When Is Memorization of Irrelevant Training Data Necessary for High-Accuracy Learning?*

Gavin Brown
Department of Computer Science
Boston University
Boston, Massachusetts, USA
grbrown@bu.edu

Mark Bun
Department of Computer Science
Boston University
Boston, Massachusetts, USA
mbun@bu.edu

Vitaly Feldman
Apple
Cupertino, California, USA
vitaly.edu@gmail.com

Adam Smith
Department of Computer Science
Boston University
Boston, Massachusetts, USA
ads22@bu.edu

Kunal Talwar
Apple
Cupertino, California, USA
kunal@kunaltalwar.org

ABSTRACT

Modern machine learning models are complex and frequently encode surprising amounts of information about individual inputs. In extreme cases, complex models appear to memorize entire input examples, including seemingly irrelevant information (social security numbers from text, for example). In this paper, we aim to understand whether this sort of memorization is necessary for accurate learning. We describe natural prediction problems in which every sufficiently accurate training algorithm must encode, in the prediction model, essentially all the information about a large subset of its training examples. This remains true even when the examples are high-dimensional and have entropy much higher than the sample size, and even when most of that information is ultimately irrelevant to the task at hand. Further, our results do not depend on the training algorithm or the class of models used for learning.

Our problems are simple and fairly natural variants of the next-symbol prediction and the cluster labeling tasks. These tasks can be seen as abstractions of text- and image-related prediction problems. To establish our results, we reduce from a family of one-way communication problems for which we prove new information complexity lower bounds.

CCS CONCEPTS

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

KEYWORDS

Memorization, Information Complexity, Overparameterization

ACM Reference Format:
Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. 2021. When Is Memorization of Irrelevant Training Data Necessary for High-Accuracy Learning?. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21), June 21–25, 2021, Virtual, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3406325.3451131

1 INTRODUCTION

Algorithms for supervised machine learning take in training data, attempt to extract the relevant information, and produce a prediction algorithm, also called a *model* or *hypothesis*. The model is used to predict a particular feature on future examples, ideally drawn from the same distribution as the training data. Such algorithms operate on a huge range of prediction tasks, from image classification to language translation, often involving highly sensitive data. To succeed, models must of course contain information about the data they were trained on. In fact, many well-known machine learning algorithms create models that explicitly encode their training data: the "model" for the k-Nearest Neighbor classification algorithm is a description of the dataset, and Support Vector Machines include points from the dataset as the "support vectors." Clearly, these models can be said to memorize at least part of their training data.

Commonly, however, memorization is an implicit, unintended side effect. In a striking recent work, Carlini et al. [13] demonstrate that modern models for next word prediction memorize large chunks of text from the training data verbatim, including personally identifiable and sensitive information such as phone numbers and addresses. Memorization of training data points by deep neural networks has also been observed in synthetic problems [29, 36]. The causes of this behavior are of interest to the foundations of both machine learning and privacy. For example, a model accidentally memorizing Social Security numbers from a text data set presents a glaring opportunity for identity theft.

In this paper, we aim to understand when this sort of memorization is unavoidable. We give natural prediction problems in which *every reasonably accurate training algorithm must encode, in the prediction model, nearly all the information about a large subset of its training examples*. Importantly, this holds even when most of that information is ultimately irrelevant to the task at hand. We

[*] Extended abstract. Full version available as [9].

STOC '21, June 21–25, 2021, Virtual, Italy
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8053-9/21/06…$15.00
https://doi.org/10.1145/3406325.3451131

show this for two types of tasks: a next-symbol prediction task (intended to abstract language modeling tasks) and a multiclass classification problem in which each class distribution is a simple product distribution in $\{0,1\}^d$ (intended to abstract a range of tasks like image labeling). Our results hold for any algorithm, regardless of its structure. We prove our statements by deriving new lower bounds on the information complexity of learning, building on the formalism of Bassily, Moran, Nachum, Shafer, and Yehudayoff [5].

We note that the word "memorization" is commonly used in the literature to refer to the phenomenon of *label memorization*, in which a learning algorithm fits arbitrarily chosen (or noisy) labels of training data points. Such memorization is a well-documented property of modern deep learning and is related to interpolation (or perfect fitting of all the training labels) [3, 25, 35, 37]. Feldman [18] recently showed that, for some problems, label memorization is *necessary* for achieving near-optimal accuracy on test data. Further, Feldman and Zhang [20] empirically demonstrate the importance of label memorization for deep learning algorithms on standard image classification datasets. In contrast, we study settings in which most of the information about entire high-dimensional (and high-entropy) training examples must be encoded by near-optimal learning algorithms.

Problem setting. We define a *problem instance p* as a distribution over labeled examples: $p \in \Delta(\mathcal{X})$, where $\mathcal{X} = \mathcal{Z} \times \mathcal{Y}$ is a space of examples (in $\mathcal{Z}$) paired with labels (in $\mathcal{Y}$). A dataset $X \in \mathcal{X}^n$ is generated by sampling i.i.d. from such a distribution. We use d to denote the dimension of the data, so X can be described in $\Theta(nd)$ bits. In contrast to the well-known PAC model of learning, we do not explicitly consider a concept class of functions. Rather, the instance p is itself drawn from a metadistribution q, dubbed the *learning task*. The learning task q is assumed to be known to the learner, but the specific problem instance is a priori unknown. We write P to denote a random instance (so P is a random variable, distributed according to q), and p to denote a particular realization. See Figure 1.

The learning algorithm A receives a sample $X \sim P^{\otimes n}$ and produces a model $M = A(X)$ that can be interpreted as a (possibly randomized) map $M : \mathcal{Z} \to \mathcal{Y}$. The model errs on a test data point (z, y) if $M(z) \neq y$ (for simplicity, we only consider misclassification error). The learner A's overall error on task q with sample size n, denoted $\mathrm{err}_{q,n}(A)$, is its expected error over P drawn from q, X drawn from $P^{\otimes n}$, and test point (Z, Y) drawn from P. That is,

$$\mathrm{err}_{q,n}(A) \overset{\text{def}}{=} \Pr_{\substack{P\sim q,\\ X\sim P^{\otimes n},(Z,Y)\sim P,\\ \text{coins of } A,M}} (M(Z) \neq Y \text{ where } M = A(X)) \qquad (1)$$

For probability calculations, we often use the shorthand "A errs" to denote a misclassification by $A(X)$ (the event above), so that $\Pr(A \text{ errs}) = \mathrm{err}_{q,n}(A)$.

Example 1.1. Consider the task of labeling the components in a mixture of N product distributions on the hypercube. An interesting special case is a uniform mixture of *uniform distributions over subcubes.* Here each component $j \in [N]$ of the mixture is specified by a sparse set of fixed indices $\mathcal{I}_j \subseteq [d]$ with values $\{b_j(i)\}_{i\in\mathcal{I}_j}$. Each labeled example is generated by picking at random a component $j \in [N]$ (which also serves as the label), for each $i \in \mathcal{I}_j$ setting $z(i) = b_j(i)$, and picking the other entries uniformly at random to obtain a feature vector $z \in \{0,1\}^d$. The labeled example is then (z, j). (See Figure 2.) A natural meta-distribution q generates each set $\mathcal{I}_j$ by adding indices i to $\mathcal{I}_j$ independently with some probability ρ, and fixes the values $b_j(i)$ at those indices uniformly at random.

