sigma_{n=0}^{N-1} \mathbb{E}_n} \cdot \sqrt{\Sigma_{n=0}^{N-1} \mathbb{E}_n}$ and lower-bound one term by $\sum_{n=0}^{N-1} \mathbb{E}_n \geq N\mathbb{E}_N$, which is true since $\mathbb{E}_n \searrow$.
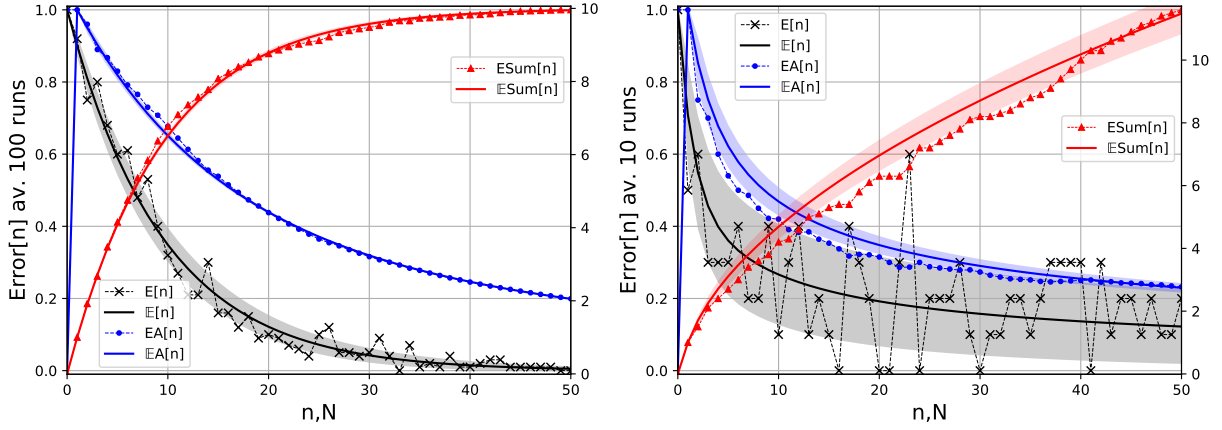
6 Experiments

Figure 1: (**Learning Curves**) *(left)* for uniform data distribution $\mathbb{P}[i_n = i] = \theta_i = \frac{1}{m}$ for $i \leq m = 10$ averaged over $k = 100$ runs. *(right)* for Zipf-distributed data $\mathbb{P}[i_n = i] = \theta_i \propto i^{-(\alpha+1)}$ for $\alpha = 1$ averaged over $k = 10$ runs. See Figures 4 and 5 for more plots for $k = 1, 10, 100, 1000$.
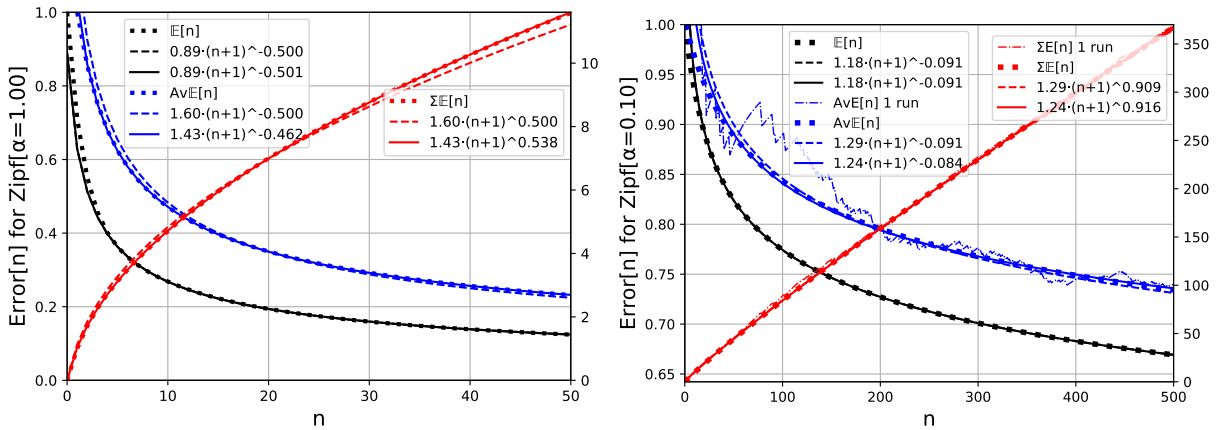

Figure 2: (**Power Law fit to Zipf-Distributed Data**) for Zipf-distributed data $\mathbb{P}[i_n = i] = \theta_i \propto i^{-(\alpha+1)}$ for $\alpha = 1$ *(left)* and $\alpha = 0.1$ *(right)*; averaged over infinitely many runs (dots), for fitted $\hat{\beta}$ (solid) and theoretical $\beta = \frac{\alpha}{1+\alpha}$ (dash), and empirical error for a single run ($k = 1$, dashdot).

We performed some control experiments to verify the correctness of the theory and claims, and the accuracy of the theoretical expressions.

Uniform and Zipf-distributed data. Figures 1 and 4 plot learning curves for uniformly distributed data $\mathbb{P}[i_n = i] = \theta_i = \frac{1}{m}$ for $i \leq m = 10$ averaged over $k = 1, 10, 100, 1000$
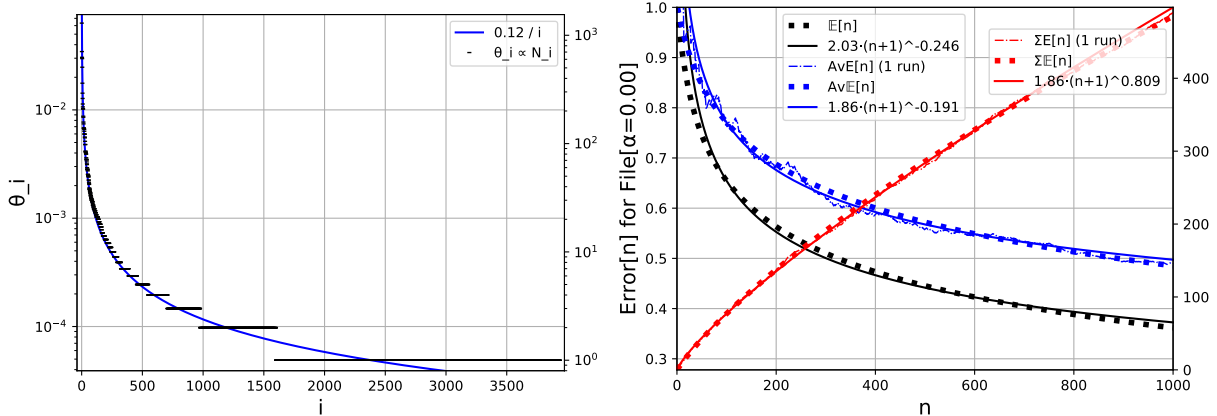
Figure 3: (**Word-Frequency in Text File, Learning Curve, Power Law**) *(left)* Log-linear plot of the relative (left scale) and absolute (right scale) frequency of words in the first 20469 words in file 'book1' of the Calgary Corpus, and fitted Zipf law. *(right)* Power law fit to learning curve for this data set for a word classification task.

runs. Figure 1 and 5 plot learning curves for Zipf-distributed data $\mathbb{P}[i_n = i] = \theta_i \propto i^{-(\alpha+1)}$ for $\alpha = 1$, also averaged over $k = 1, 10, 100, 1000$ runs. In both cases, 1000 synthetic data sets $\mathcal{D}_{50}$ have been generated, and the average is taken over k of them. Various errors are plotted as functions of the sample index/size n. The crosses are the instantaneous errors E_n averaged over k runs. The black curves are the exact expected instantaneous error $\mathbb{E}_n$. The shaded regions are 1 standard deviation $\sigma_n/\sqrt{k}$ from the theory (not empirical). Similarly the blue dots, lines, and shadings are the time-averaged errors $\overline{\mathsf{E}}_N$, their exact expectation $\mathbb{E}[\overline{\mathsf{E}}_N]$ and theoretical standard error $\sqrt{\mathbb{V}[\overline{\mathsf{E}}_N]/k}$. The red triangles, lines, and shadings are the empirical cumulative errors $\sum_{n=0}^{N} \mathsf{E}[n] \equiv N\overline{\mathsf{E}}_N$, their exact expectation $\sum_{n=0}^{N} \mathbb{E}[n] \equiv N\mathbb{E}[\overline{\mathsf{E}}_N]$, and theoretical standard error. The dashed lines connecting the empirical errors are for better visibility (only).

Fitting power laws to learning curves. We now fit power laws to the learning curves of (exactly) Zipf distributed data. Figure 2 shows fits for synthetic data with Zipf-exponents $\alpha = 1$ and $\alpha = 0.1$. The fit is "perfect" except for very small values of n. This is consistent with our approximation, which is good for $n\theta_1 \gg 1$. Theoretically we expect and empirically we found the approximation (4) to be good for $n\theta_1 \gtrsim 2$. For $\alpha = 1$ we have $\theta_1 = 0.5$, hence the approximation should be good for $n \gtrsim 4$, while for $\alpha = 0.1$ we have $\theta_1 \doteq 0.067$, hence the approximation should be good for $n \gtrsim 30$; both are consistent with the plots. To avoid clutter we only present expected curves. They perfectly match the averaged curves over infinitely many runs anyway (see Figures 1&5). The fitted power law exponents $\hat{\beta}$ are also close to the theoretical predictions $\beta = \frac{\alpha}{1+\alpha}$ ($\frac{1}{2} = 0.5$ for $\alpha = 1$ and $\frac{1}{11} \doteq 0.091$ for $\alpha = 0.1$).

