Mohammadnia Qaraei, Mohammadreza; Schultheis, Erik; Gupta, Priyanshu; Babbar, Rohit

Convex Surrogates for Unbiased Loss Functions in Extreme Classification With Missing Labels

Published in:
The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021

DOI:
10.1145/3442381.3450139

Published: 19/04/2021

Document Version
Publisher's PDF, also known as Version of record

Convex Surrogates for Unbiased Loss Functions in Extreme Classification With Missing Labels

Mohammadreza Qaraei
mohammadreza.mohammadniaqaraei@aalto.fi
Aalto University
Finland, Helsinki

Erik Schultheis
erik.schultheis@aalto.fi
Aalto University
Finland, Helsinki

Priyanshu Gupta
guptap@iitk.ac.in
IIT, Kanpur
India

Rohit Babbar
firstname.lastname@aalto.fi
Aalto University
Helsinki, Finland

ABSTRACT

Extreme Classification (XC) refers to supervised learning where each training/test instance is labeled with small subset of relevant labels that are chosen from a large set of possible target labels. The framework of XC has been widely employed in web applications such as automatic labeling of web-encyclopedia, prediction of related searches, and recommendation systems.

While most state-of-the-art models in XC achieve high overall accuracy by performing well on the frequently occurring labels, they perform poorly on a large number of infrequent (tail) labels. This arises from two statistical challenges, (i) missing labels, as it is virtually impossible to manually assign *every* relevant label to an instance, and (ii) highly imbalanced data distribution where a large fraction of labels are tail labels. In this work, we consider common loss functions that decompose over labels, and calculate unbiased estimates that compensate missing labels according to Natarajan et al. [26]. This turns out to be disadvantageous from an optimization perspective, as important properties such as convexity and lower-boundedness are lost. To circumvent this problem, we use the fact that typical loss functions in XC are convex surrogates of the 0-1 loss, and thus propose to switch to convex surrogates of its unbiased version. These surrogates are further adapted to the label imbalance by combining with label-frequency-based rebalancing.

We show that the proposed loss functions can be easily incorporated into various different frameworks for extreme classification. This includes (i) linear classifiers, such as DiSMEC, on sparse input data representation, (ii) attention-based deep architecture, AttentionXML, learnt on dense Glove embeddings, and (iii) XLNet-based transformer model for extreme classification, APLC-XLNet. Our results demonstrate consistent improvements over the respective vanilla baseline models, on the propensity-scored metrics for precision and nDCG.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450139

CCS CONCEPTS

• **Computing methodologies → Supervised learning by classification**.

KEYWORDS

Extreme classification, Missing labels, Imbalanced classification, Loss functions

ACM Reference Format:
Mohammadreza Qaraei, Erik Schultheis, Priyanshu Gupta, and Rohit Babbar. 2021. Convex Surrogates for Unbiased Loss Functions in Extreme Classification With Missing Labels. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3442381.3450139

1 INTRODUCTION

Extreme Classification (XC) refers to supervised learning where each training/test instance is labeled with small subset of relevant labels that are chosen from a large set of possible target labels. Problems with an extremely large number of labels are common in various domains such as annotating large encyclopedia [12, 27], image-classification [13], and next word prediction [25]. Further, the framework of XC can be effectively leveraged to address learning problems arising in recommendation systems, web-advertising and prediction of related searches in a search engine [1, 17, 29]. For the case of recommendation systems, by learning from similar users' buying patterns, a small subset of relevant items from a large collection can be recommended. The same argument applies for the suggestion of *related searches* in a search engine, by learning from the browsing behavior of similar users, related searches relevant to a user can be displayed from an extremely large collection of possible search queries.

With diverse applications, designing machine learning algorithms to solve XC has become a key research challenge. From the computational aspect of the learning problem, building effective extreme classifiers is faced with a scaling challenge arising due to large number (up to several millions) of output labels, input training instances, and input features. Two properties of datasets in XC which pose further problems, (i) long-tail distribution of instances among labels, and (ii) missing labels, are discussed next.

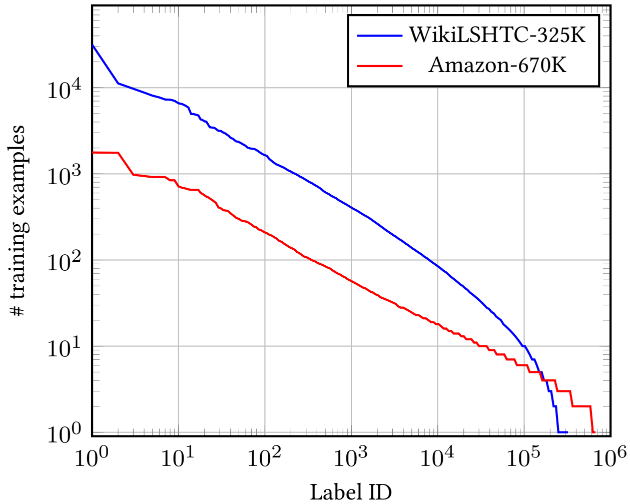

Figure 1: Label frequency in XC datasets. X-axis shows the label IDs sorted by the frequency of positive instances and Y-axis gives their number. For Amazon-670K, the power law holds very well, for WikiLSHTC-325K the decay in the far tail is a bit faster than the power law predicts.

