Label Inference Attacks from Log-loss Scores

Abhinav Aggarwal [1] **Shiva Prasad Kasiviswanathan** [1] **Zekun Xu** [1] **Oluwaseyi Feyisetan** [1] **Nathanael Teissier** [1]

Abstract

Log-loss (also known as cross-entropy loss) metric is ubiquitously used across machine learning applications to assess the performance of classification algorithms. In this paper, we investigate the problem of inferring the labels of a dataset from single (or multiple) log-loss score(s), without any other access to the dataset. Surprisingly, we show that for any finite number of label classes, it is possible to accurately infer the labels of the dataset from the reported log-loss score of a single carefully constructed prediction vector if we allow arbitrary precision arithmetic. Additionally, we present label inference algorithms (attacks) that succeed even under addition of noise to the log-loss scores and under limited precision arithmetic. All our algorithms rely on ideas from number theory and combinatorics and require no model training. We run experimental simulations on some real datasets to demonstrate the ease of running these attacks in practice.

1. Introduction

Log-loss (a.k.a. cross-entropy loss) is an important metric of choice in evaluating machine learning classification algorithms. Log-loss is based on prediction probabilities where a lower log-loss value means better predictions. Therefore, log-loss is useful to compare models not only on their output but on their probabilistic outcome.

Let $[K] = \{1, \ldots, K\}$ be a set of label classes and consider a dataset of N datapoints with true labels $\sigma \in [K]^N$. The K-ary log-loss score takes as input σ and a matrix in $[0,1]^{N \times K}$ of prediction probabilities, where the ith row is the vector of prediction probabilities $u_{i,1}, \ldots, u_{i,K}$ (with $\sum_{k \in [K]} u_{i,k} = 1$) for the ith datapoint on the K-classes.

Definition 1 (K-ary log-loss Score)**.** *Let $\mathbf{u} \in [0,1]^{N \times K}$ be a matrix such that for all $k \in [K]$ and $i \in [N]$, it holds that $\sum_{i=1}^{N} u_{i,k} = 1$. Let $\sigma \in [K]^N$ be a labeling. Then, the K-ary log-loss (or, cross-entropy loss) on $\mathbf{u}$ with respect to σ, denoted by $\textsc{LLoss}(\mathbf{u}; \sigma)$, is defined as follows:*

$$\textsc{LLoss}(\mathbf{u}; \sigma) := \frac{-1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \Big([\sigma_i = k] \cdot \ln u_{i,k} \Big),$$

where $[\sigma_i = k] = 1$ if $\sigma_i = k$ and 0, otherwise.

Typically, the $\mathbf{u} \in [0,1]^{N \times K}$ is generated by a ML model. We note that some texts use the unscaled version of log-loss and that our algorithms can be easily extended to these variants (see (Murphy, 2012) for more details on log-loss).

A special case is when $K = 2$ (binary labels) in which case the above definition reduces to the following simpler form.

Definition 2 (Binary log-loss Score)**.** *Given a vector $\mathbf{u} = (u_1, \ldots, u_N) \in [0,1]^N$ and a labeling $\sigma \in \{0,1\}^N$, the log-loss on $\mathbf{u}$ with respect to σ, denoted by $\textsc{LLoss}(\mathbf{u}; \sigma)$, is defined as follows:*

$$\textsc{LLoss}(\mathbf{u}; \sigma) := \frac{-1}{N} \ln \left(\prod_{i=1}^{N} u_i^{\sigma_i} (1 - u_i)^{1 - \sigma_i} \right).$$

Given its preeminent role in evaluating machine learning models, especially in the neural network literature (Goodfellow et al., 2016), an important question arises is whether it is possible to "exploit" the log-loss score. In particular, we ask whether the knowledge of log-loss scores leaks information about the true labels σ. We answer the question in affirmative by showing that for any finite number of label classes, it is possible to infer all of the dataset labels from just the reported log-loss scores if the prediction probability vectors are carefully constructed and this can be done *without* any model training. In fact, we present stronger inference attacks, that succeed even when the log-loss scores are perturbed by noise.

Our inference attacks have important consequences:

(i) Integrity of Machine Learning Competitions: Many machine learning (data mining) competitions such as those organized by Kaggle[1], KDDCup[2] and ILSVRC Chal-

[1]Amazon. Correspondence to: Abhinav Aggarwal <aggabhin@amazon.com>, Shiva Prasad Kasiviswanathan <kasivisw@amazon.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]https://www.kaggle.com/

[2]https://www.kdd.org/kdd-cup

Table 1. Overview of our results for binary label inference. Here, $N \geq 1$ is the number of labels to be inferred. We present attacks under both arbitrary and bounded precision arithmetic models (see Section 2 for a comparison of these models). The τ-accurate means that the error on the responses are bounded by $|\tau|$. The fourth column represents the number of arithmetic operations needed at the adversary. All our adversaries are polynomial time except for the third row.

Amount of Noise in Responses	Precision	# Log-loss Queries	#Arithmetic Operations	Reference
No noise	Arbitrary	1	$O(N)$	Theorem 1
No noise	ϕ-bits	$\Theta\left(1 + N\phi 2^{-\phi/4}\right)$	$O(N)$	Algorithm 1
τ-accurate	Arbitrary	1	$O(2^N)$	Algorithm 2
τ-accurate	ϕ-bits	$O\left(\frac{N}{\log N} + \frac{N}{\log(\phi/N\tau)}\right)$	$O\left(\frac{poly(N,\phi/\tau)}{\log(\phi/N\tau)}\right)$	Algorithm 3

lenge [3] use log-loss as their choice of evaluation metric. In particular, it is common in these competitions that the quality of a participants' solution to be assessed through a log-loss on an unknown test dataset. Our results demonstrate that an unscrupulous participant can game this system, by using the log-loss score to learn the test set labels, and thereby constructing a fake but perfect classifier with zero test error. The simplicity and efficacy of our proposed attacks make this issue a real concern.[4]

(ii) Privacy Concerns: ML models are regularly trained on sensitive datasets. Imagine an adversary who can ask log-loss scores for supplied prediction vectors. While at the onset the log-loss being a non-linear function, might look innocuous to release, our results show the extent of information leakage from these scores. In fact, we get a perfect reconstruction, a stronger privacy violation than that achieved by the *blatant non-privacy* notion (Dinur & Nissim, 2003), which only requires a large fraction of the sensitive data to be reconstructed.

Overview of Our Results. We present multiple inference attacks from log-loss scores under various constraints such as bits of arithmetic precision, noise etc. All our attacks operate only based on the ability of an adversary to query the log-loss scores on the chosen prediction vectors, without any access to the feature set or requiring any model training. All our inference attacks are also completely agnostic of the underlying classification task.

Our primary focus in this paper is on the binary label case. An overview of our main results for the binary label inference, that we discuss below, is summarized in Table 1. We start with the simplest setting, where the log-loss scores are observed in the raw (without any noise). If the adversary has the ability to perform arbitrary precision arithmetic, we show that with just one log-loss query, an adversary can recover all the labels. We extend this result to the case where the adversary performs ϕ-bits precision arithmetic, and show that the labels can be recovered with $\Theta\left(1 + N\phi 2^{-\phi/4}\right)$ log-loss queries. Both these attacks require only a polynomial-time adversary and also extend to the multiclass case.

We then move on to the more challenging setting where the scores can be perturbed with noise before the adversary observes them. Assuming that the responses are τ-accurate (i.e., within error $\pm\tau$), we show that an adversary can still recover all the labels correctly in the arbitrary precision model with just one query but now with exponential time. Interestingly, this holds independent of τ. The construction here uses large numbers (that are doubly-exponential in N) that our lower-bounds suggest are unfortunately unavoidable. In the ϕ-bits precision model, we present a polynomial-time adversary that requires $O\left(N/\log N + N/\log\left(\phi/N\tau\right)\right)$ queries.

Next, we present extensions of these attacks to other interesting noise settings such as randomly generated noise and multiplicative noise, and show how to recover labels in those settings (see Section 4.3). Finally, in Section 5, we present experiments that demonstrate the remarkable effectiveness and speed of these attacks on real and simulated datasets.

We note that while the techniques for label inference in the noised case will also hold for the (raw) unnoised case, we discuss the later separately to capture some key ideas behind our constructions. Moreover, our construction for inference from raw scores has some advantages, it uses a fewer number of queries and can be easily extended to the multiclass setting.

Overview of Our Techniques. Our attacks are based on a variety of number-theoretic and combinatorial techniques that we briefly summarize here.

- For the case where log-loss scores are returned without any noise, we use the *Fundamental Theorem of Arithmetic* (Hardy & Wright, 1979), which states that every positive integer has a unique prime factorization. We assign powers of distinct primes to different datapoints in a way that all labels can be recovered in a single query, for both binary as well as the multiclass case. Moreover, since the list of primes is well-known and

[3] http://www.image-net.org/challenges/LSVRC/

[4] If needed, an attacker can obfuscate the prediction vectors needed for our attacks in its ML models. We do not focus on this aspect here.

only a function of the number of datapoints, recovery of labels from the observed scores is efficient assuming arbitrary-precision arithmetic.

- To adapt our construction above (in the no-noise setting) to the case of bounded floating-point precision, we bound the number of log-loss queries using the well-known *Prime Number Theorem* (Poussin, 1897; Hadamard, 1896), which provides an asymptotic growth rate for the size of prime numbers. Our construction achieves label inference in an optimal number of queries, which we prove by providing matching upper and lower bounds.
- For the case of label inference from noised scores, we use a different attack strategy. Here, we reduce this problem to the *construction of sets with distinct subset sums*. For the lower-bound here, we build upon the classic result (first conjectured) by Euler (and later proved by (Benkoski & Erdős, 1974; Frenkel, 1998)), which bounds the size of the largest element in such sets with integer elements.

2. Problem Definition and Setting

We begin by formally defining our model of computation. In this paper, we discuss *perfect* label inference problem, which refers to inferring *all* the labels. We formally define label inference under different constraints in subsequent sections. We refer to the entity that runs this inference as the *adversary*.

Throughout the paper, unless otherwise stated, we focus on the binary label case. We refer to the vector $\mathbf{u}$ in Definition 2 as the *prediction vector* used by the adversary. The key ideas in our label inference algorithms are best explained by describing a vector $\mathbf{v} = (v_1, \ldots, v_N) \in \mathbb{R}^N$ and then constructing the prediction vector $\mathbf{u} = f(\mathbf{v}) := [f(v_1), \ldots, f(v_N)]$, where $f(x) = \frac{x}{1+x}$.

For notational convenience, we define:

$$\mathcal{L}_{\mathbf{v}}(\sigma) := \text{LLOSS}(f(\mathbf{v}), \sigma) = \text{LLOSS}(\mathbf{u}, \sigma). \quad (1)$$

Note that for any $\sigma \in \{0,1\}^N$, a simple algebraic manipulation of $\mathcal{L}_{\mathbf{v}}(\sigma)$ gives the following:

$$\mathcal{L}_{\mathbf{v}}(\sigma) = \frac{-1}{N} \ln\left(\frac{\prod_{i:\sigma_i=1} v_i}{(1+v_1)\ldots(1+v_N)}\right). \quad (2)$$

We will repeatedly refer to this form in our constructions. We are interested in vectors $\mathbf{v}$ for which the function $\mathcal{L}_{\mathbf{v}}$ is injective (i.e., a 1-1 correspondence). This injection will allow the adversary to ensure that the true labeling can be unambiguously recovered from the observed loss score.

Models of Computation. We present our results in two models of arithmetic computation. The first model assumes *arbitrary precision arithmetic*, which allows precise arithmetic results even with very large numbers. We refer to this as the APA model. While this model results in considerably slower arithmetic (Brent & Zimmermann, 2010), it helps an adversary to perform label inference with fewer queries.

