The Advantages of Multiple Classes for Reducing Overfitting from Test Set Reuse

Vitaly Feldman [1 2] **Roy Frostig** [1] **Moritz Hardt** [3 4]

Abstract

Excessive reuse of holdout data can lead to overfitting. However, there is little concrete evidence of significant overfitting due to holdout reuse in popular multiclass benchmarks today. Known results show that, in the worst-case, revealing the accuracy of k adaptively chosen classifiers on a data set of size n allows to create a classifier with bias of $\Theta(\sqrt{k/n})$ for any binary prediction problem. We show a new upper bound of $\tilde{O}(\max\{\sqrt{k\log(n)/(mn)}, k/n\})$ on the worst-case bias that any attack can achieve in a prediction problem with m classes. Moreover, we present an efficient attack that achieve a bias of $\Omega(\sqrt{k/(m^2 n)})$ and improves on previous work for the binary setting ($m = 2$). We also present an inefficient attack that achieves a bias of $\tilde{\Omega}(k/n)$. Complementing our theoretical work, we give new practical attacks to stress-test multiclass benchmarks by aiming to create as large a bias as possible with a given number of queries. Our experiments show that the additional uncertainty of prediction with a large number of classes indeed mitigates the effect of our best attacks.

1. Introduction

Several machine learning benchmarks have shown surprising longevity, such as the ILSVRC 2012 image classification benchmark based on the ImageNet database (Russakovsky et al., 2015). Even though the test set contains only 50,000 data points, hundreds of results have been reported on this test set. Large-scale hyperparameter tuning and experimental trials across numerous studies likely add thousands of queries to the test data. Despite this excessive data reuse, recent replication studies (Recht et al., 2018; 2019) have shown that the best performing models transfer rather gracefully to a newly collected test set collected from the same source according to the same protocol.

What matters is not only the number of times that a test (or holdout) set has been accessed, but also how it is accessed. Modern machine learning practice is *adaptive* in its nature. Prior information about a model's performance on the test set inevitably influences future modeling choices and hyperparameter settings. Adaptive behavior, in principle, can have a radical effect on generalization.

Standard concentration bounds teach us to expect a maximum error of $O(\sqrt{\log(k)/n})$ when estimating the means of k non-adaptively chosen bounded functions on a data set of size n. However, this upper bound sharply deteriorates to $O(\sqrt{k/n})$ for adaptively chosen functions, an exponential loss in k. Moreover, there exists a sequence of adaptively chosen functions, what we will call an *attack*, that causes an estimation error of $\Omega(\sqrt{k/n})$ (Dwork et al., 2014).

What this means is that in principle an analyst can overfit substantially to a test set with relatively few queries to the test set. Powerful results in *adaptive data analysis* provide sophisticated holdout mechanisms that guarantee better error bounds through noise addition (Dwork et al., 2015b) and limited feedback mechanisms (Blum & Hardt, 2015). However, the standard holdout method remains widely used in practice, ranging from machine learning benchmarks to industry validation sets, and data science competitions. If the pessimistic bound was indicative of performance in practice, the holdout method would likely be much less useful than it is.

It therefore seems evident that there are factors that prevent this worst-case overfitting from happening in practice. In this work, we isolate the number of classes in the prediction problem as one such factor that has an important effect on the amount of overfitting we expect to see. Indeed, we find that in the worst-case the number of queries required to achieve certain bias grows at least linearly with the number of classes, a phenomenon that we establish theoretically and experiment with extensively.

[1]Google Brain [2]Part of this work was done while the author was visiting the Simons Institute for the Theory of Computing. [3]University of California, Berkeley [4]Work done while at Google. Correspondence to: Vitaly Feldman <vitalyfm@google.com>, Roy Frostig <frostig@google.com>, Moritz Hardt <hardt@berkeley.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

1.1. Our Contributions

We study in both theory and experiment the effect that multiple classes have on the amount of overfitting caused by test set reuse. In doing so, we extend important developments for binary prediction to the case of multiclass prediction.

To state our results more formally, we introduce some notation. A classifier is a mapping $f\colon X \to Y$, where $Y = [m] = \{1, \ldots, m\}$ is a discrete set consisting of m classes and X is the data domain. A data set of size n is a tuple $S \in (X \times Y)^n$ consisting of n labeled examples $(x_i, y_i)_{i\in[n]}$, where we assume each point is drawn independently from a fixed underlying population. In our model, we assume that a data analyst can *query* the data set by specifying a classifier $f\colon X \to Y$ and observing its accuracy $\mathsf{acc}_S(f)$ on the data set S, which is simply the fraction of points that are correctly labeled $f(x_i) = y_i$. We denote by $\mathsf{acc}(f) = \Pr\{f(x) = y\}$ the accuracy of f over the underlying population from which (x, y) are drawn. Proceeding in k rounds, the analyst is allowed to specify a function in each round and observe its accuracy on the data set. The function chosen at a round t may depend on all previously revealed information. The analyst builds up a sequence of adaptively chosen functions $f_1, \ldots, f_k$ in this manner.

We are interested in the largest value that $\mathsf{acc}_S(f_t) - \mathsf{acc}(f_t)$ can attain over all $1 \le t \le k$. Our theoretical analysis focuses on the worst case setting where an analyst has no prior knowledge (or, equivalently, has a uniform prior) over the correct label of each point in the test set. In this setting, the highest expected accuracy achievable on the unknown distribution is $1/m$. In effect, we analyze the expected advantage of the analyst over random guesses.

