Understanding the Impact of Introducing Constraints at Inference Time on Generalization Error

Masaaki Nishino [1] **Kengo Nakamura** [1] **Norihito Yasuda** [1]

Abstract

Since machine learning technologies are being used in various practical situations, models with merely low prediction errors might not be satisfactory; prediction errors occurring with a low probability might yield dangerous results in some applications. Therefore, there are attempts to achieve an ML model whose input-output pairs are guaranteed to satisfy given constraints. Among such attempts, many previous works chose the approach of modifying the outputs of an ML model at the *inference time* to satisfy the constraints. Such a strategy is handy because we can control its output without expensive training or fine-tuning. However, it is unclear whether using constraints only in the inference time degrades a model's predictive performance. This paper analyses how the generalization error bounds change when we only put constraints in the inference time. Our main finding is that a class of loss functions preserves *the relative generalization error*, i.e., the difference in generalization error compared with the best model will not increase by imposing constraints at the inference time on multi-class classification. Some popular loss functions preserve the relative error, including the softmax cross-entropy loss. On the other hand, we also show that some loss functions do not preserve relative error when we use constraints. Our results suggest the importance of choosing a suitable loss function when we only use constraints in the inference time.

1. Introduction

Recent progress in machine learning (ML) enables models to achieve low prediction errors in many practically important tasks. As a result, machine learning is being used in a wide range of practical systems. However, models with merely low prediction errors might not be adequate for some situations since the errors that occur with a low probability potentially yield severe results. For example, we must avoid hazardous actions in safety-critical situations (Zhu et al., 2019; Leino et al., 2022). It is also important to avoid harmful generations of generative models (Gehman et al., 2020; Sheng et al., 2021). Since modern ML models are complex black-box systems with huge numbers of parameters, it is unrealistic to expect that we can extensively examine a model to guarantee the removal of such severe mistakes.

A simple but effective approach to avoid such errors is to make ML models that satisfy the given *constraints*. If we can guarantee that the outputs of ML models satisfy the given hard constraints, we can use them without excessive anxiety about the severe errors caused by violating them. Research is attempting to control a model's outputs by additional constraints (Mustafa et al., 2021; Ahmed et al., 2022; Leino et al., 2022; Hoernle et al., 2022; Qin et al., 2022). Many of these attempts use constraints only in the *inference time*, i.e., training models without constraints and using them for predictions with the trained models. Using constraints only in the inference time is more practical than using them in training and inference since users can flexibly change the constraints to control a model's output without re-training or fine-tuning. However, such ad-hoc use constraints might degrade performance.

Nishino et al. (2022) analyzed how the generalization error changes by adding constraints on the possible outputs of an ML model. They categorized the situations where constraints are used into two cases depending on whether they are accessed in the training time and named the situation where we can just access constraints only in the inference time as the *inference time verification (ITV)* setting. Nishino et al. (2022) showed that the model's generalization error, whose outputs are modified to satisfy constraints, can be bounded in ITV if the learning problem is PAC-learnable. However, how the generalization error is changed by adding constraints remains unknown for general cases.

This paper describes how generalization error changes in ITV settings, including non-PAC-learnable cases. Our key

[1] NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan. Correspondence to: Masaaki Nishino <masaaki.nishino@ntt.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

finding is that using constraints arbitrarily changes the generalization error depending on the constraints, although the difference in the generalization error of a hypothesis compared with the best possible model does not increase by imposing additional constraints on some situations. We name this property the *preservation of relative error*. We first show that relative error is preserved in a binary classification problem. Then we show conditions where relative error is preserved in multi-class classification problems. Unlike binary classification cases, relative error preservation depends on the loss function for multi-class classifications. We show a necessary and sufficient condition that loss functions preserve relative error and further demonstrate that typical loss functions, including softmax cross-entropy loss and one-versus-rest loss, satisfy the conditions. We also argue that some loss functions, including the multi-class margin loss function, cannot preserve relative error and show some worst-case results for these loss functions, indicating that relative error is much worse in the ITV setting. Our analyses show that "good models are good," i.e., a model with small relative error without constraints can be improved if we use it in the ITV setting with appropriate loss functions. This finding helps determine when the ITV is adequate and when we should retrain or fine-tune a model.

2. Related Work

Using constraints to control the outputs of machine learning models has been investigated for years, including hybrid models with logic as well as probabilistic models (Richardson & Domingos, 2006; Poon & Domingos, 2011) and neuro-symbolic AI (Manhaeve et al., 2018; Cohen, 2016; van Krieken et al., 2023; Ahmed et al., 2023). Historically, much previous research has used constraints as background knowledge to improve the prediction performance or logical rules instead of training data to deal with data scarcity problems (Mustafa et al., 2021; Chang et al., 2012). Adding constraints is again gathering attention for two reasons. First, machine learning models tend to become too huge, and the cost of training or fine-tuning them becomes excessive compared to using a model to perform predictions. Adding constraints to ML models' output is an inexpensive way to control their outputs (Qin et al., 2022; Zhang et al., 2023). Second, since machine learning technologies are being used in a wider range of situations, demand is growing for models whose input-output pairs are guaranteed to satisfy constraints (Hoernle et al., 2022; Ahmed et al., 2022; Giunchiglia et al., 2023). These trends are motivating analyses of generalization error bounds when we use constraints, especially in inference times.

Nishino et al. (2022) provided generalization analyses when constraints are placed on the output of a machine learning model. They formulated the problem setting as *learning with a verifier* module and gave generalization analyses in two different situations based on when constraints are imposed on the input-output pairs of a model. Their analyses argue that if a model is PAC-learnable, the estimated model's prediction error can be guaranteed not to exceed the other models in the hypothesis class that are modified to satisfy constraints. However, their analyses did not identify generalization error bound for general cases. Pukdee et al. (2023) also described generalization analyses when there are explainability constraints.

We address a problem setting that resembles domain adaptation. In domain adaptation problem, we use ML models trained in a different domain, and there are many theoretical results about the problem (Mansour et al., 2009; Ben-David et al., 2010; Redko et al., 2020). Adding constraints to a problem can be seen as changing the domain from which we train a model. However, our problem setting does not match a typical domain adaptation problem. In domain adaptation, we can access additional resources like unlabeled data to evaluate the closeness of domains, a strategy that is a key to deriving error bounds. In contrast, our problem setting is rather restricted since we can access a model and constraints. Therefore, we cannot directly apply the theoretical results of the domain adaptation to our problem.

3. Preliminaries

General notations Let $\mathcal{X}$ be the input domain, and let $\mathcal{Y}$ be the domain of the output labels. When a task is a binary classification, then $\mathcal{Y} = \{-1, 1\}$ and $\mathcal{Y} = [K]$ if it is a multi-class classification, where $K > 2$ is an integer and $[K] = \{1, \dots, K\}$. Let $\mathcal{D}$ be the unknown probability distribution over $\mathcal{X} \times \mathcal{Y}$, and let S be an example that consists of n samples $S = ((x_1, y_1), \dots, (x_n, y_n))$ drawn from $\mathcal{D}$ under the i.i.d assumption. Let h be a hypothesis that maps $\mathcal{X}$ to $\mathcal{Y}$. As shown below, we use slightly different definitions of h for binary and multi-class classification. Let $\mathcal{H}$ be a set of hypotheses or a hypothesis class.

