Does Learning Require Memorization? A Short Tale about a Long Tail

Vitaly Feldman*
Google Research, Brain Team

Abstract

State-of-the-art results on image recognition tasks are achieved using over-parameterized learning algorithms that (nearly) perfectly fit the training set. This phenomenon is referred to as data interpolation or, informally, as memorization of the training data. The question of why such algorithms generalize well to unseen data is not adequately addressed by the standard theoretical frameworks and, as a result, significant theoretical and experimental effort has been devoted to understanding the properties of such algorithms.

We provide a simple and generic model for prediction problems in which interpolating the dataset 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.

To the best of our knowledge, this is the first general framework that demonstrates statistical benefits of plain memorization for learning. Our results also have concrete implications for the cost of ensuring differential privacy in learning.

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 near-zero) training error. Moreover, as highlighted in the influential work of Zhang et al. [Zha+17], the zero training error is achieved even when the labels are generated at random. This phenomenon is referred to as interpolation [BHM18; BMM18] and is closely related to the concept of memorization since fitting a point with an arbitrary label requires a way to (implicitly) single out every point in the dataset and record its label.

The classical approach to understanding generalization is based on the view in which the learning algorithm selects a model from a collection of progressively richer classes of models. The algorithm needs to (explicitly or implicitly) balance the gains from better fitting the training data with the loss of using a model for which the potential gap between the training error and generalization error is larger. Perfect fitting of

*Part of this work was done while the author was visiting the Simons Institute for the Theory of Computing.

random labels implies that this balance is tuned in such a way that the learning algorithm will just memorize the labels if needed (since there is no pattern to learn). Learning is supposed to be about finding patterns and not just plain memorization so this makes no sense! So why does modern ML rely on algorithms tuned to memorize whenever needed?

This captivating contradiction between the classical theory and modern ML practice has attracted an immense amount of attention in recent years. At the same time the phenomenon is far from new. 1-Nearest neighbor [CH+67], random forests [Bre01], and Adaboost [FS97] are all known to achieve their optimal generalization error in the interpolation regime on many learning problems as well as fit random noise [Sch+98; Sch13; Wyn+17] (there are some recent examples for kernel methods as well [Zha+17; BMM18; LR18]). While specific models of the data distribution that ensure good generalization properties for some of these techniques are known (e.g. [CH+67; Wyn+17; LR18]), the focus of those works is to show that interpolation is not necessarily harmful. In particular, they do not preclude existence of algorithms for the same problem that achieve the same (or better) generalization error without interpolation.

A lot of work motivated by this question, relies on (implicit) regularization of the model complexity by the training algorithm. For example, the classical margin theory [Vap82; CV95; Sch+98] for SVMs and boosting suggests that while the dimension is large, generalization error is small due to classification with large margin. Examples of this approach in the context of DNNs can be found in [Ney+17a; BFT17; Ney+17b] (and references therein). While these notions suffice for explaining why the training algorithm will select the best model among those that *do interpolate*, it does not explain why, despite such regularization, the training error is ≈ 0.

1.1 Our contribution

We propose a conceptually simple explanation and supporting theory for why perfectly fitting the data and memorization more broadly may be necessary to achieve close-to-optimal generalization error. Unlike much of the classical theory, it is based on the view that the main hurdle to learning an accurate model is not the noise inherent in the labels but rather uncertainty due to an insufficient amount of data. Without a very large amount of data, it is naturally much harder to predict accurately on rare and atypical instances. At the same time, it has been widely observed that modern datasets used for visual object recognition and NLU tasks such as next word prediction have a long tail[1] of subpopulations [ZAR14; WRH17; VHP17; Cui+18]. For example, images of birds in the CIFAR-10 dataset include numerous different species of bird photographed from different perspectives and under different conditions (such as close-ups, in foliage and in the sky). Such subpopulations do not necessarily 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. In such "long-tailed" distributions, rare and atypical instances make up a large fraction of the data distribution (or *population*).

To reflect this property of the data it is natural to view the data distribution of points corresponding to each class as a mixture of distinct subpopulations. Importantly, not all subpopulations occur with similar frequency: some are common (or prototypical) whereas some other ones are exceedingly rare. We will be interested in understanding the case when the distribution of the frequencies of these subpopulations is long-tailed. It is also reasonable 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 mixture distribution might have some subpopulations from which few samples were observed, possibly just a single one. Therefore just a

[1] For now we use the term "long-tailed" informally.

single example might be the primary source of reliable information about the label of that subpopulation and without memorizing (and fitting) the example the model might be less accurate on the entire subpopulation of such an example.

The question is whether such an example is necessary for achieving close-to-optimal generalization error. If that sample comes from a extremely rare (or "outlier") subpopulation then not-memorizing it will not have a significant effect on the generalization error (and potentially help avoid memorizing noise). At the same time, the example may also come from a "borderline" subpopulation with frequency on the order of $1/n$. Thus being less accurate on such subpopulation will have a measurable effect on the generalization error (potentially increasing the generalization error by $\Omega(1/n)$). The key point of this work is that based on observing a single sample from a subpopulation, it is impossible to distinguish samples from "borderline" populations from those in the "outlier" ones. Therefore an algorithm can only avoid the risk of missing "borderline" subpopulations by also fitting the "outliers". In particular, an algorithm that does not fit individual samples will not have accurate predictions on subpopulations from which only a single sample was observed. In a long-tailed distribution of frequencies, the total weight of frequencies on the order of $1/n$ is significant enough that ignoring these subpopulations will hurt the generalization error substantially. Thus, for such distributions, an algorithm needs to be able to memorize the labels in order to achieve close-to-optimal accuracy.

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 and to prove the high-level statements we make. 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.

Our model is a variant of standard statistical learning models with a prior information on the distribution of frequencies of points. Equivalently, it is an average-case model in which the frequencies of points are chosen randomly according to some distribution (instead of the worst-case or fixed choice in the standard models). We will model labels in the standard way, although for consistency of notation it will also be convenient to think of labeling function as being chosen randomly from some distribution $\mathcal{F}$ over functions. Importantly, a learning algorithm does not need to be aware of the frequency prior or explicitly incorporate it to achieve optimal (or high) accuracy. However, the prior will determine whether it is necessary to fit the dataset and any meaningful analysis of the generalization error in this setting needs to be aware of the prior (a more detailed comparison of our analysis with standard approaches can be found in Sec. 2.4).

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). 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 $\pi = (\pi_1, \ldots, \pi_N)$. A marginal distribution over X is produced by picking the frequency of each point in the domain randomly and independently from the prior π of individual frequencies subject to having to sum up to 1. We note that this process is almost identical to knowing the frequencies of the elements up to a random permutation, which is the assumption underlying recent breakthroughs in the analysis of density estimation algorithms [OS15; VV16].

A labeling function is chosen randomly from some distribution $\mathcal{F}$. As usual we assume that $\mathcal{F}$ (which corresponds to the hypothesis class) is known to the learning algorithm but we will be primarily interested in the setting where $\mathcal{F}$ is rich (or uncertain) enough that achieving close to 0 generalization error requires significantly more than the available number of samples. The goal of the learning algorithm is to predict as accurately as possible on a dataset sampled i.i.d. from a data distribution randomly chosen according the

process described above.

Our main result (Thm. 2.3) directly relates the number of points that an algorithm does not fit to the sub-optimality (or excess error[2]) of the algorithm on our problem via a quantity that depends only on the frequency prior π and n. Or, equivalently, it predicts the gain in the generalization error for every additional example memorized by the algorithm. Formally, we denote by $\mathtt{errn}_S(\mathcal{A}, 1)$ 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:

$$\overline{\mathtt{err}}(\pi, \mathcal{F}, \mathcal{A}) \geq \mathtt{opt}(\pi, \mathcal{F}) + \tau_1 \cdot \mathbf{E}\left[\mathtt{errn}_S(\mathcal{A}, 1)\right]. \tag{1}$$

Here $\overline{\mathtt{err}}(\pi, \mathcal{F}, \mathcal{A})$ refers to the expected generalization error of $\mathcal{A}$ for the given prior distributions and $\mathtt{opt}(\pi, \mathcal{F})$ is 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

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

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 need to know π and therefore the comparison is fair.

The quantity τ_1 is easy to compute given π. For 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 gain of at least $\approx 0.23/n$ per *every* example the learner memorizes (since it appears only once). For comparison, the worst-case gain 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 memorize 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 (see Sec. 2.3 for more details).

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. Therefore the discussion of interpolation (or memorization) makes sense only if one assumes that the prediction on the data point in the dataset will affect the predictions on related points. To bridge this gap, we consider mixture models of disjoint subpopulations as discussed above. 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. Further, we assume that (with high probability) the 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 (Defn. 3.1) 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 nearest neighbor. We also show that it applies to linear predictors/SVMs in high dimension provided that distinct subpopulations are sufficiently uncorrelated (see Sec. 3.1). 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.* [CSG18]). This supports the possibility that they exhibit this type of coupling as well.

[2]Here we are measuring excess error relative to the optimal algorithm and not relative to the best model in some class.

Finally, we use our techniques to give lower bounds on the excess error of learning algorithms that cannot memorize labels well. For this purpose we give a simple definition of what memorizing a label means (we are not aware of a prior formal definition of this notion). Our definition (see Def. 4.1) is closely related to the classical leave-one-out notions stability but focuses on the change in the label and not the incurred loss. Using this definition, we spell out several corollaries of our main result. Roughly, we show that an algorithm that cannot memorize labels well will be unable to fit the singleton data points whenever there is some uncertainty in their labels (conditioned on the rest of the dataset). Thus such algorithms will not be able to reach close-to-optimal generalization error. We show similar corollaries for the differentially private algorithms discussed below.

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 [Aba+16; Pap+16; Pap+17; McM+18]. 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 (see Sec.4.2 for more details on why). Moreover, they result in remarkably low gap between the training and test error that is formally explained by the generalization properties of DP [Dwo+14]. 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. [Pap+17] 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.

Some of the motivation for this work comes from attempts to understand why do DP algorithms fall short of their non-private counterparts and which examples are they likely to misclassify. A thorough and recent exploration of this question can be found in the work of Carlini et al. [CEP18]. 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 accuracy of a model produced by a DP training algorithm on the example. As argued in that work and is clear from their comprehensive visualization, the examples on which a DP model errs are either outliers/mislabeled or correctly labeled but infrequent ones. In addition, the metric based on DP accuracy is well correlated with other metrics of being prototypical such as relative confidence of the (non-private) model. 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 fit the tail of the mixture distribution.

Another empirical work that provides indirect support for our theory is [Arp+17]. 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 moderately reduce the ability of the network to fit random labels without significantly impacting their performance on true data. The explanation proposed for this finding is that memorization is not necessary for learning which appears to contradict our theory. However, a closer look at the result suggests the opposite conclusion. On the real data 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 the discussion section) that fitting random examples or even interpolation are not necessary conditions for the application of our approach and for memorization being beneficial.

1.2 Related work

The analyses of the generalization error of interpolating methods date back to the classical work of [CH+67] on the nearest neighbor algorithm. At the same they have received relatively little attention until the recent wave of interest motivated by DNNs. This interest has lead to new analyses of existing interpolating methods as well new algorithmic techniques [BRT18; BHM18; LR18; BMM18; RZ18].

Some works have explored the possibility that interpolation is the result of computational benefits of optimization via SGD (*e.g.* [MBB18]). Given that methods like nearest neighbors, boosting and random forests tend to interpolate as well we would conjecture that this is not the primary reason for the ubiquity of interpolating solutions (but certainly a useful side-benefit).

Algorithmic stability [BE02; Sha+09; HRS16; FV19] is essentially the only general approach that is known to imply generalization bounds beyond those achievable via uniform convergence [Sha+09; Fel16]. 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 have better stability on average).

Recent work of Belkin et al. [Bel+18] shows empirically and also on simple synthetic models [BHX19] 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 stil 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 Memorization in Unstructured Classification

Preliminaries: For a natural number n, we use $[n]$ to denote the set $\{1, \ldots, n\}$. For a condition O we use $\mathbf{1}(O)$ to denote the $\{0,1\}$-indicator function of the condition. A dataset is specified by an ordered n-tuple of examples $S = (x_1, y_1), \ldots, (x_n, y_n)$ but we will also treat it as the multi-set of examples it includes. Let X_S denote the set of all points that appear in S.

For a probability distribution D over X, $x \sim D$ denotes choosing x by sampling it randomly from D. For any condition $O \subseteq X$ and function F over X, we denote by $\mathbf{D}_{x \sim D}[F(x) \mid x \in O]$ the probability distribution of $F(x)$, when $x \sim D$ and is conditioned on $x \in O$. For two probability distributions D_1, D_2 over the same domain we use $\mathtt{TV}(D_1, D_2)$ to denote the total variation distance between them.