In reality, an analyst typically has substantial prior knowledge about the labels and starts out with a far stronger classifier than one that predicts at random. Using domain information, models, and training data, there are many conceivable ways to label many points with high accuracy and to pare down the set of labels for points the remaining points. Indeed, our experiments explore a couple of techniques for reducing label uncertainty given a good baseline classifier. After incorporating all prior information, there is usually still a large set of points for which there remains high uncertainty over the correct label. Effectively, to translate the theoretical bounds to a practical context, it is useful to think of the dataset size n as the number of point that are hard to classify, and to think of the class count m as a number of (roughly equally likely) candidate labels for those points.

Our theoretical contributions are several upper and lower bounds on the achievable bias in terms of the number of queries k, the number of data points n, and the number of classes m. We first establish an upper bounds on the bias achievable by any attack in the uniform prior setting.

Theorem 1.1 (Informal)**.** *There is a distribution P over examples labeled by m classes such that any algorithm that makes at most k queries to a dataset $S \sim P^n$ must satisfy with high probability*

$$\max_{1\le t\le k} \mathsf{acc}_S(f_t) = \frac{1}{m} + O\left(\max\left\{\sqrt{\frac{k\log n}{nm}}, \frac{k\log n}{n}\right\}\right).$$

This bound has two regimes that emerge from the concentration properties of the binomial distribution. The more important regime for our discussion is when $k = \tilde{O}(n/m)$ for which the bound is $\tilde{O}(\sqrt{k/(nm)})$. In other words, achieving the same bias requires $O(m)$ more queries than in the binary case. What is perhaps surprising in this bound is that the difficulty of overfitting is not simply due to an increase in the amount of information per label. The label set $\{1, \ldots, m\}$ can be indexed with only $\log(m)$ bits of information.

We remark that these bounds hold even if the algorithm has access to the data points without the corresponding labels. The proofs follow from information-theoretic compression arguments and can be easily extended to any algorithm for which one can bound the amount of information extracted by the queries (e.g. via the approach in (Dwork et al., 2015a)).

Complementing this upper bound, we describe two attack algorithms that establish lower bounds on the bias in the two parameter regimes.

Theorem 1.2 (Point-wise attack, informal)**.** *For sufficiently large n and $n \ge k \ge k_{\min} = O(m \log m)$ there is an attack that uses k queries and on any dataset S outputs f such that*

$$\mathsf{acc}_S(f) = \frac{1}{m} + \Omega\left(\sqrt{\frac{k}{nm^2}}\right).$$

The algorithm underlying Theorem 1.2 outputs a classifier that computes a weighted plurality of labels given by its queries, with weights determined by the per-query accuracies observed. We remark that such an attack is rather natural, in that the function it produces is close to those produced by boosting and other common techniques for model aggregation. It also allows to easily incorporate any prior distribution over a label of each point. In addition, it is adaptive in the relatively weak sense: all queries are independent from one another except for the final classifier that combines them.

This attack is computationally efficient and we prove that it is optimal within a broad class of attacks that we call *point-wise*. Roughly speaking, such an attack predicts a label independently for each data point rather than reasoning jointly over the labels of multiple points in the test set. The

proof of Theorem 1.2 requires a relatively delicate analysis of the underlying random process.

Theorems 1.1 and 1.2 leave open a gap between bounds in the dependence on m. We conjecture that our analysis of the attack in Theorem 1.2 is asymptotically optimal and thus, considering the optimality of the attack, gives a lower bound for all point-wise attacks. If correct, our conjecture suggests that the effect of a large number of labels on mitigating overfitting is even more pronounced for such attacks.

Our second attack is based on an algorithm that exactly reconstructs the labels on a subset of the test set.

Theorem 1.3 (Reconstruction-based attack, informal)**.** *For any $k = \Omega(m \log m)$, there exists an attack $\mathcal{A}$ with access to test set points such that $\mathcal{A}$ uses k queries and on any dataset S outputs f such that*

$$\mathsf{acc}_S(f) = \min\left\{1, \frac{1}{m} + \Omega\left(\frac{k \log(k/m)}{n \log m}\right)\right\}.$$

The attack underlying Theorem 1.3 requires knowledge of the test points (in particular, is not point-wise like the previous one) and is not computationally efficient in general. For some $t \leq n$ it reconstructs the labeling of the first t points in the test set using queries that are random over the first t points and fixed elsewhere. The value t is chosen to be sufficiently small so that the answers to k random queries are sufficient to uniquely identify the correct labeling with high probability. This analysis builds on and generalizes the classical results of Erdos & Rényi (1963) and Chvátal (1983). A natural question for future work is whether similar bias can be achieved without access to test set points and by a polynomial time algorithm (currently a polynomial time algorithm is only known for the binary case (Bshouty, 2009)).

Experimental evaluation. The goal of our experimental evaluation is to come up with effective attacks to stress test multiclass benchmarks. We explore attacks based on our point-wise algorithm in particular. Although designed for worst-case label uncertainty, the point-wise attack proves applicable in a realistic setting once we reduce both the set of points and the set of labels we apply it to.

What drives performance in our experiments is the kind of prior information the attacker has. In our theory, we generally assumed a *prior-free* attacker that has no a priori information about the labels in the test set. In practice, an analyst almost always knows a model that performs better than random guessing. We therefore split our experiments into two parts: (i) simulations in the prior-free case, and (ii) effective heuristics for the ImageNet benchmark when prior information is available in the form of a well-performing model.

Our prior-free simulations it becomes substantially more difficult to overfit as the number of classes grows as predicted by our theory. Under the same simulation, restricted to two classes, we also see that our attack improves on the one proposed in (Blum & Hardt, 2015) for binary classification.

