IS THE PERFORMANCE OF MY DEEP NETWORK TOO GOOD TO BE TRUE? A DIRECT APPROACH TO ESTIMATING THE BAYES ERROR IN BINARY CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

There is a fundamental limitation in the prediction performance that a machine learning model can achieve due to the inevitable uncertainty of the prediction target. In classification problems, this can be characterized by the Bayes error, which is the best achievable error with any classifier. The Bayes error can be used as a criterion to evaluate classifiers with state-of-the-art performance and can be used to detect test set overfitting. We propose a simple and direct Bayes error estimator, where we just take the mean of the labels that show *uncertainty* of the classes. Our flexible approach enables us to perform Bayes error estimation even for weakly supervised data. In contrast to others, our method is model-free and even instance-free. Moreover, it has no hyperparameters and gives a more accurate estimate of the Bayes error than several baselines empirically. Experiments using our method suggest that a recently proposed classifier, the Vision Transformer, may have reached (or is about to reach) the Bayes error for the CIFAR-10H dataset.

1 INTRODUCTION

Comparing the prediction performance of a deep neural network with the state-of-the-art (SOTA) performance is a common approach to validating advances brought by a new proposal in machine learning research. By definition, SOTA performance such as error or $1-$AUC monotonically decreases over time, but there is a fundamental limitation in the prediction performance that a machine learning model can achieve. Hence, it is important to figure out how close the current SOTA performance is to the underlying best performance achievable (Theisen et al., 2021). For example, Henighan et al. Henighan et al. (2020) studied the scaling law (Kaplan et al., 2020) of Transformers (Vaswani et al., 2017) by distinguishing the *reducible* and *irreducible* errors.

In classification problems, one way to characterize the irreducible part is the *Bayes error* (Cover & Hart, 1967; Fukunaga, 1990; Mohri et al., 2018), which is the best achievable expected error with any measurable function. The Bayes error will become zero in the special case when class distributions are completely separable, but in practice, we usually deal with complex distributions with some class overlap. Natural images tend to have a lower Bayes error (e.g., 0.21% test error on MNIST (LeCun et al., 1998) in Wan et al. (Wan et al., 2013)), while medical images tend to have a higher Bayes error, since even medical technologists can disagree on the ground truth label (Sasada et al., 2018). If the current model's performance has already achieved the Bayes error, it is meaningless to aim for further error improvement. We may even find out that the model's error has *exceeded* the Bayes error. This may imply *test set overfitting* (Recht et al., 2018) is taking place. Knowing the Bayes error will be helpful to avoid such common pitfalls.

Estimating the Bayes error has been a topic of interest in the research community (Fukunaga & Hostetler, 1975; Fukunaga, 1990; Devijver, 1985; Berisha et al., 2016). To the best of our knowledge, all previous papers have proposed ways to estimate the Bayes error from a dataset consisting of pairs of instances and their hard labels. When instances and hard labels are available, one can also train a supervised classifier, which is known to approach the Bayes classifier (that achieves the Bayes error) with sufficient training data provided that the model is correctly specified.

This is an interesting research problem from the point of view of *Vapnik's principle*[1] (Vapnik, 2000) since we can derive the Bayes error from the Bayes classifier (and the underlying distribution) while we cannot recover the Bayes classifier from the knowledge of the Bayes error, which is just a scalar. How can we take full advantage of this property? While the Bayes error is usually defined as the best achievable expected error with any measurable function, it is known to be equivalent to the expectation of the minimum of class-posteriors with respect to classes for binary classification. Inspired by Vapnik's principle, our main idea is to skip the intermediate step of learning a function model, and we directly approximate the minimum of the class-posteriors by using soft labels (corresponding to the class probability) or uncertainty labels (corresponding to the class uncertainty[2]).

Our proposed method has two benefits. Firstly, our method is *model-free*. Since we do not learn a model, we can escape the curse of dimensionality, while dealing with high-dimensional instance would cause issues such as overfitting if we were to train a model. High dimensionality may cause performance deterioration for other Bayes error estimation methods due to divergence estimation. We experimentally show how our method can more accurately estimate the Bayes error than baselines that utilize instances and soft labels. Our model-free method is also extremely fast since we do not have any hyperparameters to tune nor a function model to train.

The second benefit is a more practical one: our method is completely *instance-free*. Suppose our final goal is to estimate the Bayes error instead of training a classifier. In that case, we do not need to collect instance-label pairs, and it may be less costly to collect soft/uncertainty labels without instances. Dealing with instances can cause privacy issues, and it can be expensive due to data storage costs especially when they are high-dimensional or can come in large quantities. It may lead to security costs to protect instances from a data breach.

As an example of an instance-free scenario, we can consider doctors who are diagnosing patients by inspecting symptoms and asking questions, without explicitly collecting or storing the patients' data in the database. In this scenario, the hospital will only have the decisions and confidence of doctors, which can be used as soft labels.

The contributions of the paper is as follows. We first propose a direct way to estimate the Bayes error from soft (or uncertainty) labels without a model nor instances. We show that our estimator is unbiased and consistent. In practice, collecting soft/uncertainty labels can be difficult since the labelling process can become noisy. We propose a modified estimator that is still unbiased and consistent even when the soft labels are contaminated with zero-mean noise. We also show that our approach can be applied to other classification problems, such as weakly supervised learning. Finally, we show the proposed methods' behavior through various experiments. Our results suggest that Vision Transformer (Dosovitskiy et al., 2021) has reached or is about to reach the Bayes error for the CIFAR-10H dataset (Peterson et al., 2019).

2 Background

Before we discuss our setup, we will review ordinary binary classification from positive and negative data and binary classification from positive-confidence data in this section.

Learning from positive-negative data Suppose a pair of d-dimensional instance $\boldsymbol{x} \in \mathbb{R}^d$ and its class label $y \in \{+1, -1\}$ follows an unknown probability distribution with probability density function $p(\boldsymbol{x}, y)$. The goal of binary classification is to train a binary classifier $g\colon \mathbb{R}^d \to \mathbb{R}$ so that the following classification risk $R(g)$ will be minimized: $R(g) = \mathbb{E}_{p(\boldsymbol{x},y)}[\ell(yg(\boldsymbol{x}))]$, where ℓ is a margin-based surrogate loss function such as the logistic loss $\ell_{\mathrm{L}}(z) = \log(1 + \exp(-z))$ or the squared loss $\ell_{\mathrm{S}}(z) = (z-1)^2$. The classification error is $R_{01}(g) = \mathbb{E}_{p(\boldsymbol{x},y)}[\ell_{01}(yg(\boldsymbol{x}))]$, where we use the zero-one loss ℓ_{01}, which takes 0 when $y \cdot g(\boldsymbol{x}) > 0$ and 1 otherwise. In empirical risk minimization, the following estimator of $R(g)$ is often used:

$$\hat{R}(g) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i g(\boldsymbol{x}_i)), \tag{1}$$

where $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}, y)$.

[1]The Vapnik's principle is known as the following: "When solving a given problem, try to avoid solving a more general problem as an intermediate step." (Sec. 1.2 of Vapnik, 2000)

[2]An uncertainty label is 1 subtracted by the dominant class posterior.

In this paper, our focus is not on the classifier itself but on the value of $R_{01}(g)$ when a good classifier is learned. This can be nicely captured by the notion of the Bayes error, defined as the following.

Definition 2.1. (Mohri et al., 2018) The Bayes error is the infimum of the classification risk with respect to all measurable functions: $\beta = \inf_{g\colon \mathbb{R}^d \to \mathbb{R},\ \text{measurable}} \mathbb{E}_{p(\boldsymbol{x},y)} \left[\ell_{01}\left(yg(\boldsymbol{x})\right)\right]$.

Learning from positive-confidence data In positive-confidence classification (Ishida et al., 2018; Shinoda et al., 2020), we only have access to positive data equipped with confidence $\{(\boldsymbol{x}_i, r_i)\}_{i=1}^{n_+}$, where $\boldsymbol{x}_i$ is a positive instance drawn independently from $p(\boldsymbol{x} \mid y = +1)$ and r_i is the positive confidence given by $r_i = p(y = +1|\boldsymbol{x}_i)$. Since we only have positive instances, we cannot directly minimize the empirical risk in Eq. 1. However, Ishida et al. (2018) showed that the following theorem holds.

Theorem 2.2. *(Ishida et al., 2018) The classification risk can be expressed as*

$$R(g) = \pi_+ \mathbb{E}_+ \left[\ell(g(\boldsymbol{x})) + \tfrac{1-r(\boldsymbol{x})}{r(\boldsymbol{x})}\ell(-g(\boldsymbol{x}))\right] \tag{2}$$

if we have $r(\boldsymbol{x}) = p(y = +1|\boldsymbol{x}) \neq 0$ for all $\boldsymbol{x}$ sampled from $p(\boldsymbol{x})$. $\mathbb{E}_+$ is the expectation over the positive distribution $p(\boldsymbol{x}|y = +1)$ and $\pi_+ = p(y = +1)$.