Text data. It is well-known that the frequency of a word in typical texts is about inversely proportional to its rank in the frequency table: The most frequent word ('the') occurs about twice as often as the second most frequent word ('a'), about three times as often as the third most frequent word, and so on. That is, word frequency follows a Zipf distribution with $\alpha \approx 0$. Figures 3&6 (left) show the frequency distributions of the first 20469 words in file 'book1' of the Calgary Corpus. Apart from the steps, caused by word frequencies being integers, the distribution is very close to Zipf. Note that more than half of the words only appear once. Figures 3 (right) shows the learning curves for any word classification task. The power-law fit is reasonably good, but not perfect. The reason is

12

the step structure of especially rare words. Indeed, many θs are equal, and only finitely many are non-zero, so the learning curve is a finite superposition of exponentials as in (3). For moderate n this mixture amalgamates to an approximately power law. For large n, the error decays exponentially as $\exp(-\theta_{min}n)$. Indeed, for larger n, Figures 6 shows that the power fit becomes worse, and the true error decays faster than the fit power law. Note that for $\alpha \leq 0$, $i^{-(\alpha+1)}$ is not summable, hence *any* such distribution must break down after some i, our approximation becomes invalid, and $\beta \leq 0$ makes no sense in any case.

7   Extensions

In the following we discuss some potential extensions of the toy model. Some look feasible, others are hard or wishful thinking. We discuss the more realistic case of noisy labels, other loss functions, continuous features, and more realistic models that generalize, e.g. deep learning algorithms.

Noisy labels. In most machine learning applications, labels (or more general targets) are themselves noisy, not just the feature vector $x \in \mathcal{X}$, e.g. $y = h(x) + Noise$. The major implications are as follows:

(a) The learning algorithms need to be a bit smarter than just memorization, e.g. predicting the average or by majority.
(b) Due to the label noise, the error cannot converge to 0 anymore but to the intrinsic "entropy", which should be subtracted before studying scaling.
(c) For absolute (locally quadratic) loss there will be an extra $n^{-1/2}$ (n^{-1}) additive error term due to parameter estimation error, hence
(d) the instantaneous loss will not decay exponentially anymore even if the model is finite.
(e) Otherwise the scaling laws for Zipf data are *unchanged*.

In summary, the error/loss should be a sum of 3 terms, at least conceptually:

(1) The parameter learning rate $n^{-1/2}$ for absolute loss, squared, i.e. n^{-1} for (locally) quadratic loss,
(2) the same power law $n^{-\beta}$ as in the deterministic case,
(3) the inherent entropy in the data.

In Appendix E we verify claims (b,c,d,e) for our toy model extended to noisy binary classification for square and absolute loss. What is remarkable is that the *instantaneous* square loss $\mathsf{Loss}_n(A)$ for noisy labels turns out to include a term proportional to the *time-averaged* (0-1) error $\mathbb{E}[\overline{\mathsf{E}}_n] = (6)$ in the deterministic case. But this "magically" ensures (c,d,e), since $\mathbb{E}[\overline{\mathsf{E}}_n] \overset{\times}{\approx} \max\{\mathbb{E}_n, \frac{1}{n}\}$, at least for the choices of $\boldsymbol{\theta}$ discussed in Section 4. For instance, for a finite model, $\mathsf{Loss}_n(A) \overset{\times}{\approx} \mathbb{E}[\overline{\mathsf{E}}_n] \overset{\times}{\approx} \frac{1}{n}$.

Other loss functions. For our deterministic toy model, the loss function has little to no influence on the results as discussed in Appendix A. For noisy labels, this also seems to be the case, except that $n^{-\beta_0}$ is now the fastest possible decay, with β_0 *depending* on the loss-function: $\beta_0 = \frac{1}{2}$ for absolute loss and $\beta_0 = 1$ for locally quadratic loss such as KL and square. Loss functions with any (other) value of $\beta_0 > 0$ are possible but rare. This is another potential universality of scaling laws, their independence from the loss function for large models.

Continuous features. Countable feature spaces have some applications, e.g. in NLP, words can be identified with integer features $i \in \mathbb{N}$. In most applications, feature spaces are (effectively) continuous, often vector spaces $\mathbb{R}^d$, and no feature ever repeats exactly ($x_n \neq x_m$ for $n \neq m$). A simple model with a continuous domain is the Dirichlet Process, or the essentially equivalent Chinese Restaurant Process (CRP) and Stick-Breaking process. In the CRP, the continuous domain is essentially reduced to an exponentially distributed countable number of sticks=features, leading to power law learning curves $n^{-\beta}$, but the exponent is restricted to $\beta = 1$, which is too limiting. But even the CRP is not exactly a special case of our toy model and much harder to analyse. In some form of "mean-field" approximation it reduces to a special case of our model. The generalized 2-parameter Poison Dirichlet Process [BH10] also only leads to $\beta = 1$. Finding analytically tractable models with continuous features that exhibit interesting learning curves remains an open problem.

Generalizing algorithms. Proper models/algorithms for continuous features need to generalize from observed inputs to similar future not-yet-observed inputs, which is at the heart of virtually all interesting machine learning model/algorithms. Such models are much more varied and harder to analyze. If the domain could be partitioned into countably many cells, each cell containing only sufficiently similar features, and this can be done a-priori and is fixed independent of the actually realized data $\mathcal{D}_n$ and most importantly independent of the data size n, we arrive back at our countable toy model (usually with noisy labels) and our analysis (nearly) applies. But it is more plausible that a suitable partitioning, e.g. clustering of data, is in itself data (size) dependent, and hence will affect the scaling. The 'perfect prediction for exact repetition' in our toy model can be viewed as an abstraction of 'classify features in the same cell alike' which itself is a toy model for 'classify similar observations alike or similarly', so maybe some of our findings or analysis tools approximately transfer.

Deep learning. (Deep) neural networks are a particularly powerful class of models/algorithms that can generalize, but are also notoriously difficult to theoretically analyse. It may be a long way from our toy model to a similar analysis of NNs. Furthermore we have not at all considered the equally interesting questions of scaling with model size and compute.

8 Discussion

Summary. We introduced a very simple model that can exhibit power laws (decrease of error with data size) consistent with recent findings in deep learning. The model is plausibly the simplest such model, and that choice was deliberate to not get bogged down with intractable math or forced into crude approximations or bounds at this early stage of investigation. Many but not all data distributions lead to power laws. We do not know whether the discovered specific relation between the Zipf exponent α and the power law exponent $\beta = \alpha/(1 + \alpha)$ is an artifact of the model, or has wider validity beyond this model. The signal-to-noise ratio for the time-averaged error tend to zero, which implies that a single experimental run suffices for stable results. Together with arguments laid out by [Hut06] for log-loss, this suggests that model selection should be based on cumulative (AUC) error, rather than final test error.

Limitations. The toy model studied in this work is admittedly totally unrealistic as a Deep Learning model, but we believe it captures the (or at least a) true reason for

the observed scaling laws w.r.t. data. Whether it has any predictive power, or can be generalized to NNs and/or scaling laws for model size and/or compute, is beyond the scope of this paper. We hope that this initial investigation spurs more advanced theoretical investigations, and ultimately lead to predictive models. We have outlined some ideas in Section 7, some (more) are hopefully feasible. In any case, finding the simplest model which captures the essence is a necessary first step, and we believe our toy model fits this bill.

Applications. Besides providing scientific insight, a good theoretical understanding of scaling laws could ultimately help choosing network architecture (depth, with, and beyond) and (network and optimization algorithm) hyper-parameters in a more principled way [CJS$^+$93] than expensive hyper-parameter sweeps. It might also help with many other choices in fine-tuning, data augmentation, pre-training, etc. [HGLS20]. Being able to extrapolate the consequences of such choices from cheap training on a small subset of the data to the whole corpus by simply fitting power laws can save significant compute. The cost of training recent models has reached millions of dollars and can exhaust and exceed even FAANGs computational resources.

References

[BH10]   Wray Buntine and Marcus Hutter. A Bayesian review of the Poisson-Dirichlet process. Technical Report arXiv:1007.0296, NICTA and ANU, Australia, 2010.

[BHM18]   Mikhail Belkin, Daniel Hsu, and Partha Mitra. Overfitting or perfect fitting? Risk bounds for classification and regression rules that interpolate. *arXiv:1806.05161 [cond-mat, stat]*, June 2018.

[BHMM19]   Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, August 2019.

[Bub15]   Sébastien Bubeck. Convex Optimization: Algorithms and Complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[Cha81]   Anne Chao. On Estimating the Probability of Discovering a New Species. *Annals of Statistics*, 9(6):1339–1342, November 1981.

[Cho20]   Kyunghyun Cho. Scaling laws of recovering Bernoulli, November 2020. Blog Post http://kyunghyuncho.me/scaling-law-of-estimating-bernoulli/.

[CJS$^+$93] Corinna Cortes, LD Jackel, Sara A Solla, Vladimir Vapnik, and John S Denker. Learning curves: Asymptotic values and rate of convergence. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, pages 327–334, 1993.

[DHM89]   Ronald A. DeVore, Ralph Howard, and Charles Micchelli. Optimal nonlinear approximation. *manuscripta mathematica*, 63(4):469–478, December 1989.

[GPS18]   Andras Gyorgy, David Pal, and Csaba Szepesvari. *Online Learning: Algorithms for Big Data*. 2018.

[Haz16]   Elad Hazan. *Introduction to Online Convex Optimization*. 2016.

[HGLS20]   Derek Hoiem, Tanmay Gupta, Zhizhong Li, and Michal M. Shlapentokh-Rothman. Learning Curves for Analysis of Deep Networks. *arXiv:2010.11029 [cs, stat]*, October 2020.

[HKHM21]   Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling Laws for Transfer. *arXiv:2102.01293 [cs]*, February 2021.

[HKK+20]   Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling Laws for Autoregressive Generative Modeling. *arXiv:2010.14701 [cs]*, November 2020.

[HNA+17]   Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep Learning Scaling is Predictable, Empirically. *arXiv:1712.00409 [cs, stat]*, December 2017.

[Hut06]   M. Hutter. Human knowledge compression prize. open ended, http://prize.hutter1.net/, 2006.

[JN17]   Mark Johnson and Dat Quoc Nguyen. How much data is enough? Predicting how accuracy varies with training data size, September 2017. http://web.science.mq.edu.au/~mjohnson/papers/Johnson17Power-talk.pdf.

[KMH+20]   Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models. *arXiv:2001.08361 [cs, stat]*, January 2020.

[Mha96]   H. N. Mhaskar. Neural Networks for Optimal Approximation of Smooth and Analytic Functions. *Neural Computation*, 8(1):164–177, January 1996.

[Pin99]   Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8:143–195, January 1999.

[RRBS19]   Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A Constructive Prediction of the Generalization Error Across Scales. In *International Conference on Learning Representations*, September 2019.

