Does Learning Require Memorization?

A Short Tale about a Long Tail*

Vitaly Feldman
Google Research[†]
Mountain View, CA, USA
vitaly.edu@gmail.com

ABSTRACT

State-of-the-art results on image recognition tasks are achieved using over-parameterized learning algorithms that (nearly) perfectly fit the training set and are known to fit well even random labels. This tendency to memorize seemingly useless training data labels is not explained by existing theoretical analyses. Memorization of the training data also presents significant privacy risks when the training data contains sensitive personal information and thus it is important to understand whether such memorization is necessary for accurate learning.

We provide a simple conceptual explanation and a theoretical model demonstrating that for natural data distributions memorization of labels is *necessary* for achieving close-to-optimal generalization error. The model is motivated and supported by the results of several recent empirical works. In our model, data is sampled from a mixture of subpopulations and the frequencies of these subpopulations are chosen from some prior. The model allows to quantify the effect of not fitting the training data on the generalization performance of the learned classifier and demonstrates that memorization is necessary whenever frequencies are long-tailed. Image and text data are known to follow such distributions and therefore our results establish a formal link between these empirical phenomena. Our results also have concrete implications for the cost of ensuring differential privacy in learning.

CCS CONCEPTS

• Theory of computation → Models of learning; Sample complexity and generalization bounds.

KEYWORDS

Generalization, Interpolation, Long-tailed Distribution, Overfitting, Privacy-preserving Learning

*Extended abstract. Full version appears as [24].

[†]Now at Apple, Cupertino, CA, USA. Part of this work was done while the author was visiting the Simons Institute for the Theory of Computing, Berkeley, CA, USA.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
STOC '20, June 22–26, 2020, Chicago, IL, USA
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6979-4/20/06.
<https://doi.org/10.1145/3357713.3384290>

ACM Reference Format:

Vitaly Feldman. 2020. Does Learning Require Memorization? A Short Tale about a Long Tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC '20)*, June 22–26, 2020, Chicago, IL, USA. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3357713.3384290>

1 INTRODUCTION

Understanding the generalization properties of learning systems based on deep neural networks (DNNs) is an area of great practical importance and significant theoretical interest. The main conceptual hurdle to adapting the classical approaches for analysis of generalization is the well-known fact that state-of-the-art approaches to training DNNs reach zero (or very low) training error even when the test error is relatively high. In fact, as highlighted in the influential work of Zhang *et al.*[54], low training error is achieved even when the labels are generated at random. The only way to fit an example whose label cannot be predicted based on the rest of the dataset is to effectively memorize it. In this work we will formalize and quantify this notion of memorization. For now we will informally say that a learning algorithm memorizes the label of some example (x, y) in its dataset S if the model output on S predicts y on x whereas when the learning algorithm is trained on S without (x, y) it is unlikely to predict y on x .

The classical approach to understanding generalization starts with the decomposition of the generalization error $\text{err}_P(h)$ relative to the data distribution P into the empirical error on the training set S and generalization gap (or estimation error):

$$\text{err}_P(h) = \text{err}_S(h) + (\text{err}_P(h) - \text{err}_S(h)).$$

The generalization gap is then upper bounded via some proxy that measures the algorithm's ability to fit arbitrary labels, for example, model complexity or algorithmic stability. For each training example whose label is memorized by the algorithm, the capacity to fit needs to increase at least as much as the training error decreases. Further, a proxy used for theoretical analysis needs to be conservative and tend to grow faster than the actual increase in the algorithm's ability to fit. Thus standard theoretical analyses suggest that algorithms which memorize labels are balancing the empirical error and the ability to fit in a suboptimal way. In particular, they cannot explain why modern ML relies on such algorithms.

This captivating disconnect between the classical theory and modern ML practice has attracted significant amount of research and broad interest in recent years (see Sec. 1.4 for an overview). At the same time the phenomenon is far from new. Random forests [14] and Adaboost [27] are known to achieve their optimal generalization error on many learning problems while fitting the training data perfectly [44, 45, 52]. There is also recent evidence that this holds for kernel methods in certain regimes as well [11, 32, 54].

