The advantages of multiple classes for reducing overfitting from test set reuse

Vitaly Feldman* Roy Frostig† Moritz Hardt‡

June 3, 2019

Abstract

Excessive reuse of holdout data can lead to overfitting. Known results show that, in the worst-case, given the accuracies of k adaptively chosen classifiers on a data set of size n, one can create a classifier with a 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 achieves a bias of $\Omega(\sqrt{k/(m^2n)})$ 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.

Our work extends developments in understanding overfitting due to adaptive data analysis to multiclass prediction problems. It also bears out the surprising fact that multiclass prediction problems are significantly more robust to overfitting when reusing a test (or holdout) dataset. This offers an explanation as to why popular multiclass prediction benchmarks, such as ImageNet, may enjoy a longer lifespan than what analysis in the context of binary classification suggests.

1 Introduction

Several machine learning benchmarks have shown surprising longevity, such as the ILSVRC 2012 image classification benchmark based on the ImageNet database [Rus+15]. 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 [Rec+18; Rec+19; YB19] 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.

*Google Brain. Part of this work was done while the author was visiting the Simons Institute for the Theory of Computing.
†Google Brain
‡University of California, Berkeley. Work done while at Google.

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 $\Omega(\sqrt{k/n})$ [Dwo+14].

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 [Dwo+15b] and limited feedback mechanisms [BH15]. However, the standard holdout method remains widely used in practice, ranging from machine learning benchmarks and data science competitions to validating scientific research and testing products during development. If the pessimistic bound were indicative of performance in practice, the holdout method would likely be much less useful than it is.

It 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 substantiate experimentally.

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 [Dwo+15a]).

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 the labels that comprise its queries, with weights determined by the per-query accuracies observed. 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 for simple incorporation of 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 rather 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, this conjecture suggests that the effect of a large number of labels on mitigating overfitting is even more pronounced for such attacks. Some support for this conjecture is given in our experimental section (Figure 4).

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 (but not their labels)—in contrast to a point-wise attack like the previous—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, with high probability, the correct labeling of t points jointly. This analysis builds on and generalizes the classical results of Erdos and Rényi [ER63] and Chvátal [Chv83]. A natural question for future work is whether a similar bias can be achieved without identifying test set points and in polynomial time (currently a polynomial time algorithm is only known for the binary case [Bsh09]).

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 the set of points and the set of labels to which we apply it.

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 [BH15] for binary classification.

Turning to real data and models, we consider the well-known 2012 ILSVRC benchmark based on ImageNet [Rus+15], 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., the 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 a 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 predicted classes for each point. For $R = 2$, this effectively reduces class count to the binary case.

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 harder than in the binary case. Such attacks also require more sophistication and hence it is natural to suspect 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. [Dwo+14; Dwo+15b] on reusable holdout mechanisms. In this work, noise addition and the tools of differential privacy serve to improve 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. [Bas+16]. Separately, computational hardness results suggest that no trivial accuracy is possible in the adaptive setting for $k > n^2$ [HU14; SU15].

Blum and Hardt [BH15] developed a limited feedback holdout mechanism, called the Ladder algorithm, that only provides feedback when an analyst improves on the previous best result significantly. This simple mechanism leads to a bound of $\log(k)^{2/3}/n^{1/3}$ on what they call the *leaderboard error*. With the help of noise addition, the bound can be improved to $\log(k)^{3/5}/n^{2/5}$ [Har17]. 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 an average of accuracies at individual points. Therefore our attacks on the test set are related to the vast literature on (approximate) recovery from linear measurements, which we cannot adequately survey here (see for example [Ver15]). 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 essentially an inner product between the query and the labels viewed in $\{\pm 1\}$.) 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 and Nissim [DN03] 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. [DMT07]) and specific classes of queries (e.g. [Kas+10; KRS13]). 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 landmark development of the notion of differential privacy [Dwo+06].

Another closely related problem is reconstruction of a pattern in $[m]^n$ from accuracy measurements. For a query $q \in [m]^n$, such a measurement returns the number of positions in which q is equal to the unknown pattern. In the binary case ($m = 2$), this problem was introduced by Shapiro [Sha60] and was studied in combinatorics and several other communities under a variety of names, such as "group testing" and "the coin weighing problem on the spring scale" (see [Bsh09] for a literature overview). In the general case, this problem is closely related to a generalization of the Mastermind board game [Wik] with only black answer pegs used. Erdos and Rényi [ER63] 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 [Bsh09]. General m was first studied by Chvátal [Chv83] who showed a bound of $O(n \log m/\log(n/m))$ for $m \leq n$ (see Doerr et al. [Doe+16] 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 (Theorem 1.3) can be seen as a generalization of the argument from Erdos and Rényi [ER63] and Chvátal [Chv83] to partial reconstruction. In contrast, our point-wise attack does not require such knowledge of the test points and it gives bounds on the 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 (Theorem 1.2) effectively uses the same combination rule as the Adaboost algorithm [FS97] and its multiclass generalization [Has+09]. 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 let μ_m^n denote the uniform distribution over $[m]^n$. We also define the average case accuracy of $\mathcal{A}$ relative to $\bar{\pi}$ by

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

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 $\mathsf{acc}_{\bar{y}}(Q) \doteq (\mathsf{acc}_{\bar{y}}(Q_j))_{j\in[k]}$.

2.1 Random variables and concentration

For completeness we include several standard concentration inequalities that we use below.

Lemma 2.1 ((Multiplicative) Chernoff bound)**.** *Let X be the average of n i.i.d. Bernoulli random variables with bias p. Then for $\alpha \in (0, 1)$*

$$\mathbf{Pr}[X \geq (1+\alpha)p] \leq e^{\frac{-\alpha^2 pn}{2+\alpha}} \text{ and}$$

$$\mathbf{Pr}[X \leq (1-\alpha)p] \leq e^{-\frac{\alpha^2 pn}{2}}.$$

We also state the Berry-Esseen theorem for the case of Bernoulli random variables.

Lemma 2.2. *Let X be the average of n i.i.d. Bernoulli random variables with bias $p \leq 1/2$. Then for every real v,*

$$|\mathbf{Pr}[X \leq v] - \mathbf{Pr}[\zeta \leq v]| = O\left(\frac{1}{\sqrt{pn}}\right),$$

where ζ is distributed according to the Gaussian distribution with mean p and variance $p(1-p)$.

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*

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

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

Proof. We first observe that for any fixed labeling $\bar{r}$, $\mathsf{acc}_{\bar{y}}(\bar{r})$ for $\bar{y}$ chosen randomly according to μ_m^n is distributed as the average of n independent Bernoulli random variables with bias $1/m$. By the Chernoff bound, for any fixed labeling $\bar{r}$,

$$\mathop{\mathbf{Pr}}_{\bar{y} \sim \mu_m^n}\left[\mathsf{acc}_{\bar{y}}(\bar{r}) \geq \frac{1}{m} + \epsilon\right] \leq e^{-\frac{mn\epsilon^2}{2+m\epsilon}}.$$

Therefore for any fixed distribution ρ over $[m]^n$, we have

$$\mathop{\mathbf{Pr}}_{\bar{r} \sim \rho, \bar{y} \sim \mu_m^n}\left[\mathsf{acc}_{\bar{y}}(\bar{r}) \geq \frac{1}{m} + \epsilon\right] \leq e^{-\frac{mn\epsilon^2}{2+m\epsilon}}. \tag{1}$$

Consider the execution of $\mathcal{A}$ with responses of the accuracy oracle fixed to some sequence of values $\alpha = (\alpha_1, \ldots, \alpha_k) \in \{0, 1/n, \ldots, 1\}^k$. We denote the resulting algorithm by $\mathcal{A}^\alpha$. It output distribution is fixed (that is independent of $\bar{y}$). Therefore by eq. (1) we have:

$$\mathop{\mathbf{Pr}}_{\bar{r} = \mathcal{A}^\alpha, \bar{y} \sim \mu_m^n}\left[\mathsf{acc}_{\bar{y}}(\bar{r}) \geq \frac{1}{m} + \epsilon\right] \leq e^{-\frac{mn\epsilon^2}{2+m\epsilon}}.$$

We denote the set $\{0, 1/n, \ldots, 1\}^k$ of possible values of α by V. Note that $|V| \leq (n+1)^k$ and thus we get:

$$\sum_{\alpha \in V} \mathop{\mathbf{Pr}}_{\bar{r} = \mathcal{A}^\alpha, \bar{y} \sim \mu_m^n}\left[\mathsf{acc}_{\bar{y}}(\bar{r}) \geq \frac{1}{m} + \epsilon\right] \leq (n+1)^k \cdot e^{-\frac{mn\epsilon^2}{2+m\epsilon}}.$$

