Training Over-parameterized Models with Non-decomposable Objectives

Anonymous Author(s)
Affiliation
Address
email

Abstract

Many modern machine learning applications come with complex and nuanced design goals such as minimizing the worst-case error, satisfying a given precision or recall target, or enforcing group-fairness constraints. Popular techniques for optimizing such non-decomposable objectives reduce the problem into a sequence of cost-sensitive learning tasks, each of which is then solved by re-weighting the training loss with example-specific costs. We point out that the standard approach of re-weighting the loss to incorporate label costs can produce unsatisfactory results when used to train over-parameterized models. As a remedy, we propose new cost-sensitive losses that extend the classical idea of logit adjustment to handle more general cost matrices. Our losses are calibrated, and can be further improved with *distilled* labels from a teacher model. Through experiments on benchmark image datasets, we showcase the effectiveness of our approach in training ResNet models with common robust and constrained optimization objectives.

1 Introduction

The misclassification error is the canonical performance measure in most treatments of classification problems [89]. While appealing in its simplicity, practical machine learning applications often come with more complex and nuanced design goals. For example, these may include minimizing the worst-case error across all classes [61], satisfying a given precision or recall target [103], enforcing minimal coverage for minority classes [30], or imposing group-fairness constraints [102]. Unlike the misclassification error, such objectives are *non-decomposable*, i.e., they cannot be expressed as a sum of losses over individual samples. This poses a non-trivial optimization challenge, which is typically addressed by reducing the problem into a sequence of cost-sensitive learning tasks [11, 25, 4, 15, 68]. Such reduction-based approaches have been successfully employed in many open-source libraries [3, 2, 1] to provide drop-in replacements for standard loss functions.

In this paper, we point out that the standard approach of re-weighting the training loss to incorporate label costs can produce unsatisfactory results when used with high capacity models, which are often trained to memorize the training data. Such *over-parameterized models* are frequently encountered in the use of modern neural networks, and have been the subject of considerable recent study [104, 70, 63]. As a remedy, we provide new calibrated losses for cost-sensitive learning that are better equipped at training over-parameterized models to optimize non-decomposable metrics, and demonstrate their effectiveness on benchmark image classification tasks. Our main contributions are as follows:

(i) we illustrate the pitfalls of using loss re-weighting in over-parameterized settings, particularly with diagonal class weights (Section 3).

(ii) we propose new logit-adjusted losses for cost-sensitive learning for both diagonal and non-diagonal gain matrices, and show that they are *calibrated* (Section 4).

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

(iii) we demonstrate that our losses provide significant gains over loss re-weighting in training ResNet models to optimize worst-case recall and to enforce coverage constraints (Section 5).

(iv) we show that the proposed approach compares favorably to post-hoc correction strategies, and can be further improved by *distilling* with a novel loss that we provide (Section 6).

1.1 Related Work

We cover prior work on class-imbalanced learning, cost-sensitive learning, and complex metrics.

Learning under class imbalance. A common setting where metrics beyond the misclassification error have been studied is the problem of class imbalance [47, 10, 32]. Here, the label distribution $\mathbf{P}(y)$ is skewed, and one seeks to ensure that performance on rare classes do not overly suffer. To this end, common metrics include the average of per-class recalls (also known as the balanced accuracy) [6, 59], quadratic mean of per-class accuracies [49], and the F-score [52].

A panoply of different techniques have been explored for this problem, spurred in part by recent interest in the specific context of neural networks (where the problem is termed *long-tail learning*) [88, 8], with the focus largely on optimizing balanced accuracy. An exhaustive survey is beyond the scope of this paper (see, e.g,. [32, 38]), but we may roughly identify three main strands of work: data modification [46, 10, 90, 100, 105], loss modification [104, 17, 9, 85, 36, 76, 94, 60, 21, 43, 91], and prediction modification [27, 41, 73, 57, 12, 39, 107]. A related recent thread seeks to improve performance on rare *subgroups* [79, 80, 83], as a means to ensure model fairness [24].

Amongst loss modification techniques, two strategies are of particular relevance. The first is *re-weighting* techniques [97, 62, 17]. As has been noted [53, 93, 58] (and as we shall subsequently verify), these may have limited effect on training samples that are perfectly separable under a given model class (as is the case with overparameterized models). The second is *margin* techniques, which enforce a class-dependent margin in the loss [9, 85, 76, 60, 91]. This seeks to ensure that rare classes are separated with greater confidence, to account for the higher uncertainty in their decision boundary.

Cost-sensitive learning. Cost-sensitive learning is a classical extension of standard multiclass classification, wherein errors on different classes incur different costs [26, 23, 82]. Strategies for this problem largely mirror those for class imbalance, which can be seen as imposing a particular set of costs dependent on the label frequencies [56]. In particular, loss modification techniques based on asymmetric weights [55, 95, 101, 5, 19, 22, 82, 109] and margins [58, 42, 35, 43] have been explored, with the latter proving useful on separable problems. Amongst these, Lin et al. [55] handle a general cost matrix for multiclass problems, but require the class scores to sum to 0, a constraint that might be difficult to impose with neural networks. Khan et al. [42] propose a multiclass loss similar to the logit adjustment idea used in this paper, but only handle a specific type of cost matrix (e.g., in their setup, the default cost matrix is a matrix of all 1s). The standard multiclass loss of Crammer and Singer [16] can also be extended to handle a general cost matrix [87], but unfortunately is not calibrated [75].

Complex metrics. There has been much work on extending the class imbalance literature to handle more complex metrics and to impose data-dependent constraints. These methods can again be divided into loss modifications [37, 71, 40, 66, 30, 25, 4, 15, 68, 48] and post-hoc prediction modifications [99, 65, 44, 67, 69, 31, 20, 64, 98, 86], and differ in how they decompose the problem into simpler cost-sensitive formulations (see [68] for a unified treatment of common reduction strategies). Most of these papers experiment with simple models, with the exception of Sanyal et al. [81], who use re-weighting strategies to train shallow DNNs, Kumar et al. [48], who train CNNs on binary-labeled problems, and Eban et al. [25], who train InceptionNets to optimize an AUC-based ranking metric. There have also been other attempts at training neural networks with ranking metrics [84, 34, 7, 74]. In contrast, our focus is on handling a general family of metrics popular in *multiclass* problems.

2 Non-decomposable Objectives

Notations. Let $[m] = \{1, \ldots, m\}$. Let Δ_m be the $(m-1)$-dimensional simplex with m coordinates. Let $\mathbf{1}_m \in \mathbb{R}^m$ denote an all 1s vector of size m, and $\mathrm{diag}(u_1, \ldots, u_m)$ denote a $m \times m$ diagonal matrix with diagonal elements $u_1, \ldots, u_m$. For vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$, $\mathbf{a}/\mathbf{b}$ denotes element-wise division. Let $\mathrm{softmax}_i(\mathbf{s}) = \frac{\exp(s_i)}{\sum_{j=1}^m \exp(s_j)}$ denote a softmax transformation of scores $\mathbf{s} \in \mathbb{R}^m$.