Since the expectation is over $p(\boldsymbol{x}|y = +1)$, we do not have to prepare negative data. Although π_+ is usually unknown, we do not need this knowledge since π_+ can be regarded as a constant when minimizing Eq. 2 to learn a classifier.

3 Estimation with ordinary data

In this section, we propose our direct approach to Bayes error estimation.

Clean soft labels First of all, it is known that the Bayes error can be rewritten as follows (Mohri et al., 2018):

$$\beta = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\min\{p(y = +1|\boldsymbol{x}), p(y = -1|\boldsymbol{x})\}\right]. \tag{3}$$

This means that we can express the Bayes error β without explicitly involving the classifier g. If we assume that we have soft labels, which are $\{c_i\}_{i=1}^n$, where $c_i = p(y = +1|\boldsymbol{x}_i)$ and $\{\boldsymbol{x}_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x})$, we can construct an extremely straightforward estimator for Bayes error estimation:

$$\hat{\beta} = \tfrac{1}{n}\textstyle\sum_{i=1}^n \min\{c_i, 1 - c_i\}. \tag{4}$$

In fact, Eq. 4 is just the empirical version of Eq. 3. However, the significance lies in the observation that this can be performed in a *model-free* and *instance-free* fashion, since taking the averages of the smaller one out of c_i and $1 - c_i$ in Eq. 4 does not require model training and does not involve $\boldsymbol{x}_i$ at all.

Note that we can further weaken the supervision by sampling *uncertainty labels*, i.e., $c'_i = \min_{y\in\{\pm1\}} p(y|\boldsymbol{x}_i)$. Practically, this would mean the label provider does not need to reveal the soft label. Although the two are similar, we can see how uncertainty labels are weaker by the fact that we cannot recover $p(y = +1|\boldsymbol{x})$ from $\min_{y\in\{\pm1\}} p(y|\boldsymbol{x})$. Furthermore, we can recover $\pi_+ = p(y = +1)$ by the expectation of soft labels $\mathbb{E}_{p(\boldsymbol{x})}[p(y = +1|\boldsymbol{x})] = \pi_+$, but π_+ cannot be recovered by averaging uncertainty labels.

We can also enjoy theoretical properties of our estimator, which we show in the following propositions.

Proposition 3.1. *The estimator $\hat{\beta}$ is an unbiased estimator of the Bayes error.*

Proposition 3.2. *For any $\delta > 0$, with probability at least $1 - \delta$, we have, $|\hat{\beta} - \beta| \leq \sqrt{\frac{1}{8n}\log\frac{2}{\delta}}$.*

The proofs are shown in App. B.1 and App. B.2, respectively. This implies that $\hat{\beta}$ is a consistent estimator to the Bayes error, i.e., $\forall\epsilon > 0$, $\lim_{n\to\infty} P(|\hat{\beta} - \beta| \geq \epsilon) = 0$. The rate of convergence is $|\hat{\beta} - \beta| = \mathcal{O}_P(1/\sqrt{n})$, where $\mathcal{O}_P$ is the stochastic big-O notation, cf. App. C. This rate is the same as classifier training and is the optimal parametric rate (Mendelson, 2008) although our method can be regarded as non-parametric. We are also free from the curse of dimensionality since the instance dimension d does not appear in the bound.

Noisy soft labels So far, our discussions were based on an assumption that we can collect soft (or uncertainty) labels that is precisely the class-posterior c_i (or c'_i). This may not be satisfied in practice, and we will present a way to weaken this assumption.

We consider a situation where noise ξ_i is added to the class-posterior values. We observe noisy soft labels $\{u_i\}_{i=1}^n$ where $u_i = c_i + \xi_i$. Since we ask labellers to give us a probability, a constraint is $0 \leq u_i \leq 1$, and we assume $\mathbb{E}[\xi_i|c_i] = 0$. This implies that ξ_i becomes smaller as c_i approaches 1 or 0. If we naively use these noisy versions, our estimator becomes

$$\tilde{\beta} = \frac{1}{n}\sum_{i=1}^n \left[\min\left(u_i, 1-u_i\right)\right]. \tag{5}$$

However, we can show the following negative result.

Theorem 3.3. *The estimator $\tilde{\beta}$ is generally a* biased *estimator of the Bayes error:* $\mathbb{E}_{\{(c_i,\xi_i)\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(c,\xi)}[\tilde{\beta}] \neq \beta$.

This is basically due to the noise inside the nonlinear $\min$ operation. We show the proof in App. B.3.

This issue can be fixed if we have an additional binary signal that shows if it is likely to be more positive or negative. Formally, if we have access to sign labels, i.e., $s_i = \text{sign}\left[p(y=+1|\boldsymbol{x}_i) - 0.5\right] \quad \forall i \in \{1, 2, \ldots, n\}$, we can divide $\{u_i\}_{i=1}^n$ into $\{u_i^+\}_{i=1}^{n_s^+}$ and $\{u_i^-\}_{i=1}^{n_s^-}$ depending on the sign label, with $n = n_s^+ + n_s^-$. Then we can construct the following estimator:

$$\hat{\beta}_{\text{Noise}} = \frac{1}{n}\left(\sum_{i=1}^{n_s^+}\left(1-u_i^+\right) + \sum_{i=1}^{n_s^-} u_i^-\right). \tag{6}$$

With this modified estimator $\hat{\beta}_{\text{Noise}}$, we can show the following propositions.

Proposition 3.4. *The estimator $\hat{\beta}_{\text{Noise}}$ is an unbiased estimator of the Bayes error.*

Proof is shown in App. B.5. The proof itself is simple, but the main idea lies in where we modified the estimator from Eq. 4 to avoid dealing with the noise inside the $\min$ operator, by utilizing the additional information of s_i.

Proposition 3.5. *For any $\delta > 0$, with probability at least $1-\delta$, we have, $|\hat{\beta}_{\text{Noise}} - \beta| \leq \sqrt{\frac{1}{2n}\log\frac{2}{\delta}}$.*

Proof is shown in App. B.6. This implies that $\hat{\beta}_{\text{Noise}}$ is a consistent estimator to the Bayes error, i.e., for $\forall\epsilon > 0$, $\lim_{n\to\infty} P(|\hat{\beta}_{\text{Noise}} - \beta| \geq \epsilon) = 0$. The rate of convergence is $\mathcal{O}_P(1/\sqrt{n})$.

In practice, one idea is to ask many labellers for each instance and then use the histogram of the hard labels as a noisy soft label. This is the main idea which we use later on in Sec. **??** with CIFAR-10H (Peterson et al., 2019). For simplicity, suppose that we have m labels for every instance i, and u_i is the sample average of those labels: $u_i = \frac{1}{m}\sum_{j=1}^m 1[y_i = 1]$. This satisfies our assumptions for noisy soft labels since $0 \leq u_i \leq 1$ and $\mathbb{E}[\xi_i \mid c_i] = 0$ for any $m \in \mathbb{N}$. If in practice we do not have access to sign labels, we cannot use the unbiased estimator $\hat{\beta}_{\text{Noise}}$ and we have no choice but to use the biased estimator $\tilde{\beta}$. However, in the following proposition, we show the bias can be bounded with respect to m.

Proposition 3.6. *The estimator $\tilde{\beta}$ is asymptotically unbiased, i.e., $\beta - \mathbb{E}[\widetilde{\beta}] \leq \frac{1}{2\sqrt{m}} + \sqrt{\log(2n\sqrt{m})/(2m)}$, where m is the number of labellers for each of the n instances.*

Proof is shown in App. B.4. The rate of convergence is $\mathcal{O}(\sqrt{\log(2nm)/m})$.

Hard labels Surprisingly, we can regard the usual hard labels as a special case of noisy soft labels. This enables us to estimate the Bayes error with hard labels and sign labels. To see why this is the case, we consider a specific type of noise:

$$\xi_i = \begin{cases} 1 - p(y=+1 \mid \boldsymbol{x}_i) & \text{w.p.} \quad p(y=+1 \mid \boldsymbol{x}_i) \\ -p(y=+1 \mid \boldsymbol{x}_i) & \text{w.p.} \quad 1 - p(y=+1 \mid \boldsymbol{x}_i). \end{cases} \tag{7}$$

Since $u_i = c_i + \xi_i$, this implies $u_i = 1$ w.p. $p(y=+1|\boldsymbol{x}_i)$ and $u_i = 0$ w.p. $p(y=-1|\boldsymbol{x}_i)$. We can easily confirm that our assumptions for noisy soft labels are satisfied: $0 \leq \xi_i \leq 1$ and $\mathbb{E}[\xi_i \mid \boldsymbol{x}_i] = 0$.