Binary classification Let $h : \mathcal{X} \to \mathbb{R}$ be a hypothesis for binary classification that predicts the label of input x, where we define the prediction of h for x as $+1$ if $h(x) > 0$, and otherwise -1. Given a sample (x, y), we call $m = yh(x)$ the margin. If m is positive, then h can classify x correctly. Otherwise, h misclassifies it. Let $\ell : \mathbb{R} \to \mathbb{R}$ be a margin-based binary loss function, which takes margin m as input and outputs a value that tends to be small with a large margin. The most important binary loss function is zero-one loss $\ell_{0\text{-}1}$, defined as $\ell_{0\text{-}1}(m) = \mathbf{1}_{m \leq 0}$, where $\mathbf{1}_\omega$ is an indicator function for event ω. The following other binary loss functions are frequently used in the literature:

- Hinge loss: $\ell(m) = \max(1 - m, 0)$,

- Ramp loss: $\ell(m) = \frac{1}{2}\min(2, \max(1-m+t, 0))$,
- Sigmoid loss: $\ell(m) = \frac{1}{1+\exp(m)}$,
- Logistic loss: $\ell(m) = \ln(1+\exp(-m))$,

where $t > 0$ is a parameter for the ramp loss. The logistic loss equals the binary cross-entropy loss if we use $h(x)$ as a logit for $x = 1$. The generalization error of hypothesis h with margin loss function ℓ is defined as $R(h) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell(yh(x))\right]$.

Multi-class classification We represent a hypothesis in multi-class classification as $\mathbf{h}(x) := (h_1(x), \ldots, h_K(x))$, where $h_y : \mathcal{X} \to \mathbb{R}$ ($y \in [K]$) is a score for class y. Hypothesis $\mathbf{h}$ predicts class label y for input x as $y = \operatorname*{argmax}_{y\in[K]} h_y(x)$. Let $\mathcal{L}(\mathbf{h}(x), y)$ be a loss function for multi-class classification. The following loss functions are typically used in the literature (Zhang, 2004; Sugiyama et al., 2022; Mohri et al., 2012):

- Softmax cross-entropy (CE):
  $\mathcal{L}(\mathbf{h}(x), y) = -h_y(x) + \ln\left(\sum_{y'\in\mathcal{Y}} \exp(h_{y'}(x))\right)$,
- One-versus-rest (OVR):
  $\mathcal{L}(\mathbf{h}(x), y) = \ell(h_y(x)) + \frac{1}{c-1}\sum_{y'\neq y} \ell(-h_{y'}(x))$,
- Pairwise comparison (PC):
  $\mathcal{L}(\mathbf{h}(x), y) = \sum_{y'\neq y} \ell(h_y(x) - h_{y'}(x))$,
- (Multi-class) margin:
  $\mathcal{L}(\mathbf{h}(x), y) = \ell\left(h_y(x) - \max_{y'\neq y} h_{y'}(x)\right)$,

where $\ell(m)$ is a binary margin-based loss function. CE is classification-calibrated, and OVR and PC are classification-calibrated if combined with specific binary margin loss functions (Sugiyama et al., 2022). The generalization error for multi-class classification with loss function $\mathcal{L}$ is defined as $R(h) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathcal{L}(\mathbf{h}(x), y)\right]$.

3.1. Learning with a Verifier

We formulate the inference problem under constraints following the *learning with a verifier* formulation of Nishino et al. (2022). We assume that the constraints are represented as *requirement function* $c : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$. A requirement function indicates whether an input-output pair satisfies the given constraints, i.e., $c(x, y) = 1$ if pair x, y satisfies them; otherwise, $c(x, y) = 0$. In other words, requirement function c defines feasible set $T_x \subseteq \mathcal{Y}$ for every $x \in \mathcal{X}$. Requirement function c can represent a wide range of constraints over input-output pair (x, y) in the literature. For example, suppose that h is a recommender system that predicts suitable item y to propose to user x. If we do not want to recommend item $y \in \bar{T}_x \subseteq \mathcal{Y}$ to customers in set $S \subseteq \mathcal{X}$, then we can represent such knowledge as requirement function c such that $c(x, y) = 0$ for $x \in S$ and $y \in \bar{T}_x$. Nishino et al. (2022) showed more examples that the requirement function can represent. For simplicity, we make an assumption that there always exists y that satisfies the constraints, that is, there exists $y \in \mathcal{Y}$ satisfying $c(x, y) = 1$ for every $x \in \mathcal{X}$. We discuss in Section 6 how easily this assumption can be relaxed.

We guarantee that the output of the hypotheses always satisfies requirement c by modifying hypothesis h to h_c and using it for predictions. In binary classification, $h_c : \mathcal{X} \to \mathbb{R}$ is defined as

$$h_c(x) = \begin{cases} -M & (c(x, +1) = 0), \\ M & (c(x, -1) = 0), \\ h(x) & \text{(otherwise)}\,, \end{cases} \tag{1}$$

where $M > 0$ is a large constant. In a multi-class classification, we assume each score function h_y for label y is modified as h_{cy} and defined as

$$h_{cy}(x) = \begin{cases} h_y(x) & (c(x, y) = 1)\,, \\ -M & (c(x, y) = 0)\,, \end{cases} \tag{2}$$

where we define $M > 0$ as a large constant satisfying $-M < \inf_{h\in\mathcal{H}, x\in\mathcal{X}, y\in\mathcal{Y}} h_y(x)$.

There are two problem settings for learning with a verifier depending on when we use constraints. The first setting uses them only in the inference phase, not in the learning phase. That is, we estimate hypothesis $\hat{h}$ from $\mathcal{H}$ with learning algorithm $A : (\mathcal{X} \times \mathcal{Y})^m \to \mathcal{H}$ and training example S, as $\hat{h} = A(S)$. We then modify $\hat{h}$ to $\hat{h}_c$ using c for predicting labels for unknown inputs x for the inference phase. This setting is called *inference time verification* (ITV). The second setting uses constraints both in the inference and training phases. Learning algorithm A estimates hypothesis h_c from $\mathcal{H}_c$ from training example S and requirement function c. This setting is called *learning time verification* (LTV).

Compared to LTV, the ITV setting is cheaper and more flexible since we can use new constraints that were unavailable in the training phase. Furthermore, some situations in which those who train a model differ from those who use pre-trained model to perform inferences, and the users have specific requirements. ITV is the only way for users to reflect on their requirements in such situations. For example, if parents want to set additional filtering rules for the videos recommended for their children, the situation corresponds to ITV.

Nishino et al. (2022) provided generalization analyses for both settings. They described a generalization error bound for the LTV setting based on Rademacher complexity that generally holds for any combination of requirement function c and hypothesis $\mathcal{H}$. On the other hand, the generalization error bound for the ITV setting is unknown except for the case where $\mathcal{H}$ is probably approximately correct (PAC)-learnable. The following sections analyze the generalization error in the ITV setting.

4. Preservation of Relative Errors

We analyze how generalization error changes in ITV, i.e., we see **how differently the generalization error with constraints $R(\mathbf{h}_c)$ can be compared with the original generalization error $R(\mathbf{h})$ of hypothesis $\mathbf{h}$**.

The change of the generalization error depends on a combination of distribution $\mathcal{D}$ and requirement function c. Obviously, $R(\mathbf{h}_c)$ can be arbitrarily worse than $R(\mathbf{h})$ if we add constraints to prohibit pairs x, y that appear with a high probability in $\mathcal{D}$. Note that if the prediction is worsened due to c, this result includes the price that all the hypotheses must pay to guarantee that the modified model satisfies the constraints. As described in the introduction, our position prioritizes satisfying the constraints rather than lowering the prediction error. Therefore, it is natural to evaluate the quality of a hypothesis by its *relative error* compared with the best possible hypothesis that satisfies requirements c. We introduce the following optimal hypothesis $\mathbf{g}^\star : \mathcal{X} \to \mathbb{R}^K$ without constraints for a multi-class classification:

$$\mathbf{g}^\star := \operatorname*{argmin}_{\mathbf{g}\in\mathbb{R}^K} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}(\mathbf{g}(x), y)\right] .$$

We can similarly define the optimal hypothesis for binary classification. $\mathbf{g}^\star$ equals a Bayes-optimal hypothesis if loss $\mathcal{L}$ is a multi-class zero-one loss, which equals a multi-class margin loss function combined with a binary zero-one loss function.

