Optimal multiclass overfitting by sequence reconstruction from Hamming queries

Jayadev Acharya ACHARYA@CORNELL.EDU
Cornell University

Ananda Theertha Suresh THEERTHA@GOOGLE.COM
Google Research, New York

Editors: Aryeh Kontorovich and Gergely Neu

Abstract

A primary concern of excessive reuse of test datasets in machine learning is that it can lead to overfitting. Multiclass classification was recently shown to be more resistant to overfitting than binary classification (Feldman et al., 2019a). In an open problem of COLT 2019, Feldman, Frostig, and Hardt ask to characterize the dependence of the amount of overfitting bias with the number of classes m, the number of accuracy queries k, and the number of examples in the dataset n. We resolve this problem and determine the amount of overfitting possible in multiclass classification. We provide computationally efficient algorithms that achieve overfitting bias of $\tilde{\Theta}(\max\{\sqrt{k/(mn)}, k/n\})$, matching the known upper bounds.

Keywords: Overfitting, dataset reuse, multiclass classification

1. Introduction

Training multiple machine learning models on the same training set leads to overfitting. A common method to overcome this is to divide the dataset into a training set, and a *holdout* (or *test*) set, where the model's accuracy on the holdout set is used as an indicator of the true generalization capability of the model (James et al., 2013). However, even the holdout dataset is used multiple times, and this leads to overfitting of models even when a holdout dataset is used (Blum and Hardt, 2015). In essence, training over the same dataset again and again can lead to the fake illusion of learning, all the while only making the performance over the true data distribution worse for future instances. As benchmark datasets such as MNIST, ImageNet, and others are trained by more and more machine learning algorithms in an adaptive fashion, where new models can be dependent on the previous models and their performance on the dataset, overfitting is an increasingly growing concern.

It was recently shown that for binary classification, after k interactive rounds with the test dataset it is possible to overfit the dataset by achieving an accuracy $\Theta(\sqrt{k/n})$ larger than the true accuracy of the algorithm (Dwork et al., 2015c). To counter this, several alternative mechanisms were proposed, such as addition of noise to the true accuracy of the predictions in each round (Dwork et al., 2015b) or revealing the accuracy of prediction in a round only if it beats all previously achieved accuracies (Blum and Hardt, 2015). Zrnic and Hardt (2019) showed improved bounds on the performance when the adaptive analyst satisfies certain constraints. This line of word broadly falls in the field of *adaptive data analysis* (Dwork et al., 2015b,c,a; Bassily et al., 2016).

However, datasets such as MNIST, CIFAR10, and ImageNet have been largely immune to overfitting, even when the feedback obtained is the true accuracy (Recht et al., 2018, 2019; Yadav and

Range of k	Previous work (Feldman et al., 2019a)	Our results
$k = O(\frac{m}{n})$	$\tilde{\Omega}\left(\sqrt{\frac{k}{m^2 n}}\right) \leq \mathrm{acc}(k,n,m) - \frac{1}{m} \leq \tilde{O}\left(\sqrt{\frac{k}{mn}}\right)$ $\mathrm{poly}(n,k,m)$ run time	$\mathrm{acc}(k,n,m) = \frac{1}{m} + \tilde{\Theta}\left(\sqrt{\frac{k}{mn}}\right)$ $\mathrm{poly}(n,k,m)$ run time
$k = \Omega(\frac{m}{n})$	$\mathrm{acc}(k,n,m) = \frac{1}{m} + \tilde{\Theta}\left(\frac{k}{n}\right)$ $\mathrm{poly}(m,k) \cdot \exp(k)$ run time	$\mathrm{acc}(k,n,m) = \frac{1}{m} + \tilde{\Theta}\left(\frac{k}{n}\right)$ $\mathrm{poly}(n,k,m)$ run time

Table 1: Results on the overfitting bias of accuracy query based algorithms.

Bottou, 2019). Recht et al. (2019) also noted that adaptivity has negligible effect on overfitting. Feldman et al. (2019a) noted that even if these datasets are not very large, they are multiclass classification problems, where the number of possible labels is large. They considered the problem of largest overfitting possible for a multiclass classification problem, generalizing the binary results stated above, where in addition to k, and n, they consider the role of the number of classes, henceforth denoted by m. In other words, they studied the following problem:

Given n, k, and m, by how much can adaptive algorithms overfit the test dataset?

1.1. Prior and new results

The question of perfect test label reconstruction dates back decades and is related to the famous game mastermind (Erdös and Rényi, 1963; Chvátal, 1983; Doerr et al., 2016). The goal here is to exactly reconstruct a sequence by making k predictions, and for each prediction observing how many locations are correct. The optimal value of k to perfectly reconstruct was resolved by Chvátal (1983). An efficient algorithm for $m = 2$ (binary sequences) was proposed by Bshouty (2009). In our setting k is much smaller than needed to perfect reconstruction, and we want to understand the guarantees on how many locations can be predicted from the queries.

The general question of characterizing the overfitting bias as a function of k, n, m was considered by Feldman et al. (2019a). They proved an information theoretic upper bound of $1/m + \tilde{O}(\max\{\sqrt{k/(mn)}, k/n\})$ on the maximum possible accuracy. Note that the two terms dominate in the ranges $k = O(n/m)$ and $k = \Omega(n/m)$ respectively.

For $k = O(n/m)$, they designed an algorithm with accuracy of $1/m + \Omega\left(\sqrt{k/(m^2 n)}\right)$. This leaves the correct relation with m open, up to a quadratic factor. In other words for a given n it is unclear whether the number of queries k needed to achieve the same accuracy should grow linearly, quadratically or somewhere in between, as a function of the number of classes m. This question's resolution was the open problem in Feldman et al. (2019b), where they also mention that from a practical viewpoint, we should give particular emphasis on computationally efficient algorithms, although even the characterization of the overfitting bias is unknown. Our main result resolves this question by proposing a computationally efficient algorithm that has an accuracy of $1/m + \Omega\left(\sqrt{k/(mn)}\right)$ for $k = O(n/m)$, matching the information-theoretic upper bound up to a logarithmic factor. The precise statement is given in Theorem 3.

For $k = \Omega(n/m)$, Feldman et al. (2019a) proposed uniformly random queries over a subset of labels, and a final prediction that is not computationally efficient and achieves an accuracy of $1/m + \tilde{\Theta}(k/n)$, matching the upper bound up to logarithmic factors. For queries similar to theirs, we provide a computationally efficient final prediction that also has the optimal accuracy. We remark

that all the k queries of our optimal algorithms are non-adaptive, and only the final predictions depend on them. Thus adaptive queries do not help. A summary of our results is given in Table 1.

Finally, our proposed algorithm for small values of k and the information theoretic bounds of Feldman et al. (2019a) differ by a factor of $O(\sqrt{\log n})$. This previously known information theoretic upper bound uses minimum description length argument. For $k = 1$, by a careful analysis of the geometry of the problem, we remove the $\sqrt{\log n}$ factor in Theorem 9, thus showing the optimal overfitting bias up to constant factors. It would be interesting to see if this can be extended to other values of k.

Organization. The rest of the paper is organized as follows. In Section 2 we give a formal problem description. In Section 2.1 we consider a simplification where the test features are known, and pose it as a sequence reconstruction problem. For this sequence reconstruction problem, in Section 3, we show that it suffices to design algorithms where the labels are drawn from the uniform distribution on $[m]$, which allows us to only consider algorithms for this case, and in Section 4 we provide an overview of our algorithms, and in Section 5 and Section 6, we detail the algorithms and prove the results for $k = O(n/m)$ and $k = \Omega(n/m)$ respectively. Finally, in Section 7, we solve the question in its generality where the test features can be unknown.