1.1 Tail Labels

An important statistical feature of the datasets in XC is that a large fraction of labels are tail labels, i.e. those which have very few training instances. Typically, the label frequency distribution follows a power law, an example of which is shown in Figure 1 for the publicly available WikiLSHTC-325K and Amazon-670K datasets [5]. Concretely, let $n_{(r)}$ denote the number of occurrences of the r-th ranked label, when ranked in decreasing order of number of training instances that belong to that label, then $n_{(r)} \approx n_{(1)} r^{-\beta}$, where $\beta > 0$ denotes the exponent of the power law.

Tail labels exhibit diversity of the label space, and contain informative content not captured by the head or torso labels. Indeed, by predicting the head labels well, yet omitting most of the tail labels, an algorithm can achieve high accuracy [34]. Such behavior is not typically desirable in real-world applications [3]. In movie recommendation systems, for instance, the head labels correspond to popular blockbusters—most likely, the user has already watched these. However, the tail of the distribution corresponds to less popular yet equally favored films, like independent movies. These are the movies that the recommendation engine should ideally focus on [32]. A similar argument applies to search engine development [30] and hash-tag recommendation in social media [14]. However, effectively predicting tail-labels can be an enormous challenge due to the extreme data imbalance problem, where a given tail label appears in only a couple of (positive) instances and does not appear in millions of others (negatives).

1.2 Missing Labels

In addition to having unfavourable statistics, when learning to classify tail labels it has been shown that one also needs to account for missing labels in the training data [18]. In a dataset where the labels for each example are chosen from a label space with thousands of elements, it is impossible to explicitly check for the presence or absence of each label, so some examples will have missing labels. Even worse, the chance for a label to be missing is higher for tail labels than for head labels. In the movie example, this means that there are more people who would have liked an independent movie, but did not because never seeing it, than there are people who would have liked a blockbuster but never saw it. However, we can typically assume that most people who claim to like a movie actually do so, i.e. that we do not have significant amounts of spurious positive labels in the training set. This leads to the propensity model introduced in Jain et al. [18], formally presented in section 2.

They showed that certain loss functions used in XC allow for the calculation of an unbiased estimate if the available data has missing labels, and also proposed the unbiased variants of common metrics in extreme classification, called propensity scored metrics, for evaluation of XC models.

Although propensity-scored metrics have become ubiquitous in XC literature for unbiased evaluation of models, to the best of our knowledge, the use of unbiased loss functions for addressing the missing labels problem in XC has been limited to those losses given in Jain et al. [18]. However, several important loss functions, such as the binary cross-entropy (BCE) and hinge loss, were not covered by their analysis. A more general theory of how to treat class-conditional noisy labels (a generalization of the missing-labels setting) is provided in Natarajan et al. [26] for binary loss functions. As many multilabel losses (hinge, squared hinge, squared error, binary cross-entropy, Hamming), can be decomposed into a sum of binary contributions, this theory can also be used in the multilabel setting.

However, the unbiased estimates turn out to be disadvantageous from an optimization standpoint, as important properties of the original loss, such as convexity and lower-boundedness, are not necessarily preserved for the unbiased estimate, see Figure 2. The optimization problems have also been observed for learning with complementary labels [10] and positive-unlabeled learning [23].

We provide an alternative based on the following argument: For a loss that is a convex surrogate of the 0-1 loss, instead of taking its unbiased version, we construct the equivalent (in the sense of being equal up to a weighting factor) convex surrogate of the unbiased estimate of the 0-1 loss, so that the resulting new loss has the desired properties by construction. Up to scaling factors, this corresponds to the idea of optimizing surrogates of a weighted 0-1 loss as presented in Natarajan et al. [26]. The surrogates can be further combined with a reweighting to address the data imbalance. We show that these loss functions, in the form of appropriate weighting factors, can be readily incorporated in state-of-the algorithms for XC, and hence easily scale to datasets with hundreds of thousand labels. Empirically, the efficacy of the proposed loss functions is demonstrated by exhibiting superior performance to existing methods, with relative improvements of as much as 20% compared to some of the recently proposed state-of-the-art baselines in extreme classification.

1.3 Our Contribution

Despite the widespread use of propensity-scored metrics in evaluation and relative comparison of XC models, training efforts have been limited to PFastreXML [18]. We aim to close this gap by providing the following contributions:

- We derive unbiased variants of loss functions commonly employed in state-of-the-art XC baselines [2, 28, 38, 41]: The BCE loss and the (squared) hinge loss, which are convex surrogates of the 0-1 loss.
- The resulting unbiased estimates are problematic in practice as convexity and lower-boundedness properties are lost (Figure 2). Therefore we propose to use the corresponding convex surrogates of the unbiased 0-1 loss, which are more amenable to optimization.
- We further rebalance the loss functions to tackle the problem of extreme class-imbalance in XC datasets.
- We show that the proposed loss functions can be easily incorporated in state-of-the-art deep and shallow XC models, leading to significant improvements in terms of propensity-scored metrics.

2 THEORY

In the extreme classification setting, it is not possible for a human annotator to consider every possible label when deciding which labels to assign to a given data point. Instead, they will look at an example and assign a set of fitting labels that comes to mind. It is reasonable to assume that any label assigned in such fashion will be correct, i.e. if the annotator were asked whether the label belonged to the example, they would confirm this. The converse is not necessarily true: If one were to ask the annotator for each label that was not chosen whether it was relevant for the example, it is likely that some would be considered relevant.