Consider a multiclass problem with an instance space $\mathcal{X} \subseteq \mathbb{R}^d$ and a label space $\mathcal{Y} = [m]$. Let D denote the underlying data distribution over $\mathcal{X} \times [m]$, $D_{\mathcal{X}}$ denote the marginal distribution over the

instances $\mathcal{X}$, and $\pi_i = \mathbf{P}(y = i)$ denote the class priors. We will use $p_i(x) = \mathbf{P}(y = i|x)$ to denote the conditional-class probability for instance x. Our goal is to learn a classifier $h : \mathcal{X} \to [m]$, and will measure its performance in terms of its confusion matrix $\mathbf{C}[h]$, where

$$C_{ij}[h] = \mathbf{E}_{(x,y)\sim D}\left[\mathbf{1}\left(y = i,\, h(x) = j\right)\right].$$

Complex learning problems. In the standard setup, one is often interested in maximizing the overall classification accuracy $\text{acc}[h] = \sum_i C_{ii}[h]$. However, in many practical settings, one may care about other metrics such as the recall for class i, $\text{rec}_i[h] = \frac{C_{ii}[h]}{\pi_i}$, the precision on class i, $\text{prec}_i[h] = \frac{C_{ii}[h]}{\sum_j C_{ji}}$, or the proportion of predictions made on class i, i.e. the coverage, $\text{cov}_i[h] = \sum_j C_{ji}$. Below are common examples of real-world design goals based on these metrics.

Example 1 (Maximizing worst-case recall). In class-imbalanced settings, where the classifier tends to generalize poorly on the rare classes, we may wish to change the training objective to directly maximize the *minimum recall* across all classes [11]:

$$\max_h \min_{i\in[m]} \frac{C_{ii}[h]}{\pi_i}. \tag{1}$$

Example 2 (Constraining per-class coverage). Another consequence of label imbalance could be that the proportion of predictions that the classifier makes for the tail classes is lower than the actual prevalence of that class. The following learning problem seeks to maximize the average recall while explicitly constraining the classifier's coverage for class j to be at least 95% of its prior π_j [15, 68]:

$$\max_h \frac{1}{m}\sum_{i=1}^{m} \frac{C_{ii}[h]}{\pi_i} \quad \text{s.t.} \quad \sum_{i=1}^{m} C_{ij}[h] \geq 0.95 \times \pi_j, \forall j \in [m]. \tag{2}$$

Other examples include constraints on recall and precision (see Table 1), as well as fairness constraints like equal opportunity, which are computed on group-specific confusion matrices [14, 31].

Reduction to cost-sensitive learning. All the problems described above are *non-decomposable*, in the sense that, they cannot be written as minimization of a sum of errors on individual examples, and hence cannot be tackled using common off-the-shelf learning algorithms. The dominant approach for solving such problems is to formulate a sequence of cost-sensitive objectives, which do take the form of an average of training errors [67, 11, 4, 15]. For example, the robust learning problem in (1) can be equivalently re-written as the following saddle-point optimization problem:

$$\max_h \min_{\boldsymbol{\lambda}\in\Delta_m} \sum_{i=1}^{m} \lambda_i \frac{C_{ii}[h]}{\pi_i}, \tag{3}$$

where λ_i is a multiplier for class i. One can then find a saddle-point for (3) by jointly minimizing the weighted objective over $\boldsymbol{\lambda} \in \Delta_m$ (using e.g. exponentiated-gradient updates) and maximizing the objective over h. Notice that for a fixed $\boldsymbol{\lambda}$, the optimization over h is a cost-sensitive (or a gain-weighted) learning problem:

$$\max_h \sum_{i,j} G_{ij} C_{ij}[h], \tag{4}$$

where $G_{ii} = \frac{\lambda_i}{\pi_i}$ and $G_{ij} = 0, \forall i \neq j$ are the rewards associated with predicting class j when the true class is i. We will refer to $\mathbf{G} \in \mathbb{R}^{m\times m}$ as the "gain" matrix, which in this case is *diagonal*. In practice, the cost-sensitive learning problem in (4) is usually replaced with multiple steps of gradient descent, interleaved with the updates on $\boldsymbol{\lambda}$.

Similar to (1), the constrained learning problem in (2) can be written as an equivalent (Lagrangian) min-max problem with Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^m_+$:

$$\max_h \min_{\boldsymbol{\lambda}\in\mathbb{R}^m_+} \frac{1}{m}\sum_{i=1}^{m} \frac{1}{\pi_i} C_{ii}[h] \,+\, \sum_{j=1}^{m} \lambda_j \Big(\sum_{i=1}^{m} C_{ij}[h] - 0.95\pi_j\Big).$$

One can find a saddle-point for this problem by jointly minimizing the Lagrangian over $\boldsymbol{\lambda} \in \mathbb{R}^m_+$ (using e.g. gradient updates) and maximizing it over h, with the maximization over h taking the form of a cost-sensitive learning problem. In this case, the gain matrix $\mathbf{G}$ is non-diagonal and is given by

	Problem	Gain Matrix	Losses
1	$\max_h \min_y \mathrm{rec}_y[h]$	$\mathrm{diag}(\boldsymbol{\lambda}/\boldsymbol{\pi})$	ℓ^{LA}
2	$\max_h \mathrm{acc}[h]$ s.t. $\mathrm{rec}_y[h] \geq \tau, \forall y$	$\mathrm{diag}(\mathbf{1}_m + \boldsymbol{\lambda}/\boldsymbol{\pi})$	ℓ^{LA}
3	$\max_h \mathrm{acc}_y[h]$ s.t. $\mathrm{prec}_y[h] \geq \tau, \forall y$	$\mathrm{diag}(\mathbf{1}_m + \boldsymbol{\lambda}) - \tau \mathbf{1}_m \boldsymbol{\lambda}^\top$	$\ell^{\mathrm{hyb}}, \ell^{\mathrm{SMS}}$
4	$\max_h \sum_y \mathrm{rec}_y[h]$ s.t. $\mathrm{cov}_y[h] \geq 0.95\pi_y, \forall y$	$\mathrm{diag}(\mathbf{1}_m/\boldsymbol{\pi}) + \mathbf{1}_m \boldsymbol{\lambda}^\top$	$\ell^{\mathrm{hyb}}, \ell^{\mathrm{SMS}}$
5	$\max_h \sum_y \mathrm{rec}_y[h]$ s.t. $\mathrm{bcov}_y[h] \geq 0.95\frac{1}{m}, \forall y$	$\mathrm{diag}(\mathbf{1}_m/\boldsymbol{\pi}) + (\mathbf{1}_m/\boldsymbol{\pi})\boldsymbol{\lambda}^\top$	ℓ^{hyb}

Table 1: Examples of complex learning problems, the associated gain matrices $\mathbf{G}$, and the proposed losses that are applicable to the setting. We use τ to denote the minimum allowed recall/precision, and $\mathrm{bcov}_y[h] = \sum_{i=1}^m \frac{1}{\pi_i} C_{iy}[h]$ is the balanced coverage metric we use in our experiments.

$G_{ii} = \frac{1}{m\pi_i} + \lambda_i$, $\forall i$ and $G_{ij} = \lambda_j$, $\forall i \neq j$. See Table 1 for the form of the gain matrix for other common constraints. In Appendix A, we provide detailed descriptions of the algorithms discussed.

Cost-sensitive losses. A standard approach for solving the cost-sensitive learning problem in (4) is to use a surrogate loss function $\ell : [m] \times \mathbb{R}^m \to \mathbb{R}_+$ that takes a label y and a m-dimensional score $\mathbf{u} \in \mathbb{R}^m$, and outputs a real value $\ell(y, \mathbf{u})$. One would then *minimize* the expected loss $\mathcal{L}(\mathbf{s}) = \mathbf{E}_{x,y}\left[\ell(y, \mathbf{s}(x))\right]$ over a class of scoring function $\mathbf{s} : \mathcal{X} \to \mathbb{R}^m$ that map each instance to an m-dimensional score. The final classifier h^* can then be obtained from the learned scoring function $\mathbf{s}^*$ by taking an argmax of its predicted scores, i.e. by constructing $h^*(x) \in \mathrm{argmax}_{i \in [m]}\, s_i^*(x)$.