Turning to real data and models, we consider the well-known 2012 ILSVRC benchmark based on ImageNet (Russakovsky et al., 2015), for which the test set consist of $50{,}000$ data points with 1000 labels. Standard models achieve accuracy of around 75% on the test set. It makes sense to assume that an attacker has access to such a model and will use the information provided by the model to overfit more effectively. We ignore the trained model parameters and only use the model's so-called *logits*, i.e., predictive scores assigned to each class for each image in the test set. In other words, the relevant information provided by the the model is a $50{,}000 \times 1000$ array.

But how exactly can we use a well-performing model to overfit with fewer queries? We experiment with three increasingly effective strategies:

1. The attacker uses the model's logits as the prior information about the labels. This gives only a minor improvement over a prior-free attack.

2. The attacker uses the model's logits to restrict the attack to the subset of the test set corresponding to the lowest "confidence" points. This strategy gives modest improvements over a prior-free attack.

3. The attacker can exploit the fact that the model has good top-R accuracy, meaning that, for every image, the R highest weighted categories are likely to contain the correct class label. The attacker then focuses only on selecting from the top R predictions for each point. For $R = 2$, this effectively reduces the number of possible labels to the binary case. As a consequence, with this strategy we're seeing an attack that's nearly as effective as an attack in the binary case would be.

In absolute terms, our best performing attack overfits by about 3% with 5000 queries.

Naturally, the multiclass setting admits attacks more effective than the prior-free baseline. However, even after making use of the prior the remaining uncertainly over multiple classes makes overfitting much harder than in the binary case. Such attacks also require more sophistication and hence it is natural to conjecture that they are less likely to be the accidental work of a well-intentioned practitioner.

1.2. Related Work

The problem of biasing results due to adaptive reuse of the test data is now well-recognized. Most relevant to us are the

developments starting with the work of Dwork et al. (2014; 2015b) on reusable holdout mechanisms. In this work, noise addition and the tools of differential privacy serve to improve on the $\sqrt{k/n}$ worst-case bias of the standard holdout method to roughly $k^{1/4}/\sqrt{n}$. The latter requires a strengthened generalization bound due to (Bassily et al., 2016). Computational hardness results suggest that no trivial accuracy is possible in the adaptive setting for $k > n^2$ (Hardt & Ullman, 2014; Steinke & Ullman, 2015).

Blum and Hardt (Blum & Hardt, 2015) developed a limited feedback holdout mechanism, called Ladder algorithm, that only provides feedback when an analyst improved on the previous best result significantly. This simple mechanism leads to a bound of $\log(k)^{2/3}/n^{1/3}$ on what is called the *leaderboard error* in their work. With the help of noise addition, the bound can be improved to $\log(k)^{3/5}/n^{2/5}$ (Hardt, 2017). Blum and Hardt also give an attack on the standard holdout mechanism that achieves the $\sqrt{k/n}$ bound for a binary prediction problem.

Accuracy on a test set is a sum of accuracies at individual points. Therefore our attacks on the test set are related to the vast literature on (approximate) recovery from linear measurements that we cannot adequately survey here (see for example (Vershynin, 2015)). The primary difference between our work and the existing literature is the focus on the multiclass setting which no longer has the simple linear structure of the binary case. (In the binary case the accuracy measurement is just an inner product between the query and the labels in viewed as $\{\pm 1\}$.) In addition, even in the binary case the closest literature (see below) focuses the analysis on prediction with high accuracy (or small error) whereas we focus on the regime where the advantage over random guessing is relatively small.

Perhaps the closest in spirit to our work are database reconstruction attacks in the privacy literature. In this context it was first demonstrated by Dinur & Nissim (2003) that sufficiently accurate answers to $O(n)$ random linear queries allow exact reconstruction of a binary database with high probability. Many additional attacks have been developed in this context allowing more general notions of errors in the answers (e.g. (Dwork et al., 2007)) and specific classes of queries (e.g. (Kasiviswanathan et al., 2010; 2013)). To the best of our knowledge this literature does not consider queries corresponding to prediction accuracy in the multiclass setting and also focuses on (partial) reconstruction as opposed to prediction bias. Defenses against reconstruction attacks have lead to the seminal development of the notion of differential privacy (Dwork et al., 2006).

Another closely related problem is reconstruction of a pattern in $[m]^n$ from accuracy measurements. That is for a query $q \in [m]^n$ such measurement returns the number of positions in which q is equal to the unknown pattern. In the binary case ($m = 2$) it was introduced by Shapiro (1960) and was studied in combinatorics and several other communities under a variety of names such as group testing and coin weighing problem on the spring scale (see (Bshouty, 2009) for a literature overview). In the general case, this problem is closely related to a generalization of the Mastermind board game (Wikipedia) with only black answer pegs used. Erdos & Rényi (1963) demonstrated that the optimal reconstruction strategy in the binary case uses $\Theta(n/\log n)$ measurements. An efficient algorithm achieving this bound was given by Bshouty (2009). General m was first studied by Chvátal (1983) who showed a bound of $O(n \log m/\log(n/m))$ for $m \leq n$ (see Doerr et al. (2016) for a recent literature overview). It is not hard to see that the setting of this reconstruction problem is very similar to our problem when the attack algorithm has access to the test set points (and only their labels are unknown). Indeed, the analysis of our reconstruction-based attack (Thm. 1.3) can be seen as a generalization of the argument from Erdos & Rényi (1963); Chvátal (1983) to partial reconstruction. In contrast, our point-wise attack does not require knowledge of the test points and it gives bounds on achievable bias (which has not been studied in the context of pattern reconstruction).