The second model is the more standard *floating point precision model*, where the arithmetic is constrained by limited precision. We denote this model as FPA(ϕ), where ϕ represents the number of bits of precision. We assume the following abstraction for the format for representing numbers in this model: 1 bit for sign, $(\phi-1)/2$ bits for the exponent and $(\phi-1)/2$ bits for the fractional part (mantissa).[5] This allows representing all numbers between $-2^{(\phi-1)/2}$ to $+2^{(\phi-1)/2}$, with a resolution of $2^{-(\phi-1)/2}$. The floating point precision model allows for more efficient arithmetic operations (Brent & Zimmermann, 2010). We refer the reader to Chapter 4 in (Knuth, 2014) for a detailed discussion on designing algorithms for standard arithmetic in these models.

Threat Model. We assume that the adversary sends a prediction vector $\mathbf{u}$ to a machine (server) that holds the (private) dataset $\sigma \in \{0,1\}^N$ (also called *labeling*) and gets back $\text{LLOSS}(\mathbf{u}, \sigma)$. We assume that the loss is computed on all labels in σ and that N is known to the adversary. In the setting where the scores are noised, we assume that the adversary knows an upper bound on the resulting error. Often the former can be inferred from the knowledge of the precision on the machine that returns the loss score to the adversary. Finally, we assume that the adversary can make multiple queries with different prediction vectors and obtain the corresponding loss scores. Since this query access can be limited in practical settings, we optimize the number of queries required by our inference algorithms and prove formal lower bounds.

We refer to the adversary as a *polynomial-time* adversary if it is restricted to only polynomial-time computations, otherwise we refer to it as an *exponential-time* adversary.

Related Work. Whitehill (2018) initiated the study of how log-loss scores can be exploited in ML competitions, which was optimized for single-query inference in (Aggarwal et al., 2020). However, the attack in (Whitehill, 2018) constructs prediction vectors (which they call probe matrices) by heuristically solving a min-max optimization problem in a space that is exponentially large in the number of labels their algorithm infers in a single query. This heuristic is based on a Monte-Carlo simulation, which severely limits the scalability of their algorithm to arbitrary large datasets (and/or to arbitrarily many number of classes). In contrast, our construction is simple and practical, which makes our attack efficient (see Section 5 for details). Additionally, the

[5] More generally, one could allocate ϕ_a bits for the exponent and ϕ_b bits for the fractional part where $\phi_a + \phi_b = \phi - 1$. This is the setting in our experiments.

algorithms in both (Whitehill, 2018; Aggarwal et al., 2020) cannot be extended to the noised case – the attacks by (Aggarwal et al., 2020), in particular, use a single log-loss query and hence, cannot be run in the finite precision setting. Additionally, the use of Twin Primes in (Aggarwal et al., 2020) is missing a discussion on efficiently constructing such primes for arbitrarily large datasets.

Label inference attacks based on other metrics such as AUC scores are also known (Whitehill, 2016; Matthews & Harel, 2013), but we do not know of any connection between these and our setting. In a recent work, (Blum & Hardt, 2015) demonstrate general techniques for safeguarding leaderboards in Kaggle-type competition settings against an adversarial boosting attack, in which the attacker observes loss scores on randomly generated prediction vectors to generate a labeling which, with probability $2/3$, gives a low loss function.

The constructions introduced in this paper are related to the ideas prevalent in the coding theory literature.[6] The idea of designing the prediction vector ($\mathbf{u}$) can be viewed as constructing a coding scheme, whose input is the true labels, with the goal of recovering (decoding) the true labels after passing it through the log-loss function (which acts as the noisy channel). In particular, our constructions in Section 4, have parallels to coding schemes based on Sidon sequences (O'Bryant, 2004) and Golomb rulers (Robinson & Bernstein, 1967). We believe that better label inference attacks could be designed by further exploring this connection with the coding theory literature.

Additional Notation. We will denote by $[n] = \{1, 2, \ldots, n\}$ and use $\mathbb{R}$ for real numbers, $\mathbb{Z}$ for integers, and $\mathbb{Z}^+$ for the set of positive integers. For any vector $v = [v_1, \ldots, v_n]$, we use $v[:a] = [v_1, \ldots, v_a]$ for $a \in [n]$. Unless specified, all logarithms use the natural base (e) and $p_1, p_2, \ldots$ will denote the primes (p_i being the i^{th} prime).

3. Label Inference from Raw Scores

We begin our discussion with label inference from scores that are reported without any noise added. We will first assume arbitrary precision arithmetic to explain the key idea behind our construction, and then extend the discussion to the case of floating-point precision. Missing details from this section are collected in the Appendix A.

We begin by formally defining the label inference problem.

Definition 3. *Let $\sigma \in \{0,1\}^N$ be an (unknown) labeling. The label inference problem is that of recovering σ given $\text{LLoss}(\mathbf{u}_1; \sigma), \ldots, \text{LLoss}(\mathbf{u}_M; \sigma)$. Here, M is the number of queries and $\mathbf{u}_i \in [0,1]^N$ are the prediction vectors.*

[6] This was pointed to us by the anonymous ICML reviewers.

3.1. Single Query Label Inference under Arbitrary Precision with Polynomial-time Adversary

For our first result, we show that it is possible to extract all ground truth labels using just one query in the arbitrary precision (APA) model. Our key tool is the Fundamental Theorem of Arithmetic (Hardy & Wright, 1979), which states that every integer has a unique prime factorization. Recall that from Equation (1), recovering σ from $\text{LLoss}(\mathbf{u}; \sigma)$ is equivalent to recovering σ from $\mathcal{L}_{\mathbf{v}}(\sigma)$. Our construction is described below.

Theorem 1. *There exists a polynomial-time adversary for the single-query label inference problem in the* APA *model.*

Proof. Let σ denote the (unknown) labeling. Define $\mathbf{v} = [p_1, \ldots, p_N]$ and $T = \prod_{i=1}^{N}(1+p_i)$. Observe that the terms in Equation (2) can be re-arranged to give $\prod_{i:\sigma_i=1} p_i = T \exp(-N\mathcal{L}_{\mathbf{v}}(\sigma))$. This gives the required injection since there is a unique product of primes for any given value of the right hand side of this equation. Moreover, the primes in this product uniquely define which elements in σ have label 1, since we use a distinct prime for each i. □

As an example, for $N = 5$, let $\mathbf{v} = [2, 3, 5, 7, 11]$. Suppose the true labeling is $[0, 1, 1, 0, 1]$. Then, the adversary observes $\mathcal{L}_{\mathbf{v}}(\sigma) = \frac{1}{5}\ln\left(\frac{2304}{55}\right)$ (obtained by plugging in $\mathbf{v}$ and σ in Equation 2). For reconstructing the labels, observe that $T = 3 \times 4 \times 6 \times 8 \times 12 = 6912$, so that all we need is to compute primes that divide $T\exp(-N\mathcal{L}_{\mathbf{v}}(\sigma)) = 165 = 3 \times 5 \times 11$. This tells us that only the labels for the second, third and fifth datapoints must be 1, which is indeed true.

We note that the construction above is not unique – all the steps in the proof would go through if we replaced $\mathbf{v}$ with *any* vector containing distinct primes (or even mutually co-prime numbers) by the unique factorization property. Moreover, since the adversary decides what primes go inside $\mathbf{v}$, it only takes $O(N)$ time to determine all the factors in the product above (e.g., by checking each p_i one by one). We further note that our proof assumes that $Te^{-N\mathcal{L}_{\mathbf{v}}(\sigma)}$ can be written precisely and unambiguously as an integer for any $\mathbf{v}$ and σ. This requires arbitrary precision. In practical scenarios, however, this may not be the case and hence, only a few labels may be correctly inferred. We discuss our construction for exact inference in this case next.

3.2. Label Inference under Bounded Precision with Polynomial-time Adversary

To work in the more realistic scenario of bounded precision within the $\text{FPA}(\phi)$ model, we are restricted in our choice of primes since the primes from Theorem 1 can get very large (as N increases). We handle this using multiple queries, inferring only a few labels at a time (details outlined in Algorithm 1). In each iteration, the prediction vectors use

Algorithm 1 Label Inference with No Noise in the $\mathsf{FPA}(\phi)$ Model (Polynomial Adversary)

Input: N (length of vector), ϕ (bits of precision)
Output: Labeling $\hat{\sigma} \in \{0,1\}^N$
Initialize $\hat{\sigma} \leftarrow [0, \ldots, 0]$.
Let m be the largest integer for which $p_m \leq 2^{(\phi-5)/4}$.
for k **in** $\{1, 2, \ldots, \lceil N/m \rceil\}$ **do**
  Set $\mathbf{v}^{(k)}$ with $\mathbf{v}^{(k)}_j = p_r$ if $j = (k-1)m + r$ for some $r \in [m]$. Else, set $\mathbf{v}^{(k)}_j = 1$.
  Obtain the loss $\ell^{(k)}$ using $\mathbf{u}^{(k)} = f\left(\mathbf{v}^{(k)}\right)$ as the prediction vector.
  Let $P^{(k)} = [p_1, \ldots, p_{|P^{(k)}|}]$ be the set of primes inside $\mathbf{v}^{(k)}$ and $\alpha^{(k)} = \prod_{p \in P^{(k)}}(1+p)$.
  Compute $q^{(k)} \leftarrow \alpha^{(k)} e^{-N\ell^{(k)} + (N - |P^{(k)}|)\ln 2}$.
  For $j \in \{1, \ldots, |P^{(k)}|\}$, if p_j divides $q^{(k)}$, then set $\hat{\sigma}_{(k-1)m+j} \leftarrow 1$.
end for
Return $\hat{\sigma}$.

primes for the bits not yet inferred, and the remaining entries are kept fixed. This way the remaining entries contribute a fixed amount to the loss, which can be subtracted at the time of label inference. Thus, using a smaller number of primes allows us to work with the available precision budget.

Theorem 2. *Let $\phi \geq 9$. There exists a polynomial-time adversary (from Algorithm 1) for the label inference problem in the $\mathsf{FPA}(\phi)$ model using $\Theta\left(1 + N\phi 2^{-\phi/4}\right)$ queries.*

Proof. We begin by proving the upper bound on the number of queries. We refer to Algorithm 1 for our proof.

Let σ denote the true labeling. Fix some $m \in [N]$ and without loss of generality, assume that N is a multiple of m, so that $\lceil N/m \rceil = N/m$. Then, in the k^{th} iteration of the for-loop in Algorithm 1, the prediction vector $\mathbf{u}^{(k)}$ uses m primes $P^{(k)} = [p_1, \ldots, p_m]$, with all other entries set to 1. Since the computation of log-loss is invariant of the relative order of the datapoints (as long as it aligns with the prediction vector), it suffices to show that the first iteration of the loop ($k = 1$) correctly recovers the first m labels. To see this, observe that Lemma 4 gives us that the log-loss score observed on $\mathbf{u}^{(k)}$ has the following form:

$$
\begin{aligned}
&-N\ell^{(1)} = -N\mathcal{L}_{\mathbf{v}^{(1)}}(\sigma) \\
&= \ln\left(\prod_{\substack{i:\sigma_i=1 \\ 1\leq i\leq M}} p_i\right) - (N-M)\ln 2 - \sum_{j=1}^{M}\ln(1+p_j) \\
\implies &\left(\prod_{j=1}^{M}(1+p_j)\right) e^{-N\ell^{(1)} + (N-M)\ln 2} = \prod_{\substack{i:\sigma_i=1 \\ 1\leq i\leq M}} p_i.
\end{aligned}
$$

The product term on the left is the same as $\alpha^{(1)}$ and the product term on the right allows for unambiguous label recovery (via unique factorization) in Step 10 of Algorithm 1.