In practice, we are provided a finite sample $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn from D, and will seek to minimize the average loss on S, given by $\widehat{\mathcal{L}}(\mathbf{s}) = \frac{1}{|S|} \sum_{(x,y) \in S} \ell(y, \mathbf{s}(x))$. We will also assume access to a held-out validation sample $S^{\mathrm{val}} = \{(x_1, y_1), \ldots, (x_{n^{\mathrm{val}}}, y_{n^{\mathrm{val}}})\}$.

One common sanity check for a "good" loss function is to confirm that it is *classification calibrated* for the learning problem of interest, i.e. to check if in the large sample limit, optimizing the loss over all scoring functions $\mathbf{s} : \mathcal{X} \to \mathbb{R}^m$ would result in the Bayes-optimal classifier for the problem. For the cost-sensitive objective in (4), the Bayes-optimal classifier is given below [51].

Proposition 1. *The optimal classifier for* (4) *for a general gain matrix* $\mathbf{G} \in \mathbb{R}^{m \times m}$ *is of the form:*

$$h^*(x) \in \mathrm{argmax}_{y \in [m]} \sum_{i=1}^m G_{iy}\, p_i(x) \;=\; \mathrm{argmax}_{y \in [m]} (\mathbf{G}^\top \mathbf{p}(x))_y.$$

3 The Perils of Over-parameterization

We now point out the problems in applying the algorithms described in the previous section to over-parameterized models, focusing particularly on the loss used for cost-sensitive learning.

Limitations of loss re-weighting. One of the most common loss function for a diagonal gain matrix $\mathbf{G}$ is a simple re-scaling of the standard cross-entropy loss with the diagonal weights:

$$\ell^{\mathrm{wt}}(y, \mathbf{s}) \;=\; -G_{y,y} \log\left(\frac{\exp(s_y)}{\sum_j \exp(s_j)}\right). \tag{5}$$

The following is a natural extension of this re-weighted loss to a general gain matrix $\mathbf{G}$, used, for example, in constrained optimization libraries [3], and also for mitigating label noise [72]:

$$\ell^{\mathrm{wt}}(y, \mathbf{s}) \;=\; -\sum_{i=1}^m G_{y,i} \log\left(\frac{\exp(s_i)}{\sum_j \exp(s_j)}\right). \tag{6}$$

While this weighted loss is calibrated for $\mathbf{G}$ (see, e.g., [72]), it is often inadequate when training an over-parameterized model on a *finite* sample. This is evident when $\mathbf{G}$ is a diagonal matrix: such a model will usually memorize the training labels and achieve *zero* training loss for every class, *irrespective* of what we choose for the outer weighting. In such separable settings, it is unclear how the outer weighting will impact the model's out-of-sample performance on different classes.

For clarity, we provide a simple illustration in Figure 1, where we use a re-weighted loss to train ResNet-56 models on 10000 images from CIFAR-10 with different diagonal gain matrices $\mathbf{G}$. We assign a weight of ω to the first class "airplane", and a weight of 1 on all other classes (i.e. set $G_{1,1} = \omega$

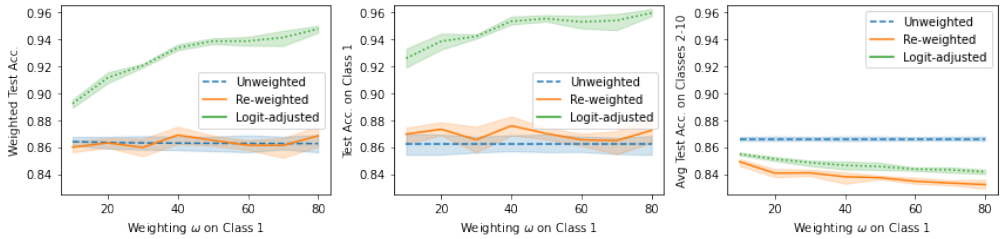

Figure 1: Training ResNet-56 on a subset of CIFAR-10 with an unweighted, re-weighted (cf. (5)) and logit-adjusted (cf. (7)) cross-entropy loss. All models achieve near-zero training error.

and $G_{y,y} = 1, \forall y \neq 1$), and plot the normalized weighted accuracy $\sum_y G_{yy} C_{yy}[h] / \sum_y G_{yy}$ on the test set as we vary ω. Compared to an unweighted cross-entropy loss, the re-weighted loss does not produce a significant change to the test metric, whereas the logit-adjusted loss (which we will discuss in Section 4) yields substantially better values. Probing closer into the test accuracy for each class, we see that the re-weighted loss yields slightly better accuracies for class 1, but is significantly worse-off on the other classes, suggesting that re-weighting has the effect of excessively focusing on the class with the higher weight at the cost of the other classes. Sagawa et al. [80] also make a similar observation when using importance weights to improve worst-group generalization, while Cui et al. [17], Cao et al. [9] observe that up-weighting the minority classes can lead to unstable optimization.

When $\mathbf{G}$ is not diagonal, the minimum training loss may not be zero. Nonetheless, the following proposition sheds some light on the scores learned by a model that achieves minimum training loss.

Proposition 2. *Let* $\widehat{\mathcal{L}}^{\text{wt}}(\mathbf{s}) = \frac{1}{|S|} \sum_{(x,y) \in S} \ell^{\text{wt}}(y, \mathbf{s}(x))$ *denote the average loss on training sample* S *with* ℓ^{wt} *defined as in* (6) *for a gain matrix* $\mathbf{G}$. *Then a scoring function* $\widehat{\mathbf{s}}$ *that achieves the minimum value of* $\widehat{\mathcal{L}}^{\text{wt}}(\mathbf{s})$ *over all* $\mathbf{s} : \mathcal{X} \to \mathbb{R}^m$ *is of the form:* $\text{softmax}_i(\widehat{\mathbf{s}}(x)) = \frac{G_{y,i}}{\sum_j G_{y,j}}, \forall (x,y) \in S$.

Notice that the model output is invariant to re-scaling of the rows in $\mathbf{G}$, i.e., one can multiply each row of the gain matrix $\mathbf{G}$ by a different scalar, and the values memorized by the model for each training example will remain unchanged. On the other hand, re-scaling a column of $\mathbf{G}$ does substantially change the score learned for each example. While this does not tell us about how the model would behave on unseen new examples, our experience has been that loss re-weighting is usually effective when used to control the amount of out-of-sample predictions that a model makes for a certain class (by suitably scaling the column for that class in $\mathbf{G}$), but does not work well when used to emphasize greater accuracy on a particular class. Consequently, we find that loss re-weighting usually fares better with metrics like coverage that depend only on the model predictions and not on the true labels, than with metrics like recall which depend on both the predictions and labels.

Existing remedies for better generalization. The literature offers some remedy to improve model generalization with non-standard objectives. Sagawa et al. [80] improve performance on rare sub-groups by regularizing the losses based on the size of each group, but their solution applies to a specific learning problem. Other remedies such as smarter re-weighting strategies [54, 17] or deferred re-weighting schedules [9] have proven to work well in specialized imbalanced settings.