To capture this effect, the notion of propensity is introduced. The propensity p of a label (for an example) is defined as the probability of the label being present, given that when explicitly asked, the ground-truth annotator would confirm it. For any given label j, we denote with $Y_j \in \{0,1\}$ whether the label is present in the annotated dataset, and with $Y_j^* \in \{0,1\}$ whether it should be present in the ground-truth. Formally, the propensity model described above can be specified as

$$\mathcal{P}\left\{Y = 1 | Y^* = 1\right\} =: p, \qquad \text{(missing labels)} \qquad (1)$$

$$\mathcal{P}\left\{Y = 1, Y^* = 0\right\} = 0. \qquad \text{(no spurious labels)} \qquad (2)$$

An empirical model for estimating propensities from label frequencies is given in [18]. They postulate that the propensity for a label j can be approximated by

$$p_j = (1 + C\exp(-A\log(n_j + B)))^{-1}, \qquad (3)$$

where A, B and $C = (\log N - 1)(B+1)^A$ are dataset dependent parameters, n_j denotes the number of positives for label j, and N is the number of training instances. This model has become standard in the community.

2.1 Unbiased Estimates

In the work of [18], the authors proposed to take into account the missing labels by replacing stochastic estimates of the form $l^*(Y)$ by unbiased estimates g s.t. $\mathbb{E}[g(Y)] = \mathbb{E}[l^*(Y^*)]$. They derived expressions for cases in which $y = 0$ implies $l^*(y, \hat{y}) = 0$ (e.g. P@k), as well as for the Hamming loss.

Loss	l_+^*	l_-^*
0-1 Loss	$\mathbb{I}[\hat{z} \leq 0]$	$\mathbb{I}[\hat{z} > 0]$
Hinge Loss	$\max(1-\hat{z}, 0)$	$\max(1+\hat{z}, 0)$
Sq. Hinge Loss	$\max(1-\hat{z}, 0)^2$	$\max(1+\hat{z}, 0)^2$
BCE	$-\log(\hat{y})$	$-\log(1-\hat{y})$

Table 1: Positive and negative parts of common losses.

A more general formulation is given in Natarajan et al. [26, Lemma 7], where unbiased losses for the binary classification setting are derived. This reduces to the missing labels scenario, relevant to our work, when the noise rates are $\rho_+ = (1-p)$ and $\rho_- = 0$. Under our propensity model, this is stated below in the form of the following corollary :

Corollary 1. *Let $l^* : \{0,1\} \times \mathbb{R} \longrightarrow \mathbb{R}$ be a function and define $l_+^* := l^*(1, \cdot)$ as well as $l_-^* := l^*(0, \cdot)$. Then the function $l : \{0,1\} \times \mathbb{R} \longrightarrow \mathbb{R}$ defined as*

$$l_+(\hat{y}) := p^{-1}\left(l_+^*(\hat{y}) + (p-1)l_-^*(\hat{y})\right) \qquad (4)$$

$$l(y, \hat{y}) := \begin{cases} l_+(\hat{y}) & y = 1 \\ l_-^*(\hat{y}) & y = 0 \end{cases} \qquad (5)$$

allows to calculate an unbiased estimate of l^*:

$$\mathbb{E}[l^*(Y^*, \hat{y})] = \mathbb{E}[l(Y, \hat{y})]. \qquad (6)$$

For an intuitive understanding of this result, consider that when we observe a label with propensity p, we know that in reality there are expected to be a total of $1/p$ instances with this label, so we have wrongly used the loss function l_-^* on $1/p - 1$ instances. Thus we should assign $l_+^* + (1/p - 1)(l_+^* - l_-^*)$ to the current instance to compensate for that, which is exactly what Equation 4 specifies.

By linearity, the result can also be used for any multilabel loss function that decomposes over labels, and it suffices to calculate the unbiased estimator in the binary case. Some losses are more easily defined over a prediction space of $\{-1, 1\}$ using the quantity $z := 2y - 1$. We will use this notation when appropriate, and in that case define also $\hat{z} = 2\hat{y} - 1$. Below, we derive the unbiased estimates for the losses listed in Table 1.

0-1 Loss. The 0-1 loss is given by $l^*(z, \hat{z}) = \mathbb{I}[\hat{z} \leq 0, z > 0] + \mathbb{I}[\hat{z} > 0, z \leq 0]$, which results in the unbiased estimate

$$l_+(\hat{z}) = \begin{cases} 1/p & \hat{z} < 0 \\ 1 - 1/p & \hat{z} \geq 0 \end{cases} \qquad (7)$$

For optimization purposes, when a constant shift does not matter, the slightly simpler formulation

$$\tilde{l}_+(\hat{z}) = (2/p - 1)\mathbb{I}[\hat{z} \leq 0] \qquad (8)$$

can be used. Note that composing the binary 0-1 loss for multiple labels leads to the Hamming loss.