Given a set of n labeled examples and a test point z' (drawn from the same distribution, but missing its label), the learner's job is to infer the label of the mixture component which generated z'. ■

Given q, a particular meta-distribution, and n, the number of samples in the data set, there exists a learner A_{OPT} (called *Bayes-optimal*) that minimizes the overall error on the task q. For any given task, this minimal error will be our reference; for $\epsilon \geq 0$, we call a learner *ϵ-suboptimal* (for q and n) if its error is within ϵ of that of A_{OPT} on samples of size n, that is, $\mathrm{err}_{q,n}(A) \leq \mathrm{err}_{q,n}(A_{OPT}) + \epsilon$. In our problem of cluster labeling (Example 1.1), the optimal learner works roughly as follows: for each component j of the mixture, produce a set $\hat{\mathcal{I}}_j$ of features that are fixed in the samples from that component (indices which take two values in different samples are irrelevant to classification). We have $\mathcal{I}_j \subseteq \hat{\mathcal{I}}_j$, but with few samples from cluster j, $\hat{\mathcal{I}}_j$ will contain many irrelevant indices. The optimal learner will balance the Hamming distance on $\hat{\mathcal{I}}_j$ against the probability of achieving a set of fixed indices of that size, accounting for the fact that we expect half of the non-fixed indices to match. In our analysis, it will suffice to analyze the simpler, but still low-error, learner that compares Hamming distances to a single sample from each component.

1.1 Our Contributions

We present natural prediction problems q where any algorithm with near-optimal accuracy on q must memorize $\Omega(nd)$ bits about the sample. Furthermore, this memorized information is really about the specific sample X and not about the data distribution P.

Theorem 1.1 (Informal)**.** *For all n and d, there exist natural tasks q for which any algorithm A that satisifes, for small constant ϵ,*

$$\mathrm{err}_{q,n}(A) \leq \mathrm{err}_{q,n}(A_{OPT}) + \epsilon$$

also satisfies

$$I(X; M \mid P) = \Omega(nd)\,,$$

where $P \sim q$ is the distribution on labeled examples, $X \sim P^{\otimes n}$ is a sample of size n from P, $M = A(X)$ is the model, and examples lie in $\{0,1\}^d$ (so $H(X) \leq nd$). The asymptotic expression holds for any sequence of n, d pairs; the constant depends only on ϵ.

To interpret this result, recall that conditional mutual information is defined via two conditional entropy terms: $I(X; M \mid P) = H(X \mid P) - H(X \mid M, P)$. Consider an informed observer who knows the full data distribution P (but not X). The term $I(X; M|P)$ captures how the observer's uncertainty about X is reduced after observing the model M. Since P is a full description of the problem instance, the term $H(X \mid P)$ captures the uncertainty about what is "unique to the data set," such as noise or irrelevant features. So $I(X; M \mid P) = \Omega(nd)$ means that *not only must the learning algorithm encode a constant fraction of the information it receives, but also that a constant fraction of what it encodes is irrelevant to the task.* For one of the problems we consider, we even get that $I(X; M \mid P) = (1 - o(1))H(X' \mid P)$, where X' is a subset of X of

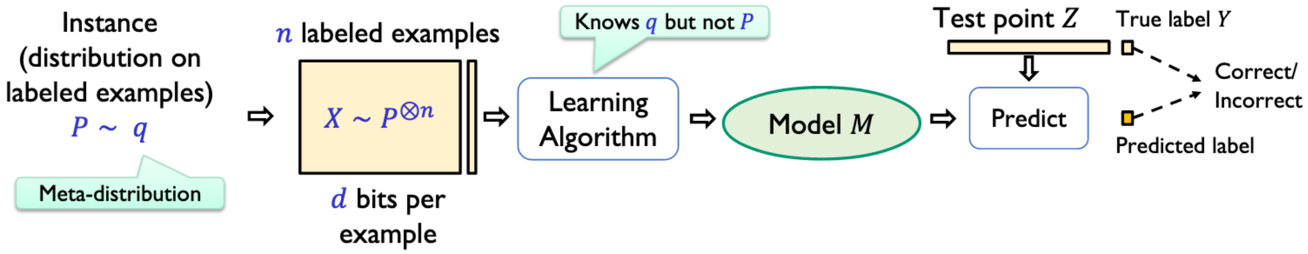

Figure 1: Problem setting. We aim to understand the information about the data X that is encoded in the model description M.

expected size $\Omega(n)$ and entropy $H(X' \mid P) = \Omega(nd)$ (see Theorem 1.2). That is, a subset of examples is encoded nearly completely in the model.

The meta-distribution q captures the learner’s initial uncertainty about the problem, and is essential to the result: if the exact distribution P were known to the learner A, it could simply ignore X and write down an optimal classifier for P as its model M. In that case, we would have $I(X; M \mid P) = 0$. That said, since conditional information is an expectation over realizations of P, our result also means that for every learner, there is a particular worst-case p (in the support of q) such that $I(M; X)$ is large. Such worst-case bounds were considered in a series of related papers [5, 27, 28], with which we compare below.

Our results lower bound mutual information. The statements do not directly shed light on whether a computationally efficient attacker, given access to the classifier, could recover some or all of the training data. Our proofs do suggest limited forms of recovery for some adversaries, but we leave the investigation of efficient recovery, and attacks against specific learning algorithms, as areas for future research.

We study two classes of learning tasks. The first is a next-symbol prediction problem (intended to abstract language modeling tasks). The second is the cluster labeling problem, partially introduced in Example 1.1, where individual classes are mixtures of product distributions over the Boolean hypercube. The exact problems are defined in Section 1.2.

In all the tasks we consider, data are drawn from a mixture of subpopulations. We consider settings where there are likely to be $\Omega(n)$ components of the mixture distribution from which the data set contains exactly one example. Leveraging new communication complexity bounds, we show that $\Omega(d)$ bits about most of these “singleton” examples must be encoded in M for the algorithm to perform well on average.

Returning to the cluster labeling problem in Example 1.1, recall that the learner receives an nd-bit data set, which has entropy $\Theta(nd)$, even conditioned on P (when ρ, the probability of fixing an index, is bounded away from 1). This “remaining uncertainty” $H(X \mid P)$ is, ignoring lower-order terms, exactly the uncertainty about the values of the irrelevant features. Showing $I(X; M \mid P) = \Omega(nd)$, then, establishes not only that the model must contain a large amount of information about X, but also that it must encode a large amount of information about the unfixed features, information completely irrelevant to the classification task at hand.