The work that most closely relates to our setting is Cotter et al. [13], who propose a simple modification to the algorithms discussed in Section 2, with the use of two datasets: a training sample S, and a held-out validation sample S^{val}. They suggest using the validation sample to perform updates on the multipliers and in turn the gain matrix $\mathbf{G}$, and the training sample to solve the resulting cost-sensitive learning problem in (4). The latter would typically involve optimizing the empirical risk on the training set $\frac{1}{|S|} \sum_{(x,y) \in S} \ell_{\mathbf{G}}(y, \mathbf{s}(x))$, for some cost-sensitive loss $\ell_{\mathbf{G}}$. The intuition here is that, even when a model achieves very low training error, the estimate of $\mathbf{G}$ will accurately reflect the model's performance on held-out examples. While this modification improves our estimate of $\mathbf{G}$, it only provides a partial solution for over-parameterized settings, where the model would still struggle to generalize well if $\ell_{\mathbf{G}}$ happens to be the simple re-weighted loss in (6).

4 Cost-sensitive Losses Based on Logit Adjustment

We present new cost-sensitive losses that seek to avoid the problems mentioned above. We build on recent work on *logit adjustment*, shown previously to be effective in long tail settings [76, 60, 91].

Diagonal gain matrix. When the gain matrix $\mathbf{G}$ is diagonal, Proposition 1 tells us that the Bayes-optimal classifier for the resulting weighted accuracy metric is of the form $h^*(x) \in \operatorname{argmax}_{y \in [m]} G_{yy}\, p_y(x)$, where $p_y(x) = \mathbf{P}(y|x)$. Intuitively, we would like the learned scoring function $\mathbf{s}(x)$ to mimic the Bayes-optimal scores $G_{yy}\, p_y(x)$ for each x. In particular, we would like the maximizer of $s_y(x)$ over labels y to match the Bayes-optimal label for x.

One way to facilitate this is to adjust the scores $s_y(x)$ based on the diagonal weights G_{yy}, and to compute a loss on the adjusted scores. Specifically, we shift $s_y(x)$ to $\bar{s}_y(x) = s_y(x) - \log(G_{yy})$, and optimize the shifted scores so that their softmax transformation matches the class probabilities $\mathbf{p}(x)$. This would then encourage the original scorer $\mathbf{s}$ to be monotonic in the Bayes optimal scores:

$$\frac{\exp(\bar{s}_y(x))}{\sum_j \exp(\bar{s}_j(x))} = p_y(x) \iff \bar{s}_y(x) \propto \ln(p_y(x)) \iff s_y(x) \propto \ln(G_{yy}\, p_y(x)).$$

In practice, $\mathbf{p}(x)$ is not directly available to us, and so we employ the following logit-adjusted cross-entropy loss that implements this idea with labels y drawn according to $\mathbf{p}(x)$:

$$\ell^{\text{LA}}(y, \mathbf{s}) = -\log\left(\frac{\exp(s_y - \log(G_{yy}))}{\sum_j \exp(s_j - \log(G_{jj}))}\right). \tag{7}$$

This loss is a simple generalization of the one analyzed in Ren et al. [76], Menon et al. [60], Wang et al. [91], in which the logits are adjusted based on the class priors to optimize the balanced error rate. Such approaches have historical precedent in the class imbalance literature [73, 108, 12].

Proposition 3. *The logit-adjusted loss ℓ^{LA} is calibrated for a diagonal gain matrix $\mathbf{G}$.*

Unlike the weighted loss in (5), changing the diagonal entries of $\mathbf{G}$ in (7) changes the *operating point* of the learned scoring function. In separable settings, this would mean that the diagonal weights will determine the operating point at which the model achieves zero training error, and tend to push the separator towards classes that have higher weights, thus helping improve model generalization. An alternate explanation posed by Menon et al. [60] re-writes the loss in a pair-wise form:

$$\ell^{\text{LA}}(y, \mathbf{s}) = \log\left(1 + \sum_{j \neq y} \exp\left(\delta_{yj} - (s_y - s_j)\right)\right), \tag{8}$$

where $\delta_{yj} = \log(G_{yy}/G_{jj})$ can be seen as the *relative margin* between class y and class j. This tells us that the loss encourages a larger separation between classes that have different weights. In Appendix C, we elaborate on this margin interpretation and point out that this loss can in fact be seen as a soft-approximation to the more traditional margin-based loss of Crammer and Singer [16].

General gain matrix. When the gain matrix $\mathbf{G}$ is non-diagonal, a simple logit adjustment no longer works. In this case, we propose a hybrid approach that combines logit adjustment with an outer weighting. To this end, we prescribe factorizing the gain matrix into a product $\mathbf{G} = \mathbf{M}\mathbf{D}$ for some diagonal matrix $\mathbf{D} \in \mathbb{R}^{m \times m}$, and $\mathbf{M} = \mathbf{G}\mathbf{D}^{-1}$. The proposed loss then adjusts the logits to account for the diagonal entries of $\mathbf{D}$ and applies an outer weighting to account for $\mathbf{M}$:

$$\ell^{\text{hyb}}(y, \mathbf{s}) = -\sum_{i=1}^{m} M_{yi} \log\left(\frac{\exp(s_i - \log(D_{ii}))}{\sum_j \exp(s_j - \log(D_{jj}))}\right). \tag{9}$$

In practice, $\mathbf{D}$ can be chosen to reflect the relative importance of the classes and include information such as the class priors, which cannot be effectively incorporated as part of the outer weighting. One simple choice could be $\mathbf{D} = \operatorname{diag}(1/\pi_1, \ldots, 1/\pi_m)$. Another intuitive choice could be to $\mathbf{D} = \operatorname{diag}(G_{11}, \ldots, G_{mm})$, so that the residual matrix $\mathbf{M} = \mathbf{G}\mathbf{D}^{-1}$ that serves as the outer weighting has 1s in all its diagonal entries, and thus equal weights on the diagonal loss terms.

Proposition 4. *For any diagonal matrix $\mathbf{D} \in \mathbb{R}^{m \times m}$ with $D_{yy} > 0, \forall y$, and $\mathbf{M} = \mathbf{G}\mathbf{D}^{-1}$, the hybrid loss ℓ^{hyb} is calibrated for $\mathbf{G}$.*

In some special cases, we may be able to avoid the outer-weighting: e.g., when the gain matrix is the sum of a diagonal matrix and a matrix with equal rows, i.e. $\mathbf{G} = \operatorname{diag}(\boldsymbol{\alpha}) + \mathbf{1}\boldsymbol{\beta}^\top$ (cf. Table 1, rows 3 & 4), the Bayes-optimal classifier from Proposition 1 is of the form $h^*(x) = \operatorname{argmax}_{y \in [m]} \alpha_y p_y(x) + \beta_y$. The additive terms β_y in the optimal classifier can then be encoded in the loss as a shift to the softmax output, giving us the following *softmax-shifted* loss: for constant $C > 0$,

$$\ell^{\text{SMS}}(y, \mathbf{s}) = -\log\left(C \frac{\exp(s_y - \log(\alpha_y))}{\sum_j \exp(s_j - \log(\alpha_j))} - \beta_y/\alpha_y\right). \tag{10}$$

Method	CIFAR-10-LT		CIFAR-100-LT		TinyImgNet-LT	
	Avg Rec	Min Rec	Avg Rec	Min HT Rec	Avg Rec	Min HT Rec
ERM	0.754	0.589	0.444	0.076	0.319	0.010
LA with class priors	0.790	0.658	0.489	0.200	0.350	0.095
CSL [Re-weighted]	0.731	0.592	0.415	0.083	0.306	0.098
CSL [Logit-adjusted]	0.773	0.721	0.466	0.418	0.326	0.310

Table 2: Results of maximizing worst-case recall on CIFAR-10 and the minimum of the head and tail recalls on CIFAR-100 and Tiny-ImageNet. The highest entry in each column is shaded. The proposed logit-adjusted loss yields significantly better worst-case recall than all the baselines.