2. Problem formulation

Let $\mathcal{X}$ denote the feature space, and $\mathcal{Y} = [m] := \{1, 2, \ldots, m\}$ be the set of labels. Let $S := \{(x_1, z_1), \ldots, (x_n, z_n)\}$ be a test set with n examples with $(x_i, z_i) \in \mathcal{X} \times \mathcal{Y}$. A classifier f is a (possibly randomized) mapping from $\mathcal{X}$ to $\mathcal{Y}$, and the accuracy of f on S is

$$\mathrm{acc}_S(f) := \frac{1}{n} \sum_{j=1}^{n} \mathbb{I}_{f(x_j)=z_j}.$$

As is most common in machine learning, we consider query access to the accuracy on the dataset S. Each query consists of a function $f^i : \mathcal{X} \to \mathcal{Y}$, and the accuracy oracle returns $\mathrm{acc}_S(f^i)$. A *k-query algorithm* $\mathcal{A}$ makes k queries $f^1, \ldots, f^k$ to S, and based on $f^1, \ldots, f^k$ and $\mathrm{acc}_S(f^1), \ldots, \mathrm{acc}_S(f^k)$, outputs a classifier $\hat{f} = \mathcal{A}^S$. The queries are allowed to be randomized and adaptive, namely f^i can depend on $f^1, \ldots, f^{i-1}$ and on $\mathrm{acc}_S(f^1), \ldots, \mathrm{acc}_S(f^{i-1})$. The accuracy of $\mathcal{A}$ is

$$\mathrm{acc}(\mathcal{A}, S) := \mathbb{E}_{\hat{f}=\mathcal{A}^S} \left[\mathrm{acc}(\hat{f}) \right],$$

where the expectation is over the randomization in $\mathcal{A}$. The worst case accuracy of $\mathcal{A}$ is

$$\mathrm{acc}(\mathcal{A}) := \inf_S \mathbb{E}_{\hat{f}=\mathcal{A}^S} \left[\mathrm{acc}_S(\hat{f}) \right],$$

the worse case expected accuracy over all data sets $S \in (\mathcal{X} \times \mathcal{Y})^n$. Our goal is to characterize

$$\mathrm{acc}(k, n, m) := \sup_{\mathcal{A}} \mathrm{acc}(\mathcal{A}) = \sup_{\mathcal{A}} \inf_S \mathbb{E}_{\hat{f}=\mathcal{A}^S} \left[\mathrm{acc}(\hat{f}) \right],$$

the accuracy that can be achieved by an algorithm after making k queries on any S. In this framework, the baseline accuracy is $1/m$, since without making any queries (when $k = 0$), the best accuracy possible is $1/m$, achieved by making a uniformly random prediction for each $x \in \mathcal{X}$. The

overfitting bias of an algorithm is $\mathrm{acc}_S(\mathcal{A}) - 1/m$, and we are interested in $\mathrm{acc}(k, n, m) - 1/m$, the maximum overfitting bias possible in the worst case.

A simplification. For a test set S, let $S_{\mathcal{X}}$ be the set of features $\{x_1, \ldots, x_n\}$ of the test set S. We first start with the variant of the problem, where the adversary has access to the test features $S_{\mathcal{X}}$. We will remove this assumption and solve the problem in its generality in Section 7. The assumption allows us to restate the overfitting problem as a sequence reconstruction problem in Section 2.1, which can be of independent interest. In this case, in order to overfit on S, the adversary can provide its predictions on the $S_{\mathcal{X}}$ as $f(x_1), \ldots, f(x_n)$ instead of specifying the entire function from $\mathcal{X} \to [m]$. Hence, instead of specifying classifiers $f : \mathcal{X} \to [m]$ as queries, we specify it as length-n sequences $\bar{q} = (q_1, q_2, \ldots, q_n) \in [m]^n$, where $q_i = f(x_i)$. $\bar{q}$ corresponds to our *guesses* for the true labels $\bar{z} = z_1, z_2, \ldots, z_n$ of examples in S. The accuracy query oracle then returns $\frac{1}{n}\sum_{j=1}^{n} \mathbb{I}_{q_j = z_j}$, the fraction of labels correctly predicted by the query on the test set S. In Section 5, and 6 we provide optimal overfitting algorithms in this model. Finally, in Section 7, we remove this assumption and extend algorithms to the scenario when test features are unknown to the adversary.

2.1. Sequence reconstruction from Hamming distance queries

Let $\bar{z} = z_1, \ldots, z_n \in [m]^n$ be an unknown sequence that corresponds to the labels of examples in S. For a query $\bar{q} = q_1, \ldots, q_n \in [m]^n$, an accuracy oracle returns

$$h(\bar{q}, \bar{z}) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{q_i = z_i},$$

the fraction of correctly predicted locations. The Hamming distance $d_{\mathrm{ham}}(\bar{q}, \bar{z})$ between $\bar{q}$ and $\bar{z}$ is related to $h(\bar{q}, \bar{z})$ as

$$d_{\mathrm{ham}}(\bar{q}, \bar{z}) = n(1 - h(\bar{q}, \bar{z})).$$

Therefore, the query that returns the fraction of matches is equivalent to a query that returns the Hamming distance between the query and the underlying sequence. The objective is to *adaptively* ask k queries and then output an estimate $\hat{\bar{z}} = \hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n$ for $\bar{z}$. The performance of the algorithm is measured by

$$h(\mathcal{A}, \bar{z}) = \mathbb{E}\left[h(\hat{z}_i, z_i)\right],$$

where the expectation is over the algorithm's randomization. The question of perfectly reconstructing $\bar{z}$ is well and long studied (Erdös and Rényi, 1963; Chvátal, 1983), and our work resolves this problem when only partial reconstruction is possible due to limited number of queries. Similar to worst case accuracy in the previous section, the algorithms are evaluated on their worst performance

$$h(\mathcal{A}) := \min_{\bar{z} \in [m]^n} h(\mathcal{A}, \bar{z}),$$

and the goal is to find an algorithm that maximizes this worst case performance,

$$h(k, n, m) := \max_{\mathcal{A}} h(\mathcal{A}).$$

Owing to the discussions above, we remark that $h(k, n, m) = \mathrm{acc}(k, n, m)$. Now, under the assumption that the test set features $S_{\mathcal{X}}$ are known to the adversary, it can provide as each query its predictions over the examples in $S_{\mathcal{X}}$, and arbitrary predictions for $x \notin S_{\mathcal{X}}$. Since the accuracy

responses depend only on $S_{\mathcal{X}}$, and the goal is to overfit for S, the question of overfitting reduces to the question of predicting a sequence under Hamming queries. Until Section 7 we consider the overfitting problem as a sequence reconstruction problem, and then in Section 7 generalize to the case when the test features are unknown.

3. Reduction to average case

Instead of worst-case $\bar{z}$, a natural question is to ask what happens if $\bar{z} \sim p$, where p is a distribution over $[m]^n$. For an algorithm $\mathcal{A}$, let

$$h(\mathcal{A}, p) := \mathbb{E}_{\bar{z}\sim p}\left[h(\mathcal{A}, \bar{z})\right].$$

For p and any $\mathcal{A}$,

$$h(\mathcal{A}, p) \geq h(\mathcal{A}). \tag{1}$$

A key observation in our work is to prove that we can assume the labels to be generated from u_m^n, the uniform distribution over $[m]^n$. In Theorem 2 we show that for any algorithm $\mathcal{A}$, there exists an algorithm $\mathcal{A}'$ such that

$$h(\mathcal{A}') = h(\mathcal{A}, \mathrm{u}_m^n).$$

In fact, we will provide an efficient construction to obtain $\mathcal{A}'$ from $\mathcal{A}$. Hence, in the rest of the paper, we design efficient algorithms whose performance on u_m^n matches the upper bound, and thereby proving their optimality. Theorem 2 can also be used to show a stronger result equating the worst case and average case performance.