On a technical level, our results are related to those of Bassily, Moran, Nachum, Shafer, and Yehudayoff [5], [27] and Nachum and Yehudayoff [28], who study lower bounds on the mutual information $I(X; M)$ achievable by a PAC learner for a given class of Boolean functions $\mathcal{H}$. Specifically, for the class $\mathcal{H}_{thresh}$ of threshold functions on $[2^d]$, they give a learning task[1] for which every *proper and consistent* learning algorithm (i.e., one that is limited to outputting a function in $\mathcal{H}_{thresh}$ that labels the training data perfectly) satisfies $I(X; M \mid P) = \Omega(\log d)$ [5, 27]. Furthermore, Nachum et al. [27] extend this result to provide a hypothesis class $\mathcal{H}$ with VC dimension n over the input space $[n] \times \{0,1\}^d$ such that learners receiving $\Omega(n)$ samples must leak at least $I(X; M \mid P) = \Omega(n \cdot \log(d - \log n))$ bits about the input via their message. The direct sum construction in [27] is similar to our construction: they build a learning problem out of a product of simpler problems and relate the difficulty of the overall problem to that of the components.

Even more closely related is concurrent work of Livni and Moran [24, Theorem 2, setting $m = 2$], which gives settings in which the PAC-Bayes framework cannot yield good generalization bounds. Their result implies that sufficiently accurate algorithms for learning thresholds over $[2^d]$ must leak $\Omega(\log\log d)$ bits of information. (This can be extended to a lower bound of $\Omega(n \log\log d)$ for learning products of thresholds from samples of size n.) It is unclear if those techniques can yield bounds that scale linearly with d.

As we show in the full version of the paper [9], our results on next-symbol prediction can be cast in terms of learning threshold functions. As such, our results provide an alternative to those of [6], [27], and [24]. First, they are quantitatively stronger: we give a lower bound of $(1-o(1))nd$ rather than $\Omega(n \log d)$ or $\Omega(n \log\log d)$. Second, our bounds and those of [24] apply to all sufficiently accurate learners, whereas those of [6] and [27] require the learner to be proper and consistent (an incomparable assumption, in the regimes of interest).

Implications. While the problems we describe are intentionally simplified to allow for clean theoretical analysis, they rely on properties found in natural learning problems such as clustering of examples, noise, and a fine-grained subpopulation structure [38]. Our results thus suggest that memorization of irrelevant information, often observed in practice, is a fundamental property of learning and not an artifact of particular deep learning techniques.

[1] The results of Bassily et al. [5], Nachum et al. [27], Nachum and Yehudayoff [28] are formulated in terms of worst-case information leakage over a class of problems. They imply the existence of a single hard meta-distribution q by a minimax argument.

Our proofs rely on an assumption of independence between subpopulations. While this is a natural assumption for mixture models broadly, it is a significant simplification for a model of natural language or images. We believe that one could prove weaker but still meaningful statements about memorization under relaxed versions of the independence assumption. The crucial ingredients are that samples contain significant information about their subpopulation alongside irrelevant information and that the learning algorithm is unable to discern which is which. Independent subpopulations make for easier proofs and cleaner statements, but do not seem to be a requirement for memorization.

Our results have implications for learning algorithms that (implicitly) limit memorization. One class of such algorithms aims to compress models (for example to reduce memory usage), since description length upper bounds the mutual information. Differentially private algorithms [16] form another such class. It is known that differential privacy implies a bound on the mutual information $I(X; M \mid P)$ [10, 15, 26, 31]. Our results imply that such algorithms might not be able to achieve the same accuracy as unrestricted algorithms. In itself, that is nothing new: there is a long line of work on differentially private learning [for example 8, 23], including a number of striking separations from nonprivate algorithms [2, 6, 11]. There are also well established attacks on statistical and learning algorithms for high-dimensional problems (starting with [14]; see [17] for a survey of the theory, and a recent line of work on membership inference attacks [32] for empirical results). However, our results show a novel aspect of the limits of private learning: in the settings we consider, *successful learners must memorize exactly those parts of the data that are most likely to be sensitive*—unique samples from small subpopulations, including their peculiar details (modeled here as noisy or irrelevant features).

Variations on the main result. Different learning tasks exhibit variations and refinements of this central result. The mutual information lower bound implies that the model itself must be large, occupying at least $\Omega(nd)$ bits. But for some tasks we present, there exist ϵ-suboptimal models needing only $O(n \log(n/\epsilon) \log d)$ bits to write down (in the parameter regime we consider, where the problem scales with n). That is, with n samples the learning algorithm must output a model exponentially larger than what is required with sufficient data (or exact knowledge of P). In particular, for a given target accuracy level, there is a gradual drop in the size of the model, and the information specific to X, that is necessary (starting at $\Theta(n_0 d)$ where n_0 is the minimal sample size needed for that accuracy, and tending to $O(n_0 \log n_0 \log d)$ as the sample size n grows).

Another variation of our results gives a qualitatively stronger lower bound. For some tasks, we are able to demonstrate that *entire* samples must be memorized, in the following sense:

Theorem 1.2 (Informal)**.** *There exist natural tasks q for which every data set X has a subset of records X_S such that*

- *$\mathbb{E}[|X_S|] = \Omega(n)$ and $H(X_S \mid P) = \Omega(nd)$, and*
- *any algorithm A that satisifes $\mathrm{err}_{q,n}(A) \leq \mathrm{err}_{q,n}(A_{OPT}) + \epsilon$ also satisfies*

$$I(X_S; M \mid P) = (1 - o(1))H(X_S \mid P).$$

This statement implies $I(X; M \mid P) = \Omega(nd)$, but is a qualitatively different statement: as the learning algorithm's accuracy approaches optimal, it must reveal *everything* about these examples in X_S, at least information-theoretically. For these tasks, there is no costless compression the learning algorithm can apply: any reduction will increase the achievable error.

Finally, in addition to the memorization of noise or irrelevant features, in some settings we show how near-optimal models may be forced to memorize examples that are themselves entirely "useless," i.e. could be ignored with only a negligible loss in accuracy. That is, not only must irrelevant details of useful examples be memorized, but one must also memorize examples that come from very low-probability parts of the distribution. Unlike our main results, which hold for uniform mixtures over the subpopulations, this behavior relies on a particular type of mixture structure and the long-tailed distribution setup of [18]. We explain the concept and the statement more carefully in the full version [9].

1.2 Techniques: Subpopulations, Singletons, and Information Complexity

The learning tasks q we consider share a basic structure: each distribution P consists of a mixture, with coefficients $D \in \Delta([N])$, over subpopulations $j \in [N]$, each with a different "component distribution" C_j over labeled examples. The mixture coefficients D may be deterministic (e.g. uniform) or random; for this extended abstract, we focus on the uniform mixture setting, with $N = n$ (so the number of subpopulations is the same as the sample size). The C_j's are themselves sampled i.i.d. from a meta-distribution q_c.

As in [18], we look at how the learning algorithm behaves on the subset of examples that are *singletons*, that is, sole representatives (in X) of their subpopulation. For any data set X, let $X_S \subseteq X$ denote the subset of singletons. We consider mixture weights D where X_S has size $\Omega(n)$ with high probability. We show that for our tasks, a successful learner must roughly satisfy $I(X_S; M \mid P) = \Omega(d|X_S|)$, where d is the dimension of the data. Our results rely on the learning algorithm doing almost as well as possible with the size-n sample they are given. That requires us to adapt the distribution to n. For any fixed distribution we consider, if the sample is large enough, our proofs will yield weaker guarantees. For instance, if instead of $n = N$ samples from the uniform mixture over subpopulations, we draw $n = 2N$, then we will get fewer singletons, although we still expect $|X_S| = \Omega(n)$. If we increase the sample size to $n = \Omega(N \log N)$, with high probability the data set will contain no singletons.