Method	CIFAR-10-LT		CIFAR-100-LT		TinyImgNet-LT	
	Avg Rec	Min Cov (tgt: 0.095)	Avg Rec	Min HT Cov (tgt: 0.010)	Avg Rec	Min HT Cov (tgt: 0.005)
ERM	0.754	0.058	0.444	0.001	0.319	0.000
LA with class priors	0.790	0.076	0.489	0.005	0.350	0.002
CSL [Re-weighted]	0.739	0.092	0.408	0.010	0.289	0.004
CSL [Hybrid A]	0.759	0.095	0.474	0.010	0.350	0.005
CSL [Hybrid B]	0.760	0.097	0.475	0.010	0.349	0.005

Table 3: Results of maximizing average recall, while constraining the coverage for each class to be at least 95% of $\frac{1}{m}$ on CIFAR-10, and constraining the head and tail coverages to be at least 95% of $\frac{1}{m}$ on CIFAR-100 and TinyImageNet. The models are evaluated on a balanced test set and hence we set the coverage target to $0.95\frac{1}{m}$. All metrics and targets are rounded off to 3 decimal places. The highest entry and those comparable to it (within 0.002) are shaded. The maximum recall among rows where the coverage is closest to the target is underlined. The proposed hybrid losses yield coverage values $\geq$ target, while achieving a higher average recall than the re-weighted loss.

Proposition 5. *ℓ^{SMS} is calibrated for the gain matrix $\mathbf{G} = \text{diag}(\boldsymbol{\alpha}) + \mathbf{1}\boldsymbol{\beta}^\top$ when $C = 1 + \sum_y \beta_y/\alpha_y$.*

See Appendix D for a practical variant this loss that avoids a negative value within the log.

5 Experimental Comparison with Loss Re-weighting

We now showcase that the proposed losses provide substantial gains over loss re-weighting through experiments on three benchmark datasets: CIFAR-10, CIFAR-100 [45], and TinyImageNet [50, 77] (a subset of the ImageNet dataset with 200 classes). Similar to [18, 9, 60], we use long-tail versions of these datasets by downsampling examples with an exponential decay in the per-class sizes. We set the imbalance ratio $\frac{\max_i \mathbf{P}(y=i)}{\min_i \mathbf{P}(y=i)}$ to 100 for CIFAR-10 and CIFAR-100, and to 83 for TinyImageNet (the slightly smaller ratio is to ensure that the smallest class is of a reasonable size). In each case, we use a balanced validation sample of 5000 held-out images, and a balanced test set of the same size. We trained ResNet-56 models on CIFAR-10 and CIFAR-100, and ResNet-18 models on TinyImageNet, using SGD with momentum. See Appendix E for details of the hyper-parameters used.

Cost-sensitive learning and baselines. We consider two learning objectives: maximizing worst-case recall, and maximizing average recall subject to coverage constraints. We employ the "two dataset" approach of Cotter et al. [13] discussed in Section 3 to solve these problems, with the validation set used for updates on the gain matrix $\mathbf{G}$, and the training set used for the cost-sensitive learning (CSL) step. Since our goal here is to demonstrate that the the proposed losses fair better at solving the CSL step than loss re-weighting, we use a large validation sample to get good estimates of the gain matrix $\mathbf{G}$. For the CIFAR datasets, we perform 32 SGD steps on the cost-sensitive loss for every update on $\mathbf{G}$, and for TinyImageNet, we perform 100 SGD steps for every update on $\mathbf{G}$. In each case, we compare the results with empirical risk minimization (ERM) with the standard cross-entropy loss, and as a representative method that seeks to maximize the average per-class recall (i.e. balanced accuracy), we include the logit-adjusted loss of Menon et al. [60] in which the adjustments are based on the class priors alone (LA with class priors).

Maximizing worst-case recall. In our first task, we maximize the worst-case recall over all 10 classes on CIFAR-10 (cf. (1)). Given the larger number of classes, this can be a very restrictive goal for CIFAR-100 and TinyImageNet, leading to poor overall performance. So for these datasets, we will consider a simpler goal, where we measure the average recall on the bottom 10% of the

Method	C10-LT Per-class Recall		C100-LT Per-class Recall		C100-LT Head-Tail Recall		TIN-LT Head-Tail Recall	
	Avg	**Min**	**Avg**	**Min**	**Avg**	**Min**	**Avg**	**Min**
ERM + PS	0.771	0.722	0.286	0.090	0.450	0.447	0.311	0.310
CSL [Logit-adjusted]	0.773	0.721	0.342	0.120	0.450	0.446	0.325	0.323
CSL [Logit-adjusted] + PS	0.755	0.708	0.315	0.120	0.457	0.451	0.331	0.319

Table 4: Improvements with post-shifting: Results of maximizing the minimum recall over *all classes* (col 1–2) and over just the head and tail classes (col 3–4). Proposed approach compares favorably to ERM + PS, and is the best at maximizing worst-case recall *over all 100 classes* on CIFAR-100.

Method	CIFAR-10-LT		CIFAR-100-LT	
	Avg Rec	**Min Rec**	**Avg Rec**	**Min HT Rec**
Distilled ERM	0.757	0.585	0.455	0.058
Distilled LA with teacher priors	0.817	0.708	0.509	0.248
Distilled CSL [Re-weighted]	0.736	0.571	0.451	0.082
Distilled CSL [Logit-adjusted]	0.781	0.734	0.462	0.458
Distilled CSL [Hybrid-distilled]	0.775	0.744	0.473	0.467

Table 5: Improvements with self-distillation: Performance of student model on maximizing worst-case recall on CIFAR-10 and the minimum of the head and tail recalls on CIFAR-100. Both the teacher and student use a ResNet-56 architecture. The highest entry in each column is shaded.

smallest-sized classes (the "tail" recall), and the average recall on the top 90% of the classes (the "head" recall), and seek to maximize the minimum of the head and tail recalls:

$$\max_h \min \Big\{ \frac{1}{|\mathcal{H}|} \sum_{y \in \mathcal{H}} \mathrm{rec}_y[h], \ \frac{1}{|\mathcal{T}|} \sum_{y \in \mathcal{T}} \mathrm{rec}_y[h] \Big\},$$

where $\mathcal{H} \subset [m]$ and $\mathcal{T} \subset [m]$ denote the set of head and tail labels respectively. The gain matrix $\mathbf{G}$ for these problems is diagonal. In Table 2, we present results of maximizing the worst-case recall metrics using the proposed logit-adjusted loss for a diagonal $\mathbf{G}$ in (7). Compared to loss re-weighting, the prescribed loss provides huge gains in the worst-case recalls. In fact, on the CIFAR datasets, loss re-weighting fairs significantly worse than using the simple logit adjustment with class priors, which as expected performs the best on the average recall. Our approach improves substantially over this baseline on the worst-case recall, at the cost of a moderate decrease in the average recall.