Corollary 1 *For any k, n, m,*

$$h(k, n, m) = \max_{\mathcal{A}} h(\mathcal{A}) = \max_{\mathcal{A}} h(\mathcal{A}, u_m^n).$$

Proof Let $\mathcal{A}^* = \arg\max_{\mathcal{A}} h(\mathcal{A})$. By (1),

$$\max_{\mathcal{A}} h(\mathcal{A}) = h(\mathcal{A}^*) \leq h(\mathcal{A}^*, \mathrm{u}_m^n) \leq \max_{\mathcal{A}} h(\mathcal{A}, \mathrm{u}_m^n).$$

Let $\mathcal{A}^*_{\mathrm{u}_m^n} = \arg\max_{\mathcal{A}} h(\mathcal{A}, \mathrm{u}_m^n)$ be the optimal algorithm under uniform distribution. By Theorem 2 below, there exists a $\mathcal{A}^{*'}_{\mathrm{u}_m^n}$ such that

$$\max_{\mathcal{A}} h(\mathcal{A}) \geq h(\mathcal{A}^{*'}_{\mathrm{u}_m^n}) = h(\mathcal{A}^*_{\mathrm{u}_m^n}, \mathrm{u}_m^n) = \max_{\mathcal{A}} h(\mathcal{A}, \mathrm{u}_m^n).$$

The corollary follows by combining the above two equations. ■

We now formally show the construction of $\mathcal{A}'$ from $\mathcal{A}$.

Theorem 2 *For any randomized and adaptive algorithm $\mathcal{A}$, there exists an algorithm $\mathcal{A}'$ such that*

$$h(\mathcal{A}') = h(\mathcal{A}, u_m^n).$$

Proof Any algorithm $\mathcal{A}$ proceeds as follows. It chooses the first query $\bar{q}^1 \in [m]^n$ according to some distribution. Then for each $i = 2, \ldots, k$, based on the previous queries $\{\bar{q}^1, \ldots, \bar{q}^{i-1}\}$, and accuracy responses $\{h(\bar{q}^1, \bar{z}), \ldots, h(\bar{q}^{i-1}, \bar{z})\}$, it chooses the next (possibly randomized) query $\bar{q}^i$. The final guess $\hat{\bar{z}}$ is determined from all the k queries and their accuracy responses.

We construct $\mathcal{A}'$ from $\mathcal{A}$ as follows. Let $\bar{\pi} = \pi_1, \ldots, \pi_n$ be n permutations, each chosen independently and uniformly at random from S_m, the set of all permutations on $[m]$. For a sequence $\bar{z}$, let $\bar{\pi}(\bar{z}) = \pi_1(z_1), \pi_2(z_2), \ldots, \pi_n(z_n)$. Then for any $\bar{z}$, $\bar{\pi}(\bar{z})$ is distributed according to u_m^n.

Now let $\mathcal{A}^{\bar{\pi}}$ be the following algorithm. If the first query of $\mathcal{A}$ is $\bar{q}^1$, the first query of $\mathcal{A}^{\bar{\pi}}$ is $\bar{\pi}(\bar{q}^1)$. Then for $i = 2, \ldots, k$, based on the previous queries $\{\bar{q}^j, \forall j < i\}$ and outputs $\{h(\bar{\pi}(\bar{q}^j), \bar{z}) \forall j < i\}$, if $\mathcal{A}$ queries $\bar{q}^i$, then $\mathcal{A}^{\bar{\pi}}$ queries $\bar{\pi}(\bar{q}^i)$. Finally, if $\mathcal{A}$ outputs $\hat{\bar{z}}$, then $\mathcal{A}^{\pi}$ outputs $\bar{\pi}(\hat{\bar{z}})$. Now for any query $\bar{q}$,

$$h(\bar{\pi}(\bar{q}), \bar{z}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{\pi_i(q_i)=z_i} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{q_i=\pi_i^{-1}(z_i)} = h(\bar{q}, \bar{\pi}^{-1}(\bar{z})).$$

Similarly, it can be shown that for the final output

$$h(\mathcal{A}^{\bar{\pi}}, \bar{z}) = h(\bar{\pi}(\hat{\bar{z}}), \bar{z}) = h(\hat{\bar{z}}, \bar{\pi}^{-1}(\bar{z})) = h(\mathcal{A}, \bar{\pi}^{-1}(\bar{z})). \tag{2}$$

Therefore $\mathcal{A}^{\bar{\pi}}$ achieves the same expected accuracy on $\bar{z}$ that $\mathcal{A}$ achieves on $\bar{\pi}^{-1}(\bar{z})$. Alternatively, $\mathcal{A}^{\bar{\pi}}$ can be viewed as follows. If the first query of $\mathcal{A}$ is $\bar{q}^1$, $\mathcal{A}^{\bar{\pi}}$ queries $\bar{q}^1$ on $\bar{\pi}^{-1}(\bar{z})$. Then for each $i = 2, \ldots, k$, based on the previous queries $\{\bar{q}^j, \forall j < i\}$ and outputs $\{h(\bar{q}, \bar{\pi}^{-1}(\bar{z})), \forall j < i\}$, if $\mathcal{A}$ queries $\bar{q}^i$, then $\mathcal{A}^{\bar{\pi}}$, queries $\bar{q}^i$ on $\bar{\pi}^{-1}(\bar{z})$. Finally if $\mathcal{A}$ returns output $\hat{\bar{z}}$, then $\mathcal{A}^{\bar{\pi}}$ outputs $\bar{\pi}^{-1}(\hat{\bar{z}})$ as an estimate of $\bar{\pi}^{-1}(\bar{z})$. Thus by (2),

$$\mathbb{E}_{\mathcal{A},\bar{\pi}}\left[h(\mathcal{A}^{\bar{\pi}}, \bar{z})\right] = \mathbb{E}_{\mathcal{A}}\left[\mathbb{E}_{\bar{\pi}}\left[h(\mathcal{A}^{\bar{\pi}}, \bar{z})\right]\right] = \mathbb{E}_{\mathcal{A}}\left[\mathbb{E}_{\bar{\pi}}\left[h(\mathcal{A}, \bar{\pi}^{-1}(\bar{z}))\right]\right] = \mathbb{E}_{\mathcal{A}}\left[h(\mathcal{A}, \mathrm{u}_m^n)\right],$$

where the last equality uses the fact that $\bar{\pi}^{-1}(\bar{z})$ is distributed according to u_m^n. Hence,

$$h(\mathcal{A}^{\bar{\pi}}) = \min_{\bar{z}} \mathbb{E}_{\mathcal{A},\bar{\pi}}\left[h(\mathcal{A}^{\bar{\pi}}, \bar{z})\right]) = \mathbb{E}_{\mathcal{A}}\left[h(\mathcal{A}, \mathrm{u}_m^n)\right].$$

Therefore, choosing $\mathcal{A}'$ to be $\mathcal{A}^{\bar{\pi}}$, where $\bar{\pi}$ are randomly chosen permutations proves the theorem. ■

4. Overview of the algorithms

By the previous section, it suffices to design algorithms assuming that the labels $\bar{z}$ are drawn from u_m^n, namely each label is uniformly and independently distributed on $[m]$.

We first consider the case $k = O(n/m)$. Feldman et al. (2019a) proposed random queries, where each q_j^i is independently and uniformly drawn from $[m]$. Our queries on the other hand are highly correlated across the examples. We divide the examples into essentially k groups, and all the examples within a group are predicted with the same label. We will now summarize our algorithm for $k = 1$, and a sketch that its overfitting bias is the optimal $\Omega(\sqrt{1/mn})$, improving from $\Omega(\sqrt{1/(m^2 n)})$. The extension to larger k is based on similar principles. Our single query for $k = 1$ consists of predicting all the labels to be '1'. If the accuracy on this query is at least $1/m$, we predict all labels as '1' as our final prediction, otherwise we predict all labels to be '2'.

The number of examples with a particular label is $\text{Bin}(n; 1/m)$, and for two different labels, the number of examples with two different labels are negatively associated. Using arguments about their variance, and other elementary tools, we show that this algorithm obtains a standard deviation advantage over random predictions. Here the standard deviation of the number of examples with a particular label is $\sqrt{n/m}$, which we use to prove our result. The extension to larger k is similar in spirit, where we divide the examples into $k-1$ groups, and perform a similar operation over each group. The pseudo code of the algorithm is given in Figure 1, and a complete analysis in Section 5.