Clearly, for every $\bar{y}$, the accuracy oracle $\mathcal{O}(\bar{y})$ outputs some responses in V. Therefore,

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{\bar{y} \sim \mu_m^n, \bar{r} = \mathcal{A}^{\mathcal{O}(\bar{y})}}\left[\mathsf{acc}_{\bar{y}}(\bar{r}) \geq \frac{1}{m} + \epsilon\right] \\
&\qquad \leq \sum_{\alpha \in V} \mathop{\mathbf{Pr}}_{\bar{r} = \mathcal{A}^\alpha, \bar{y} \sim \mu_m^n}\left[\mathsf{acc}_{\bar{y}}(\bar{r}) \geq \frac{1}{m} + \epsilon\right] \\
&\qquad \leq (n+1)^k \cdot e^{-\frac{mn\epsilon^2}{2+m\epsilon}}.
\end{aligned}$$

Now, if $\frac{k\ln(n+1)+\ln(1/\delta)}{n} \geq \frac{1}{m}$ then by definition of b and ϵ,

$$\begin{aligned}\epsilon &= 2\max\left\{\sqrt{\frac{b}{nm}}, \frac{b}{n}\right\} \\ &= 2\frac{b}{n} \geq \frac{2}{m}.\end{aligned}$$

Therefore we obtain that, $\frac{mn\epsilon^2}{2+m\epsilon} \geq \frac{n\epsilon}{2}$ and

$$(n+1)^k \cdot e^{-\frac{mn\epsilon^2}{2+m\epsilon}} \leq e^{k\ln(n+1)-\frac{n\epsilon}{2}} = e^{\ln\delta} = \delta.$$

Otherwise (when $\frac{k\ln(n+1)+\ln(1/\delta)}{n} < \frac{1}{m}$) we have that

$$\epsilon = 2\sqrt{\frac{b}{nm}} < \frac{2}{m}.$$

In this case $\frac{mn\epsilon^2}{2+m\epsilon} \geq \frac{mn\epsilon^2}{4}$ and

$$(n+1)^k \cdot e^{-\frac{mn\epsilon^2}{2+m\epsilon}} \leq e^{k\ln(n+1)-\frac{mn\epsilon^2}{4}} = e^{\ln\delta} = \delta.$$

□

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 [Dwo+15a].*

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.

Algorithm 1 The $\texttt{NB}_{\bar{\pi}}$ overfitting attack algorithm.

input Query access to a test set of n points over m labels, query budget k, and priors $\bar{\pi} = (\pi_i)_{i\in[n]}$.

Draw k queries $Q \in [m]^{n\times k}$ uniformly.

Submit queries $Q^1, \ldots, Q^k$ and receive corresponding accuracies $\bar{\alpha} = (\alpha_1, \ldots, \alpha_k)$.

For $i \in [n]$, compute:

$$z_i \leftarrow \operatorname*{argmax}_{\ell\in[m]} \left\{ \pi_i(\ell) \prod_{j\in[k], Q_{i,j}=\ell} \alpha_j \prod_{j\in[k], Q_{i,j}\neq\ell} \frac{(1-\alpha_j)}{m-1} \right\},$$

breaking any ties among maximizers uniformly at random.

output Predictions $\bar{z} = (z_1, \ldots, z_n)$

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:

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

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 $\texttt{NB}$ and $\texttt{NB}_{\bar{\pi}}$, respectively. The latter is summarized in Algorithm 1.

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

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

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 $\bar{s}$,

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

□

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

Proof. A point-wise attack $\mathcal{A}$ that uses a query matrix $Q \sim \mu_m^{n\times k}$ is fully specified by some algorithm $\mathcal{B}$ that takes as input the query values for the point $\bar{s} \in [m]^k$ and accuracy values $\bar{\alpha} = (\alpha_1, \ldots, \alpha_k)$ and outputs a label. By definition,

$$\begin{aligned} \mathsf{acc}(\mathcal{A}, \bar{\pi}) &= \mathop{\mathbf{E}}_{\bar{y},Q}\left[\frac{1}{n}\sum_{i\in[n]} \mathsf{Ind}(y_i = \mathcal{B}(Q_i, \mathsf{acc}_{\bar{y}}(Q))\right] \\ &= \frac{1}{n}\sum_{i\in[n]} \Pr_{\bar{y},Q}\left[y_i = \mathcal{B}(Q_i, \mathsf{acc}_{\bar{y}}(Q))\right], \end{aligned}$$

where $\bar{y} \sim \bar{\pi}$ and $Q \sim \mu_m^{n\times k}$ (and the same in the rest of the proof). Now for every fixed $i \in [n]$,

$$\begin{aligned} &\Pr_{\bar{y},Q}\left[y_i = \mathcal{B}(Q_i, \mathsf{acc}_{\bar{y}}(Q))\right] \\ &\quad = \sum_{\bar{\alpha}\in V} \Pr_{\bar{y},Q|\bar{\alpha}}\left[y_i = \mathcal{B}(Q_i, \bar{\alpha})\right] \cdot \Pr_{\bar{y},Q}[\mathsf{acc}_{\bar{y}}(Q) = \bar{\alpha}], \end{aligned}$$

where by $\bar{y}, Q \mid \bar{\alpha}$ we denote the distribution of Q and $\bar{y}$ conditioned on $\mathsf{acc}_{\bar{y}}(Q) = \bar{\alpha}$ and by V we denote the set of all possible accuracy vectors. For every fixed $\bar{\alpha} \in V$,

$$\begin{aligned} &\Pr_{\bar{y},Q|\bar{\alpha}}\left[y_i = \mathcal{B}(Q_i, \bar{\alpha})\right] \\ &= \sum_{\bar{s}\in[m]^k} \Pr_{\bar{y},Q|\bar{\alpha}}\left[y_i = \mathcal{B}(\bar{s}, \bar{\alpha}) \mid Q_i = \bar{s}\right] \cdot \Pr_{\bar{y},Q|\bar{\alpha}}\left[Q_i = \bar{s}\right]. \end{aligned}$$

For every fixed $\bar{s} \in [m]^k$, $\mathcal{B}(\bar{s}, \bar{\alpha})$ outputs a random label and the algorithm's randomness is independent of Q and $\bar{y}$. Hence,

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{\bar{y},Q|\bar{\alpha}}[y_i = \mathcal{B}(\bar{s}, \bar{\alpha}) \mid Q_i = \bar{s}] \\
&= \sum_{\ell \in [m]} \mathop{\mathbf{Pr}}_{\bar{y},Q|\bar{\alpha}}[y_i = \ell \mid Q_i = \bar{s}] \cdot \mathop{\mathbf{Pr}}_{\bar{y},Q|\bar{\alpha}}[\mathcal{B}(\bar{s}, \bar{\alpha}) = \ell] \\
&\leq \max_{\ell \in [m]} \mathop{\mathbf{Pr}}_{\bar{y},Q|\bar{\alpha}}[y_i = \ell \mid Q_i = \bar{s}] .
\end{aligned}$$

Moreover, the equality is achieved by the algorithm that outputs any value in

$$\mathsf{Opt}(\bar{s}, \bar{\alpha}) \doteq \operatorname*{argmax}_{\ell \in [m]} \mathop{\mathbf{Pr}}_{\bar{y},Q|\bar{\alpha}}[y_i = \ell \mid Q_i = \bar{s}] .$$

It remains to verify that $\mathtt{NB}_{\bar{\pi}}$ computes a value in $\mathsf{Opt}(\bar{s}, \bar{\alpha})$. Applying the Bayes rule we get

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{\bar{y},Q|\bar{\alpha}}[y_i = \ell \mid Q_i = \bar{s}] \\
&= \frac{\mathbf{Pr}_{\bar{y},Q|\bar{\alpha}}[Q_i = \bar{s} \mid y_i = \ell] \cdot \mathbf{Pr}_{\bar{y},Q|\bar{\alpha}}[y_i = \ell]}{\mathbf{Pr}_{\bar{y},Q|\bar{\alpha}}[Q_i = \bar{s}]}.
\end{aligned}$$

Now, the denominator is independent of ℓ and thus does not affect the definition of $\mathsf{Opt}(\bar{s}, \bar{\alpha})$. The distribution Q is uniform over all possible queries, and thus for every pair of vectors $\bar{y}, \bar{y}'$,

$$\mathop{\mathbf{Pr}}_{Q}[\mathsf{acc}_{\bar{y}}(Q) = \bar{\alpha}] = \mathop{\mathbf{Pr}}_{Q}[\mathsf{acc}_{\bar{y}'}(Q) = \bar{\alpha}].$$

Therefore the marginal of distribution $\bar{y} \sim \bar{\pi}, Q \sim \mu_m^{n \times k} \mid \mathsf{acc}_{\bar{y}}(Q) = \bar{\alpha}$ over label vectors is not affected by conditioning. That is, it is equal to $\bar{\pi}$. Therefore

$$\mathop{\mathbf{Pr}}_{\bar{y},Q|\bar{\alpha}}[y_i = \ell] = \mathop{\mathbf{Pr}}_{\bar{y},Q|\bar{\alpha}}[y_i = \ell] = \pi_i(\ell).$$

By Lemma 4.1 we obtain that

$$\mathop{\mathbf{Pr}}_{\bar{y},Q|\bar{\alpha}}[Q_i = \bar{s} \mid y_i = \ell] = \mathsf{conf}(\ell, \bar{s}, \bar{\alpha}).$$

This implies that maximizing $\mathbf{Pr}_{\bar{y},Q|\bar{\alpha}}[y_i = \ell \mid Q_i = \bar{s}]$ is equivalent to maximizing $\pi_i(\ell) \cdot \mathsf{conf}(\ell, \bar{s}, \bar{\alpha})$. Hence $\mathtt{NB}_{\bar{\pi}}$ achieves the optimal expected accuracy.