One-Way Information Complexity of Singletons. We show that a good learner implies a good strategy for a related one-way communication game, the "singletons-only" game. In this game, nature generates k distributions $C_1, \ldots, C_k$, i.i.d. from the meta-distribution on clusters q_c, along with a uniformly random index $j^* \in [k]$. One player, Alice, receives a list $(x_1, \ldots, x_k)$ of labeled examples, where $x_j \sim C_j$. A second player Bob, receives only the feature vector z from a fresh draw $(z, y) \sim C_{j^*}$. Alice sends a single message M to Bob, who predicts a label $\hat{y}$. Alice and Bob win if $\hat{y} = y$.

Example 1.2 (Nearest neighbor, Figure 2)**.** For the hypercube task corresponding to Example 1.1, let q_{HC} be the distribution from which the $\{C_j\}$ are sampled. In the k-sample singletons-only game

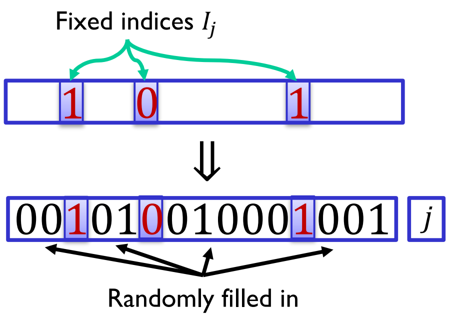

Figure 2: In hypercube labeling, each subpopulation j is associated with a sparse set of fixed indices. The example is generated by filling in the other indices randomly. The label is j.

for q_{HC}, there are k sets of fixed indices $\{(\mathcal{I}_j, b_j)\}_{j\in[k]}$. Alice gets a list $X' = (x_1, \dots, x_k) \in (\{0,1\}^d)^k$, where for every example j, we have: $\forall i \in \mathcal{I}_j, x_j(i) = b_j(i)$ and $\forall i \notin \mathcal{I}_j, x_j(i) = \text{Bernoulli}(1/2)$. The label, j, is implicit in the ordered list. Bob receives z for a random index j^* and must predict j^*.

Equivalently, we may view the game as a version of the nearest neighbor problem, treating Alice's input list $(x_1, \dots, x_k)$ as uniformly random in $(\{0,1\}^d)^k$, and Bob's input as a corrupted version of the one of the x_j's. If each $\mathcal{I}_j$ is built by adding every index independently with probability ρ, one can quickly check that generating z from the same distribution as x_j is equivalent to setting $z = BSC_{\frac{1-\rho}{2}}(x_{j^*})$, where $BSC_{\frac{1-\rho}{2}}$ is the binary symmetric channel that flips each bit of x_{j^*} independently with probability $\frac{1-\rho}{2}$. If Bob were to see Alice's entire input, his best strategy would be to guess index of the point in $(x_1, ..., x_k)$ that is nearest to z. One can show that he succeeds with high probability as long as $\rho \geq c\sqrt{\frac{\ln k}{d}}$ for $c > \sqrt{2}$.

This straightforward strategy requires Alice to send nd bits. We ask: can Bob still succeed with high probability when Alice sends $o(nd)$ bits? ■

One novel technical result bounds the information complexity of this nearest neighbor problem.

Lemma 1.3 (Informal; see Lemma 4.1). *For all $k, d \in \mathbb{N}, c > \sqrt{2}, \rho = c\sqrt{\frac{\ln k}{d}}$, and ϵ_k sufficiently small, the one-way information complexity of ϵ_k-suboptimal protocols for the k-sample singletons-only hypercube labeling task (Example 1.2) is $I(X'; M) \geq \frac{1-2h(\epsilon_k)}{c^2 \ln 2} \cdot kd$, where h is the binary entropy function.*

We prove this using the strong data processing inequality, directly analogous to its recent use for bounding the one-way information complexity of the Gap-Hamming problem [22].

The proof of this result is subtle, and does not proceed by separately bounding the information complexity of solving each of the k subproblems implicit in the singletons-only task. The parameter ρ is large enough that one can reliably detect proximity to any *one* of Alice's inputs with a message of size $\frac{d}{\log k} = o(d)$. Our proof crucially uses the fact that Bob must select from among k possibilities. It shows that his optimal strategy is to detect proximity to each of Alice's inputs with failure probability $\approx 1/k$, with his total failure probability controlled by a union bound.

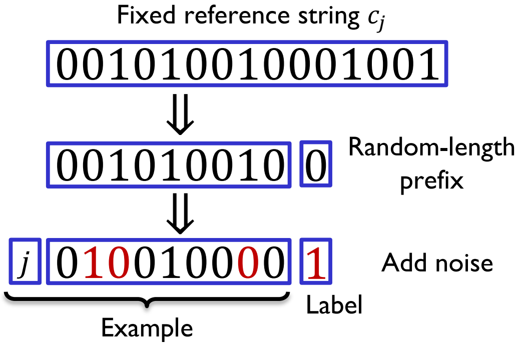

Figure 3: In next-symbol prediction, each subpopulation j is associated with a reference string. Examples contain j paired with a noisy prefix of random length. The label is the next bit, which may also be corrupted.

Next-bit Prediction and One-Shot Learning. Inspired by the empirical results of [12, 13], we demonstrate a sequence prediction task which requires memorization. Each subpopulation j is associated with a fixed "reference string," and samples from the subpopulation are noisy prefixes of this string.

Example 1.3 (Next-Symbol Prediction, Figure 3)**.** In the next-symbol prediction task the component distribution q_{NSP} draws a reference string $c_j \in \{0,1\}^d$ uniformly at random. Samples from j are generated by randomly picking a length $\ell \in \{0, \dots, d-1\}$, then generating $z \sim BSC_{\delta/2}(c_j(1:\ell))$ for some noise parameter $\delta \in [0,1)$. We pair z with a subpopulation identifier, so the example is (j, z). The label is a noisy version of the next bit: $y \sim BSC_{\delta/2}(c_j(\ell+1))$. ■

Unlike cluster problems, where the label is the subpopulation, each subpopulation can be treated independently by the learning algorithm. The core of our lower bound for this task, then, is to prove a "one-shot" lower bound on the setting where both Alice and Bob each receive a single example from the same subpopulation.

Lemma 1.4 (Informal; see Lemma 3.1)**.** *For sufficiently small ϵ, any algorithm that is ϵ-suboptimal on the (noiseless) one-shot next-symbol prediction task satisfies*

$$I(X; M) \geq \frac{d+1}{2}\left(1 - h\left(2\epsilon\right)\right). \tag{2}$$

Note that $\frac{d+1}{2}$ is the average length of Alice's input and that the $\log d$ term arises from uncertainty about that length, which Alice need not convey. The proof proceeds by establishing that Bob's correctness is tied to his ability to output Alice's relevant bit. For any fixed length of Alice's input, the problem is similar to a communication complexity problem called Augmented Indexing. We adapt the approach of a well-known elementary proof [4, 19].