When $k = \Omega(n/m)$, Feldman et al. (2019a) proposed an algorithm with optimal overfitting bias, which is however not computationally efficient. They choose a number $t = \tilde{\Theta}(k)$ such that it is possible to recover the labels of the first t examples perfectly from the queries. They achieve this by performing uniform queries over the first t examples, and constant queries over the remaining (see Figure 2). Their guarantees are based on results from a similar problem studied in Erdös and Rényi (1963); Chvátal (1983), which perform a brute force search over all possible labelings of the t examples, and thus are not computationally efficient. We will make a small modification to their queries for simplicity of analysis. We will also predict the last $n-t$ examples with all one's. However, for each of the first t examples, we ensure that among the k queries there are exactly k/m of each label. Instead of reconstructing all the t examples simultaneously, we predict one example's label at a time, with a success probability of at least $3/4$. We also remark that a slight modification of our algorithm can be used with the queries as proposed by Feldman et al. (2019a) to give an efficient optimal algorithm.

5. Small k

We show that the algorithm in Figure 1 achieves an overfitting bias of $\Omega(\sqrt{k/mn})$ by proving the following theorem.

Theorem 3 *Let $n \geq m$. For $1 \leq k \leq 1 + n/2m$, $\mathcal{A}^{\textit{small}}$ in Figure 1 satisfies*

$$h(\mathcal{A}^{\textit{small}}, u_m^n) \geq \frac{1}{m} + \frac{1}{8}\sqrt{\frac{k}{mn}}.$$

We prove this theorem for $k = 1$, and $k > 1$ separately in the next two sections.

5.1. $k = 1$

The query and final prediction. For $\ell = 1, \ldots, m$, let N_ℓ be the number of examples with label ℓ. Since, the labels are uniformly distributed, $(N_1, \ldots, N_m)$ is distributed Multinomial $(n; \frac{1}{m}, \ldots, \frac{1}{m})$. Our query is to predict all the labels as '1', namely $q_j^1 = 1$ for $1 \leq j \leq n$. The accuracy observed is then N_1/n. If $N_1 \geq n/m$, then we predict all labels as '1', namely $\hat{z}_j = 1$ for all j, otherwise we output all the labels as '2'. The pseudocode is provided in Figure 1.

The number of correctly predicted labels is then given by $N_1 \cdot \mathbb{I}_{N_1 \geq n/m} + N_2 \cdot \mathbb{I}_{N_1 < n/m}$, and the expected accuracy is

$$h(\mathcal{A}^{\text{small}}, \mathrm{u}_m^n) = \frac{1}{n} \cdot \mathbb{E}\left[N_1 \cdot \mathbb{I}_{N_1 \geq n/m} + N_2 \cdot \mathbb{I}_{N_1 < n/m}\right].$$

Hence, Theorem 3 for $k = 1$, follows from the following lemma.

Algorithm $\mathcal{A}^{\text{small}}$

If $k = 1$**:**

1. Let $\bar{q}^1$ be the all one query i.e., $q_j^1 = 1, \forall j$.
2. Predict
$$\hat{z}_j = \begin{cases} 1 \text{ for all } j, & \text{if } h(\bar{q}^1, \bar{z}) \geq 1/m, \\ 2 \text{ for all } j, & \text{otherwise.} \end{cases}$$

If $k > 1$**:**

1. Divide $[n]$ into $k-1$ blocks $B_1, \ldots, B_{k-1}$ such that $n/(k-1) \geq |B_i| \geq n/2(k-1)$.
2. For $1 \leq i \leq k$:
$$q_j^i = \begin{cases} 1, & \text{if } j \in B_1 \cup B_2 \cup \ldots B_{i-1}, \\ 2, & \text{otherwise.} \end{cases}$$
3. Predict
$$\hat{z}_j = \begin{cases} 1 \text{ for all } j \in B_i, & \text{if } h(\bar{q}^{i+1}, \bar{z}) \geq h(\bar{q}^i, \bar{z}), \\ 2 \text{ for all } j \in B_i, & \text{otherwise.} \end{cases}$$

Figure 1: Algorithm for small values of k.

Lemma 4 (Appendix A.1) *Let $n \geq m \geq 2$. If $(N_1, \ldots, N_m)$ is distributed Multinomial $(n; \frac{1}{m}, \ldots, \frac{1}{m})$,*

$$\mathbb{E}\left[N_1 \cdot \mathbb{I}_{N_1 \geq n/m} + N_2 \cdot \mathbb{I}_{N_1 < n/m}\right] \geq \frac{n}{m} + \frac{1}{4}\sqrt{\frac{n}{m}}.$$

5.2. $1 < k \leq 1 + n/2m$

The queries and the final prediction. We divide the n examples into $k-1$ (consecutive) blocks $B_1, \ldots, B_{k-1}$ of almost equal sizes. For $i = 1, \ldots, k$, the ith query predicts '1' for all the examples in $B_1 \cup \ldots \cup B_{i-1}$ and it predicts '2' for the remaining examples, namely $q_j^i = 1$ if $j \in B_1 \cup \ldots \cup B_{i-1}$, and $q_j^i = 2$ otherwise. Therefore, accuracy of the $(i+1)$th query is larger than the ith query if and only if in B_i, there are more examples with label '1' than those with '2'. Our final prediction is to predict all examples in B_i as '1' if there are more '1's, otherwise we predict all examples in B_i as '2'. The pseudocode is given in Figure 1.

Proof [Proof of Theorem 3 for $k > 1$.] Let $N_{i,\ell}$ be the number of examples in B_i with label 'ℓ'. Then $(N_{1,\ell}, \ldots, N_{m,\ell})$ is Multinomial $(|B_i|; \frac{1}{m}, \ldots, \frac{1}{m})$. Our final predictions correctly predicts $\max\{N_{i,1}, N_{i,2}\}$ examples in B_i. We use the following lemma to bound the expected overfitting bias.

Lemma 5 (Appendix A.2) *Let $n' \geq m \geq 2$. If $(N_1, \ldots, N_m)$ is distributed Multinomial $(n'; \frac{1}{m}, \ldots, \frac{1}{m})$,*

$$\mathbb{E}\left[\max\{N_1, N_2\}\right] \geq \frac{n'}{m} + \frac{1}{4}\sqrt{\frac{n'}{m}}.$$

Summing over the blocks, the expected total number of correct predictions made by our algorithm is

$$\sum_{i=1}^{k-1} \mathbb{E}\left[\max\{N_{i,1}, N_{i,2}\}\right] \geq \frac{n}{m} + \frac{(k-1)}{4}\sqrt{\frac{n}{2(k-1)m}} \tag{3}$$

$$\geq \frac{n}{m} + \frac{1}{8}\sqrt{\frac{nk}{m}}, \tag{4}$$

where (3) follows from Lemma 5 and the fact that $|B_i| \geq n/(2(k-1)) \geq m$. (4) follows from the fact that $k \geq 2$. Normalizing by n proves the theorem. ■

6. Large k, $k = \Omega(n/m)$

In this section, we propose an efficient algorithm with the optimal overfitting bias for the large k case. In particular, we prove the following theorem.

Theorem 6 *Let $k > 9m \log m$. Algorithm $\mathcal{A}^{large}$ in Figure 2 satisfies,*

$$h(\mathcal{A}^{large}, u_m^n) \geq \frac{1}{m} + \frac{k}{36n \log m}.$$

Our queries. Our queries are a small modification to that of Feldman et al. (2019a) that is slightly easier to analyze. Suppose Q denote the $k \times n$ matrix whose (i,j)th entry is q_j^i. Let $t := 1 + \frac{k}{9 \log m}$. We choose the last $n - t$ columns of Q to be 1. The first t columns of Q are chosen independently from the following distribution: Each column is picked uniformly from all the $\binom{k}{\frac{k}{m}, \frac{k}{m}, \ldots, \frac{k}{m}}$ ways such that there are exactly k/m occurrences of each label $\ell \in [m]$, namely, for $1 \leq j \leq t$, and any $\ell \in [m]$, $|\{i : q_j^i = \ell\}| = k/m$.[1] We remark that this modification is only for simplifying the proof of optimality, and in fact we can tweak our final prediction slightly to provide an algorithm that has the optimal overfitting bias and using their queries.