To obtain the second part of the claim we note that the expected accuracy of $\mathtt{NB}$ does not depend on the target labels $\bar{y}$ (the queries and the decision algorithm are invariant to an arbitrary permutation of labels at any point). That is for any $\bar{y}, \bar{y}' \in [m]^n$,

$$\mathop{\mathbf{E}}_{\bar{r}=\mathtt{NB}^{\mathcal{O}(\bar{y})}}[\mathsf{acc}_{\bar{y}}(\bar{r})] = \mathop{\mathbf{E}}_{\bar{r}=\mathtt{NB}^{\mathcal{O}(\bar{y}')}}[\mathsf{acc}_{\bar{y}'}(\bar{r})].$$

This means that the worst case accuracy of $\mathtt{NB}$ is the same as its average-case accuracy for labels drawn from the uniform distribution μ_m^n. In addition, $\mathtt{NB}_{\mu_m^n}$ is equivalent to $\mathtt{NB}$. Therefore,

$$\mathsf{acc}(\mathcal{A}) \leq \mathsf{acc}(\mathcal{A}, \mu_m^n) \leq \mathsf{acc}(\mathtt{NB}_{\mu_m^n}, \mu_m^n) = \mathsf{acc}(\mathtt{NB}).$$

□

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

The key to our proof of Theorem 4.4 is the following lemma about biased and Poissonized multinomial random variables that we prove in Appendix A.

Lemma 4.3. *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 $\operatorname{argmax}(\bar{c})$ 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 $\gamma \leq \frac{1}{8\sqrt{\lambda m}}$,*

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

Given this lemma the rest of the analysis follows quite easily.

Theorem 4.4. *For any $m \geq 2$, $n \geq k \geq k_{\min} = O(\ln n + m \ln m)$, we have that*

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

Proof. Let $\gamma = \frac{\sqrt{1-1/m}}{3\sqrt{mn}}$ and we consider a point-wise attack algorithm $\mathcal{B}$ that given a vector $\bar{s} \in [m]^k$ of query values at a point and a vector $\bar{\alpha}$ of accuracies computes the set of indices $J \subseteq [k]$, where $\alpha_j \geq \frac{1}{m} + \gamma$. We denote $t = |J|$. The algorithm then samples v from $\mathsf{Pois}(\lambda)$ for $\lambda = k/8$. If $v \leq t$ then let J' denote the first v elements in J, otherwise we let $J' = J$. $\mathcal{B}$ outputs the plurality label of labels in $\bar{s}_{J'} = (s_j)_{j\in J'}$.

To analyze the algorithm, we denote the distribution over $\bar{s}$, conditioned on the accuracy vector being $\bar{\alpha}$ and correct label of the point being y by $\rho(\bar{\alpha}, y)$. Our goal is to lower bound the success probability of $\mathcal{B}$

$$\mathop{\mathbf{Pr}}_{\bar{s}\sim\rho(\bar{\alpha},y)}[\mathcal{B}(\bar{s}, \bar{\alpha}) = y].$$

Lemma 4.1 implies that elements of $\bar{s}$ are independent and for every $j \in [k]$, s_j is equal to y with probability α_j and $\frac{1-\alpha_j}{m-1}$, otherwise. Therefore for every $j \in J$, s_j is biased by at least γ towards the correct label y. We will further assume that s_j is biased by exactly γ since larger bias can only increase the success probability of $\mathcal{B}$.

Now let $\delta = \mathbf{Pr}[v > t]$. The distribution of $|J'|$ is δ close in total variation distance to $\mathsf{Pois}(\lambda)$. By Lemma 4.3, this means that

$$\mathbf{Pr}[\mathsf{plu}(\bar{s}_{J'}) = y] \geq \frac{1}{m} + \Omega\left(\frac{\sqrt{k}\gamma}{\sqrt{m}}\right) - \delta = \frac{1}{m} + \Omega\left(\frac{\sqrt{k}}{m\sqrt{n}}\right) - \delta, \tag{2}$$

where we used the assumptions $k \geq k_{\min}$ and $n \geq k$ to ensure that the conditions $\lambda \geq 2m\ln(4m)$ and $\gamma \leq \frac{1}{8\sqrt{\lambda m}}$ hold.

Hence to obtain our result it remains to estimate δ. We view t as jointly distributed with $\bar{\alpha}$. Let ϕ denote the distribution of $\bar{\alpha}$ for $Q \sim \mu_n^{n\times k}$ and any vector $\bar{y}$. For every $j \in [k]$, α_j is distributed according to the binomial distribution $\mathsf{Bin}(n, 1/m)$. By using the Berry-Esseen theorem (Lemma 2.2), we obtain that

$$\mathbf{Pr}\left[\alpha_j \geq \frac{1}{m} + \frac{\sigma}{3}\right] \geq \mathbf{Pr}\left[\zeta \geq \frac{\sigma}{3}\right] - O\left(\sqrt{\frac{m}{n}}\right),$$

where $\sigma^2 = \frac{1-1/m}{mn}$ and ζ is normally distributed with mean 0 and variance σ^2. In particular, for sufficiently large n,

$$\mathbf{Pr}\left[\alpha_j \geq \frac{1}{m} + \gamma\right] \geq 1/3.$$

Now by Chernoff bound (Lemma 2.1), we obtain that for sufficiently large k,

$$\mathbf{Pr}\left[t \leq \frac{k}{4}\right] \leq e^{-k/96}.$$

In addition, by the concentration of $\mathsf{Pois}(k/8)$ (Lemma A.2) we obtain that

$$\mathbf{Pr}\left[v \geq \frac{k}{4}\right] \leq e^{-k/32}.$$

Therefore, by the union bound, $\delta \leq e^{-k/96} + e^{-k/32}$ and thus for $k \geq k_{\min}$ we will have that $\delta = o(1/n) = o(\sqrt{k}/(m\sqrt{n}))$. Plugging this into eq. (2) we obtain the claim. □

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 and Rényi [ER63] and generalized to arbitrary m by Chvátal [Chv83]. 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.

Proof. Let $\bar{y} \neq \bar{y}' \in [m]^n$ be an arbitrary pair of indices. We describe the difference between $\bar{y}$ and $\bar{y}'$ using the set of indices where they differ $I = \Delta(\bar{y}, \bar{y}') = \{i \mid y_i \neq y'_i\}$ and the vectors restricted to this set $\bar{y}_I = (y_i)_{i\in I}$ and $\bar{y}'_I = (y'_i)_{i\in I}$. It is easy to see from the definition that for any query $\bar{q}$,

$$\mathsf{acc}_{\bar{y}}(\bar{q}) - \mathsf{acc}_{\bar{y}'}(\bar{q}) = \frac{1}{n}\sum_{i\in I}\left(\mathsf{Ind}(y_i = q_i) - \mathsf{Ind}(y'_i = q_i)\right).$$

In particular, the difference is fully determined by $I = \Delta(\bar{y}, \bar{y}')$, $\bar{y}_I$ and $\bar{y}'_I$.