1.1 Our Contribution

We propose a conceptually simple explanation and supporting theory for why memorization of labels may be necessary to achieve close-to-optimal generalization error. It is based on the view that the primary hurdle to learning an accurate model is not the noise inherent in the labels but rather an insufficient amount of data to predict accurately on rare and atypical instances. Such instances are usually referred in practice as the “long tail” of the data distribution. It has been widely observed that modern datasets used for visual object recognition and text labeling follow the classical long-tailed distributions such as Zipf distribution (or more general power law distributions).

To formalize the notion of having a “long tail” we will model the data distribution of each class as a mixture of distinct subpopulations. For example, images of birds include numerous different species photographed from different perspectives and under different conditions (such as close-ups, in foliage and in the sky) [49]. Naturally, the subpopulations may have different frequencies (which correspond to mixture coefficients). We model the informal notion of long-tailed data distributions as distributions in which the frequencies of subpopulations are long-tailed. The long-tailed nature of subpopulation frequencies is known in datasets for which additional human annotations are available. A detailed discussion of this phenomenon in the SUN object detection benchmark [53] can be found in the work of Zhu *et al.*[55].

Additional evidence that classes can be viewed as long-tailed mixtures of subpopulations comes from extreme multiclass problems. Specifically, these problems often have more than 10,000 fine-grained labels and the number of examples per class is long-tailed [3, 4, 21, 30, 49, 51]. Observe that fine-grained labels in such problems correspond to subcategories of coarser classes (for example, different species of birds all correspond to the “bird” label in a coarse classification problem). We also remark that subpopulations do not have to directly correspond to human-definable categories. They are the artifacts of the representation used by the learning algorithm which are often relatively low-level.

It is natural to assume that before seeing the dataset the learning algorithm does not know the frequencies of subpopulations and may not be able to predict accurately on a subpopulation without observing any examples from it. A dataset of n samples from a long-tailed mixture distribution will have some subpopulations from which just a single one example was observed (and some subpopulations from which none at all). By our assumption, to fit such an example the learning algorithm will need to memorize the label. The question is whether this is necessary for achieving close-to-optimal generalization error. If the example comes from an extremely rare (or “outlier”) subpopulation then memorizing it has no significant benefits. At the same time, it is possible that the example comes from a subpopulation with frequency on the order of $1/n$. Fitting such an example is likely to improve the accuracy on the entire subpopulation and thus has a measurable effect on the generalization error.

The key point of this work is that the only information that the algorithm has about the frequency of a subpopulation comes from the empirical frequency of its representatives. Thus given a unique representative it is statistically impossible to infer whether the

true frequency is on the order of $1/n$ or much smaller. In addition, having a single representative from a subpopulation can make it impossible for the learning algorithm to tell whether the provided label of that example is accurate. Therefore an algorithm can only avoid the risk of missing subpopulations with significant frequency by also memorizing examples from the “outlier” subpopulations and mislabeled examples. Importantly, in a long-tailed distribution of frequencies, the total weight of frequencies on the order of $1/n$ is substantial enough that ignoring these subpopulations will hurt the generalization error. Thus, for such distributions, an algorithm needs to be able to memorize the labels in order to achieve close-to-optimal generalization.

1.2 Technical Overview

On a technical level our primary contribution is turning this intuitive but informal explanation into a formal model that allows to quantify the trade-offs involved. This model also allows to quantify the cost of ensuring differential privacy (and even weaker forms of stability) when learning from natural data distributions.

Our starting point is a simple model for classification problems that incorporates the long tail of frequencies in the data distribution. The goal of the model is to isolate the discussion of the effect of memorization on the accuracy from other aspects of modeling subpopulations. More formally, in our problem the domain X is unstructured and has size N (each point will correspond to a subpopulation in the more general model). As usual, we will assume that the labeling function belongs to some class of functions F known to the learning algorithm. We will be primarily interested in the setting where F is rich (or computationally hard) enough that for a significant fraction of the points the learning algorithm cannot predict the label of a point well without observing it in the dataset. In particular, fitting some of the examples will require memorizing their labels.