Hinge Loss. The hinge loss is $l^*(z, \hat{z}) = \max(1 - z\hat{z}, 0)$. Thus

$$l_+(\hat{z}) = p^{-1}(\max(1 - \hat{z}, 0) + (p-1)\max(1 + \hat{z}, 0)). \quad (9)$$

Therefore, using

$$\mathbb{I}\{y = 1\} = (z+1)/2, \qquad \mathbb{I}\{y = 0\} = (1-z)/2, \quad (10)$$

the re-weighted loss becomes (brown line, Figure 2)

$$l(z, \hat{z}) = \frac{1+z}{2}\frac{\max(1-\hat{z}, 0) + (p-1)\max(1+\hat{z}, 0)}{p} + \frac{1-z}{2}\max(1+\hat{z}, 0). \quad (11)$$

Binary Cross-Entropy. For the BCE loss, (4) gives

$$l_+(\hat{y}) = p^{-1}(-\log\hat{y} + (1-p)\log(1-\hat{y})), \quad (12)$$

which results in the unbiased BCE given by

$$l(y, \hat{y}) = -\frac{y}{p}\log\hat{y} + \frac{y(1-p) - p + py}{p}\log(1-\hat{y}) = -\frac{y}{p}\log\hat{y} - \left(1 - \frac{y}{p}\right)\log(1-\hat{y}) \quad (13)$$

This result also follows directly from the fact that the BCE loss is linear in y.

2.2 Surrogates of Reweighted 0-1 Loss

The examples above show that many desirable properties of the original loss functions, such as convexity and non-negativety, may not hold for the unbiased estimates (see Figure 2). Even more problematic, for hinge and BCE loss the result is not lower-bounded, making the corresponding optimization problem ill-defined.

Therefore, this section provides an alternative to the unbiased estimators based on using convex surrogates to the unbiased estimate of the 0-1 loss. First, we show that this is equivalent, up to constant factors, to the weighted 0-1 loss approach of Natarajan et al. [26, Thm. 16]: This theorem, for the task of optimizing accuracy with missing labels, suggests to optimize a surrogate to the α-weighted 0-1 loss

$$U_\alpha := (1-\alpha)\mathbb{I}[y = 1, \hat{y} \le 0] + \alpha\mathbb{I}[y = 0, \hat{y} > 0]. \quad (14)$$

Setting $\alpha = 0.5p$ corresponds to the missing labels setting. By rescaling such that the second coefficient becomes 1, we recover the shifted version of the unbiased 0-1 loss of (8)

$$(2-p)/p\mathbb{I}[y = 1, \hat{y} \le 0] + \mathbb{I}[y = 0, \hat{y} > 0]. \quad (15)$$

This suggests a simple strategy for dealing with missing labels when the objective function is a surrogate of the 0-1 loss: Multiply the l^*_+ part of the loss by $2/p - 1$. The same approach was used by Chou et al. [10] to improve training with complementary labels. They observed that the biased gradients resulting from the convex surrogate of the bias-corrected 0-1 loss were better aligned with the true gradients than the unbiased, but high-variance gradients from the unbiased estimate of a convex surrogate of the 0-1 loss.

For the squared hinge loss, this results in the following variation

$$l_+(\hat{z}) = \frac{2-p}{p}\max(1-\hat{z}, 0)^2. \quad (16)$$

The BCE loss can also be interpreted in this way, if we reparametrize it to act on unscaled logits instead of normalized probabilities, which turns the BCE into the logistic loss $l(z, \hat{z}) = \log(2)^{-1}\log(1 + \exp(-z\hat{z}))$. This is a surrogate for the 0-1 loss and the arguments above apply. It relates to BCE by

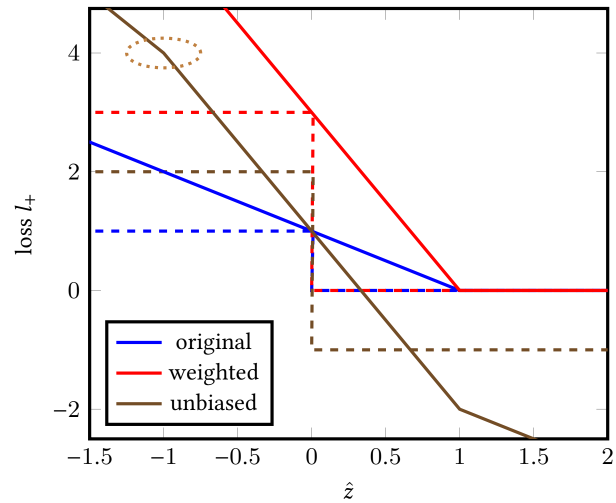

Figure 2: (Best viewed in Color) Visualization of 0-1-loss (dashed), hinge loss (solid) $y = 1$. The blue lines show the original loss functions without propensity re-weighting. The brown lines indicate the unbiased estimates for $p = 0.5$ (the dotted brown ellipse indicates the kink where the non-convexity appears). The red lines are the reweighted loss functions, i.e. the shifted 0-1 loss and the hinge loss upper bound.

$$l_{\log}(y, \hat{y}) = l_{\mathrm{BCE}}(y, \sigma(\hat{y})), \quad (17)$$

where σ is the logistics function.

2.3 Losses for Imbalanced Data

In the extreme setting, problems arise not only from missing labels, but also from the fact that most labels will be tail labels, that is the fraction of instances where this label is present will be very low. In such cases, even a trivial predictor that always predicts the absence of the label will get low loss values.

For a total of N examples, let $C^+(n, N)$ be the reweighting factor as a function of imbalance. For extreme classification, the imbalance becomes so large that weighting by inverse frequency becomes ineffective to achieve competitive performance. Instead, methods such as those based on *class-balanced weighting* of the loss [11] have been suggested.