An attack on a test set is related to a boosting algorithm. The goal of a boosting algorithm is to output a high-accuracy predictor by combining the information from multiple low-accuracy ones. A query function to the test set that has some correlation with the target function gives a low-accuracy predictor on the test set and an attack algorithm needs to combine the information from these queries to get the largest possible prediction accuracy on the test set. Indeed, our optimal point-wise attack (Thm. 1.2) effectively uses the same combination rule as the seminal Adaboost algorithm (Freund & Schapire, 1997) and its multiclass generalization (Hastie et al., 2009). Note that in our setting one cannot modify the weights of individual points in the test set (as is required by boosting). On the other hand, unlike a boosting algorithm, an attack algorithm can select which predictors to use as queries. Another important difference is that boosting algorithms are traditionally analyzed in the setting when the algorithm achieves high-accuracy, whereas we deal primarily with the more delicate low-accuracy regime.

2. Preliminaries

Let $S = (x_i, y_i)_{i \in [n]}$ denote the test set, where $(x_i, y_i) \in X \times Y$. Let $m = |Y|$ and without loss of generality we assume that $Y = [m]$. For $f\colon X \to Y$ its accuracy on the test set is $\mathsf{acc}_S(f) = \frac{1}{n}\sum_{i \in [n]} \mathsf{Ind}(f(x_i) = y_i)$ We are interested in overfitting attack algorithms that do not have access to the test set S. Instead, they have query access to accuracy on the test set S, i.e. for any classifier $f\colon X \to Y$

the algorithm can obtain the value $\mathsf{acc}_S(f)$. We refer to each such access as a query, and we denote the execution of an algorithm $\mathcal{A}$ with access to accuracy on the test S and $\mathcal{A}^{\mathcal{O}(S)}$. In addition, in some settings the attack algorithm may also have access to the set of points $x_1, \ldots, x_n$.

A k-query test set overfitting attack is an algorithm that, given access to at most k accuracy queries on some unknown test set S, outputs a function f. For any such possibly randomized algorithm $\mathcal{A}$ we define

$$\mathsf{acc}(\mathcal{A}) \doteq \inf_{S \in (X \times Y)^n} \mathop{\mathbf{E}}_{f=\mathcal{A}^{\mathcal{O}(S)}} [\mathsf{acc}_S(f)].$$

An algorithm is non-adaptive if none of its queries depend on the accuracy values of previous queries (however the output function depends on the accuracies so a query for that function is adaptive).

The main attack we design will be from a restricted class of *point-wise* attacks. We define an attack is *point-wise* if its queries and output function are generated for each point individually (while still having access to accuracy on the entire dataset). More formally, $\mathcal{A}$ is defined using an algorithms $\mathcal{B}$ that evaluated queries and the final classifier. A query f_ℓ at x is defined as the execution of $\mathcal{B}$ on values $f_1(x), \ldots, f_{\ell-1}(x)$ and the corresponding accuracies: $\mathsf{acc}_S(f_1), \ldots, \mathsf{acc}_S(f_{\ell-1})$. Similarly, for k query attack, the value of the final classifier f at x is defined as the execution of $\mathcal{B}$ on $f_1(x), \ldots, f_k(x)$ and $\mathsf{acc}_S(f_1), \ldots, \mathsf{acc}_S(f_k)$. An important property of point-wise attacks is that they can be easily implemented without access to data points. Further, the accuracy they achieve depends only on the vector of target labels.

Our upper bounds on the bias will apply even to algorithms that have access to points $x_1, \ldots, x_n$. The accuracy of such algorithms depends only on target labels. Hence for most of the discussion we describe the test set by the vector of labels $\bar{y} = (y_1, \ldots, y_n)$. Similarly, we specify each query by a vector of labels on the points in the dataset $\bar{q} = (q_1, \ldots, q_n) \in [m]^n$. Accordingly, we use $\bar{y}$ in place of the test set and $\bar{q}$ in place of a classifier in our definitions of accuracy and access to the oracle (e.g. $\mathsf{acc}_{\bar{y}}(\bar{q})$ and $\mathcal{A}^{\mathcal{O}(\bar{y})}$).

In addition to worst-case (expected) accuracy, we will also consider the average-case accuracy of the attack algorithm on randomly sampled labels. The random choice of labels may reflect the uncertainty that the attack algorithm has about the labels. Hence it is natural to refer to it as a prior distribution. In general, the prior needs to be specified on all points in X, but for point-wise attacks or attacks that have access to points it is sufficient to specify a vector $\bar{\pi} = (\pi_1, \ldots, \pi_n)$, where each π_i is a probability mass function on $[m]$ corresponding to the prior on y_i. We use $\bar{y} \sim \bar{\pi}$ to refer to $\bar{y}$ being chosen randomly with each y_i sampled independently from π_i. We also define the average case accuracy of $\mathcal{A}$ relative to $\bar{\pi}$ by

$$\mathsf{acc}(\mathcal{A}, \bar{\pi}) \doteq \mathop{\mathbf{E}}_{\bar{y} \sim \bar{\pi}} \left[\mathop{\mathbf{E}}_{\bar{r}=\mathcal{A}^{\mathcal{O}(\bar{y})}} [\mathsf{acc}_{\bar{y}}(\bar{r})] \right].$$

Note that for every $\bar{\pi}$, $\mathsf{acc}(\mathcal{A}) \leq \mathsf{acc}(\mathcal{A}, \bar{\pi})$.

For a matrix of query values $Q \in [m]^{n \times k}$, $i \in [n]$ and $j \in [k]$, we denote by Q^j the j-th column of the matrix (which corresponds to query j) and by Q_i the i-th row of the matrix: $(Q_{i,1}, \ldots, Q_{i,k})$ (which corresponds to all query values for point i). For a matrix of queries Q and label vector $\bar{y}$ we denote by $\mathsf{acc}_{\bar{y}}(Q) \doteq (\mathsf{acc}_{\bar{y}}(Q_j))_{j \in [k]}$.