[SB14]   Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014.

[SK20]   Utkarsh Sharma and Jared Kaplan. A Neural Scaling Law from the Dimension of the Data Manifold. *arXiv:2004.10802 [cs, stat]*, April 2020.

A Other Loss Functions

We can (slightly) generalize the learning algorithm A to other loss functions and behaviors on $i_{n+1} \notin i_{1:n}$. We continue to assume that A' suffers $\mathsf{Loss}_n = 0$ if $i \in i_{1:n}$ by using stored $\mathcal{D}_n$. Assume A' suffers Loss_n if $i_{n+1} \notin i_{1:n}$, then

$$\mathbb{E}[\mathsf{Loss}_n] \;=\; \mathbb{E}[\mathsf{Loss}_n|i_{n+1} \notin i_{1:n}] \cdot \mathbb{P}[i_{n+1} \notin i_{1:n}] + 0 \cdot \mathbb{P}[i \in i_{1:n}] \;=\; \mathbb{E}[\mathsf{Loss}_n|i_{n+1} \notin i_{1:n}] \cdot \mathbb{E}_n$$

The second factor $\mathbb{E}_n$ is our primary object of study. The first factor is often constant or bounded by constants: For instance for A in (1), $\mathsf{Loss}_n = 1$ if $i_{n+1} \notin i_{1:n}$, hence $\mathbb{E}[\mathsf{Loss}_n] = \mathbb{E}_n$. Assume a classification problem with K labels $\mathcal{Y} = \{0, ..., K-1\}$, and modify A to randomize its output (samples $y \in \mathcal{Y}$ uniformly) if $i_{n+1} \notin i_{1:n}$, then there is a $1/K$ chance to accidentally predict the correct label, hence $\mathbb{E}[\mathsf{Loss}_n|i_{n+1} \notin i_{1:n}] = 1 - 1/K$, where the expectation is now also w.r.t. the algorithm's randomness. For continuous $\mathcal{Y}$ and if A' samples y from any (non-atomic) density over $\mathcal{Y}$ if $i_{n+1} \notin i_{1:n}$, the probability of accidentally correctly predicting y_{n+1} is 0, hence $\mathbb{E}[\mathsf{Loss}_n] = \mathbb{E}_n$. For binary classification ($K = 2$) we could also let A' predict $y = \frac{1}{2}$ and use absolute loss $\mathsf{Loss}_n = |y - y_{n+1}|$, in which case $\mathbb{E}[\mathsf{Loss}_n|i_{n+1} \notin i_{1:n}] = \frac{1}{2}$. If A' instead samples y uniformly from $[0; 1]$, then $\mathbb{E}[\mathsf{Loss}_n|i_{n+1} \notin i_{1:n}] = \int_0^1 |y - y_{n+1}| dy = \frac{1}{4}$. If also $\mathcal{Y} = [0; 1]$, then $\mathbb{E}[\mathsf{Loss}_n|i_{n+1} \notin i_{1:n}] = \int_0^1 |y - y_{n+1}| dy = \frac{1}{4} + (\frac{1}{2} - y_{n+1})^2$, hence $\frac{1}{4}\mathbb{E}_n \leq \mathbb{E}[\mathsf{Loss}_n] \leq \frac{1}{2}\mathbb{E}_n$.

More generally, for any compact and uniformly rounded set $\mathcal{Y} \subseteq \mathbb{R}^d$, for any loss of the form $\mathsf{Loss}_n := \ell(||y - y_{n+1}||)$, for any norm $||\cdot||$, for any continuous strictly increasing $\ell \geq 0$, and A' sampling y from any density $p_{A'}(y) > 0$ over $\mathcal{Y}$ if $i_{n+1} \notin i_{1:n}$, we will show that for some constants $c_1, c_2 > 0$,

$$c_1 \mathbb{E}_n \;\leq\; \mathbb{E}[\mathsf{Loss}_n] \;\leq\; c_2 \mathbb{E}_n, \qquad\qquad \text{or} \quad \mathbb{E}[\mathsf{Loss}_n] \stackrel{\times}{=} \mathbb{E}_n \;\; \text{for short}$$

and this fact holds even more generally. Since a multiplicative constant in the loss is irrelevant from a scaling perspective, all scaling results for $\mathbb{E}_n$ also apply to this (slightly) more general setting.

The proof of this is as follows: A uniformly rounded set by definition can be represented as a union of ε-balls $B_\varepsilon(\tilde{y}) := \{y : ||y - \tilde{y}|| \leq \varepsilon\}$ for some fixed $\varepsilon > 0$, i.e. $\mathcal{Y} = \bigcup_{\tilde{y} \in \tilde{\mathcal{Y}}} B_\varepsilon(\tilde{y})$ for some $\tilde{\mathcal{Y}}$. Then

$$\begin{aligned}
\mathbb{E}[\mathsf{Loss}_n|i_{n+1} \notin i_{1:n}] \;&=\; \int_{\mathcal{Y}} \ell(||y - y_{n+1}||) p_{A'}(y) dy & (8)\\
&\stackrel{(a)}{\geq}\; \delta \int_{B_\varepsilon(\tilde{y})} \ell(||y - y_{n+1}||) dy \\
&\stackrel{(b)}{\geq}\; \delta \int_{B_\varepsilon(y_{n+1})} \ell(||y - y_{n+1}||) dy \\
&\stackrel{(c)}{=}\; \delta \int_{B_\varepsilon(0)} \ell(||z||) dz \;=:\; c_1 \stackrel{(d)}{>} 0
\end{aligned}$$

In (a), $\tilde{y} \in \tilde{\mathcal{Y}}$ is chosen such that $y_{n+1} \in B_\varepsilon(\tilde{y}) \subseteq \mathcal{Y}$, and $p_{A'} > \delta > 0$, since $p_{A'} > 0$ and $\mathcal{Y}$ is compact. In (b), $B_\varepsilon(y_{n+1})$ can be obtained from $B_\varepsilon(\tilde{y})$ by cutting out the moon $B_\varepsilon(\tilde{y}) \setminus B_\varepsilon(y_{n+1})$, and point-mirroring the moon at $\frac{1}{2}(\tilde{y} + y_{n+1})$. The flip brings every point closer to y_{n+1}, hence decreases the integral since ℓ is monotone increasing. (c) just recenters the integral, which now is obviously independent of y_{n+1}. It is non-zero (d), since ℓ is strictly increasing. Since ℓ is continuous and $\mathcal{Y}$ is compact, ℓ is upper bounded by $\ell_{max} < \infty$. This immediately implies (8) is upper bounded by $c_2 := \ell_{max} < \infty$.

B Derivation and Approximation of Expectation

Expectation. 2012(2a-d) Recall the error of (the basic form of) Algorithm A is $\mathsf{E}_n = [\![i_{n+1} \notin i_{1:n}]\!]$. Hence the probability that Algorithm A makes an error under distribution $\boldsymbol{\theta}$ given data $\mathcal{D}_n$ is

$$\mathbb{E}[\mathsf{E}_n|\mathcal{D}_n] \;=\; \mathbb{P}[A(i_{n+1},\mathcal{D}_n) \neq y_{n+1}|\mathcal{D}_n] \;=\; \sum_{i \notin i_{1:n}} \mathbb{P}[i_{n+1}=i] \;=\; \sum_{i \notin i_{1:n}} \theta_i \qquad (9)$$

The expectaion of this w.r.t. $\mathcal{D}_n$ is

$$\mathbb{E}_n \;:=\; \mathbb{E}[\mathsf{E}_n] \;=\; \mathbb{E}[\mathbb{E}[\mathsf{E}_n|\mathcal{D}_n]] \;=\; \sum_{i_{1:n}} \mathbb{P}[A(i_{n+1},\mathcal{D}_n) \neq y_{n+1}|\mathcal{D}_n]\mathbb{P}[\mathcal{D}_n]$$

$$= \; \sum_{i_{1:n}} \Big(\sum_{i \notin i_{1:n}} \theta_i \Big) \prod_{t=1}^{n} \theta_{i_t} \;=\; \sum_{i_{1:n}} \Big(\sum_{i=1}^{\infty} [\![i \neq i_1 \wedge ... \wedge i \neq i_n]\!]\theta_i \Big) \prod_{t=1}^{n} \theta_{i_t}$$

$$= \; \sum_{i=1}^{\infty} \theta_i \sum_{i_{1:n}} \prod_{t=1}^{n} [\![i \neq i_t]\!]\theta_{i_t} \;=\; \sum_{i=1}^{\infty} \theta_i \prod_{t=1}^{n} \sum_{i_t \neq i} \theta_{i_t} \;=\; \sum_{i=1}^{\infty} \theta_i (1-\theta_i)^n$$

The result can actually more easily be derived as

$$\mathbb{E}_n \;=\; \mathbb{P}[i_{n+1} \notin i_{1:n}] \;=\; \sum_{i=1}^{\infty} \mathbb{P}[i_{n+1}=i \wedge i_1 \neq i \wedge ... \wedge i_n \neq i]$$

$$= \; \sum_{i=1}^{\infty} \mathbb{P}[i_{n+1}=i] \prod_{t=1}^{n} \mathbb{P}[i_t \neq i] \;=\; \sum_{i=1}^{\infty} \theta_i (1-\theta_i)^n \qquad (10)$$

but the former derivation is more suitable for generalization to other loss functions and noisy labels.

Approximation. Let $f : (0;\infty) \to (0;\infty)$ be a continuously differentiable and decreasing extension of $\boldsymbol{\theta} : \mathbb{N} \to \mathbb{R}$, i.e. $f(i) := \theta_i$ and $f'(x) < 0$. Let $g(x) := f(x)e^{-nf(x)}$. Since $u \mapsto ue^{-nu}$ is unimodal with maximum $1/en$ and at $u = 1/n$ and f is monotone, $g(x)$ is unimodal with maximum $g_{max} = 1/en$ at $x_{max} = f^{-1}(\frac{1}{n})$. We hence can use (18) (any $a \in [0;1]$) to upper bound the sum in (10) by an integral as follows:

$$\mathbb{E}_n \;\overset{(10)}{=}\; \sum_{i=1}^{\infty} \theta_i (1-\theta_i)^n \;\leq\; \sum_{i=1}^{\infty} \theta_i e^{-n\theta_i} \;=\; \sum_{i=1}^{\infty} f(i)e^{-nf(i)}$$

$$\overset{(18)}{\leq}\; g_{max} + \int_{a}^{\infty} f(x)e^{-nf(x)}dx \;=\; \frac{1}{en} + \int_{0}^{f(a)} \frac{ue^{-nu}du}{|f'(f^{-1}(u))|}$$

where the last equality follows from a reparametrization $u = f(x)$ and $f(\infty) = 0$ and $dx = du/f'(x)$ and $f' < 0$.