This implies that for a randomly chosen $\bar{q} \sim \mu_m^n$, $\mathsf{acc}_{\bar{y}}(\bar{q}) - \mathsf{acc}_{\bar{y}'}(\bar{q})$ is distributed as a sum of $w = |I|$ independent random variables from distribution that is equal $1/n$ with probability $1/m$, $-1/n$ with probability $1/m$ and 0 otherwise. Equivalently, this distribution can be seen as a sum

$$\frac{1}{n} \sum_{i \in [w]} b_i \sigma_i,$$

where each b_i is independent Bernoulli random variable with bias $2/m$ and each σ_i is an independent Rademacher random variable. We use v to denote the random variable

$$v = \sum_{i \in [w]} b_i \sigma_i$$

and let b denote the jointly distributed value

$$b = \sum_{i \in [w]} b_i.$$

We first deal with shift $\beta = 0$. For this we will first need to upper-bound the probability $p_w \doteq \mathbf{Pr}[v = 0]$. Note that conditioned on $b = j$, v is distributed as sum of j Rademacher random variables. Standard bounds on the central binomial coefficient imply that for even $j \geq 2$,

$$\mathbf{Pr}[v = 0 \mid b = j] \leq \frac{1}{\sqrt{j}}$$

and for odd j, $\mathbf{Pr}[v = 0 \mid b = j] = 0$. In particular, for all $j \geq 1$, $\mathbf{Pr}[v = 0 \mid b = j] \leq 1/2$.

This gives us that

$$\begin{aligned} \mathbf{Pr}[v = 0] &\leq \mathbf{Pr}[b = 0] + \frac{1}{2} \mathbf{Pr}[b > 1] \\ &= \frac{1}{2} + \frac{1}{2} \left(1 - \frac{2}{m}\right)^w \\ &\leq \frac{1}{2} + \frac{1}{2} e^{-\frac{2w}{m}}. \end{aligned} \tag{3}$$

Now using the multiplicative Chernoff bound we get that

$$\mathbf{Pr}\left[b \leq \frac{w}{m}\right] \leq e^{-\frac{w}{6m}}.$$

This implies that

$$\begin{aligned} \mathbf{Pr}[v = 0] &\leq \mathbf{Pr}\left[b < \frac{w}{m}\right] + \sqrt{\frac{m}{w}} \mathbf{Pr}\left[b \geq \frac{w}{m}\right] \\ &= e^{-\frac{w}{6m}} + \sqrt{\frac{m}{w}}. \end{aligned} \tag{4}$$

Given a matrix Q of k randomly and independently chosen queries we have

$$\mathop{\mathbf{Pr}}_{Q \sim \mu_m^{n \times k}} \left[\mathsf{acc}_{\bar{y}}(Q) = \mathsf{acc}_{\bar{y}'}(Q)\right] \leq p_w^k.$$

There are at most $\binom{n}{w}m^{2w}$ possible differences between a pair of vectors $\bar{y}, \bar{y}'$. Therefore, by the union bound for every w, probability that there exists a pair of vectors $\bar{y}, \bar{y}'$ that differ in w positions and for which the accuracies on all k queries are identical is at most

$$\binom{n}{w}m^{2w} \cdot p_w^k.$$

If $1 \leq w < 2m$ then eq. (3) implies that

$$p_w \leq \frac{1}{2} + \frac{1}{2}e^{-\frac{2w}{m}} \leq \frac{1}{2} + \frac{1}{2}\left(1 - \frac{w}{m}\right) \leq e^{-\frac{w}{2m}}$$

and our union bound is

$$\begin{aligned}
&\binom{n}{w}m^{2w} \cdot e^{-\frac{kw}{2m}} \\
&\qquad \leq \left(\frac{nem^2}{w}\right)^w \cdot e^{-\frac{kw}{2m}} \\
&\qquad \leq e^{w\ln(enm^2) - \frac{kw}{2m}} \\
&\qquad \leq \left(\frac{1}{4n^2}\right)^w \leq \frac{1}{2n^2},
\end{aligned}$$

where we used the condition that

$$k \geq 20m\ln(nm) \geq 2m\ln(2en^3m^2).$$

If $2m \leq w < 6m$ then eq. (3) implies that

$$p_w \leq \frac{1}{2} + \frac{1}{2}e^{-\frac{2w}{m}} \leq \frac{1}{2} + \frac{1}{2}e^{-1} \leq e^{-1/3}$$

and our union bound is

$$\begin{aligned}
&\binom{n}{w}m^{2w} \cdot e^{-\frac{k}{3}} \\
&\qquad \leq e^{w\ln(enm/2) - \frac{k}{3}}.
\end{aligned}$$

This bound is maximized for $w = 6m$ giving $e^{6m\ln(nm) - \frac{k}{3}}$. Using the condition

$$k \geq 20m\ln(nm) \geq 18m\ln(enm/2) + 3\ln(2n^2)$$

we get an upper bound of $\frac{1}{2n^2}$.

If $w \geq 6m$ then eq. (4) implies that $p_w \leq \sqrt{\frac{m}{4w}}$ and our union bound is

$$\binom{n}{w}m^{2w} \cdot \left(\sqrt{\frac{m}{4w}}\right)^k.$$

This bound is maximized for $w = n$, which gives an upper bound of

$$m^{2w} \cdot \left(\frac{n}{4m}\right)^{k/2} \leq \frac{1}{2n^2},$$

Algorithm 2 The reconstruction-based overfitting attack algorithm.

input Query access to a test set of n points over m labels, example budget $t \leq n$.
Draw k queries $R \in [m]^{t \times k}$ uniformly over $[m]^{t \times k}$.
Let $Q \in [m]^{n \times k}$ be the matrix that extends R by appending $n - t$ rows of ones.
Submit queries $Q^1, \ldots, Q^k$ and receive corresponding accuracies $\bar{\alpha} = (\alpha_1, \ldots, \alpha_k)$
Compute $\bar{z} = (z_1, \ldots, z_n) \in [m]^n$ as any vector satisfying $\mathsf{acc}_{\bar{z}}(Q) = \bar{\alpha}$.
Draw random predictions $z'_1, \ldots, z'_{n-t}$ uniformly over $[m]^{n-t}$.
output Predictions $(z_1, \ldots, z_t, z'_1, \ldots, z'_{n-t})$

where we used the condition that

$$k \geq \frac{5n \ln m}{\ln(n/4m)} \geq 2\frac{2n\ln(m) + \ln(2n^2)}{\ln(n/4m)}.$$

Now by using a union bound over all values of $w \in [n]$ we get that probability that there exist distinct $\bar{y}, \bar{y}' \in [m]^n$ such that $\mathsf{acc}_{\bar{y}}(Q) = \mathsf{acc}_{\bar{y}'}(Q)$ is at most $1/(2n)$. Now to deal with any other $\beta \in \{1/n, \ldots, 1\}$ (we only need to treat positive βs since the definition is symmetric) we observe that for $m \geq 3$ and any w,

$$\mathbf{Pr}[v = n\beta] \leq \mathbf{Pr}[v = 0] = p_w.$$

By the same argument this implies that probability that there exist distinct $\bar{y}, \bar{y}' \in [m]^n$ such that

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

is at most $1/(2n)$. Taking the union bound over all values of β we obtain the claim. □

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.

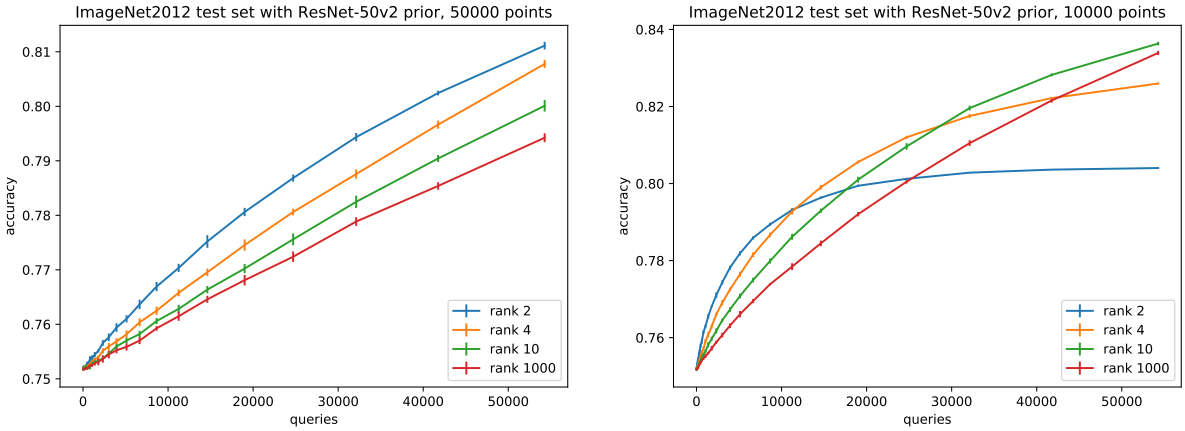
Corollary 4.6. *For any $k \geq 40m\ln(m)$, there exists an attack $\mathcal{A}$ with access to points such that*

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

Proof. We first let t be the largest value for which Thm. 4.5 guarantees existence of a set of queries of size k that allows to fully recover t labels from shifted accuracies. Using the bound from Thm. 4.5 we get that $t = \Omega\left(\frac{k\ln(k/m)}{\ln m}\right)$. If $t \geq n$ then we recover the labels and output them. Otherwise, let $R \in [m]^{t \times k}$ be the set of queries that recovers t labels and let $\bar{y}_{[t]}$ be the first t values of $\bar{y}$. We extend R to a set Q of queries over n labels by appending a fixed query $(1, 1, \ldots, 1)$ over the remaining $n - t$ coordinates.