Nothing is known a priori about the frequency of any individual point aside from a prior distribution over the frequencies described by a list of N frequencies $\pi = (\pi_1, \dots, \pi_N)$. Our results are easiest to express when the objective of the learning algorithm is to minimize the expectation of the error over a random choice of the marginal distribution D over X from some distribution $\mathcal{D}$ (instead of the more usual worst-case error). In addition, for convenience of notation we will also measure the error with respect to a random choice of the labeling function from some distribution $\mathcal{F}$ over F . That is, the objective of a learning algorithm $\mathcal{A}$ is defined as:

$$\overline{\text{err}}(\mathcal{D}, \mathcal{F}, \mathcal{A}) := \mathbf{E}_{D \sim \mathcal{D}, f \sim \mathcal{F}} \left[\mathbf{E}_{S \sim (D, F)^n, h = \mathcal{A}(S)} \left[\mathbf{Pr}_{x \sim D} [h(x) \neq f(x)] \right] \right].$$

Specifically, we consider the following distribution over marginal distributions on X : the frequency of each point in the domain is chosen randomly and independently from the prior π of individual frequencies and then normalized to 1. This process results in a distribution $\mathcal{D}$ over marginal distributions that is similar to choosing the frequencies of the elements to be a random permutation of the elements of π . Models measuring the worst-case error over all the permutations of a list of frequencies underlie the recent breakthroughs in the analysis of density estimation algorithms [40, 48]. We believe that our results can be extended to this worst-case model as well and leave such an extension for future work. (The extension

to measuring the worst-case error over the choice of $f \in F$, on the other hand, is straightforward).

Our main result directly relates the number of points that an algorithm does not fit to the sub-optimality (or excess error) of the algorithm via a quantity that depends only on the frequency prior π and n . Here excess error is measured relative to the optimal algorithm and not relative to the best model in some class. Formally, we denote by $\text{err}_S(\mathcal{A}, 1)$ the number of examples that appear once in the dataset S and are mislabeled by the classifier that $\mathcal{A}$ outputs on S . A special case of our theorem states:

$$\overline{\text{err}}(\pi, \mathcal{F}, \mathcal{A}) \geq \text{opt}(\pi, \mathcal{F}) + \tau_1 \cdot \mathbb{E}[\text{err}_S(\mathcal{A}, 1)]. \quad (1)$$

Here $\overline{\text{err}}(\pi, \mathcal{F}, \mathcal{A})$ refers to the expected generalization error of $\mathcal{A}$ and $\text{opt}(\pi, \mathcal{F})$ is the minimum achievable error by any algorithm (expectations are with respect to the process that generates the learning problem and also sampling of the dataset). The important quantity here is

$$\tau_1 := \frac{\mathbb{E}_{\alpha \sim \bar{\pi}^N} [\alpha^2 \cdot (1 - \alpha)^{n-1}]}{\mathbb{E}_{\alpha \sim \bar{\pi}^N} [\alpha \cdot (1 - \alpha)^{n-1}]},$$

where $\bar{\pi}^N$ is the actual marginal distribution over frequencies that results from our process and is, basically, a slightly smoothed version of π . We note that the optimal algorithm in this case does not depend on π and thus our modeling does not require the learning algorithm to know π to achieve near-optimal generalization error.

The quantity τ_1 is easy to compute given π . As a quick numerical example, for the prototypical long-tailed Zipf distribution (where the frequency of the i -th most frequent item is proportional to $1/i$) over the universe of size $N = 100,000$ and $n = 10,000$ samples, one gets the expected loss of at least $\approx 0.23/n$ per every example the learner does not fit. For comparison, the worst-case loss in this setting is determined by the least frequent element and is close to 0. Given that the expected fraction of samples that appear once is $\approx 35\%$, an algorithm that does not fit well will be suboptimal by $\approx 8\%$ (with the optimal top-1 error for 10 balanced classes being $\approx 31\%$ in this case). More generally, we show that τ_1 can be lower bounded by the total weight of the part of the prior π which has frequency on the order of $1/n$ and also that the absence of frequencies on this order will imply negligible τ_1 .