For a lower bound, we need a lower bound on $(1-\theta_i)^n$. For $0 \leq x \leq 2\varepsilon$ we have

$$e^{-x} \;\leq\; 1 - x + \tfrac{1}{2}x^2 \;=\; 1 - (1-\tfrac{1}{2}x)x \;\leq\; 1 - (1-\varepsilon)x$$

Inserting $x = \theta_i/(1-\varepsilon)$, we get

$$1 - \theta_i \;\geq\; e^{-\theta_i/(1-\varepsilon)} \quad \text{for} \quad \theta_i \leq 2\varepsilon(1-\varepsilon)$$

18

Let i_0 be an index such that $\theta_{i_0} \leq 2\varepsilon(1-\varepsilon)$. We define $\tilde{n} := n/(1-\varepsilon)$. That $\tilde{n}$ is not an integer is no problem, and could even be avoided by renormalizing θ_i instead. Similarly as for the upper bound, we get a lower bound

$$\mathbb{E}_n \;\geq\; \sum_{i=i_0}^{\infty} \theta_i(1-\theta_i)^n \;\geq\; \sum_{i=i_0}^{\infty} \theta_i e^{-\tilde{n}\theta_i} \;=\; -\sum_{i=1}^{i_0-1} \theta_i e^{-\tilde{n}\theta_i} + \sum_{i=1}^{\infty} \theta_i e^{-\tilde{n}\theta_i}$$

$$\geq\; -\sum_{i=1}^{i_0-1} \theta_i e^{-\tilde{n}\theta_{i_0}} - \tilde{g}_{max} + \int_a^{\infty} f(x) e^{-\tilde{n}f(x)} dx$$

$$\geq\; -e^{-\tilde{n}\theta_{i_0}} - \frac{1}{e\tilde{n}} + \int_0^{f(a)} \frac{u e^{-\tilde{n}u} du}{|f'(f^{-1}(u))|}$$

For θ_i not decaying faster than exponential, i.e. if $\theta_i \geq \gamma\theta_{i-1}$ $\forall i$ for some $\gamma < 1$, there always exist i_0 such that $2\gamma\varepsilon(1-\varepsilon) \leq \theta_{i_0}/ \leq 2\varepsilon(1-\varepsilon)$. This finally leads to

$$\mathbb{E}_n \;\geq\; -e^{-2\gamma\varepsilon n} - \frac{1}{en} + \int_0^{f(a)} \frac{u e^{-\tilde{n}u} du}{|f'(f^{-1}(u))|}$$

Since we can choose ε arbitrarily small, combining both bounds, and choosing $a \to 0$ and f such that $f(x) \to \infty$ for $x \to 0$, we have

$$\left| \mathbb{E}_n - \mathbb{E}_n^{\int} \right| \;\leq\; \frac{1}{en} + o(1/n) \quad \text{with} \quad \mathbb{E}_n^{\int} \;:=\; \int_0^{\infty} \frac{u e^{-nu} du}{|f'(f^{-1}(u))|} \tag{11}$$

The integral is dominated by $u \lesssim \frac{1}{n}$, so for large n is determined by the asymptotics of $f'(f^{-1}(u))$ for $u \to 0$. Assume

$$|f'(f^{-1}(u))| \;\approx\; c' u^{\delta} \quad \text{for} \quad u \to 0 \quad \text{for some} \quad c' \text{ and } \delta \tag{12}$$

Substitution $u = v/n$ leads to

$$\mathbb{E}_n^{\int} \;\approx\; \int_0^{\infty} \frac{u e^{-nu}}{c' u^{\delta}} du \;=\; \frac{n^{\delta-2}}{c'} \int_0^{\infty} v^{1-\delta} e^{-v} dv \;=\; \frac{\Gamma(2-\delta)}{c'} n^{\delta-2} \tag{13}$$

where Γ is the Gamma function.

Zipf distribution. For Zipf-distributed $\theta_i = \alpha i^{-(\alpha+1)}$, let $u = f(x) = \alpha \cdot x^{-(\alpha+1)}$. This implies

$$x = f^{-1}(u) = (u/\alpha)^{-\frac{1}{1+\alpha}} \quad \text{and} \quad f'(x) = -\alpha(\alpha+1)x^{-(\alpha+2)} = -\alpha(\alpha+1)(u/\alpha)^{\frac{\alpha+2}{\alpha+1}}$$

hence approximation (12) is actually exact with $\delta = \frac{\alpha+2}{\alpha+1}$ and $c' = \alpha(\alpha+1)/\alpha^{(\alpha+2)/(\alpha+1)}$ leading to

$$\mathbb{E}_n^{\int} = c_\alpha n^{-\beta} \quad \text{with} \quad c_\alpha = \frac{\Gamma(2-\delta)}{c'} = \frac{\alpha^{1/(1+\alpha)}}{1+\alpha} \Gamma\left(\frac{\alpha}{1+\alpha}\right) \quad \text{and} \quad \beta = \delta - 2 = \frac{\alpha}{1+\alpha}$$

Note that $\tilde{n}^{-\beta} = n^{-\beta} + o(\frac{1}{n})$ for e.g. $\varepsilon = \frac{2}{n}\ln n$, and $e^{-\varepsilon n} = 1/n^2$. Numerically one can check that $c_\alpha \leq 1.214$ for all $\alpha > 0$ and $c_\alpha \geq 0.886$ for (the interesting) $\alpha \leq 1$, that is, c_α is nearly independent of α. $c_1 = \frac{1}{2}\sqrt{\pi} \doteq 0.886$ and $c_{0.1} = 1.177$ are in excellent agreement with the fit curves in Figure 2.

C   Time-Averaged Expectation and Variance

Time-averaged expectation. We now consider the time-averaged error

$$\overline{\mathsf{E}}_N \; := \; \frac{1}{N} \sum_{n=0}^{N-1} \mathsf{E}_n$$

We derive the expressions for its expectation $\mathbb{E}[\overline{\mathsf{E}}_N]$ and variance $\mathbb{V}[\overline{\mathsf{E}}_N]$ stated in Section 5. The expectation is trivial:

$$\mathbb{E}[\overline{\mathsf{E}}_N] \; = \; \frac{1}{N} \sum_{n=0}^{N-1} \mathbb{E}[\mathsf{E}_n] \; = \; \frac{1}{N} \sum_{i=1}^{\infty} \theta_i \sum_{n=0}^{N-1} (1-\theta_i)^n \; = \; \frac{1}{N} \sum_{i=1}^{\infty} [1-(1-\theta_i)^N]$$

Time-averaged variance. For the variance, we first compute $\overline{\mathsf{E}}_N^2$, then $\mathbb{E}[\overline{\mathsf{E}}_N^2]$, then $\mathbb{V}[\overline{\mathsf{E}}_N] = \mathbb{E}[\overline{\mathsf{E}}_N^2] - \mathbb{E}[\overline{\mathsf{E}}_N]^2$:

$$\overline{\mathsf{E}}_N^2 \; = \; \frac{1}{N^2} \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} \mathsf{E}_n\mathsf{E}_m \; \overset{(b)}{=} \; \frac{2}{N^2} \sum_{n=0}^{N-1}\sum_{m=0}^{n-1} \mathsf{E}_n\mathsf{E}_m + \frac{1}{N^2} \sum_{n=0}^{N-1} \mathsf{E}_n^2$$

where (b) breaks up the double sum into lower=upper and diagonal terms. Since $m < n$ we have

$$
\begin{aligned}
\mathsf{E}_n \cdot \mathsf{E}_m \; &= \; [\![i_{n+1} \notin i_{1:n}]\!] \cdot [\![i_{m+1} \notin i_{1:m}]\!] \\
&= \; \sum_{i=1}^{\infty} [\![i_{n+1} = i \wedge i_n \neq i \wedge ... \wedge i_1 \neq i]\!] \cdot \sum_{j=1}^{\infty} [\![i_{m+1} = j \wedge i_m \neq j \wedge ... \wedge i_1 \neq j]\!] \\
&= \; \sum_{i,j} [\![i_{n+1} = i \wedge i_n \neq i \wedge ... \wedge i_{m+2} \neq i \wedge i_{m+1}\,{\overset{=\,j}{\neq\,i}} \wedge i_m\,{\overset{\neq\,j}{\neq\,i}} \wedge ... \wedge i_1\,{\overset{\neq\,j}{\neq\,i}}]\!]
\end{aligned}
$$

For $i = j$, $i_{m+1}\,{\overset{=j}{\neq i}}$ (meaning $i_{m+1} = j \wedge i_{m+1} \neq i$) is a contradiction, so we can limit the sum to $i \neq j$. Taking the expectation and noting that $\mathsf{E}_n \in \{0, 1\}$ and all i_t are independent and $\mathbb{P}[i_t = i] = \theta_i$ we get

$$
\begin{aligned}
\mathbb{E}[\mathsf{E}_n \cdot \mathsf{E}_m] \; &= \; \sum_{i \neq j} \mathbb{P}[i_{n+1} = i]\mathbb{P}[i_n \neq i]...\mathbb{P}[i_{m+2} \neq i]\mathbb{P}[i_{m+1}\,{\overset{=\,j}{\neq\,i}}]\mathbb{P}[i_m\,{\overset{\neq\,j}{\neq\,i}}]...\mathbb{P}[i_1\,{\overset{\neq\,j}{\neq\,i}}] \\
&= \; \sum_{i \neq j} \theta_i (1-\theta_i)^{n-m-1} \theta_j (1-\theta_i-\theta_j)^m, \quad \text{hence}
\end{aligned}
$$

$$\mathbb{E}[\sum_{n=0}^{N-1}\sum_{m=0}^{n-1} \mathsf{E}_n\mathsf{E}_m] \; = \; \sum_{i \neq j} b_{ijN} = \tfrac{1}{2} \sum_{i \neq j} b_{ijN} + b_{jiN}, \quad \text{where}$$