There are two non-commuting ways of implementing this approach in the missing-labels case:

(1) Treat the optimization problem that has been corrected for missing labels as an imbalanced classification problem and apply reweighting to the loss function l.
(2) Treat the original problem as an imbalanced classification problem, i.e. re-weight l^*, and then correct for the missing labels. This is the approach discussed as *cost-sensitive* classification Natarajan et al. [26].

Note that, in the second strategy, one first needs to correct the true number of positive samples $n^* = n/p$ based on the propensity,

Dataset	# Training	# Test	# Labels	# Features	APpL	ALpP	A	B
EURLex-4K	15,539	3,809	**3993**	5,000	25.7	5.3	0.55	1.5
AmazonCat-13K	1,186,239	306,782	**13,330**	203,882	448.5	5.04	0.55	1.5
Wikipedia-31K	14,146	6,616	**30,938**	101,938	8.5	18.6	0.55	1.5
WikiLSHTC-325K	1,778,351	587,084	**325,056**	1,617,899	17.4	3.2	0.5	0.4
Wikipedia-500K	1,813,391	783,743	**501,070**	2,381,304	24.7	4.7	0.5	0.4
Amazon-670K	490,499	153,025	**670,091**	135,909	3.9	5.4	0.6	2.6

Table 2: The statistics of the multilabel datasets used in our experiments. APpL denotes the average points per label and ALpP is the average labels per point respectively. A **and** B **refer to the parameters of the propensity model.**

before applying the reweighting function. Thus the two strategies differ in whether one replaces $l_+ \leftarrow C^+(n,N)l_+$ or $l_+^* \leftarrow C^+(n/p,N)l_+^*$. This latter results in

$$l_+(\hat{y}) = p^{-1}\left(C^+(n/p,N)l_+^*(\hat{y}) - (1-p)l_-^*(\hat{y})\right). \tag{18}$$

For the 0-1 loss, the two variations are

$$l_+(\hat{z}) = C^+(n,N)(2/p-1)\mathbb{I}[\hat{z}<0] \quad \text{and} \tag{19}$$

$$l_+(\hat{z}) = (C^+(n/p,N)/p + 1/p - 1)\mathbb{I}[\hat{z}<0]. \tag{20}$$

Choosing to adapt according to the number of noisy labels, and class-balanced weighting with Cui et al. [11], the squared-hinge-based convex surrogate loss becomes

$$\frac{1-\beta}{1-\beta^n}\cdot\frac{2-p}{p}\cdot\max(1-\hat{z},0)^2. \tag{21}$$

where $\beta < 1$ is a hyperparameter usually close to 1. The same argument can be applied to the other convex surrogates for the unbiased loss functions discussed in the previous section.

3 EXPERIMENTAL SETUP

To show the practical applicability of the proposed losses, we incorporate them into a shallow and two deep state-of-the-art XC models. These are evaluated on multiple publicly available datasets from the extreme classification repository [5] that span three orders of magnitude in terms of their label set size, see Table 2. The data for the shallow method consists of sparse bag-of-words representation, whereas the deep methods are based on raw text input encoded using pretrained word embeddings. Despite these rather different methods and datasets, we observe significant improvements when replacing the respective training objectives with the corresponding rebalanced convex surrogates, as demonstrated by the experiments below.

For the shallow model, we apply the weighting scheme to `DiSMEC` [2], a one-vs-rest linear SVM with multilabel setting for XC. In `DiSMEC`, the weight vector w_j for each label j is learnt by minimizing a combination of squared hinge loss and l_2-regularization. Separating the squared hinge loss into the contributions from false negatives and false positives for label j, this is given by the following optimization problem:

$$\min_{w_j} \|w_j\|_2^2 + C_j^+ W_j^+ \sum_{i\in\mathcal{L}_j^+} \max(0, 1-(w_j^T x_i + b_j))^2 + \sum_{i\in\mathcal{L}_j^-} \max(0, 1+(w_j^T x_i + b_j))^2, \tag{22}$$

where $\mathcal{L}_j^+$ ($\mathcal{L}_j^-$) denotes the set of positive (negative) training samples corresponding to label j. The hyperparameters C_j^+ are the weighting factors to rebalance the classes, and W_j^+ are the factors we introduce to compensate for the missing labels. In the base `DiSMEC` model [2], these are all equal to 1.

To evaluate the proposed methods in deep learning models, we use the rebalanced convex surrogate BCE loss in `AttentionXML` [41] and `APLC-XLNet` [38], two state-of-the art approaches for deep extreme classification. `AttentionXML` employs a BiLSTM layer over pre-trained 300-dimensional word embeddings, followed by an attention layer. This model is minimizing the following BCE loss function:

$$l(\mathbf{y},\hat{\mathbf{y}}) = -\sum_{j=1}^{L} C_j^+ W_j^+ y_j \log \hat{y}_j + (1-y_j)\log(1-\hat{y}_j). \tag{23}$$

In You et al. [41] the parameters W_j^+ and C_j^+ are equal to 1.