Next we define $\mathbf{f}_c^\star$ as an optimal hypothesis that satisfies requirement function c. Hypotheses satisfying c correspond to $\mathbf{g}_c : \mathcal{X} \to \mathbb{R}^K$ that satisfies $g_{cy}(x) = -M$ if $c(x, y) = 0$, where $M > 0$ is a large constant. We define an optimal hypothesis satisfying constraints as

$$\mathbf{f}_c^\star := \operatorname*{argmin}_{\mathbf{g}_c} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}(\mathbf{g}_c(x), y)\right] .$$

We define the relative error as follows.

Definition 4.1. We define the *relative error* of $\mathbf{h}$ without constraints as $R(\mathbf{h}) - R(\mathbf{g}^\star)$. Similarly the relative error with constraints c is defined as $R(\mathbf{h}_c) - R(\mathbf{f}_c^\star)$. The relative error is preserved for loss function $\mathcal{L}$ if

$$R(\mathbf{h}) - R(\mathbf{g}^\star) \geq R(\mathbf{h}_c) - R(\mathbf{f}_c^\star) \tag{3}$$

holds for any combinations of $\mathbf{h}$, $\mathcal{D}$, and c.

If the relative error is preserved for loss function $\mathcal{L}$, it means that hypothesis $\mathbf{h}$ with small relative error $R(\mathbf{h}) - R(\mathbf{g}^\star)$ will achieve small relative error $R(\mathbf{h}_c) - R(\mathbf{f}_c^\star)$ in ITV settings. Therefore, we can confidently use $\mathbf{h}$ with a small relative error in ITV settings if we know the relative error is preserved.

Note that modified hypothesis $\mathbf{h}_c$ is defined by revising $\mathbf{h}$ to satisfy c per Equation (2). On the other hand, $\mathbf{f}_c^\star$ can be different from the one obtained by modifying $\mathbf{g}^\star$ to satisfy c. Below we discuss how this difference affects whether a loss function preserves the relative error.

In the following, we show situations where relative error is preserved. We first show that relative error is always preserved in binary classification and then the multi-class classification result. Unlike the binary classification case, relative error preservation for a multi-class classification depends on the loss function.

4.1. Binary Classification

We show that the relative error is preserved for binary classification.

Theorem 4.2. *Let $h : \mathcal{X} \to R$ be a hypothesis for binary classification, and let $c : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ be a requirement function. Let h_c be a modified hypothesis defined following Equation (1) with large constant $M > 0$, where $\ell : \mathbb{R} \to \mathbb{R}$ is a binary loss function. Let $g^\star$ be a minimizer of the generalization error, and let $f_c^\star$ be a minimizer of it while satisfying constraints c. Then for every requirement c, distribution $\mathcal{D}$, and hypothesis h, relative error $R(h_c) - R(f_c^\star)$ is preserved, i.e., it is not larger than $R(h) - R(g^\star)$ if $R(g^\star) > -\infty$.*

Proof. Let $\mathcal{C} \subseteq \mathcal{X}$ be a subset of $\mathcal{X}$ defined as $\mathcal{C} := \{x \mid c(x, -1) = c(x, +1) = 1, x \in \mathcal{X}\}$. If $x \notin \mathcal{C}$, then $h_c(x) = f_c^\star(x)$ for any h_c. If $x \in \mathcal{C}$, $h_c(x) = h(x)$, there exists optimal $f_c^\star$ satisfying $g^\star(x) = f_c^\star(x)$. Therefore, $R(h_c) - R(f_c^\star)$ becomes

$$\int \sum_{y\in\{-1,+1\}} p(x, y) \left[\ell(yh_c(x)) - \ell(yf_c^\star(x))\right] dx$$

$$= \int \mathbf{1}_{x\in\mathcal{C}} \left\{ \sum_{y\in\{-1,+1\}} p(x, y) \left[\ell(yh(x)) - \ell(yg^\star(x))\right] \right\} dx .$$

Since $\sum_{y\in\{-1,+1\}} p(x, y) \left[\ell(yh(x)) - \ell(yg^\star(x))\right]$ is always non-negative for every $x \in \mathcal{X}$, the above equation is not larger than $R(h) - R(g^\star)$. □

Adding constraints corresponds to uniquely determining output y for x satisfying $c(x, +1) = 0$ or $c(x, -1) = 0$. Therefore, it is intuitive that adding constraints will not increase the relative error.

4.2. Losses that Preserve Relative Errors

Unlike the binary classification case, relative error is not always preserved for multi-class classifications. Preservation depends on the loss function we employ. First, we show a necessary and sufficient condition for preserving the relative error. Then we show typical loss functions that preserve the relative error by using them.

Theorem 4.3. *Let $\mathbf{g}^\star : \mathcal{X} \to \mathbb{R}^K$ be an optimal hypothesis that minimizes the generalization error when using multi-class loss function $\mathcal{L}$, and let $c : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ be a requirement function. Let $\mathbf{f}^\star_c : \mathcal{X} \to \mathbb{R}^K$ be an optimal hypothesis with requirements c where $\mathbf{f}^\star_c$ satisfies the conditions of Equation (2) for some $M > 0$. Let $\hat{\mathbf{g}}_c : \mathcal{X} \to \mathbb{R}^K$ be the hypothesis obtained by substituting the value of $g^\star_y(x)$ with $-M$ when $c(x, y) = 0$. Then $\mathcal{L}$ preserves the relative error for any $\mathbf{h}$, $\mathcal{D}$, and c if and only if:*
(i) $R(\hat{\mathbf{g}}_c) = R(\mathbf{f}^\star_c)$, and
(ii) $R(\mathbf{h}) - R(\mathbf{g}^\star) \geq R(\mathbf{h}_c) - R(\hat{\mathbf{g}}_c)$
for any $\mathbf{h}$, c, and $\mathcal{D}$.

Proof. By substituting $\hat{\mathbf{g}}_c$ of condition (ii) with $\mathbf{f}^\star_c$ following condition (i), we can prove the if statement.

We next prove the only if statement. We assume that the relative error is preserved and $R(\hat{\mathbf{g}}_c) > R(\mathbf{f}^\star_c)$ holds. Substituting $\mathbf{h} = \mathbf{g}^\star$ to Equation (3), we have $0 \geq R(\hat{\mathbf{g}}_c) - R(\mathbf{f}^\star_c)$, which conflicts with the assumption. Hence $R(\hat{\mathbf{g}}_c) = R(\mathbf{f}^\star_c)$ holds. Finally, we assume that the relative error is preserved, but hypothesis $\mathbf{h}$ argues that condition (ii) does not hold. Then, since

$$R(\mathbf{h}) - R(\mathbf{g}^\star) < R(\mathbf{h}_c) - R(\hat{\mathbf{g}}_c) = R(\mathbf{h}_c) - R(\mathbf{f}^\star_c) \, ,$$

it contradicts the assumption that relative loss is preserved. Hence, condition (ii) also holds. □

We will prove that some surrogate loss functions preserve relative error when adding constraints by checking whether the loss satisfies the above two sufficient conditions. We first show that the softmax cross-entropy (CE) loss function can preserve relative error. CE loss is one of the most widely used surrogate loss functions for multi-class classification.