20

$$b_{ijN} := \sum_{n=0}^{N-1} \sum_{m=0}^{n-1} \theta_i (1-\theta_i)^{n-m-1} \theta_j (1-\theta_i-\theta_j)^m$$

$$= \theta_i \theta_j \sum_{m=0}^{N-1} (1-\theta_i-\theta_j)^m \sum_{n=m+1}^{N-1} (1-\theta_i)^{n-m-1}$$

$$= \theta_i \theta_j \sum_{m=0}^{N-1} (1-\theta_i-\theta_j)^m \frac{1}{\theta_i} [1-(1-\theta_i)^{N-m-1}]$$

$$= \theta_j \sum_{m=0}^{N-1} (1-\theta_i-\theta_j)^m - \theta_j (1-\theta_i)^{N-1} \sum_{m=0}^{N-1} \left(\frac{1-\theta_i-\theta_j}{1-\theta_i}\right)^m$$

$$= \theta_j \frac{1}{\theta_i+\theta_j} [1-(1-\theta_i-\theta_j)^N] - \theta_j (1-\theta_i)^{N-1} \frac{1-\theta_i}{\theta_j} \left[1 - \left(\frac{1-\theta_i-\theta_j}{1-\theta_i}\right)^N\right]$$

$$= \frac{\theta_j}{\theta_i+\theta_j} [1-(1-\theta_i-\theta_j)^N] - (1-\theta_i)^N + (1-\theta_i-\theta_j)^N, \quad \text{hence}$$

$$b_{ijN} + b_{jiN} = 1 - (1-\theta_i)^N - (1-\theta_j)^N + (1-\theta_i-\theta_j)^N$$

The diagonal term is easy:

$$\frac{1}{N^2} \sum_{n=0}^{N-1} \mathbb{E}[\mathsf{E}_n^2] = \frac{1}{N^2} \sum_{n=0}^{N-1} \mathbb{E}[\mathsf{E}_n] = \frac{1}{N} \sum_{n=0}^{N-1} \mathbb{E}[\bar{\mathsf{E}}_N] = \frac{1}{N^2} \sum_{i=1}^{\infty} 1 - (1-\theta_i)^N$$

Putting everything together, non-diagonal $i \neq j$ and diagonal $i = j$ expressions we get our final expression

$$\mathbb{E}[\bar{\mathsf{E}}_N^2] = \frac{1}{N^2} \sum_{i \neq j} [1 - (1-\theta_i)^N - (1-\theta_j)^N + (1-\theta_i-\theta_j)^N] + \frac{1}{N^2} \sum_{i=1}^{\infty} [1 - (1-\theta_i)^N]$$

In order to get the variance of $\bar{\mathsf{E}}_N$ we have to subtract the squared expected error

$$\mathbb{E}[\bar{\mathsf{E}}_N]^2 = \left(\frac{1}{N} \sum_{i=1}^{\infty} 1 - (1-\theta_i)^N\right)^2 = \frac{1}{N^2} \sum_{i,j} [1 - (1-\theta_i)^N] \cdot [1 - (1-\theta_j)^N]$$

$$= \frac{1}{N^2} \sum_{i \neq j} [1 - (1-\theta_i)^N - (1-\theta_j)^N + (1-\theta_i)^N (1-\theta_j)^N]$$

$$+ \frac{1}{N^2} \sum_{i=1}^{\infty} [1 - 2(1-\theta_i)^N + (1-\theta_i)^{2N}]$$

where we expanded the product and separated the $i \neq j$ from the $i = j$ terms, which now easily leads to

$$\mathbb{V}[\bar{\mathsf{E}}_N] = \mathbb{E}[\bar{\mathsf{E}}_N^2] - \mathbb{E}[\bar{\mathsf{E}}_N]^2$$

$$= \frac{1}{N^2} \sum_{i \neq j} [(1-\theta_i-\theta_j)^N - (1-\theta_i)^N (1-\theta_j)^N] + \frac{1}{N^2} \sum_{i=1}^{\infty} [(1-\theta_i)^N - (1-\theta_i)^{2N}]$$

Approximation. We can approximate the variance similarly to the expectation $\mathbb{E}_n$. We

21

only provide a heuristic derivation analogous to (4):

$$\mathbb{V}[\overline{\mathsf{E}}_N] \overset{(a)}{\approx} \frac{1}{N^2}\sum_{i\neq j}[e^{-(\theta_i-\theta_j)N} - e^{-\theta_i N}e^{-\theta_j N}] \ + \ \frac{1}{N^2}\sum_{i=1}^{\infty}[e^{-\theta_i N} - e^{-2\theta_i N}]$$

$$\overset{(b)}{\approx} 0 \ + \ \frac{1}{N^2}\int_1^{\infty} e^{-f(x)N} - e^{-2f(x)N}dx \overset{(c)}{=} \frac{1}{N^2}\int_0^{\theta_1}\frac{e^{-uN}-e^{-2uN}}{|f'(f^{-1}(u))|}du$$

$$\overset{(d)\times}{\approx} \frac{1}{N^2|f'(f^{-1}(\frac{\ln 2}{N}))|}\int_0^{\theta_1} e^{-uN} - e^{-2uN}du \overset{(e)}{\approx} \frac{1}{2N^3|f'(f^{-1}(\frac{\ln 2}{N}))|}$$

$$\overset{\times}{\approx} \frac{1}{2N}\mathbb{E}_N \begin{cases} \overset{\times}{\approx} \frac{1}{N}\mathbb{E}[\overline{\mathsf{E}}_N] & \text{if } \mathbb{E}_N\overset{\times}{\approx}N^{-\beta} \\ \leq \frac{1}{2N}\mathbb{E}[\overline{\mathsf{E}}_N] & \text{always} \end{cases}$$

(a) follows from $1-\theta_i \approx e^{-\theta_i}$, (b) by setting $f(i) := \theta_i$, and replacing the sums by integrals, (c) follows from a reparametrization $u = f(x)$ and $f(1) = \theta_1$ and $f(\infty) = 0$ and $dx = du/f'(x)$ and $f' < 0$. The numerator $e^{-uN}-e^{-2uN}$ is maximal and (strongly) concentrated around $u = \ln 2/n$, hence $u \approx \ln 2/n$ gives most of the integral's contribution. Therefore in (d) replacing u by $\ln 2/n$ in the denominator can be a reasonable approximation. (e) follows from $\int_0^{\theta_1} ... \approx \int_0^{\infty} ... = 1/2N$ for $\theta_1 N \gg 1$. We could use this approximation for various concrete f, but if we substitute $u = 1/N$ instead of $\ln 2/N$ in (d) we only make a multiplicative error $(\times)$, and the expression nicely reduces to the *instantaneous* expected error $\mathbb{E}_N$. For slowly decreasing error $\mathbb{E}_N\overset{\times}{\approx}N^{-\beta}$, we have $\mathbb{E}[\mathsf{E}_N]\overset{\times}{\approx}\mathbb{E}[\overline{\mathsf{E}}_N]$. In general $\mathbb{E}[\mathsf{E}_N] \leq \mathbb{E}[\overline{\mathsf{E}}_N]$, since $\mathbb{E}_n$ is monotone decreasing.

D   Upper Bound for SubExponential Distribution

Here we show that $\mathbb{E}_n^{\int}\overset{\times}{\leq}\frac{1}{n}$ for $\theta_i\overset{\times}{\leq}e^{-\gamma i}$, hence also $\mathbb{E}_n\overset{\times}{\leq}\frac{1}{n}$ by (11). Interestingly, the standard approximation technique of splitting integrals over f (naturally at $u = f(x) = 1/n$ or otherwise) and bounding each, only leads to a cruder $O(\frac{1}{n}\ln n)$. To get $\frac{1}{n}$, we will maximizing $\mathbb{E}_n^{\int}[f] = \int f(x)e^{-nf(x)}dx$ w.r.t. f under constraints $\int f(x) = 1$ and $f(x) \leq \overline{f}(x) := be^{-\gamma x}$. This can be done by Lagrange multipliers, or more formally using the Karush–Kuhn–Tucker saddle-point theorem: A necessary condition is that variational derivative of

$$L_n[f] \ := \ \int_0^{\infty}f(x)e^{-nf(x)}dx + \lambda_n\left[\int_0^{\infty}f(x)dx - 1\right] - \int_0^{\infty}\mu_n(x)\max\{0, f(x)-\overline{f}(x)\}dx$$

w.r.t. f is 0 for some $\mu_n(x)$.

$$\frac{\delta L_n[f]}{\delta f(x)} \ = \ [1 - nf(x)]e^{-nf(x)} \ + \ \lambda_n \ - \ \mu_n(x)[\![f(x) \geq \overline{f}(x)]\!] \ \overset{!}{=} \ 0$$

For all x for which $f(x) < \overline{f}(x)$, this implies that f must be the same, i.e. $f(x) = c$. For all x for which $f(x) \geq \overline{f}(x)$, the constraints imply $f(x) = \overline{f}(x)$. Jointly this implies that $f(x) = \min\{c, \overline{f}(x)\} =: f^*(x)$, and that this is an extremal point of $\mathbb{E}_n^{\int}[f]$. Since ue^{-nu} has a unique extremal point, and this is a maximum, this implies that f^* maximizes $\mathbb{E}_n^{\int}[f]$. Let $x_c := \frac{1}{\gamma}\ln\frac{b}{c}$ be the transition point where $c = \overline{f}(x_c) = be^{-\gamma x_c}$. Now we can split the integral for f^* get the desired result. The reason this now works is because the

22

splitting point x_c is independent of n, while for f it had to depend on n:

$$\mathbb{E}_n^f[f] \leq \mathbb{E}_n^f[f^*] = \int_0^{x_c} ce^{-nc}dx + \int_{x_c}^{\infty} \overline{f}(x)e^{-n\overline{f}(x)}dx$$

$$\leq x_c ce^{-nc} + \int_0^{\infty} \overline{f}(x)e^{-n\overline{f}(x)}dx$$

$$\stackrel{(a)}{=} x_c ce^{-nc} + \frac{1}{\gamma n} \stackrel{(b)}{\leq} \frac{x_c}{n} + \frac{1}{\gamma n} \stackrel{(c)}{=} \frac{1}{\gamma n}[1 + \ln \tfrac{b}{c}] = O(\tfrac{1}{n})$$

(a) follows from change of variables $x \rightsquigarrow u := \overline{f}(x)$ and computing the integral as done before in the exponential case. (b) follows from $ce^{-nc} \leq \frac{1}{n} \, \forall c$. (c) follows from inserting x_c.