3. Upper Bound

In this section we formally establish the upper bound on bias that can be achieved by any overfitting attack on a multiclass problem. The upper bound assumes that the attacker does not have any prior knowledge about the test set. That is, its prior distribution is uniform over all possible labelings.

The upper bound applies to algorithms that have access to the points in the test set. The upper bound has two distinct regimes. For $k = \tilde{O}(n/m)$ the upper bound on bias is $O\left(\sqrt{\frac{k \log n}{nm}}\right)$ and so the highest bias achieved in this regime is $\tilde{O}(1/m)$ (i.e. total accuracy improves by at most a constant factor). For $k \geq n/m$, the upper bound is $O\left(\frac{k \log n}{n}\right)$. Note that, in this regime, the attacker pays on average one query to improve the accuracy by one data point (up to log factors).

The proof of the upper bound relies on a simple description length argument, showing that finding a classifier with desired accuracy and non-negligible probability of success requires learning many bits about the target labeling.

Theorem 3.1. *Let m, n, k be positive integers and μ_m^n denote the uniform distribution over $[m]^n$. Then for every k-query attack algorithm $\mathcal{A}$, $\delta > 0$, $b = k \ln(n+1) + \ln(1/\delta)$, and*

$$\epsilon = 2 \cdot \max\left\{ \sqrt{\frac{b}{nm}}, \frac{b}{n} \right\},$$

$$\mathop{\mathbf{Pr}}_{\bar{y} \sim \mu_m^n, \bar{r}=\mathcal{A}^{\mathcal{O}(\bar{y})}} \left[\mathsf{acc}_{\bar{y}}(\bar{r}) \geq \frac{1}{m} + \epsilon \right] \leq \delta.$$

Remark 3.2. *The upper bound applies to arbitrary test set access models that limit the number of bits revealed. Specifically, if the information that the attacker learns about the labeling can be represented using t bits then the same upper bound applies for $b = t + \ln(1/\delta)$. It can also be easily generalized to algorithms whose output has bounded (approximate) max-information with the labeling (Dwork et al., 2015a).*

This upper bound can also be converted to a simpler one on the expected accuracy by setting $\delta = 1/n$ and noticing that accuracy is bounded above by 1. Therefore, for

$$\epsilon = \frac{1}{n} + 2 \cdot \max\left\{\sqrt{\frac{(k+1)\ln(n+1)}{nm}}, \frac{(k+1)\ln(n+1)}{n}\right\},$$

we have $\mathsf{acc}(\mathcal{A}, \mu_m^n) \leq \frac{1}{m} + \epsilon$.

4. Test Set Overfitting Attacks

In this section we will examine two attacks that both rely on queries chosen uniformly at random. Our first attack will be a point-wise attack that simply estimates the probability of each of the labels for the point, given the per-query accuracies, and then outputs the most likely label. We will show that this algorithm is optimal among all point-wise algorithms and then analyze the bias of this attack.

We then analyze the accuracy of an attack that relies on access to data points and is not computationally efficient. While such an attack might not be feasible in many scenarios (and we do not evaluate it empirically), it demonstrates the tightness of our upper bound on the optimal bias. This attack is based on exactly reconstructing part of the test set labels. Omitted proofs appear in the supplemental material.

4.1. Point-wise Attack

The queries in our attack are chosen randomly and uniformly. A point-wise algorithm can implement this easily because each coordinate of such a query is independent of all the rest. Hence we only need to describe how the label of the final classifier on each point is output, given the vector of the point's k labels $\bar{s} = (s_1, \ldots, s_k)$ from each query, and given the corresponding accuracies $\bar{\alpha} = (\alpha_1, \ldots, \alpha_k)$. To output the label our algorithm computes for each of the possible labels the probability of the observed vector of queries given the observed accuracies. Specifically, if the correct label is $\ell \in [m]$ then the probability of observing s_j given accuracy α_j is α_j if $s_j = \ell$ and $\frac{(1-\alpha_j)}{m-1}$ otherwise. Accordingly, for each label ℓ the algorithm considers:

$$\mathsf{conf}(\ell, \bar{s}, \bar{\alpha}) = \prod_{j \in [k], s_j = \ell} \alpha_j \times \prod_{j \in [k], s_j \neq \ell} \frac{(1-\alpha_j)}{m-1}.$$

It then predicts the label that maximizes conf, and in case of ties it picks one of the maximizers randomly.

This algorithm also naturally incorporates the prior distribution over labels $\bar{\pi} = (\pi_i)_{i \in [n]}$. Specifically, on point i the algorithm outputs the label that maximizes $\pi_i(\ell) \cdot \mathsf{conf}(\ell, \bar{s}, \bar{\alpha})$. Note that the version without a prior is equivalent to one with the uniform prior. We refer to these versions of the attack algorithm as $\mathtt{NB}$ and $\mathtt{NB}_{\bar{\pi}}$, respectively.

We will start by showing that $\mathsf{conf}(\ell, \bar{s}, \bar{\alpha})$ accurately computes the probability of query values.