Now to recover $\bar{y}_{[t]}$ we need to observe that, if there exists a vector $\bar{z} \in [m]^t$ such that

$$t \cdot \mathsf{acc}_{\bar{z}}(R) = n \cdot \mathsf{acc}_{\bar{y}}(Q) - (n - t)\beta(1, 1, \ldots, 1)$$



(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.

for some fixed value β, then $\bar{y}_{[t]} = \bar{z}$. This follows from the fact that

$$t \cdot \mathsf{acc}_{\bar{y}_{[t]}}(R) = n \cdot \mathsf{acc}_{\bar{y}}(Q) - (n-t)\beta'(1,1,\ldots,1),$$

where β' is the accuracy of all 1 labels on the last $n-t$ coordinates of $\bar{y}$. This implies that

$$\mathsf{acc}_{\bar{z}}(R) = \mathsf{acc}_{\bar{y}_{[t]}}(R) + \left(\frac{n-t}{t}\right)(\beta' - \beta)(1,1,\ldots,1).$$

By the property of R this implies that $\bar{z} = \bar{y}_{[t]}$. Having found $\bar{z} = \bar{y}_{[t]}$ we output a labeling that is equal to $\bar{z}$ on the first t labels and is random and uniform over the rest. The expected accuracy of this labeling is

$$\frac{t}{n} + \frac{1}{m}\left(1 - \frac{t}{n}\right) = \frac{1}{m} + \frac{t}{n}\frac{m-1}{m} = \frac{1}{m} + \Omega\left(\frac{k\ln(k/m)}{n\ln m}\right).$$

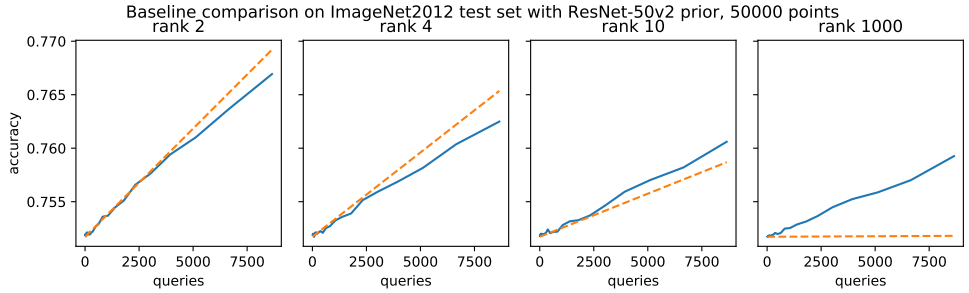
□

The resulting reconstruction-based attack is summarized as Algorithm 2.

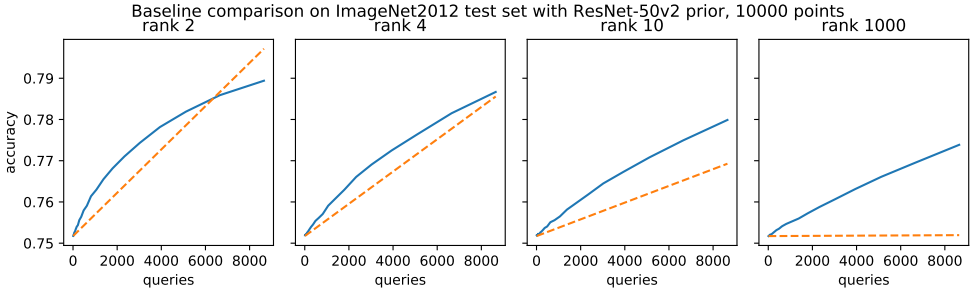
5 Experimental evaluation

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.



(a) Full test set



(b) 10K least confident points

Figure 2: The average accuracies depicted in Figure 1 in comparison with an analytical baseline: the expected performance of the linear scan attack at the same number of queries.

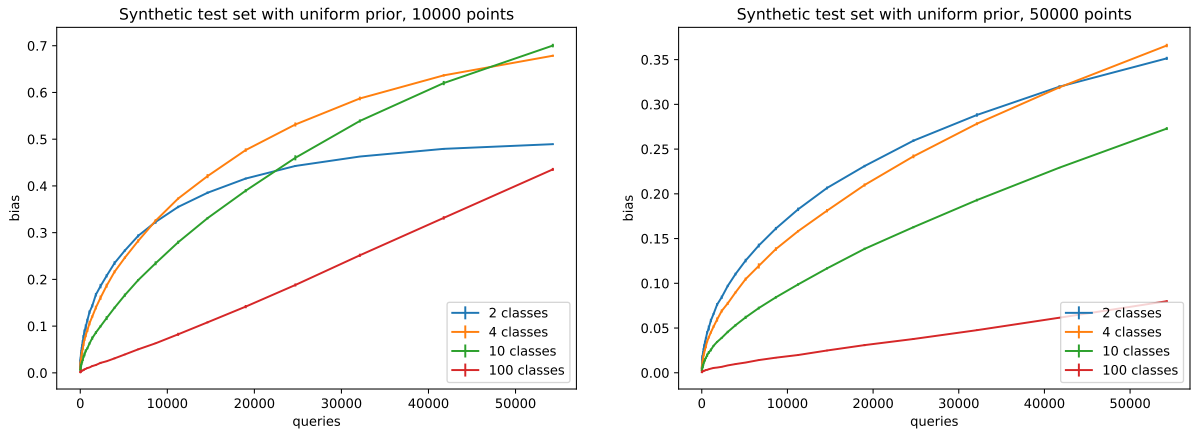
Figure 3 shows the observed advantage of the attack over the population error rate of $1/m$, across a range of query budgets, on test sets of size 10,000 and 50,000 respectively.[1]

Figure 4 shows the number of queries at which a fixed advantage over $1/m$ is first attained, while the number of class labels m varies, on a test set of size 100,000. To maintain a fixed value of the bound in Theorem 4.4, an increase in the number of classes m requires a quadratic increase in the number of queries k. The endpoints of the curves in Figure 4, on a log-log scale, form lines of slope greater than 1, supporting the conjecture that, to attain a fixed bias, the number of queries k grows superlinearly with m.

In the binary classification setting, we compare to the majority-based attack proposed by Blum and Hardt [BH15], 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. Figure 5 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.

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



(a) 10K random points

(b) 50K random points

Figure 3: Average bias (with negligible standard deviation bars), over 10 attack trials, of the NB attack against uniformly random test sets comprising different class counts. Note that the maximum achievable bias, which is $1 - 1/m$ for m classes, differs for each curve.

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).

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 Figure 2: 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.