If we assign $y = +1$ for $u_i = 1$ and $y = -1$ for $u_i = 0$, this will be equivalent to providing a hard label based on a Bernoulli distribution $\mathrm{Be}(p(y = +1|\boldsymbol{x}_i))$ for an instance $\boldsymbol{x}_i$ sampled from $p(\boldsymbol{x})$. This corresponds to the usual data generating process in supervised learning.

4 ESTIMATION WITH POSITIVE-CONFIDENCE DATA

In this section, we move our focus to the problem of estimating the Bayes error with positive-confidence (Pconf) data (Sec. 2). First, we show the following theorem.

Theorem 4.1. *The Bayes error can be expressed with positive-confidence data and class prior:*

$$\beta_{\mathrm{Pconf}} := \pi_+ \left(1 - \mathbb{E}_+ \left[\max\left(0, 2 - \frac{1}{r(\boldsymbol{x})}\right)\right]\right) = \beta. \tag{8}$$

Recall that $\mathbb{E}_+$ is $\mathbb{E}_{p(\boldsymbol{x}|y=+1)}$ and $r(\boldsymbol{x}) = p(y = +1|\boldsymbol{x})$. The proof is shown in App. B.7. Based on Theorem 4.1, we can derive the following estimator:

$$\hat{\beta}_{\mathrm{Pconf}} = \pi_+ \left(1 - \frac{1}{n_+} \sum_{i=1}^{n_+} \max\left(0, 2 - \frac{1}{r_i}\right)\right), \tag{9}$$

where $r_i = p(y = +1|\boldsymbol{x}_i)$ for $\boldsymbol{x}_i \sim p(\boldsymbol{x}|y = +1)$, and n_+ is the number of Pconf data. We can estimate the Bayes error in an instance-free fashion, even if we do not have any r_i from the negative distribution. All we require is instance-free positive data and π_+. A counter-intuitive point of Eq. 9 is that we can throw away r_i when $r_i < 0.5$, due to the $\max$ operation. If the labeler is confident that $r_i < 0.5$, we do not need the specific value of r_i. Note that in practice, $r_i = 0$ may occur for positive samples. This will cause an issue for classifier training (since Eq. 2 will diverge), but such samples can be ignored in Bayes error estimation (due to the $\max$ in Eq. 9).

As discussed in Sec. 2, π_+ was unnecessary when learning a Pconf classifier. Fortunately, this is available from other sources depending on the task, e.g., market share, government statistics, or other domain knowledge (Quadrianto et al., 2009; Patrini et al., 2014). Instead of the Bayes error itself, we may be interested in the following: 1) the relative comparison between the Bayes error and the classifier's error, 2) whether the classifier's error has exceeded the Bayes error or not. In these cases, π_+ is unnecessary since we can divide both errors (Eq. 9 and empirical version of Eq. 2) by π_+ without changing the order of the two.

We can derive the following two propositions.

Proposition 4.2. *The estimator $\hat{\beta}_{\mathrm{Pconf}}$ is an unbiased estimator of the Bayes error.*

Proposition 4.3. *For any $\delta > 0$, with probability at least $1-\delta$, we have, $|\hat{\beta}_{\mathrm{Pconf}} - \beta| \leq \sqrt{\frac{\pi_+^2}{2n_+} \log \frac{2}{\delta}}$.*

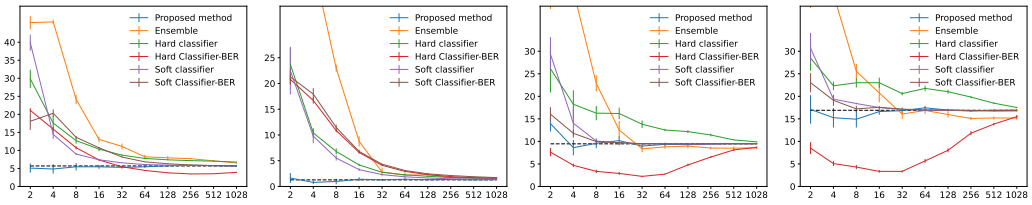
Proofs are shown in App. B.8 and App. B.9, respectively. This implies that the estimator $\hat{\beta}_{\mathrm{Pconf}}$ is consistent to the Bayes error, i.e., for $\forall \epsilon > 0$, $\lim_{n\to\infty} P(|\hat{\beta}_{\mathrm{Pconf}} - \beta| \geq \epsilon) = 0$. The rate of convergence is $\mathcal{O}_P(1/\sqrt{n_+})$. We can also see that the upper bound is equivalent to the upper bound in Prop. 3.2 when the data is balanced, i.e., $\pi_+ = 0.5$.

5 EXPERIMENTS

In this section, we perform experiments with synthetic data to investigate the behavior of our methods. Then, we use benchmark datasets to compare the estimated Bayes error with SOTA classifiers.

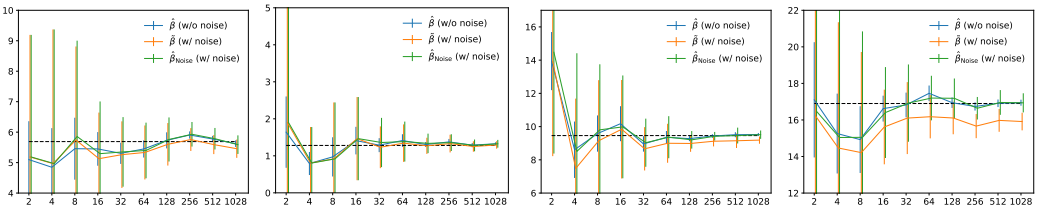
5.1 ILLUSTRATION

Positive-negative data **Setup:** We use two Gaussian distributions to construct a binary classification dataset. We then derive the positive-class posterior analytically for each data and use it as our label. We use Eq. 4 to estimate the Bayes error. We change the number of training samples per class $(2, 4, 8, \ldots, 1028)$ to observe how quickly we can approach the true Bayes error. We perform experiments with four different setups (A, B, C, and D) and the details are explained in the supplementary material. To compare with the true Bayes error of the setup, we take the



(a) Setup A (b) Setup B (c) Setup C (d) Setup D

Figure 1: Comparison of the proposed method (blue) with the true Bayes error (black), a recent Bayes error estimator "Ensemble" (orange) (Noshad et al., 2019), and the test error of a multilayer perceptron (hard labels: green, soft labels: purple). We also use the classifier's probabilistic output and plug it into the Bayes error equation in Eq. 3, shown as "Classifier-BER" (hard labels: red, soft labels: brown). The vertical/horizontal axis is the test error or estimated Bayes error/# of training samples per class, respectively. Instance-free data was used for Bayes error estimation, while fully labelled data was used for training classifiers. We plot the mean and standard error for 10 trials.



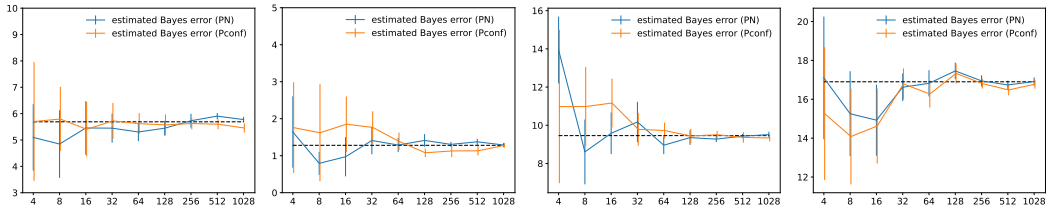
(a) Setup A (b) Setup B (c) Setup C (d) Setup D

Figure 2: Comparison of the estimated Bayes error with and without noise on the labels, with mean and standard error for 10 trials. The vertical/horizontal axis is the estimated Bayes error/the number of training samples, respectively. With noisy labels, the estimation tends to become inaccurate with a larger standard error. As we increase sample size, the difference between clean labels becomes smaller, but a small bias remains. With our modified estimator, the bias vanishes.

average of $c(\boldsymbol{x})$ for $10,000$ samples. We expect this to be a good substitute because we are using the analytically derived class posteriors for the soft labels. Although our problem setting does not require instances, we also compare with a recent Bayes error estimator that uses instances and hard labels (Noshad et al., 2019) and classifiers trained from instances and hard labels. We train a multi-layer perceptron model with one hidden layer (500 units) with the following cross-entropy loss: $-\mathbf{1}[y_i = +1] \log \hat{p}_\theta(y = +1|\boldsymbol{x}_i) - \mathbf{1}[y_i = -1] \log \hat{p}_\theta(y = -1|\boldsymbol{x}_i)$, where $\hat{p}_\theta(y|\boldsymbol{x})$ is the probabilistic output of the trained classifier and θ represents the model parameters. We chose the cross entropy loss because it is a proper scoring rule (Kull & Flach, 2015). We use stochastic gradient descent for 150 epochs with a learning rate of 0.005 and momentum of 0.9. We also compare with another baseline ("classifier-BER"), where we plug in $\hat{p}_\theta(y|\boldsymbol{x})$ to Eq. 3. Finally, we add a stronger baseline "soft classifier" which utilizes soft labels. If the soft label is r_i for sample $\boldsymbol{x}_i$, then the cross-entropy loss is modified as follows: $-r_i \log \hat{p}_\theta(y = +1|\boldsymbol{x}_i) - (1 - r_i) \log \hat{p}_\theta(y = -1|\boldsymbol{x}_i)$.