Lemma 4.1. *Let $\mu_m^{n \times k}$ denote the uniform distribution over k queries. Then for every $\bar{y} \in [m]^n$, accuracy vector $\bar{\alpha}$, $\bar{s} \in [m]^k$, $i \in [n]$ and $j \in [k]$,*

$$\Pr_Q\left[Q_{i,j} = s_j \mid \mathsf{acc}_{\bar{y}}(Q) = \bar{\alpha}\right] = \begin{cases} \alpha_j & \text{if } s_j = y_i, \\ \frac{1-\alpha_j}{m-1} & \text{otherwise.} \end{cases}$$

Further $Q_{i,j}$ are independent conditioned on $\mathsf{acc}_{\bar{y}}(Q) = \bar{\alpha}$. That is

$$\begin{aligned} &\Pr_Q\left[Q_i = \bar{s} \mid \mathsf{acc}_{\bar{y}}(Q) = \bar{\alpha}\right] \\ &\quad = \prod_{j \in [k], s_j = y_i} \alpha_j \times \prod_{j \in [k], s_j \neq y_i} \frac{(1-\alpha_j)}{m-1} \\ &\quad = \mathsf{conf}(y_i, \bar{s}, \bar{\alpha}). \end{aligned}$$

Proof. For every fixed value $\bar{y}$, the distribution $Q \sim \mu_m^{n \times k}$ conditioned on $\mathsf{acc}_{\bar{y}}(Q) = \bar{\alpha}$ is uniform over all query matrices that satisfy $\mathsf{acc}_{\bar{y}}(Q) = \bar{\alpha}$. This implies that for every j the marginal distribution over Q^j is uniform over the set $\{\bar{q} \mid \mathsf{acc}_{\bar{y}}(\bar{q}) = \alpha_j\}$. We denote this distribution $\rho_{\bar{y},\alpha_j}$. In addition, Q conditioned on $\mathsf{acc}_{\bar{y}}(Q) = \bar{\alpha}$ is just the product over marginals $\rho_{\bar{y},\alpha_1} \times \cdots \times \rho_{\bar{y},\alpha_k}$. It is easy to see from the definition of $\rho_{\bar{y},\alpha_j}$, that for every $q \in [m]$,

$$\Pr_{\bar{q} \sim \rho_{\bar{y},\alpha_j}}[\bar{q}_i = q] = \begin{cases} \alpha_j & \text{if } q = y_i, \\ \frac{1-\alpha_j}{m-1} & \text{otherwise.} \end{cases}$$

Thus for every $\bar{s}$,

$$\begin{aligned} &\Pr_Q\left[Q_i = \bar{s} \mid \mathsf{acc}_{\bar{y}}(Q) = \bar{\alpha}\right] \\ &\quad = \prod_{j \in [k], s_j = y_i} \alpha_j \times \prod_{j \in [k], s_j \neq y_i} \frac{(1-\alpha_j)}{m-1} \\ &\quad = \mathsf{conf}(\ell, \bar{s}, \bar{\alpha}). \end{aligned}$$

□

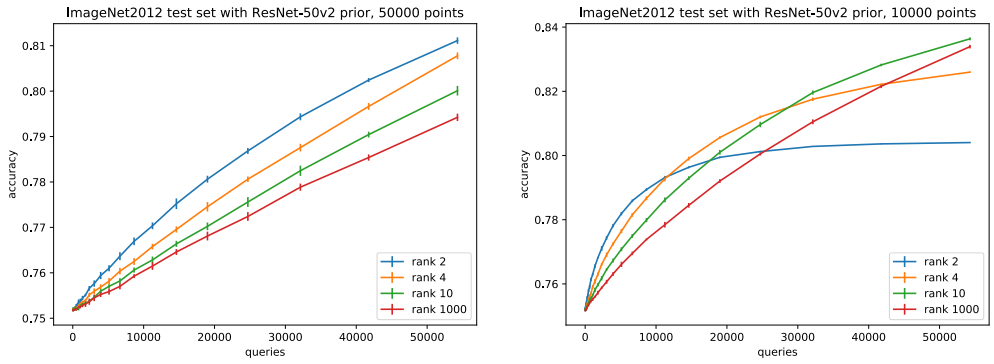
This lemma allows us to conclude that our algorithm is optimal for this setting.

Theorem 4.2. *Let $\bar{\pi} = (\pi_1, \ldots, \pi_n)$ be an arbitrary prior on n labels. Let $\mathcal{A}$ be an arbitrary point-wise attack using k randomly and uniformly chosen queries. Then*

$$\mathsf{acc}(\mathcal{A}, \bar{\pi}) \leq \mathsf{acc}(\mathtt{NB}_{\bar{\pi}}, \bar{\pi}).$$

In particular, $\mathsf{acc}(\mathcal{A}) \leq \mathsf{acc}(\mathtt{NB})$.

We now provide the analysis of a lower bound on the bias achieved by $\mathtt{NB}$. Our analysis will apply to a simpler algorithm that effectively computes the plurality label among



(a) Full test set

(b) 10K least confident points

Figure 1. Average accuracy (with standard deviation bars), over 10 attack trials, of the $\texttt{NB}_{\bar{\pi}}$ attack against the ImageNet test set. The attacker's gains improve when the effective class count, as indicated by *rank* (the value R used in the top-R heuristic) is reduced, illustrating the increasing vulnerability of the test set when classes are removed.

those for which accuracy is sufficiently high (larger than the mean plus one standard deviation). Further, to simplify the analysis, we take the number of queries to be a draw from the Poisson distribution. This Poissonization step ensures that the counts of the times each label occurs are independent. The optimality of the NB attack implies that the bias achieved by NB is at least as large as that of this simpler attack.

Theorem 4.3. *For any $m \geq 2$, $n \geq k \geq k_{\min} = O(\ln n + m \ln m)$, we have that*

$$\mathsf{acc}(\texttt{NB}) = \frac{1}{m} + \Omega\left(\frac{\sqrt{k}}{m\sqrt{n}}\right).$$

The key to our proof of Theorem 4.3 is the following lemma about biased and Poissonized multinomial random variables.