The constant c is actually determined by

$$\int_0^{\infty} \overline{f}(x)dx = \int_0^{x_c} cdx + \int_{x_c}^{\infty} be^{-\gamma x}dx = x_c c + \frac{b}{\gamma}e^{-\gamma x_c} = \frac{c}{\gamma}\ln\frac{b}{c} + \frac{c}{\gamma} \stackrel{!}{=} 1$$

One can show that this requires $c \approx \gamma/\ln b$ for large b. For a (cruder) bound: The integral increases with c for the relevant range $c < b$, and inserting $c = \gamma/b$ reveals that $\int < 1$ for $b > [1 - \ln\gamma]^{-1}$ and $\gamma < e$, hence solution $c^* \geq \gamma/b$, leading to $\mathbb{E}_n^f[f] \leq \frac{1}{\gamma n}[1 + \ln(b^2/\gamma)]$ with a rather mild dependence on b.

E   Noisy Labels

Here we generalize our model to noisy labels. We first derive generic expression expressions for (somewhat) general algorithm and loss. We then instantiate them for frequency estimation and square loss. Finally we outline how to derive similar expressions for the absolute loss.

General loss.   Consider a binary classification problem where labels $y_t \in \{0, 1\}$ are noisy. Let $\gamma_i := \mathbb{P}[y_t = 1 | i_t = i]$ be the probability that feature $i \in \mathbb{N}$ is labelled as 1. The probability of observing feature i itself remains $\theta_i := \mathbb{P}[i_t = i]$ as in the deterministic case. Algorithm $A_i := A(i, \mathcal{D}_n) \in [0; 1]$ now aims to predict γ_i. The square loss if predicting A_i while the true label is y, and its expectation w.r.t. γ_i, are

$$\mathsf{Loss}_n(A|\mathcal{D}_n, i_{n+1} = i, y_{n+1} = y) = (y - A_i)^2$$
$$\mathsf{Loss}_n(A|\mathcal{D}_n, i_{n+1} = i) = \gamma_i(1 - A_i)^2 + (1 - \gamma_i)(0 - A_i)^2 = (\gamma_i - A_i)^2 + \gamma_i(1 - \gamma_i)$$

The most naive learning algorithm would predict γ_i from observed frequencies: $A_i = k_i/n_i$ if feature i occurred $n_i := \#\{t \leq n : i_t = i\}$ times and has label 1 for $k_i := \#\{t \leq n : i_t = i, y_t = 1\}$ times. Obviously $k_i/n_i \to \gamma_i$ provided $n_i \to \infty$, hence $\mathsf{Loss}_n(A|\mathcal{D}_n, i_{n+1} = i)$ converges to the intrinsic label "entropy" $\gamma_i(1 - \gamma_i)$, rather than to 0, which has to be subtracted for a power law analysis to make sense. Similarly the expectation of log-loss $-y\ln A_i - (1-y)\ln(1 - A_i)$ w.r.t. γ_i leads to Kullback-Leibler loss $\mathrm{KL}(\gamma_i||A_i) + \mathrm{Entropy}$ $H(\gamma_i)$. More generally let us assume $A(i, \mathcal{D}_n)$ depends (somehow) only on k_i and n_i (e.g. Laplace rule), and hence

$$\mathsf{Loss}_n(A|\mathcal{D}_n, i_{n+1} = i) = \ell(\gamma_i, A_i) = \ell(\gamma_i, k_i, n_i) = \mathsf{Loss}_n(A|k_i, n_i, i_{n+1} = i)$$

for some function ℓ. We now take the expectation over $\mathcal{D}_n$:

$$
\begin{aligned}
L_i &:= \mathsf{Loss}_n(A|i_{n+1} = i) \\
&= \sum_{\mathcal{D}_n} \mathsf{Loss}_n(A|\mathcal{D}_n, i_{n+1} = i)\mathbb{P}[\mathcal{D}_n] \\
&= \sum_{n_i=0}^{n} \sum_{k_i=0}^{n_i} \left(\sum_{\mathcal{D}_n:k_i,n_i} \mathbb{P}[\mathcal{D}_n] \right) \ell(\gamma_i, k_i, n_i)
\end{aligned}
$$

where $\sum_{\mathcal{D}_n:k_i,n_i}$ means that the sum is restricted to $\mathcal{D}_n$ for which i $((i,1))$ appears n_i (k_i) times. The probability of each of this happening is binomial:

$$
\sum_{\mathcal{D}_n:k_i,n_i} \mathbb{P}[\mathcal{D}_n] = \sum_{i_{1:n}:n_i} \mathbb{P}[i_{1:n}] \sum_{y_{1:n}:k_i} \mathbb{P}[y_{1:n}|i_{1:n}] = \binom{n}{n_i}\theta_i^{n_i}(1-\theta_i)^{n-n_i}\binom{n_i}{k_i}\gamma_i^{k_i}(1-\gamma_i)^{n_i-k_i}
$$

This is obvious or follows by explicit calculation of the sums and some algebra. Putting everything together and finally taking the expectation over i we get

$$
\mathsf{Loss}_n(A|n_i, i_{n+1} = i) = \sum_{k_i=0}^{n_i} \binom{n_i}{k_i}\gamma_i^{k_i}(1-\gamma_i)^{n_i-k_i}\ell(\gamma_i, k_i, n_i) \tag{14}
$$

$$
L_i \equiv \mathsf{Loss}_n(A|i_{n+1} = i) = \sum_{n_i=0}^{n} \binom{n}{n_i}\theta_i^{n_i}(1-\theta_i)^{n-n_i}\,\mathsf{Loss}_n(A|n_i, i_{n+1} = i) \tag{15}
$$

$$
\mathsf{Loss}_n(A) = \sum_{i=1}^{\infty} \theta_i L_i \tag{16}
$$

In the deterministic case $\gamma_i \in \{0,1\}$ and our memorizing algorithm (1) with 0-1 loss, $\ell(\gamma_i, k_i, n_i) = [\![n_i = 0]\!]$ is independent k_i, so the k_i sums the binomial to 1, and the n_i-sum collapses to $n_i = 0$, leading back to $\mathsf{Loss}_n(A) = \sum_{i=1}^{\infty} \theta_i(1-\theta_i)^n = \mathbb{E}_n =$ (10).

Square loss. For noisy labels, $\gamma_i \in [0;1]$, frequency estimator $A_i = k_i/n_i$, square loss $\ell(\gamma_i, k_i, n_i) = (\gamma_i - k_i/n_i)^2$ for $n_i > 0$ (with "Entropy" $\gamma_i(1-\gamma_i)$ removed), and keeping $\ell(\gamma_i, k_i, 0) = 1$, we proceed as follows: The k_i-sum in (14) becomes the variance of k_i/n_i, hence $\mathsf{Loss}_n(A|n_i, i_{n+1} = i) = \gamma_i(1-\gamma_i)/n_i$. Unfortunately plugging this into the next n_i-sum in (15) leads to a hypergeometric function. We tried various approximations, all leading essentially to the same end result. The simplest approximation is to approximate $\gamma_i(1-\gamma_i)/n_i$ by $\gamma_i(1-\gamma_i)/(n_i + 1)$, which is asymptotically correct and within a factor

of 2 also valid for $1 \leq n_i < \infty$, and avoids hypergeometric functions altogether:

$$
\begin{aligned}
L_i \;&=\; \sum_{n_i=0}^{n} \binom{n}{n_i} \theta_i^{n_i}(1-\theta_i)^{n-n_i} \, \mathsf{Loss}_n(A|n_i, i_{n+1}=i) \\[1mm]
&=\; (1-\theta_i)^n \;+\; \sum_{n_i=1}^{n} \binom{n}{n_i}\theta_i^{n_i}(1-\theta_i)^{n-n_i}\frac{\gamma_i(1-\gamma_i)}{n_i} \\[1mm]
&\approx\; (1-\theta_i)^n \;+\; \sum_{n_i=1}^{n} \binom{n}{n_i}\theta_i^{n_i}(1-\theta_i)^{n-n_i}\frac{\gamma_i(1-\gamma_i)}{n_i+1} \\[1mm]
&=\; (1-\theta_i)^n \;-\; (1-\theta_i)^n\gamma_i(1-\gamma_i) \;+\; \sum_{n_i=0}^{n} \binom{n}{n_i}\theta_i^{n_i}(1-\theta_i)^{n-n_i}\frac{\gamma_i(1-\gamma_i)}{n_i+1} \\[1mm]
&\stackrel{(a)}{=}\; [1-\gamma_i(1-\gamma_i)](1-\theta_i)^n \;+\; \frac{\gamma_i(1-\gamma_i)}{(n+1)\theta_i}\sum_{k=1}^{n+1}\binom{n+1}{k}\theta_i^k(1-\theta_i)^{n+1-k} \\[1mm]
&\stackrel{(b)}{=}\; [1-\gamma_i(1-\gamma_i)](1-\theta_i)^n \;+\; \frac{\gamma_i(1-\gamma_i)}{(n+1)\theta_i}[1-(1-\theta_i)^{n+1}]
\end{aligned}
$$

where (a) follows from substituting $n_i = k-1$ and rearranging terms, and (b) from adding and subtracting the missing $k=0$ contribution, and the fact that a complete binomial sums to 1. If we assume that the noise level is the same for all features, i.e. $\gamma_i = \gamma$ or $\gamma_i = 1-\gamma$, then

$$
\begin{aligned}
\mathsf{Loss}_n(A) \;\equiv\; \sum_{i=1}^{\infty}\theta_i L_i \;&\approx\; [1-\gamma(1-\gamma)]\sum_{i=1}^{\infty}\theta_i(1-\theta_i)^n \;+\; \frac{\gamma(1-\gamma)}{n+1}\sum_{i=1}^{\infty}[1-(1-\theta_i)^{n+1}] \\[1mm]
&=\; [1-\gamma(1-\gamma)]\mathbb{E}_n \;+\; \gamma(1-\gamma)\mathbb{E}[\overline{\mathsf{E}}_{n+1}]
\end{aligned}
$$

Again, in the deterministic case $\gamma \in \{0,1\}$, we get back $\mathsf{Loss}_n(A) = \mathbb{E}_n$.