Results: We construct a dataset with the same underlying distribution ten times and report the means and standard error for the final epoch in Fig. 1. We can see how the estimated Bayes error is relatively accurate even when we have few samples. The benefit against classifier training is clear: the MLP model requires much more samples before the test error approaches the Bayes error. The classifier is slightly better when it can use soft labels but is not as good as our proposed method. It is interesting how the "hard classifier-BER" tends to underestimate the Bayes error (except Fig. 1b). This can be explained by the overconfidence of neural networks (Guo et al., 2017), which will lead to a small $\min_{y \in \{\pm 1\}} \hat{p}(y|\boldsymbol{x})$. This underestimation does not occur with "soft classifier-BER," implying that there are less room to be overconfident when we learn with the modified cross-entropy loss.

Positive-negative data with noise **Setup:** Next, we perform experiments when we have noise on the labels. We follow the assumption in Sec. 3, and we design the noise to have mean 0 and



(a) Setup A (b) Setup B (c) Setup C (d) Setup D

Figure 3: Comparison of the estimated Bayes error with two different data: PN (positive and negative) and Pconf (positive-confidence). The vertical axis is the estimated Bayes error, and the horizontal axis is the number of training samples. We plot the mean and standard error for 10 trials. We used instance-free positive and negative data for the PN method, while we used only instance-free positive data for the Pconf method. We can observe how the difference between the two becomes smaller as we increase the number of training samples.

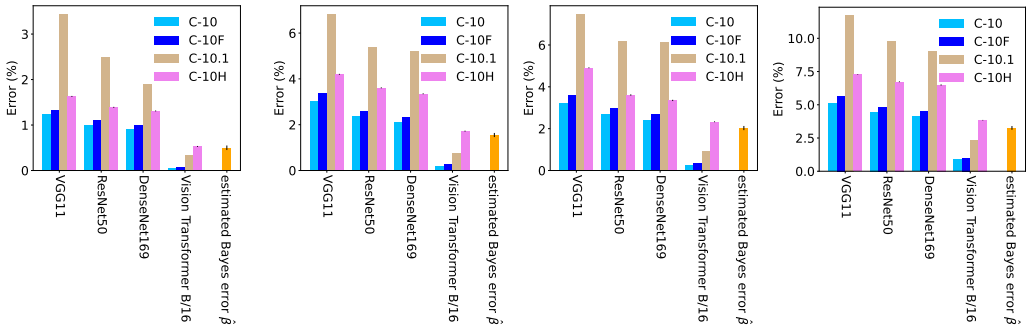
the label to be between 0 and 1 after adding this noise, to be a valid probability. For this purpose, we add noise to the underlying class posterior and use a truncated Gaussian distribution $p(z)$ with standard deviation 0.4. We truncate the left tail $\{z \in \mathbb{R} | z \leq \mu - \min(1-\mu, \mu - 0)\}$ and the right tail $\{z \in \mathbb{R} | z \geq \mu + \min(1-\mu, \mu - 0)\}$ where $\mu \in [0, 1]$ is the mean of the distribution, i.e., the true class posterior. We compare three estimators: 1) Original estimator $\hat{\beta}$ in Eq. 4 with clean labels, 2) original estimator $\tilde{\beta}$ in Eq. 4 with noisy labels, and 3) proposed estimator $\hat{\beta}_{\mathrm{Noise}}$ in Eq. 6 with noisy labels. We repeat the experiment 10 times with different data realizations. **Results:** The mean and standard error for 10 trials are shown in Fig. 2, for varying number of training samples. As expected, using noisy soft labels lead to worse results for $\tilde{\beta}$, as we can see with the larger standard errors. Increasing the sample size is effective to some degree, but the bias does not vanish. However, if we use a modified estimator $\hat{\beta}_{\mathrm{Noise}}$, the bias vanishes completely and works well even *with* noise.

Positive-confidence data **Setup:** Similarly, we perform experiments for positive-confidence (Pconf) data. The data is generated so that the class prior π_+ is 0.5, and we assume this information is available. We only use the confidence of data from the positive distribution and do not use the confidence of data from the negative distribution. We use this data to estimate $\hat{\beta}_{\mathrm{Pconf}}$ and compare it with the $\hat{\beta}$ derived earlier. **Results:** We report the mean and standard error for 10 trials in Fig. 3. Although Bayes error estimation tends to be inaccurate with Pconf data for fewer samples, the difference between the two disappears with more samples.

5.2 Experiments with benchmark datasets: CIFAR-10

Setup: We use CIFAR-10 (C10) (Krizhevsky, 2009), ciFAIR-10 (C10F) (Barz & Denzler, 2020), CIFAR-10.1 (C10.1) (Recht et al., 2018), and CIFAR-10H (C10H) (Peterson et al., 2019). C10 is an image dataset of 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. However, the test set in C10 is known to include complete duplicates and near-duplicates from the training dataset. C10F replaces the duplicates in the test set with a different image. C10.1 is a new test set to evaluate C10 classifiers. It ensures the underlying distribution to be as close to the original distribution as possible, but it does not include any images from the original test set. C10H is a dataset with additional human annotations for the C10 test set. Each image in the test set has class labels annotated by around 50 different human labelers. We can take the proportion to construct soft labels that we will use for Bayes error estimation. Although these datasets are originally multi-class setups with 10 classes, we re-construct binary datasets in the following four ways: animals vs. artifacts, land vs. other (water/sky), odd vs. even, and first five vs. last five. See the details after Sec. 6.

We compare the estimated Bayes error $\hat{\beta}$ with the base model of Vision Transformer (Dosovitskiy et al., 2021) with 16×16 input patch size (ViT B/16), DenseNet169 (Huang et al., 2017), ResNet50 (He et al., 2016), and VGG11 (Simonyan & Zisserman, 2015). For ViT B/16, we downloaded the official pretrained model (Dosovitskiy et al., 2021) and used the implementation by Jeon (Jeon, 2020) to fine-tune on C10. For others, we use the models provided in Phan (Phan, 2021), which are trained on C10. Although these are ten-class classifiers, we transform them into binary classifiers by



(a) Animals vs. artifacts (b) Land vs. other (c) Odd vs. even (d) First 5 vs. last 5

Figure 4: Comparing the estimated Bayes error with four classifiers and four setups (Figures 4a, 4b, 4c, 4d). Although it seems that Vision Transformer B/16 has surpassed the Bayes error for all four setups for CIFAR-10 (C10), ciFAIR-10 (C10F), and CIFAR-10.1 (C10.1), while it is still comparable or larger than the Bayes error for CIFAR-10H (C10H). In Sec. 5.2, we discuss how this has to be investigated carefully. The black error bar for the Bayes error is the 95% confidence interval. The black error bar for the C10H is the standard error for sampling test labels 20 times.

swapping the predictions into positive and negative based on the setups (App. **??**). For example, a prediction of "cat" would be swapped with a positive label in "animals vs. artifacts".

Results: We compare the estimated Bayes error $\hat{\beta}$ with classifier training in Fig. 4. As expected, the test error (shown in light blue) is lower for newer network architectures, with ViT achieving the lowest test error. An interesting observation is that the test error of ViT has exceeded the estimated Bayes error in all four setups, which theoretically should not take place. We first suspected that ViT's error is too low (overfitting to the test set; Sec. A) but found out this is not necessarily the case. We investigated the degree of overfitting to the test set by performing experiments with C10F and C10.1. The former dataset excludes the duplicated images between training and test set, and the latter dataset is a fresh test set that is designed to be close to the original distribution of C10. The test error for the C10F dataset becomes slightly worse and much worse for C10.1. This is what we expected and implies that overfitting to the test set may be happening to some degree. However, ViT's error is *still* below the estimated Bayes error even for C10.1.

To explain why ViT's error is still below the estimated Bayes error, we discuss the discrepancy between the underlying $p(y|\boldsymbol{x})$ for C10 and C10H. C10 is constructed with a three-step procedure: first, collect images by searching for specific terms in search engines, downscale them to 32×32 (Torralba et al., 2008), and finally ask labelers to filter out mislabelled images (Krizhevsky, 2009). On the other hand, C10H constructs the labels by asking labelers to choose the correct class out of ten classes after seeing each *downscaled* image of C10. This makes it visually harder for human labellers. Hence, the labelling process will become less confident, and this may lead to a higher entropy of the distribution of the human labels. According to Eq. 4, this will lead to an overestimation of $\hat{\beta}$.