Next, we prove that in the $\mathsf{FPA}(\phi)$ model, a polynomial-time adversary can infer at most $2^{\phi/4}/\phi$ labels per query.

To see this bound on the number of labels inferred per iteration, first observe that:

$$
\begin{aligned}
&\min_{\substack{i,j\in[N] \\ p_i\neq p_j}} \left|\frac{p_i}{1+p_i} - \frac{p_j}{1+p_j}\right| \geq \frac{p_m}{1+p_m} - \frac{p_{m-1}}{1+p_{m-1}} \\
&= \frac{p_m - p_{m-1}}{(1+p_m)(1+p_{m-1})} \geq \frac{p_m - p_{m-1}}{4p_m p_{m-1}} \\
&= \frac{1}{4}\left(\frac{1}{p_{m-1}} - \frac{1}{p_m}\right) \geq \frac{1}{4}\left(\frac{1}{p_m - 1} - \frac{1}{p_m}\right) \geq \frac{1}{4p_m^2},
\end{aligned}
$$

where the first line follows from the fact that $p_1 < \cdots < p_m$, and the second line from $p_m \geq 3$ and $p_{m-1} \geq 2$. Thus, in the $\mathsf{FPA}(\phi)$ model, we can only use m that is large enough so that the following continues to holds:

$$
\begin{aligned}
&\frac{\phi - 1}{2} \geq \log_2(4p_m^2) = 2 + 2\log_2 p_m \\
\implies &\phi \geq 5 + 4\log_2 p_m.
\end{aligned}
$$

From this, we obtain that the largest prime p_m that can be used must be at most $2^{(\phi-5)/4}$, as mentioned in Algorithm 1 (which establishes the optimality of our construction).

Finally, from Lemma 3, since $p_m = \Theta(m \log m)$, we obtain $m = \Theta\left(2^{\phi/4}/\phi\right)$, which gives the number of queries as $\Theta\left(N\phi 2^{-\phi/4}\right)$ for our inference attack. □

Note that the bound on the number of queries is asymptotically tight. We prove this using the Prime Number Theorem (Hadamard, 1896; Poussin, 1897), which describes the asymptotic distribution of primes among the integers. In Section 5, we present experimental results to show Algorithm 1 can be used for label inference on real datasets.

3.3. Extension to the Multiclass Case

Our construction using primes in Theorem 1 can be extended to multiple classes as well. We now use Definition 1. We prove that using the powers of a distinct prime for each datapoint, we can infer all the labels in a single query – in particular, using vector $\mathbf{v}_K$ of the following form:

$$
v_{i,k} = \frac{p_i^{k-1}}{\sum_{j=1}^{K} p_i^{j-1}} \quad \forall (i,k) \in [N] \times [K].
$$

The proof of correctness follows from the Fundamental Theorem of Arithmetic (see Appendix A.2 for details).

Theorem 3. *There exists a polynomial-time adversary for K-ary label inference in the* APA *model using only a single log-loss query. For inference in the* FPA(ϕ) *model, it suffices to issue* $O\left(1 + NKh(\phi)\right)$ *queries, where the following holds when* $K < N$*:*

$$h(\phi) = O\left(\frac{(\ln \phi)^2}{(\phi + (N-K)\ln K)^{2/3}}\right).$$

Observe that $\lim_{\phi\to\infty} h(\phi) = 0$, as expected.

4. Label Inference from Noised Scores

In this section, we describe a label inference attack for binary labels that works even when the reported scores are noised before the adversary gets to see them. We do not place any assumption on the noise distribution except that the adversary knows an upper bound on the amount of resulting error. Compared to attacks in the unnoised case presented in the previous section, our attacks here use a larger number of queries (for the same number of bits of precision) and currently only work for the binary label case.

We begin by formally defining the label inference problem in presence of noise. Missing details from this section are collected in the Appendix B.

Definition 4. *Let* $\tau > 0$ *and* $\sigma \in \{0,1\}^N$ *be the (unknown) labeling. The* τ*-robust label inference problem is that of recovering* σ *given* $\ell_1, \ldots, \ell_M$*, where for all* $i \in [M]$*, it holds that* $|\text{LLOSS}(\mathbf{u}_i; \sigma) - \ell_i| \le \tau$*. Here,* M *is the number of queries and* $\mathbf{u}_i \in [0,1]^N$ *are the prediction vectors.*

As before, we discuss the results in both APA and FPA arithmetic. In this case, we also make a distinction between exponential- and polynomial-time adversaries.

4.1. τ-Robust Label Inference under Arbitrary Precision with Exponential-time Adversary

As in Section 3, because of the equivalence between $\mathcal{L}_{\mathbf{v}}(\sigma)$ and $\text{LLOSS}(\mathbf{u}; \sigma)$ for $\mathbf{u} = f(\mathbf{v})$, we focus on recovering σ from $\mathcal{L}_{\mathbf{v}}(\sigma)$. We start with a definition of a vector $\mathbf{v}$ that helps with label inference in the presence of noise. To illustrate our key ideas, we first focus on an exponential-time adversary, and later extend the results to a polynomial-time adversary.

Definition 5. *Let* $\tau > 0$*. In the* APA *model, we say that a vector* $\mathbf{v} \in (0,\infty)^N$ *is* τ*-robust if for all labelings* $\sigma \in \{0,1\}^N$ *and all* ℓ *such that* $|\mathcal{L}_{\mathbf{v}}(\sigma) - \ell| \le \tau$*, there exists an algorithm (Turing Machine)* $\mathcal{A}$ *such that* $\mathcal{A}(\ell, N, \tau, \mathbf{v}) = \sigma$*.*

We show that for all $\tau > 0$, there exists a τ-robust vector in the APA model that can recover the true labeling in a single query. We do so by constructing a vector $\mathbf{v}$ such that for any two different labelings $\sigma_1, \sigma_2 \in \{0,1\}^N$, it holds that $|\mathcal{L}_{\mathbf{v}}(\sigma_1) - \mathcal{L}_{\mathbf{v}}(\sigma_2)| > 2\tau$, so that the inference from the noised loss is unambiguous. To achieve this co-domain separation, we reduce our problem to constructing sets with a given minimum difference between its arbitrary subset sums. The construction from this reduction will help design our vector $\mathbf{v}$. The main steps of our approach are outlined in Algorithm 2 and described below.

Algorithm 2 Label Inference with Bounded Error in the APA Model (Exponential Adversary)

Input: N, upper bound on error $\tau > 0$
Output: Labeling $\hat{\sigma} \in \{0,1\}^N$
Set $\mathbf{u} \leftarrow f(\mathbf{v})$, where $v_i \leftarrow 3^{2^i N\tau}$.
Obtain the loss score ℓ using $\mathbf{u}$ as the prediction vector.
Return $\hat{\sigma} \leftarrow \arg\min_{\sigma\in\{0,1\}^N} |\mathcal{L}_{\mathbf{u}}(\sigma) - \ell|$.

Let $\Delta(\mathbf{v}) := \min_{\sigma_1,\sigma_2\in\{0,1\}^N} |\mathcal{L}_{\mathbf{v}}(\sigma_1) - \mathcal{L}_{\mathbf{v}}(\sigma_2)|$ be the magnitude of the minimum difference in the loss scores computed on any two distinct labelings. For any set S, let $\mu(S) := \min_{S_1,S_2\subseteq S} \left|\sum_{s_1\in S_1} s_1 - \sum_{s_2\in S_2} s_2\right|$ denote the magnitude of the minimum difference between any two subset sums in S. Remember that our goal is to ensure that $\Delta(\mathbf{v})$ is large (in particular, more than 2τ). The following lemma will be helpful in constructing such a vector $\mathbf{v}$.

Lemma 1. *Let* $\mathbf{v} = [v_1, \ldots, v_N]$ *be a vector with all entries distinct and positive. Define* $\ln \mathbf{v} := [\ln v_1, \ldots, \ln v_N]$*. Then, it holds that* $\Delta(\mathbf{v}) = \frac{1}{N}\mu(\ln \mathbf{v})$*.*

Now, let $\mathcal{S}_N = \{1, 2^1, \ldots, 2^{N-1}\}$. Observe that $\mu(\mathcal{S}_N) = 1$. This follows from the fact that $\mathcal{S}_N$ contains all N distinct powers of 2 (from 0 to $N-1$), and hence, every subset of $\mathcal{S}_N$ has a distinct sum since there are 2^N subsets of $\mathcal{S}_N$ and each subset corresponds to a unique integer in $[2^N - 1]$. Using this set, we can achieve the desired separation between loss scores by scaling the elements in $\mathcal{S}_N$ as suggested by Lemma 1 and the construction in Algorithm 2. We prove the correctness of this approach in the following.

Theorem 4. *For any* $\tau > 0$*, there exists an exponential-time adversary (from Algorithm 2) for single-query* τ*-robust label inference in the* APA *model.*

We note a few keys remarks about Algorithm 2. First, note that the exact knowledge of τ is not required – any upper bound τ_{max} suffices. This eliminates the need for the adversary to know the exact noise generation process and compute the bound τ_{max} purely from its knowledge of the environment (*e.g.* precision on the channel through which the scores are communicated). Second, at first glance, it may seem too good to be true that we can handle arbitrary noise levels added to the noise scores. Intuitively, too much noise must render any signal completely useless when recovering meaningful information. However, we remind the reader that Algorithm 2 requires arbitrary precision. In practice,

any finite precision will limit the resolution to which the difference between the loss scores can be controlled using vectors that contain entries that are exponentially large in N and τ. As we will show next, multiple queries are required in this case, which, instead of separating scores over all the labelings at once, only separates scores over labelings that differ in a few bits (which can be significantly smaller in number). Third, note that the adversary iterates over the entire exponentially large (2^N) space of the labelings to recover the true labeling σ (see step 5). While this is feasible if we assume an all-powerful adversary, in more realistic scenarios where the adversary is limited to only polynomial computations, at most $O(\log N)$ bits must be inferred at a time. We discuss this approach in more detail below. Lastly, observe that Algorithm 2 is a single-query inference. Even with the caveats mentioned above, this algorithm is a certificate to the guarantee that even noisy loss scores leak sufficient information about the private labels, which, given enough computation power, can be extracted unambiguously. A natural question to ask is if it is possible to perform this inference using smaller numbers than what our construction uses. We now explore if this is possible.

Optimality of Single-Query τ-Robust Inference. We now prove that any solution to the single-query Robust Label Inference Problem must use a prediction vector that has large entries. At a high level, we do this by first reducing the problem of constructing sets with distinct subset sums into the problem of constructing τ-robust vectors[7]. Having established the equivalence between the two problems through this reduction, we then prove our lower bound by generalizing a classic result by Euler, which states that any set of positive integers with distinct subset sums must have at least one element that is exponentially large in the number of elements in the set (Benkoski & Erdős, 1974; Frenkel, 1998). We state this result as a theorem since it may be of independent mathematical interest (missing details in Appendix B.2). We let $\mathbb{Q}^+$ denote the set of positive rational numbers and for any set S, let $||S||_\infty = \max_{s\in S}|s|$.

Theorem 5. *For any set $S \subset \mathbb{Q}^+$ with $\mu(S) > \lambda$ for some $\lambda \in [0,\infty)$, it must hold that $||S||_\infty = \Omega(\lambda 2^{|S|})$.*

The optimality of our construction of prediction vectors can now be proven as follows.