The goal of the learning algorithm is to predict the labels given a dataset $S = (x_1, y_1), \ldots (x_n, y_n)$ consisting of i.i.d. samples from some unknown distribution P over $X \times Y$. For any function $h \colon X \to Y$ and distribution P over $X \times Y$, we denote $\mathtt{err}_P(h) := \mathbf{E}_{(x,y)\sim P}[h(x) \neq y]$. As usual, for a randomized learning algorithm $\mathcal{A}$ we denote its expected generalization error on dataset S by

$$\mathtt{err}_P(\mathcal{A}, S) := \mathop{\mathbf{E}}_{h \sim \mathcal{A}(S)} \left[\mathtt{err}_P(h)\right],$$

where $h \sim \mathcal{A}(S)$ refers to h being the output of a (possibly) randomized algorithm. We also denote by $\mathtt{err}_P(\mathcal{A}) := \mathbf{E}_{S \sim P^n}\left[\mathtt{err}_P(\mathcal{A}, S)\right]$ the expectation of the generalization error of $\mathcal{A}$ when examples are drawn randomly from P.

2.1 Problem setup

To capture the main phenomenon we are interested in we start by considering a simple and general prediction problem in which the domain does not have any underlying structure (such as the notion of distance). The domains X and Y are discrete, $|X| = N$ and $|Y| = m$ (for concreteness one can think of $X = [N]$ and $Y = [m]$).

The prior information about the labels is encoded using a distribution $\mathcal{F}$ over functions from X to Y. The key assumption is that nothing is known a priori about the frequency of any individual point aside from a prior distribution over the individual frequencies. One natural approach to capturing this assumption is to assume that the frequencies of the elements in X are known up to a permutation. That is, a distribution over X is defined by picking a random permutation of elements of the prior $\pi = (\pi_1, \ldots, \pi_N)$. This modeling approach was used in several recent breakthrough works on density estimation [OS15; VV16]. In our case it does not appear to lead to a sufficiently clear and general connection between memorization and generalization. Exact knowledge of the entire frequency prior is also an unnecessarily strong of an assumption in most learning problems. We therefore use a related but different way to model the frequencies (which we have not encountered in prior work). In our model the frequency of each point in X is chosen randomly an independently from the list of possible frequencies π subject to having to sum up to 1.

More formally, let $\mathcal{D}_\pi^X$ denote the distribution over probability mass functions on X defined as follows. For every $x \in X$ sample p_x randomly, independently and uniformly from the elements of π. Define the corresponding probability mass function on X as $D(x) = \frac{p_x}{\sum_{x \in X} p_x}$. This definition can be naturally generalized to sampling from a general distribution π over frequencies (instead of just the uniform over a list of frequencies). We also denote by $\bar{\pi}^N$ the resulting marginal distribution over the frequency of any single element in x. That is,

$$\bar{\pi}^N(\alpha) := \mathop{\mathbf{Pr}}_{D \sim \mathcal{D}_\pi^X}[D(x) = \alpha].$$

Note that, while π is used to define the process, the actual distribution over individual frequencies the process results in is $\bar{\pi}^N$ and our bounds will be stated in terms of properties of $\bar{\pi}^N$. At the same time, this distinction is not particularly significant for applications of our result since, as we will show later, $\bar{\pi}^N$ is essentially a slightly smoothed version of π.

The key property of this way to generate the frequency distribution is that it allows us to easily express the expected frequency of a sample conditioned on observing it in the dataset. Specifically, in Appendix A we prove the following lemma:

Lemma 2.1. *For any frequency prior π, $x \in X$ and a sequence of points $V = (x_1, \ldots, x_n) \in X^n$ that includes x exactly ℓ times, we have*

$$\mathop{\mathbf{E}}_{D \sim \mathcal{D}_\pi^X, U \sim D^n}[D(x) \mid U = V] = \frac{\mathbf{E}_{\alpha \sim \bar{\pi}^N}\left[\alpha^{\ell+1} \cdot (1-\alpha)^{n-\ell}\right]}{\mathbf{E}_{\alpha \sim \bar{\pi}^N}\left[\alpha^{\ell} \cdot (1-\alpha)^{n-\ell}\right]}.$$

An instance of our learning problem is generated by picking a marginal distribution D randomly from $\mathcal{D}_\pi^X$ and picking the true labeling function randomly according to $\mathcal{F}$. We refer to the distribution of the labeled examples $(x, f(x))$ for $x \sim D$ by (D, f). We abbreviate $\mathcal{D}_\pi^X$ as $\mathcal{D}$ whenever the prior and X are clear from the context.

We are interested in evaluating the generalization error of a classification algorithm on instances of our learning problem. Our results apply (via a simple adaption) to the more common setup in statistical learning theory where $\mathcal{F}$ is a set of functions and worst case error with respect to a choice of $f \in \mathcal{F}$ is considered.

However for simplicity of notation and consistency with the random choice of D, we focus on the expectation of the generalization error on a randomly chosen learning problem:

$$\overline{\mathtt{err}}(\pi, \mathcal{F}, \mathcal{A}) := \mathop{\mathbf{E}}_{D\sim\mathcal{D}, f\sim\mathcal{F}}[\mathtt{err}_{D,f}(\mathcal{A})]\,.$$

2.2 The cost of not fitting

Ability of an algorithm $\mathcal{A}$ to memorize the labels of points in the dataset is closely related to how well it can fit the dataset, or equivalently, to the lowest empirical error $\mathcal{A}$ can achieve. We will now demonstrate that for our simple problem there exists a precise relationship between how well an algorithm fits the labels of the points it observed and the excess generalization error of the algorithm. This relationship will be determined by the prior $\bar{\pi}^N$ and n. Importantly, this relationship will hold even when optimal achievable generalization error is high, a regime not covered by the usual analysis in the "realizable" setting.

In our results the effect of not fitting an example depends on the number of times it occurs in the dataset and therefore we count examples that $\mathcal{A}$ does not fit separately for each possible multiplicity. More formally,

Definition 2.2. *For a dataset $S \in (X \times Y)^n$ and $\ell \in [n]$, let $X_{S=\ell}$ denote the set of points x that appear exactly ℓ times in S. For a function $h\colon X \to Y$ let*

$$\mathtt{errn}_S(h, \ell) := |\{x \in X_{S=\ell} \mid h(x) \neq y\}|$$

and let

$$\mathtt{errn}_S(\mathcal{A}, \ell) := \mathop{\mathbf{E}}_{h\sim\mathcal{A}(S)}[\mathtt{errn}_S(h, \ell)].$$

It is not hard to see (and we show this below) that in this noiseless setting the optimal expected generalization error is achieved by memorizing the dataset. Namely, by the algorithm that outputs the function that on the points in the dataset predicts the observed label and on points outside the dataset predicts the most likely label according to the posterior distribution on $\mathcal{F}$. We will now quantify the excess error of any algorithm that does not fit the labels of all the observed data points. Our result holds for every single dataset (and not just in expectation). To make this formal, we define $\mathcal{G}$ to be the probability distribution over triplets (D, f, S) where $D \sim \mathcal{D}_\pi^X$, $f \sim \mathcal{F}$ and $S \sim (D, f)^n$. For any dataset $Z \in (X \times Y)^n$, let $\mathcal{G}(|Z)$ denote the marginal distribution over distribution-function pairs conditioned on $S = Z$. That is:

$$\mathcal{G}(|Z) := \mathop{\mathbf{D}}_{(D,f,S)\sim\mathcal{G}}[(D, f) \mid S = Z].$$

We then define the expected error of $\mathcal{A}$ conditioned on dataset being equal to Z as

$$\overline{\mathtt{err}}(\pi, \mathcal{F}, \mathcal{A} \mid Z) := \mathop{\mathbf{E}}_{(D,f)\sim\mathcal{G}(|Z)}[\mathtt{err}_{D,f}(\mathcal{A}, Z)]\,.$$

We will also define $\mathtt{opt}(\pi, \mathcal{F} \mid Z)$ to be the minimum of $\overline{\mathtt{err}}(\pi, \mathcal{F}, \mathcal{A}' \mid Z)$ over all algorithms $\mathcal{A}'$.

Theorem 2.3. *Let π be a frequency prior with a corresponding marginal frequency distribution $\bar{\pi}^N$, and $\mathcal{F}$ be a distribution over Y^X. Then for every learning algorithm $\mathcal{A}$ and every dataset $Z \in (X \times Y)^n$:*

$$\overline{\mathtt{err}}(\pi, \mathcal{F}, \mathcal{A} \mid Z) \geq \mathtt{opt}(\pi, \mathcal{F} \mid Z) + \sum_{\ell\in[n]} \tau_\ell \cdot \mathtt{errn}_Z(\mathcal{A}, \ell),$$

where

$$\tau_\ell := \frac{\mathbf{E}_{\alpha\sim\bar{\pi}^N}\left[\alpha^{\ell+1}\cdot(1-\alpha)^{n-\ell}\right]}{\mathbf{E}_{\alpha\sim\bar{\pi}^N}\left[\alpha^{\ell}\cdot(1-\alpha)^{n-\ell}\right]}.$$

In particular,

$$\overline{\texttt{err}}(\pi,\mathcal{F},\mathcal{A}) \geq \texttt{opt}(\pi,\mathcal{F}) + \mathop{\mathbf{E}}_{D\sim\mathcal{D}_\pi^X, f\sim\mathcal{F}, S\sim(D,f)^n}\left[\sum_{\ell\in[n]}\tau_\ell\cdot\texttt{errn}_S(\mathcal{A},\ell)\right].$$

Proof. We denote the marginal distribution of $\mathcal{G}(|Z)$ over D by $\mathcal{D}(|Z)$ and the marginal distribution over f by $\mathcal{F}(|Z)$. We begin by noting that for every $f'\colon X\to Y$ consistent with the examples in Z, the distribution of D conditioned on $f=f'$ is still $\mathcal{D}(|Z)$, since D is chosen independently of any labeling. Therefore we can conclude that $\mathcal{G}(|Z)$ is equal to the product distribution $\mathcal{D}(|Z)\times\mathcal{F}(|Z)$.

Let $X_{Z>0} := X\setminus X_{Z=0}$ denote the set of points in X that appear in one of the examples in Z. To prove the claim we will prove that

$$\overline{\texttt{err}}(\pi,\mathcal{F},\mathcal{A}\mid Z) = \sum_{\ell\in[n]}\tau_\ell\cdot\texttt{errn}_Z(\mathcal{A},\ell) + \sum_{x\in X_{Z=0}}\mathop{\mathbf{E}}_{h\sim\mathcal{A}(Z),f\sim\mathcal{F}(|Z)}[h(x)\neq f(x)]\cdot p(x,Z), \tag{2}$$

where $p(x,Z) := \mathbf{E}_{D\sim\mathcal{D}(|Z)}[D(x)]$. This will imply the claim since the bottom expression is minimized when for all $\ell\in[n]$, $\texttt{errn}_Z(\mathcal{A},\ell)=0$ and for all $x\in X_{Z=0}$,

$$\mathop{\mathbf{E}}_{h\sim\mathcal{A}(Z),f\sim\mathcal{F}(|Z)}[h(x)\neq f(x)] = \min_{y\in Y}\mathop{\mathbf{E}}_{f\sim\mathcal{F}(|Z)}[f(x)\neq y].$$

Moreover, this minimum is achieved by the algorithm $\mathcal{A}^*$ that fits the examples in Z and predicts the label y that minimizes $\mathbf{E}_{f\sim\mathcal{F}(|Z)}[f(x)\neq y]$ on all the points in $X_{Z=0}$. Namely,

$$\begin{aligned}\sum_{x\in X_{Z=0}}\mathop{\mathbf{E}}_{h\sim\mathcal{A}(Z),f\sim\mathcal{F}(|Z)}[h(x)\neq f(x)]\cdot p(x,Z) &\geq \sum_{x\in X_{Z=0}}\min_{y\in Y}\mathop{\mathbf{E}}_{f\sim\mathcal{F}(|Z)}[f(x)\neq y]\cdot p(x,Z)\\ &= \overline{\texttt{err}}(\pi,\mathcal{F},\mathcal{A}^*\mid Z) = \texttt{opt}(\pi,\mathcal{F}\mid Z).\end{aligned}$$

Plugging this into eq. (2) gives the first claim.