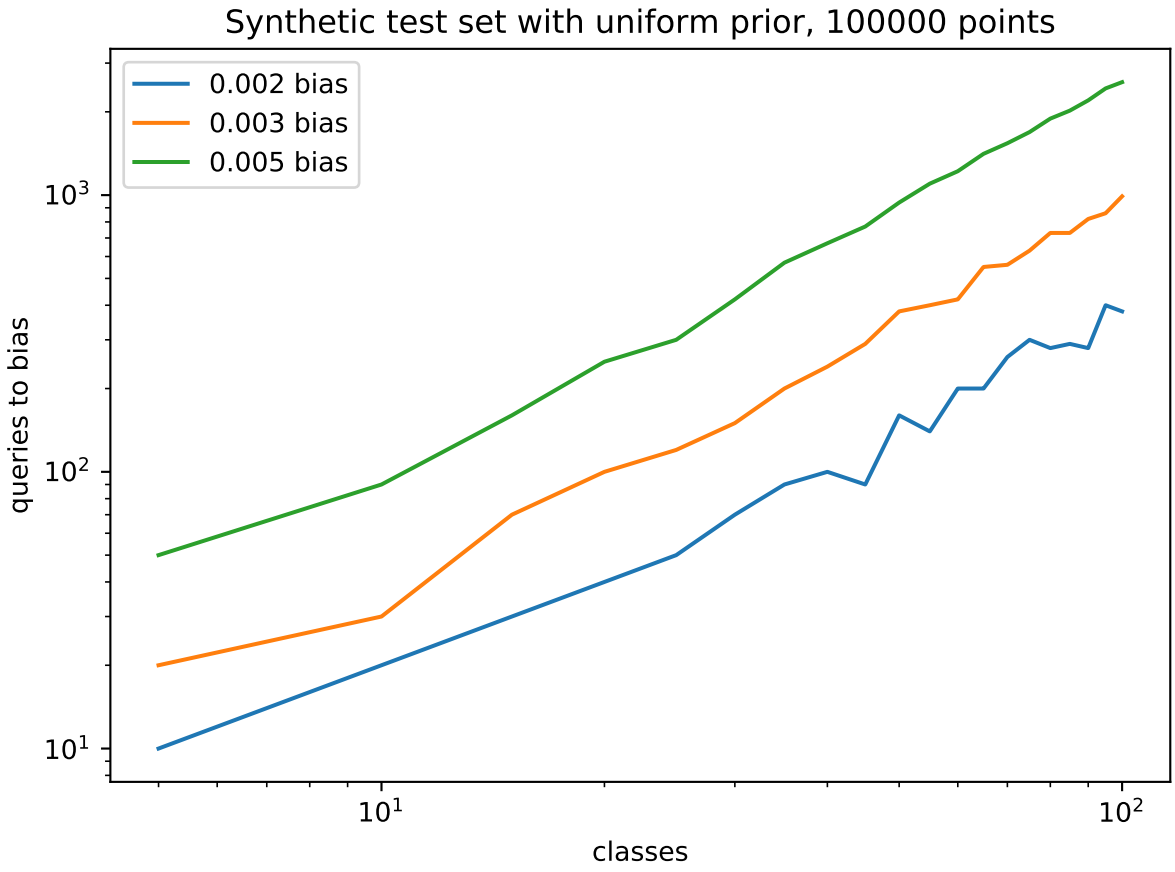


Figure 4: The number of queries at which a fixed advantage over $1/m$ is first attained, while the number of class labels m varies, on a randomly generated test set of size 100,000. The endpoints of the curves form slopes (under the log-log axis scaling) of roughly 1.2 (for the 0.002 bias curve) and 1.3 (for the other curves), suggesting that, to attain a fixed bias, the number of queries k must indeed grow superlinearly with m, as supported by the bound in Theorem 4.4.

Acknowledgements

We thank Clément Canonne for his suggestion to use Poissonization in the proof of Theorem 4.4. 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.

References

[Bas+16] R. Bassily, K. Nissim, A. D. Smith, T. Steinke, U. Stemmer, and J. Ullman. "Algorithmic stability for adaptive data analysis". In: *STOC*. 2016, pp. 1046–1059.

[BH15] A. Blum and M. Hardt. "The Ladder: A Reliable Leaderboard for Machine Learning Competitions". In: *CoRR* abs/1502.04585 (2015).

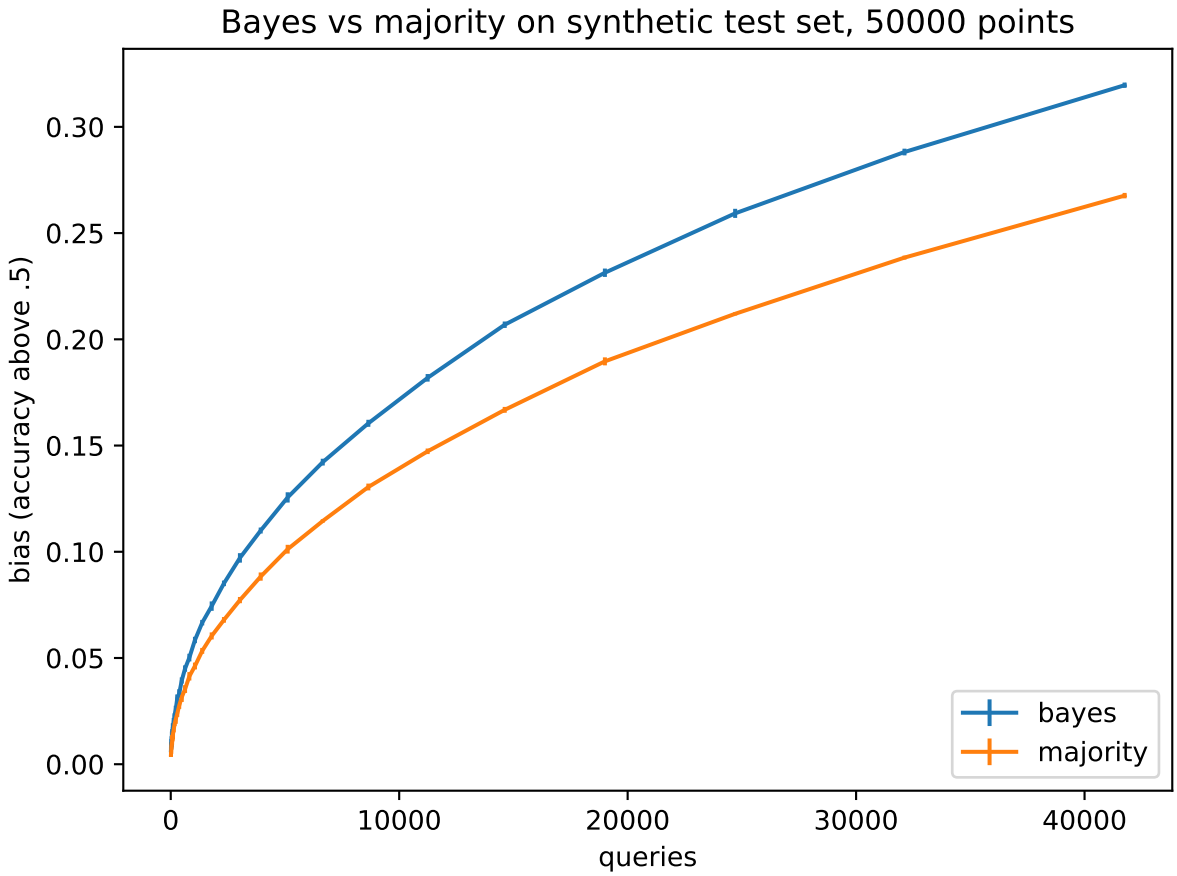


Figure 5: Average bias (with negligible standard deviation bars), over 10 attack trials, of two attacks—`NB` and the majority attack of Blum and Hardt [BH15]—against uniformly random binary-labeled test sets.

[Bsh09] N. H. Bshouty. "Optimal Algorithms for the Coin Weighing Problem with a Spring Scale". In: *COLT*. 2009.

[Can17] C. Canonne. *A short note on Poisson tail bounds*. `https://github.com/ccanonne/probabilitydistributiontoolbox/blob/master/poissonconcentration.pdf`. 2017.

[Chv83] V. Chvátal. "Mastermind". In: *Combinatorica* 3.3 (1983), pp. 325–329.

[DMT07] C. Dwork, F. McSherry, and K. Talwar. "The price of privacy and the limits of LP decoding". In: *Proceedings of STOC*. ACM. 2007, pp. 85–94.

[DN03] I. Dinur and K. Nissim. "Revealing information while preserving privacy". In: *PODS*. 2003, pp. 202–210.

[Doe+16] B. Doerr, C. Doerr, R. Spöhel, and H. Thomas. "Playing mastermind with many colors". In: *Journal of the ACM (JACM)* 63.5 (2016), p. 42.

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

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

[Dwo+15a] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. "Generalization in Adaptive Data Analysis and Holdout Reuse". In: *CoRR* abs/1506 (2015). Extended abstract in NIPS 2015.

[Dwo+15b] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. "The reusable holdout: Preserving validity in adaptive data analysis". In: *Science* 349.6248 (2015), pp. 636–638. eprint: `http://www.sciencemag.org/content/349/6248/636.full.pdf`.

[ER63] P. Erdos and A. Rényi. "On two problems of information theory". In: *Magyar Tud. Akad. Mat. Kutató Int. Közl* 8 (1963), pp. 229–243.

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

[Har17] M. Hardt. "Climbing a shaky ladder: Better adaptive risk estimation". In: *CoRR* abs/1706.02733 (2017). arXiv: `1706.02733`.

[Has+09] T. Hastie, S. Rosset, J. Zhu, and H. Zou. "Multi-class adaboost". In: *Statistics and its Interface* 2.3 (2009), pp. 349–360.

[HU14] M. Hardt and J. Ullman. "Preventing False Discovery in Interactive Data Analysis Is Hard". In: *FOCS*. 2014, pp. 454–463.

[Kas+10] S. P. Kasiviswanathan, M. Rudelson, A. Smith, and J. Ullman. "The price of privately releasing contingency tables and the spectra of random matrices with correlated rows". In: *Proceedings of STOC*. ACM. 2010, pp. 775–784.