The final prediction. Upon making the queries described above, we make the final prediction on one example at a time. We predict the last $n - t$ queries as '1', namely $\hat{z}_j = 1$ for $j > t$. We then show we can predict each of the first t labels correctly with probability at least $3/4$. Consider the jth example, for $1 \leq j \leq t$. For a label $\ell \in [m]$, consider the k/m queries such that $q_j^i = \ell$, and consider the average of the accuracies returned for these queries. Our prediction for the jth example is the label for which this average accuracy is the largest, namely

$$\hat{z}_j = \arg\max_{\ell \in [m]} \sum_{i : q_j^i = \ell} h(\bar{q}^i, \bar{z}).$$

The queries and predictions are described in Figure 2. We prove that our algorithm has the optimal bias up to logarithmic factors.

Proof [Proof of Theorem 6] By symmetry of our queries and the reconstruction, note that the probability that $\hat{z}_j = z_j$ is the same for all $j = 1, \ldots, t$. We will only consider $j = 1$, and prove that $\Pr(\hat{z}_j = z_j) > 3/4$. Let ℓ^* be the true label of the first example. Let W denote the number of 1's

1. We assume that k is an integer multiple of m for simplicity. Same results hold without the assumption.

Algorithm $\mathcal{A}^{\text{large}}$

Queries

1. Let $t := 1 + \frac{k}{9 \log m}$.
2. For $1 \le i \le k$ and $j > t$, $q_j^i = 1$.
3. For $1 \le j \le t$, independent across j, let $q_j^1, \ldots, q_j^k$ be uniformly chosen from all sequences in $[m]^k$ that have each element in $[m]$ appearing exactly k/m times.

Predict

1. For $1 \le j \le t$, $\hat{z}_j = \arg\max_{\ell \in [m]} \sum_{i: q_j^i = \ell} h(\bar{q}^i)$, breaking ties randomly.
2. For $j > t$, let $\hat{z}_j = 1$.

Figure 2: Algorithm for large values of k.

in the last $n - t$ examples. For $\ell \in [m]$, let A_ℓ be the total number of correctly predicted examples by all the queries that predict the first examples as '1', *i.e.,*

$$A_\ell := n \cdot \sum_{i: q_1^i = \ell} h(\bar{q}^i, \bar{z}) = \sum_{i: q_1^i = \ell} \sum_{j=1}^{n} \mathbb{I}_{q_j^i = \ell} = \frac{k}{m} \cdot \mathbb{I}_{\ell = \ell^*} + \frac{k}{m} \cdot W + \sum_{i: q_1^i = \ell} \sum_{j=2}^{t} \mathbb{I}_{q_j^i = \ell}.$$

Let

$$M_\ell := \sum_{i: q_1^i = \ell} \sum_{j=2}^{t} \mathbb{I}_{q_j^i = \ell},$$

then for $\ell \neq \ell^*$

$$A_{\ell^*} - A_\ell = \frac{k}{m} + M_{\ell^*} - M_\ell.$$

Now q_j^i and $q_{j'}^i$ are independent for $j \neq j'$, namely the queries are independent across examples. Further, from basic balls and bins results for a fixed j, $\mathbb{I}_{q_j^i = \ell}$ are negatively associated across i. Therefore, M_ℓ for any ℓ will satisfy the Chernoff bounds: For $\varepsilon < 1$

$$\Pr\left(|M_\ell - \mathbb{E}[M_\ell]| > \varepsilon \mathbb{E}[M_\ell]\right) \le 2 \exp\left(-\frac{\varepsilon^2}{3} \mathbb{E}[M_\ell]\right). \tag{5}$$

Now for each ℓ by the linearity of expectations,

$$\mathbb{E}[M_\ell] = (t-1) \cdot \frac{k}{m} \cdot \frac{1}{m}.$$

Suppose ε is such that $\varepsilon \mathbb{E}[M_\ell] \le \frac{k}{2m}$, and $\frac{\varepsilon^2}{3} \mathbb{E}[M_\ell] \ge 3 \log m$, then by (5) and the union bound

$$\Pr\left(\arg\max_{\ell \in [m]} A_\ell \neq \ell^*\right) < (m-1) \cdot \frac{2}{m^3} \le \frac{1}{4},$$

and with probability at least $3/4$, $\hat{\bar{z}}_1 = z_1$. Now, $\varepsilon \mathbb{E}[M_\ell] < \frac{k}{2m}$ holds for

$$\varepsilon \le \frac{m^2}{(t-1)k} \frac{k}{2m} = \frac{m}{(t-1)},$$

and $\frac{\varepsilon^2}{3}\mathbb{E}[M_\ell] \geq 3\log m$ holds for $\varepsilon \geq \sqrt{9\log m \cdot \frac{m^2}{(t-1)k}}$. Therefore, we can find a suitable ε whenever

$$\sqrt{9\log m \cdot \frac{m^2}{(t-1)k}} \leq \frac{m}{t-1} < 1.$$

If we choose $t = 1 + \frac{k}{9\log m}$ and $k > 9m\log m$, then the condition above holds. Therefore, the expected number of correctly predicted labels is at least

$$\frac{3}{4}\cdot t + \frac{1}{m}(n-t) = \frac{n}{m} + t\cdot\left(\frac{3}{4} - \frac{1}{m}\right) \geq \frac{n}{m} + \frac{k}{36\log m},$$

proving the result. ■

7. Overfitting without test features

As stated in Section 2, the results so far assume that the adversary has knowledge of the test features. We note that the above results also hold when the test features are unknown, but the test set is indexed and is always evaluated in a particular order. In this case, the adversary can create a classifier $f : \mathcal{X} \times [n] \to \mathcal{Y}$, that only looks at the index of the test sample and uses it to query. In particular, $f(x, i) = q_i$.

However, in the more general setting, we may not have access to the features of the test set, and there may not be a fixed ordering of the test examples. In this case, instead of query being a length-n sequence, the adversary in the ith query needs to provide a classifier $f^i : \mathcal{X} \to \mathcal{Y}$. We will now generalize the algorithms in the previous sections into algorithms whose each query is a classifier over the entire feature space. The guarantees for our new algorithms will be the same as those of Theorem 3 and Theorem 6 up to constant factors. These extensions work under a natural assumption that that all the n test features in $S_\mathcal{X}$ are distinct.

Recall that f is true underlying mapping from $\mathcal{X}$ to $\mathcal{Y}$. Let $\mathcal{F}$ be the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. For a test set S, $S_\mathcal{X}$ is the set of features $x_1, \ldots, x_n$, i.e., the examples with their labels dropped. With these definitions, let $\mathrm{acc}(\mathcal{A}, S_\mathcal{X}, f) := \mathrm{acc}(\mathcal{A}, S)$. For an algorithm $\mathcal{A}$ and a distribution p over $\mathcal{F}$, let

$$\mathrm{acc}(\mathcal{A}, S_\mathcal{X}, p) = \mathbb{E}_{f\sim p}[\mathrm{acc}(\mathcal{A}, S_\mathcal{X}, f)].$$

Similar to Theorem 2, we first show that uniformly random f are the hardest to overfit. Let u_m be a distribution over $\mathcal{F}$ such that when $f \sim \mathrm{u}_m$, then for each $x \in \mathcal{X}$, $f(x)$ is independently and uniformly distributed over $[m]$. Hence, as before, it suffices to consider random functions generated by u_m.

Theorem 7 (Appendix B.1) *For any randomized adaptive algorithm $\mathcal{A}$, there exists algorithm $\mathcal{A}'$ such that*

$$\mathrm{acc}(\mathcal{A}', S) = \mathrm{acc}(\mathcal{A}, S_\mathcal{X}, u_m).$$

As before, Theorem 7 can also be used to show a stronger result equating the worst case and average case performance. The proof is similar to Corollary 1 and we omit it.