Upper and lower bounds for general γ_i. Let us introduce the shortcuts

$$
E_i^n \;:=\; (1-\theta_i)^n \;\leq\; \frac{1-(1-\theta_i)^{n+1}}{(n+1)\theta_i} \;=:\; \overline{E}_i^{n+1}
$$

The inequality is reminiscent of $\mathbb{E}_n \equiv \mathbb{E}[\mathsf{E}_n] \leq \mathbb{E}[\overline{\mathsf{E}}_{n+1}]$, which followed from $\mathbb{E}_n$ being monotone deceasing, or can directly be proven as

$$
\frac{1}{\theta_i}[1-(1-\theta_i)^{N+1}] \;=\; \sum_{n=0}^{N}(1-\theta_i)^n \;\geq\; \sum_{n=0}^{N}(1-\theta_i)^N \;=\; (N+1)(1-\theta_i)^N
$$

For a lower bound on L_i, with $c_i := \gamma_i(1-\gamma_i)$ and $\underline{c} := \inf_i\{c_i\}$, we get

$$
\begin{aligned}
L_i \;&\geq\; (1-c_i)E_i^n + c_i\overline{E}_i^{n+1} \\[1mm]
&=\; (1-c_i)E_i^n + (c_i-\underline{c})\overline{E}_i^{n+1} + \underline{c}\,\overline{E}_i^{n+1} \\[1mm]
&\geq\; (1-c_i)E_i^n + (c_i-\underline{c})E_i^n + \underline{c}\,\overline{E}_i^{n+1} \\[1mm]
&\geq\; (1-\underline{c})E_i^n + \underline{c}\,\overline{E}_i^{n+1}
\end{aligned}
$$

Taking the $\boldsymbol{\theta}$-expectation we get

$$
\mathsf{Loss}_n(A) \;\equiv\; \sum_{i=1}^{\infty}\theta_i L_i \;\geq\; (1-\underline{c})\mathbb{E}_n + \underline{c}\,\mathbb{E}[\overline{\mathsf{E}}_{n+1}]
$$

Note that if γ_i includes deterministic labels $\gamma_i \in \{0, 1\}$ (or arbitrarily close to deterministic), then $\underline{c} = 0$, and the best we can conclude is $\mathsf{Loss}_n(A) \geq \mathbb{E}_n$, which holds for trivial reasons: A new feature causes one unit of error in the deterministic as well as the noisy label case, but there can be further errors for noisy labels.

Similarly for an upper bound with $\overline{c} := \sup_i\{c_i\}$, we get

$$
\begin{aligned}
L_i &\leq (1 - c_i)E_i^n + 2c_i\overline{E}_i^{n+1} \\
&= (1 - \overline{c})E_i^n + (\overline{c} - c_i)E_i^n + 2c_i\overline{E}_i^{n+1} \\
&\leq (1 - \overline{c})E_i^n + 2(\overline{c} - c_i)\overline{E}_i^{n+1} + 2c_i\overline{E}_i^{n+1} \\
&= (1 - \overline{c})E_i^n + 2\overline{c}\overline{E}_i^{n+1}
\end{aligned}
$$

Taking the $\boldsymbol{\theta}$-expectation we get

$$
\mathsf{Loss}_n(A) \leq (1 - \overline{c})\mathbb{E}_n + 2\overline{c}\mathbb{E}[\overline{E}_{n+1}] \leq \tfrac{3}{4}\mathbb{E}_n + \tfrac{1}{2}\mathbb{E}[\overline{E}_{n+1}]
$$

where the last inequality holds for any $\gamma_i \in [0; 1]$ ($\overline{c} = \frac{1}{4}$). In the deterministic case, $\overline{c} = 0$, and $\mathsf{Loss}_n(A) \leq \mathbb{E}_n$ as it should be.

Noise bounded away from 0/1. If we assume that γ_i are bounded away from 0 and 1 ($\underline{c} > 0$), the above bounds imply that within a multiplicative constant

$$
\mathsf{Loss}_n(A) \overset{\times}{=} \mathbb{E}_n + \mathbb{E}[\overline{E}_n]
$$

This is quite remarkable, that the *instantaneous* square loss for noisy labels includes a term proportional to the *time-averaged* (0-1) error in the deterministic case. As we have seen in the main paper, roughly, as long as $\mathbb{E}_n$ goes to 0 slower than $1/n$, $\mathbb{E}_n$ and $\mathbb{E}[\overline{E}_n]$ have the same asymptotics, which in turn implies that the results for deterministic classification transfer to noisy labels, in particular α-Zipf-distributed data lead to β-power law learning curves with $\beta = \frac{\alpha}{1+\alpha}$. While $\mathbb{E}$ can decay faster than $1/n$, e.g. exponentially for finite models, $\mathbb{E}[\overline{E}_n]$ and hence $\mathsf{Loss}_n(A)$ can never decay faster than $1/n$. As discussed in the introduction, the reason is that the accuracy to which parameters (here γ_i) can be estimated from n_i i.i.d. data is $\overset{\times}{=} n_i^{-1/2} \overset{\times}{\geq} n^{-1/2}$, which squares to $\mathsf{Loss}_n(A) \overset{\times}{\geq} n^{-1}$ for (locally) quadratic loss.

Absolute loss. For absolute loss $\ell = |\gamma_i - k_i/n_i|$ there is no closed-form solution for (14). For large n_i, the binomial is approximately Gaussian (in k_i/n_i) with mean γ_i and variance $\gamma_i(1 - \gamma_i)/n_i$, and (14) can be evaluated to $\sqrt{\gamma_i(1 - \gamma_i)/n_i}$. Plugging this into (15) we can approximate the n_i-sum for $n\theta_i \ll 1$ and for $n\theta_i \gg 1$. Plugging each into (16), and approximating the i-sum for α-Zipf-distributed θ_i one can show that each, again, scales as $n^{-\beta}$ with $\beta = \frac{\alpha}{1+\alpha}$, but the latter has an additional $n^{-1/2}$ term. For $\alpha < 1$ this is swamped by $n^{-\beta}$, but for $\alpha > 1$ it dominates the learning curve. The intuition for this happening is explained in the main text.

F   Approximating Sums by Integrals

Sums $\sum_{i=1}^{\infty} g(i)$ can be approximated by integrals $\int_1^{\infty} g(x)dx$. To upper bound the approximation error classically requires computing a cumbersome integral (Euler-Maclaurin remainder) or only works for finite sums (Trapezoid rule). In the following we derive an

upper bound on the approximation accuracy, suitable for our purpose. First, note that for a monotone increasing function

$$\int_{n-1}^{m} g(x)dx \;\leq\; \sum_{i=n}^{m} g(i) \;\leq\; \int_{n}^{m+1} g(x)dx \tag{17}$$

with inequalities reversed for monotone decreasing functions. Consider now any measurable function $g : [0;\infty) \to [0;\infty)$ increasing up to $g_{max} = g(x_{max})$ and thereafter decreasing (In our application $g(x) = f(x)e^{-nf(x)}$). Let $i_{m-1} \leq x_{max} \leq i_m$. We split the integral into the increasing and decreasing part and use (17) to lower-bound the error:

$$\sum_{i=1}^{\infty} g(i) \;=\; \sum_{i=1}^{i_{m-1}} g(i) + \sum_{i=i_m}^{\infty} g(i) \;\geq\; \int_{0}^{i_{m-1}} g(x)dx + \int_{i_m}^{\infty} g(x)dx$$

$$=\; \int_{0}^{\infty} g(x)dx - \int_{i_m}^{i_{m-1}} g(x)dx \;\geq\; \int_{0}^{\infty} g(x)dx - g_{max}$$

To obtain an upper bound we have to exclude i_{m-1} and i_m from the sums:

$$\sum_{i=1}^{\infty} g(i) \;=\; \sum_{i=1}^{i_{m-2}} g(i) + \sum_{i=i_{m+1}}^{\infty} g(i) + \; [g(i_{m-1}) + g(i_m)]$$

$$\leq\; \int_{1}^{i_{m-1}} g(x)dx + \int_{i_m}^{\infty} g(x)dx + \; \min\{g(i_{m-1}), g(i_m)\} + \max\{g(i_{m-1}), g(i_m)\}$$

$$\leq\; \int_{1}^{i_{m-1}} g(x)dx + \int_{i_m}^{\infty} g(x)dx + \int_{i_{m-1}}^{i_m} g(x)dx + g_{max} \;=\; \int_{1}^{\infty} g(x)dx + g_{max}$$

Together this leads to the following bound on the approximation error:

$$\left| \sum_{i=1}^{\infty} g(i) - \int_{a}^{\infty} g(x)dx \right| \;\leq\; g_{max} \tag{18}$$

for any=every choice of $a \in [0;1]$. Without further assumptions on g, this bound is tight. For the lower bound consider $g(x) = g_{max}$ for $i_{m-1} < x < i_m$ and 0 otherwise. For the upper bound consider $g(i_m) = g_{max}$ and 0 otherwise.