Constraining coverage. The next task we consider for CIFAR-10 seeks to ensure that when evaluated on a balanced dataset, the model makes the same proportion of predictions for each class. This leads us to the optimization problem shown in row 5 of Table 1, where we wish to maximize the average recall, constraining the model's "balanced coverage" on each class $\mathrm{bcov}_y[h] = \sum_{i=1}^m \frac{1}{\pi_i} C_{iy}[h]$ to be at least 95% of $\frac{1}{m}$, where m is the number of classes. Because the validation and test samples are already balanced, the model's coverage on these datasets is the same as its balanced coverage. For CIFAR-100 and TinyImageNet, we consider the simpler goal of maximizing the average recall over all classes, with constraints on the model's average coverage over the head labels, and its average coverage over the tail labels to be both at least 95% of $\frac{1}{m}$:

$$\max_h \frac{1}{m} \sum_{y \in [m]} \mathrm{rec}_y[h] \quad \text{s.t.} \quad \frac{1}{|\mathcal{H}|} \sum_{y \in \mathcal{H}} \mathrm{bcov}_y[h] \geq 0.95 \frac{1}{m}, \qquad \frac{1}{|\mathcal{T}|} \sum_{y \in \mathcal{T}} \mathrm{bcov}_y[h] \geq 0.95 \frac{1}{m}.$$

The gain matrix $\mathbf{G}$ for these problems (as shown in Table 1) is non-diagonal, and does not have a special structure that we can exploit. We will therefore use the hybrid loss function that we provide in (9) for a general $\mathbf{G}$, and try out both variants suggested: with the diagonal matrix $\mathbf{D} = \mathrm{diag}(1/\pi_1, \ldots, 1/\pi_m)$ (variant "A"), and with $\mathbf{D} = \mathrm{diag}(G_{11}, \ldots, G_{mm})$ (variant "B"). In this case, loss re-weighting is able to match the coverage targets for CIFAR-100 and TinyImageNet and comes in a close-second on CIFAR-10, but fairs poorly on the average recall. The proposed losses perform significantly better on this metric, while yielding coverage values that are closest to the target. Between the two hybrid variants, there isn't a clear winner.

6 Improvements with Post-shifting and Distillation

Having demonstrated that the proposed losses can provide significant gains over loss re-weighting, we explore ways to further improve their performance.

Does post-shifting provide further gains? Post-hoc correction strategies have generally shown to be very effective in optimizing evaluation metrics [65, 44, 98, 86, 60], often matching the performance of more direct methods that modify the training loss. They are implemented in two steps: (i) train a base scoring model $\mathbf{s} : \mathcal{X} \to \mathbb{R}^m$ using ERM, (ii) construct a classifier that estimates the Bayes-optimal label for a given x by applying a gain matrix $\mathbf{G} \in \mathbb{R}^{m \times m}$ to the predicted probabilities:

$$h(x) \in \operatorname{argmax}_{y \in [m]} \sum_{i=1}^{m} G_{iy} \eta_i(x), \quad \text{where} \quad \boldsymbol{\eta}(x) = \operatorname{softmax}(\mathbf{s}(x)).$$

The coefficients $\mathbf{G}$ are usually chosen to maximize the given evaluation metric on a held-out validation set, using either a simple grid search (when the number of classes is small) or more sophisticated optimization tools [67, 86]. Table 4 shows the results of post-shifting the ERM-trained model for the tasking of maximizing worst-case recall (see Appendix E for details of how we fit the post-shift coefficients). While post-shifting does considerably improve the performance of ERM, on the more difficult problem of maximizing the worst-case recall over *all 100 classes* in CIFAR-100, our proposed approach of modifying the training loss fairs significantly better on both the average and minimum recalls. In some cases, our model over-fits to the validation sample as a result of post-shifting.

New cost-sensitive loss for distillation. Another technique that has proven effective in boosting the performance of neural networks is *knowledge distillation*, wherein soft predictions from a "teacher" model $\mathbf{p}^t : \mathcal{X} \to \Delta_m$ are used as labels to train a "student" model [33, 78, 28, 96, 106]. All the losses discussed so far can be easily applied to a distillation setup, where the training labels can be replaced with an expectation over the teacher's label distribution: $\mathbf{E}_{x \sim D_{\mathcal{X}}} \left[\sum_{y=1}^{m} p_y^t(x) \ell(y, \mathbf{s}(x)) \right]$. In applying these losses for cost-sensitive learning, it is important that the class priors π_y used to construct the gain matrices (see Table 1) be replaced with the teacher's prior distribution $\pi_y^t = \frac{1}{|S|} \sum_{(x,y) \in S} p_y^t(x)$. This is particularly important when the teacher is trained on a dataset with a different prior distribution, or its outputs are re-calibrated to yield soft predictions.

We also provide an additional loss for distillation that seeks to better exploit the teacher predictions by encoding them as a part of the logit adjustment. This is an extension of the hybrid loss proposed in (9) for a general gain matrix $\mathbf{G}$, where we pick a diagonal matrix $\mathbf{D} \in \mathbb{R}^{m \times m}$, and compute the residual matrix $\mathbf{M} = \mathbf{G}\mathbf{D}^{-1}$. For given teacher labels $\mathbf{z} \in \Delta_m$ and student scores $\mathbf{s} \in \mathbb{R}^d$, the proposed loss takes the following form with a hyper-parameter $\gamma \in [0, 1]$:

$$\ell^{\text{dis}}(\mathbf{z}, \mathbf{s}) = -\sum_{y=1}^{m} \bar{z}_y^{1-\gamma} \log \left(\frac{\exp(s_y - \log(D_{yy}) - \gamma \log(\bar{z}_y))}{\sum_j \exp(s_j - \log(D_{jj}) - \gamma \log(\bar{z}_j))} \right), \text{ where } \bar{\mathbf{z}} = \mathbf{M}^\top \mathbf{z}. \quad (11)$$

The loss applies the matrix $\mathbf{M}$ to the teacher labels, and uses a portion of the transformed teacher scores (determined by the parameter γ) as an outer weighting, and uses the remainder to perform an additional adjustment to the logits. The calibration properties of this surrogate rely on the teacher labels mimicking the conditional-class probabilities $\mathbf{p}(x)$, and are discussed in Appendix F.

We re-run the worst-case recall experiments on CIFAR-10 and CIFAR-100 from Section 5 with distillation. We employ *self-distillation* and use the same ResNet-56 architecture for both the teacher and the student. We first train the teacher using ERM, and use its labels to train a student on the same dataset. For the hybrid-distilled loss in (11), we set $\mathbf{D}$ to a diagonal matrix of inverse teacher priors $1/\pi_1^1, \ldots, 1/\pi_m^t$ and pick the parameter γ from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ using the validation set. As shown in Table 5, distillation uniformly improves the performance for all methods, with the hybrid-distilled loss yielding the best worst-case recall.

In conclusion, we have proposed new cost-sensitive losses for training over-parameterized models with non-decomposable metrics, and have shown significant gains over standard loss re-weighting. Our approach compares favorably to post-shifting, and can be further improved with distillation. A limitation of our losses is that while they are calibrated, these guarantees only hold in the large sample limit. In the future, we wish to probe further into why our loses improve generalization even in finite sample settings, and to develop a more formal understanding of their margin properties. More broadly, when applied to objectives with fairness constraints, the techniques presented here could be used to improve performance of complex neural networks on under-represented samples. Carefully studying their empirical behaviour in such settings, to ensure they do not introduce unforeseen additional biases, is an important direction for future study.

References

[1] Fairlearn Library. URL https://fairlearn.org/.

[2] Global Objectives Library. URL https://git.dst.etit.tu-chemnitz.de/external/tf-models/-/tree/master/research/global_objectives.

[3] TensorFlow Constrained Optimization Library. URL https://github.com/google-research/tensorflow_constrained_optimization.