We now prove eq. (2).

$$\begin{aligned}\overline{\texttt{err}}(\pi,\mathcal{F},\mathcal{A}\mid Z) &= \mathop{\mathbf{E}}_{(D,f)\sim\mathcal{G}(|Z),h\sim\mathcal{A}(Z)}\left[\texttt{err}_{D,f}(h)\right]\\ &= \mathop{\mathbf{E}}_{(D,f)\sim\mathcal{G}(|Z),h\sim\mathcal{A}(Z)}\left[\sum_{x\in X}\mathbf{1}\left(h(x)\neq f(x)\right)\cdot D(x)\right]\\ &= \sum_{x\in X_{Z>0}}\mathop{\mathbf{E}}_{(D,f)\sim\mathcal{G}(|Z),h\sim\mathcal{A}(Z)}\left[\mathbf{1}\left(h(x)\neq f(x)\right)\cdot D(x)\right] \qquad (3)\\ &+ \sum_{x\in X_{Z=0}}\mathop{\mathbf{E}}_{(D,f)\sim\mathcal{G}(|Z),h\sim\mathcal{A}(Z)}\left[\mathbf{1}\left(h(x)\neq f(x)\right)\cdot D(x)\right]. \qquad (4)\end{aligned}$$

Using the fact that $\mathcal{G}(|Z)=\mathcal{D}(|Z)\times\mathcal{F}(|Z)$, for every $x\in X_{Z=0}$ we get

$$\begin{aligned}\mathop{\mathbf{E}}_{(D,f)\sim\mathcal{G}(|Z),h\sim\mathcal{A}(Z)}\left[\mathbf{1}\left(h(x)\neq f(x)\right)\cdot D(x)\right] &= \mathop{\mathbf{Pr}}_{h\sim\mathcal{A}(Z),f\sim\mathcal{F}(|Z)}[h(x)\neq f(x)]\cdot\mathop{\mathbf{E}}_{D\sim\mathcal{D}(|Z)}[D(x)]\\ &= \mathop{\mathbf{Pr}}_{h\sim\mathcal{A}(Z),f\sim\mathcal{F}(|Z)}[h(x)\neq f(x)]\cdot p(x,Z).\end{aligned}$$

Hence we obtain that the term in line (4) is exactly equal to the second term on the right hand side of eq. (2).

For the term in line (3), we pick an arbitrary $x \in X_{Z=\ell}$ for some $\ell \in [n]$. We can decompose

$$\mathop{\mathbf{E}}_{(D,f)\sim\mathcal{G}(|Z),h\sim\mathcal{A}(Z)}[\mathbf{1}\left(h(x) \neq f(x)\right) \cdot D(x)] = \mathop{\mathbf{Pr}}_{h\sim\mathcal{A}(Z)}[h(x) \neq f(x)] \cdot \mathop{\mathbf{E}}_{D\sim\mathcal{D}(|Z)}[D(x)]$$

since additional conditioning on $h(x) \neq f(x)$ does not affect the distribution of $D(x)$ (as mentioned, $\mathcal{G}(|Z)$ is a product distribution). Let V denote the sequence of points in the dataset Z. The labels of these points do not affect the conditioning of D and therefore by Lemma 2.1,

$$\mathop{\mathbf{E}}_{D\sim\mathcal{D}(|Z)}[D(x)] = \mathop{\mathbf{E}}_{D\sim\mathcal{D},U\sim D^n}[D(x) \mid U = V] = \frac{\mathbf{E}_{\alpha\sim\bar{\pi}^N}\left[\alpha^{\ell+1} \cdot (1-\alpha)^{n-\ell}\right]}{\mathbf{E}_{\alpha\sim\bar{\pi}^N}\left[\alpha^{\ell} \cdot (1-\alpha)^{n-\ell}\right]} = \tau_\ell.$$

By combining these two equalities we obtain that, as claimed in eq.(2), line (3) is equal to

$$\begin{aligned}
(3) &= \sum_{x\in X_{Z>0}} \mathop{\mathbf{E}}_{(D,f)\sim\mathcal{G}(|Z),h\sim\mathcal{A}(Z)}[\mathbf{1}\left(h(x) \neq f(x)\right) \cdot D(x)] \\
&= \sum_{\ell\in[n]} \sum_{x\in X_{Z=\ell}} \tau_\ell \cdot \mathop{\mathbf{Pr}}_{h\sim\mathcal{A}(Z)}[h(x) \neq f(x)] \\
&= \sum_{\ell\in[n]} \tau_\ell \cdot \mathtt{errn}_Z(\mathcal{A},\ell).
\end{aligned}$$

To obtain the second part of the theorem we denote by $\mathcal{S}$ the marginal distribution of $\mathcal{G}$ over S. Observe that

$$\mathtt{opt}(\pi,\mathcal{F}) = \mathop{\mathbf{E}}_{Z\sim\mathcal{S}}[\mathtt{opt}(\pi,\mathcal{F} \mid Z)]$$

since the optimal algorithm is given Z as an input. The second claim now follows by taking the expectation over the marginal distribution over S:

$$\begin{aligned}
\overline{\mathtt{err}}(\pi,\mathcal{F},\mathcal{A}) &= \mathop{\mathbf{E}}_{Z\sim\mathcal{S}}[\overline{\mathtt{err}}(\pi,\mathcal{F},\mathcal{A} \mid Z)] \\
&\geq \mathop{\mathbf{E}}_{Z\sim\mathcal{S}}\left[\mathtt{opt}(\pi,\mathcal{F} \mid Z) + \sum_{\ell\in[n]} \tau_\ell \cdot \mathtt{errn}_Z(\mathcal{A},\ell)\right] \\
&= \mathtt{opt}(\pi,\mathcal{F}) + \sum_{\ell\in[n]} \tau_\ell \cdot \mathop{\mathbf{E}}_{Z\sim\mathcal{S}}\left[\mathtt{errn}_Z(\mathcal{A},\ell)\right].
\end{aligned}$$

□

2.3 From tails to bounds

Given a frequency prior π, Theorem 2.3 gives a general and easy way to compute the effect of not fitting an example in the dataset. We now spell out some simple and easier to interpret corollaries of this general result and show that the effect can be very significant. The primary case of interest is $\ell = 1$, namely examples that appear only once in S, which we refer to as *singleton* examples. In order to fit those, an algorithm needs memorize their labels (see Section 4.1 for a more detailed discussion). We first note that the expected number

of singleton examples is determined by the weight of the entire tail of $\bar{\pi}^N$. Specifically, the expected fraction of the distribution D contributed by frequencies in the range $[\beta_1, \beta_2]$ is defined as:

$$\begin{aligned}\texttt{weight}(\bar{\pi}^N, [\alpha, \beta]) &:= \mathop{\mathbf{E}}_{D\sim\mathcal{D}}\left[\sum_{x\in X} D(x)\cdot \mathbf{1}\left(D(x)\in[\beta_1,\beta_2]\right)\right] \\ &= N\cdot \mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}\left[\alpha\cdot\mathbf{1}\left(\alpha\in[\beta_1,\beta_2]\right)\right].\end{aligned}$$

At the same time the expected number of singleton points is:

$$\begin{aligned}\texttt{single}(\bar{\pi}^N) &:= \mathop{\mathbf{E}}_{D\sim\mathcal{D}, V\sim D^n}\left[|X_{V=1}|\right] = \mathop{\mathbf{E}}_{D\sim\mathcal{D}}\left[\sum_{x\in X}\mathop{\mathbf{Pr}}_{V\sim D^n}[x\in X_{V=1}]\right] \\ &= \mathop{\mathbf{E}}_{D\sim\mathcal{D}}\left[\sum_{x\in X} n\cdot D(x)(1-D(x))^{n-1}\right] \\ &= \sum_{x\in X} n\mathop{\mathbf{E}}_{D\sim\mathcal{D}}\left[D(x)(1-D(x))^{n-1}\right] \\ &= nN\cdot\mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}\left[\alpha(1-\alpha)^{n-1}\right].\end{aligned}$$

For every $\alpha \le 1/n$ we have that $(1-\alpha)^{n-1} \ge 1/3$ (for sufficiently large n). Therefore:

$$\texttt{single}(\bar{\pi}^N) \ge nN\cdot\mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}\left[\alpha(1-\alpha)^{n-1}\cdot\mathbf{1}\left(\alpha\in\left[0,\frac{1}{n}\right]\right)\right] \ge \frac{n}{3}\texttt{weight}\left(\bar{\pi}^N,\left[0,\frac{1}{n}\right]\right). \tag{5}$$

We will now show that the expected cost of not fitting any of the singleton examples is lower bounded by the weight contributed by frequencies on the order of $1/n$. Our bounds will be stated in terms of the properties of $\bar{\pi}^N$ (as opposed to π itself) and therefore, before proceeding, we briefly explain the relationship between these two.

Relationship between π and $\bar{\pi}^N$: Before the normalization step, for every $x \in X$, p_x is distributed exactly according to π (that is uniform over $(\pi_1, \ldots, \pi_N)$. Therefore, it is sufficient to understand the distribution of the normalization factor conditioned on $p_x = \pi_i$ for some i. Under this condition the normalization factor s_i is distributed as the sum of $n-1$ independent samples from π plus π_i. The mean of each sample is exactly $1/N$ and thus standard concentration results can be used to obtain that s_i is concentrated around $\frac{N-1}{N} + \pi_i$. Tightness of this concentration depends on the properties of π, most importantly, the largest value $\pi_{\max} := \max_{j\in[N]}\pi_j$ and $\mathbf{Var}[\pi] := \frac{1}{N}\sum_{j\in[N]}(\pi_j - \frac{1}{N})^2 \le \pi_{\max}$. For $\pi_{\max} = o(1)$, $\bar{\pi}^N$ can be effectively seen as convolving each π_i multiplicatively by a factor whose inverse is a Gaussian-like distribution of mean $1 - 1/N + \pi_i$ and variance $\mathbf{Var}(\pi)$. More formally, using Bernstein's (or Bennett's) concentration inequality (*e.g.* [Sri02]) we can easily relate the total weight in a certain range of frequencies under $\bar{\pi}^N$ to the weight in a similar range under π.

Lemma 2.4. *Let $\pi = (\pi_1, \ldots, \pi_N)$ be a frequency prior and $\bar{\pi}^N$ be the corresponding marginal distribution over frequencies. For any $0 < \beta_1 < \beta_2 < 1$ Then for and any $\gamma > 0$,*

$$\texttt{weight}(\bar{\pi}^N, [\beta_1, \beta_2]) \ge \frac{(1-\delta)}{1 - \frac{1}{N} + \beta_2 + \gamma}\cdot\texttt{weight}\left(\pi, \left[\frac{\beta_1}{1-\frac{1}{N}+\beta_1-\gamma}, \frac{\beta_2}{1-\frac{1}{N}+\beta_2+\gamma}\right]\right),$$

where $\pi_{\max} := \max_{j\in[N]} \pi_j$, $\mathbf{Var}[\pi] := \sum_{j\in[N]}(\pi_j - \frac{1}{N})^2$ *and* $\delta := 2 \cdot e^{\frac{-\gamma^2}{2(N-1)\mathbf{Var}(\pi)+2\gamma\pi_{\max}/3}}$.

Note that

$$\mathbf{Var}[\pi] \leq \frac{1}{N}\sum_{j\in[N]} \pi_j^2 \leq \frac{\pi_{\max}}{N} \cdot \sum_{j\in[N]} \pi_j = \frac{\pi_{\max}}{N}.$$

By taking $\gamma = 1/4$, we can ensure that the boundaries of the frequency interval change by a factor of at most (roughly) $4/3$. For such γ we will obtain $\delta \leq 2e^{-1/(40\pi_{\max})}$ and in particular $\pi_{\max} \leq 1/200$ will suffice for making the correction $(1-\delta)$ at least $99/100$ (which is insignificant for our purposes).

Bounds for $\ell = 1$: We now show a simple lower bound on τ_1 in terms of $\texttt{weight}(\bar{\pi}^N, [1/2n, 1/n])$ (similar results hold for other choices of the interval $[c_1/n, c_2/n]$). We also do not optimize the constants in the bounds as our goal is to demonstrate the qualitative behavior.