Theorem 6. *For sufficiently large N and all $\tau > 0$, any τ-robust vector $\mathbf{v}$ must have $||\mathbf{v}||_\infty = \Omega\left(e^{2^N N\tau}\right)$.*

We emphasize that this lower bound is only when a single log-loss query is used for inference (like used in Algorithm 2). This is because if multiple queries can be issued, then smaller numbers in the vector construction may suffice. For example, using N queries with vectors of the form $\mathbf{v} = [k, 1, \ldots, 1]$, where $k > e^{2N\tau}$ suffice.

[7] Recall we used the reverse direction earlier in our construction.

Algorithm 3 Label Inference with Bounded Error in the FPA(ϕ) Model (Polynomial Adversary)

1: **Input:** N, upper bound on error $\tau > 0$, ϕ
2: **Output:** Labeling $\hat{\sigma} \in \{0,1\}^N$
3: Initialize $\hat{\sigma} \leftarrow [0, \ldots, 0]$.
4: Let $m \leftarrow \min\left\{\lceil \log_2 N\rceil, \left\lfloor \log_2\left(\frac{\phi-8}{N\tau\ln 2}\right)\right\rfloor\right\}$.
5: **for** i **in** $\left\{1, \ldots, \left\lceil \frac{N}{m}\right\rceil\right\}$ **do**
6: Form a vector $\mathbf{v}$ with $\mathbf{v}_j = 3^{2^r N\tau}$ if $j = (i-1)m + r$ for some $r \in [m]$. Else, set $\mathbf{v}_j = 1$.
7: Let $\mathbf{u} = f(\mathbf{v}) = \left[\frac{v_1}{1+v_1}, \ldots, \frac{v_N}{1+v_N}\right]$. Obtain the loss score ℓ using $\mathbf{u}$ as the prediction vector.
8: Let $\Sigma_{w,m} = 0^{(w-1)m}(0+1)^m 0^{N-wm}$ (expressed as a regular expression – truncated to N bits) denote the set of all labelings that have 0's in the first $(w-1)m$ and last $(N-wm)$ positions.
9: Compute $\sigma' \leftarrow \arg\min_{\sigma\in\Sigma_{i,m}} |\mathcal{L}_{\mathbf{u}}(\sigma) - \ell|$.
10: Set $\hat{\sigma}_k = \sigma'_k$ for $k \in \{(i-1)m+1, \ldots, im\}$, while keeping all other entries in $\hat{\sigma}$ unchanged.
11: **end for**
12: **Return** $\hat{\sigma}$.

4.2. τ-Robust Label Inference under Bounded Precision with Polynomial-time Adversary

We now present our algorithm for label inference under bounded number of bits of precision. As discussed above, at most $O(\log N)$ bits must be inferred at a time since any larger amount will be infeasible in polynomial time. The idea behind our construction is similar to that in Algorithm 1, only this time, we construct prediction vectors that offer robustness to the noise added. We outline the main steps in Algorithm 3 and discuss it below. Note that Algorithm 3 only requires an upper bound on τ.

Lemma 2. *Let $\tau > 0$ be a bound on the resulting error and $m \le N$ be an integer. Let $\mathbf{v}_m = \left[3e^{2N\tau}, 3e^{4N\tau}, \ldots, 3e^{2^m N\tau}, 1, \ldots, 1\right]$. Then, for any distinct $\sigma_1, \sigma_2 \in \{0,1\}^N$, it holds that if $\sigma_1[:m] = \sigma_2[:m]$, then $\mathcal{L}_{\mathbf{v}_m}(\sigma_1) = \mathcal{L}_{\mathbf{v}_m}(\sigma_2)$. Else, we have $|\mathcal{L}_{\mathbf{v}_m}(\sigma_1) - \mathcal{L}_{\mathbf{v}_m}(\sigma_2)| > 2\tau$.*

If using the construction vector $\mathbf{v}_m$ from this above lemma, for any $m \le N$, the loss scores computed on vector $\mathbf{v}_m$ are at least 2τ apart for all labelings that differ in at least one index in $[m]$. Moreover, if the first m bits are the same for any two labelings, this lemma tells us that the loss scores will be the same as well. This is the key idea that allows unambiguous label inference in Algorithm 3. We formally state our result about the correctness of this algorithm and compute a bound on the number of bits required as follows.

Theorem 7. *For any error bounded by $\tau > 0$ and $\phi \ge 8 + \lceil N\tau\ln 2\rceil$, there exists a polynomial-time adversary (from Algorithm 3) for the τ-label inference problem in the*

$\mathsf{FPA}(\phi)$ *model using* $O\left(\frac{N}{\log N} + \frac{N}{\log(\phi/N\tau)}\right)$ *queries.*

4.3. Extension to Other Noise Models

In the previous section, we considered the case where the log-loss responses were all accurate within $\pm\tau$ for some $\tau > 0$. We now give an overview of how our attacks could be presence of random and multiplicative noise (missing details in Appendices B.4 and B.5, respectively).

Random Noise. The results above on bounded error can be extended to handle randomly generated (additive) noise. We achieve this by safeguarding against the worst case magnitude of the noise that can be added for bounded noise distributions. For cases where this distribution is unbounded, we allow for some error tolerance.

To see why this works, observe that for any bounded distribution, say over some interval $[a, b] \subset \mathbb{R}$, the error is bounded by $\max\{|a|, |b|\}$. Thus, using any $\max\{|a|, |b|\}$-robust vector will suffice. For distributions with unbounded support, however, this upper bound does not exist. However, given a failure probability $\delta > 0$, it is possible to compute this bound for vector robustness for any distribution. For example, in the APA model, in case of subexponential noise[8] with parameters λ^2 and $\nu > 0$, and $\delta \in (0, 1)$, a τ-robust vector $\mathbf{v}$ constructed as in Algorithm 2 (with $\tau = \left(2(\lambda + \nu)\sqrt{\ln 2/\delta}\right)$) will succeed in label inference with probability at least $1 - \delta$.

Multiplicative Noise. We briefly explore extending the analysis above to the case of multiplicative error. In this case, the adversary observes score ℓ that satisfies $(1-\alpha)\mathcal{L}_{\mathbf{v}}(\sigma) \leq \ell \leq (1+\alpha)\mathcal{L}_{\mathbf{v}}(\sigma)$, where the bound $\alpha \in [0, 1)$ is known to the adversary. We prove that for any $\alpha \leq 1/8$, label inference can be done, $\lceil \log_2\left(\frac{1}{\alpha}\right) - 2 \rceil$ labels at a time, using vectors that are $(2\ln 2)\alpha$-robust. Observe that when $\alpha \geq 1/4$, then no value of τ satisfies the constraint above, implying that vectors from Algorithm 3 cannot be used with any number of queries. The noise is more than what these vectors can guarantee handling.

5. Experimental Observations

We evaluate our attacks on both simulated binary labelings and real binary classification datasets fetched from the UCI machine learning dataset repository[9]. In this section, we focus on the binary label experiments, deferring the multiclass experiments to Appendix C. The results show that our algorithms are surprisingly efficient, even with a large number of datapoints. All experiments are run on a 64-bit machine with 2.6GHz 6-Core processor, using the standard IEEE-754 double precision format (1 bit for sign, 11 bits for exponent, and 53 bits for mantissa). For ensuring reproducibility, the entire experiment setup is submitted as part of the supplementary material.

Results on Simulated Binary Labelings. The first row in Figure 1 shows the plots for label inference with no noise, where we use the attack based on primes from Theorem 1. The accuracy reported is with respect to 10000 randomly generated binary labelings for each N (length of the vector to be inferred). For $N \leq 10$ all labels are correctly recovered (see Figure 1(a)). For $N \geq 47$, the maximum accuracy falls to zero. This is not unexpected because of the limited floating point precision on the machine. The run time plot shows that this inference happens in only a few milliseconds (see Figure 1(b)). The third plot in row 1 shows multi-query label inference with no noise using Algorithm 1. We use $N/5$ queries.[10] The results show that by changing the number of queries from 1 to $N/5$, we now have accuracy of 100% even up to $N = 10,000$, with the corresponding average run time shown in the fourth plot (Figure 1(d)). The average runtime is order of few milliseconds, even when $N = 10000$, demonstrating the efficiency of this attack.

The second row in Figure 1 shows the plots for robust label inference with bounded noise. The setup is similar to the first row, except that a bounded noise of scale 0.01, 0.1, or 1 is added to the log-loss scores (the noise is from about 1% to 100% of the raw log-loss score). In Figure 1(e), the label inference is performed using Algorithm 2. As the noise scale increases from 0.01 to 0.1 and 1, the length of the private vector on which we can recover all labelings correctly drops from 12 to 8 and 6, respectively. The run time displayed in the Figure 1(f) is much higher than in the unnoised case (Figure 1(b)) because the label inference algorithm for the noised setting involves iterating through 2^N labelings to find out the one that is closest to the reported noisy log-loss. The third plot shows the accuracy for multi-query label inference with bounded noise using Algorithm 3. With multiple queries calibrated by the noise scale, we are able to recover all labelings correctly in all three noise cases, with the corresponding run time displayed in Figure 1(f).

Results on Real Binary Classification Datasets. We now discuss (unnoised) label inference on real datasets. The list of datasets we use is as follows:

i. **D1** (IMDB movie review for sentiment analysis (Maas et al., 2011)) – 0 (negative review) or 1 (positive review);
ii. **D2** (Banknote Authentication) – 0 (fine) or 1 (forged);
iii. **D3** (Wisconsin Cancer) – 0 (benign) and 1 (malignant);
iv. **D4** (Haberman's Survival) – 0 (survived) and 1 (died).

As our attacks construct prediction vectors that are indepen-

[8] A random variable X is subexponential with parameters λ^2 and ν if $\mathbb{E}(e^{sX}) \leq \exp\left(\lambda^2 s^2/2\right)$ for all $|s| < 1/\nu$

[9] https://archive.ics.uci.edu/ml/machine-learning-databases

[10] We did not optimize for the number of queries – probably a smaller number of queries suffice for 100% reconstruction.

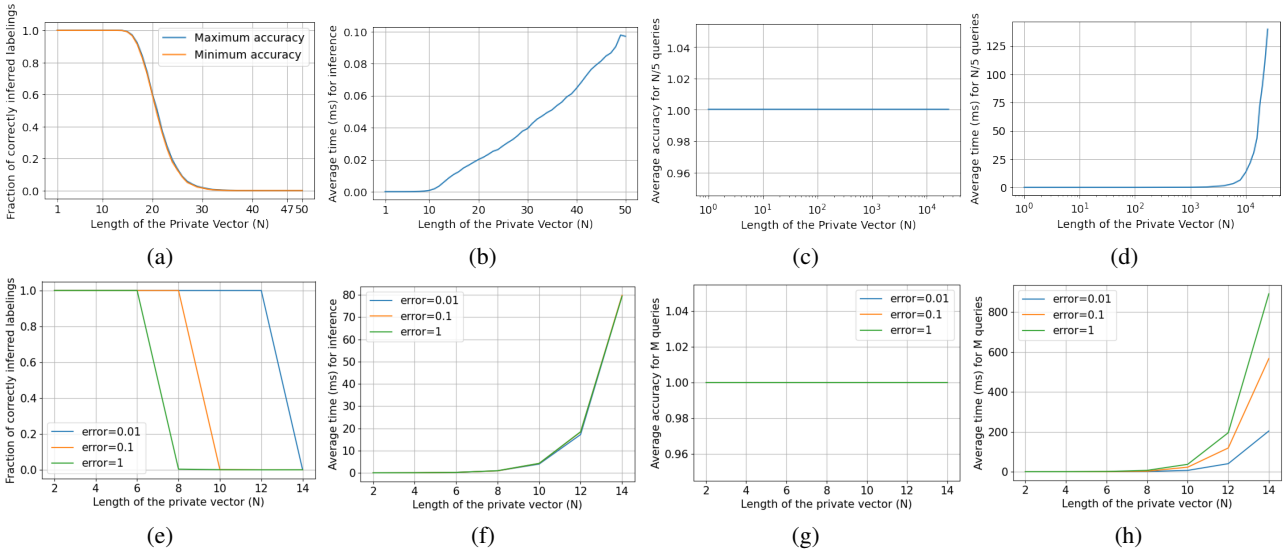


Figure 1. Results on simulated binary labelings. The first row shows the performance of the label inference without noise using for single-query (a) and (b), and multi-query (c) and (d). The second row shows the performance of label inference with bounded error (scale = 0.01, 0.1, and 1) for single-query (e) and (f), and multi-query (g) and (h). Here, M is the number of queries from Algorithm 3.