Continuous data distributions: Naturally, our simple setting in which individual points have significant probability cannot be applied to continuous and high-dimensional ML problems where each individual point has an exponentially small (in the dimension) probability. In this more general setting the prediction on the example itself has negligible effect on the generalization error. To show how the effects we demonstrated in the simple discrete setting can extend to continuous distributions, we consider mixture models of subpopulations. In our model, the frequencies of subpopulations (or mixture coefficients) are selected randomly according to the prior π as before. The labeling function is also chosen as before and is assumed to be constant over every subpopulation.

The discussion of fitting makes sense only if one assumes that the prediction on the data point in the dataset will affect the predictions on related points. Thus we assume that (with high probability) the learning algorithm's prediction on a single point from a subpopulation will be correlated with the prediction on a random example

from the same subpopulation. We refer to this condition as coupling and show that eq. (1) still holds up to the adjustment for the strength of the coupling.

Intuitively, it is clear that this form of "coupling" is likely to apply to "local" learning rules such as the nearest neighbors algorithm. Indeed, our assumption can be seen as a more abstract version of geometric smoothness conditions on the marginal distribution of the label used in analysis of such methods (e.g. [17]). We also show that it applies to linear predictors/SVMs in high dimension provided that distinct subpopulations are sufficiently uncorrelated. Deep neural networks are known to have some of the properties of both nearest neighbor rules and linear classifiers in the last-hidden-layer representation (e.g. [18]). Thus they are likely to exhibit this type of coupling as well.

Noisy labels: Our analysis so far does not consider some of the potentially significant costs of fitting noisy examples. We first note that random label noise is easy to incorporate into the model. The optimal strategy in our setting is to predict the argmax of the posterior distribution over the label of each point. If it holds that observing an example (x, y) makes the posterior probability of label y larger by at least some Δ than the posterior probability of any of the other labels (at x) then we will obtain the same result but with excess error scaled by Δ . In other words, despite the label noise, memorization might still be beneficial for these examples. At the same time, memorization of noisy examples *within* a frequent subpopulation is likely to be harmful for generalization and would not be explained by our model. Memorization of such examples by some algorithms can be explained by the fact that the learning algorithms we use are tuned for data distribution in which noise on typical examples is extremely rare. For such data the benefits of memorization outweigh the costs of memorizing some harmful noise.

From fitting to memorization and privacy: Our results so far have expressed suboptimality in terms of fitting the data. To relate these results to memorization we give a simple definition of what memorizing a label of a point in the dataset means (we are not aware of a prior formal definition of this notion). Formally, for a dataset $S = (x_i, y_i)_{i \in [n]}$ and $i \in [n]$ define

$$\text{mem}(\mathcal{A}, S, i) := \left| \Pr_{h \sim \mathcal{A}(S)} [h(x_i) = y_i] - \Pr_{h \sim \mathcal{A}(S^{(i)})} [h(x_i) = y_i] \right|,$$

where $S^{(i)}$ denotes the dataset that is S with (x_i, y_i) removed. This definition is closely related to the classical leave-one-out notion of stability but focuses on the change in the label and not the incurred loss. An immediate corollary of this definition is that an algorithm that cannot memorize labels will be unable to fit the examples whose labels it cannot predict based on the rest of the dataset. Two natural situations in which the algorithm will not be able to predict these labels are learning a complex labeling function (e.g. having large VC dimension) and computational hardness of finding a simple model of the data. A direct corollary of our results is that in these situations an algorithm that does not memorize labels will not be able to reach close-to-optimal generalization error. In particular, in these conditions differentially private algorithms will not achieve the same accuracy as non-private algorithms.

1.3 Known Empirical Evidence

The best results (that we are aware of) on modern benchmarks that are achieved without interpolation are those for differentially private (DP) training algorithms [1, 34, 41, 42]. While not interpolating is not the goal, the properties of DP imply that a DP algorithm with the privacy parameter $\epsilon = O(1)$ cannot memorize individual labels. Moreover, they result in remarkably low gap between the training and test error that is formally explained by the generalization properties of DP [22]. However, the test error results achieved in these works are well below the state-of-the-art using similar models and training algorithms. For example, Papernot *et al.* [42] report accuracy of 98% and 82.7% on MNIST and SVHN as opposed to 99.2% and 92.8%, respectively when training the same models without privacy.