Lemma 2.5. *For every frequency prior π and sufficiently large n, N,*

$$\tau_1 \geq \frac{1}{5n} \cdot \texttt{weight}\left(\bar{\pi}^N, \left[\frac{1}{3n}, \frac{2}{n}\right]\right).$$

If, in addition, $\pi_{\max} \leq 1/200$, then

$$\tau_1 \geq \frac{1}{7n} \cdot \texttt{weight}\left(\pi, \left[\frac{1}{2n}, \frac{1}{n}\right]\right).$$

Proof. We first observe that the denominator of τ_1 satisfies

$$\mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}\left[\alpha(1-\alpha)^{n-1}\right] \leq \mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}[\alpha] = \frac{1}{N}.$$

Now, by simple calculus, for every $\alpha \in \left[\frac{1}{3n}, \frac{2}{n}\right]$ and sufficiently large n,

$$\alpha^2(1-\alpha)^{n-1} \geq \frac{1}{5n} \cdot \alpha.$$

Therefore

$$\begin{aligned}\tau_1 &= \frac{\mathbf{E}_{\alpha\sim\bar{\pi}^N}\left[\alpha^2(1-\alpha)^{n-1}\right]}{\mathbf{E}_{\alpha\sim\bar{\pi}^N}\left[\alpha(1-\alpha)^{n-1}\right]} \\ &\geq \frac{\frac{1}{5n}\cdot\mathbf{E}_{\alpha\sim\bar{\pi}^N}\left[\alpha\cdot\mathbf{1}\left(\alpha\in\left[\frac{1}{3n},\frac{2}{n}\right]\right)\right]}{\frac{1}{N}} = \frac{1}{5n}\cdot\texttt{weight}\left(\bar{\pi}^N,\left[\frac{1}{3n},\frac{2}{n}\right]\right).\end{aligned}$$

To obtain the second part of the claim we apply Lemma 2.4 for $\gamma = 1/4$ (as discussed above). To verify, observe that for sufficiently large n and N, $\frac{\frac{1}{3n}}{1-\frac{1}{N}+\frac{1}{3n}-1/4} \leq \frac{1}{2n}$ and $\frac{\frac{2}{n}}{1-\frac{1}{N}+\frac{2}{n}+1/4} \geq \frac{1}{n}$, and $\frac{(1-\delta)}{1-\frac{1}{N}+\frac{2}{n}+\gamma} \geq \frac{3}{4}$. □

The value of $\tau_1 = \Omega(1/n)$ corresponds to paying on the order of $1/n$ in generalization error for every example that is not fit by the algorithm. Hence if the total weight of frequencies in the range of $1/n$ is at least some θ then the algorithm that does not fit them will be suboptimal by θ times the fraction of such examples

in the dataset. By eq. (5), the fraction of such examples themselves is determined by the weight of the entire tail $\texttt{weight}(\bar{\pi}^N, [0, 1/n])$.

We can contrast this situation with the case where there are no frequencies that are on the order of $1/n$. Even when the data distribution has no elements with such frequency, the total weight of the frequencies in the tail and as a result the fraction of singleton points might be large. Still as we show in such case the cost of not fitting such examples will be negligible.

Lemma 2.6. *Let π be a frequency prior such that for some $\theta \leq \frac{1}{2n}$, $\texttt{weight}\left(\bar{\pi}^N, \left[\theta, \frac{t}{n}\right]\right) = 0$, where $t = \ln(1/(\theta\beta)) + 2$ for $\beta := \texttt{weight}\left(\bar{\pi}^N, [0, \theta]\right)$. Then $\tau_1 \leq 2\theta$.*

Proof. We first observe that the numerator of τ_1 is at most:

$$\begin{aligned}\mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}\left[\alpha^2(1-\alpha)^{n-1}\right] \leq \max_{\alpha\in[t/n,1]} \alpha^2(1-\alpha)^{n-1} \cdot \mathop{\mathbf{Pr}}_{\alpha\sim\bar{\pi}^N}\left[\alpha \geq \frac{t}{n}\right] \\ + \mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}\left[\alpha^2(1-\alpha)^{n-1} \cdot \mathbf{1}\left(\alpha \leq \theta\right)\right].\end{aligned}$$

By Markov's inequality, $\mathbf{E}_{\alpha\sim\bar{\pi}^N}[\alpha] = \frac{1}{N}$ implies

$$\mathop{\mathbf{Pr}}_{\alpha\sim\bar{\pi}^N}\left[\alpha \geq \frac{t}{n}\right] \leq \frac{n}{tN}.$$

In addition, by our definition of t,

$$\max_{\alpha\in[t/n,1]} \alpha^2(1-\alpha)^{n-1} \leq \frac{t}{n}\left(1-\frac{t}{n}\right)^{n-1} \leq \frac{t\beta\theta}{en}.$$

Therefore the first term in the numerator is upper bounded by $\frac{n}{tN}\frac{t\beta\theta}{en} \leq \frac{\beta\theta}{eN}$. At the same time the second term in the numerator satisfies:

$$\begin{aligned}\mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}\left[\alpha^2(1-\alpha)^{n-1} \cdot \mathbf{1}\left(\alpha \leq \theta\right)\right] &\geq \theta(1-\theta)^{n-1} \cdot \mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}\left[\alpha \cdot \mathbf{1}\left(\alpha \leq \theta\right)\right] \\ &\geq \theta\left(1-\frac{1}{2n}\right)^{n-1} \cdot \frac{\texttt{weight}\left(\bar{\pi}^N, [0,\theta]\right)}{N} \geq \frac{\theta\beta}{2N}.\end{aligned}$$

Therefore the second term is at least as large as the first term and we obtain that:

$$\begin{aligned}\mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}\left[\alpha^2(1-\alpha)^{n-1}\right] &\leq 2 \cdot \mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}\left[\alpha^2(1-\alpha)^{n-1} \cdot \mathbf{1}\left(\alpha \leq \theta\right)\right] \\ &\leq 2\theta \cdot \mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}\left[\alpha(1-\alpha)^{n-1} \cdot \mathbf{1}\left(\alpha \leq \theta\right)\right] \\ &\leq 2\theta \cdot \mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}\left[\alpha(1-\alpha)^{n-1}\right].\end{aligned}$$

Thus $\tau_1 \leq 2\theta$ as claimed. □

For $\theta = 1/(2n^2)$, under the conditions of Lemma 2.6 we will obtain that the suboptimality of the algorithm that does not fit any of the singleton examples is at most $1/n$.

2.4 Comparison with standard approaches to generalization

We now briefly demonstrate that standard approaches for analysis of generalization error cannot be used to derive the conclusions of this section and do not capture our simple problem whenever $N \geq n$. For concreteness, we will use $m = 2$ with a uniform prior over all labelings. Without any structure to labels a natural class of algorithms for the problem are algorithm that pick a subset of points whose labels are memorized and predict randomly the other points in the domain.

First of all, it is clear that any approach that does not make any assumption on the marginal distribution D cannot adequately capture the generalization error of such algorithms. A distribution independent generalization bound needs to apply to the uniform distribution over X. For this distribution the expected generalization error for a randomly chosen labeling function f will be at least $(1 - n/N)/2$. In particular, no non-trivial bound on generalization error will be possible for $N \gg n$. For sufficiently large N the differences in the generalization error of different algorithms will be negligible and therefore such notion will not be useful for guiding the choice of the algorithm.

Notions that are based on the algorithm knowing the input distribution D are not applicable to our setting. Indeed the main difficulty is that the algorithm does not know the exact frequencies of the singleton elements. An algorithm that knows D would not need to fit the points whose frequency is less than say $1/n^2$. Thus the algorithm would be able to achieve excess error of at most $1/n$ without fitting the dataset which is inconsistent with the conclusions in our model. As an extreme example, consider the case when the prior contains $n/2$ points of frequency $1/n$ and $n^2/2$ points of frequency $1/n^2$. In this case, the algorithm that knows D can fit only 50% of the points in the dataset and achieve excess generalization error of around $1/(2n)$. In contrast, our analysis shows that an algorithm that only knows the prior and fits only 50% of the dataset will be suboptimal by $> 13\%$.

Fairly tight data-dependent bounds on the generalization error can be obtained via the notion of empirical Rademacher complexity [Kol10; BM02]. Empirical Rademacher complexity for a dataset S and the class of all Boolean functions on X that memorize k points is $\geq \min\{k, |X_S|\}/n$. Similar bound can also be obtained via weak notions of stability such as average leave-one-out stability [BE02; RMP05; Muk+06; Sha+10]

$$\texttt{LOOstab}(P, \mathcal{A}) := \frac{1}{n} \sum_{i \in [n]} \mathop{\mathbf{E}}_{S \sim P^n} \left[\left| \mathop{\mathbf{Pr}}_{h \sim \mathcal{A}(S)} [h(x_i) = y_i] - \mathop{\mathbf{Pr}}_{h \sim \mathcal{A}(S^{\setminus i})} [h(x_i) = y_i] \right| \right], \tag{6}$$

where $S^{\setminus i}$ refers to S with i-th example removed. If we were to use either of these notions to pick k (the number of points to memorize), we would end up not fitting any of the singleton points. The simple reason for this is that, just like a learning algorithm cannot distinguish between "outlier" and "borderline" points given S in this setting, neither will any bound. Therefore any true upper bound on the generalization error that is not aware of the prior on the frequencies needs to be correct when all the points that occur once are "outliers". Fitting any of the outliers does not improve the generalization error at all and therefore such upper bounds on the generalization error cannot be used to correctly guide the choice of k.

3 Prediction in Mixture Models

Our problem setting in Section 2 considers discrete domains without any structure on X. The results also focus on elements of the domain whose frequency is on the order of $1/n$. Naturally, practical prediction problems are high-dimensional with each individual point having exponentially small (in the dimension) probability. Therefore direct application of our analysis from Section 2 for the unstructured case makes little

sense. Indeed, any learning algorithm $\mathcal{A}$ can be modified to a learning algorithm $\mathcal{A}'$ that does not fit any of the points in the dataset and achieves basically the same generalization error as $\mathcal{A}$ simply by modifying $\mathcal{A}$'s predictions on the training data to different labels and vice versa (any algorithm can be made to fit the dataset without any effect on its generalization).

At the same time in high dimensional settings the points have additional structure that can be exploited by a learning algorithm. Most machine learning algorithms are very likely to produce the same prediction on points that are sufficiently "close" in some representation. The representation itself may be designed based on domain knowledge or data-derived (in a not too sensitive way). This is clearly true about k-NN, SVMs/linear predictors and has been empirically observed for neural networks once the trained representation in the last hidden layer is considered.

The second important aspect of natural image and text data is that it is a mixture of numerous subpopulations (as evidenced by the popularity of mixture models). The relative frequency of these subpopulations is known to have a long-tailed distribution, especially in more natural datasets such as iNaturalist [VH+18] that are not artificially balanced for sub-categories [ZAR14; WRH17; VHP17; Cui+18]. A natural way to think of and a common way to model subpopulations (or mixture components) is as consisting of points that are similar to each other yet sufficiently different from other points in the domain.

We capture the essence of these two properties using the following model that applies the ideas we developed in Section 2 to mixture models. To keep the main points clear we keep the model relatively simple by making relatively strong assumptions on the structure. (we will later briefly discuss several ways in which the model's assumptions can be relaxed or generalized).

We model the unlabeled data distribution as a mixture of a large number of fixed distribution $M_1, \ldots, M_N$. For simplicity, we assume that these distributions have disjoint support, namely M_i is supported over X_i and $X_i \cap X_j = \emptyset$ for $i \neq j$ (without loss of generality $X = \cup_{i \in [N]} X_i$). For $x \in X$ we denote i_x to be the index of the sub-domain of x and by X_x (or M_x) the sub-domain (or subpopulation, respectively) itself.

The unknown distribution $M(x) \equiv \sum_{i \in [N]} \alpha_i M_i(x)$ for some vector of mixture coefficients $(\alpha_1, \ldots, \alpha_N)$ that sums up to 1. We describe it as a distribution $D(x)$ over $[N]$ (that is $\alpha_i = D(i)$). As in our unstructured model, we assume that nothing is known a priori about the mixture coefficients aside from (possibly) a prior $\pi = (\pi_1, \ldots, \pi_N)$ described by a list of frequencies. The mixture coefficients are generated, as before, by sampling D from $\mathcal{D}_\pi^{[N]}$. We denote by M_D the distribution over X defined as $M_D(x) \equiv \sum_{i \in [N]} D(i) M_i(x)$.

We assume that the entire subpopulation X_i is labeled by the same label and the label prior is captured via an arbitrary distribution $\mathcal{F}$ over functions from $[N]$ to Y. Note that such prior can be used to reflect typical case where a subpopulation that is similarly "close" to subpopulations i_1 and i_2 is likely to have the same label as i_1 or i_2.The labeling function L for the entire domain X is sampled by first sampling $f \sim \mathcal{F}$ and defining $L_f(x) = f(i_x)$.

To model the properties of the learning algorithm we assume that for every point x in a dataset S the distribution over predictions $h(x)$ for a random predictor output by $\mathcal{A}(S)$ is close to (or at least not too different) from the distribution over predictions that $\mathcal{A}$ produces over the entire subpopulation of x. This follows the intuition that labeling x will have a measurable effect on the prediction over the entire subpopulation. This effect may depend on the number of other points from the same subpopulation and therefore our assumption will be parameterized by ℓ parameters.