[4] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69, 2018.

[5] Francis R. Bach, David Heckerman, and Eric Horvitz. Considering cost asymmetry in learning classifiers. *Journal of Machine Learning Research*, 7(63):1713–1741, 2006. URL http://jmlr.org/papers/v7/bach06a.html.

[6] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, pages 3121–3124, 2010. doi: 10.1109/ICPR.2010.764.

[7] Sebastian Bruch, Masrour Zoghi, Mike Bendersky, and Marc Najork. Revisiting approximate metric optimization in the age of deep neural networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19)*, pages 1241–1244, 2019.

[8] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *arXiv:1710.05381 [cs, stat]*, October 2017.

[9] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019.

[10] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research (JAIR)*, 16:321–357, 2002.

[11] Robert S Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. In *Advances in Neural Information Processing Systems*, pages 4705–4714, 2017.

[12] Guillem Collell, Drazen Prelec, and Kaustubh R. Patil. Reviving threshold-moving: a simple plug-in bagging ensemble for binary and multiclass imbalanced data. *CoRR*, abs/1606.08698, 2016.

[13] Andrew Cotter, Maya Gupta, Heinrich Jiang, Nathan Srebro, Karthik Sridharan, Serena Wang, Blake Woodworth, and Seungil You. Training well-generalizing classifiers for fairness metrics and other data-dependent constraints. In *International Conference on Machine Learning*, pages 1397–1405. PMLR, 2019.

[14] Andrew Cotter, Maya Gupta, and Harikrishna Narasimhan. On making stochastic classifiers deterministic. In *Advances in Neural Information Processing Systems*, 2019.

[15] Andrew Cotter, Heinrich Jiang, Serena Wang, Taman Narayan, Seungil You, Karthik Sridharan, and Maya R. Gupta. Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. *Journal of Machine Learning Research (JMLR)*, 20 (172):1–59, 2019.

[16] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001.

[17] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019.

[18] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019.

[19] M.A. Davenport, R.G. Baraniuk, and C.D. Scott. Controlling false alarms with support vector machines. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5, pages V–V, 2006. doi: 10.1109/ICASSP.2006.1661344.

[20] Krzysztof Dembczyński, Wojciech Kotłowski, Oluwasanmi Koyejo, and Nagarajan Natarajan. Consistency analysis for binary classification revisited. In *International Conference on Machine Learning*, pages 961–969. PMLR, 2017.

[21] Zongyong Deng, Hao Liu, Yaoxing Wang, Chenyang Wang, Zekuan Yu, and Xuehong Sun. PML: progressive margin loss for long-tailed age classification. *CoRR*, abs/2103.02140, 2021. URL https://arxiv.org/abs/2103.02140.

[22] Jacek P. Dmochowski, Paul Sajda, and Lucas C. Parra. Maximum likelihood in cost-sensitive learning: Model specification, approximations, and upper bounds. *Journal of Machine Learning Research*, 11(108):3313–3332, 2010.

[23] Pedro Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, page 155–164, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 1581131437. doi: 10.1145/312129.312220. URL https://doi.org/10.1145/312129.312220.

[24] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science Conference (ITCS)*, pages 214–226, 2012.

[25] Elad Eban, Mariano Schain, Alan Mackey, Ariel Gordon, Ryan Rifkin, and Gal Elidan. Scalable learning of non-decomposable objectives. In *Artificial Intelligence and Statistics*, pages 832–840. PMLR, 2017.

[26] Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, page 973–978, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558608125.

[27] Tom Fawcett and Foster Provost. Combining data mining and machine learning for effective user profiling. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 8–13. AAAI Press, 1996.

[28] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018.

[29] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[30] Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael P Friedlander. Satisfying real-world goals with dataset constraints. In *Advances in Neural Information Processing Systems*, pages 2415–2423, 2016.

[31] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, pages 3315–3323, 2016.

[32] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[33] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

[34] Chen Huang, Shuangfei Zhai, Walter Talbott, Miguel Bautista Martin, Shih-Yu Sun, Carlos Guestrin, and Josh Susskind. Addressing the loss-metric mismatch with adaptive loss alignment. In *International Conference on Machine Learning*, pages 2891–2900. PMLR, 2019.

[35] Arya Iranmehr, Hamed Masnadi-Shirazi, and Nuno Vasconcelos. Cost-sensitive support vector machines. *Neurocomputing*, 343:50–64, 2019.

[36] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[37] Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, pages 377–384. ACM, 2005.

[38] Justin Johnson and Taghi Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6:27, 03 2019.

[39] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Eighth International Conference on Learning Representations (ICLR)*, 2020.

[40] Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. Online and stochastic gradient methods for non-decomposable loss functions. *Advances in Neural Information Processing Systems*, 2014.

[41] Grigoris Karakoulas and John Shawe-Taylor. Optimizing classifiers for imbalanced training sets. In *Proceedings of the 11th International Conference on Neural Information Processing Systems*, NIPS'98, page 253–259, Cambridge, MA, USA, 1998. MIT Press.

[42] Salman H. Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A. Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3573–3587, 2018. doi: 10.1109/TNNLS.2017.2732482.

[43] Ganesh Ramachandra Kini, Orestis Paraskevas, Samet Oymak, and Christos Thrampoulidis. Label-imbalanced and group-sensitive classification under overparameterization. *CoRR*, abs/2103.01550, 2021. URL https://arxiv.org/abs/2103.01550.

[44] Oluwasanmi O Koyejo, Nagarajan Natarajan, Pradeep K Ravikumar, and Inderjit S Dhillon. Consistent binary classification with generalized performance metrics. In *NIPS*, pages 2744–2752, 2014.

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

[46] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of the International Conference on Machine Learning (ICML)*, 1997.

[47] Miroslav Kubat, Robert Holte, and Stan Matwin. Learning when negative examples abound. In Maarten van Someren and Gerhard Widmer, editors, *Proceedings of the European Conference on Machine Learning (ECML)*, volume 1224 of *Lecture Notes in Computer Science*, pages 146–153. Springer Berlin Heidelberg, 1997. ISBN 978-3-540-62858-3.

[48] Abhishek Kumar, Harikrishna Narasimhan, and Andrew Cotter. Implicit rate-constrained optimization of non-decomposable objectives. In *International Conference on Machine Learning (ICML)*, 2021, to appear.

[49] Steve Lawrence, Ian Burns, Andrew D. Back, Ah Chung Tsoi, and C. Lee Giles. Neural network classification and prior class probabilities. In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, page 299–313, Berlin, Heidelberg, 1998. Springer-Verlag. ISBN 3540653112.

[50] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. CS 231N, 2015.

[51] Yoonkyung Lee, Yi Lin, and Grace Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81, 2004.

[52] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, page 3–12, Berlin, Heidelberg, 1994. Springer-Verlag. ISBN 038719889X.

[53] Yaoyong Li, Hugo Zaragoza, Ralf Herbrich, John Shawe-Taylor, and Jaz S. Kandola. The perceptron algorithm with uneven margins. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, page 379–386, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 1558608737.

[54] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[55] Yi Lin, Yoonkyung Lee, and Grace Wahba. Support vector machines for classification in nonstandard situations. *Mach. Learn.*, 46(1–3):191–202, March 2002. ISSN 0885-6125.

[56] Charles Ling and Victor Sheng. Cost-sensitive learning and the class imbalance problem. *Encyclopedia of Machine Learning*, 01 2010.

[57] Marcus A. Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *ICML 2003 Workshop on Learning from Imbalanced Datasets*, 2003.