Proposition 4.4. *Let $\mathbf{h} : \mathcal{X} \to \mathbb{R}^K$ be a hypothesis in a vector form, and let $c : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ be a requirement function. Let $\mathbf{h}_c$ be a modified hypothesis, made by the values of $\mathbf{h}$ by setting $h_{cy}(x)$ satisfying $c(x, y) = 0$ to a constant M satisfying*

$$\frac{\exp(h_c y(x))}{\sum_y \exp(h_{cy}(x))} = \frac{\exp(-M)}{\sum_y \exp(h_{cy}(x))} = \delta \, ,$$

where δ is a small probability. We suppose that optimal hypothesis $\mathbf{f}^\star_c$ with requirement c satisfies

$$\frac{\exp f^\star_{cy}(x)}{\sum_y \exp(f^\star_{cy}(x))} = \frac{\exp -M}{\sum_y \exp(f^\star_{cy}(x))} = \delta \, ,$$

if $c(x, y) = 0$. Then for every requirement $c(x, y)$ and hypothesis $\mathbf{h} : \mathbf{X} \to \mathbb{R}^K$, relative error $R(\mathbf{h}_c) - R(\mathbf{f}^\star_c)$ is not larger than $R(\mathbf{h}) - R(\mathbf{g}^\star)$.

Proof. For simplicity, we consider a case where $\mathcal{X}$ is a singleton and write $h_y(x)$ and $g_y(x)$ as h_y, g_y by omitting x. Let $q^\star_y$ be the probability defined by $\mathbf{g}^\star$ as $q^\star_y = \frac{\exp(g^\star_y)}{\sum_{y'} \exp(g^\star_{y'})}$.

We first show that $R(\hat{\mathbf{g}}_c) = R(\mathbf{f}^\star_c)$ holds. Optimal hypothesis $\mathbf{g}^\star$, which minimizes the CE loss, achieves $q^\star_y = p_y$ for every $y \in [K]$, where p_y is the probability that y appears in distribution $\mathcal{D}$. Optimal hypothesis $\mathbf{f}^\star_c$ with constraints c corresponds to distribution $q^\star_{cy}$ satisfying $q^\star_{cy} = \delta$ if $c(x, y) = 0$ and minimizes

$$\sum_{y \in \mathcal{C}_0} p_y \ln \delta + \sum_{y \in \mathcal{C}_1} p_y \ln q^\star_{cy} \, ,$$

where $\mathcal{C}_0 \subseteq \mathcal{Y}$ is a set of $y \in \mathcal{Y}$ satisfying $c(x, y) = 0$, and $\mathcal{C}_1$ is a set of y satisfying $c(x, y) = 1$. By minimizing the above error with respect to $q^\star_{cy}$ under the constraints where $\sum_y q^\star_{cy} = 1$, optimal $q^\star_{cy}$ for $y \in \mathcal{C}_1$ equals

$$q^\star_{cy} = \frac{p_y}{\sum_{y' \in \mathcal{C}_1} p_{y'}} \, ,$$

when we set M to a large value so that δ is sufficiently small. Therefore, $\hat{\mathbf{g}}_c$ gives the same distribution with $\mathbf{f}^\star_c$, and condition (i) of Theorem 4.3 holds.

We next show that condition (ii) of Theorem 4.3 holds. The relative CE error for hypothesis $\mathbf{h}$ without constraints is

$$R(\mathbf{h}) - R(\mathbf{g}^\star) = -\sum_{y \in \mathcal{Y}} p_y \ln \frac{q_y}{p_y} \, ,$$

where q_y is the probability defined by $q_y = \exp(h_y)/\sum_{y'} \exp(h'_y)$. Using $R(\mathbf{f}^\star_c) = R(\hat{\mathbf{g}}_c)$, the relative error with CE loss with constraints becomes

$$\begin{aligned} R(\mathbf{h}_c) - R(\hat{\mathbf{g}}_c) &= -\sum_{y \in \mathcal{C}_1} p_y \ln \left[\frac{q_y}{p_y} \cdot \frac{\sum_{y' \in \mathcal{C}_1} p'_y}{\sum_{y' \in \mathcal{C}_1} q'_y} \right] \\ &= -\sum_{y \in \mathcal{C}_1} p_y \ln \frac{q_y}{p_y} - p_{\mathcal{C}_1} \ln \frac{p_{\mathcal{C}_1}}{q_{\mathcal{C}_1}}, \end{aligned}$$

where we use $p_{\mathcal{C}_1} = \sum_{y \in \mathcal{C}_1} p_y$ and $q_{\mathcal{C}_1} = \sum_{y \in \mathcal{C}_1} q_y$. The difference in relative error $(R(\mathbf{h}) - R(\mathbf{g}^\star)) - (R(\mathbf{h}_c) - R(\mathbf{f}^\star_c))$ is

$$\sum_{y \in \mathcal{C}_0} p_y \ln \frac{p_y}{q_y} + p_{\mathcal{C}_1} \ln \frac{p_{\mathcal{C}_1}}{q_{\mathcal{C}_1}} \, .$$

The above equation coincides with the Kullback-Leibler divergence between distributions p and q over the union of the elements of $\mathcal{C}_0$ and $\mathcal{C}_1$. Therefore, the equation is always non-negative for any combination of p and q if $q_y > 0$

for any $y \in \mathcal{Y}$. q_y is always nonzero since it is defined as the softmax of $\mathbf{h}$. Therefore, the relative error with CE loss with constraints is never larger than without constraints. Extending the above results to general $\mathcal{X}$ is straightforward. □

Similar results hold when we employ the OVR loss combined with any binary loss function if an optimal solution exists that minimizes the generalization error.

Proposition 4.5. *Let $\mathbf{h} : \mathcal{X} \to \mathbb{R}^K$ be a hypothesis, and let $c : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ be a requirement function. Let $\mathbf{h}_c$ be the hypothesis obtained by modifying $\mathbf{h}$ by following Equation (2). Let $\mathbf{g}^\star \in \mathbb{R}^K$ be an optimal hypothesis that minimizes the generalization error without constraints, and let $\mathbf{f}_c^\star \in \mathbb{R}^K$ be an optimal hypothesis minimizing the generalization error and satisfying $f^\star_{cy} = -M$ if $c(x, y) = 0$. For any c, $\mathcal{D}$, and $\mathbf{h}$, relative error $R(\mathbf{h}_c) - R(\mathbf{f}_c^\star)$ is not larger than $R(\mathbf{h}) - R(\mathbf{g}^\star)$ where the OVR loss is combined with any binary surrogate function if $\mathbf{g}^\star$ exists such that $R(\mathbf{g}^\star) > -\infty$.*

Proof. For simplicity, we again consider a case where $\mathcal{X}$ is a singleton. We first show that optimal solution $\mathbf{f}_c^\star$ coincides with $\hat{\mathbf{g}}_c$, the vector obtained by modifying $\mathbf{g}^\star$ by setting $\hat{g}_{cy} = -M$ when $c(x, y) = 0$. The generalization error with the OVR loss is

$$R(\mathbf{h}) = \sum_{y\in\mathcal{Y}} p_y \left[\ell(h_y) + \frac{1}{K-1}\sum_{y'\neq y}\ell(-h_{y'})\right]$$

where $\ell(m)$ is a binary surrogate loss function. We reformulate the equation as the sum of terms corresponding to every $y \in \mathcal{Y}$:

$$R(\mathbf{h}) = \sum_{y\in\mathcal{Y}} \left[p_y \ell(h_y) + \frac{1-p_y}{K-1}\ell(-h_y)\right] \tag{4}$$

where we use $\sum_{y'\neq y} p_{y'} = 1 - p_y$. The above equation is linear with $\ell(h_y)$ and $\ell(-h_y)$. Therefore, we can obtain optimal solution $g^\star$ by independently minimizing each term related to y by setting the best g_y. Using constraints corresponds to restricting values g_y to $-M$ if $c(x, y) = 0$. Therefore, the terms related to y with $c(x, y) = 0$ in Equation (4) become constant for every hypothesis $\mathbf{h}_c$. Optimal hypothesis $\mathbf{f}_c^\star$ also satisfies that $f^\star_{cy}$ is a constant when $c(x, y) = 0$. If $c(x, y) = 1$, $f^\star_{cy}$ is set to minimize the term in Equation (4) related to y. Therefore, both $f^\star_{cy}$ and $g^\star_y$ give the same optimal value for terms related to y satisfying $c(x, y) = 1$, and we can obtain optimal hypothesis $\mathbf{f}_c^\star$ by modifying the values of $\mathbf{g}^\star$ corresponding to $y \in \mathcal{C}_0$ to constant $-M$.