Definition 3.1. *Let X be a domain partitioned into sub-domains $\{X_i\}_{i \in [N]}$ with subpopulations $\{M_i\}_{i \in [N]}$ over the sub-domains. For a dataset S, let $X_{S=\ell}$ denote the union of subpopulations X_i such that points from X_i appear exactly ℓ times in S. For $\Lambda = (\lambda_1, \ldots, \lambda_n)$, we say that an algorithm $\mathcal{A}$ is* Λ-subpopulation-

coupled *if for every* $S \in (X \times Y)^n$, $x \in X_{S=\ell}$,

$$\mathtt{TV}\left(\mathop{\mathbf{D}}_{h\sim\mathcal{A}(S)}[h(x)], \mathop{\mathbf{D}}_{x'\sim M_x, h\sim\mathcal{A}(S)}[h(x')]\right) \leq 1-\lambda_\ell.$$

Note that we do not restrict the algorithm to be coupled in this sense over subpopulations that are not represented in the data. This distinction is important since predictors output by most natural algorithms vary over regions from which no examples were observed. As a result the setting here cannot be derived by simply collapsing points in the sub-domain into a single point and applying the results from the unstructured case. However, the analysis and the results in Sec. 2 still apply essentially verbatim to this more general setup. All we need is to extend the definition of $\mathtt{errn}_S(\mathcal{A},\ell)$ to look at the multiplicity of sub-domains and not points themselves and count mistakes just once per sub-domain. For a function $h\colon X \to Y$ let

$$\mathtt{errn}_S(h,\ell) = \frac{1}{\ell}\sum_{i\in[n]} \mathbf{1}\left(x_i \in X_{S=\ell} \text{ and } h(x_i) \neq y_i\right).$$

As before, $\mathtt{errn}_S(\mathcal{A},\ell) = \mathbf{E}_{h\sim\mathcal{A}(S)}[\mathtt{errn}_S(h,\ell)]$. With this definition we get the following generalization of Theorem 2 (we only state the version for the total expectation of the error but the per-dataset version holds as well):

Theorem 3.2. *Let $\{M_i\}_{i\in[N]}$ be subpopulations over sub-domains $\{X_i\}_{i\in[N]}$ and let π and $\mathcal{F}$ be some frequency and label priors. Then for every Λ-subpopulation-coupled learning algorithm $\mathcal{A}$:*

$$\overline{\mathtt{err}}(\pi,\mathcal{F},\mathcal{A}) \geq \mathtt{opt}(\pi,\mathcal{F}) + \mathop{\mathbf{E}}_{D\sim\mathcal{D}_\pi^{[N]}, f\sim\mathcal{F}, S\sim(M_D,L_f)^n}\left[\sum_{\ell\in[n]} \lambda_\ell\tau_\ell \cdot \mathtt{errn}_S(\mathcal{A},\ell)\right],$$

where τ_ℓ is defined in Thm. 2.3.

We now briefly discuss how the modeling assumptions can be relaxed. We first note that it suffices for subpopulation coupling to hold with high probability over the choice of dataset S from the marginal distribution over the datasets $\mathcal{S}$. Namely, if the property in Definition 3.1 holds with probability $1-\delta$ over the choice of $S \sim \mathcal{S}$ (where, $\mathcal{S}$ is the marginal distribution over the datasets) then the conclusion of the theorem holds up to an additional δ. This follows immediately from the fact that Theorem 3.2 holds for every dataset separately.

The assumption that the components of the mixture are supported on disjoint subdomains is potentially quite restrictive as it does not allow for ambiguous data points (for which Bayes optimal error is > 0). Subpopulations are also often modeled as Gaussians (or other distributions with unbounded support). If the probability of the overlap between the subpopulations is sufficiently small, then one can reduce this case to the disjoint one by modifying the components M_i to have disjoint supports while changing the marginal distribution over S by at most δ in the TV distance (and then appealing to the same argument as above). Dealing with a more general case allowing general overlap is significantly messier but the basic insight still applies: observing a single point sampled from some subpopulation increases the expectation of the frequency of the subpopulation under the posterior distribution. That increase can make this expectation significant which can making it necessary to memorize the label of the point.

3.1 Examples

We will now provide some intuition on why one would expect the (in-expectation) Λ-subpopulation-coupling to hold for some natural classes of algorithms. Our goal here is not to propose or justify specific models of data but rather to relate properties of known learning systems (and corresponding properties of data) to subpopulation coupling. Importantly, we aim to demonstrate that the coupling emerges from the interaction between the algorithm and the geometric properties of the data distribution and not from any explicit knowledge of subpopulations.

Local algorithms: A simple example of a class of algorithms that will exhibit subpopulation coupling is k-NN-like algorithms and other algorithms that are in some sense locally smooth. If subpopulations are sufficiently "clustered" so that including the example (x, y) in the predictor will affect the prediction in the neighborhood of x and the total weight of affected neighborhood is some fraction λ_1 of the subpopulation, then we will obtain subpopulation coupling with λ_1. In the more concrete (and extreme case), when for every point $x \in X$, the most distant point in X_x is closer than the closest point from the other subpopulations we will get that any example from a subpopulation will cause a 1-NN classifier to predict in the same way over the entire subpopulation. In particular, it would make it Λ-subpopulation-coupled for $\Lambda = (1, \ldots, 1)$.

Linear classifiers: A more interesting case to understand is that of linear classifiers and by extension SVMs and (in a limited sense) neural networks. We will examine a high-dimensional setting, where $d \gg n$. We will assume that points within each subpopulation are likely to have relatively large inner product whereas for every subpopulation most points will, with high probability have, a substantially large component that is orthogonal to the span of random samples from other populations. These conditions are impossible to satisfy when $d \leq n$ but are easy to satisfy when d is sufficiently large. Formally, we assume most of the points in most datasets sampled from our distribution will satisfy the following condition:

Definition 3.3. *Let $X \subset \mathbb{R}^d$ be a domain partitioned into subdomains $\{X_i\}_{i\in[N]}$. We say that a sequence of points $V = (x_1, \ldots, x_n)$ is (τ, θ)-independent if it holds that*

- *for all i, j such that $x_i, x_j \in X_t$ for some t, $\langle x_i, x_j\rangle \geq \tau\|x_i\|_2\|x_j\|_2$ and*
- *for all i such that $x_i \in X_t$, and any $v \in \mathtt{span}(V \setminus X_t)$, $|\langle x_i, v\rangle| \leq \theta\|x\|_2\|v\|_2$.*

We consider the performance of linear classifiers that approximately maximize the margin. Here, by "approximately" we will simply assume that they output classifiers that achieve at least $1/2$ of the optimal margin achievable when separating the same points in the given dataset. Note that algorithms with this property are easy to implement efficiently via SGD on the cross-entropy loss [Sou+18] and also via simple regularization of the Perceptron algorithm [SSS05]. We will also assume that the linear classifiers output by the algorithm lie in the span of the points in the dataset[3] Formally, we define approximately margin-maximizing algorithms in this multi-class setting (for convenience, restricted to the homogeneous case) as follows:

Definition 3.4. *An algorithm $\mathcal{A}$ is an* approximately margin maximizing m-class linear classifier *if given a dataset $S = ((x_1, y_1), \ldots, (x_n, y_n)) \in (X \times [m])^n$ it outputs m linear classifiers $w_1, \ldots, w_m$ satisfying:*

- *for every $k \in [m]$, w_k lies in the span of $x_1, \ldots, x_n$;*

[3] A linear classifier can always be projected to the span of the points without affecting the margins. This assumption allows us to avoid having to separately deal with spurious correlations between unseen parts of subpopulations and the produced classifiers.

- *for every x, the prediction of $\mathcal{A}$ on x depends only on the predictions of the classifiers* $\mathsf{sign}(\langle x, w_k \rangle)$ *and;*

- *for every $k \in [m]$, let $V_- := \{x \in X_S \mid \langle x, w_k \rangle < 0\}$ and $V_+ := \{x \in X_S \mid \langle x, w_k \rangle \geq 0\}$. If V_- can be linearly separated from V_+ by a homogeneous linear separator with margin γ_k then for all $x \in X_S$, $|\langle x, w_k \rangle| \geq \frac{\gamma_k}{2} \|x\|_2$.*

We now show that linear classifiers over distributions that produce datasets independent in the sense of Definition 3.3 will have high subpopulation coupling. In order to guarantee strong coupling, we will assume that the set V of points in a random dataset together with the set of points V' that consists of additional samples from every mixture present in V (namely, $V' \sim \prod_{j \in [N]_{S=1}} M_j$) satisfy the independence condition with high probability. Formally, we establish the following result (the proof can be found in Appendix B).

Theorem 3.5. *Let $X \subset \mathbb{R}^d$ be a domain partitioned into sub-domains $\{X_i\}_{i \in [N]}$ with subpopulations $\{M_i\}_{i \in [N]}$ over the sub-domains. Let $\mathcal{A}$ be any approximately margin maximizing m-class linear classifier and π be a frequency prior. Assume that for $D \sim \mathcal{D}_\pi^{[N]}$ and $V \sim M_D^n$, $V' \sim \prod_{j \in [N]_{S=1}} M_j$, with probability at least $1 - \delta^2$, $V \cup V'$ is $(\tau, \tau^2/(8\sqrt{n}))$-independent for some $\tau \in (0, 1/2]$. Then for any labeling prior $\mathcal{F}$, $\mathcal{A}$ is Λ-subpopulation-coupled with probability $1 - \delta$ and $\lambda_1 \geq 1 - \delta$.*

As a simple example of subpopulations that will produce sets of points that are $(\tau, \tau^2/(8\sqrt{n}))$ independent with high probability we pick each M_i to be a spherically-symmetric distribution supported on a ball of radius 1 around some center z_i of norm 1. We also pick the centers randomly and independently from the uniform distribution on the unit sphere. It is not hard to see that, by the standard concentration properties of spherically-symmetric distributions, a set V of t samples from an arbitrary mixture of such distributions will be (τ, θ)-independent with high probability for $\tau \geq 1/2 - o(1)$ and $\theta = \tilde{O}(\sqrt{t/d})$. Thus for $t < 2n$, $d = \tilde{O}(n^2)$ suffices to ensure that $\theta \leq \tau^2/(8\sqrt{n})$.

4 Memorization, Privacy and Stability

So far we have discussed memorization by learning algorithms informally. Below we will provide a simple definition of label memorization. We will show that bounds on the memorization ability of an algorithm can be easily translated into bounds on how well the algorithm can fit the dataset whenever there is enough uncertainty in the labels. This result immediately implies bounds on the generalization error of such algorithms in our problem setting. We will then show that (even relatively weak forms of) differential privacy imply that the algorithm cannot memorize well.

To keep the notation cleaner we will discuss these results in the context of our simpler model from Sec.2 but they can be easily adapted to our mixture model setting. For simplicity of notation, we will also focus on memorization of singleton elements.

4.1 Memorization

To measure the ability of an algorithm $\mathcal{A}$ to memorize labels we will look at how much the labeled example (x, y) affects the prediction of the model on x. This notion will be defined per specific dataset and example but in our applications we will use the expectation of this value when the dataset is drawn randomly.

Definition 4.1. *For a dataset $S = (x_i, y_i)_{i \in [n]}$ and $i \in [n]$ define*

$$\mathtt{mem}(\mathcal{A}, S, i) := \max\left\{0, \mathop{\mathbf{Pr}}_{h \sim \mathcal{A}(S)}[h(x_i) = y_i] - \mathop{\mathbf{Pr}}_{h \sim \mathcal{A}(S^{\setminus i})}[h(x_i) = y_i]\right\},$$

where $S_{x \setminus i}$ denotes the dataset that is S with (x_i, y_i) removed. We say that $\mathcal{A}$ is γ-memorization bounded for singletons if for all $S \in (X, Y)^n$ and all i such that $x_i \in X_{S=1}$ we have $\mathtt{mem}(\mathcal{A}, S, i) \leq \gamma$.