Table 2. Experimental results on real datasets using Algorithm 1. Here, N is the number of test samples in the dataset and $\mathbf{Acc}_q$ is the fraction of labels correctly inferred with q queries.

Dataset	N	$\mathbf{Acc}_1$	$\mathbf{Acc}_{N/5}$	$\mathbf{Time}_{N/5}$
D1	25,000	0.4891	1.0	53.41 ms
D2	1,372	0.4446	1.0	0.2 ms
D3	569	0.3448	1.0	0.06 ms
D4	306	0.2647	1.0	0.03 ms

dent of the dataset contents, we ignore the dataset features in our experiments. Our results are summarized in Table 2 with $N/5$ log-loss queries. In Figure 2, for **D1**, we plot the results as we double the queries from 1 to $N/5$. Note while the accuracy is low to start with, as soon as we get sufficiently large number of queries we get perfect recovery.

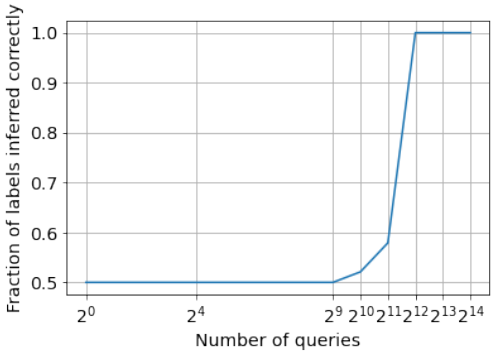


Figure 2. Accuracy of label inference on dataset **D1** as a function of the number of queries used by the adversary. For $N/5 = 5000$ queries, all labels have been correctly inferred.

6. Conclusion

In this paper, we discussed multiple label inference attacks from log-loss scores. These attacks allow an adversary to efficiently recover all test labels in a small number of queries, even when the observed scores can be erroneous due to perturbation or precision constraints (or both), without any access to the underlying dataset or model training. These results shed light into the amount of information that log-loss scores leak about the test datasets.

A natural question to ask is if there are ways to defend against such inference attacks. One way is to apply Warner's randomized response mechanism (Warner, 1965) to the test labels before computing the loss score. This way, when an adversary runs our inference attacks, the labels that it recovers will be protected by plausible deniability. Yet another way is to report the scores on a randomly chosen subset of the test dataset. In this case, bounding the loss is not possible when the size of this subset is unknown.

Acknowledgements

We would like to thank the anonymous reviewers of ICML 2021 for their feedback. We are also grateful to Prakash Krishnamurthy, Shengsheng Liu, Huiming Song and the scientific publications team at Amazon for their helpful comments on the earlier drafts of this paper and providing the necessary resources for this research.

References

Aggarwal, A., Xu, Z., Feyisetan, O., and Teissier, N. On primes, log-loss scores and (no) privacy. *Workshop on Privacy in Natural Language Processing (PrivateNLP), The 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Benkoski, S. and Erdős, P. On weird and pseudoperfect numbers. *Mathematics of Computation*, 28(126):617–623, 1974.

Blum, A. and Hardt, M. The ladder: A reliable leaderboard for machine learning competitions. *arXiv preprint arXiv:1502.04585*, 2015.

Brent, R. P. and Zimmermann, P. *Modern computer arithmetic*, volume 18. Cambridge University Press, 2010.

Dinur, I. and Nissim, K. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 202–210, 2003.

Dubhashi, D. P. and Panconesi, A. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

Frenkel, P. Integer sets with distinct subset sums. *Proceedings of the American Mathematical Society*, 126(11): 3199–3200, 1998.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Hadamard, J. Sur la distribution des zéros de la fonction $\zeta(s)$ et ses conséquences arithmétiques. *Bulletin de la Societé mathematique de France*, 24:199–220, 1896.

Hardy, G. and Wright, E. Statement of the fundamental theorem of arithmetic. *Proof of the Fundamental Theorem of Arithmetic," and" Another Proof of the Fundamental Theorem of Arithmetic*, 1(2.10):3, 1979.

Knuth, D. E. *Art of computer programming, volume 2: Seminumerical algorithms*. Addison-Wesley Professional, 2014.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011.

Matthews, G. J. and Harel, O. An examination of data confidentiality and disclosure issues related to publication of empirical roc curves. *Academic radiology*, 20(7):889–896, 2013.

Murphy, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

O'Bryant, K. A complete annotated bibliography of work related to sidon sequences. *arXiv preprint math/0407117*, 2004.

Poussin, C. J. d. L. V. *Recherches analytiques sur la théorie des nombres premiers*, volume 1. Hayez, 1897.

Robinson, J. p. and Bernstein, A. A class of binary recurrent codes with limited error propagation. *IEEE Transactions on Information Theory*, 13(1):106–113, 1967.

Warner, S. L. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

Whitehill, J. Exploiting an oracle that reports auc scores in machine learning contests. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Whitehill, J. Climbing the kaggle leaderboard by exploiting the log-loss oracle. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Appendix for "Label Inference Attacks from Log-loss Scores"

In this technical appendix, we present the missing details and omitted proofs from the main body of our paper.

A. Missing Details from Section 3

A.1. Label Inference under Bounded Precision with Polynomial-Time Adversary

For our results in this section, we make use of the well-known Prime Number Theorem (PNT), which describes the asymptotic distribution of the prime numbers among the positive integers. We use this result to quantify the number of queries required for our label inference attack in the $\mathsf{FPA}(\phi)$ model. The form of PNT which will be helpful for our analysis is stated as follows (we state it as a Lemma for use in our proof for the main result in this section).

Lemma 3. *(Hadamard, 1896; Poussin, 1897) The n^{th} prime number p_n satisfies $p_n \sim n \log n$, where $\sim$ means that the relative error of this approximation approaches 0 as n increases.*

We note that this bound on p_n implies that $p_n = \Theta(n \log n)$ also holds. We begin with proving the following lemma.

Lemma 4. *Let $M \leq N$ be a positive integer and $\mathbf{v} = [v_1, \ldots, v_M, 1, \ldots, 1] \in (\mathbb{R}^+)^N$ be a real valued vector. Then, for any labeling $\sigma \in \{0,1\}^N$, it holds that:*

$$\mathcal{L}_{\mathbf{v}}(\sigma) = -\frac{1}{N} \ln \left(\prod_{\substack{i:\sigma_i=1 \\ 1 \leq i \leq M}} v_i \right) + \textit{constant},$$

where the constant term is independent of σ.

Proof. From the definition of log-loss in Equation (2), we compute the following:

$$\begin{aligned} N\mathcal{L}_{\mathbf{v}}(\sigma) &= -\ln \left(\frac{\prod_{i:\sigma_i=1} v_i}{(1+v_1)\cdots(1+v_N)} \right) \\ &= -\ln \left(\frac{\prod_{\substack{i:\sigma_i=1 \\ 1 \leq i \leq M}} v_i}{2^{N-M}(1+v_1)\cdots(1+v_M)} \right) = -\ln \left(\prod_{\substack{i:\sigma_i=1 \\ 1 \leq i \leq M}} v_i \right) + \textit{constant}, \end{aligned}$$

where $\textit{constant} = (N-M)\ln 2 + \sum_{j=1}^{M} \ln(1+v_j)$. Dividing both sides by N, we obtain the desired result. □

A.2. Extension to the Multiclass Case

We begin by stating our result for the single-query label inference in the APA model.

Theorem 8. *There exists a polynomial-time adversary for K-ary label inference in the* APA *model using only a single log-loss query.*

Proof. We only focus on $K \geq 3$ (since $K = 2$ is equivalent to Theorem 1). Let $\sigma^* \in [K]$ be the true labeling.

Define the matrix $\mathbf{v}$ as follows:

$$\mathbf{v}_{i,k} = \frac{p_i^{k-1}}{\sum_{j=1}^{K} p_i^{j-1}} \qquad \forall (i,k) \in [N] \times [K].$$

We can perform the following algebraic manipulation of the log loss (using Definition 1) as follows:

$$\begin{aligned}\textsc{LLoss}\left(\mathbf{v};\sigma^*\right) &= \frac{-1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\left([\sigma_i^*=k]\cdot\ln\mathbf{v}_{i,k}\right) = \frac{-1}{N}\ln\left(\prod_{i=1}^{N}\mathbf{v}_{i,\sigma_i^*}\right)\\ &= \frac{-1}{N}\ln\prod_{i=1}^{N}\left(\frac{p_i^{\sigma_i^*-1}}{\sum_{j=1}^{K}p_i^{j-1}}\right) = \frac{-1}{N}\ln\left(\frac{\prod_{i=1}^{N}p_i^{\sigma_i^*-1}}{\prod_{i=1}^{N}\sum_{j=1}^{K}p_i^{j-1}}\right).\end{aligned}$$

Rearranging the terms above, we obtain the following:

$$\prod_{i=1}^{N}p_i^{\sigma_i^*-1} = \exp\left(-N\cdot\textsc{LLoss}\left(\mathbf{v};\sigma^*\right)\right)\left(\prod_{i=1}^{N}\sum_{j=1}^{K}p_i^{j-1}\right).$$

Thus, similar to the binary case, from the Fundamental theorem of arithmetic, given the value on the right hand side above, we can uniquely factorize this value to obtain the true labels on the left (which can be done efficiently since the list of primes and the number of classes are known). □

To extend this algorithm for the $\mathsf{FPA}(\phi)$ model, we first make some important observations about using multiple queries for label inference (similar to the binary case). We analyze the case where $K < N$ and focus only on inferring the first set of labels (the analysis for other cases follow using similar principles). We let $m < K$ denote an upper bound on the number of labels that we will infer in a single query (note that previously, m was the exact number of labels inferred per query). Moreover, for simplicity, we will assume that m divides both K and N. This will not change the asymptotic number of queries required by our inference attack.