Lemma 4.4. *For $\gamma \geq 0$ let ρ_γ denote the categorical distribution ρ_γ over $[m]$ such that $\mathbf{Pr}_{s\sim\rho_\gamma}[s = m] = \frac{1}{m} + \gamma$ and for all $y \neq m$, $\mathbf{Pr}_{s\sim\rho_\gamma}[s = y] = \frac{1}{m} - \frac{\gamma}{m-1}$. For an integer t, let $\mathsf{Mnom}(t, \rho_\gamma)$ be the multinomial distribution over counts corresponding to t independent draws from ρ_γ. For a vector of counts $\bar{c}$, let $\mathrm{argmax}(\bar{c})$ denote the index of the largest value in $\bar{c}$. If several values achieve the maximum then one of the indices is picked randomly. Then for $\lambda \geq 2m\ln(4m)$ and $\gamma \leq \frac{1}{8\sqrt{\lambda m}}$,*

$$\mathop{\mathbf{Pr}}_{t\sim\mathsf{Pois}(\lambda),\bar{c}\sim\mathsf{Mnom}(t,\rho_\gamma)}[\mathrm{argmax}(\bar{c}) = m] \geq \frac{1}{m} + \Omega\left(\frac{\gamma\sqrt{\lambda}}{\sqrt{m}}\right)$$

4.2. Reconstruction-based Attack

Our second attack relies on a probabilistic argument, showing that any dataset's label vector is, with high probability, uniquely identified by the accuracies of $O\left(\max\left\{\frac{n\ln m}{\ln(n/m)}, m\ln(nm)\right\}\right)$ uniformly random queries. This argument was first used for the binary label case by Erdos & Rényi (1963) and generalized to arbitrary m by Chvátal (1983). We further generalize it to allow identification when the accuracy values are known only up to a fixed shift. This is needed as we apply this algorithm to a subset of labels such that the accuracy on the remaining labels is unknown. Formally, the unique identification property follows.

Theorem 4.5. *Say that a query matrix $Q \in [m]^{n\times k}$ recovers any label vector from shifted accuracies if there do not exist distinct $\bar{y}, \bar{y}' \in [m]^n$ and shift $\beta \in \mathbb{R}$ such that*

$$\mathsf{acc}_{\bar{y}}(Q) = \mathsf{acc}_{\bar{y}'}(Q) + \beta \cdot (1, 1, \ldots, 1).$$

For $m \geq 3$ and $k = \max\left\{\frac{5n\ln m}{\ln(n/4m)}, 20m\ln(nm)\right\}$, with probability at least $1/2$ over the choice of random $Q \sim \mu_m^{n\times k}$, Q recovers any label vector from shifted accuracies.

Naturally, if for all distinct labeling $\bar{y}, \bar{y}'$, $\mathsf{acc}_{\bar{y}}(Q) \neq \mathsf{acc}_{\bar{y}'}(Q)$ then we can recover the unknown labeling $\bar{y}$ simply by trying out all possible labeling $\bar{y}'$ and picking the one for which the $\mathsf{acc}_{\bar{y}}(Q) = \mathsf{acc}'_{\bar{y}}(Q)$. Thus an immediate implication of Thm. 4.5 is that there exists a fixed set of $k = O\left(\max\left\{\frac{n\ln m}{\ln(n/m)}, m\ln(nm)\right\}\right)$ queries that can be used to reconstruct the labels. In particular, this gives an attack algorithm with accuracy 1. If k is not sufficiently large for reconstructing the entire set of labels then it can be used to reconstruct a sufficiently small subset of the labels (and predict the rest randomly). Hence we obtain the following bound on achievable bias.

Corollary 4.6. *For any $k \geq 40m\ln(m)$, there exists an attack $\mathcal{A}$ with access to points such that*

$$\mathsf{acc}(\mathcal{A}) = \min\left\{1, \frac{1}{m} + \Omega\left(\frac{k\ln(k/m)}{n\ln m}\right)\right\}.$$

5. Experimental Evaluation

This section presents a variety of experiments intended (i) to corroborate formal bounds, (ii) to provide a comparison to previous attack in the binary classification setting, and (iii) to explore the practical application of the NB attack from Section 4.1.

To visualize the attack's performance, we first simply simulate our attack directly on a test set of labels generated uniformly at random from m classes. The attack assumes the uniform prior over the same labels and Figures in the supplemental material show the observed advantage of the attack over the population error rate of $1/m$, across a range of query budgets, on test sets of size 10,000 and 50,000 respectively.[1]

In the binary classification setting, we compare to the majority-based attack proposed by Blum & Hardt (2015), under the same synthetic dataset. Recall that the NB attack is based on a majority (more generally, plurality) weighted by the per-query accuracies. The majority function is weighted only by ± 1 values, as a means of ensuring non-negative correlation of each query with the test set labels. It does not consider low- and high-accuracy queries differently, where NB does. A figure in the supplemental material shows the observed relative advantage of the NB attack. Note that simulating uniformly random binary labels places both attacks on similar starting grounds: the attacks otherwise differ in that $\texttt{NB}_{\bar{\pi}}$ can incorporate a prior distribution $\bar{\pi}$ over class labels to its advantage.

Our remaining experiments aim to overfit to the ImageNet test set associated with the 2012 ILSVRC benchmark. As a form of prior information, we incorporate the availability of a standard and (nearly) state of the art model. Specifically, we train a ResNet-50v2 model over the ImageNet training set. On the test set, this model achieves a prediction accuracy of 75.1% and a top-R accuracy of 85.3%, 91.0%, and 95.3% for $R = 2, 4$, and 10, respectively.

As is common practice in classification, the ResNet model is trained under the cross-entropy loss (a.k.a. the multiclass logistic loss). That is, it is trained to output scores (logits) that define a probability distribution over classes, from which it predicts the maximally-probable class label. We use the model's logits—a 50,000 by 1000 array—as the sole source of side information for attack. All results are summarized in Figure 1, several highlights of which follow.