In this definition we measure the effect simply as the total variation distance between the distributions of the indicator of the label being y, but other notions of distance could be appropriate in other applications. For notion of distance our definition of memorization is closely related to leave-one-out stability of the algorithm (see eq. (6)). Indeed, it is easy to see from this definition that LOO stability upper bounds the expected memorization:

$$\frac{1}{n} \mathop{\mathbf{E}}_{S \sim P^n}\left[\sum_{i \in [n]} \mathtt{mem}(\mathcal{A}, S, i)\right] \leq \mathtt{LOOstab}(P, \mathcal{A}).$$

A simple corollary of this definition is that $\mathcal{A}$ cannot fit the label y if there is enough uncertainty in the label. To measure the uncertainty in a distribution ρ over labels we will simply use the maximum probability of any specific label:

$$\|\rho\|_\infty := \max_{y \in Y} \rho(y).$$

Lemma 4.2. *Let ρ be an arbitrary distribution over Y. For a dataset $S = (x_i, y_i)_{i \in [n]}$, $i \in [n]$ and $y \in Y$, let $S^{i \leftarrow y}$ denote the dataset S with (x_i, y) in place of example (x_i, y_i). Then we have:*

$$\mathop{\mathbf{E}}_{y \sim \rho, h \sim \mathcal{A}(S^{i \leftarrow y})}[h(x) = y] \leq \|\rho\|_\infty + \mathop{\mathbf{E}}_{y \sim \rho}[\mathtt{mem}(\mathcal{A}, S^{i \leftarrow y}, i)].$$

In particular, for every distribution D and labeling prior $\mathcal{F}$,

$$\mathop{\mathbf{E}}_{f \sim \mathcal{F}, S \sim (D,f)^n}[\mathtt{errn}_S(\mathcal{A}, 1)] \geq \mathop{\mathbf{E}}_{f \sim \mathcal{F}, S \sim (D,f)^n}\left[\sum_{i \in [n],\ x_i \in X_{S=1}} 1 - \|\mathcal{F}(x_i | S^{\setminus i})\|_\infty - \mathtt{mem}(\mathcal{A}, S, i)\right],$$

where $\mathcal{F}(x_i | S^{\setminus i})$ denotes the conditional distribution over the label of x_i after observing all the other examples:

$$\mathcal{F}(x_i | S^{\setminus i}) = \mathop{\mathbf{D}}_{f \sim \mathcal{F}, S \sim (D,f)^n}[f(x_i) \mid \forall j \neq i, f(x_j) = y_j].$$

Proof. Let ρ' denote the marginal distribution of $h(x)$ for $h \sim \mathcal{A}(S^{\setminus i})$. By Definition 4.1, for every y,

$$\mathop{\mathbf{Pr}}_{h \sim \mathcal{A}(S^{i \leftarrow y})}[h(x) = y] \leq \mathop{\mathbf{Pr}}_{h \sim \mathcal{A}(S^{\setminus i})}[h(x) = y] + \mathtt{mem}(\mathcal{A}, S^{i \leftarrow y}, i) = \rho'(y) + \mathtt{mem}(\mathcal{A}, S^{i \leftarrow y}, i).$$

Thus,

$$\mathop{\mathbf{E}}_{y \sim \rho, h \sim \mathcal{A}(S^{i \leftarrow y})}[h(x) = y] \leq \|\rho\|_\infty + \mathop{\mathbf{E}}_{y \sim \rho}[\mathtt{mem}(\mathcal{A}, S^{i \leftarrow y}, i)].$$

The rest of the claim follows from the definition of $\mathtt{errn}_S(\mathcal{A}, 1)$ and observing that an expectation is taken on $f \sim \mathcal{F}$ that ensures that for every point the error will be averaged over all labelings of the point according to conditional distribution of the corresponding label. □

Using these definitions and Lemma 4.2, we immediately obtain the following example corollary on the excess error of any algorithm that is γ-memorization bounded for singletons.

Corollary 4.3. *In the setting of Thm. 2.3, let $\mathcal{A}$ be any algorithm γ-memorization bounded for singletons. Then*

$$\overline{\texttt{err}}(\pi, \mathcal{F}, \mathcal{A}) \geq \texttt{opt}(\pi, \mathcal{F}) + \tau_1 \cdot \mathop{\mathbf{E}}_{D\sim D_\pi^X, f\sim\mathcal{F}, S\sim(D,f)^n} \left[\sum_{i\in[n],\ x_i\in X_{S=1}} 1 - \|\mathcal{F}(x_i|S^{\setminus i})\|_\infty - \gamma \right].$$

The bound in this corollary depends on the expectation of the uncertainty in the label $\|\mathcal{F}(x_i|S^{\setminus i})\|_\infty$. While, in general, this quantity might be hard to estimate it might be relatively easy to get a sufficiently strong upper bound. For example, if for $f \sim \mathcal{F}$ the labeling is k-wise independent for k that upper-bounds the typical number of distinct points (or subpopulations in the general case) then with high probability we will have that $\|\mathcal{F}(x_i|S^{\setminus i})\|_\infty \leq \max_{x\in X} \|\rho_x\|_\infty$, where ρ_x is the marginal distribution of $f(x)$ for $f \sim \mathcal{F}$.

4.2 Privacy

Memorization of the training data can be undesirable in a variety of settings. For example, in the context of user data privacy, memorization is known to lead to ability to mount black-box membership inference attacks (that discover the presence of a specific data point in the dataset) [Sho+17; LBG17; Lon+18; Tru+18] as well as ability to extract planted secrets from language models [Car+19]. The most common approaches toward defending such attacks are based on the notion of differential privacy [Dwo+06] that are formally known to limit the probability of membership inference by requiring that the output distribution of the learning algorithm is not too sensitive to individual data points. Despite significant recent progress in training deep learning networks with differential privacy, they still lag substantially behind the state-of-the-art results trained without differential privacy [SS15; Aba+16; Pap+16; Wu+17; Pap+17; McM+18]. While some of this lag is likely to be closed by improved techniques, our results imply that the some of this gap is inherent due to the data being long-tailed. More formally, we will show that the requirements differential privacy imply a lower bound on the value of $\texttt{errn}$ (for simplicity just for $\ell = 1$). We will prove that this limitation applies even to algorithms that satisfy a very weak form of privacy: label privacy for predictions. It protects only the privacy of the label as in [CH11] and also with respect to algorithms that only output a prediction on an (arbitrary) fixed point [DF18]. Formally, we define:

Definition 4.4. *Let $\mathcal{A}$ be an algorithm that given a dataset $S \in (X \times Y)^n$ outputs a random predictor $h\colon X \to Y$. We say that $\mathcal{A}$ is* (ϵ, δ)-differentially label-private prediction *algorithm if for every $x \in X$ and two datasets S, S' that only differ in a label of a single element we have for any subset of labels Y',*

$$\mathop{\mathbf{Pr}}_{h\sim\mathcal{A}(S)}[h(x) \in Y'] \leq e^\epsilon \cdot \mathop{\mathbf{Pr}}_{h\sim\mathcal{A}(S')}[h(x) \in Y'] + \delta.$$

It is easy to see that any algorithm that satisfies this notion of privacy is $(e^\epsilon - 1 + \delta)$-memorization bounded for singletons. A slightly more careful analysis in this case gives the following analogues of Lemma 4.2 and Corollary 4.3.

Theorem 4.5. *Let $\mathcal{A}$ be an (ϵ, δ)-differentially label-private prediction algorithm and let ρ be an arbitrary distribution over Y. For a dataset $S = (x_i, y_i)_{i\in[n]}$, $i \in [n]$ and $y \in Y$, we have:*

$$\mathop{\mathbf{E}}_{y\sim\rho, h\sim\mathcal{A}(S^{i\leftarrow y})}[h(x) = y] \leq e^\epsilon \cdot \|\rho\|_\infty + \delta.$$

In particular, in the setting of Thm. 2.3, for every distribution D and labeling prior $\mathcal{F}$,

$$\mathop{\mathbf{E}}_{f\sim\mathcal{F},S\sim(D,f)^n}[\mathtt{errn}_S(\mathcal{A},1)] \geq \mathop{\mathbf{E}}_{f\sim\mathcal{F},S\sim(D,f)^n}\left[\sum_{i\in[n],\ x_i\in X_{S=1}} 1-e^{\epsilon}\cdot\|\mathcal{F}(x_i|S^{\setminus i})\|_{\infty}-\delta\right].$$

and, consequently,

$$\overline{\mathtt{err}}(\pi,\mathcal{F},\mathcal{A}) \geq \mathtt{opt}(\pi,\mathcal{F}) + \tau_1\cdot\mathop{\mathbf{E}}_{D\sim D_{\pi}^{X},f\sim\mathcal{F},S\sim(D,f)^n}\left[\sum_{i\in[n],\ x_i\in X_{S=1}} 1-e^{\epsilon}\cdot\|\mathcal{F}(x_i|S^{\setminus i})\|_{\infty}-\delta\right].$$

Proof. Let ρ' denote the marginal distribution of $h(x)$ for $h\sim\mathcal{A}(S)$. By the definition of (ϵ,δ)-differential label privacy for predictions, for every y,

$$\mathop{\mathbf{Pr}}_{h\sim\mathcal{A}(S^{i\leftarrow y})}[h(x_i)=y] \leq e^{\epsilon}\cdot\mathop{\mathbf{Pr}}_{h\sim\mathcal{A}(S)}[h(x)=y]+\delta = e^{\epsilon}\rho'(y)+\delta$$

Thus,

$$\mathop{\mathbf{E}}_{y\sim\rho,h\sim\mathcal{A}(S^{i\leftarrow y})}[h(x_i)=y] \leq \mathop{\mathbf{E}}_{y\sim\rho}\left[e^{\epsilon}\rho'(y)+\delta\right] \leq e^{\epsilon}\|\rho\|_{\infty}+\delta.$$

The rest of the claim follows as before. □

This theorem is easy to extend to any subpopulation from which only ℓ examples have been observed using the group privacy property of differential privacy. This property implies that if ℓ labels are changed then the resulting distributions are $(\ell\epsilon,\ell e^{\ell-1}\delta)$-close (in the same sense) [DR14]. The total weight of subpopulations that have at most ℓ examples for a small value of ℓ is likely to be significant in most modern datasets. Thus this may formally explain at least some of the gap in the results currently achieved using differentially private training algorithms and those achievable without the privacy constraint.

Uniform stability: A related notion of stability is uniform prediction stability [BE02; DF18] that, in the context of prediction, requires that changing any point in the dataset does not change the label distribution on any point by more than δ in total variation distance. This notion is useful in ensuring generalization [BE02; FV19] and as a way to ensure robustness of predictions against data poisoning. In this context, δ-uniform stability implies $(0,\delta)$-differential privacy for predictions and therefore Theorem 4.5 also implies limitations of such algorithms.

5 Discussion

Several additional points and comments about our approach are worth making.

Our modeling ignores some of the potentially significant costs of fitting/memorizing 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 a singleton 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 for $\ell=1$ scaled by Δ. In other words, despite the label noise, memorization might still be beneficial for these examples.

At the same time, our model does not rule out effective noise/outlier removal on samples *within* a subpopulation. Such procedures may improve the generalization error while also turning the model into non-interpolating. Some learning algorithms may be able to fit noisy examples within subpopulations while avoiding interference with correctly classified ones. Our model does not directly address the benefits of interpolation in such cases. That said, given that memorization has benefits, these benefits may simply outweigh the costs (at least in some error regimes).

Our model is not specific to the interpolation regime and applies also in setting where the learning algorithm cannot fit the entire dataset due to capacity restrictions or progressively stronger regularization. Effectively, interpolation and fitting of random labels are just symptoms of the benefits of memorization. The phenomenon itself is likely to manifest itself beyond the interpolation regime. Adapted to a non-interpolating regime, our theory predicts that achieving close-to-optimal error (in the presence of long-tailed mixtures) requires a larger generalization gap than that explainable by standard analyses. In other words, even without interpolation, our work justifies rebalancing the model's capacity (as measured by its ability to fit arbitrary examples) and training error in favor of memorization. Such skewed balance is common in ML practice but has previously lacked a general justification in theory.

Our model is agnostic to specific mechanisms used to effectively discover representations in which there is little interference between subpopulations. Understanding of these mechanisms in the context of DNNs remains an important and challenging problem.

Finally, while we believe that there is already significant empirical evidence supporting our modeling assumptions and conclusions, additional empirical investigation could further elucidate several aspects of our model and verify its predictions. Such investigation is currently ongoing but is outside the scope of this manuscript.

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 [CEP18] 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 Misha Belkin, Edith Cohen, Roy Frostig, Daniel Hsu, Tomer Koren, Sasha Rakhlin, Kunal Talwar and Chiyuan Zhang for insightful feedback and suggestions on this work.

References

[Aba+16] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. "Deep learning with differential privacy". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2016, pp. 308–318.

[Arp+17] D. Arpit, S. Jastrzkebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al. "A closer look at memorization in deep networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 233–242.

[BE02] O. Bousquet and A. Elisseeff. "Stability and generalization". In: *JMLR* 2 (2002), pp. 499–526.

[Bel+18] M. Belkin, D. Hsu, S. Ma, and S. Mandal. "Reconciling modern machine learning and the bias-variance trade-off". In: *arXiv preprint arXiv:1812.11118* (2018).

[BFT17] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky. "Spectrally-normalized margin bounds for neural networks". In: *Advances in Neural Information Processing Systems*. 2017, pp. 6240–6249.

[BHM18] M. Belkin, D. J. Hsu, and P. Mitra. "Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate". In: *Advances in Neural Information Processing Systems*. 2018, pp. 2300–2311.

[BHX19] M. Belkin, D. Hsu, and J. Xu. "Two models of double descent for weak features". In: *arXiv preprint arXiv:1903.07571* (2019).

[BM02] P. Bartlett and S. Mendelson. "Rademacher and Gaussian Complexities: Risk Bounds and Structural Results". In: *Journal of Machine Learning Research* 3 (2002), pp. 463–482.