Define the matrices $\mathbf{v}^{(m)}$ and $\mathbf{u}^{(m)}$ as follows:

$$\mathbf{v}_{i,k}^{(m)} = \begin{cases} p_i^{k-1} & \forall(i,k)\in[m]\times[m]\\ 1 & \text{otherwise.}\end{cases}, \text{ and}$$

$$\mathbf{u}_{i,k}^{(m)} = \frac{\mathbf{v}_{i,k}^{(m)}}{\sum_{j=1}^{K}\mathbf{v}_{i,j}^{(m)}} = \begin{cases} \frac{p_i^{k-1}}{\sum_{j=1}^{m}p_i^{j-1}+(K-m)} & \forall(i,k)\in[m]\times[m]\\ \frac{1}{\sum_{j=1}^{m}p_i^{j-1}+(K-m)} & \text{otherwise.}\end{cases}$$

Now, substituting this in Definition 1, we obtain the following:

$$\begin{aligned}\textsc{LLoss}\left(\mathbf{u}^{(m)};\sigma^*\right) &= \frac{-1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\left([\sigma_i^*=k]\cdot\ln\mathbf{u}_{i,k}^{(m)}\right)\\ &= \frac{-1}{N}\sum_{i=1}^{m}\sum_{k=1}^{m}\left([\sigma_i^*=k]\cdot\ln\mathbf{u}_{i,k}^{(m)}\right) + \frac{-1}{N}\sum_{i=m+1}^{N}\sum_{k=m+1}^{K}\left([\sigma_i^*=k]\cdot\ln\mathbf{u}_{i,k}^{(m)}\right)\\ &= \frac{-1}{N}\sum_{i=1}^{m}\ln\left(\frac{p_i^{\sigma_i^*-1}\cdot[\sigma_i^*\le m]}{\sum_{j=1}^{m}p_i^{j-1}+(K-m)}\right) - \frac{1}{N}\sum_{i=m+1}^{N}\ln\left(\frac{1}{\sum_{j=1}^{m}p_i^{j-1}+(K-m)}\right)\\ &= \frac{1}{N}\sum_{i=1}^{m}\ln\left(\sum_{j=1}^{m}p_i^{j-1}+(K-m)\right) - \frac{1}{N}\sum_{i=1}^{m}\ln\left(p_i^{\sigma_i^*-1}\cdot[\sigma_i^*\le m]\right) + \frac{(N-m)\ln K}{N}\\ &= \frac{1}{N}\sum_{i=1}^{m}\ln\left(\sum_{j=1}^{m}p_i^{j-1}+(K-m)\right) - \frac{1}{N}\ln\left(\prod_{i=1}^{m}\left(p_i^{\sigma_i^*-1}\cdot[\sigma_i^*\le m]\right)\right) + \frac{(N-m)\ln K}{N}.\end{aligned}$$

Thus, when the score $\ell^{(m)} := \textsc{LLoss}\left(\mathbf{u}^{(m)};\sigma^*\right)$ is observed, the adversary can compute the following:

$$\prod_{i=1}^{m}\left(p_i^{\sigma_i^*-1}\cdot[\sigma_i^*\le m]\right) = \exp\left(-N\ell^{(m)} + (N-m)\ln K + \sum_{i=1}^{m}\ln\alpha(i,m,K)\right),$$

where $\alpha(i,m,K) = \sum_{j=1}^{m} p_i^{j-1} + (K-m)$. Using the Fundamental Theorem of Arithmetic, the product on the left will allow recovering labels for datapoints that have labels in $[m]$.

Restatement of Theorem 3. *There exists a polynomial-time adversary for K-ary label inference in the* APA *model using only a single log-loss query. For inference in the* FPA(ϕ) *model, it suffices to issue* $O\left(1 + NKh(\phi)\right)$ *queries, where*

$$h(\phi) = O\left(\frac{(\ln \phi)^2}{(\phi + (N-K)\ln K)^{2/3}}\right).$$

Proof. The largest value of m that works in the FPA(ϕ) model can be obtained (asymptotically) by setting $(\phi-1)/2 \geq \log_2\left(\prod_{i=1}^{m}\left(p_i^{\sigma_i^*-1}\cdot[\sigma_i^* \leq m]\right)\right)$, which will allow enough resolution for this disambiguation. Simplifying this, we obtain:

$$\begin{aligned}
\phi &\geq 1 + 2\log_2\left(\exp\left(-N\ell^{(m)} + (N-m)\ln K + \sum_{i=1}^{m}\ln\alpha(i,m,K)\right)\right)\\
&\geq 1 + \frac{2}{\ln 2}\left((N-K)\ln K + \sum_{i=1}^{m}\ln\left(\sum_{j=1}^{m} p_i^{j-1}\right)\right)\\
&\geq 1 + \frac{2}{\ln 2}\left((N-K)\ln K + (m-1)\sum_{i=1}^{m}\ln p_i\right)\\
&\gtrsim 3\left((N-K)\ln K + (m-1)\sum_{i=1}^{m} i\ln i\right) \quad \text{(Using the Prime Number Theorem – see Lemma 3)}\\
&\geq c\left((N-K)\ln K + m^3\ln m\right) \qquad \text{for some constant } c>0.
\end{aligned}$$

This gives an upper bound on m by simplifying $m^3\ln m \leq \phi/c - (N-K)\ln K$, which gives $m\ln m \leq \left(\frac{\phi/c-(N-K)\ln K}{3}\right)^{1/3}$, or equivalently, $m \leq c'\left(\frac{(\phi+(N-K)\ln K)^{1/3}}{\ln\phi}\right)$ for some constant $c'>0$ (here we use the fact that $x\ln x = y$ holds when $x = \Theta(y/\ln y)$). Setting $m = m^*$, where m^* is this upper bound, we can then obtain the number of queries as $NK/(m^*)^2$, which gives $h(\phi) = 1/(m^*)^2$, or equivalently,

$$h(\phi) = O\left(\frac{(\ln \phi)^2}{(\phi + (N-K)\ln K)^{2/3}}\right).$$

□

We defer optimization of this algorithm and the detailed discussion of Robust Label Inference for multi-class classification to future work.

B. Missing Details from Section 4

B.1. τ-Robust Label Inference under Arbitrary Precision with Exponential-time Adversary

Restatement of Lemma 1. *Let* $\mathbf{v} = [v_1,\ldots,v_N]$ *be a vector with all entries distinct and positive. Define* $\ln\mathbf{v} := [\ln v_1,\ldots,\ln v_N]$. *Then, it holds that* $\Delta(\mathbf{v}) = \frac{1}{N}\mu(\ln\mathbf{v})$.

Proof. Without loss of generality, assume that σ_1 and σ_2 are such that $\mathcal{L}_{\mathbf{v}}(\sigma_1) \geq \mathcal{L}_{\mathbf{v}}(\sigma_2)$. This happens when the following holds (from Equation (2)):

$$\mathcal{L}_{\mathbf{v}}(\sigma_1) \geq \mathcal{L}_{\mathbf{v}}(\sigma_2) \iff \prod_{i:\sigma_1(i)=1} v_i \leq \prod_{j:\sigma_2(j)=1} v_j.$$

Under this condition, we can write the following:

$$N\Delta(\mathbf{v}) = \min_{\sigma_1,\sigma_2\in\{0,1\}^N} N\left(\mathcal{L}_{\mathbf{v}}(\sigma_1) - \mathcal{L}_{\mathbf{v}}(\sigma_2)\right) = \min_{\substack{\sigma_1,\sigma_2\in\{0,1\}^N \\ \sigma_1\neq\sigma_2}} \ln\exp\left(N\mathcal{L}_{\mathbf{v}}(\sigma_1) - N\mathcal{L}_{\mathbf{v}}(\sigma_2)\right)$$

$$= \min_{\sigma_1,\sigma_2\in\{0,1\}^N} \ln\left(\frac{\exp\left(-N\mathcal{L}_{\mathbf{v}}(\sigma_2)\right)}{\exp\left(-N\mathcal{L}_{\mathbf{v}}(\sigma_1)\right)}\right) = \min_{\sigma_1,\sigma_2\in\{0,1\}^N} \ln\left(\frac{\prod_{j:\sigma_2(j)=1} v_j}{\prod_{i:\sigma_1(i)=1} v_i}\right)$$

$$= \min_{\sigma_1,\sigma_2\in\{0,1\}^N} \left(\sum_{j:\sigma_2(j)=1} \ln v_j - \sum_{i:\sigma_1(i)=1} \ln v_i\right) = \mu\left(\ln\mathbf{v}\right),$$

where the last step follows from interpreting σ_1 and σ_2 as selecting elements from $\ln\mathbf{v}$ (by choosing which elements get labeled 1 and the ones that do not). □

Restatement of Theorem 4. *For any $\tau > 0$, there exists an exponential-time adversary (from Algorithm 2) for the τ-robust label inference problem in the APA model using only a single log-loss query.*

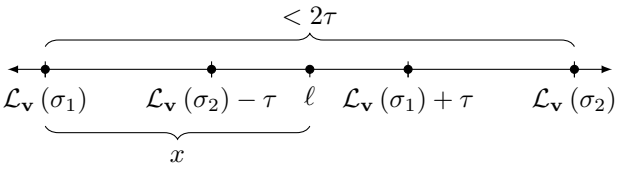
Proof. Let $S = \{s_1, \ldots, s_N\}$ be a set such that $\mu(S) \geq 2N\tau$. We know that S exists, for example, by scaling each element of $\mathcal{S}_\circ$ by $2N\tau$. Let $\mathbf{v}$ be constructed such that $v_i = 3\exp(s_i)$. To see that $\mathbf{v}$ is τ-robust, it suffices to prove that $\Delta(\mathbf{v}) > 2\tau$. Now, observe that $(\ln 3)s_i = \ln\mathbf{v}_i$. From Lemma 1, we get $\Delta(\mathbf{v}) = \frac{\ln 3}{N}\mu(S) = (2\ln 3)\tau > 2\tau$. □

B.2. Optimality of Single-Query τ-Robust Inference

We begin with the following two lemmas.

Lemma 5. *A vector $\mathbf{v}$ is τ-robust if and only if $\Delta(\mathbf{v}) > 2\tau$.*

Proof. It suffices to prove that for any $\Delta(\mathbf{v}) > 2\tau$ for any τ-robust vector $\mathbf{v}$ (since the other direction follows from our construction in Algorithm 2). We prove this by contradiction. The idea is to construct a score from which a unique labeling cannot be unambiguously derived. Let $\mathbf{v}$ be a τ-robust vector and $\mathcal{A}$ be a Turing Machine such that $\mathcal{A}(\ell, N, \tau, \mathbf{v}) = \sigma$ for all $\sigma \in \{0,1\}^N$ and all ℓ that satisfies $|\ell - \mathcal{L}_{\mathbf{v}}(\sigma)| \leq \tau$. Without loss of generality, let σ_1, σ_2 be two distinct labelings for which $0 < \mathcal{L}_{\mathbf{v}}(\sigma_2) - \mathcal{L}_{\mathbf{v}}(\sigma_1) < 2\tau$. It follows that $\mathcal{L}_{\mathbf{v}}(\sigma_2) - \tau < \mathcal{L}_{\mathbf{v}}(\sigma_1) + \tau$ (see schematic below).



Let $\ell = \left(\mathcal{L}_{\mathbf{v}}(\sigma_1) + \mathcal{L}_{\mathbf{v}}(\sigma_2)\right)/2$ and $x = \ell - \mathcal{L}_{\mathbf{v}}(\sigma_1)$. Clearly, $x < \tau$, and thus, $\mathcal{A}(\ell, N, \tau, \mathbf{v}) = \sigma_1$. However, we can show similarly that $\mathcal{L}_{\mathbf{v}}(\sigma_2) - \ell < \tau$, and hence $\mathcal{A}(\ell, N, \tau, \mathbf{v}) = \sigma_2$. This is a contradiction since $\sigma_1 \neq \sigma_2$, and hence, it must be true that $\mathcal{L}_{\mathbf{v}}(\sigma_2) - \mathcal{L}_{\mathbf{v}}(\sigma_1) > 2\tau$. The result in the Lemma statement then follows by observing that if this condition holds for any σ_1, σ_2, then $\Delta(\mathbf{v}) = \min_{\sigma_1,\sigma_2} |\mathcal{L}_{\mathbf{v}}(\sigma_2) - \mathcal{L}_{\mathbf{v}}(\sigma_1)| > 2\tau$ as well. □

Lemma 6. *For every τ-robust vector $\mathbf{v} = [v_1, \ldots, v_N]$ with distinct entries in $(1, \infty)$, it is possible, for every $s > 0$, to construct a set S such that $\mu(S) > s$.*

Proof. From Lemma 1, we know that $\Delta(\mathbf{v}) = \mu(\ln\mathbf{v})/N$. Since $\mathbf{v}$ is τ-robust, it must be true that $\Delta(\mathbf{v}) > 2\tau$ (by Lemma 5), from which we obtain that $\mu(\ln\mathbf{v}) > 2N\tau$. The result in the lemma statement then follows by setting $S = \{s_1, \ldots, s_N\}$, where $s_i = (s\ln v_i)/(2N\tau)$. □

Restatement of Theorem 5. *For any set $S \subset \mathbb{Q}^+$ with $\mu(S) > \lambda$ for some $\lambda \in [0, \infty)$, it holds that $||S||_\infty = \Omega(\lambda 2^{|S|})$.*

Proof. We prove this result in three steps. First, we show that the bound holds whenever λ as well as all the elements in S are positive integers. To see this, observe that if $\mu(S) > 1 > 0$, Euler's result gives us that $||S||_\infty = \Omega(2^{|S|})$. Now, let

$S' = \{s\lambda \mid s \in S\}$. Then, $\mu(S') > \lambda$. If we suppose that $||S'||_\infty = o(\lambda 2^{|S|})$, then $||S'/T||_\infty = ||S||_\infty = o(2^{|S|})$, which is a contradiction to above.