[58] Hamed Masnadi-Shirazi and Nuno Vasconcelos. Risk minimization, probability elicitation, and cost-sensitive SVMs. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 759–766, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

[59] Aditya Menon, Harikrishna Narasimhan, Shivani Agarwal, and Sanjay Chawla. On the statistical consistency of algorithms for binary classification under class imbalance. In *International Conference on Machine Learning*, pages 603–611, 2013.

[60] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

[61] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, 2019.

[62] Katharina Morik, Peter Brockhausen, and Thorsten Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML)*, pages 268–277, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-612-2.

[63] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020.

[64] Harikrishna Narasimhan. Learning with complex loss functions and constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 1646–1654, 2018.

[65] Harikrishna Narasimhan, Rohit Vaish, and Shivani Agarwal. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *Advances in Neural Information Processing Systems*, pages 1493–1501, 2014.

[66] Harikrishna Narasimhan, Purushottam Kar, and Prateek Jain. Optimizing non-decomposable performance measures: A tale of two classes. In *International Conference on Machine Learning*, pages 199–208. PMLR, 2015.

[67] Harikrishna Narasimhan, Harish Ramaswamy, Aadirupa Saha, and Shivani Agarwal. Consistent multiclass algorithms for complex performance measures. In *ICML*, pages 2398–2407, 2015.

[68] Harikrishna Narasimhan, Andrew Cotter, and Maya Gupta. Optimizing generalized rate metrics with three players. In *Advances in Neural Information Processing Systems*, pages 10746–10757, 2019.

[69] Nagarajan Natarajan, Oluwasanmi Koyejo, Pradeep Ravikumar, and Inderjit Dhillon. Optimal classification with multivariate losses. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2016.

[70] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[71] Shameem Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. Optimizing f-measures by cost-sensitive classification. In *Advances in Neural Information Processing Systems*, pages 2123–2131, 2014.

[72] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017.

[73] Foster Provost. Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI-2000 Workshop on Imbalanced Data Sets*, 2000.

[74] Qi Qi, Youzhi Luo, Zhao Xu, Shuiwang Ji, and Tianbao Yang. Stochastic optimization of area under precision-recall curve for deep learning with provable convergence. *CoRR*, abs/2104.08736, 2021. URL https://arxiv.org/abs/2104.08736.

[75] Harish G Ramaswamy and Shivani Agarwal. Convex calibration dimension for multiclass loss matrices. *The Journal of Machine Learning Research*, 17(1):397–441, 2016.

[76] Jiawei Ren, Cunjun Yu, shunan sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4175–4186. Curran Associates, Inc., 2020.

[77] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[78] Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gülcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. In *ICLR (Poster)*, 2016.

[79] S. Sagawa, A. Raghunathan, P. W. Koh, and P. Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning (ICML)*, 2020.

[80] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

[81] Amartya Sanyal, Pawan Kumar, Purushottam Kar, Sanjay Chawla, and Fabrizio Sebastiani. Optimizing non-decomposable measures with deep networks. *Machine Learning*, 107(8): 1597–1620, 2018.

[82] Clayton Scott. Calibrated asymmetric surrogate losses. *Electronic Journal of Statistics*, 6(none): 958 – 992, 2012. doi: 10.1214/12-EJS699. URL https://doi.org/10.1214/12-EJS699.

[83] N. Sohoni, J. Dunnmon, G. Angus, A. Gu, and C. Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[84] Yang Song, Alexander Schwing, Raquel Urtasun, et al. Training deep neural networks via direct loss minimization. In *International Conference on Machine Learning*, pages 2169–2177. PMLR, 2016.

[85] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, and J. Yan. Equalization loss for long-tailed object recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11659–11668, 2020.

[86] S. K. Tavker, H. G. Ramaswamy, and H. Narasimhan. Consistent plug-in classifiers for complex objectives and constraints. In *Advances in Neural Information Processing Systems*, 2020.

[87] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, Yasemin Altun, and Yoram Singer. Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, 6(9), 2005.

[88] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017.

[89] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[90] B.C. Wallace, K.Small, C.E. Brodley, and T.A. Trikalinos. Class imbalance, redux. In *Proc. ICDM*, 2011.

[91] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[92] Robert C Williamson, Elodie Vernet, and Mark D Reid. Composite multiclass losses. *Journal of Machine Learning Research*, 17:1–52, 2016.

[93] Shan-Hung Wu, Keng-Pei Lin, Chung-Min Chen, and Ming-Syan Chen. Asymmetric support vector machines: Low false-positive learning under the user tolerance. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 749–757, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581934.

[94] Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. Distribution-balanced loss for multi-label classification in long-tailed datasets. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 162–178, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58548-8.

[95] Xiaoyun Wu and Rohini Srihari. New ν-support vector machines and their sequential minimal optimization. In *AAAI*, pages 824–831, 01 2003.

[96] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.

[97] Yu Xie and Charles F. Manski. The logit model and response-based samples. *Sociological Methods & Research*, 17(3):283–302, 1989.

[98] Bowei Yan, Sanmi Koyejo, Kai Zhong, and Pradeep Ravikumar. Binary classification with karmic, threshold-quasi-concave metrics. In *International Conference on Machine Learning*, pages 5531–5540. PMLR, 2018.

[99] Nan Ye, Kian Ming Chai, Wee Sun Lee, and Hai Leong Chieu. Optimizing f-measures: a tale of two approaches. In *Proceedings of the 29th International Conference on Machine Learning*, pages 289–296. Omnipress, 2012.

[100] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for face recognition with under-represented data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[101] B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Third IEEE International Conference on Data Mining*, pages 435–442, 2003. doi: 10.1109/ICDM.2003.1250950.

[102] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970, 2017.

[103] Ao Zhang, Nan Li, Jian Pu, Jun Wang, Junchi Yan, and Hongyuan Zha. Tau-fpl: Tolerance-constrained learning in linear time. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pages 4398–4405. AAAI Press, 2018.

[104] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[105] Junjie Zhang, Lingqiao Liu, Peng Wang, and Chunhua Shen. To balance or not to balance: A simple-yet-effective approach for learning with long-tailed distributions, 2019.

[106] Shaoyu Zhang, Chen Chen, Xiyuan Hu, and Silong Peng. Balanced knowledge distillation for long-tailed learning. *CoRR*, abs/2104.10510, 2021. URL https://arxiv.org/abs/2104.10510.

[107] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *CVPR*, 2021.

[108] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 18(1), 2006.

[109] Zhi-Hua Zhou and Xu-Ying Liu. On multi-class cost-sensitive learning. *Computational Intelligence*, 26(3):232–257, 2010.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See last para of Section 6
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See last para of Section 6
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 4
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix B
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Unfortunately, due to internal library dependencies, we are unable to provide code at this stage. We'll provide a self-contained codebase with the final version of the paper.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix E
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Unfortunately, owing to the non-trivial computational requirements of training on large-scale image datasets (particularly in distillation setups), it was prohibitive to run all methods for multiple trials. Thus, for consistency, our reported results in Sections 5–6 are limited to one trial. However, as a sanity check, we did perform multiple runs of the baseline method, and find that the gains shown by our method are significantly greater than the variance across trials. For example, in the illustrative experiments in Section 3, where we do average our results over 5 trials and report error bars, one can see that the gains our proposed approach offers are substantially greater than the variance.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix E
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 5
   (b) Did you mention the license of the assets? [N/A] The datasets we experiment with are widely used in the literature and have their licences publicly available.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The data we use has no personally identifiable information or offensive content
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]