[BMM18] M. Belkin, S. Ma, and S. Mandal. "To Understand Deep Learning We Need to Understand Kernel Learning". In: *ICML*. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 541–549. URL: http://proceedings.mlr.press/v80/belkin18a.html.

[Bre01] L. Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[BRT18] M. Belkin, A. Rakhlin, and A. B. Tsybakov. "Does data interpolation contradict statistical optimality?" In: *arXiv preprint arXiv:1806.09471* (2018).

[Car+19] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song. "The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks". In: *Usenix Security (to appear)*. 2019.

[CEP18] N. Carlini, U. Erlingsson, and N. Papernot. "Prototypical Examples in Deep Learning: Metrics, Characteristics, and Utility". In: (2018). URL: https://openreview.net/forum?id=r1xyx3R9tQ.

[CH11] K. Chaudhuri and D. Hsu. "Sample Complexity Bounds for Differentially Private Learning". In: *COLT*. 2011, pp. 155–186.

[CH+67] T. M. Cover, P. E. Hart, et al. "Nearest neighbor pattern classification". In: *IEEE transactions on information theory* 13.1 (1967), pp. 21–27.

[CSG18] G. Cohen, G. Sapiro, and R. Giryes. "DNN or k-NN: That is the Generalize vs. Memorize Question". In: *arXiv preprint arXiv:1805.06822* (2018).

[Cui+18] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie. "Large Scale Fine-Grained Categorization and Domain-Specific Transfer Learning". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[CV95] C. Cortes and V. Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297.

[DF18] C. Dwork and V. Feldman. "Privacy-preserving Prediction". In: *Conference On Learning Theory*. 2018, pp. 1693–1702.

[DR14] C. Dwork and A. Roth. *The Algorithmic Foundations of Differential Privacy*. Vol. 9. 3-4. 2014, pp. 211–407. URL: http://dx.doi.org/10.1561/0400000042.

[Dwo+06] C. Dwork, F. McSherry, K. Nissim, and A. Smith. "Calibrating noise to sensitivity in private data analysis". In: *TCC*. 2006, pp. 265–284.

[Dwo+14] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. "Preserving Statistical Validity in Adaptive Data Analysis". In: *CoRR* abs/1411.2664 (2014). Extended abstract in STOC 2015.

[Fel16] V. Feldman. "Generalization of ERM in Stochastic Convex Optimization: The Dimension Strikes Back". In: *CoRR* abs/1608.04414 (2016). Extended abstract in NIPS 2016. URL: http://arxiv.org/abs/1608.04414.

[FS97] Y. Freund and R. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139.

[FV19] V. Feldman and J. Vondrák. "High probability generalization bounds for uniformly stable algorithms with nearly optimal rate". In: *CoRR* abs/1902.10710 (2019). arXiv: 1902.10710. URL: http://arxiv.org/abs/1902.10710.

[HRS16] M. Hardt, B. Recht, and Y. Singer. "Train faster, generalize better: Stability of stochastic gradient descent". In: *ICML*. 2016, pp. 1225–1234. URL: http://jmlr.org/proceedings/papers/v48/hardt16.html.

[Kol10] V. Koltchinskii. "Rademacher Complexities and Bounding the Excess Risk in Active Learning". In: *JMLR* 11 (2010), pp. 2457–2485.

[LBG17] Y. Long, V. Bindschaedler, and C. A. Gunter. "Towards Measuring Membership Privacy". In: *CoRR* abs/1712.09136 (2017). arXiv: 1712.09136. URL: http://arxiv.org/abs/1712.09136.

[Lon+18] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen. "Understanding Membership Inferences on Well-Generalized Learning Models". In: *CoRR* abs/1802.04889 (2018). arXiv: 1802.04889. URL: http://arxiv.org/abs/1802.04889.

[LR18] T. Liang and A. Rakhlin. "Just interpolate: Kernel" ridgeless" regression can generalize". In: *arXiv preprint arXiv:1808.00387* (2018).

[MBB18] S. Ma, R. Bassily, and M. Belkin. "The Power of Interpolation: Understanding the Effectiveness of SGD in Modern Over-parametrized Learning". In: *ICML*. 2018, pp. 3331–3340. URL: http://proceedings.mlr.press/v80/ma18a.html.

[McM+18] B. McMahan, D. Ramage, K. Talwar, and L. Zhang. "Learning Differentially Private Recurrent Language Models". In: *International Conference on Learning Representations (ICLR)*. 2018. URL: https://openreview.net/pdf?id=BJ0hF1Z0b.

[Muk+06] S. Mukherjee, P. Niyogi, T. Poggio, and R. Rifkin. "Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization". In: *Advances in Computational Mathematics* 25.1-3 (2006), pp. 161–193.

[Ney+17a] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. "Exploring generalization in deep learning". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5947–5956.

[Ney+17b] B. Neyshabur, R. Tomioka, R. Salakhutdinov, and N. Srebro. "Geometry of optimization and implicit regularization in deep learning". In: *arXiv preprint arXiv:1705.03071* (2017).

[OS15] A. Orlitsky and A. T. Suresh. "Competitive distribution estimation: Why is good-turing good". In: *NIPS*. 2015, pp. 2143–2151.

[Pap+16] N. Papernot, M. Abadi, Ú. Erlingsson, I. J. Goodfellow, and K. Talwar. "Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data". In: *CoRR* abs/1610.05755 (2016). arXiv: 1610.05755. URL: http://arxiv.org/abs/1610.05755.

[Pap+17] N. Papernot, M. Abadi, Ú. Erlingsson, I. J. Goodfellow, and K. Talwar. "Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data". In: *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. 2017.

[RMP05] A. Rakhlin, S. Mukherjee, and T. Poggio. "Stability Results In Learning Theory". In: *Analysis and Applications* 03.04 (2005), pp. 397–417.

[RZ18] A. Rakhlin and X. Zhai. "Consistency of Interpolation with Laplace Kernels is a High-Dimensional Phenomenon". In: *arXiv preprint arXiv:1812.11167* (2018).

[Sch13] R. E. Schapire. "Explaining adaboost". In: *Empirical inference*. Springer, 2013, pp. 37–52.

[Sch+98] R. Schapire, Y. Freund, P. Bartlett, and W. Lee. "Boosting the margin: a new explanation for the effectiveness of voting methods". In: *Annals of Statistics* 26.5 (1998), pp. 1651–1686.

[Sha+09] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. "Stochastic Convex Optimization". In: *COLT*. 2009.

[Sha+10] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. "Learnability, Stability and Uniform Convergence". In: *Journal of Machine Learning Research* 11 (2010), pp. 2635–2670. URL: `http://portal.acm.org/citation.cfm?id=1953019`.

[Sho+17] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. "Membership Inference Attacks Against Machine Learning Models". In: *2017 IEEE Symposium on Security and Privacy, SP 2017*. 2017, pp. 3–18.

[Sou+18] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. "The implicit bias of gradient descent on separable data". In: *The Journal of Machine Learning Research* 19.1 (2018), pp. 2822–2878.

[Sri02] K. Sridharan. *A gentle introduction to concentration inequalities*. Tech. rep. 2002.

[SS15] R. Shokri and V. Shmatikov. "Privacy-preserving deep learning". In: *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. ACM. 2015, pp. 1310–1321.

[SSS05] S. Shalev-Shwartz and Y. Singer. "A new perspective on an old perceptron algorithm". In: *International Conference on Computational Learning Theory*. Springer. 2005, pp. 264–278.

[Tru+18] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei. "Towards demystifying membership inference attacks". In: *arXiv preprint arXiv:1807.09173* (2018).

[Vap82] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. New York: Springer-Verlag, 1982.

[VH+18] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. "The inaturalist species classification and detection dataset". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8769–8778.

[VHP17] G. Van Horn and P. Perona. "The devil is in the tails: Fine-grained classification in the wild". In: *arXiv preprint arXiv:1709.01450* (2017).

[VV16] G. Valiant and P. Valiant. "Instance optimal learning of discrete distributions". In: *STOC*. ACM. 2016, pp. 142–155.

[WRH17] Y.-X. Wang, D. Ramanan, and M. Hebert. "Learning to model the tail". In: *Advances in Neural Information Processing Systems*. 2017, pp. 7029–7039.

[Wu+17] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. F. Naughton. "Bolt-on Differential Privacy for Scalable Stochastic Gradient Descent-based Analytics". In: *SIGMOD*. 2017, pp. 1307–1322.

[Wyn+17] A. J. Wyner, M. Olson, J. Bleich, and D. Mease. "Explaining the success of adaboost and random forests as interpolating classifiers". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 1558–1590.

[ZAR14] X. Zhu, D. Anguelov, and D. Ramanan. "Capturing long-tail distributions of object subcategories". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 915–922.

[Zha+17] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. "Understanding deep learning requires rethinking generalization". In: *ICLR*. 2017. URL: https://openreview.net/forum?id=Sy8gdB9xx.

A Proof Lemma 2.1

The key property of our problem definition is that it allows to decompose the probability of a dataset (under the entire generative process) into a probability of seeing one of the points in the dataset and the probability of seeing the rest of the dataset under a similar generative process. Specifically, we prove the following lemma.

Lemma A.1. *For $x \in X$, a sequence of points $V = (x_1, \ldots, x_n) \in X^n$ that includes x exactly ℓ times, let $V \setminus x$ be equal to V with all the elements equal to x omitted. Then for any frequency prior π and α in the support of $\bar{\pi}^N$, we have*

$$\Pr_{D\sim\mathcal{D}_\pi^X, U\sim D^n}[U = V \mid D(x) = \alpha] = \alpha^\ell \cdot (1-\alpha)^{n-\ell} \cdot \Pr_{D'\sim\mathcal{D}_\pi^{X\setminus\{x\}}, U'\sim D^{n-\ell}}[U' = V \setminus x].$$

In particular:

$$\Pr_{D\sim\mathcal{D}_\pi^X, U\sim D^n}[U = V] = \mathop{\mathbf{E}}_{\alpha\sim\bar{\pi}^N}\left[\alpha^\ell \cdot (1-\alpha)^{n-\ell}\right] \cdot \Pr_{D'\sim\mathcal{D}_\pi^{X\setminus\{x\}}, U'\sim D^{n-\ell}}[U' = V \setminus x].$$

Proof. We consider the distribution of $D \sim \mathcal{D}_\pi^X$ conditioned on $D(x) = \alpha$ (which, by our assumption, is an event with positive probability). We denote this distribution by $\mathcal{D}_\pi^X(|D(x) = \alpha)$. From the definition of $\mathcal{D}_\pi^X$ we get that a random sample D from $\mathcal{D}_\pi^X(|D(x) = \alpha)$ can generated by setting $D(x) = \alpha$, then for all $z \in X \setminus \{x\}$, sampling p_z from π and normalizing the results to sum to $1 - \alpha$. That is, defining

$$D(z) = (1-\alpha)\frac{p_z}{\sum_{z\in X\setminus\{x\}} p_z}.$$

From here we obtain that an equivalent way to generate a random sample from $\mathcal{D}_\pi^X(|D(x) = \alpha)$ is to sample D' from $\mathcal{D}_\pi^{X\setminus\{x\}}$ and then multiply the resulting p.m.f. by $1 - \alpha$ (with $D(x) = \alpha$ as before). Naturally, for any D,

$$\Pr_{U\sim D^n}[U = V] = \prod_{i\in[n]} D(x_i).$$

Now we denote by I_{-x} the subset of indices of elements of V that are different from x: $I_x = \{i \in [n] \mid x_i \neq x\}$. We can now conclude:

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{D\sim\mathcal{D}_\pi^X, U\sim D^n}[U = V \mid D(x) = \alpha] &= \mathop{\mathbf{Pr}}_{D\sim\mathcal{D}_\pi^X(|D(x)=\alpha), U\sim D^n}[U = V] \\
&= \mathop{\mathbf{E}}_{D\sim\mathcal{D}_\pi^X(|D(x)=\alpha)}\left[\prod_{i\in[n]} D(x_i)\right] \\
&= \mathop{\mathbf{E}}_{D\sim\mathcal{D}_\pi^X(|D(x)=\alpha)}\left[\alpha^\ell \prod_{i\in I_{-x}} D(x_i)\right] \\
&= \alpha^\ell \cdot (1-\alpha)^{n-\ell} \cdot \mathop{\mathbf{E}}_{D\sim\mathcal{D}_\pi^X(|D(x)=\alpha)}\left[\prod_{i\in I_{-x}} \frac{D(x_i)}{1-\alpha}\right] \\
&= \alpha^\ell \cdot (1-\alpha)^{n-\ell} \cdot \mathop{\mathbf{E}}_{D'\sim\mathcal{D}_\pi^{X\setminus\{x\}}}\left[\prod_{i\in I_{-x}} D'(x_i)\right] \\
&= \alpha^\ell \cdot (1-\alpha)^{n-\ell} \cdot \mathop{\mathbf{Pr}}_{D'\sim\mathcal{D}_\pi^{X\setminus\{x\}}, U'\sim D^{n-\ell}}[U' = V \setminus x].
\end{aligned}$$

The second part of the claim follows directly from the fact that, by definition of $\bar{\pi}^N$,

$$\mathop{\mathbf{Pr}}_{D\sim\mathcal{D}_\pi^X, U\sim D^n}[D(x) = \alpha] = \bar{\pi}^N(\alpha).$$

□

We can now prove Lemma 2.1 which we restate here for convenience.