1.3 Related Concepts

Our results are closely related to a number of other lines of work in machine learning. First, as discussed in the introduction, one can view our results as a significant strengthening of recent results on label memorization [18] and information-theoretic lower bounds for learning thresholds [5, 24, 27].

Representation Complexity. Another closely related concept is *probabilistic representation complexity* [7, 19]. For given error parameter ϵ, the representation complexity $\mathsf{PRep}_\epsilon(C)$ of a class C of concepts (functions from $\mathcal{Z}$ to $\mathcal{Y}$) is roughly the smallest number of bits needed to represent a hypothesis that approximates (up to expected error ϵ) a concept $c \in C$ on an example distribution $P_z \in \Delta(\mathcal{Z})$, in the worst case over pairs (c, P_z).[2] This complexity measure characterizes the sample complexity of "pure" differentially private learners for C [7].

Interpreted in our setting, representation complexity aims to understand the *length* of the message M, when the task q is a distribution over pairs (c, P_z) (that is, where the data distribution P consists of examples drawn from P_z and labeled with c). By a minimax argument, one can show that $\mathsf{PRep}_\epsilon(C)$ lower bounds not only M's length, but also the information it contains about P: one can find q such that $I(P; M)$ is at least $\mathsf{PRep}_\epsilon(C)$. This does imply that $I(X; M)$ must be large, but it says nothing about the information in M that is specific to a particular sample X: in fact, the bound is saturated by learners that get enough data to construct a hypothesis that is just a function of P, so that $I(X; M \mid P)$ is small.

The bounds we prove here are qualitatively stronger. We give settings where the analogue of representation complexity is small (namely, a learner that knows P can construct a model of size about $n \log(n/\epsilon) \log d$), but where a learner which only gets a training sample must write down a very large model ($\Omega(nd)$ bits) to do well.

Time-Space Tradeoffs for Learning. A recent line of work establishes time-space tradeoffs for learning: problems where any learning algorithm requires either a large memory or a large number of samples (see [21] for a summary of results). The prime example is parity learning over d bits, which is shown to require either $\Omega(d^2)$ bits of memory or exponentially many samples. The straightforward algorithm for parity learning requires $O(d)$ samples, so this result shows that any feasible algorithm must store, up to constant factors, as many bits as are required to store the dataset [30].

Our work sets a specific number of samples under which learning is feasible and, for that number of samples, establishes an information lower bound on the *output* of the algorithm. This implies not only a communication lower bound but also one on memory usage: the algorithm must store the model immediately prior to releasing it. Some of our tasks exhibit the property that, with additional data, an algorithm can output a substantially smaller model. These learning tasks might exhibit a time-space tradeoff, although not one as dramatic as the requirement of exponentially many samples. Intuitively, the underlying concept in parity learning must be learned "all at once." Our problem instances can be learned "piece-by-piece," as the algorithm learns sections independently of the rest of the sample.

Information Bottlenecks. Our work fits into the broad category of information bottleneck results in information theory [33]. An information bottleneck is a compression scheme for extracting from a random variable V all the information relevant for the estimation of another random variable W while discarding all irrelevant information in V. In our setting, one may take $V = X$ to be the data set, and W to be the true distribution P (where the loss of a model is its misclassification error). This form of information bottleneck was recently described in general terms [1]. Our results lower bound the extent to which nontrivial compression is possible, showing that the Markov chain $P - X - M$ must in particular satisfy $I(X; M) \gg I(M; P)$.

Information bottlenecks have been put forward as a theory of how implicit feature representations evolve during training [34]. That line of work studies how the prediction process transforms information from a test datum during prediction (i.e. as one moves through layers of a neural network), and is thus distinct from our study of how learning algorithms are able to extract information from training data sets.

1.4 Organization of This Extended Abstract

In Section 2, we state and discuss the main reduction in the paper, connecting the learning task to the associated communication game. In Sections 3 and 4 we state and prove the information complexity lower bounds for our two types of learning problems. Appendix A contains additional details related to the reduction lemma.

The full version of the paper [9] describes the general setup (beyond uniform mixtures) and presents detailed theorems and the additional necessary analysis of the learning tasks. It presents additional results, including lower bounds for threshold learning and implications for differentially-private algorithms.

2 CENTRAL REDUCTION

To connect our information complexity lower bounds to the machine learning setting, where data are drawn i.i.d., we show that any learning algorithm for the standard setting implies a sequence of protocols for the task-specific communication games (one for each possible number of singletons). If the learning algorithm is near-optimal, then this sequence must also be near-optimal on average. For simplicity we focus on the setting of a uniform mixture over subpopulations and set the number of samples equal to the number of subpopulations, so $n = N$. In this setting, we expect approximately $\frac{n}{e}$ singletons, and with high probability will receive at least $\frac{n}{3}$ of them. The proof of the reduction, and the more general non-uniform statement, can be found in the full version of the paper [9].

Before stating the result, let us give names to the standard learning task and the singletons-only communication game.

Definition 2.1. We call our standard learning task $\mathsf{Learn}(n, q_c)$. A problem instance P is generated by independently drawing, for each subpopulation $j \in [N]$, a component distribution $C_j \sim q_c$. The data set of n i.i.d. samples is generated by, for each sample, picking a subpopulation j uniformly at random and sampling $(z_i, y_i) \sim C_j$. One test sample (z, y) is drawn independently from the same process, and the model predicts a label based on z. ■

[2] See [19] for an exact definition.

Definition 2.2. We denote by $\mathrm{Singletons}(k, q_c)$ the singletons-only communication game. In this task, k component distributions C_j are sampled i.i.d. from q_c. Alice receives a tuple of k data points $((z_1, y_1), \ldots, (z_k, y_k))$, where for each j we draw $(z_j, y_j) \sim C_j$. One test sample is drawn by picking a j^* uniformly at random and sampling $(z, y) \sim C_{j^*}$, and Bob predicts a label based on z alone. ■

Now let us state the reduction. Recall that for each task there is a Bayes-optimal learner that minimizes misclassification error and that we say an algorithm is ϵ-suboptimal if its error is within ϵ of the optimal error.

Lemma 2.1 (Central Reduction, Uniform Setting)**.** *Suppose we have the following lower bound for every k: any algorithm A^k that is ϵ_k-suboptimal for* $\mathrm{Singletons}(k, q_c)$ *satisfies*

$$I(X'; A^k(X')) \geq k \cdot f(\epsilon_k),$$

for some convex and non-increasing function $f(\cdot)$. Then for any algorithm A that is ϵ-suboptimal on $\mathrm{Learn}(n, q_c)$*,*

$$I(X_S; M \mid K) \geq \frac{n}{3} \cdot f\Big(3\epsilon + \phi_1(q_c, A) + \phi_2(q_c)\Big),$$

where X_S is the subset of singletons and $\phi_1(q_c, A)$ and $\phi_2(q_c)$ are task-specific error terms defined in Appendix A.