To adjust for the distribution discrepancy, we also plot the test errors when we sample a test label for each sample from the C10H dataset (repeated 20 times) and show this in Fig. 4 with pink. Note that this generally will not go below the estimated Bayes error. For example, consider an extreme case with zero test error for the original test set. If a test sample has $p(y = +1|\boldsymbol{x}) = 0.8$ in C10H, then it will incur an error of 0.2 on average for the new test labels. After adjusting for the discrepancy, we can see how the test error of ViT is on par with or slightly higher than the estimated Bayes error. This implies that we have reached (or is about to reach) the Bayes error for C10H.

5.3 Experiments with benchmark datasets: Fashion-MNIST

Although there are many research areas within machine learning that utilize soft labels or multiple hard labels,[3] to our knowledge there is only one large-scale dataset with over 10 annotations per

[3]Some examples are soft-label classification (Nguyen et al., 2014), Pconf and SC-conf classification (Ishida et al., 2018; Cao et al., 2021), detecting superhuman classifiers (Xu et al., 2021), probability estimation (Liu et al., 2021), and our Bayes error estimation.

sample (Peterson et al., 2019), which we used in Sec. 5.2. Hence, it would be beneficial to construct another benchmark dataset with multiple annotations. This will also demonstrate how our Bayes error estimator can be used for more difficult benchmark datasets compared with CIFAR-10 in Sec. 5.2.

Data collection: We chose the Fashion-MNIST dataset (Xiao et al., 2017) as a more challenging dataset. This is a 10 class classification dataset with small fashion images with categories such as dress, bag, and sneaker. Other popular datasets such as CIFAR-100 (Krizhevsky, 2009) and ImageNet (Deng et al., 2009) are problematic because they are known to be essentially multi-label classification datasets rather than single-label (Shankar et al., 2020; Beyer et al., 2020; Yun et al., 2021; Wei et al., 2022). Since the Bayes error is defined for the single-labeled case, this means we have to define a similar but different and novel concept that corresponds to the lowest error for multi-label classification if we want to estimate the best possible performance on datasets such as CIFAR-100 and ImageNet. This is definitely an interesting direction to pursue in the future, but is out of the scope of the current paper since we are focusing on the Bayes error.

We used Amazon Mechanical Turk (Buhrmester et al., 2016) to collect multiple annotations per each image in the test dataset with 10,000 images. We designed the task so that the worker is instructed to annotate 10 randomly chosen fashion images, with a multiple choice of 10 labels. We showed several correct images for each of categories in the beginning of the task, so that the worker can have a good understanding of each labels. For each task with 10 image annotations, we paid \$0.06 or \$0.08 to the worker. We removed the duplicates when the same worker provide a label for the same image more than once. We further removed the annotation when the worker provided an answer within 30 seconds, which we considered as too short, e.g., when we performed the task, it took us 45 seconds when we skipped reading the instructions. After removing the shorter ones, the mean and median answering time was about 147 and 78 seconds, respectively. Finally, we ended up with around 67 annotations per image. The least number was 53 and the largest number was 79 annotations.

Setup and results: Since aiming for SOTA performance on Fashion-MNIST is less common in our community compared with CIFAR-10, we simply chose ResNet-18 (He et al., 2016) for our model architecture. We trained for 100 epochs with batch size 128, learning rate 0.001 on the original labels (no data augmentation) and reconstructed the 10-class predictions into binary predictions by grouping shirt, dress, coat, pullover, and t-shirt/top as the positive class, and the other 5 categories as the negative class.

We compared the binary predictions with a test label sampled from the multiple annotations (repeated 10 times) and derived the error, which is basically the same procedure of the pink bar in Fig. 4. The error was 3.852% ($\pm$ 0.041%), where percentage in parenthesis is the standard error. On the other hand, the estimated Bayes error was 3.478% ($\pm$ 0.079%), where percentage in parenthesis is the 95% confidence interval. We can see how the estimated Bayes error is higher than the results for CIFAR-10H in Fig. 4, implying that Fashion-MNIST is more difficult than CIFAR-10. This is intuitive as the 28×28 grayscale images are ambiguous while the original 51×73 PNG image is much more clear, e.g., see Fig. 1 in Xiao et al. (2017), and also consistent with the results in Theisen et al. (2021). The ResNet-18 error is approaching the estimated Bayes error, but may still have room for small improvement. Since we made a simple choice of model architecture, we may be able to close this gap by other architectures that are more suited for Fashion-MNIST images, adding data augmentation, and trying extensive hyper-parameter tuning.

6 Conclusion

We proposed a model-free and instance-free Bayes error estimation method that is unbiased and consistent. We showed how we can similarly estimate it with uncertainty labels, noisy soft labels, or positive-confidence. In experiments, we demonstrated how the proposed method could be helpful when we want to compare the SOTA performance for benchmark datasets with the Bayes error.

Although we did not detect strong evidence of test set overfitting in Sec. 5.2 and Sec. 5.3, we are interested in investigating other benchmark datasets in the future. A current limitation is that we focus on binary classification. Since binary classification is popular in industry, e.g., click prediction, fraud detection, anomaly detection, churn prediction, we believe estimating the Bayes error for binary classification problems is interesting on its own. However, in the future we are interested in working on the multi-class case (and the multi-label case we discussed in Sec. 5.3).

Ethics statement: Constructing soft labels may rely on collecting human annotations. This may potentially have a negative impact, e.g., cost of the annotations. We are hoping to lower the cost of annotations in our future work. We will only publish our annotations we collected in Sec. 5.3 in a way that does not contain any private information, e.g., only the category counts for each images and drop all other worker-related information.

Reproducibility statement: In the synthetic experiments, we used four setups: A, B, C, D. A is a 10-dimensional isotropic setup with means $\boldsymbol{\mu}_+ = [0, \ldots, 0]^\top$ and $\boldsymbol{\mu}_- = [1, \ldots, 1]^\top$ with identity matrices for the covariance matrices. B is the same as A except that the dimension is doubled to 20. C and D are 2-dimensional anisotropic setups and is borrowed from the first two synthetic datasets in Ishida et al. (Ishida et al., 2018), where the code is provided in `https://github.com/takashiishida/pconf`.

We used data and code provided in `https://cvjena.github.io/cifair/` for ciFAIR-10 (Barz & Denzler, 2020). We used data and code provided in `https://github.com/jcpeterson/cifar-10h` for CIFAR-10H (Peterson et al., 2019). We used data and code provided in `https://github.com/modestyachts/CIFAR-10.1` for CIFAR-10.1 (Recht et al., 2019). We used data provided in `https://www.cs.toronto.edu/~kriz/cifar.html` for CIFAR-10. We are not aware of any personally identifiable information or offensive content in the datasets we used. We also collected annotations by using Amazon Mechanical Turk (Buhrmester et al., 2016). We explained the details of how we collected the annotations in Sec. 5.3, and we plan to remove all personally identifiable information when we publish the data after acceptance of the paper.

We used `https://github.com/mrtnoshad/Bayes_Error_Estimator` for the ensemble method (Noshad et al., 2019). We partially used `https://github.com/takashiishida/flooding` for training classifiers in synthetic experiments with Gaussian distributions. We used `https://github.com/huyvnphan/PyTorch_CIFAR10` for VGG11, ResNet50, and DenseNet169 models trained on CIFAR-10. We used `https://github.com/jeonsworld/ViT-pytorch` to fine-tune ViT and downloaded pretrained weights from `https://console.cloud.google.com/storage/browser/vit_models;tab=objects?pli=1&prefix=&forceOnObjectsSortingFiltering=false`. All of our experiments were conducted with a single Mac Pro except when we trained ResNet-18 in Sec. 5.3 and fine-tuned ViT in Sec. 5.2, which was when we utilized Google Colab Pro+.

The details of the four binary datasets that are used in Sec. 5.2 is as follows.

- Animals vs. artifacts: we divide the dataset into cat, deer, dog, frog, and horse for the positive class and plane, car, ship, truck, and bird for the negative class.
- Land vs. other (water/sky): we divide the dataset into car, truck, cat, deer, dog, horse for the positive class and plane, ship, bird, and frog for the negative class.
- Odd vs. even: we divide the dataset into odd and even class numbers, with plane, bird, deer, frog, and ship for the positive class and car, cat, dog, horse, and truck for the negative class.
- First five vs. last five: we divide the dataset into the first five and last five classes based on the class number, with plane, car, bird, cat, and deer for the positive class and dog, frog, horse, ship, and truck for the negative class.

REFERENCES

Han Bao, Gang Niu, and Masashi Sugiyama. Classification from pairwise similarity and unlabeled data. In *ICML*, 2018.

Björn Barz and Joachim Denzler. Do we train on test data? Purging CIFAR of near-duplicates. *Journal of Imaging*, 6:41, 2020.