The motivation and inspiration for this work comes in part from attempts to understand why do DP algorithms fall short of their non-private counterparts and which examples are they more likely to misclassify. A thorough and recent exploration related to this question can be found in the work of Carlini *et al.* [16]. They consider different ways to measure how “prototypical” each of the data points is according to several natural metrics and across MNIST, CIFAR-10, Fashion-MNIST and ImageNet datasets and compare between these metrics. One of those metrics is the highest level of privacy that a DP training algorithm can achieve while still correctly classifying an example that is correctly classified by a non-private model. As argued in that work and is clear from their comprehensive visualization, the examples on which a DP model errs are either outliers or atypical ones. In addition, the metric based on DP is well correlated with other metrics of being prototypical such as relative confidence of the (non-private) model and human annotation. Their concepts of most and least prototypical map naturally to the frequency of subpopulation in our model. Thus their work supports the view that the reason why learning with DP cannot achieve the same accuracy as non-private learning is that it cannot memorize the tail of the mixture distribution. This view also explains the recent empirical results showing that the decrease in accuracy is larger for less well represented subpopulations [5].

Another empirical work that provides indirect support for our theory is [2]. It examines the relationship between memorization of random labels and performance of the network for different types of regularization techniques. The work demonstrates that for some regularization techniques it is possible to reduce the ability of the network to fit random labels without significantly impacting their performance on true labels. The explanation proposed for this finding is that memorization is not necessary for learning. While it may appear to contradict our theory, a closer look at the result suggests the opposite conclusion. On the true labels almost all their regularization techniques still reach near perfect train accuracy with test accuracy of at most 78%. The only two techniques that do not quite interpolate (though still reaching around 97% train accuracy) are *exactly* the ones that do exhibit clear correlation between ability to fit random labels and test accuracy (see “input binary mask” and “input gaussian” in their Figs. 10,11). We remark (and elaborate in Section 2) that fitting random examples or even interpolation are not necessary conditions for the application of our approach and for memorization being beneficial.

Our theory also makes a number of predictions that merit empirical investigation. For example, the model predicts that for typical image and text datasets and learning algorithms, there is a significant fraction of examples whose memorization is necessary for predicting accurately on examples from the same subpopulation in the test set. More formally, there exist examples in the training set such that for each of them (1) the label is memorized by the learning algorithm in the sense defined above; (2) there exists a dependent example in the test set in the following sense: the accuracy of the model on the dependent test example drops significantly when the corresponding example from the training set is removed (with no significant effect on the accuracy on the other test examples). This prediction has been already confirmed on MNIST, CIFAR-10/100, SVHN and ImageNet datasets revealing numerous visually similar pairs of relatively atypical examples [26].

1.4 Related Work

One line of research motivated by the empirical phenomena we discuss here studies implicit regularization in the overparameterized regime (namely, when the parameter space is large enough that the learning algorithm can perfectly fit the dataset). For example, the classical margin theory [19, 44, 50] for SVMs and boosting suggests that, while the ambient dimension is large, the learning algorithm implicitly maximizes the margin. The generalization gap can then be upper bounded in terms of the margin. Examples of this approach in the context of DNNs can be found in [6, 31, 37–39] (and references therein). These notions imply that it is beneficial to overparameterize and suffice for explaining why the training algorithm will select the best model among those that do fit the training set. However implicit regularization does not explain why, despite the regularization, the training error is near zero even when the generalization error is large.