`APLC-XLNet` fine-tunes XLNet [37], a pretrained transformer, on extreme classification datasets. To reduce the complexity of computing BCE in the large label space of XC datasets, `APLC-XLNet` partitions labels into a head and several tail clusters based on the frequency of the labels. Then the BCE loss is computed as Equation 23 with a slight difference that L does not comprise labels in tail clusters without any positive label, and $\hat{y}_j$ is computed by chain rule when label j belongs to a tail cluster (see Equations 2 and 5 of [38]). The same as the two other models, the hyperparameters W_j^+ and C_j^+ are equal to 1 in the ordinary `APLC-XLNet`.

We now use the convex surrogates of the bias corrected 0-1 loss (15) and corresponding formulations to handle data imbalance, developed in sections 2.2 and 2.3, to set the appropriate values for the weighting parameters W_j^+ and C_j^+. Firstly, the propensity weighted (**PW**) variant of squared hinge loss and BCE loss can be obtained by setting W_j^+ according to Equation 15. Secondly, as suggested in Equation 21, C_j^+ can be set based on the frequency of label j to rebalance the unbiased loss function (**PW-cb**) for better processing of imbalanced data.

Hence, in our experiments, we use the following two variants for the squared hinge loss (22) in `DiSMEC` and the BCE loss (23) in the deep models:

(1) **PW**: $W_j^+ = \frac{2}{p_j} - 1$.
(2) **PW-cb**: $W_j^+ = \frac{2}{p_j} - 1$, and $C_j^+ = \frac{1-\beta}{1-\beta^{n_j}}$ which is the class-balanced term introduced in [11]. We use $\beta = 0.9$ as we experimentally observed that larger values for β can improve

Loss Function	PSP@1	PSP@3	PSP@5	PnD@3	PnD@5	P@1	P@3	P@5	nD@3	nD@5	PS	vanilla
EURLex-4K												
Original	41.20	45.40	49.30	44.30	46.90	**82.40**	68.50	57.70	72.50	66.70	0.00	0.00
PW	43.05	47.39	50.58	46.19	48.37	82.17	**70.01**	**58.77**	**73.18**	**67.65**	3.89	**0.98**
PW-cb	**43.48**	**48.81**	**51.25**	**47.36**	**49.15**	82.25	68.80	57.18	72.26	66.32	**5.71**	−0.29
AmazonCat-13K												
Original	51.41	61.02	65.86	65.20	68.80	93.40	**79.1**	**64.1**	**87.7**	**85.8**	0.00	**0.00**
PW	61.58	68.99	73.11	**67.03**	**70.58**	93.43	78.74	63.91	87.45	85.54	11.50	−0.21
PW-cb	**64.95**	**71.35**	**74.37**	63.55	68.50	**93.54**	78.50	63.44	87.26	85.07	**13.26**	−0.47
Wikipedia-31K												
Original	13.60	13.10	13.80	13.20	13.60	85.20	74.60	65.90	77.10	70.40	0.00	0.00
PW	**14.9**	14.02	14.35	14.23	14.38	84.62	75.33	66.57	77.56	70.94	7.28	0.33
PW-cb	12.67	**15.87**	**18.28**	**15.05**	**16.76**	**85.77**	**78.17**	**68.53**	**80.08**	**72.96**	**12.86**	**2.94**
WikiLSHTC-325K												
Original	29.10	35.60	39.50	35.90	39.40	64.40	42.50	31.50	58.50	58.40	0.00	0.00
PW	34.24	37.22	40.78	38.44	41.57	64.60	**42.73**	**31.64**	58.83	58.74	9.28	0.46
PW-cb	**37.12**	**40.36**	**43.57**	**41.61**	**44.61**	**65.27**	42.68	31.48	**59.11**	**58.96**	**17.99**	**0.84**
Amazon-670K												
Original	27.80	30.60	34.20	28.80	30.70	**44.70**	**39.70**	**36.10**	**42.10**	**40.50**	0.00	**0.00**
PW	30.61	32.79	34.97	31.71	32.63	43.71	39.12	35.82	41.45	40.10	7.67	−1.53
PW-cb	**31.24**	**33.27**	**35.51**	**32.26**	**33.22**	41.89	37.81	34.92	40.04	39.02	**9.59**	−4.86
Wikipedia-500K												
Original	31.20	33.40	37.00	33.70	37.10	70.20	50.60	39.70	42.10	40.50	0.00	0.00
PW	**32.80**	**35.40**	**38.61**	**35.92**	**38.85**	**70.86**	**50.82**	**39.91**	**62.30**	**60.62**	**5.32**	**16.75**
PW-cb	30.32	31.56	33.52	31.83	33.88	66.38	45.69	34.85	56.62	54.12	−5.80	5.89

Table 3: Comparison of the original and the proposed reweighted variants of squared hinge loss in `DiSMEC` algorithm. For space reasons, we have abbreviated nDCG@k with nD@k and PSnDCG@k with PnD@k. The last two columns show the improvement relative to `DiSMEC`, averaged over propensity-scored and vanilla metrics respectively, as per (30). In almost all the cases, except PW-cb on Wikipedia-500K, the proposed methods improve propensity-scored metrics (PSP@k and PnD@k) without a significant drop in the vanilla metrics.

propensity scored metrics but lead to significant drop in vanilla metrics.

In the above methods, p_j is computed based on the empirical model of [18] as Equation 3.