We next show $R(\mathbf{h}) - R(\mathbf{g}^\star) \geq R(\mathbf{h}_c) - R(\hat{\mathbf{g}}_c)$ for any c. $(R(\mathbf{h}) - R(\mathbf{g}^\star)) - (R(\mathbf{h}_c) - R(\hat{\mathbf{g}}_c))$ equals

$$\sum_{y\in\mathcal{C}_0}\left[p_y(\ell(h_y) - \ell(g^\star_y)) + \frac{1-p_y}{K-1}(\ell(-h_y) - \ell(-g^\star_y))\right]$$

since the terms that correspond to $y \in \mathcal{C}_1$ are zero. The above equation is non-negative since $m = g^\star_y$ minimizes term $p_y\ell(m) + \frac{1-p_y}{K-1}\ell(-m)$ for every y. □

Finally, we show that the relative error preserves with the PC loss when we employ either zero-one, sigmoid, or a ramp binary loss as a binary surrogate loss function.

Proposition 4.6. *Under the same assumption made on $\mathbf{h}$, c $\mathbf{g}^\star$, and $\mathbf{f}_c^\star$ in Proposition 4.5, the difference of relative error $R(\mathbf{h}_c) - R(\mathbf{f}_c^\star)$ is not larger than $R(\mathbf{h}) - R(\hat{\mathbf{g}}_c)$ where the PC loss is combined with the zero-one, sigmoid, and ramp surrogate functions if there exists $\mathbf{g}^\star$ that minimizes the generalization error.*

Proof. We prove the case where we employ the zero-one loss function. We first show that $R(\mathbf{f}_c^\star) = R(\hat{\mathbf{g}}_c)$. Then we show that the difference of relative error $R(\mathbf{h}_c) - R(\hat{\mathbf{g}}_c)$ is smaller than $R(\mathbf{h}) - R(\mathbf{g}^\star)$.

We assume $\mathcal{X}$ is a singleton for simplicity. The generalization error of the PC loss with the zero-one binary loss is

$$R(\mathbf{h}) = \sum_{y\in[K]} p_y \sum_{y'\neq y}\left[\ell_{\text{0-1}}(h_y - h_{y\prime})\right] .$$

We can minimize the generalization error with $\mathbf{g} \in \mathbb{R}^K$ where g_y follows the same order with p_y, i.e., if $p_{\pi(1)} \geq p_{\pi(2)} \geq \cdots \geq p_{\pi(k)}$, then $\mathbf{g}$ that satisfies $g_{\pi(1)} > g_{\pi(2)} > \cdots g_{\pi(k)}$ minimizes the generalization error, where $\pi : [K] \to [K]$ is a permutation over $[K]$. Optimal solution $\mathbf{g}^\star$ satisfies the order constraint.

If we add constraints, terms $\ell(g_y - g_{y\prime})$ are constant if $c(x, y) = 0$ or $c(x, y') = 0$. Therefore, the generalization error becomes:

$$R(\mathbf{h}_c) = \sum_{y\in\mathcal{C}_1} p_y \sum_{y'\in\mathcal{C}_1, y'\neq y} \ell_{\text{0-1}}(h_y - h_{y\prime}) + C \tag{5}$$

where C is a constant. Similar to the unconstrained case, $\mathbf{f}^\star_{\mathbf{c}}$ minimizes Equation (5) and satisfies the ordering constraint of p_y for $y \in \mathcal{C}_1$. If we modify $\mathbf{g}^\star$ by setting $g_y = -M$ for $y \in \mathcal{C}_0$, then the resulting vector satisfies the ordering constraint over p_y for $y \in \mathcal{C}_1$, and therefore, the vector obtained by modifying $\mathbf{g}^\star$ is optimal solution $\mathbf{f}_c^\star$ for the constrained problem.

We next show that $R(\mathbf{h}_c) - R(\hat{\mathbf{g}}_c)$ is not larger than $R(\mathbf{h}) - R(\mathbf{g}^\star)$. $R(\mathbf{h}_c) - R(\hat{\mathbf{g}}_c)$ equals:

$$\sum_{y\in\mathcal{C}_1} p_y \sum_{y'\in\mathcal{C}_1, y'\neq y} (\ell(h_y - h_{y'}) - \ell(g^\star_y - g^\star_{y'}))$$

where we use $\ell(h_{cy} - h_{cy'}) - \ell(g^\star_{cy} - g^\star_{cy'}) = 0$ if $y \in \mathcal{C}_0$ or $y' \in \mathcal{C}_0$. Using this equation, we show that the difference in relative error is

$$\begin{aligned}&(R(\mathbf{h}) - R(\mathbf{g}^\star)) - (R(\mathbf{h}_c) - R(\hat{\mathbf{g}}_c))\\&= \sum_{y\in\mathcal{C}_1} p_y \sum_{y'\in\mathcal{C}_0} (\ell(h_y - h_{y'}) - \ell(g^\star_y - g^\star_{y'}))\\&+ \sum_{y\in\mathcal{C}_0} p_y \sum_{y'\in[K]} (\ell(h_y - h_{y'}) - \ell(g^\star_y - g^\star_{y'}))\\&= \sum_{(y,y')\in[K]^2\setminus(\mathcal{C}_1)^2, y\neq y'} \big[p_y(\ell(h_y - h_{y'}) - \ell(g^\star_y - g^\star_{y'}))\\&+ p_{y'}(\ell(h_{y'} - h_y) - \ell(g^\star_{y'} - g^\star_y))\big]\end{aligned}$$

where we reformulate the formula as the sum of the error corresponding to pair y, y'. Since optimal solution $\mathbf{g}^\star$ minimizes term $p_y\ell(g^\star_y - g^\star_{y'}) + p_{y'}\ell(g^\star_{y'} - g^\star_y)$ for every pair $(y, y') \in [K]^2, y \neq y'$, the above equation is always non-negative.

When we use the sigmoid or ramp loss function, we can almost identically prove the error preservation using the conditions shown in Theorem 4.3. The difference is we need an additional condition for optimal solution $\mathbf{g}^\star$ so that $|g^\star_y - g^\star_{y'}|$ are sufficiently large for every $y \neq y'$ unless $y, y' \in \mathcal{C}_1$. □

4.3. Extension to structured predictions

Some recent neuro-symbolic AI works use constraints in structured prediction tasks (Ahmed et al., 2022; Giunchiglia & Lukasiewicz, 2022; Dragone et al., 2021), which are formulated as a multi-class classification where $\mathcal{Y} = (y_1, \dots, y_L) = \{0,1\}^L$ for some fixed L. We show that the relative error is preserved for this task if we use binary cross-entropy as a loss function. For sample $(x, \mathbf{y}) \in \mathcal{X}\times\mathcal{Y}$, the binary cross-entropy loss is defined as

$$\sum_{j=1}^{L} -y_j \ln q_j(x) - (1 - y_j)\ln(1 - q_j(x))$$

where $q_j(x) \in [0,1]$ is the hypothesis output, which represents the probability that $y_j = 1$. We assume that a hypothesis outputs vector $(q_1(x), \dots, q_L(x)) \in [0,1]^L$, which defines conditional probability $q(\mathbf{y} \mid x) = \prod_j q_j(y_j \mid x) = \prod_j (y_j q_j(x) + (1 - y_j)(1 - q_j(x))$.