Another line of research studies generalization properties of learning algorithms that fit the training data perfectly, often referred to as *interpolating* [10, 11]. For example, a classical work of Cover and Hart [20] gives bounds on the generalization error of the 1-nearest neighbor algorithm. Recent wave of interest in such methods has lead to new analyses of existing interpolating methods as well as new algorithmic techniques [7, 9–12, 29, 32, 35, 43, 52]. These works bypass the classical approach to generalization outlined above and demonstrate that interpolating methods can generalize while tolerating some amount of noise. In particular, they show that interpolation can be “harmless” in the sense that interpolating methods can in some cases achieve asymptotically optimal generalization error. At the same time, for the problems studied in these works there also exist non-interpolating algorithms with the same (or better) generalization guarantees. Thus these works do not explain why interpolation is common in the first place. We also remark that while interpolating the training set (with high generalization error) requires memorization, memorization also occurs without interpolation. For example, experiments of Zhang *et al.* [54] show that 9% training error is achieved by a standard deep learning algorithm on completely randomly labeled 1000-class ImageNet dataset (with generalization error being 99.9%).

It is known that in the convex setting SGD converges faster when all the loss functions have a joint minimizer [36, 47] and therefore it

has been suggested that interpolation is the result of computational benefits of optimization via SGD [33]. However this hypothesis is not well supported by empirical evidence since interpolation does not appear to significantly affect the speed with which the neural networks are trained [54]. In addition, methods like nearest neighbors, boosting, and bagging are not trained via SGD but tend to interpolate the data as well.

Algorithmic stability [13, 25, 28, 46] is essentially the only general approach that is known to imply generalization bounds beyond those achievable via uniform convergence [23, 46]. However it runs into exactly the same conceptual issue as capacity-based bounds: average stability needs to be increased by at least $1/n$ to fit an arbitrary label. In fact, an interpolating learning algorithm does not satisfy any non-trivial uniform stability (but may still be on-average stable). Recent work of Belkin *et al.* [8] shows empirically and also on simple synthetic models [9] that classical bias-variance trade-off curve may coexist with the interpolating regime in which the generalization error improves as the class capacity grows. Thus the generalization error as a function of some way to measure model capacity is a “double-descent” curve. The proposed explanation is that the over-parameterized regime allows the learning algorithm to implicitly rely on inductive bias that is better suited to the data distribution (much like increasing the dimension may help find a solution with a larger margin). While the phenomenon deserves further understanding and the proposed explanation is compelling, it still does not explain why the model returned by this method interpolates the training data (that is, why the training error does not increase once the appropriate inductive bias is exploited).

2 DISCUSSION

Our work provides a natural and simple learning model in which memorization of labels and, in some cases interpolation, are necessary for achieving nearly optimal generalization when learning from a long-tailed data distribution. It suggests that the reason why many modern ML methods reach their best accuracy while (nearly) perfectly fitting the data is that these methods are (implicitly) tuned to handle the long tails of natural data distributions. Our model explicitly incorporates the prior distribution on the frequencies of subpopulations in the data and we argue that such modeling is necessary to avoid the disconnect between the classical view of generalization and the practice of ML. We hope that the insights derived from our approach will serve as the basis for future theoretical analyses of generalization that more faithfully reflect modern datasets and learning techniques. A recent example that such modeling has practical benefits can be found in [15].

ACKNOWLEDGEMENTS

Part of the inspiration and motivation for this work comes from empirical observations that differentially private algorithms have poor accuracy on atypical examples. I’m grateful to Nicholas Carlini, Ulfar Erlingsson and Nicolas Papernot for numerous illuminating discussions of experimental work on this topic [16] and to Vitaly Shmatikov for sharing his insights on this phenomenon in the context of language models. I would like to thank my great colleagues Peter Bartlett, Misha Belkin, Olivier Bousquet, Edith Cohen, Roy Frostig, Daniel Hsu, Phil Long, Yishay Mansour, Mehryar Mohri,

Tomer Koren, Sasha Rakhlin, Adam Smith, Kunal Talwar, Greg Valiant, Chiyuan Zhang and anonymous STOC reviewers for insightful feedback and suggestions on this work.