For the very large labels spaces in Wikipedia-500K and Amazon-670K, a label tree has been used to speed up the computations of `AttentionXML`. The individual labels form the leaves in the tree, which are clustered under their parent nodes describing meta-labels. The non-leaf nodes are considered positives if *any* of their child nodes is positive, which means that correct calculation of the propensity of a meta-label does not result in the weighted average of the mean of its children, but needs to take into account the higher-order co-occurrence statistics. As a much simpler alternative, we opted to calculate the propensities of the meta-labels using the empirical propensity model (3) with the counts based on the number of instances belonging to the clusters. The computational advantage arises because only the descendants of positive nodes are evaluated. For the tree-based `AttentionXML` models, on the intermediate levels, n_j for cluster j required for computing Equation 3 is the number of training instances belonging to that cluster.

As the weighting factors may have large values, they can disrupt the learning process in deep learning models. We suspect that this is because in deep models we are no longer solving independent binary problems, but have to learn shared features in the hidden layers. As infrequent, low propensity labels get strongly upweighted by the PW losses, they can cause an increase in variance of the gradients that may hamper the learning of the shared features. A similar effect has been observed by Kang et al. [21], who noticed that for learning good representations, instance-balanced data is preferable to class-balanced data. However, such a separation is not possible in `AttentionXML`, because the last weights are shared across labels.

An approach that can stabilize the training in the deep learning based XC models is to prevent disproportionally large contributions from a single example, which we achieve by following [8] and normalizing the weighting factors in the deep models by

$$\eta_j \leftarrow \frac{\eta_j}{\sum_j \eta_j} \times L, \tag{24}$$

where η_j is W_j^+ in PW or $W_j^+ \times C_j^+$ in PW-cb. This rescaling with the same factor across all labels does not affect the relative contribution

Loss Function	PSP@1	PSP@3	PSP@5	PnD@3	PnD@5	P@1	P@3	P@5	nD@3	nD@5	PS	vanilla
EURLex-4K												
Original	44.80	51.66	54.54	50.10	51.99	**87.27**	**73.68**	**61.53**	**77.11**	**71.14**	0.00	**0.00**
PW	47.32	**53.62**	**55.84**	**51.90**	**53.47**	85.84	73.01	61.00	76.37	70.48	**3.98**	−1.16
PW-cb	**47.35**	52.97	55.51	51.43	53.17	84.73	72.38	60.97	75.63	70.19	3.44	−1.96
AmazonCat-13K												
Original	54.75	69.26	76.45	65.24	70.07	**96.11**	**82.51**	**67.3**	**91.46**	**89.49**	0.00	**0.00**
PW	57.89	71.95	77.63	68.15	72.05	95.09	81.11	66.14	89.86	88.10	4.03	−1.47
PW-cb	**61.35**	**73.95**	**78.78**	**70.58**	**73.91**	94.71	81.36	66.75	89.95	88.54	**7.93**	−1.31
Wikipedia-31K												
Original	15.88	17.12	18.11	16.54	17.55	**87.37**	**78.52**	**69.39**	**80.67**	**73.87**	0.00	**0.00**
PW	14.06	16.27	17.24	15.84	16.55	86.50	77.05	67.69	79.30	72.27	−7.10	−1.70
PW-cb	**20.82**	**20.66**	**20.84**	**20.68**	**20.79**	82.17	70.14	61.91	72.89	66.49	**23.58**	−8.83
Amazon-670K												
Original	30.36	33.74	37.12	32.98	35.12	47.65	42.53	38.83	45.12	**43.75**	0.00	0.00
PW	**31.32**	**34.62**	**37.55**	**33.76**	**35.76**	47.25	42.35	38.64	44.79	43.21	**2.38**	−0.76
PW-cb	30.22	33.91	37.23	32.96	35.21	**47.68**	**42.77**	**39.06**	**45.21**	43.63	0.01	**0.20**
Wikipedia-500K												
Original	30.85	39.14	44.22	36.79	39.79	76.80	58.42	46.03	69.87	68.06	0.00	0.00
PW	**34.59**	**42.04**	**46.59**	**39.86**	**42.61**	**77.23**	58.39	46.01	**70.19**	68.31	**8.74**	**0.31**
PW-cb	30.88	39.30	44.42	36.82	39.92	77.04	**58.47**	**46.19**	70.12	**68.32**	0.24	0.30

Table 4: Comparison of the original and the proposed reweighted variants of BCE loss in `AttentionXML` algorithm. The weighting factors of PW and PW-cb are normalized as per Equation 24. The columns are the same as in Table 3. The proposed losses improve propensity scored metrics in most of the cases, while the vanilla metrics are close to those of the original model.

of each label towards the loss, thus effectively downscaling the contribution of head and high-propensity labels as opposed to upscaling low-propensity tail labels.

It must also be noted that our proposed variants of loss functions do not lead to any significant computational overhead in terms of training and prediction over the base algorithms - `DiSMEC`, `AttentionXML`, and `APLC-XLNet`. Consequently, the resulting algorithms remains scalable to even larger datasets with millions of labels[1].

3.1 Evaluation metrics

With applications of XC arising in recommendation systems and web-advertising, the objective of an algorithm in this domain is to correctly recommend/advertise among the top-k slots. Thus, for evaluation of the methods, we use precision at k (P@k) and normalized discounted cumulative gain at k (nDCG@k), and their propensity scored variants. These are standard metrics in XC, which are defined below.