Lemma A.2 (Lemma 2.1 restated)**.** *For any frequency prior π, $x \in X$ and a sequence of points $V = (x_1, \ldots, x_n) \in X^n$ that includes x exactly ℓ times, we have*

$$\mathop{\mathbf{E}}_{D\sim\mathcal{D}_\pi^X, U\sim D^n}[D(x) \mid U = V] = \frac{\mathbf{E}_{\alpha\sim\bar{\pi}^N}\left[\alpha^{\ell+1}\cdot(1-\alpha)^{n-\ell}\right]}{\mathbf{E}_{\alpha\sim\bar{\pi}^N}\left[\alpha^\ell\cdot(1-\alpha)^{n-\ell}\right]}.$$

Proof. We first observe that by the Bayes rule and Lemma A.1:

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{D\sim\mathcal{D}_\pi^N, U\sim D^n}[D(x) = \alpha \mid U = V] &= \frac{\mathbf{Pr}_{D\sim\mathcal{D}_\pi^N, U\sim D^n}[U = V \mid D(x) = \alpha]\cdot\mathbf{Pr}_{D\sim\mathcal{D}_\pi^N, U\sim D^n}[D(x) = \alpha]}{\mathbf{Pr}_{D\sim\mathcal{D}_\pi^N, U\sim D^n}[U = V]} \\
&= \frac{\alpha^\ell\cdot(1-\alpha)^{n-\ell}\cdot\mathbf{Pr}_{D'\sim\mathcal{D}_\pi^{X\setminus\{x\}}, U'\sim D^{n-\ell}}[U' = V\setminus x]\cdot\bar{\pi}^N(\alpha)}{\mathbf{E}_{\beta\sim\bar{\pi}^N}\left[\beta^\ell\cdot(1-\beta)^{n-\ell}\right]\cdot\mathbf{Pr}_{D'\sim\mathcal{D}_\pi^{X\setminus\{x\}}, U'\sim D^{n-\ell}}[U' = V\setminus x]} \\
&= \frac{\alpha^\ell\cdot(1-\alpha)^{n-\ell}\cdot\bar{\pi}^N(\alpha)}{\mathbf{E}_{\beta\sim\bar{\pi}^N}\left[\beta^\ell\cdot(1-\beta)^{n-\ell}\right]}.
\end{aligned}$$

This leads to the claim:

$$\begin{aligned}\mathop{\mathbf{E}}_{D\sim\mathcal{D}_\pi^N,U\sim D^n}[D(x)\mid U=V] &= \sum_{\alpha\in\mathtt{supp}(\bar\pi^N)}\alpha\cdot\mathop{\mathbf{Pr}}_{D\sim\mathcal{D}_\pi^N,U\sim D^n}[D(x)=\alpha\mid U=V]\\ &= \sum_{\alpha\in\mathtt{supp}(\bar\pi^N)}\alpha\cdot\frac{\alpha^\ell\cdot(1-\alpha)^{n-\ell}\cdot\bar\pi^N(\alpha)}{\mathbf{E}_{\beta\sim\bar\pi^N}\left[\beta^\ell\cdot(1-\beta)^{n-\ell}\right]}\\ &= \frac{\mathbf{E}_{\alpha\sim\bar\pi^N}\left[\alpha^{\ell+1}\cdot(1-\alpha)^{n-\ell}\right]}{\mathbf{E}_{\alpha\sim\bar\pi^N}\left[\alpha^\ell\cdot(1-\alpha)^{n-\ell}\right]}.\end{aligned}$$

□

B Proof of Theorem 3.5

Theorem B.1 (Thm. 3.5 restated)**.** *Let $X\subset\mathbb{R}^d$ be a domain partitioned into sub-domains $\{X_i\}_{i\in[N]}$ with subpopulations $\{M_i\}_{i\in[N]}$ over the sub-domains. Let $\mathcal{A}$ be any approximately margin maximizing m-class linear classifier and π be a frequency prior. Assume that for $D\sim\mathcal{D}_\pi^{[N]}$ and $V\sim M_D^n$, $V'\sim\prod_{j\in[N]_{S=1}}M_j$, with probability at least $1-\delta^2$, $V\cup V'$ is $(\tau,\tau^2/(8\sqrt{n}))$-independent for some $\tau\in(0,1/2]$. Then for any labeling prior $\mathcal{F}$, $\mathcal{A}$ is Λ-subpopulation-coupled with probability $1-\delta$ and $\lambda_1\geq 1-\delta$.*

Proof. For the given priors π and $\mathcal{F}$, let $S=((x_1,y_1),\ldots,(x_n,y_n))$ be a dataset sampled from $(M_D,L_f)^n$ for $D\sim\mathcal{D}_\pi^{[N]}$ and $f\sim\mathcal{F}$. Let $V=(x_1,\ldots,x_n)$. Let $T:=[N]_{S=1}$ and let $V'=(x'_j)_{j\in K}$ be sampled from $\prod_{j\in T}M_j$, that is additional samples from every subpopulation with a single sample.

We will show that for any $V\cup V'$ that is $(\tau,\theta:=\tau^2/(8\sqrt{n}))$ independent, the output $w_1,\ldots,w_m$ of any approximately margin maximizing m-class linear classifier $\mathcal{A}$ gives predictions on V' that are consistent with those on V (which are defined by S): if $x_i\in X_t$ for $t\in T$ then for every $k\in[m]$,

$$\mathsf{sign}(\langle w_k,x'_t\rangle)=\mathsf{sign}(\langle w_k,x_i\rangle).$$

This implies that the prediction of the classifier on x'_t is identical to that on x_i. By our assumption, $V\cup V'$ is *not* $(\tau,\tau/(4\sqrt{n}))$ independent with probability at most δ^2. By Markov's inequality, probability over the choice of V such that the probability over the choice of V' that $V\cup V'$ is not (τ,θ) independent with probability more than δ is at most δ. By our definition of V', the marginal distribution of x'_t is exactly M_t. Therefore this implies that with probability at least $1-\delta$ over the choice of the dataset S, for every $x\in X_{S=1}$, and $x'\sim M_x$ we have

$$\mathtt{TV}\left(\mathop{\mathbf{D}}_{h\sim\mathcal{A}(S)}[h(x)],\mathop{\mathbf{D}}_{x'\sim M_x,h\sim\mathcal{A}(S)}[h(x')]\right)\leq\delta$$

as required by Definition 3.1 (for $\ell=1$).

To prove the stated consistency property for $V\cup V'$ that is (τ,θ) that is independent, we will first show that every class can be separated from the other ones with margin γ of $\Omega(1/\sqrt{n})$. We will then use the properties of approximately margin maximizing classifiers and, again, independence to obtain consistency.

For any vector v, we denote $\bar v:=v/\|v\|_2$. To show that the margin is large, we define the weights explicitly by using one representative point from every subpopulation in V. Without loss of generality, we can assume that these representatives are $x_1,\ldots,x_r$ for some $r\leq n$. Let $z_1,\ldots,z_r\in\{\pm1\}$ be an arbitrary

partition of these representatives into positively and negatively labeled ones. We define $w := \sum_{j\in[r]} z_j \bar{x}_j$ and consider the linear separator given by $\bar{w}$.

To evaluate the margin we first observe that $\|w\|_2 \leq \sqrt{2r}$. This follows via induction on r:

$$\begin{aligned}\left\|\sum_{j\in[r]} z_j\bar{x}_j\right\|_2^2 &= \left\|\sum_{j\in[r-1]} z_j\bar{x}_j\right\|_2^2 + \|\bar{x}_j\|_2^2 + 2z_r\left\langle \sum_{j\in[r-1]} z_j\bar{x}_j, \bar{x}_r\right\rangle \\ &\leq 2(r-1)+1+2\frac{\tau^2}{8\sqrt{n}}\left\|\sum_{j\in[r-1]} z_j\bar{x}_j\right\|_2 \\ &\leq \frac{4(r-1)}{3}+1+\frac{1}{16\sqrt{n}}\cdot\sqrt{2(r-1)} \leq 2r.\end{aligned}$$

Now for $i \in [n]$ assume that $x_i \in X_t$ and (without loss of generality) that x_r is the representative of subdomain X_t.

Then

$$\begin{aligned}z_r\langle \bar{x}_i, \bar{w}\rangle &= \frac{1}{\|w\|_2}\left(\langle \bar{x}_i, \bar{x}_r\rangle + z_r\left\langle \bar{x}_i, \sum_{j\in[r-1]} z_j\bar{x}_j\right\rangle\right) \\ &\geq \frac{1}{\|w\|_2}\left(\tau - \frac{\tau^2}{8\sqrt{n}}\left\|\sum_{j\in[r-1]} z_j\bar{x}_j\right\|_2\right) \\ &\geq \frac{\tau}{\|w\|_2}\left(1-\frac{\tau\sqrt{2(r-1)}}{8\sqrt{n}}\right) \geq \frac{\tau}{2\sqrt{n}}.\end{aligned}$$

Thus we obtain that x_i is labeled in the same way as its representative x_r and with margin of at least $\frac{\tau}{2\sqrt{n}}$. This holds for all $i \in [n]$ and therefore $\bar{w}$ shows that the desired separation can be achieved with margin of at least $\frac{\tau}{2\sqrt{n}}$.

Let $w_1, \ldots, w_k$ be the linear separators returned by $\mathcal{A}$. Let w be one of them. By our assumptions on $\mathcal{A}$, w separates V with margin of at least $\gamma := \frac{\tau}{4\sqrt{n}}$ and further it lies in the span on V. Namely, there exist $\alpha_1, \ldots, \alpha_n$ such that $w = \sum_{i\in[n]} \alpha_i \bar{x}_i$.

We now pick an arbitrary singleton point from V. Without loss of generality we assume that it is x_n, $\langle x_n, w\rangle \geq \gamma\|x_n\|_2$ and let $x \in V'$ be the point from the same subdomain X_t. Let $v := \sum_{i\in[n-1]} \alpha_i\bar{x}_i$ be the part of w that excludes x_n. By our assumption x_n is a singleton and therefore the points in $(x_1, \ldots, x_{n-1})$ are from other subdomains. By the independence of V, this implies that $|\langle \bar{x}_n, v\rangle| \leq \theta\|v\|_2$ and $|\langle \bar{x}, v\rangle| \leq \theta\|v\|_2$.

Now we need to show that the margin condition implies that α_n is sufficiently large. Specifically,

$$\gamma \leq \langle \bar{x}_n, w\rangle = \alpha_n + \langle \bar{x}_n, v\rangle \leq \alpha_n + \theta\|v\|_2,$$

and thus

$$\alpha_n \geq \gamma - \theta\|v\|_2 \geq \gamma - \theta(1+\alpha_n),$$

where we used the fact that, by triangle inequality, $\|v\|_2 \leq \|w\|_2 + \|\alpha_n\bar{x}_n\|_2 \leq 1 + \alpha_n$. This implies that

$\alpha_n \geq \frac{\gamma-\theta}{1+\theta}$. We can now ready to bound $\langle \bar{x}, w \rangle$

$$\langle \bar{x}, w \rangle = \langle \alpha_n \bar{x}, \bar{x}_n \rangle + \langle \bar{x}, v \rangle \geq \alpha_n \tau - \theta \|v\|_2 \geq \alpha_n \tau - \theta(1+\alpha_n) = \alpha_n(\tau - \theta) - \theta$$
$$\geq \frac{(\gamma-\theta)(\tau-\theta)}{1+\theta} - \theta \geq \frac{\left(\frac{\tau}{4\sqrt{n}} - \frac{\tau^2}{8\sqrt{n}}\right)\left(\tau - \frac{\tau^2}{8\sqrt{n}}\right)}{1+\frac{\tau^2}{8\sqrt{n}}} - \frac{\tau^2}{8\sqrt{n}} > 0.$$

where the last inequality assumes that $n \geq 4$. Thus we obtain that for every $w \in \{w_1, \ldots, w_m\}$, every point in $V' \cap X_{S=1}$ will be classified by w in the same way as the point from the same subpopulation in S. $\square$