Informally, the error terms quantify by how much an algorithm can beat the optimal error on a subset of the probability space by focusing on that setting. For example, ϕ_1 characterizes this quantity for the event that the test sample is not from a singleton subpopulation. In the full version of the paper [9], we show that $\phi_1(q_{HC}, A)$ and $\phi_2(q_{HC})$, for the hypercube labeling task, are both $O(n^{-\alpha})$ for some $\alpha > 0$ and any algorithm A. Similarly we show for next-symbol prediction that $\phi_1(q_{NSP}, A), \phi_2(q_{NSP}) \leq 0$ for all algorithms A.

As we saw above, the functions $f(\cdot)$ that show up in our lower bounds are simple, depending on ϵ through a term like $1 - h(\epsilon)$, where $h(\cdot)$ is the binary entropy function.

3 NEXT-SYMBOL PREDICTION

We now present the information complexity lower bound for Next-Symbol Prediction. Recall from Example 1.3 that each data point comes with a subpopulation identifier. This fact allows us to focus on the "one-shot" communication game, where Alice receives a single example. The lower bound for $\mathrm{Singletons}(k, q_{NSP})$ is then exactly k times the lower bound for $\mathrm{Singletons}(1, q_{NSP})$.

The core of this proof is based on an existing lower bound [4, 19] for *Augmented Index*, a well-known communication complexity task.

Lemma 3.1. *Fix a noise level $\delta \in [0, 1)$. For any $\epsilon_1 < \frac{(1-\delta)^2}{4}$, any learning algorithm A that is ϵ_1-suboptimal on* $\mathrm{Singletons}(1, q_{NSP})$ *satisfies*

$$I(X; M) \geq \frac{d+1}{2}\left(1 - h\left(\frac{2\epsilon_1}{(1-\delta)^2}\right)\right).$$

Proof. Let random variable L_A be the length of Alice's input. Since we know $H(X) = H(X, L_A) = H(L_A) + H(X \mid L_A) = \log d + \frac{d+1}{2}$, to lower bound the mutual information we must provide an upper bound on $H(X \mid M)$.

Define G to be the "good event" that (i) Alice's input is at least as long as Bob's and (ii) the relevant bits were not rerandomized. G happens with probability $\frac{1}{2} \cdot \frac{d+1}{d}(1-\delta)^2$. Conditioned on G, the optimal algorithm is correct and, conditioned on $\bar{G}$, *any* algorithm has accuracy $\frac{1}{2}$. The main idea of the proof is that, conditioned on G, "correctness" and "outputting Alice's data" are the same event.

We change the additive error ϵ_1 into a multiplicative error γ. Let $\gamma \overset{\text{def}}{=} \Pr[A \text{ errs} \mid G]$. We can write

$$\begin{aligned}\Pr[A \text{ errs}] &= \Pr[A \text{ errs} \mid G]\Pr[G] + \Pr[A \text{ errs} \mid \bar{G}](1 - \Pr[G]) \\ &= \frac{1}{2} - \frac{\Pr[G]}{2}(1 - 2\gamma),\end{aligned}$$

and $\Pr[A_{OPT} \text{ errs}] = \frac{1}{2} - \frac{\Pr[G]}{2}$. By the definition of suboptimality we have $\epsilon_1 = \Pr[A \text{ errs}] - \Pr[A_{OPT} \text{ errs}] = \Pr[G] \cdot \gamma$. Since $\Pr[G] \geq \frac{1}{2} \cdot (1-\delta)^2$, we have $\gamma \leq \frac{2\epsilon_1}{(1-\delta)^2}$, which implies $\gamma \leq \frac{1}{2}$.

Let random variables X and Z denote Alice and Bob's inputs, respectively, and write L_A and L_B for the lengths of their inputs. Since L_A is fixed given X, we can apply the chain rule for entropy and bound

$$\begin{aligned}H(X \mid M) = H(X, L_A \mid M) &= H(X \mid M, L_A) + H(L_A \mid M) \\ &\leq \mathop{\mathbb{E}}_{\ell}[H(X \mid M, L_A = \ell)] + \log d.\end{aligned}$$

We will fix ℓ and bound $H(X \mid M, L_A = \ell)$. Define γ_ℓ to be $\Pr[A \text{ errs} \mid G, L_A = \ell]$. Assume $\gamma_\ell \leq \frac{1}{2}$ without loss of generality; for any algorithm with a $\gamma_\ell > \frac{1}{2}$ there is one with the same information cost that achieves lower error by reversing the decision.

Recall that G implies neither Alice nor Bob's $L_B + 1$-th bits are rerandomized, so Bob is correct if and only if he outputs Alice's $L_B + 1$-th bit. We now show that, if Bob can output Alice's bits, her message must contain a lot of information about her input. Crucially, conditioned on $L_A = \ell$, the good event G is independent of Alice's input X, since $L_B \perp L_A$ and Alice's string is uniformly random whether or not any bit is flipped. So $H(X \mid M, L_A = \ell) = H(X \mid M, G, L_A = \ell)$. Below, in (3), we apply the chain rule for entropy over Alice's bits, including her label. In (4) we replace $X_1^{\ell_B}$ with $Z_1^{\ell_B}$, which is a noisy version and thus can only increase uncertainty about X_{ℓ_B+1}.

$$H(X \mid M, G, L_A = \ell) = \sum_{\ell_B=0}^{\ell} H(X_{\ell_B+1} \mid X_1^{\ell_B}, M, G, L_A = \ell) \tag{3}$$

$$\leq \sum_{\ell_B=0}^{\ell} H(X_{\ell_B+1} \mid Z_1^{\ell_B}, M, G, L_A = \ell). \tag{4}$$

Now we relate the index ℓ_B to the random variable L_B, the length of Bob's input, and observe that $\Pr[L_B = \ell_B \mid G, L_A = \ell] = \frac{1}{\ell+1}$, since event G requires that $L_A \geq L_B$. So we have

$$\begin{aligned}H(X \mid M, L_A = \ell) &\leq (\ell+1)\sum_{\ell_B=0}^{\ell-1}\big(\Pr[L_B = \ell_B \mid G, L_A = \ell] \\ &\qquad \times H(X_{\ell_B+1} \mid Z_1^{\ell_B}, M, G, L_A = \ell)\big) \\ &= (\ell+1) \cdot H(X_{L_B+1} \mid Z, M, G, L_A = \ell) \\ &\leq (\ell+1) \cdot h(\gamma_\ell),\end{aligned}$$

applying Fano's inequality and using the assumption that $\gamma_\ell \le \frac{1}{2}$, since this is exactly Bob's task and, conditioned on G and $L_A = \ell$, he fails with probability at most γ_ℓ.