Next, assume that $S \subset \mathbb{Q}^+$ and λ still be an integer. In this case, each element of S can be written as $s_i = p_i/q_i$ (in the lowest form). Let $Q = q_1 q_2 \cdots q_N$ and $S' = \{sQ \mid s \in S\}$. Then, each element of S' is an integer and since $\mu(S') > \lambda Q$, which is also an integer, from the discussion above, we have $||S'||_\infty = \Omega(\lambda Q 2^N)$, which gives $||S||_\infty = \Omega(\lambda 2^N)$.

Finally, let λ be an arbitrary positive real. In this case, $\mu(S) > \lambda$ implies $\mu(S) > \lfloor\lambda\rfloor$, which is an integer. Hence, $||S||_\infty = \Omega(\lfloor\lambda\rfloor 2^N) = \Omega(\lambda 2^N)$, as desired. □

Restatement of Theorem 6: *For sufficiently large N and all $\tau > 0$, any τ-robust vector $\mathbf{v}$ must have $||\mathbf{v}||_\infty = \Omega\left(e^{2^N N\tau}\right)$.*

Proof. From Lemma 1, we know that for any vector $\mathbf{v}$ with distinct positive entries, it holds that $\Delta(\mathbf{v}) = \mu(\ln \mathbf{v})/N$. For this vector to be τ-robust, we argue that $\Delta(\mathbf{v}) > 2\tau$ (see Lemma 5), which is the same as setting $\mu(\ln \mathbf{v}) > 2N\tau$. From Theorem 5, for this to hold, it must be true that $||\ln \mathbf{v}||_\infty = \Omega\left(2^N N\tau\right)$. From the definition of $\ln \mathbf{v}$, this gives $||\mathbf{v}||_\infty = \Omega\left(\exp\left(2^N N\tau\right)\right)$, as desired. □

B.3. τ-Robust Label Inference under Bounded Precision with Polynomial-time Adversary

We prove an additional lemma before proving our next result.

Lemma 7. *Let $\tau > 0$ be a bound on the resulting error and $m \leq N \in \mathbb{Z}^+$ be an integer. Let $\mathbf{v}_m = \left[3e^{2m\tau}, 3e^{4m\tau}, \ldots, 3e^{2^m m\tau}, 1, \ldots, 1\right]$. Then, for any distinct $\sigma_1, \sigma_2 \in \{0,1\}^N$ where $\sigma_1[:m] \neq \sigma_2[:m]$ (i.e. σ_1 and σ_2 differ some index $i \leq m$), it holds that:*

$$|\mathcal{L}_{\mathbf{v}_m}(\sigma_1) - \mathcal{L}_{\mathbf{v}_m}(\sigma_2)| > 2m\tau/N.$$

Proof. From Lemma 4, we can write that:

$$|\mathcal{L}_{\mathbf{v}_m}(\sigma_1) - \mathcal{L}_{\mathbf{v}_m}(\sigma_2)| = \frac{1}{N}\left|\sum_{\substack{i:\sigma_1(i)=1\\1\leq i\leq m}} \ln v_m(i) - \sum_{\substack{k:\sigma_2(k)=1\\1\leq k\leq m}} \ln v_m(k)\right|$$

$$= \frac{\ln 3}{N}\left|\sum_{\substack{i:\sigma_1(i)=1\\1\leq i\leq m}} 2^i m\tau - \sum_{\substack{k:\sigma_2(k)=1\\1\leq k\leq m}} 2^k m\tau\right| = \frac{m\tau \ln 3}{N}\left|\sum_{\substack{i:\sigma_1(i)=1\\1\leq i\leq m}} 2^i - \sum_{\substack{k:\sigma_2(k)=1\\1\leq k\leq m}} 2^k\right| > \frac{2m\tau}{N},$$

where the last step follows from the fact that σ_1 and σ_2 differ in at least one element amongst the first m entries. □

Restatement of Lemma 2. *Let $\tau > 0$ be a bound on the resulting error and $m \leq N \in \mathbb{Z}^+$ be an integer. Let $\mathbf{v}_m = \left[3e^{2N\tau}, 3e^{4N\tau}, \ldots, 3\exp\left(2^m N\tau\right), 1, \ldots, 1\right]$. Then, for any distinct $\sigma_1, \sigma_2 \in \{0,1\}^N$, the following hold:*

(a) If $\sigma_1[:m] = \sigma_2[:m]$, then $\mathcal{L}_{\mathbf{v}_m}(\sigma_1) = \mathcal{L}_{\mathbf{v}_m}(\sigma_2)$.

(b) Else, we have $|\mathcal{L}_{\mathbf{v}_m}(\sigma_1) - \mathcal{L}_{\mathbf{v}_m}(\sigma_2)| > 2\tau$.

The proof directly follows from Lemma 7.

Restatement of Theorem 7. *For any error bounded by $\tau > 0$ and $\phi \geq 8 + \lceil N\tau \ln 2\rceil$, there exists a polynomial-time adversary (from Algorithm 3) for the τ-label inference problem in the $\mathsf{FPA}(\phi)$ model using $O\left(\frac{N}{\log N} + \frac{N}{\log(\phi/N\tau)}\right)$ queries.*

Proof. To see how many queries suffice, we first compute the number of bits necessary to represent the prediction vector and the loss scores up to sufficient resolution. For $\mathbf{u}_m = f(\mathbf{v}_m)$, we observe the following for sufficiently large N, in

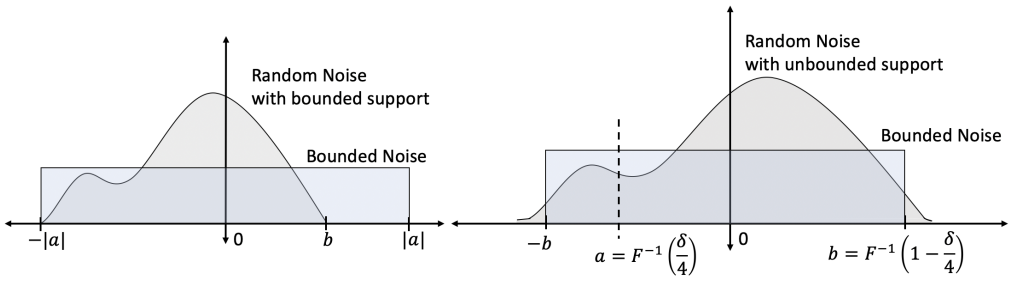


Figure 3. Schematic of the reduction of the random noise case to bounded noise for label inference. The picture on the left allows for a construction of a robust vector, whereas the picture on the right allows for robust vectors.

particular, when $N\tau > \ln(16)$:

$$\min_{\substack{i,j\in[N]\\ \mathbf{u}_m(i)\neq\mathbf{u}_m(j)}} |\mathbf{u}_m(i)-\mathbf{u}_m(j)| = \frac{3e^{2^m N\tau}}{1+3e^{2^m N\tau}} - \frac{3e^{2^{m-1}N\tau}}{1+3e^{2^{m-1}N\tau}} \geq \frac{\exp\left(-2^{m-1}N\tau\right)}{16},$$

which, to work within the $\mathsf{FPA}(\phi)$ model, would require $(\phi-1)/2 \geq 4+2^{m-1}N\tau\ln 2$, or equivalently, $\phi \geq 8+2^m N\tau\ln 2$ bits. Moreover, we can bound the loss computed on $\mathbf{v}_m$ on any σ as follows:

$$\max_{\sigma\in\{0,1\}^N} \mathcal{L}_{\mathbf{v}_m}(\sigma) \leq \frac{1}{N}\ln\left(2^{N-m}\prod_{j=1}^{m}(1+3e^{2^j N\tau})\right) \leq 2\left(\ln 2 + (2^m-1)\tau\right) \leq 2^{m+1}\tau,$$

which can be represented (along with sufficient resolution, since $\Delta(\mathbf{v}_m) \geq 2\tau$ by construction) within the number of bits described above. Thus, in the $\mathsf{FPA}(\phi)$ model, the maximum value of m that we can set is obtained by setting $2^m N\tau\ln 2 + 8 \leq \phi$, which gives:

$$m_{max} = \left\lfloor \log_2\left(\frac{\phi-8}{N\tau\ln 2}\right)\right\rfloor.$$

From this, we obtain that $\frac{N}{m_{max}} = O\left(\frac{N}{\log N} + \frac{N}{\log(\phi/N\tau)}\right)$ queries suffice.

Given this bound, it suffices to show that in the i^{th} iteration of the for-loop in Algorithm 3, the vector σ' contains the true labels for indices in $\{(i-1)m+1,\ldots,im\}$. Without loss of generality, assume $i=1$, so that the vector $\mathbf{v} = \left[3e^{2N\tau}, 3e^{4N\tau},\ldots,3\exp\left(2^m N\tau\right),1,\ldots,1\right]$. From Lemma 2(b), it follows that for any distinct $\sigma_1,\sigma_2\in\{0,1\}^N$ where $\sigma_1[:m] \neq \sigma_2[:m]$, it holds that:

$$|\mathcal{L}_{\mathbf{v}}(\sigma_1) - \mathcal{L}_{\mathbf{v}}(\sigma_2)| > 2\tau. \tag{3}$$

Thus, when the loss ℓ is observed on $\mathbf{u} = f(\mathbf{v})$ and $\arg\min_\sigma |\mathcal{L}_{\mathbf{u}}(\sigma) - \ell| = \{\sigma^{(1)},\ldots,\sigma^{(k)}\}$, then it must be true that $\sigma^{(k_1)}[:m] = \sigma^{(k_1)}[:m]$ for all $k_1,k_2\in[k]$ (or else, it would contradict the inequality in (3)). Furthermore, from Lemma 2(a), it follows that the true labeling must be present in the set $\{\sigma^{(1)},\ldots,\sigma^{(k)}\}$. Thus, at the end of iteration $i=1$, the vector $\hat{\sigma}$ has recovered the first m bits in σ. For all other iterations, note that the vector $\mathbf{v}$ is just a cyclic rotation to the right by m elements, and hence, the bits in σ are recovered, m at a time, in Algorithm 3. □

B.4. Extension to Other Noise Models: Random Noise

The discussion of norm-bounded noise above allows easy extension to handling randomly generated (additive) noise. We achieve this by safeguarding against the worst case magnitude of the noise that can be added for bounded noise distributions. For cases where this distribution is unbounded, we allow for some error tolerance. We begin with formally defining the label inference problem in this setting and then present our analysis. We will restrict ourselves to arbitrary precision in this section and defer the extension to $\mathsf{FPA}(\phi)$ for future work.