[KRS13] S. P. Kasiviswanathan, M. Rudelson, and A. Smith. "The power of linear reconstruction attacks". In: *Proceedings of SODA*. SIAM. 2013, pp. 1415–1433.

[Rec+18] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. "Do CIFAR-10 Classifiers Generalize to CIFAR-10?" In: *CoRR* abs/1806.00451 (2018).

[Rec+19] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. "Do ImageNet Classifiers Generalize to ImageNet?" In: *CoRR* abs/1902.10811 (2019).

[Rus+15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. "Imagenet large scale visual recognition challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.

[Sha60] H. S. Shapiro. "Problem E 1399". In: *American Mathematical Monthly* 67 (1960), p. 82.

[SU15] T. Steinke and J. Ullman. "Interactive Fingerprinting Codes and the Hardness of Preventing False Discovery". In: *COLT*. 2015, pp. 1588–1628.

[Ver15] R. Vershynin. "Estimation in high dimensions: a geometric perspective". In: *Sampling theory, a renaissance*. Springer, 2015, pp. 3–66.

[Wik] Wikipedia. *Mastermind (board game)*.

[YB19] C. Yadav and L. Bottou. "Cold Case: The Lost MNIST Digits". In: *arXiv* 1905.10498 (2019).

A Proof of Lemma 4.3

We start with some definitions and properties of the Poisson distribution that we will need in the proof.

A Poisson random variable V, with parameter λ, is the random variable that for all non-negative integers t, satisfies $\mathbf{Pr}[V = t] = e^{-\lambda}\frac{\lambda^t}{t!}$. We denote its density by $\mathsf{Pois}(\lambda)$. For $U \sim \mathsf{Pois}(\lambda_1)$ and $U \sim \mathsf{Pois}(\lambda_2)$, $U + V$ is distributed according to $\mathsf{Pois}(\lambda_1 + \lambda_2)$.

We will use the following result referred to as Poissonization of a multinomial random variable.

Fact A.1. *Let $\rho(\bar{p})$ be a categorical distribution over $[m]$ defined by a vector of probabilities $\bar{p} = (p_1, \ldots, p_m)$ and let $\mathsf{Mnom}(k, \bar{p})$ be the multinomial distribution over counts corresponding to k independent draws from $\rho(\bar{p})$. Then for any $\lambda > 0$ and $V \sim \mathsf{Pois}(\lambda)$ we have that $\mathsf{Mnom}(V, \bar{p})$ is distributed as*

$$\mathsf{Pois}(p_1\lambda) \times \mathsf{Pois}(p_2\lambda) \times \cdots \times \mathsf{Pois}(p_m\lambda).$$

We will need a relatively tight bound on the concentration of a Poisson random variable. Its simple proof can be found, for example, in a note byCanonne [Can17].

Lemma A.2 ([Can17])**.** *For any $\lambda > 0, x \geq 0$,*

$$\mathop{\mathbf{Pr}}_{V\sim\mathsf{Pois}(\lambda)}[V \geq \lambda + x] \leq e^{-(\lambda+x)\ln\left(1+\frac{x}{\lambda}\right)-x} \text{ and}$$

$$\mathop{\mathbf{Pr}}_{V\sim\mathsf{Pois}(\lambda)}[V \leq \lambda - x] \leq e^{-(\lambda-x)\ln\left(1-\frac{x}{\lambda}\right)-x}.$$

In particular,

$$\mathop{\mathbf{Pr}}_{V\sim\mathsf{Pois}(\lambda)}[V \geq \lambda + x] \leq e^{\frac{-x^2}{2(\lambda+x)}} \text{ and}$$

$$\mathop{\mathbf{Pr}}_{V\sim\mathsf{Pois}(\lambda)}[V \leq \lambda - x] \leq e^{\frac{-x^2}{2(\lambda+x)}}.$$

Using this concentration inequality we show that the density of the Poisson random variable can be related in a tight way to the corresponding tail probability.

Lemma A.3. *For any $\lambda > 0$ and integer $t \geq 0$ and $x = |t - \lambda|$,*

$$\mathop{\mathbf{Pr}}_{V\sim\mathsf{Pois}(\lambda)}[V = t] \geq \frac{e^{-t\ln\left(\frac{t}{\lambda}\right)-x}}{e\sqrt{t}}.$$

In particular, for $t \geq \lambda$,

$$\mathop{\mathbf{Pr}}_{V\sim\mathsf{Pois}(\lambda)}[V = t] \geq \frac{\mathbf{Pr}_{V\sim\mathsf{Pois}(\lambda)}[V \geq t]}{e\sqrt{t}}$$

and $t \leq \lambda$,

$$\mathop{\mathbf{Pr}}_{V\sim\mathsf{Pois}(\lambda)}[V = t] \geq \frac{\mathbf{Pr}_{V\sim\mathsf{Pois}(\lambda)}[V \leq t]}{e\sqrt{t}}.$$

Proof. If $t \geq \lambda$ (and $x = t - \lambda$) then by definition and using Stirling's approximation of the factorial we get:

$$\begin{aligned}\mathop{\mathbf{Pr}}_{V\sim\mathsf{Pois}(\lambda)}[V = t] &= e^{-\lambda}\frac{\lambda^t}{t!} \geq e^{-\lambda}\frac{\lambda^t}{e\sqrt{t}e^{-t}t^t} \\ &= \frac{e^x}{e\sqrt{t}}\left(\frac{\lambda}{\lambda+x}\right)^{\lambda+x} = \frac{e^x}{e\sqrt{t}}\left(\frac{\lambda+x}{\lambda}\right)^{-(\lambda+x)} \\ &= \frac{1}{e\sqrt{t}}e^{-(\lambda+x)\ln\left(1+\frac{x}{\lambda}\right)-x} \\ &\geq \frac{\mathbf{Pr}_{V\sim\mathsf{Pois}(\lambda)}[V \geq \lambda + x]}{e\sqrt{t}},\end{aligned}$$

where we used Lemma A.2 to obtain the last inequality. The case when $t \leq \lambda$ is proved analogously. □

We are now ready to prove Lemma 4.3 which we restate here for convenience.

Lemma A.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 $\gamma \leq \frac{1}{8\sqrt{\lambda m}}$,*

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

Proof. Let $\bar{c} = (c_1, \ldots, c_m)$ denote the vector of label counts sampled from $\mathsf{Mnom}(t, \rho_\gamma)$ for t sampled randomly from $\mathsf{Pois}(\lambda)$. We first use Fact A.1 to conclude that $\bar{c}$ is distributed according to

$$\mathsf{Pois}(\lambda') \times \cdots \times \mathsf{Pois}(\lambda') \times \mathsf{Pois}\left(\lambda' + \frac{\gamma\lambda m}{m-1}\right)$$

for $\lambda' = \left(\frac{1}{m} - \frac{\gamma}{m-1}\right)\lambda$.

The next step is to reduce the problem to that of analyzing the product distribution of identical Poisson random variables. Specifically, we view the count of the "true" label m as the sum of two independent Poisson random variables $c'_m \sim \mathsf{Pois}(\lambda')$ and $d_m \sim \mathsf{Pois}\left(\frac{\gamma\lambda m}{m-1}\right)$. We also denote by $\bar{c}'$ the vector $(c_1, \ldots, c_{m-1}, c'_m)$. Note that $\bar{c}'$ consists of independent and identically distributed samples from $\mathsf{Pois}(\lambda')$.

Let $z = \max_{j\in[m-1]} c_j$. By definition, if $c_m > z$ then $\mathrm{argmax}(\bar{c}) = m$ and if $c_m = z$ then $\mathbf{Pr}[\mathrm{argmax}(\bar{c}) = m] \leq 1/2$, where the probability is taken solely with respect to the random choice of the index that maximizes the count. This implies that if $d_m \geq 1$ then

$$\mathbf{Pr}[\mathrm{argmax}(\bar{c}) = m] \geq \mathbf{Pr}[\mathrm{argmax}(\bar{c}') = m] + \frac{1}{2}\mathsf{Ind}(c'_m \in [z - d_m + 1, z]).$$

Now taking the probability over the random choice of $\bar{c}$ we get

$$\mathbf{Pr}[\mathrm{argmax}(\bar{c}) = m] \geq \mathbf{Pr}[\mathrm{argmax}(\bar{c}') = m] + \frac{1}{2}\mathbf{Pr}[c'_m \in [z - d_m + 1, z]]. \tag{5}$$

By symmetry of the distribution of $\bar{c}'$ we obtain that $\mathbf{Pr}[\mathrm{argmax}(\bar{c}') = m] = \frac{1}{m}$. To analyze the second term, we first consider the case where $\mathbf{E}[d_m] = \frac{\gamma\lambda m}{m-1} \leq 1$. For this case we simply bound

$$\mathbf{Pr}[c'_m \in [z - d_m + 1, z]] \geq \mathbf{Pr}[c'_m = z] \cdot \mathbf{Pr}[d_m \geq 1] \tag{6}$$

(recall that d_m and c'_m are independent).