Visar Berisha, Alan Wisler, Alfred O. Hero, and Andreas Spanias. Empirically estimable classification bounds based on a nonparametric divergence measure. *IEEE Transactions on Signal Processing*, 64:580 – 591, 2016.

Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality data? 2016.

Yuzhou Cao, Lei Feng, Senlin Shu, Yitian Xu, Bo An, Gang Niu, and Masashi Sugiyama. Multi-class classification from single-class data with confidences. In *arXiv:2106.08864*, 2021.

O. Chapelle, B. Schölkopf, and A. Zien (eds.). *Semi-Supervised Learning*. MIT Press, Cambridge, Massachusetts, USA, 2006.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

Pierre A. Devijver. A multiclass, k-NN approach to Bayes risk estimation. *Pattern Recognition Letters*, 3:1 – 6, 1985.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

M. C. du Plessis, G. Niu, and M. Sugiyama. Analysis of learning from positive and unlabeled data. In *NeurIPS*, 2014.

Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edition, 1990.

Keinosuke Fukunaga and Larry D. Hostetler. k-nearest-neighbor Bayes-risk estimation. *IEEE Transactions on Information Theory*, 21:285 – 293, 1975.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative modeling. In *arXiv:2010.14701v2*, 2020.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

Takashi Ishida, Gang Niu, and Masashi Sugiyama. Binary classification from positive-confidence data. In *NeurIPS*, 2018.

Eunkwang Jeon. Vision transformer. `https://github.com/jeonsworld/ViT-pytorch`, 2020. Accessed: 2021-10-01.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. In *arXiv:2001.08361*, 2020.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *NAACL*, 2021.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1×1 convolutions. In *NeurIPS*, 2018.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Meelis Kull and Peter Flach. Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 68–85. Springer, 2015.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. Question and answer test-train overlap in open-domain question answering datasets. In *EACL*, 2020.

Gang Li, Jan Zrimec, Boyang Ji, Jun Geng, Johan Larsbrink, Aleksej Zelezniak, Jens Nielsen, and Martin KM Engqvist. Performance of regression models as a function of experiment noise. *Bioinformatics and Biology Insights*, 15:1–10, 2021.

Sheng Liu, Aakash Kaku, Weicheng Zhu, Matan Leibovich, Sreyas Mohan, Boyang Yu, Laure Zanna, Narges Razavian, and Carlos Fernandez-Granda. Deep probability estimation. In *ICML*, 2021.

Nan Lu, Gang Niu, Aditya Krishna Menon, and Masashi Sugiyama. On the minimal supervision for training any binary classifier from only unlabeled data. In *ICLR*, 2019.

Horia Mania, John Miller, Ludwig Schmidt, Moritz Hardt, and Benjamin Recht. Model similarity mitigates test set overuse. In *NeurIPS*, 2019.

Shahar Mendelson. Lower bounds for the empirical minimization algorithm. *IEEE Transactions on Information Theory*, 54(8):3797–3803, 2008.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2nd edition, 2018.

Quang Nguyen, Hamed Valizadegan, and Milos Hauskrecht. Learning classification models with soft-label information. *Journal of the American Medical Informatics Association*, 21(3):501–508, 2014.

Morteza Noshad, Li Xu, and Alfred Hero. Learning to benchmark: Determining best achievable misclassification error from training data. In *arXiv:1909.07192*, 2019.

Giorgio Patrini, Richard Nock, Paul Rivera, and Tiberio Caetano. (almost) no label no cry. In *NeurIPS*, 2014.

Joshua C. Peterson, Ruairidh M. Battleday, Thomas L. Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *ICCV*, 2019.

Huy Phan. Pytorch_cifar10. `https://github.com/huyvnphan/PyTorch_CIFAR10`, 2021. Accessed: 2021-10-01.

Novi Quadrianto, Alex J Smola, Tiberio S Caetano, and Quoc V Le. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10(10), 2009.

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

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *ICML*, 2019.

Cedric Renggli, Luka Rimanic, Nora Hollenstein, and Ce Zhang. Evaluating Bayes error estimators on real-world datasets with FeeBee. In *NeurIPS Datasets and Benchmarks track*, 2021.

Rebecca Roelofs, Vaishaal Shankar, Benjamin Recht, Sara Fridovich-Keil, Moritz Hardt, John Miller, and Ludwig Schmidt. A meta-analysis of overfitting in machine learning. In *NeurIPS*, 2019.

Tomoya Sakai, Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Semi-supervised classification based on classification from positive and unlabeled data. In *ICML*, 2017.

Keiko Sasada, Noriko Yamamoto, Hiroki Masuda, Yoko Tanaka, Ayako Ishihara, Yasushi Takamatsu, Yutaka Yatomi, Waichiro Katsuda, Issei Sato, Hirotaka Matsui, and on behalf of the Kyushu regional department of the Japanese Society of Laboratory Hematology. Inter-observer variance and the need for standardization in the morphological classification of myelodysplastic syndrome. *Leukemia Research*, 69:54–59, 2018.

Salimeh Yasaei Sekeh, Brandon Oselio, and Alfred O. Hero. Learning to bound the multi-class Bayes error. *IEEE Transactions on Signal Processing*, 68:3793 – 3807, 2020.

Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. Evaluating machine accuracy on imagenet. In *ICML*, 2020.

Kazuhiko Shinoda, Hirotaka Kaji, and Masashi Sugiyama. Binary classification from positive data with skewed confidence. In *IJCAI*, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Ryan Theisen, Huan Wang, Lav R. Varshney, Caiming Xiong, and Richard Socher. Evaluating state-of-the-art classification models against Bayes optimality. In *NeurIPS*, 2021.

Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1958–1970, 2008.

Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer, New York, USA, 2nd edition, 2000.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *ICML*, 2013.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *ICLR*, 2022.

Roman Werpachowski, András György, and Csaba Szepesvári. Detecting overfitting via adversarial examples. In *NeurIPS*, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Qiongkai Xu, Christian Walder, and Chenchen Xu. Humanly certifying superhuman classifiers. *arXiv preprint arXiv:2109.07867*, 2021.

Chhavi Yadav and Léon Bottou. Cold case: The lost MNIST digits. In *NeurIPS*, 2019.

Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, Junsuk Choe, and Sanghyuk Chun. Re-labeling imagenet: from single to multi-labels, from global to localized labels. In *CVPR*, 2021.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1): 44–53, 2018.

A Related work

Bayes error estimation has been a topic of interest for nearly half a century (Fukunaga & Hostetler, 1975). Several works have derived upper and lower bounds of the Bayes error and proposed ways to estimate those bounds. Various f-divergences, such as the Bhattacharyya distance (Fukunaga, 1990) or the Henze-Penrose divergence (Berisha et al., 2016; Sekeh et al., 2020), were studied. Other approaches include Noshad et al. (Noshad et al., 2019), which directly estimated the Bayes error with f-divergence representation instead of using a bound, and Theisen et al. (Theisen et al., 2021),

which computed the Bayes error of generative models learned using normalizing flows (Kingma & Dhariwal, 2018). More recently, Renggli et al. (Renggli et al., 2021) proposed a method to evaluate Bayes error estimators on real-world datasets, for which we usually do not have access to the true Bayes error.

All of these works that estimate the Bayes error are based on instance-label pairs. Not only do they require instances, but they also usually have a model, which may need careful tuning of hyperparameters to avoid overfitting. Li et al. (Li et al., 2021) proposed to estimate the maximal performance for regression problems, while we focus on the classification problem.

Weakly supervised learning (Zhou, 2018) tries to learn with a weaker type of supervision, e.g., semi-supervised learning (Chapelle et al., 2006; Sakai et al., 2017), positive-unlabeled learning (du Plessis et al., 2014), similar-unlabeled learning (Bao et al., 2018), and unlabeled-unlabeled learning (Lu et al., 2019).

On the other hand, the focus of most previous works on estimating the Bayes error is on ordinary classification problems with full supervision. To our knowledge, Bayes error estimation for weakly supervised data has not been studied. We discussed how our formulation introduced in this paper is flexible and can be extended to weakly supervised learning in Sec. 4.

Overfitting to the test dataset is one of the issues that concerns the machine learning community (Mania et al., 2019; Werpachowski et al., 2019; Kiela et al., 2021). Models can overfit to the test dataset by training on the test set, *explicitly* or *implicitly*. Barz et al. (Barz & Denzler, 2020) reported that 3% of the test data in CIFAR-10 has duplicates and near-duplicates in the training set. Lewis et al. (Lewis et al., 2020) studied popular open-domain question answering datasets and reported that 30% of test questions have a near-duplicate in the training data. These are examples of when we may explicitly train on the test set. Barz et al. (Barz & Denzler, 2020) and Lewis et al. (Lewis et al., 2020) showed how removing the duplicates will lead to substantially worse results.