REFERENCES

- [1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 308–318.
- [2] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR.org, 233–242.
- [3] Rohit Babbar and Bernhard Schölkopf. 2017. Dismec: Distributed sparse machines for extreme multi-label classification. In *Proceedings of the tenth ACM international conference on web search and data mining*. ACM, 721–729.
- [4] Rohit Babbar and Bernhard Schölkopf. 2019. Data scarcity, robustness and extreme multi-label classification. *Machine Learning* (18 Mar 2019).
- [5] Eugene Bagdasaryan and Vitaly Shmatikov. 2019. Differential Privacy Has Disparate Impact on Model Accuracy. *CoRR* abs/1905.12101 (2019). [arXiv:1905.12101](http://arxiv.org/abs/1905.12101)
- [6] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. 2017. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*. 6240–6249.
- [7] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. 2019. Benign Overfitting in Linear Regression. *arXiv preprint arXiv:1906.11300* (2019).
- [8] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2018. Reconciling modern machine learning and the bias-variance trade-off. *arXiv preprint arXiv:1812.11118* (2018).
- [9] Mikhail Belkin, Daniel Hsu, and Ji Xu. 2019. Two models of double descent for weak features. *arXiv preprint arXiv:1903.07571* (2019).
- [10] Mikhail Belkin, Daniel J Hsu, and Partha Mitra. 2018. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. In *Advances in Neural Information Processing Systems*. 2300–2311.
- [11] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. 2018. To Understand Deep Learning We Need to Understand Kernel Learning. In *ICML (Proceedings of Machine Learning Research)*, Vol. 80. PMLR, 541–549. <http://proceedings.mlr.press/v80/belkin18a.html>
- [12] Mikhail Belkin, Alexander Rakhlin, and Alexandre B Tsybakov. 2018. Does data interpolation contradict statistical optimality? *arXiv preprint arXiv:1806.09471* (2018).
- [13] Olivier Bousquet and André Elisseeff. 2002. Stability and generalization. *JMLR* 2 (2002), 499–526.
- [14] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.
- [15] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arachis, and Tengyu Ma. 2019. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. *arXiv preprint arXiv:1906.07413* (2019).
- [16] Nicholas Carlini, Ulfar Erlingsson, and Nicolas Papernot. 2018. Prototypical Examples in Deep Learning: Metrics, Characteristics, and Utility. (2018). <https://openreview.net/forum?id=r1yx3R9tQ>
- [17] Kamalika Chaudhuri and Sanjoy Dasgupta. 2014. Rates of Convergence for Nearest Neighbor Classification. In *NIPS*. 3437–3445. <http://papers.nips.cc/paper/5439-rates-of-convergence-for-nearest-neighbor-classification>
- [18] Gilad Cohen, Guillermo Sapiro, and Raja Giryes. 2018. DNN or k-NN: That is the Generalize vs. Memorize Question. *arXiv preprint arXiv:1805.06822* (2018).
- [19] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.
- [20] Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory* 13, 1 (1967), 21–27.
- [21] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. 2018. Large Scale Fine-Grained Categorization and Domain-Specific Transfer Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [22] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. 2014. Preserving Statistical Validity in Adaptive Data Analysis. *CoRR* abs/1411.2664 (2014). Extended abstract in STOC 2015.
- [23] Vitaly Feldman. 2016. Generalization of ERM in Stochastic Convex Optimization: The Dimension Strikes Back. *CoRR* abs/1608.04414 (2016). <http://arxiv.org/abs/1608.04414> Extended abstract in NIPS 2016.
- [24] Vitaly Feldman. 2019. Does Learning Require Memorization? A Short Tale about a Long Tail. *CoRR* abs/1906.05271 (2019). [arXiv:1906.05271](http://arxiv.org/abs/1906.05271) <http://arxiv.org/abs/1906.05271>
- [25] Vitaly Feldman and Jan Vondrák. 2019. High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. *CoRR* abs/1902.10710 (2019). [arXiv:1902.10710](http://arxiv.org/abs/1902.10710) <http://arxiv.org/abs/1902.10710>