By definition of $\mathsf{Pois}\left(\frac{\gamma\lambda m}{m-1}\right)$ we obtain that

$$\mathbf{Pr}[d_m \geq 1] = 1 - e^{-\gamma\lambda m/(m-1)} \geq \frac{\lambda\gamma m}{2(m-1)} \geq \frac{\lambda\gamma}{2}, \tag{7}$$

where we used the fact that $e^{-a} \leq 1 - a/2$ whenever $a \leq 1$ and our assumption that $\frac{\gamma\lambda m}{m-1} \leq 1$.

Hence it remains to lower bound $\mathbf{Pr}[c'_m = z]$. To this end, let u and v be the $1/2$ and $1 - 1/(4m)$ quantiles of $\mathsf{Pois}(\lambda')$, respectively. That is, $u = \max\{t \mid \mathbf{Pr}_{V\sim\mathsf{Pois}(\lambda')}[V \geq t] \geq 1/2\}$ and $v = \max\{t \mid \mathbf{Pr}_{V\sim\mathsf{Pois}(\lambda')}[V \geq t] \geq 1 - 1/(4m)\}$. By the union bound, $\mathbf{Pr}[z \geq v + 1] \leq \frac{m-1}{4m} < 1/4$. In addition, by the standard properties of Poisson distribution $\mathbf{Pr}_{V\sim\mathsf{Pois}(\lambda')}[V \geq \lfloor\lambda'\rfloor] \geq 1/2$ which implies that $\mathbf{Pr}[z \geq \lfloor\lambda'\rfloor] \geq 1/2$ and thus $u \geq \lfloor\lambda'\rfloor$.

Thus we have an interval such that

$$\mathbf{Pr}[z \in [u, v]] \geq \frac{1}{4}.$$

By Lemma A.3, we have that for every $t \in [u, v]$,

$$\mathbf{Pr}[c'_m = t] = \mathop{\mathbf{Pr}}_{V\sim\mathsf{Pois}(\lambda')}[V = t] \geq \frac{\mathbf{Pr}_{V\sim\mathsf{Pois}(\lambda')}[V \geq t]}{e\sqrt{t}} \geq \frac{1}{4e\sqrt{v}m}. \tag{8}$$

Observe that by our assumption, $\lambda \geq 2\ln(4m)$ and $\frac{\gamma}{m-1}\lambda \leq \lambda'/2$. Hence $\lambda' \geq \ln(4m)$. By Lemma A.2 this implies that

$$v \leq \lambda' + 3\sqrt{\lambda'\ln(4m)} \leq 4\lambda'.$$

Using the independence of z and c'_m we can conclude that

$$\mathbf{Pr}[c'_m = z] \geq \mathbf{Pr}[z \in [u, v]] \cdot \min_{t\in\{u,u+1,\ldots,v\}} \mathbf{Pr}[c'_m = t] \geq \frac{1}{4} \cdot \frac{1}{4e\sqrt{v}m} \geq \frac{1}{32e\sqrt{\lambda'}m}.$$

By combining this bound with eq.(7), plugging it into eq.(6) and recalling that $\lambda' = \left(\frac{1}{m} - \frac{\gamma}{m-1}\right)\lambda \geq \frac{\lambda}{2m}$ we obtain that

$$\mathbf{Pr}[\mathrm{argmax}(\bar{c}) = m] \geq \frac{1}{m} + \frac{1}{2} \cdot \frac{\lambda\gamma}{2} \cdot \frac{1}{32e\sqrt{\lambda'}m} = \frac{1}{m} + \Omega\left(\frac{\sqrt{\lambda}\gamma}{\sqrt{m}}\right).$$

We now consider the other case where $\mathbf{E}[d_m] = \frac{\gamma\lambda m}{m-1} > 1$ which requires a similar but somewhat more involved treatment. We first note that we can assume that $\frac{\gamma\lambda m}{m-1} \geq 12$. For the case when $\frac{\gamma\lambda m}{m-1} \in [1, 12]$ we note that it holds that

$$\mathbf{Pr}[d_m \geq 1] \geq 1 - e^{-1} > \frac{1}{20}\gamma\lambda.$$

Thus we can still use the same analysis as before to obtain our claim. By Lemma A.2, for $\nu \geq 12$, $\mathbf{Pr}_{V\sim\mathsf{Pois}(\nu)}[V \in [\nu/2, 2\nu]] \geq 1/2$. In particular, under the assumption that $\frac{\gamma\lambda m}{m-1} \geq 12$,

$$\mathbf{Pr}\left[d_m \in \left[\frac{\gamma\lambda m}{2(m-1)}, \frac{2\gamma\lambda m}{m-1}\right]\right] \geq \frac{1}{2}.$$

We also define u and v as before. Using independence of z and d_m we obtain that with probability at least $\frac{1}{4} \cdot \frac{1}{2}$, we have that $d_m \in \left[\frac{\gamma\lambda m}{2(m-1)}, \frac{2\gamma\lambda m}{m-1}\right]$ and $z \in [u, v-1]$. In particular, with probability at least $1/8$, $d_m \geq \frac{\gamma\lambda m}{2(m-1)} > \gamma\lambda/2$ and $[z - d_m + 1, z] \subseteq [u', v']$, where $u' = u - \frac{2\gamma\lambda m}{m-1}$ and $v' = v - \frac{\gamma\lambda m}{2(m-1)}$. The interval $[z - d_m + 1, z]$ includes d_m integer points and therefore

$$\mathbf{Pr}[c'_m \in [z - d_m + 1, z]] \geq \frac{1}{8} \cdot \frac{\gamma\lambda}{2} \cdot \min_{t \in \{u', u'+1, \ldots, v'\}} \mathbf{Pr}[c'_m = t]. \tag{9}$$

To analyze the lowest value of the probability mass function of $\mathsf{Pois}(\lambda')$ on the integers in the interval $[u', v']$ we first note that $v' \leq v$ and thus for $t \in [\lambda', v']$ our bound in eq. (8) applies. For $t \in [u', \lambda')$ we will first prove that under the assumptions of the lemma $u' \geq \lambda' - \sqrt{\lambda'}$ and then show that for $t \in [\lambda' - \sqrt{\lambda'}, \lambda')$, $\mathbf{Pr}[c'_m = t] \geq \frac{1}{e^2\sqrt{\lambda'}}$. Plugging this lower bound together with the one in eq. (8) into eq. (9) we obtain the claim:

$$\mathbf{Pr}[c'_m \in [z - d_m + 1, z]] \geq \frac{\gamma\lambda}{16} \cdot \min\left\{\frac{1}{8e\sqrt{\lambda'}m}, \frac{1}{e^2\sqrt{\lambda'}}\right\} = \Omega\left(\frac{\sqrt{\lambda}\gamma}{\sqrt{m}}\right).$$

We now complete these two missing steps. By our definition of u',

$$u' \geq \lfloor \lambda' \rfloor - \frac{2\gamma\lambda m}{m-1} \geq \lambda' - 4\gamma\lambda - 1 \geq \lambda' - \sqrt{\lambda'}$$

which follows from the assumption that $\gamma \leq \frac{1}{8\sqrt{\lambda m}}$ and assumption $\lambda \geq 2m\ln(4m)$ implying that $\lambda' > 5$. Now by Lemma A.3 and, using the monotonicity of the pmf of $\mathsf{Pois}(\lambda')$ until λ', we have that for $t \in [\lambda' - \sqrt{\lambda'}, \lambda')$,

$$\begin{aligned}
\mathbf{Pr}[c'_m = t] &\geq \frac{e^{-(\lambda' - \sqrt{\lambda'})\ln\left(\frac{\lambda' - \sqrt{\lambda'}}{\lambda'}\right) - \sqrt{\lambda'}}}{e\sqrt{\lambda' - \sqrt{\lambda'}}} \\
&\geq \frac{e^{(\lambda' - \sqrt{\lambda'})\left(\frac{1}{\sqrt{\lambda'}} - \frac{1}{2\lambda'}\right) - \sqrt{\lambda'}}}{e\sqrt{\lambda' - \sqrt{\lambda'}}} \\
&= \frac{e^{-\frac{1}{2} + \frac{1}{2\sqrt{\lambda'}}}}{e\sqrt{\lambda' - \sqrt{\lambda'}}} \geq \frac{1}{e^2\sqrt{\lambda'}},
\end{aligned}$$

where we used the Taylor series of $\ln(1-x) = -x - x^2/2 - x^3/3 - \ldots$ to obtain the second line. □