G   List of Notation

Symbol   **Explanation**

$a/b \cdot c = (a/b) \cdot c$ but $a/bc = a/(bc)$

$[\![\text{Bool}]\!]$ 1 if Bool=True, 0 if Bool=False

$\#\mathcal{S}$ Number of elements in set $\mathcal{S}$

$\mathbb{P}, \mathbb{E}, \mathbb{V}$ Probability, Expectation, Variance

$i \in i_{1:n}$ is short for $i \in \{i_1, ..., i_n\}$

$:=$ defined as

$\equiv$ equal by (earlier) definition

$\approx$ Asymptotically or approximately equal (depending on context)

$\overset{\times}{=}$ Equal within a multiplicative constant

$\overset{\times}{\approx}$ Asymptotically or approximately proportional (depending on context)

$\doteq$ Equal within the stated number of numerical digits

$i, j \in \mathbb{N}$ natural number "feature"

$t, n, m \in \mathbb{N}$ time/sample index

$N \in \mathbb{N}$ sample size

θ_i probability of feature i

A tabular learning algorithm

$h : \mathbb{N} \to \mathcal{Y}$ classifier, e.g. binary $\mathcal{Y} = \{0, 1\}$

$f : \mathbb{R} \to \mathbb{R}$ Theoretical data distribution/scaling $f(i) = \theta_i$

$\mathcal{D}_n$ Data consisting of n (feature,label) pairs

E_n Instantaneous Error of A on i_{n+1} predicting y_{n+1} from $\mathcal{D}_n$

$\mathbb{E}_n$ Expectation of Instantaneous Error E_n w.r.t. $\mathcal{D}_n$

$\bar{\mathsf{E}}_N$ Time-Averaged Error from $n = 1, ..., N$

α Exponent of Zipf distributed data frequency

β Exponent of power law for error as a function of data size

γ Decay rate for exponential data distribution
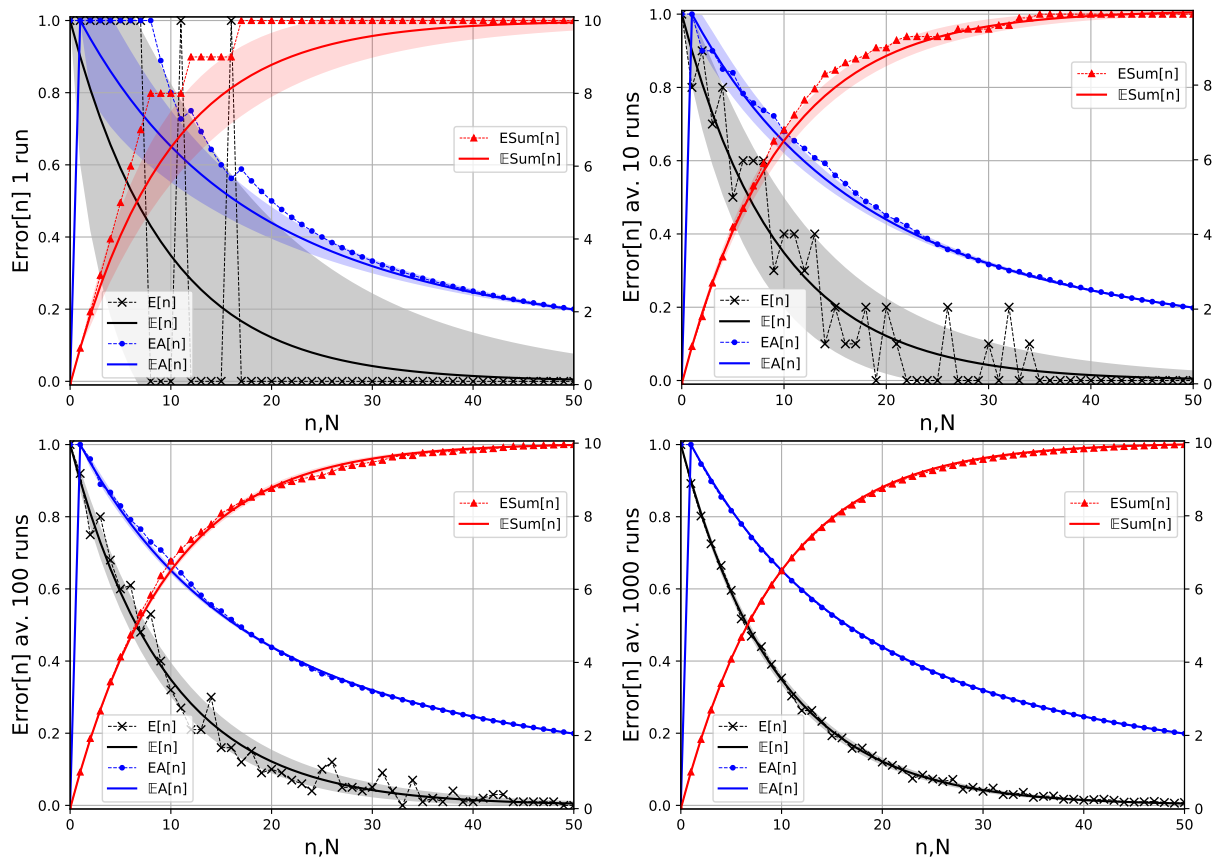
H   More Figures

Figure 4: (**Learning Curves**) for uniform data distribution $\mathbb{P}[i_n = i] = \theta_i = \frac{1}{m}$ for $i \leq m = 10$ averaged over $k = 1, 10, 100, 1000$ runs.
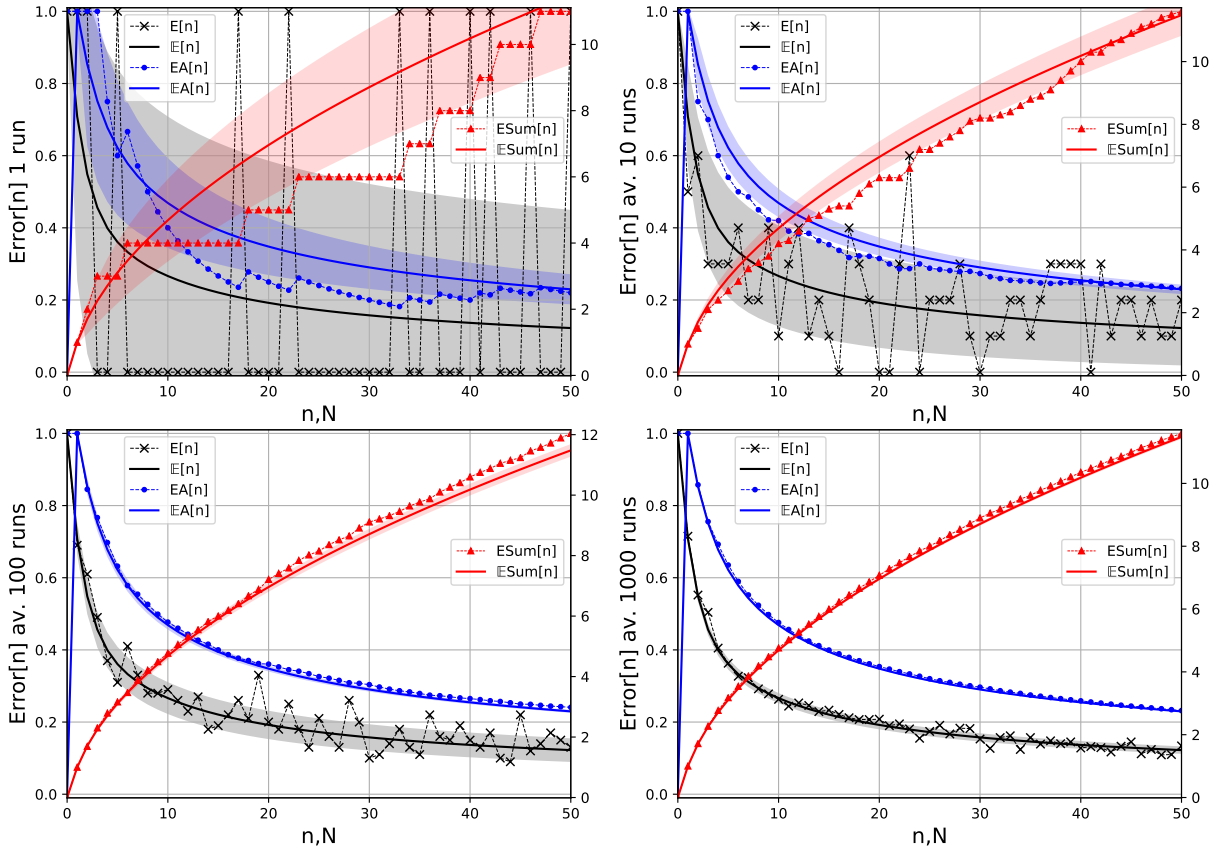
Figure 5: (**Learning Curves**) for Zipf-distributed data $\mathbb{P}[i_n = i] = \theta_i \propto i^{-(\alpha+1)}$ for $\alpha = 1$ averaged over $k = 1, 10, 100, 1000$ runs.
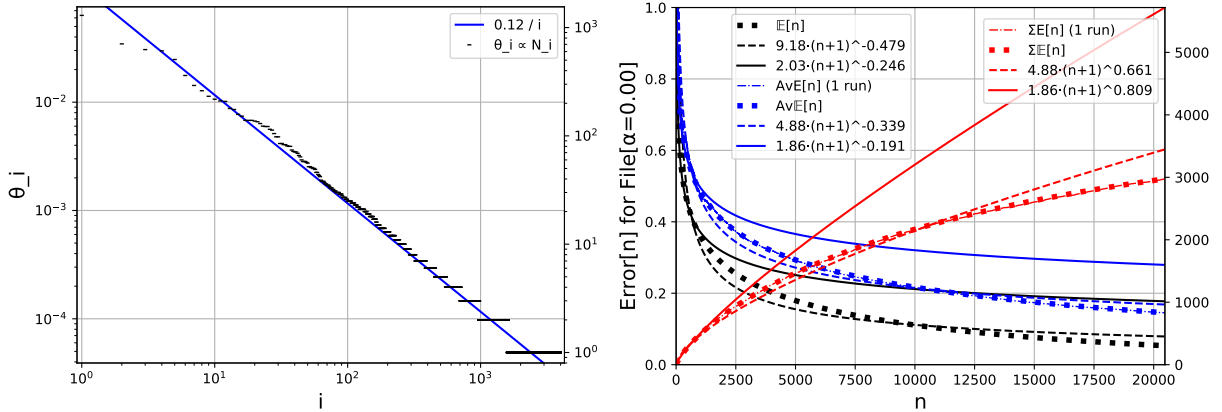

Figure 6: (**Word-Frequency in Text File, Learning Curve, Power Law**) *(left)* Relative (left scale) and absolute (right scale) of frequency of words in the first 20469 words in file 'book1' of the Calgary Corpus, and fitted Zipf law, which is a straight line in the Log-log plot. *(right)* Power law fit to learning curve for this data set for a word classification task. For large n, low frequency words break the Zipf law and hence the power law. The solid line is the same as in Figure 3 (right) fit to the reliable region $n \leq 1000$ and then extrapolated. The dashed line is fit over the whole range of n.