Definition 6. *Let $\mathcal{D}$ be a probability distribution and $\delta \in [0,1]$ be the error tolerance. We say that a vector $\mathbf{v}$ is $\mathcal{D}$-robust if for all $\tau \sim \mathcal{D}$ and $\sigma \in \{0,1\}^N$, there exists an adversary (Turing Machine) $\mathcal{A}$ that can recover (within a single query) σ from $\ell = \mathcal{L}_{\mathbf{v}}(\sigma) + \tau$ with probability at least $1-\delta$,* i.e. $\Pr[\mathcal{A}(\ell, N, \mathcal{D}, \mathbf{v}) = \sigma] \geq 1-\delta$.

Our main result in the section is stated in the theorem below (see Figure 3 for a proof sketch).

Theorem 9. *For any error tolerance $\delta \in (0,1)$, it is possible to construct a $\mathcal{D}$-robust vector for all distributions $\mathcal{D}$.*

Proof. We first look at the case when $\mathcal{D}$ has bounded support over $\mathbb{R}$. Let $[a,b] \subset \mathbb{R}$ be the support of $\mathcal{D}$. Let $\tau = \max\{|a|,|b|\}$. Then, for any $\eta \sim \mathcal{D}$, it holds that $|(\mathcal{L}_{\mathbf{v}}(\sigma)+\eta) - \mathcal{L}_{\mathbf{v}}(\sigma)| = |\eta| \leq \tau$ and hence, any τ-robust vector is also $\mathcal{D}$-robust.

For the case when $\mathcal{D}$ has unbounded support, let $\eta \sim \mathcal{D}$, and $a, b \in \mathbb{R}$ be such that $\Pr(\eta \in (-\infty, a) \cup (b, \infty)) < \delta$. To see that this can be done, let $F : \mathbb{R} \to [0,1]$ be the cumulative distribution function for $\mathcal{D}$, which is a nondecreasing, right-continuous function with $F(-\infty) = 0$, $F(\infty) = 1$, and $F^{-1}(p) = \inf\{x \in \mathbb{R} : F(x) \geq p\}$. For any fixed $\delta \in (0,1)$, we can set $a = F^{-1}(\delta/4)$ and $b = F^{-1}(1-\delta/4)$ so that $\Pr(\eta \in [a,b]) \geq \delta/2$, which makes $\Pr(\eta \in (-\infty, a) \cup (b, \infty)) \leq \delta/2 < \delta$. Note that a and b are always finite by definition of the cumulative distribution function as the point mass is zero on $\pm\infty$. □

To see why this works, observe that for any bounded distribution, say over some interval $[a,b] \subset \mathbb{R}$, the amount of noise added is never more than $\max\{|a|,|b|\}$. Thus, any $\max\{|a|,|b|\}$-robust vector can unambiguously recover the labels from the noised scores. For distributions with unbounded support, however, such an upper bound does not exist. Given the error tolerance, it is possible to compute this bound for any distribution and robustness be defined accordingly. We explain this for subexponential noise below.

Recall that a random variable $X \in \mathbb{R}$ is said to be subexponential (denoted $X \sim \mathsf{subE}(\lambda^2, \nu)$) with parameters $\lambda^2, \nu > 0$ if $\mathbb{E}(X) = 0$ and its moment generating function satisfies $\mathbb{E}(e^{sX}) \leq \exp\left(\lambda^2 s^2/2\right)$ for all $|s| < 1/\nu$. Given any tolerance $\delta \in (0,1)$, it can be shown that there exists a $\mathsf{subE}(\lambda^2, \nu)$-robust vector for all $\lambda, \nu > 0$. We formally state this result below.

Theorem 10. *For all $\lambda, \nu > 0$ and $\delta \in (0,1)$, any vector that is $\left(2(\lambda+\nu)\sqrt{\ln\left(\frac{2}{\delta}\right)}\right)$-robust is $\mathsf{subE}(\lambda^2, \nu)$-robust as well. In particular, with probability at least $1 - o(1)$, there exists an adversary for the single query Robust Label Inference problem for $\mathsf{subE}(\lambda^2, \nu)$ noise.*

Proof. We begin with reminding the reader a concentration bound that will be useful in our proof.

Lemma 8 ((Dubhashi & Panconesi, 2009))**.** *Let $Y \sim \mathsf{subE}(\lambda^2, \nu)$ be a zero-mean random variables for some $\lambda, \nu > 0$. Then, for all $t > 0$, the following holds:*

$$\Pr(|Y| > t) \leq 2\exp\left(-\frac{1}{2}\left(\frac{t^2}{\lambda^2} \wedge \frac{t}{\nu}\right)\right),$$

where $a \wedge b := \min\{a,b\}$.

Given this result, we follow the general construction shown in the proof of Theorem 9, we compute $a > 0$ such that for any $Y \sim \mathsf{subE}(\lambda^2, \nu)$, it holds that $\Pr(Y \in (-a, a)) \geq 1-\delta$. This would allow all a-robust vectors to be robust with probability at least $1-\delta$, as needed.

To compute a, observe that from Lemma 8, we can set $t = a$ and set the bound on the probability to be at most δ, as follows:

$$\Pr(|Y| > a) \leq 2\exp\left(-\frac{1}{2}\left(\frac{a^2}{\lambda^2} \wedge \frac{a}{\nu}\right)\right) < \delta.$$

A simple algebraic manipulation for the inequality on the right gives the condition that $\frac{a^2}{\lambda^2} \wedge \frac{a}{\nu} > 2\ln\left(\frac{2}{\delta}\right)$. We solve the two cases here separately. If $a < \lambda^2/\nu$, then $\frac{a^2}{\lambda^2} \wedge \frac{a}{\nu} = \frac{a^2}{\lambda^2}$, and hence, this gives $a < \lambda\sqrt{2\ln\left(\frac{2}{\delta}\right)}$. Else, we have $\frac{a^2}{\lambda^2} \wedge \frac{a}{\nu} = \frac{a}{\nu}$, which gives $a < 2\nu\ln\left(\frac{2}{\delta}\right)$. It suffices to take the maximum of these two limits and hence, $\max\{\lambda\sqrt{2\ln\left(\frac{2}{\delta}\right)}, 2\nu\ln\left(\frac{2}{\delta}\right)\}$

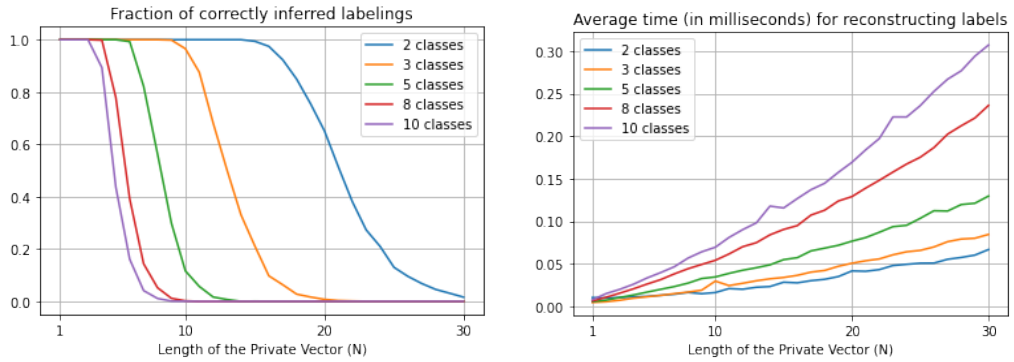


Figure 4. Empirical results for multi-class label inference. The plot on the top shows that due to fixed-precision, the accuracy drops to zero faster as the number of classes increases, since larger primes powers are used in the prediction vectors. The plot below shows that the inference time increases with the number of classes.

works. We can further simplify this expression using the upper bound $2(\lambda+\nu)\sqrt{\ln\left(\frac{2}{\delta}\right)}$ which follows from the fact that $\delta \in (0,1)$. □

For example, let $Z = \mathcal{N}(0,1)$ denote the standard normal random variable. Then, we know that the Chi-squared random variable with one degree of freedom follows the law for $Z^2 = \mathsf{subE}(2,4)$. Thus, from the Corollary above, we obtain that for with probability at least $1 - o(1)$, any vector that can handle up to $12\sqrt{\ln N}$ amount of bounded noise can also handle Z^2 noise. If we are allowed up to, say $\delta = 0.1$, then any $12\sqrt{\ln 20} \approx 20.77$-robust vector suffices.

B.5. Extension to Other Noise Models: Multiplicative Noise

We briefly explore extending the analysis above to the case of multiplicative noise. In this case, the adversary observes score ℓ that satisfies:

$$(1-\alpha_1)\mathcal{L}_{\mathbf{v}}(\sigma) \le \ell \le (1+\alpha_2)\mathcal{L}_{\mathbf{v}}(\sigma),$$

where $\alpha_1 \in [0,1)$ and $\alpha_2 \ge 0$ are the known bounds on the rate of the multiplicative noise. We show that if the rate is small enough, then it is possible to reduce this case to that of additive noise. To see this, we begin with rewriting the inequality above as $|\ell - \mathcal{L}_{\mathbf{v}}(\sigma)| \le \alpha^* \cdot |\mathcal{L}_{\mathbf{v}}(\sigma)|$, where $\alpha^* = \max\{\alpha_1, \alpha_2\}$. Now, if we use the construction of a τ-robust vector from Algorithm 2, it is easy to compute that $|\mathcal{L}_{\mathbf{v}}(\sigma)| \le 2^{N+1}\tau$ for any σ, which would require ensuring $2^{N+1}\tau\alpha^* < 2\tau$ – this is not possible for any $\tau > 0$ unless $\alpha^* < 2^{-N}$. Thus, we require multiple queries.

Suppose we only infer $1 \le m \le N$ labels in a single query (using $M = m$ in Algorithm 3), then this will require $\left(\ln 2 + 2^{m+1}\tau\right)\alpha^*$ to be at most τ. It suffices to set $\alpha^* \le 1/2^{m+2}$. Then, the quantity on the left becomes at most

$$\frac{\ln 2 + 2^{m+1}\tau}{2^{m+2}} = \frac{\ln 2}{2^{m+2}} + \frac{\tau}{2} \le \tau,$$

whenever $\tau \ge \frac{\ln 2}{2^{m+1}}$. We state this formally as follows.

Theorem 11. *For any $\alpha^* \le \frac{1}{8}$, label inference can be done, $\left\lceil \log_2\left(\frac{1}{\alpha^*}\right) - 2 \right\rceil$ labels at a time, using vectors that are $(2\ln 2)\alpha^*$-robust.*

Observe that when $\alpha \ge \frac{1}{4}$, then no value of τ satisfies the constraint above, implying that robust vectors from Algorithm 3 cannot be used with any number of queries. The noise is more than what these vectors can guarantee handling. We defer label inference for this case to future work.

C. Experimental Results for Multi-Class Label Inference

Similar to Section 5, we evaluate our attacks on simulated labelings (with no noise) in the multi-class setting (see Figure 4). The results show that our algorithms are efficient, even with a large number of datapoints. All experiments are run on a

64-bit machine with 2.6GHz 6-Core processor, using the standard IEEE-754 double precision format (1 bit for sign, 11 bits for exponent, and 53 bits for mantissa). For ensuring reproducibility, the entire experiment setup is submitted as part of the supplementary material.

The plot on the left in Figure 4 shows the accuracy of label inference, where we use the attack based on primes from Theorem 8. The accuracy reported is with respect to 100 randomly generated labelings for each N (length of the vector to be inferred). We vary the number of classes from 2 to 10 for these experiments. Similar to the results in Figure 1, the maximum accuracy falls to zero when N increases, because of the limited floating point precision on the machine. As the number of classes increase, this drop happens sooner (*i.e.* for a smaller N), because the powers of primes get larger.

The plot on the right shows the run time for our attack. The results show that this inference happens in only a few milliseconds.