Overfitting to the test set can also happen implicitly because research is conducted on the same test data that is public for many years (often with a fixed train/test partition), in contrast to how engineers in industry can sample fresh unseen data for evaluation. Recht et al. (Recht et al., 2018) constructed CIFAR-10.1, an alternative unseen test set for CIFAR-10 classifiers and showed how deep learning models became worse with respect to the test error, with a 4% to 10% increase. Yadav et al. (Yadav & Bottou, 2019) and Recht et al. (Recht et al., 2019) showed similar results for MNIST (LeCun et al., 1998) and ImageNet (Deng et al., 2009), respectively. With closer inspection, however, Recht et al. (Recht et al., 2019) discussed how the accuracy improvement proportional to that in the original test set was observed in the new test set, for both CIFAR-10 and ImageNet. Roelofs et al. (Roelofs et al., 2019) studied Kaggle competition datasets by comparing accuracies based on a public test set (which is used repeatedly by practitioners) and a private test set (which is used to determine the final rankings), but found little evidence of overfitting.

Constructing new test datasets is one way to examine the degree of overfitting to the test dataset. In Sec. 5, we discussed how estimating the Bayes error can be another way to detect overfitting to the test dataset. If models have lower test error than the estimated Bayes error, that is a strong signal for this type of overfitting. We compared the test error of recent classifiers with the estimated Bayes error in Sec. 5.

B PROOFS

B.1 PROOF OF PROPOSITION 3.1

Proof.

$$\mathbb{E}_{\{\boldsymbol{x}_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x})}[\hat{\beta}] = \mathbb{E}_{\{\boldsymbol{x}_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x})}\left[\frac{1}{n}\sum_{i=1}^n [\min\{p(y=+1|\boldsymbol{x}_i), p(y=-1|\boldsymbol{x}_i)\}]\right] \tag{10}$$

$$= \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\boldsymbol{x}_i \sim p(\boldsymbol{x})}[\min\{p(y=+1|\boldsymbol{x}_i), p(y=-1|\boldsymbol{x}_i)\}] \tag{11}$$

$$= \frac{1}{n}\sum_{i=1}^n \beta \tag{12}$$

$$= \beta. \tag{13}$$

□

B.2 PROOF OF PROPOSITION 3.2

Proof. Since $0 \leq \min[p(y=+1|\boldsymbol{x}), p(y=-1|\boldsymbol{x})] \leq 0.5$, we can use the Hoeffding's inequality as follows:

$$P(|\hat{\beta} - \beta| \geq \epsilon) \leq 2\exp\left(-\frac{2n\epsilon^2}{\left(\frac{1}{2}\right)^2}\right) = 2\exp(-8n\epsilon^2). \tag{14}$$

Therefore, for any $\delta > 0$, with probability at least $1-\delta$, we have,

$$|\hat{\beta} - \beta| \leq \sqrt{\frac{1}{8n}\log\frac{2}{\delta}}. \tag{15}$$

□

B.3 PROOF THAT $\tilde{\beta}$ IS A BIASED ESTIMATOR

Proof.

$$\mathbb{E}_{\{c_i,\xi_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(c,\xi)}[\tilde{\beta}] = \mathbb{E}_{\{c_i,\xi_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(c,\xi)}\left[\frac{1}{n}\sum_{i=1}^n [\min\{u_i, 1-u_i\}]\right] \tag{16}$$

$$= \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{(c_i,\xi_i)\sim p(c,\xi)}[\min\{u_i, 1-u_i\}] \tag{17}$$

$$\neq \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{c_i \sim p(c)}\left[\min\{\mathbb{E}_{\xi_i \sim p(\xi|c_i)}[u_i], 1-\mathbb{E}_{\xi_i \sim p(\xi|c_i)}[u_i]\}\right] \tag{18}$$

$$= \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{c_i \sim p(c)}[\min\{c_i, 1-c_i\}] \tag{19}$$

$$= \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\boldsymbol{x}_i \sim p(\boldsymbol{x})}[\min\{p(y=1|\boldsymbol{x}_i), p(y=-1|\boldsymbol{x}_i)\}] \tag{20}$$

$$= \frac{1}{n}\sum_{i=1}^n \beta = \beta \tag{21}$$

□

B.4 PROOF OF PROPOSITION 3.6

For simplicity, suppose that we have m labels for every sample i, and u_i is the sample average of those labels:

$$u_i = \frac{1}{m}\sum_{j=1}^{m} 1[y_i = 1]. \tag{22}$$

Since $1[y_i = 1] \in [0, 1]$, using Hoeffding's inequality, we get

$$\begin{aligned}
&\Pr\left[\bigcup_{i=1}^{n}\{|u_i - c_i| \geq \epsilon\} \mid c_i\right] && (23)\\
&\leq \sum_{i=1}^{n}\Pr[|u_i - c_i| \geq \epsilon \mid c_i] \quad \text{(union bound)} && (24)\\
&\leq 2n\exp\left(-\frac{2m\epsilon^2}{(1-0)^2}\right) && (25)\\
&= 2n\exp\left(-2m\epsilon^2\right). && (26)
\end{aligned}$$

for any $\epsilon > 0$. In other words, for any $\delta' > 0$, with probability at least $1 - \delta'$, we have

$$|u_i - c_i| < \sqrt{\frac{1}{2m}\log\frac{2n}{\delta'}} =: \epsilon'(\delta') \tag{27}$$

for all $i = 1, \ldots, n$. Note that

$$\begin{aligned}
\beta_i &:= \min\{c_i, 1 - c_i\} && (28)\\
&= \min\{\mathbb{E}[u_i \mid c_i], \mathbb{E}[1 - u_i \mid c_i]\} && (29)\\
&\geq \mathbb{E}[\min\{u_i, 1 - u_i\} \mid c_i] && (30)\\
&\geq \mathbb{E}[\tilde{\beta} \mid c_i]. && (31)
\end{aligned}$$

The conditional bias can be bounded as

$$\begin{aligned}
\beta_i - \mathbb{E}[\tilde{\beta} \mid c_i] &= \beta_i - \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\min\{u_i, 1 - u_i\} \mid c_i\right] && (32)\\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\min\{c_i, 1 - c_i\} - \min\{u_i, 1 - u_i\} \mid c_i\right] && (33)\\
&= \frac{1}{n}\sum_{i=1}^{n}\Pr[|u_i - c_i| \geq \epsilon'(\delta') \mid c_i]\mathbb{E}[\min\{c_i, 1 - c_i\} - \min\{u_i, 1 - u_i\} \mid c_i, |u_i - c_i| \geq \epsilon'(\delta')] && (34)\\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\Pr[|u_i - c_i| < \epsilon'(\delta') \mid c_i]\mathbb{E}[\min\{c_i, 1 - c_i\} - \min\{u_i, 1 - u_i\} \mid c_i, |u_i - c_i| < \epsilon'(\delta')] && (35)\\
&\leq \frac{1}{n}\sum_{i=1}^{n}\Pr[|u_i - c_i| \geq \epsilon'(\delta) \mid c_i](1/2 - 0) && (36)\\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\Pr[|u_i - c_i| < \epsilon'(\delta') \mid c_i](\min\{c_i, 1 - c_i\} - \min\{c_i - \epsilon'(\delta'), 1 - c_i - \epsilon'(\delta')\}) && (37)\\
&\leq \delta'/2 + \epsilon'(\delta'), && (38)
\end{aligned}$$

where the last line follows from Eq. 27. Eq. 37 holds because $\min\{c_i, 1 - c_i\} \in [0, 1/2]$, $\min\{u_i, 1 - u_i\} \in [0, 1/2]$, and $|u_i - c_i| < \epsilon'(\delta')$ implies that $u_i > c_i - \epsilon'(\delta')$ and $1 - u_i > 1 - c_i - \epsilon'(\delta')$.