By rewriting the expectation, we can apply Jensen's inequality and push the expectation inside the binary entropy function, getting

$$\begin{aligned} H(X \mid M, L_A) &\le \mathbb{E}[\ell+1] \cdot h\left(\frac{\mathbb{E}[(\ell+1)\gamma_\ell]}{\mathbb{E}[\ell+1]}\right) \\ &= \frac{d+1}{2} \cdot h\left(\frac{2 \cdot \mathbb{E}[(\ell+1)\gamma_\ell]}{d+1}\right) \\ &= \frac{d+1}{2} \cdot h(\gamma). \end{aligned}$$

The last equality follows with a few lines of algebra from the facts that $\gamma \cdot \Pr[G] = \mathbb{E}_\ell\left[\gamma_\ell \Pr[G \mid L_A = \ell]\right]$, $\Pr[G] = \frac{d+1}{2d} \cdot (1-\delta)^2$, and $\Pr[G \mid L_A = \ell] = \frac{\ell+1}{d} \cdot (1-\delta)^2$. Using $\gamma \le \frac{2\epsilon_1}{(1-\delta)^2}$ gives us

$$H(X \mid M) \le \frac{d+1}{2} \cdot h\left(\frac{2\epsilon_1}{(1-\delta)^2}\right) + \log d$$

which, combined with $H(X) = \frac{d+1}{2} + \log d$, finishes the proof. □

4 HYPERCUBE LEARNING

We turn to the lower bound for Singletons(k, q_{HC}), the communication game associated with Hypercube Labeling. As discussed above in Example 1.2, Singletons(k, q_{HC}) is equivalent to the following communication game, which removes the need to analyze the set of fixed indices.

Definition 4.1 (Nearest of k Neighbors). Alice receives k strings $x_1, \dots, x_k \in \{0,1\}^d$, drawn i.i.d. from the uniform distribution. Bob receives a string $y \sim BSC_{\frac{1-\rho}{2}}(x_j)$ for some index $j \in [k]$, also chosen uniformly at random. They succeed if Bob outputs j. ■

Lemma 4.1. *Set $\rho = \frac{c\sqrt{\ln n}}{\sqrt{d}}$ for any $n > 1$ and constant $c > \sqrt{2}$. For every $k \le n$, any one-way communication protocol for Nearest of k Neighbors with error at most $\epsilon_k \le \frac{1}{10}$ satisfies*

$$I(X; M) \ge \frac{kd}{c^2 \ln 2} \cdot \frac{\log k}{\log n} \cdot (1 - 2h(\epsilon_k)).$$

Note that, because of the $\log k$ factor, this does not exactly match Lemma 2.1, which asks for a lower bound of the form $k \cdot f(\epsilon)$ for convex and non-increasing $f(\cdot)$. In the full version of this paper [9] we circumvent this issue by using the fact that the number of singletons will concentrate about its mean, making $\frac{\log k}{\log n}$ roughly constant.

The proof of Lemma 4.1 is adapted from one by Hadar et al. [22] and relies on the following strong data processing inequality for binary symmetric channels.

Lemma 4.2 (SDPI). *Suppose we have a Markov chain $M - X - Y$ where $X \sim \text{Uniform}(\{0,1\}^d)$ and $Y \sim \text{BSC}_{\frac{1-\rho}{2}}(X)$. Then $I(M;Y) \le \rho^2 I(M;X)$.*

Proof of Lemma 4.1. Since Bob, with access to M and test sample Z, can guess the index $J \in_R [k]$ with error ϵ_k, we have via Fano's inequality that

$$\begin{aligned} I(J; M, Z) &= H(J) - H(J \mid M, Z) \\ &\ge \log k - (\epsilon_k \log k + h(\epsilon_k)) \\ &\ge (1 - 2h(\epsilon_k)) \log k, \end{aligned} \tag{5}$$

since for all $\epsilon_k \le \frac{1}{2}$, $h(\epsilon_k) \ge \epsilon_k$. For this lower bound to be nontrivial, we require $h(\epsilon_k) < \frac{1}{2}$, which is satisfied by taking $\epsilon_k \le \frac{1}{10}$. We now upper bound $I(J; M, Z)$. Let P refer to the joint and marginal distributions defined by the learning task. We apply the "radius" property of mutual information and take Q to be the product of marginals over M and Z: $Q_{M,Z} = P_M \times P_Z$:

$$\begin{aligned} I(J; M, Z) &= \inf_{Q_{M,Z} \in \Delta((M,Z))} \mathop{\mathbb{E}}_j \left[D_{KL}\left(P_{M,Z|J=j} \| Q_{M,Z}\right)\right] \\ &\le \mathop{\mathbb{E}}_j \left[D_{KL}\left(P_{M,Z|J=j} \| P_M \times P_Z\right)\right]. \end{aligned}$$

Next, note that $M \perp J$ and $Z \perp J$, so we have

$$\begin{aligned} &\mathop{\mathbb{E}}_j \left[D_{KL}(P_{M,Z|J=j} \| P_M \times P_Z)\right] \\ &\quad = \mathop{\mathbb{E}}_j \left[D_{KL}\left(P_{M,Z|J=j} \| P_{M|J=j} \times P_{Z|J=j}\right)\right] \\ &\quad = \mathop{\mathbb{E}}_j \left[I(M; Z \mid J = j)\right]. \end{aligned}$$

Now we apply the SDPI. For any fixed j, M depends on the test sample Z only through data point X_j, since Z is a noisy version of X_j. We can marginalize out $\{X_i\}_{i \ne j}$ and apply Lemma 4.2:

$$\mathop{\mathbb{E}}_j \left[I(M; Z \mid J = j)\right] \le \mathop{\mathbb{E}}_j \left[\rho^2 I(M; X_j \mid J = j)\right].$$

But for any index i, the mutual information between M and X_i is independent of J; it depends only on Alice's protocol. So

$$\mathop{\mathbb{E}}_j \left[\rho^2 I(M; X_j \mid J = j)\right] = \mathop{\mathbb{E}}_i \left[\rho^2 I(M; X_i)\right]$$

and, combining these steps and writing out the expectation, we have

$$I(J; M, Z) \le \mathop{\mathbb{E}}_i \left[\rho^2 I(M; X_i)\right] = \frac{\rho^2}{k} \sum_{i=1}^{k} I(M; X_i). \tag{6}$$

Applying the chain rule for mutual information and the independence of the $\{X_i\}$, we get

$$\begin{aligned} \sum_i I(M; X_i) &= \sum_i H(X_i) - H(X_i \mid M) \\ &= \sum_i H(X_i \mid X_1^{i-1}) - H(X_i \mid M) \\ &\le \sum_i H(X_i \mid X_1^{i-1}) - H(X_i \mid M, X_1^{i-1}) \\ &= I(M; X). \end{aligned}$$

Therefore, combining Equations (5) and (6),

$$(1 - 2h(\epsilon_k)) \log k \le I(J; M, Z) \le \frac{\rho^2}{k} I(M; X).$$

Plugging in $\rho = \frac{c\sqrt{\ln n}}{\sqrt{d}}$ and changing the natural log to base 2, we see that $\frac{\rho^2}{k} = \frac{c^2 \ln 2 \cdot \log n}{kd}$. Rearranging, we get a lower bound on $I(X; M)$. □

ACKNOWLEDGMENTS

We thank Ankit Garg for helpful conversations about the information complexity of the Gap-Hamming problem. Gavin Brown and Adam Smith are supported in part by NSF award CCF-1763786 as well as a Sloan Foundation research award. Mark Bun is supported by NSF award CCF-1947889.