Corollary 8 *For any k, n, m,*

$$\max_\mathcal{A} \mathrm{acc}(\mathcal{A}) = \max_\mathcal{A} \mathrm{acc}(\mathcal{A}, u_m).$$

7.1. Algorithms without test features for small k

We will now provide the modifications to the previously proposed algorithms $\mathcal{A}^{\text{small}}$ and $\mathcal{A}^{\text{large}}$, which are optimal even without knowledge of the features. For $k = 1$, recall that $\mathcal{A}^{\text{small}}$ queried using the all one query. Even when the test features are unknown, we query a function f^1 such that $f^1(x) = 1, \forall x \in \mathcal{X}$. For $k > 1$, recall that in $\mathcal{A}^{\text{small}}$, we divided the examples into $k-1$ blocks with almost equal sizes. In particular, our guarantee for $\mathcal{A}^{\text{small}}$ holds when the number of examples in each block is at least $2n/(k-1)$. This is possible to do when we have access to the test set features. Without knowing the features of the test set, we propose the following. Let $g : \mathcal{X} \to [k-1]$ be a random mapping from $\mathcal{X}$ to $[k-1]$, such that for each $x \in \mathcal{X}$ $g(x)$ is independently and uniformly distributed over $[k-1]$. For $j = 1, \ldots, k-1$, let

$$B_j := \{x \in \mathcal{X} : g(x) = j\}. \tag{6}$$

For $k > 1$, the only modification is in step (2) of $\mathcal{A}^{\text{small}}$. We predict '1' for all symbols $x \in B_1 \cup B_2 \ldots B_{i-1}$, and '2' otherwise. The algorithm and the analysis is in Appendix B.2.

7.2. Algorithms without test features for large k

Recall that in $\mathcal{A}^{\text{large}}$ in Figure 2, we made queries that ensured that each of the first t examples were queried precisely k/m times with each query, and the remaining $n-t$ examples are always queried with all '1's. This is not possible to do precisely without access to the features since we cannot choose a set that has exactly t of the examples in S. We make small modifications to make it work when features are unknown. Let $\mathcal{X}_t \subset \mathcal{X}$ be a randomly chosen subset of $\mathcal{X}$ such that each element in $x \in \mathcal{X}$ is in $\mathcal{X}_t$ with probability t/n (this requires the knowledge of n). For each $x \notin \mathcal{X}_t$, let $f^i(x) = 1 \forall i$. For each $x \in \mathcal{X}_t$, let $f^1(x), f^2(x), \ldots, f^k(x)$ be uniformly chosen from all sequences in $[m]^k$ that have each label in $[m]$ appearing exactly k/m times. Since $\mathcal{X}_t$ is chosen at random, the expected number of examples is $\mathcal{X}_t$ is t, and by the Chernoff bound, this value concentrates around t, and therefore the guarantees of the algorithm still remains the same up to constant factors. The rest of the analysis is similar to that of Theorem 6 and we omit it. The precise algorithm is given in Appendix B.2.

8. Information theoretic upper bound

Our proposed algorithm $\mathcal{A}^{\text{small}}$ and the information theoretic bounds of Feldman et al. (2019a) differ by a factor of $O(\sqrt{\log n})$. This previously known information theoretic upper bound uses minimum description length argument. By a careful analysis that uses Corollary 1, we show that the $\sqrt{\log n}$ factor can be removed when $k = 1$. It would be interesting to see if this can be extended to other values of k. Furthermore, for $k = 1$ as the proof of Theorem 9 shows $\mathcal{A}^{\text{small}}$ is optimal including up to the constants[2].

Theorem 9 (Appendix C) *For $k = 1$,*

$$\max_{\mathcal{A}} h(\mathcal{A}) \leq \frac{1}{m} + \frac{1}{2}\sqrt{\frac{1}{n(m-1)}}.$$

2. We note that the results in Theorem 9 and Theorem 3 differ by a constant factor due to the analysis technique.

Acknowledgments

Authors thank Vitaly Feldman, Roy Frostig, and Satyen Kale for helpful comments and suggestions. Authors thank Vitaly Feldman for suggesting methods to extend algorithms to the scenario when test features are unknown. JA is supported by NSF-CCF-1846300 (CAREER), and a Google Faculty Research Award.

References

Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1046–1059, 2016.

Daniel Berend and Aryeh Kontorovich. A sharp estimate of the binomial mean absolute deviation with applications. *Statistics & Probability Letters*, 83(4):1254–1259, 2013.

Avrim Blum and Moritz Hardt. The ladder: a reliable leaderboard for machine learning competitions. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 1006–1014. JMLR.org, 2015.

Nader H Bshouty. Optimal algorithms for the coin weighing problem with a spring scale. In *COLT*, volume 2009, page 82. Citeseer, 2009.

Vasek Chvátal. Mastermind. *Combinatorica*, 3(3-4):325–329, 1983.

Benjamin Doerr, Carola Doerr, Reto Spöhel, and Henning Thomas. Playing mastermind with many colors. *Journal of the ACM (JACM)*, 63(5):42, 2016.

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

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015b.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 117–126. ACM, 2015c.

Paul Erdös and Alfred Rényi. On two problems of information theory. *Magyar Tud. Akad. Mat. Kutató Közl*, 8, 1963.

Vitaly Feldman, Roy Frostig, and Moritz Hardt. The advantages of multiple classes for reducing overfitting from test set reuse. In *International Conference on Machine Learning*, 2019a.

Vitaly Feldman, Roy Frostig, and Moritz Hardt. Open problem: How fast can a multiclass test set be overfit? In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 3185–3189, Phoenix, USA, 25–28 Jun 2019b. PMLR.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019.

Chhavi Yadav and Léon Bottou. Cold case: The lost mnist digits. *arXiv preprint arXiv:1905.10498*, 2019.

Tijana Zrnic and Moritz Hardt. Natural analysts in adaptive data analysis. In *International Conference on Machine Learning*, pages 7703–7711, 2019.

Appendix A. Properties of the multinomial distribution

A.1. Proof of Lemma 4

Let N_2' be an independent copy of N_2. Since N_1 and N_2 are negatively correlated,

$$\begin{aligned}\mathbb{E}\left[N_1 \cdot \mathbb{I}_{N_1 \geq n/m} + N_2 \cdot \mathbb{I}_{N_1 < n/m}\right] &\geq \mathbb{E}\left[N_1 \cdot \mathbb{I}_{N_1 \geq n/m} + N_2' \cdot \mathbb{I}_{N_1 < n/m}\right] \\ &= \mathbb{E}\left[N_1 \cdot \mathbb{I}_{N_1 \geq n/m} + \frac{n}{m} \cdot \mathbb{I}_{N_1 < n/m}\right] \\ &= \mathbb{E}\left[\left(N_1 - \frac{n}{m}\right) \cdot \mathbb{I}_{N_1 \geq n/m}\right] + \frac{n}{m}.\end{aligned}$$

Let X be a random variable with $\mathbb{E}[X] = a$. Since $|X - a| = (X - a)\mathbb{I}_{X \geq a} + (a - X)\mathbb{I}_{X < a}$,

$$\mathbb{E}\left[|X - a|\right] = 2 \cdot \mathbb{E}\left[(X - a)\mathbb{I}_{X \geq a}\right].$$

Using this with $X = N_1$, and $\mathbb{E}\left[N_1\right] = n/m$ gives

$$\mathbb{E}\left[\left(N_1 - \frac{n}{m}\right) \cdot \mathbb{I}_{N_1 \geq n/m}\right] + \frac{n}{m} = \frac{1}{2}\mathbb{E}\left[\left|N_1 - \frac{1}{m}\right|\right] + \frac{n}{m}.$$

Berend and Kontorovich (2013, Theorem 1) showed that for $Y \sim \mathrm{Bin}(n; p)$ with $\frac{1}{n} \leq p \leq 1 - \frac{1}{n}$

$$\mathbb{E}\left[|Y - np|\right] \geq \sqrt{\frac{np(1-p)}{2}}. \tag{7}$$

Using this with $Y = N_1$, and $p = 1/m \geq 1/n$, we obtain

$$\frac{1}{2}\mathbb{E}\left[\left|N_1 - \frac{1}{m}\right|\right] + \frac{n}{m} \geq \frac{n}{m} + \frac{1}{2}\sqrt{\frac{n}{2m}\left(1 - \frac{1}{m}\right)} \geq \frac{n}{m} + \frac{1}{4}\sqrt{\frac{n}{m}},$$

where the final step uses $m \geq 2$. Plugging this back proves the lemma.