Since using the above binary cross-entropy loss corresponds to multi-class cross-entropy loss under the assumption that a hypothesis gives factored conditional probability $\prod_j q_j(y_j \mid x)$, the relative error is preserved for this problem[1].

5. Losses that Do Not Preserve the Relative Error

In the previous section, we showed how relative error can be preserved for binary classification. We also show that three typical surrogate loss functions for multi-class classification preserve relative error for multi-class classification. However, such preservation does not hold for all loss functions. We prove that the multi-class margin loss function cannot preserve relative error by checking condition (i) of the Theorem 4.3.

First, we show that using the multi-class margin loss in combination with a specific binary loss function cannot preserve the relative error.

Proposition 5.1. *If $\mathcal{Y} = [K]$ and $K > 2$, using multi-class margin loss function $\mathcal{L}(\mathbf{h}(x), y) = \max_{y'\neq y} \ell(h_y(x) - h_{y'}(x))$ in combination with the zero-one, ramp, and sigmoid loss does not preserve the relative error. Moreover, if $K > 3$ and the binary loss function is differentiable at 0 and $\ell'(0) \neq 0$, then the margin loss does not preserve the relative error.*

Proof. We show situations where $R(\hat{\mathbf{g}}_c)$ differs from $R(\mathbf{f}^\star_c)$ and use condition (i) of Theorem 4.3. We again assume that $\mathcal{X}$ is a singleton. Without loss of generality, we assume that label distribution p_y satisfies $p_1 \geq p_2 \geq \cdots \geq p_K$.

We first prove the case where ℓ is a zero-one loss. The generalization error for $\mathbf{h}$ is

$$R(\mathbf{h}) = \sum_{y\in[K]} p_y \left[\ell\left(h_y - \max_{y'\neq y} h_{y'}\right)\right].$$

Since ℓ is a zero-one loss function, any $\mathbf{h}$ satisfying $h_1 > \max_{y'\in\{2,\dots,K\}} h_{y'}$ minimizes the generalization error. For example, $(h_1, \dots, h_K) = (1, 0, \dots, 0)$ minimizes the error. If we put requirement c that satisfies $c(x, y_1) = 0$ and otherwise $c(x, y) = 0$, then optimal solution $\mathbf{f}^\star_c$ under the constraint must satisfy $f^\star_{c1} = -M$ and $f^\star_{c2} > f^\star_{cy'}$ for $y' \in \{3, \dots, K\}$. On the other hand, modifying $\mathbf{g}^\star = (1, 0, \dots, 0)$ results in $\hat{\mathbf{g}} = (-M, 0, \dots, 0)$, which is not an optimal solution under the constraints. Therefore, the margin loss combined with the binary zero-one loss does not preserve relative error due to Theorem 4.3. Extending the results to the ramp and sigmoid loss is easy by setting $\mathbf{g}^\star = (T, 0, \dots, 0)$ where $T > 0$ is a sufficiently large constant.

Next we consider a condition where $K > 3$ and the loss is differentiable at 0 and $\ell'(0) \neq 0$. Without loss of generality, we assume that optimal solution $\mathbf{g}^\star$ satisfies $g^\star_1 \geq g^\star_2 \geq \cdots \geq g^\star_K$. The generalization error becomes:

$$R(\mathbf{g}^\star) = p_1\ell(g^\star_1 - g^\star_2) + \sum_{y\in\{2,\dots K\}} p_y\ell(g^\star_y - g^\star_1).$$

[1] We need to slightly modify the procedure for obtaining modified hypothesis q_c from q. Instead of modifying a scoring function, we obtain modified probability distribution $\mathbf{q}_c$ by changing $q(\mathbf{y} \mid x)$ to a very small probability when $c(x, \mathbf{y}) = 0$ and normalizing the distribution.

Since $g^\star_y$ for $f \in \{3, \dots, K\}$ only appears in term $p_y \ell(g^\star_y - g^\star_1)$ and the error is linear with $\ell(g^\star_y - g^\star_1)$, there exists an optimal solution satisfying $g^\star_3 = \cdots = g^\star_K = c$ if the error is bounded. Next we add requirement c such that $c(x, y) = 0$ for $y = 1, 2$. Then there exists $\hat{\mathbf{g}}_c = (-M, -M, c, \dots, c)$ whose error is

$$R(\hat{\mathbf{g}}_c) = (p_1 + p_2)\ell(-M - c) + \sum_{y \in \{3,\dots,K\}} p_y \ell(0) \,.$$

If $\ell'(0) \neq 0$, then subtracting value, $\delta > 0$, from every $\hat{g}_{cy}$, $y \in \{4, \dots, K\}$ decreases the error depending on probabilities $p_3, \dots, p_K$. Hence, we have $\hat{\mathbf{g}}_c \neq \mathbf{f}^\star_c$, and the loss does not preserve the relative error due to Theorem 4.3. □

Next we show that the PC loss cannot preserve relative errors when combined with a hinge or logistic loss.

Proposition 5.2. *Under the same assumption made on* $\mathbf{h}$*,* c $\mathbf{g}^\star$ *and* $\mathbf{g}^\star_c$ *in Proposition 4.5, the PC loss function combined with a hinge loss function does not preserve the relative error for* $K > 2$*.*

Proof. We prove by showing that $R(\hat{\mathbf{g}}_c)$ differs from $R(\mathbf{f}^\star_c)$ for a hinge loss. Adding requirement c changes term $\ell(g_{y'} - g_y)$ to $1 - (M + g_y)$ if $c(x, y') = 0$ and $c(x, y) = 1$. This term becomes smaller as we move g_y close to M. Therefore, $R(\hat{\mathbf{g}}_c) \neq R(\mathbf{f}^\star_c)$ for some $\mathcal{D}$. Similar results hold for a logistic loss. □

5.1. Worst-case Analyses

We show that some loss functions for multi-class classification do not preserve the relative error. This means the error might be much worse when we employ these errors. How large can the degradation be when we use a loss function that does not preserve relative errors? The following theorem shows that combining the multi-class margin loss and some binary loss functions achieves the worst relative error.

Theorem 5.3. *For every hypothesis* $\mathbf{h} : \mathcal{X} \to \mathbb{R}^K$ *that satisfies* $h_y(x) \neq h_{y'}(x)$ *for any* $x \in \mathcal{X}$ *and* $y \neq y'$*, there exists a combination of distribution* $\mathcal{D}$ *and requirement* c *that achieves* $R(\mathbf{h}) - R(\mathbf{g}^\star) = 0$ *and* $R(\mathbf{h}_c) - R(\mathbf{f}^\star_c) \geq 1/2$ *when we use the multi-class margin loss in combination with the zero-one, ramp, or sigmoid binary loss functions.*

Proof. We design distribution $\mathcal{D}$ as (i) by assigning largest conditional probability $p(y|x)$ to y satisfying $y = \operatorname{argmax} h_y(x)$ for every x and (ii) by assigning the smallest conditional probability $p(y|x)$ to y where $h_y(x)$ is the second largest value for every x. Let $\hat{y}_x = \operatorname*{argmax}_y p(y|x)$. Next we set c as a mapping satisfying $c(x, y) = 0$ for $y = \operatorname*{argmax}_{y \in \mathcal{Y}} h_y(x)$, and otherwise $c(x, y) = 1$.