Further taking the expectation on both sides, we get a bound on the bias:

$$\beta - \mathbb{E}[\tilde{\beta}] = \mathbb{E}[\beta_i - \mathbb{E}[\tilde{\beta} \mid c_i]] \tag{39}$$
$$\leq \delta'/2 + \epsilon'(\delta') \tag{40}$$
$$= \delta'/2 + \sqrt{\frac{1}{2m} \log \frac{2n}{\delta'}}. \tag{41}$$

Setting δ' to $m^{-1/2}$ yields

$$\beta - \mathbb{E}[\widetilde{\beta}] \leq \frac{1}{2\sqrt{m}} + \sqrt{\frac{1}{2m} \log(2n\sqrt{m})} = \mathcal{O}\left(\sqrt{\frac{\log(2nm)}{m}}\right). \tag{42}$$

B.5 Proof of Proposition 3.4

Proof. $\hat{\beta}_{\text{Noise}}$ can be expressed as

$$\hat{\beta}_{\text{Noise}} = \frac{1}{n} \sum_{i=1}^{n} \{\mathbf{1}[c_i \geq 0.5](1 - u_i) + \mathbf{1}[c_i < 0.5]u_i\}. \tag{43}$$

Recall that $u_i = c_i + \xi_i$. Hence, we have

$$\mathbb{E}[\hat{\beta}_{\text{Noise}}] = \frac{1}{n} \sum_{i=1}^{n} \left(\mathbb{E}_{\{c_i, \xi_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(r,\xi)} [\mathbf{1}[c_i \geq 0.5](1 - u_i)] + \mathbb{E}_{\{c_i, \xi_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(r,\xi)} [\mathbf{1}[c_i < 0.5]u_i] \right) \tag{44}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \Big(\mathbb{E}_{\{c_i, \xi_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(r,\xi)} [\mathbf{1}[c_i \geq 0.5](1 - c_i) - \mathbb{E}_{\{c_i, \xi_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(r,\xi)} [\mathbf{1}[c_i \geq 0.5]\xi_i]$$
$$+ \mathbb{E}_{\{c_i, \xi_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(r,\xi)} [\mathbf{1}[c_i < 0.5]c_i] + \mathbb{E}_{\{c_i, \xi_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(r,\xi)} [\mathbf{1}[c_i < 0.5]\xi_i] \Big). \tag{45}$$

The second term in Eq. 45 is,

$$\mathbb{E}_{\{c_i, \xi_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p(r,\xi)} [\mathbf{1}[c_i \geq 0.5]\xi_i] = \int \int \mathbf{1}[c_i \geq 0.5] \cdot \xi_i \cdot p(c_i, \xi_i) \mathrm{d}c_i \mathrm{d}\xi_i \tag{46}$$
$$= \int_{c_i \geq 0.5} \left(\int \xi_i p(\xi_i | c_i) \mathrm{d}\xi_i \right) p(c_i) \mathrm{d}c_i \tag{47}$$
$$= \int_{c_i \geq 0.5} \mathbb{E}[\xi_i | c_i] p(c_i) \mathrm{d}c_i \tag{48}$$
$$= 0. \tag{49}$$

Similarly, the fourth term in Eq. 45 becomes zero. Then,

$$\mathbb{E}[\hat{\beta}_{\text{Noise}}] = \frac{1}{n} \sum_{i=1}^{n} \left(\mathbb{E}_{c_i \sim p(c)} [\mathbf{1}[c_i \geq 0.5](1 - c_i) + \mathbb{E}_{c_i \sim p(c)} [\mathbf{1}[c_i < 0.5]c_i] \right) \tag{50}$$
$$= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{c_i \sim p(c)} \left[\min\{c_i, 1 - c_i\} \right] \tag{51}$$
$$= \mathbb{E}_{c \sim p(c)} \left[\min\{c, 1 - c\} \right] \tag{52}$$
$$= \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})} \left[\min\{p(y = 1 | \boldsymbol{x}), p(y = -1 | \boldsymbol{x})\} \right] \tag{53}$$
$$= \beta. \tag{54}$$

□

B.6 PROOF OF PROPOSITION 3.5

Proof. $\hat{\beta}_{\mathrm{Noise}}$ can be expressed as

$$\hat{\beta}_{\mathrm{Noise}} = \frac{1}{n}\sum_{i=1}^{n} \check{b}_i, \tag{55}$$

where

$$\check{b}_i := \mathbf{1}[c_i \geq 0.5] \cdot (1 - u_i) + \mathbf{1}[c_i < 0.5] \cdot u_i. \tag{56}$$

Since we can derive that $0 \leq \check{b}_i \leq 1$, we can again use the Hoeffding's inequality:

$$P(|\hat{\beta}_{\mathrm{Noise}} - \beta| \geq \epsilon) \leq 2\exp\left(-\frac{2n\epsilon^2}{(1-0)^2}\right) = 2\exp\left(-2n\epsilon^2\right). \tag{57}$$

Therefore, for any $\delta > 0$, with probability at least $1 - \delta$, we have,

$$|\hat{\beta}_{\mathrm{Noise}} - \beta| \leq \sqrt{\frac{1}{2n}\log\frac{2}{\delta}}. \tag{58}$$

□

B.7 PROOF OF THEOREM 4.1

Proof.

$$\begin{aligned}
\beta &= \mathbb{E}_{p(\boldsymbol{x})}[\min\{p(y=+1|\boldsymbol{x}), p(y=-1|\boldsymbol{x})\}] && (59)\\
&= \int \min\{p(y=+1|\boldsymbol{x}), p(y=-1|\boldsymbol{x})\} p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} && (60)\\
&= \int \min\{p(y=+1|\boldsymbol{x}), p(y=-1|\boldsymbol{x})\} \frac{\pi_+ p(\boldsymbol{x}|y=+1)}{p(y=+1|\boldsymbol{x})}\mathrm{d}\boldsymbol{x} && (61)\\
&= \int \min\{p(y=+1|\boldsymbol{x}), 1-p(y=+1|\boldsymbol{x})\} \frac{\pi_+ p(\boldsymbol{x}|y=+1)}{p(y=+1|\boldsymbol{x})}\mathrm{d}\boldsymbol{x} && (62)\\
&= \int \min\left\{1, \frac{1-p(y=+1|\boldsymbol{x})}{p(y=+1|\boldsymbol{x})}\right\} \pi_+ p(\boldsymbol{x}|y=+1)\mathrm{d}\boldsymbol{x} && (63)\\
&= \pi_+ \mathbb{E}_+\left[\min\left\{1, \frac{1-r(\boldsymbol{x})}{r(\boldsymbol{x})}\right\}\right] && (64)\\
&= \pi_+\left(1 + \mathbb{E}_+\left[\min\left\{0, \frac{1}{r(\boldsymbol{x})} - 2\right\}\right]\right) && (65)
\end{aligned}$$

□

B.8 PROOF OF PROPOSITION 4.2

Proof.

$$\begin{aligned}
\mathbb{E}_{\{\boldsymbol{x}_i\}_{i=1}^n \overset{\mathrm{i.i.d.}}{\sim} p(\boldsymbol{x}|y=+1)}[\hat{\beta}_{\mathrm{Pconf}}] &= \pi_+ - \frac{\pi_+}{n}\sum_{i=1}^{n}\mathbb{E}_+\left[\max\left\{0, 2 - \frac{1}{r(\boldsymbol{x}_i)}\right\}\right] && (66)\\
&= \pi_+\left(1 - \mathbb{E}_+\left[\max\left\{0, 2 - \frac{1}{r(\boldsymbol{x})}\right\}\right]\right) && (67)\\
&= \pi_+\left(1 + \mathbb{E}_+\left[\min\left\{0, \frac{1}{r(\boldsymbol{x})} - 2\right\}\right]\right) && (68)\\
&= \pi_+\left(1 + \left(\frac{\beta}{\pi_+} - 1\right)\right) && (69)\\
&= \beta && (70)
\end{aligned}$$

In the third equality, we used $\max\{a, b\} = -\min\{-a, -b\}$. In the fourth equality, we used $\mathbb{E}_+\left[\min\left\{0, \frac{1}{r(\boldsymbol{x})} - 2\right\}\right] = \frac{\beta}{\pi_+} - 1$ which can be derived from Eq. 65. □

B.9 PROOF OF PROPOSITION 4.3

We can rewrite $\hat{\beta}_{\mathrm{Pconf}}$ as

$$\hat{\beta}_{\mathrm{Pconf}} = \frac{1}{n_+} \sum_{i=1}^{n_+} \left(\pi_+ - \pi_+ \max \left\{ 0, 2 - \frac{1}{r(\boldsymbol{x}_i)} \right\} \right) \tag{71}$$

Although $1/r(\boldsymbol{x}_i)$ is not bounded, since $0 \leq \pi_+ - \pi_+ \max \left(0, 2 - \frac{1}{r(\boldsymbol{x}_i)} \right) \leq \pi_+$, we can use the Hoeffding's inequality to derive,

$$P(|\hat{\beta}_{\mathrm{Pconf}} - \beta| \geq \epsilon) \leq 2 \exp \left(\frac{-2n\epsilon^2}{\pi_+^2} \right) \tag{72}$$

Therefore, for any $\delta > 0$, with probability at least $1 - \delta$ we have,

$$|\hat{\beta}_{\mathrm{Pconf}} - \beta| \leq \sqrt{\frac{\pi_+^2}{2n} \log \frac{2}{\delta}}. \tag{73}$$

C STOCHASTIC BIG-O NOTATION

Suppose $\{X_n\}$ $(n = 1, 2, \ldots)$ is a sequence of random variables and $\{a_n\}$ $(n = 1, 2, \ldots)$ is a sequence of positive numbers, respectively. We say $X_n = O_p(a_n)$ when for all $\epsilon > 0$, there exists $C \in \mathbb{R}$ and finite N such that for all $n > N$, $P(\frac{|X_n|}{a_n} > C) < \epsilon$.