A TASK-SPECIFIC ERROR TERMS

Our Lemma 2.1 involves two error terms: $\phi_1(q_c, A)$ and $\phi_2(q_c)$. For "natural" algorithms these terms will be negative, but in general they may be positive. In the full version of the paper [9], we show that these terms for q_{HC} are bounded above by $n^{-\alpha}$ for some constant $\alpha > 0$ and are bounded above by 0 for q_{NSP}.

Recall A_{OPT}, the Bayes-optimal algorithm for Learn(n, q). Let E_1 be the event that the test sample comes from a "singleton subpopulation," that is, a subpopulation with exactly one representative in the data set. $\phi_1(q_c, A)$ captures how well A performs above optimal when the test sample comes from a non-singleton subpopulation:

$$\begin{aligned}\phi_1(q_c, A) \overset{\text{def}}{=} 3 \cdot \Pr[\bar{E}_1]\big(&\Pr[A_{OPT} \text{ errs on Learn}(n, q_c) \mid \bar{E}_1] \\ &- \Pr[A \text{ errs on Learn}(n, q_c) \mid \bar{E}_1]\big)\end{aligned}$$

The leading factor of 3 comes from the fact that we are working with the uniform mixture, and we deal with it in a slightly different manner in the full version [9].

For the second term, let random variable K be the number of singletons in the data set. Define the following difference term:

$$\begin{aligned}\Delta_k \overset{\text{def}}{=} &\Pr[A_{OPT} \text{ errs on Learn}(n, q_c) \mid E_1, K = k] \\ &- \inf_{A'} \Pr[A' \text{ errs on Singletons}(k, q_c)].\end{aligned}$$

Then $\phi_2(q_c)$ is defined as an average over k:

$$\phi_2(q_c) \overset{\text{def}}{=} 3 \sum_{k=1}^{N} \Pr[K = k \mid E_1] \cdot \Delta_k.$$

REFERENCES

[1] Alexander A Alemi. 2020. Variational predictive information bottleneck. In *Symposium on Advances in Approximate Bayesian Inference*. PMLR, 1–6.

[2] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. 2019. Private PAC learning implies finite Littlestone dimension. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*. 852–860.

[3] Devansh Arpit, Stanislaw Jastrzkebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 233–242.

[4] Ziv Bar-Yossef, Thathachar S Jayram, Ravi Kumar, and D Sivakumar. 2004. An information statistics approach to data stream and communication complexity. *J. Comput. System Sci.* 68, 4 (2004), 702–732.

[5] Raef Bassily, Shay Moran, Ido Nachum, Jonathan Shafer, and Amir Yehudayoff. 2018. Learners that use little information. In *Algorithmic Learning Theory*. PMLR, 25–55.

[6] Raef Bassily, Adam Smith, and Abhradeep Thakurta. 2014. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE, 464–473.

[7] Amos Beimel, Kobbi Nissim, and Uri Stemmer. 2019. Characterizing the Sample Complexity of Pure Private Learners. *Journal of Machine Learning Research* 20, 146 (2019), 1–33.

[8] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. 2005. Practical privacy: the SuLQ framework. In *Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 13-15, 2005, Baltimore, Maryland, USA*, Chen Li (Ed.). ACM, 128–138. https://doi.org/10.1145/1065167.1065184

[9] Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. 2020. When is Memorization of Irrelevant Training Data Necessary for High-Accuracy Learning? *arXiv preprint arXiv:2012.06421* (2020).

[10] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*. Springer, 635–658.

[11] Mark Bun, Jonathan Ullman, and Salil Vadhan. 2018. Fingerprinting codes and the price of approximate differential privacy. *SIAM J. Comput.* 47, 5 (2018), 1888–1938.

[12] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*. 267–284.

[13] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2020. Extracting Training Data from Large Language Models. (2020). arXiv:2012.07805 [cs.CR]

[14] Irit Dinur and Kobbi Nissim. 2003. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 202–210.

[15] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toni Pitassi, Omer Reingold, and Aaron Roth. 2015. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems*. 2350–2358.

[16] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings (Lecture Notes in Computer Science, Vol. 3876)*, Shai Halevi and Tal Rabin (Eds.). Springer, 265–284. https://doi.org/10.1007/11681878_14

[17] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. 2017. Exposed! a survey of attacks on private data. *Annual Review of Statistics and Its Application* 4 (2017), 61–84.

[18] Vitaly Feldman. 2020. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 954–959.

[19] Vitaly Feldman and David Xiao. 2014. Sample complexity bounds on differentially private learning via communication complexity. In *Conference on Learning Theory*. PMLR, 1000–1019.

[20] Vitaly Feldman and Chiyuan Zhang. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems* 33 (2020).

[21] Sumegha Garg, Ran Raz, and Avishay Tal. 2018. Extractor-based time-space lower bounds for learning. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. 990–1002.

[22] Uri Hadar, Jingbo Liu, Yury Polyanskiy, and Ofer Shayevitz. 2019. Communication complexity of estimating correlations. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. 792–803.

[23] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM J. Comput.* 40, 3 (2011), 793–826.

[24] Roi Livni and Shay Moran. 2020. A limitation of the pac-bayes framework. *Advances in Neural Information Processing Systems* 33 (2020).

[25] Siyuan Ma, Raef Bassily, and Mikhail Belkin. 2018. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In *International Conference on Machine Learning*. PMLR, 3325–3334.

[26] Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil P. Vadhan. 2010. The Limits of Two-Party Differential Privacy. In *FOCS*. 81–90.

[27] Ido Nachum, Jonathan Shafer, and Amir Yehudayoff. 2018. A direct sum result for the information complexity of learning. *arXiv preprint arXiv:1804.05474* (2018).

[28] Ido Nachum and Amir Yehudayoff. 2019. Average-case information complexity of learning. In *Algorithmic Learning Theory*. PMLR, 633–646.

[29] Adityanarayanan Radhakrishnan, Mikhail Belkin, and Caroline Uhler. 2019. Overparameterized neural networks can implement associative memory. *arXiv preprint arXiv:1909.12362* (2019).

[30] Ran Raz. 2018. Fast learning requires good memory: A time-space lower bound for parity learning. *Journal of the ACM (JACM)* 66, 1 (2018), 1–18.

[31] Ryan Rogers, Aaron Roth, Adam Smith, and Om Thakkar. 2016. Max-information, differential privacy, and post-selection hypothesis testing. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 487–494.

[32] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 3–18.

[33] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057* (2000).

[34] Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*. IEEE, 1–5.

[35] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. 2019. Small ReLU networks are powerful memorizers: a tight analysis of memorization capacity. In *Advances in*

Neural Information Processing Systems. 15558–15569.

[36] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Michael C Mozer, and Yoram Singer. 2019. Identity crisis: Memorization and generalization under extreme overparameterization. *arXiv preprint arXiv:1902.04698* (2019).

[37] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* (2016).

[38] Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. 2014. Capturing long-tail distributions of object subcategories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 915–922.