We consider a case where we employ the zero-one binary loss function. Under this combination of $\mathcal{D}$ and c, optimal solution $\mathbf{g}^\star$ for the unconstrained setting satisfies $\operatorname*{argmax}_y g^\star_y(x) = \hat{y}_x$, and relative error $R(\mathbf{h}) - R(\mathbf{g}^\star)$ is zero. We fix x to a specific value and omit it for simplicity. If we add constraints c, optimal solution $\mathbf{f}^\star_c$ assigns largest score $f^\star_{cy}$ to y with highest probability $p(y)$ among $y \in [K] \setminus \hat{y}$ to minimize the generalization error, which for the solution is $1 - \max_{y \in [K]\setminus\hat{y}} p(y)$. On the other hand, the generalization error for $\mathbf{h}_c$ is $1 - \min_{y \in [K]\setminus\hat{y}} p(y)$ since $\mathcal{D}$ assigns the smallest probability to $y = \operatorname*{argmax}_{y \in [K]\setminus\hat{y}} h_y$. Therefore, the relative error under the constraints is $\max_{y \in [K]\setminus\hat{y}} p(y) - \min_{y \in [K]\setminus\hat{y}} p(y)$, which becomes largest when $\max_{y \in [K]\setminus\hat{y}} p(y) = 1/2 - \epsilon$, where $\epsilon > 0$, and $\min_{y \in [K]\setminus\hat{y}} p(y) = 0$. Extending the above discussion to general $\mathcal{X}$ is straightforward. □

6. Discussion

Relation to the previous results in the ITV setting Nishino et al. (2022) gave generalization analyses in the ITV setting. Their main findings on the ITV setting are two-fold. First, the generalization error with the multi-class zero-one loss of learned hypothesis $\hat{h}_c$ is not worse than the other hypotheses, that is,

$$R(\hat{h}_c) \leq \inf_{h_c \in \mathcal{H}_c} R(h_c) + \epsilon$$

with high probability if hypothesis class $\mathcal{H}$ is PAC-learnable. Second, if $\mathcal{H}$ is not PAC-learnable, then there exists $\hat{h}$ and c such that $R(\hat{h}_c)$ can be larger than other hypotheses in $\mathcal{H}_c$ even with a sufficient number of training examples.

Two major deviations seem apparent between the results in (Nishino et al., 2022) and those in our paper since the previous results do not depend on the loss functions. We show that these results are consistent. The first deviation is that the former paper showed a bound for the generalization error with zero-one multi-class loss if the problem is PAC-learnable; the relative error is not preserved when we use the margin loss combined with the zero-one binary loss function, which coincides with the zero-one multi-class loss function. These results are consistent since Nishino et al. (2022) assumed PAC-learnability. If a problem is PAC-learnable, then target distribution $\mathcal{D}$ must be deterministic, i.e., $\mathcal{D}$ assigns $p(y \mid x) = 1$ for some y for every x with $p(x) > 0$. In such a case, the relative error will be preserved for any constraints.

The second deviation is that the previous paper argues that the generalization error of $\hat{\mathbf{h}} = A(S)$ is not bounded if the problem is not PAC-learnable; our results show that the relative error will be preserved if we use an appropriate multi-class loss function. These results are also consistent since the former paper focused on how the generalization error compared with *other hypotheses* where $\mathcal{H}$ changes, and ours focused on how the error compared with *the best model* changed. We show an example consistent with both statements; we have $R(\mathbf{h}) - R(\mathbf{g}^\star) \geq R(\mathbf{h}_c) - R(\mathbf{f}_c^\star)$ for all $\mathbf{h} \in \mathcal{H}$ if we use the CE loss. However, there might exists $\mathbf{h}'$ satisfying $R(\mathbf{h}') > R(\hat{\mathbf{h}})$ and $R(\mathbf{h}'_c) < R(\hat{\mathbf{h}}_c)$.

Limitation of ITV setting The above discussion identifies a limitation of the ITV setting even if we employ a loss function preserving the relative error. The generalization error of hypothesis $\hat{\mathbf{h}}_c$ would be preserved, but we might overlook other hypotheses $\mathbf{h}'$ where $R(\mathbf{h}'_c)$ is much smaller than $R(\hat{\mathbf{h}}_c)$ in the ITV setting. Our next question: how can a good $R(\mathbf{h}'_c)$ be compared with $R(\hat{\mathbf{h}}_c)$? If we use the CE loss, perhaps $R(\mathbf{h}'_c) - R(\mathbf{f}_c^\star) = 0$ even if $R(\mathbf{h}') - R(\mathbf{g}^\star)$ is large. Can we estimate such a best hypothesis in $\mathcal{H}$ in the ITV setting? That is an open problem, although we expect answering it will be difficult if we cannot access the constraints during the learning phase. If the relative error of the estimated hypothesis is not small, then we expect a subpar ITV performance, which might be regarded as a trade-off for its convenience. In contrast, while the LTV setting lacks the convenience inherent in the ITV, it possesses potential for preserving performance.

Intuitive explanation of loss functions that preserve errors We next describe insights into why the multi-class margin loss cannot preserve relative error by comparing it with PC loss, which does retain the relative error. We assume that the PC and margin losses employ zero-one binary loss. As shown in the proof, the minimizer of the margin loss assigns the largest score to label $\hat{y}$ with highest probability $p(y|x)$, although the margin loss cannot distinguish solutions assigning different scores to $h_{y'}$ for $y^`prime \neq \hat{y}$. Therefore, score $g_{y'}(x)$ for $y' \neq \hat{y}$ might not reflect conditional distribution $p(yx)$. If we add constraints where $c(x, \hat{y}) = 0$, the modified hypothesis is not guaranteed to work well.

In contrast, as shown in the proof of Proposition 4.6, the minimizer of the PC loss achieves a minimum when $g_y^\star$ follows the order of $p(y \mid x)$. Therefore, we expect a hypothesis with a small relative error with PC loss to work when we prohibit some y. This example shows the importance of the preservation of relative error when we use constraints to restrict a model's prediction.

Relaxing the feasibility assumption In Section 3.1, we assumed that there exists $y \in \mathcal{Y}$ such that $c(x, y) = 1$ for all $x \in \mathcal{X}$. Here we discuss a way to relax that assumption. Let $\mathcal{I}_c = \{x \mid x \in \mathcal{X}, \forall y \in \mathcal{Y} : c(x, y) = 0\}$. $x \in \mathcal{I}_c$ is *infeasible* since no y satisfies the requirements. Otherwise, x is *feasible*. To deal with infeasible x, we have to extend each hypothesis $\mathbf{h}_c$ to be able to reject such input x. Similarly, we must extend loss function $\mathcal{L}$ to calculate the loss value if hypothesis $\mathbf{h}_c$ rejects input x.