First, we consider plugging the model's predictive distribution in as the prior $\bar{\pi}$ in the $\texttt{NB}_{\bar{\pi}}$ attack, yielding modest gains, e.g. a 0.42% accuracy boost after 5200 queries (averaged over 10 simulations of the attack).

[1]The number of points in these synthetic test sets is chosen to mirror the CIFAR-10 and ImageNet test sets.

Next, we observe that the model is highly confident about many of its predictions. Recalling the dependence on the test set size n in our upper bound, we consider a simple heuristic for culling points. Namely, we select the 10K points for which the model is least confident of its prediction in order to attack a test set that is a fifth of the original size. This heuristic presents a trade-off: one reduces n to 10K, but commits to leaving intact the errors made by the model on the 40K more confident points. Applying this heuristic improves gains further, e.g. to a 1.44% accuracy boost after 5200 queries.

Finally, we consider another heuristic to reduce m, the effective number of classes in the attack, per this paper's focus on the multiple class count. Observing that the model has a high top-R accuracy (i.e. recall at R) for relatively small values of R, it is straightforward to apply the $\texttt{NB}_{\bar{\pi}}$ attack not to the original classes, but to selecting (pointwise) which of the model's top-R predictions to take. This heuristic presents a trade-off as well: one reduces m down to R, but commits to perform no better than the top-R accuracy of the model, a quantity that increases with R. Applying this heuristic together with the previous improves the attacker's advantage further. For instance, at $R = 2$, we observe a 3.0% accuracy boost after 5200 queries.

To put these numbers in perspective, we compare to a straightforward analytical baseline in supplemental material: the expected performance of the "linear scan attack." Namely, this is an attack that begins with a random query vector and successively submits queries by modifying the label of one point at a time, discovering the label's true value whenever the observed test set accuracy increases.

Acknowledgements

We thank Clément Canonne for his suggestion to use Poissonization in the proof of Theorem 4.3. We thank Chiyuan Zhang for his crucial help in the setup of our ImageNet experiment. We thank Kunal Talwar, Tomer Koren, and Yoram Singer for insightful discussion throughout the course of this work.

References

Bassily, R., Nissim, K., Smith, A. D., Steinke, T., Stemmer, U., and Ullman, J. Algorithmic stability for adaptive data analysis. In *STOC*, pp. 1046–1059, 2016.

Blum, A. and Hardt, M. The ladder: A reliable leaderboard for machine learning competitions. *CoRR*, abs/1502.04585, 2015. URL http://arxiv.org/abs/1502.04585.

Bshouty, N. H. Optimal algorithms for the coin weighing problem with a spring scale. In *COLT*, 2009. URL

http://www.cs.mcgill.ca/%7Ecolt2009/papers/004.pdf#page=1.

Chvátal, V. Mastermind. *Combinatorica*, 3(3):325–329, 1983. URL https://doi.org/10.1007/BF02579188.

Dinur, I. and Nissim, K. Revealing information while preserving privacy. In *PODS*, pp. 202–210, 2003.

Doerr, B., Doerr, C., Spöhel, R., and Thomas, H. Playing mastermind with many colors. *Journal of the ACM (JACM)*, 63(5):42, 2016.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *TCC*, pp. 265–284, 2006.

Dwork, C., McSherry, F., and Talwar, K. The price of privacy and the limits of lp decoding. In *Proceedings of STOC*, pp. 85–94. ACM, 2007.

Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O., and Roth, A. Preserving statistical validity in adaptive data analysis. *CoRR*, abs/1411.2664, 2014. Extended abstract in STOC 2015.

Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O., and Roth, A. Generalization in adaptive data analysis and holdout reuse. *CoRR*, abs/1506, 2015a. Extended abstract in NIPS 2015.

Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O., and Roth, A. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349 (6248):636–638, 2015b. doi: 10.1126/science.aaa9375. URL http://www.sciencemag.org/content/349/6248/636.abstract.

Erdos, P. and Rényi, A. On two problems of information theory. *Magyar Tud. Akad. Mat. Kutató Int. Közl*, 8: 229–243, 1963.

Freund, Y. and Schapire, R. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Hardt, M. Climbing a shaky ladder: Better adaptive risk estimation. *CoRR*, abs/1706.02733, 2017. URL http://arxiv.org/abs/1706.02733.

Hardt, M. and Ullman, J. Preventing false discovery in interactive data analysis is hard. In *FOCS*, pp. 454–463, 2014.

Hastie, T., Rosset, S., Zhu, J., and Zou, H. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360, 2009.

Kasiviswanathan, S. P., Rudelson, M., Smith, A., and Ullman, J. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In *Proceedings of STOC*, pp. 775–784. ACM, 2010.

Kasiviswanathan, S. P., Rudelson, M., and Smith, A. The power of linear reconstruction attacks. In *Proceedings of SODA*, pp. 1415–1433. SIAM, 2013.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do CIFAR-10 classifiers generalize to CIFAR-10? *CoRR*, abs/1806.00451, 2018.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do ImageNet classifiers generalize to ImageNet? *CoRR*, abs/1902.10811, 2019.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015.

Shapiro, H. S. Problem e 1399. *American Mathematical Monthly*, 67:82, 1960.

Steinke, T. and Ullman, J. Interactive fingerprinting codes and the hardness of preventing false discovery. In *COLT*, pp. 1588–1628, 2015. URL http://jmlr.org/proceedings/papers/v40/Steinke15.html.

Vershynin, R. Estimation in high dimensions: a geometric perspective. In *Sampling theory, a renaissance*, pp. 3–66. Springer, 2015.

Wikipedia. Mastermind (board game). URL https://en.wikipedia.org/wiki/Mastermind_(board_game).