- [26] Vitaly Feldman and Chiyuan Zhang. 2019. Finding the memorized examples via fast influence estimation (working title). Unpublished manuscript. Short summary in <https://www.youtube.com/watch?v=YWy2Iwn-1S8>.
- [27] Y. Freund and R. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. System Sci.* 55, 1 (1997), 119–139.
- [28] Moritz Hardt, Ben Recht, and Yoram Singer. 2016. Train faster, generalize better: Stability of stochastic gradient descent. In *ICML*. 1225–1234. <http://jmlr.org/proceedings/papers/v48/hardt16.html>
- [29] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. 2019. Surprises in High-Dimensional Ridgeless Least Squares Interpolation. *arXiv preprint arXiv:1903.08560* (2019).
- [30] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision* 123, 1 (2017), 32–73.
- [31] Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. 2018. Algorithmic Regularization in Over-parameterized Matrix Sensing and Neural Networks with Quadratic Activations. In *Conference On Learning Theory*. 2–47.
- [32] Tengyuan Liang and Alexander Rakhlin. 2018. Just interpolate: Kernel" ridgeless" regression can generalize. *arXiv preprint arXiv:1808.00387* (2018).
- [33] Siyuan Ma, Raef Bassily, and Mikhail Belkin. 2018. The Power of Interpolation: Understanding the Effectiveness of SGD in Modern Over-parametrized Learning. In *ICML*. 3331–3340. <http://proceedings.mlr.press/v80/ma18a.html>
- [34] Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning Differentially Private Recurrent Language Models. In *International Conference on Learning Representations (ICLR)*. <https://openreview.net/pdf?id=BJ0hF1Z0b>
- [35] Vidya Muthukumar, Kailas Vodrahalli, and Anant Sahai. 2019. Harmless interpolation of noisy data in regression. *arXiv preprint arXiv:1903.09139* (2019).
- [36] Deanna Needell, Rachel Ward, and Nati Srebro. 2014. Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz algorithm. In *NIPS*. 1017–1025. <http://papers.nips.cc/paper/5355-stochastic-gradient-descent-weighted-sampling-and-the-randomized-kaczmarz-algorithm.pdf>
- [37] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. 2017. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*. 5947–5956.
- [38] Behnam Neyshabur, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. 2017. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071* (2017).
- [39] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. 2015. In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning. In *ICLR*. <http://arxiv.org/abs/1412.6614>
- [40] Alon Orlitsky and Ananda Theertha Suresh. 2015. Competitive distribution estimation: Why is good-turing good. In *NIPS*. 2143–2151.
- [41] Nicolas Papernot, Martin Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. 2016. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. *CoRR* abs/1610.05755 (2016). arXiv:1610.05755 <http://arxiv.org/abs/1610.05755>
- [42] Nicolas Papernot, Martin Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. 2017. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.
- [43] Alexander Rakhlin and Xiyu Zhai. 2019. Consistency of Interpolation with Laplace Kernels is a High-Dimensional Phenomenon. In *COLT*, Vol. 99. PMLR, 2595–2623. <http://proceedings.mlr.press/v99/rakhlin19a.html>
- [44] R. Schapire, Y. Freund, P. Bartlett, and W. Lee. 1998. Boosting the margin: a new explanation for the effectiveness of voting methods. *Annals of Statistics* 26, 5 (1998), 1651–1686.
- [45] Robert E Schapire. 2013. Explaining adaboost. In *Empirical inference*. Springer, 37–52.
- [46] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. 2009. Stochastic Convex Optimization. In *COLT*.
- [47] Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. 2010. Smoothness, Low Noise and Fast Rates. In *NIPS*. 2199–2207. <http://papers.nips.cc/paper/3894-smoothness-low-noise-and-fast-rates.pdf>
- [48] Gregory Valiant and Paul Valiant. 2016. Instance optimal learning of discrete distributions. In *STOC*. ACM, 142–155.
- [49] Grant Van Horn and Pietro Perona. 2017. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450* (2017).
- [50] V. N. Vapnik. 1982. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York.
- [51] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. 2017. Learning to model the tail. In *Advances in Neural Information Processing Systems*. 7029–7039.
- [52] Abraham J Wyner, Matthew Olson, Justin Bleich, and David Mease. 2017. Explaining the success of adaboost and random forests as interpolating classifiers. *The Journal of Machine Learning Research* 18, 1 (2017), 1558–1590.
- [53] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 3485–3492.
- [54] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *ICLR*. <https://openreview.net/forum?id=Sy8gdB9xx>
- [55] Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. 2014. Capturing long-tail distributions of object subcategories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 915–922.