A.2. Proof of Lemma 5

Note that

$$\max\{N_1, N_2\} = \frac{N_1 + N_2}{2} + \frac{|N_1 - N_2|}{2}.$$

Since N_1 and N_2 are distributed $\text{Bin}(n'; 1/m)$, $\mathbb{E}[N_1 + N_2] = \frac{2n'}{m}$. Let N_2' be an independent copy of N_2. Since N_1 and N_2 are negatively correlated,

$$\mathbb{E}[|N_1 - N_2|] \geq \mathbb{E}[|N_1 - N_2'|] \geq \mathbb{E}\left[\left|N_1 - \frac{n'}{m}\right|\right] \tag{8}$$

$$\geq \sqrt{\frac{\mathbb{E}\left[\left(N_1 - \frac{n'}{m}\right)^2\right]}{2}} \tag{9}$$

$$= \sqrt{\frac{n'}{2m}\left(1 - \frac{1}{m}\right)}$$

$$\geq \sqrt{\frac{n'}{4m}}, \tag{10}$$

where (8) follows from Jensen's inequality, (9) from (7), and (10) uses $m \geq 2$.

Appendix B. Extensions to unknown test features

B.1. Proof of Theorem 7

The proof is similar to that of Theorem 2. Recall that an algorithm $\mathcal{A}$ proceeds as follow. It chooses the first query f^1 from some distribution over $\mathcal{F}$. For $i = 2, \ldots, k$, based on the $\{f^1, \ldots, f^{i-1}\}$, and accuracy responses $\{\text{acc}(f^1, \bar{z}), \ldots, \text{acc}(f^{i-1}, \bar{z})\}$, it chooses the next, possibly randomized, f^i. The final guess $\hat{f}$ is designed based on all the k queries and their accuracy responses.

We construct $\mathcal{A}'$ from $\mathcal{A}$ as follows. For each $x \in \mathcal{X}$, let π_x be an independent and uniformly sampled permutation over $[m]$. For $f \in \mathcal{F}$, let $f_\pi \in \mathcal{F}$ be $f_\pi(x) := \pi_x(f(x))$. Then, note that for any f, f_π is distributed according to u_m.

Let $\mathcal{A}^\pi$ be the following algorithm. If the first query of $\mathcal{A}$ is f^1, then the first query of $\mathcal{A}^\pi$ is $\pi(f^1)$. For $i = 2, \ldots, k$, based on the previous queries $\{f_\pi^j, \forall j < i\}$ and outputs $\{\text{acc}(f_\pi^j, \bar{z}) \forall j < i\}$, if $\mathcal{A}$ queries f^i, then $\mathcal{A}^\pi$ queries f_π^i. Finally, if $\mathcal{A}$ outputs $\hat{f}$, then $\mathcal{A}^\pi$ outputs $\pi(\hat{f})$. Now for any f^j and the true f,

$$\text{acc}(f_\pi^i, f) = \frac{1}{n}\sum_{i=1}^n \mathbb{I}_{\pi_{x_i}(f(x_i)) = f(x_i)} = \frac{1}{n}\sum_{i=1}^n \mathbb{I}_{f(x_i) = \pi_{x_i}^{-1}(f(x_i))} = \text{acc}(f, \pi^1(f)).$$

Similarly, it can be shown that for the final output

$$\text{acc}(\mathcal{A}^\pi, S_\mathcal{X}, f) = \text{acc}(\pi(\hat{f}), f) = \text{acc}(\hat{f}, f_{\pi^{-1}}) = \text{acc}(\mathcal{A}, S_\mathcal{X}, f_{\pi^{-1}}). \tag{11}$$

Therefore $\mathcal{A}^\pi$ achieves the same expected accuracy on f that $\mathcal{A}$ achieves on $f_{\pi^{-1}}$. Alternatively, $\mathcal{A}^\pi$ can be viewed as follows. If the first query of $\mathcal{A}$ is f^1, $\mathcal{A}^\pi$ queries f^1 on $f_{\pi^{-1}}$. Then for each $i = 2, \ldots, k$, based on the previous queries $\{f^j, \forall j < i\}$ and accuracy responses $\{\text{acc}(f^i, f_{\pi^{-1}}), \forall j <$

Algorithm $\mathcal{A}^{\text{small}}_{\text{unknown}}$

If $k = 1$**:**

1. Let f^1 be the all one query i.e., $f^1(x) = 1, \forall x \in \mathcal{X}$.
2. Predict
$$\hat{z}_j = \begin{cases} 1 \text{ for all } x, & \text{if } \mathrm{acc}(f^1, \bar{z}) \geq 1/m, \\ 2 \text{ for all } x, & \text{otherwise.} \end{cases}$$

If $k > 1$**:**

1. Partition $\mathcal{X}$ into $k-1$ blocks $B_1, \ldots, B_{k-1}$ using the function g defined in (6).
2. For $1 \leq i \leq k$:
$$f^i(x) = \begin{cases} 1, & \text{if } x \in B_1 \cup B_2 \cup \ldots B_{i-1}, \\ 2, & \text{otherwise.} \end{cases}$$
3. Predict
$$\hat{f} = \begin{cases} 1 \text{ for all } x \in B_i, & \text{if } h(f^{i+1}, \bar{z}) \geq h(f^i, \bar{z}), \\ 2 \text{ for all } x \in B_i, & \text{otherwise.} \end{cases}$$

Figure 3: Algorithm for small values of k without the test features.

$i\}$, if $\mathcal{A}$ queries f^i, then $\mathcal{A}^\pi$, queries f^i on $f_{\pi^{-1}}$. Finally if $\mathcal{A}$ returns output $\hat{f}$, then $\mathcal{A}^\pi$ outputs $\hat{f}_{\pi^{-1}}$ as an estimate of $f_{\pi^{-1}}$. Thus by (11),

$$\mathbb{E}_{\mathcal{A},\pi}\left[h(\mathcal{A}^\pi, f)\right] = \mathbb{E}_{\mathcal{A}}\left[\mathbb{E}_\pi\left[\mathrm{acc}(\mathcal{A}^\pi, f)\right]\right] = \mathbb{E}_{\mathcal{A}}\left[\mathbb{E}_\pi\left[\mathrm{acc}(\mathcal{A}, f_{\pi^{-1}})\right]\right] = \mathbb{E}_{\mathcal{A}}\left[h(\mathcal{A}, \mathrm{u}_m)\right],$$

where the last equality uses the fact that $f_{\pi^{-1}}$ is distributed according to u_m. Hence,

$$h(\mathcal{A}^\pi) = \min_f \mathbb{E}_{\mathcal{A},\pi}\left[\mathrm{acc}(\mathcal{A}^\pi, f)\right]) = \mathbb{E}_{\mathcal{A}}\left[\mathrm{acc}(\mathcal{A}, \mathrm{u}_m)\right].$$

Choosing $\mathcal{A}'$ to be $\mathcal{A}^\pi$, where π are randomly chosen permutations proves the theorem.

B.2. Algorithms

We provide the complete algorithm for $\mathcal{A}^{\text{small}}_{\text{unknown}}$ and $\mathcal{A}^{\text{large}}_{\text{unknown}}$ in Figures 3 and 4 respectively. As discussed in Section 7.2, the proof for large values of k is similar to that of Theorem 6. We now outline the sketch the proof for small values of k.

The analysis of $\mathcal{A}^{\text{small}}_{\text{unknown}}$ for $k > 1$ is similar to that of Theorem 3, except we need to incorporate the condition that each B_i is now not guaranteed to have size $n/(2(k-1))$. We modify the proof of Theorem 3 as follows.

Recall that Let $N_{i,\ell}$ be the number of examples in B_i with label 'ℓ'. Then $(N_{1,\ell}, \ldots, N_{m,\ell})$ is Multinomial $(|B_i|; \frac{1}{m}, \ldots, \frac{1}{m})$. Our final predictions correctly predicts $\max\{N_{i,1}, N_{i,2}\}$ examples in B_i. Hence, summing over the blocks the total expected number of correct predictions by our

Algorithm $\mathcal{A}^{\text{large}}_{\text{unknown}}$

Queries

1. Let $t := 1 + \frac{k}{9 \log m}$, and $\mathcal{X}_t \subset \mathcal{X}$ is defined in Section 7.2.
2. For $1 \le i \le k$ and $x \notin \mathcal{X}_t$, $f^i(x) = 1$.
3. Independently for each $x \in \mathcal{X}_t$, let $f^1(x), \ldots, f^k(x)$ be uniformly chosen from all sequences in $[m]^k$ that have each label in $[m]$ appearing exactly k/m times.