There are multiple ways to extend $\mathbf{h}_c$ and $\mathcal{L}$ to deal with the rejection of infeasible x. However, it is natural to assume that $\mathcal{L}(\mathbf{h}_c(x), y) = \mathcal{L}(\mathbf{f}_c^\star(x), y)$ for any $x \in \mathcal{I}_c$, $y \in \mathcal{Y}$ $\mathbf{h}_c$, and $\mathbf{f}_c^\star$ since whether we reject x depends only on x and c. Next we represent distribution $\mathcal{D}$ as $\mathcal{D} = \lambda\mathcal{D}_I + (1-\lambda)\mathcal{D}_F$, where $\lambda \in [0, 1]$ and $\mathcal{D}_I$ is a probability distribution defined over a subset of $\mathcal{X} \times \mathcal{Y}$ such that $x \in \mathcal{I}_c$ and $\mathcal{D}_F$ is a distribution defined over a subset of $\mathcal{X} \times \mathcal{Y}$ such that $x \notin \mathcal{I}_c$. Then the relative error between $\mathbf{h}$ and $\mathbf{g}^\star$, $\mathbf{h}_c$ and $\mathbf{f}_c^\star$ under distribution $\mathcal{D}$ becomes

$$\begin{aligned} R_{\mathcal{D}}(\mathbf{h}) - R_{\mathcal{D}}(\mathbf{g}^\star) &= \lambda(R_{\mathcal{D}_I}(\mathbf{h}) - R_{\mathcal{D}_I}(\mathbf{g}^\star)) \\ &\quad + (1-\lambda)(R_{\mathcal{D}_F}(\mathbf{h}) - R_{\mathcal{D}_F}(\mathbf{g}^\star)) \\ R_{\mathcal{D}}(\mathbf{h}_c) - R_{\mathcal{D}}(\mathbf{f}_c^\star) &= \lambda(R_{\mathcal{D}_I}(\mathbf{h}_c) - R_{\mathcal{D}_I}(\mathbf{f}_c^\star)) \\ &\quad + (1-\lambda)(R_{\mathcal{D}_F}(\mathbf{h}_c) - R_{\mathcal{D}_F}(\mathbf{f}_c^\star)) \,, \end{aligned}$$

where we use $R_{\mathcal{D}}$ to represent that the error is computed for the distribution $\mathcal{D}$. Since $R_{\mathcal{D}_I}(\mathbf{h}_c) - R_{\mathcal{D}_I}(\mathbf{f}_c^\star) = 0$ from the assumption, the relative error for infeasible x is preserved, i.e., $R_{\mathcal{D}_I}(\mathbf{h}) - R_{\mathcal{D}_I}(\mathbf{g}^\star) \geq R_{\mathcal{D}_I}(\mathbf{h}_c) - R_{\mathcal{D}_I}(\mathbf{f}_c^\star) = 0$. Therefore, the above equations show that the relative error is preserved if it is done under feasible distribution $\mathcal{D}_F$. In this way, we can relax the feasibility assumption.

7. Conclusion

This paper analyzed the effect of adding constraints to modify the outputs of previously trained models. Relative generalization error is preserved when we use a class of multi-class loss functions. The class contains important functions, including cross-entropy loss and one-versus-rest functions. We describe a necessary and sufficient condition where a loss function can preserve relative error. We also show a hardness result of the margin-based loss function, which does not preserve the relative error.

Adding constraints to perform predictions with pre-trained models is a realistic choice for exploiting machine learning models because larger models are used in many practical applications. Our results give a theoretical understanding of this topic and emphasize the importance of selecting appropriate loss functions.

Acknowledgements

The authors thank the anonymous reviewers for their valuable feedback, corrections, and suggestions. This work was supported by JST PRESTO (Grant Number JPMJPR20C7, Japan), JST CREST (Grant Number JPMJCR22D3, Japan), and JSPS KAKENHI (Grant Number JP20H05963, Japan).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Ahmed, K., Teso, S., Chang, K.-W., Van den Broeck, G., and Vergari, A. Semantic probabilistic layers for neuro-symbolic learning. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 29944–29959. Curran Associates, Inc., 2022.

Ahmed, K., Chang, K.-W., and Van den Broeck, G. Semantic strengthening of neuro-symbolic learning. *AISTATS*, pp. 10252–10261, February 2023.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

Chang, M.-W., Ratinov, L., and Roth, D. Structured learning with constrained conditional models. *Mach. Learn.*, 88 (3):399–431, sep 2012. ISSN 0885-6125. doi: 10.1007/s10994-012-5296-5.

Cohen, W. W. Tensorlog: A differentiable deductive database. *CoRR*, abs/1605.06523, 2016.

Dragone, P., Teso, S., and Passerini, A. Neuro-Symbolic constraint programming for structured prediction. *International Workshop on Neural-Symbolic Learning and Reasoning*, 2021.

Gehman, S., Gururangan, S., Sap, M., Choi, Y., and Smith, N. A. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3356–3369, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.301.

Giunchiglia, E. and Lukasiewicz, T. Multi-Label classification neural networks with hard logical constraints. *J. Artif. Intell. Res.*, 72:759–818, January 2022.

Giunchiglia, E., Stoian, M. C., Khan, S., Cuzzolin, F., and Lukasiewicz, T. ROAD-R: the autonomous driving dataset with logical requirements. *Mach. Learn.*, 112 (9):3261–3291, September 2023.

Hoernle, N., Karampatsis, R. M., Belle, V., and Gal, K. Multiplexnet: Towards fully satisfied logical constraints in neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5):5700–5709, Jun. 2022. doi: 10.1609/aaai.v36i5.20512.

Leino, K., Fromherz, A., Mangal, R., Fredrikson, M., Parno, B., and Păsăreanu, C. Self-correcting neural networks for safe classification. In Isac, O., Ivanov, R., Katz, G., Narodytska, N., and Nenzi, L. (eds.), *Software Verification and Formal Methods for ML-Enabled Autonomous Systems*, pp. 96–130, Cham, 2022. Springer International Publishing. ISBN 978-3-031-21222-2.

Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., and De Raedt, L. Deepproblog: Neural probabilistic logic programming. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation: Learning bounds and algorithms. In *Proceedings of The 22nd Annual Conference on Learning Theory (COLT 2009)*, Montréal, Canada, 2009.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. The MIT Press, 2012. ISBN 026201825X.

Mustafa, W., Lei, Y., Ledent, A., and Kloft, M. Fine-grained generalization analysis of structured output prediction. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 2841–2847. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/391. Main Track.

Nishino, M., Nakamura, K., and Yasuda, N. Generalization analysis on learning with a concurrent verifier. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 4177–4188. Curran Associates, Inc., 2022.

Poon, H. and Domingos, P. Sum-product networks: A new deep architecture. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, pp. 337–346, Arlington, Virginia, USA, 2011. AUAI Press. ISBN 9780974903972.

Pukdee, R., Sam, D., Kolter, J. Z., Balcan, M.-F. F., and Ravikumar, P. Learning with explanation constraints. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 49883–49926. Curran Associates, Inc., 2023.

Qin, L., Welleck, S., Khashabi, D., and Choi, Y. Cold decoding: Energy-based constrained text generation with langevin dynamics. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 9538–9551. Curran Associates, Inc., 2022.

Redko, I., Morvant, E., Habrard, A., Sebban, M., and Bennani, Y. A survey on domain adaptation theory. *CoRR*, abs/2004.11829, 2020.

Richardson, M. and Domingos, P. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, 2006.

Sheng, E., Chang, K.-W., Natarajan, P., and Peng, N. Societal biases in language generation: Progress and challenges. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4275–4293, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.330.

Sugiyama, M., Bao, H., Ishida, T., Lu, N., Sakai, T., and Niu, G. *Machine Learning from Weak Supervision: An Empirical Risk Minimization Approach*. Adaptive Computation and Machine Learning series. MIT Press, 2022. ISBN 9780262047074.

van Krieken, E., Thanapalasingam, T., Tomczak, J., van Harmelen, F., and Ten Teije, A. A-NeSI: A scalable approximate method for probabilistic neurosymbolic inference. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 24586–24609. Curran Associates, Inc., 2023.

Zhang, H., Dang, M., Peng, N., and den Broeck, G. V. Tractable control for autoregressive language generation, 2023.

Zhang, T. Statistical analysis of some multi-category large margin classification methods. *J. Mach. Learn. Res.*, 5: 1225–1251, dec 2004. ISSN 1532-4435.

Zhu, H., Xiong, Z., Magill, S., and Jagannathan, S. An inductive synthesis framework for verifiable reinforcement learning. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2019, pp. 686–701, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367127. doi: 10.1145/3314221.3314638.