Predict

1. For $x \in \mathcal{X}t$, $\hat{f}(x) = \arg\max_{\ell \in [m]} \sum_{i: f^i(x) = \ell} h(f^i, \bar{z})$, breaking ties randomly.
2. For $x \notin \mathcal{X}_t$, let $\hat{f}(x) = 1$.

Figure 4: Algorithm for large values of k without the test features.

algorithm conditioned on B_i is

$$
\begin{aligned}
&\sum_{i=1}^{k-1} \mathbb{E}\left[\max\{N_{i,1}, N_{i,2}\}\right] \\
&= \sum_{i=1}^{k-1} \mathbb{E}\left[\max\{N_{i,1}, N_{i,2}\}\right] 1_{|B_i| \ge n/(2(k-1))} + \sum_{i=1}^{k-1} \mathbb{E}\left[\max\{N_{i,1}, N_{i,2}\}\right] 1_{|B_i| < n/(2(k-1))} \\
&\ge \sum_{i=1}^{k-1} \mathbb{E}\left[\max\{N_{i,1}, N_{i,2}\}\right] 1_{|B_i| \ge n/(2(k-1))} + \sum_{i=1}^{k-1} \mathbb{E}\left[\frac{N_{i,1} + N_{i,2}}{2}\right] 1_{|B_i| < n/(2(k-1))} \\
&\ge \sum_{i=1}^{k-1} \frac{|B_i|}{m} + \frac{(k-1)}{4}\sqrt{\frac{n}{2(k-1)m}} 1_{|B_i| \ge n/(2(k-1))} \\
&= \frac{n}{m} + \sum_{i=1}^{k-1} \frac{(k-1)}{4}\sqrt{\frac{n}{2(k-1)m}} 1_{|B_i| \ge n/(2(k-1))},
\end{aligned}
$$

where the second inequality follows from Lemma 5. Recall that for a binomial distribution, the median is larger than $\lfloor np \rfloor$. The lemma follows by observing that since $|B_i| \sim \text{Bin}\left(n; \frac{1}{k-1}\right)$ and $n/((k-1)) \ge 2$, $\Pr(|B_i| \ge n/(2(k-1)) \ge 1/2)$.

Appendix C. Proof of Theorem 9

By Corollary 1,

$$\max_{\mathcal{A}} h(\mathcal{A}) = \max_{\mathcal{A}} h(\mathcal{A}, \mathrm{u}_m^n).$$

Hence it suffices to consider sequences generated by the uniform distribution. We first argue that there is a deterministic algorithm that maximizes $h(\mathcal{A}, \mathrm{u}_m^n)$. Let A be the set of all deterministic algorithms. Let $\mathcal{A}^*$ be the optimal algorithm. Recall that any randomized algorithm can be written as a distribution over deterministic algorithms. Let $\lambda_{\mathcal{A}}$ is the probability that the randomized algorithm

$\mathcal{A}^*$ assigns to a deterministic algorithm $\mathcal{A} \in \mathsf{A}$. Then,

$$h(\mathcal{A}^*, \mathrm{u}_m^n) = \sum_{\mathcal{A} \in \mathsf{A}} \lambda_{\mathcal{A}} h(\mathcal{A}, \mathrm{u}_m^n) \leq \max_{\mathcal{A} \in \mathsf{A}} h(\mathcal{A}, \mathrm{u}_m^n),$$

Hence, there exists a deterministic algorithm which performs as good as $\mathcal{A}^*$ and there exists an optimal deterministic algorithm.

Since the algorithm is deterministic, by the symmetry of u_m^n, it suffices to consider the first query $\bar{q}$ as the all one sequence i.e., $q_i = 1$ for $i = 1, \ldots, n$. After this query, let $\hat{\bar{z}}$ be the estimate of the optimal deterministic algorithm.

$$\begin{aligned} h(\mathcal{A}, \mathrm{u}_m^n) &= \mathbb{E}_{z \in \mathrm{u}_m^n}[h(\bar{z}, \hat{\bar{z}})] \\ &= \mathbb{E}_{h(\bar{q}, \bar{z})}[\mathbb{E}_{z \sim \mathrm{u}_m^n}[h(\bar{z}, \hat{\bar{z}}) | h(\bar{q}, \bar{z})]] \\ &= \sum_{i=1}^{n} \mathbb{E}_{h(\bar{q}, \bar{z})}[\mathbb{E}_{z \sim \mathrm{u}_m^n}[h(z_i, \hat{z}_i) | h(\bar{q}, \bar{z})]], \end{aligned}$$

where the first equality follows by law of conditional expectations and the second equality follows by the linearity of expectations. Without loss of generality consider $i = 1$,

$$\begin{aligned} \mathbb{E}_{z \in \mathrm{u}_m^n}[h(z_1, \hat{z}_1) | h(\bar{q}, \bar{z}) = r] &= \sum_{j \in [m]} \Pr(z_1 = i | h(\bar{q}, \bar{z}) = r) h(z_1, \hat{z}_1) \\ &\leq \max_{j \in [m]} \Pr(z_1 = i | h(\bar{q}, \bar{z}) = r) \\ &= \max_{j \in [m]} \Pr(z_1 = i | h(\bar{q}, \bar{z}) = r) \\ &= \max_{j \in [m]} \frac{\Pr(h(\bar{q}, \bar{z}) = r | z_1 = i) \Pr(Z_1 = i)}{\Pr(h(\bar{q}, \bar{z}) = r)}, \end{aligned}$$

where the last equality follows by Bayes rule. Note that $h(\bar{q}, \bar{z}) \sim \text{Bin}\,(n; 1/m)$ and conditioned on $z_1 = 1$ $h(\bar{q}, \bar{z}) \sim \text{Bin}\,(n-1; 1/m) + 1$ and conditioned on $z_1 = j \neq 1$, $h(\bar{q}, \bar{z}) \sim \text{Bin}\,(n-1; 1/m)$. Hence, the above quantity can be simplified to

$$\begin{aligned} \max_{j \in [m]} \frac{\Pr(h(\bar{q}, \bar{z}) = r | z_1 = i) \Pr(Z_1 = i)}{\Pr(h(\bar{q}, \bar{z}) = r)} &= \frac{1}{m} \max\left(\frac{\binom{n-1}{r-1} m}{\binom{n}{r}}, \frac{\binom{n-1}{r} m}{\binom{n}{r}(m-1)} \right) \\ &= \frac{1}{m} \max\left(\frac{rm}{n}, \frac{(n-r)m}{n(m-1)} \right) \\ &= \frac{1}{2m} \left(\frac{rm}{n} + \frac{(n-r)m}{n(m-1)} + \left| \frac{rm}{n} - \frac{(n-r)m}{n(m-1)} \right| \right) \\ &= \frac{1}{2m} \left(\frac{rm}{n} + \frac{(n-r)m}{n(m-1)} + \frac{m|rm-n|}{n(m-1)} \right). \end{aligned}$$

Let $r = h(\bar{q}, \bar{z})$. Then, $r \sim \text{Bin}\,(n-1; 1/m)$ Combining the above equations together,

$$\begin{aligned} h(\mathcal{A}^*, \mathsf{u}_m^n) &\leq \frac{1}{2m}\mathbb{E}_r\left[\frac{rm}{n} + \frac{(n-r)m}{n(m-1)} + \frac{m|rm-n|}{n(m-1)}\right] \\ &= \frac{1}{m} + \frac{\mathbb{E}[|rm-n|]}{2n(m-1)} \\ &\leq \frac{1}{m} + \frac{\sqrt{\mathbb{E}[(rm-n)^2]}}{2n(m-1)} \\ &= \frac{1}{m} + \frac{1}{2\sqrt{n(m-1)}}, \end{aligned}$$

where we used $\mathbb{E}\left[\left(r - \frac{n}{m}\right)^2\right] = n \cdot \frac{1}{m} \cdot (1 - \frac{1}{m})$.