For each test sample with observed ground truth label vector $\mathbf{y} \in \{0,1\}^L$ and predicted vector $\hat{\mathbf{y}} \in \mathbb{R}^L$, propensity scored variants of P@k and nDCG@$k$are given by :

$$\text{PSP@}k(\mathbf{y}, \hat{\mathbf{y}}) \coloneqq \frac{1}{k} \sum_{\ell \in \text{top}_k(\hat{\mathbf{y}})} \frac{\mathbf{y}_\ell}{p_\ell} \tag{25}$$

[1]The codes for the experiments are available at: https://github.com/xmc-aalto/PWXMC

$$\text{PSnDCG@}k(\mathbf{y}, \hat{\mathbf{y}}) \coloneqq \frac{\text{PSDCG@}k}{\sum_{\ell=1}^{\min(k, \|\mathbf{y}\|_0)} \frac{1}{\log(\ell+1)}} \tag{26}$$

$$\text{PSDCG@}k(\mathbf{y}, \hat{\mathbf{y}}) \coloneqq \sum_{\ell \in \text{top}_k(\hat{\mathbf{y}})} \frac{\mathbf{y}_\ell}{p_\ell \log(\ell+1)}, \tag{27}$$

where $\text{top}_k(\hat{\mathbf{y}})$ returns the k largest indices of $\hat{\mathbf{y}}$. Setting $p_\ell = 1$ recovers the vanilla metrics.

To match against the best possible performance attainable by any system, as suggested in [18], we define, for M test samples,

$$\mathbb{G}(\{\hat{\mathbf{y}}\}) = -\frac{1}{M} \sum_{i=1}^{M} \mathbb{L}(\mathbf{y}_i, \hat{\mathbf{y}}_i), \tag{28}$$

where $\mathbb{L}(\cdot,\cdot)$ and $\mathbb{G}(\cdot)$ signify loss and gain respectively. We use

$$100 * \mathbb{G}(\{\hat{\mathbf{y}}\}) / \mathbb{G}(\{\mathbf{y}\}) \tag{29}$$

as the performance metric. The loss $\mathbb{L}(\cdot,\cdot)$ can take two forms, (i) $\mathbb{L}(\mathbf{y}_i, \hat{\mathbf{y}}_i) = -\text{PSnDCG@}k$, and (ii) $\mathbb{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\text{PSP@}k$. This leads to the metrics which are used in our comparison in Table 3 (denoted PSP@k and PnD@k), and evaluated for $k = 1, 3, 5$.

A collection of results from recent papers on datasets in Table 2 for algorithms developed over the last few years is given on the extreme classification repository [5].

There are two main reasons for using propensity scored metrics in XC. The first is theoretically grounded, and is that they provide (for an accurate propensity model) an unbiased estimate of the true loss even if the test data is missing labels. However, the propensity models used are typically only empirical approximations. As such,

Loss Function	PSP@1	PSP@3	PSP@5	PnD@3	PnD@5	P@1	P@3	P@5	nD@3	nD@5	PS	vanilla
EURLex-4K												
Original	42.17	49.77	52.86	47.72	49.87	**86.95**	**74.37**	**62.07**	**77.28**	**68.04**	0.00	**0.00**
PW	45.25	51.41	54.04	49.82	51.62	86.24	73.84	61.28	76.50	67.29	4.67	−0.96
PW-cb	**46.80**	**52.52**	**54.07**	**50.96**	**52.12**	86.06	73.24	60.61	76.19	66.68	**6.85**	−1.55
AmazonCat-13K												
Original	52.54	65.07	71.35	61.66	65.87	**94.58**	**79.77**	**64.58**	**83.28**	**71.9**	0.00	**0.00**
PW	**55.41**	**67.04**	**72.24**	**63.91**	**67.43**	93.89	79.14	64.23	82.62	71.45	**3.54**	−0.70
PW-cb	55.41	66.90	71.69	63.81	67.09	93.87	79.09	64.11	82.59	71.36	3.26	−0.78
Wikipedia-31K												
Original	14.84	15.85	16.99	15.58	16.36	**89.13**	**78.72**	**69.49**	**81.15**	**74.14**	0.00	**0.00**
PW	18.29	19.18	19.74	18.95	19.35	86.40	73.72	64.59	76.59	69.50	20.60	−5.23
PW-cb	**19.07**	**19.52**	**19.79**	**19.41**	**19.61**	83.36	67.91	57.98	71.44	63.62	**23.52**	−11.57
Amazon-670K												
Original	25.05	28.96	32.35	27.94	30.23	43.35	**38.72**	**35.18**	**39.77**	**37.04**	0.00	**0.00**
PW	27.02	29.81	32.22	29.09	30.73	**43.36**	38.57	34.82	39.66	36.79	4.01	−0.39
PW-cb	**28.69**	**30.92**	**32.83**	**30.35**	**31.65**	43.25	38.68	34.94	39.72	36.87	**8.44**	−0.31
Wikipedia-500K												
Original	29.83	35.24	38.31	33.68	35.55	**72.63**	**50.35**	**38.44**	**55.35**	**45.75**	0.00	**0.00**
PW	**31.89**	**36.86**	**39.70**	**35.42**	**37.15**	72.27	50.13	38.21	55.11	45.50	**5.28**	−0.50
PW-cb	31.84	35.12	37.04	34.18	35.34	71.68	48.53	36.62	53.71	44.05	1.79	−2.94

Table 5: Comparison of the original and the proposed reweighted variants of BCE loss in `APLC-XLNet` algorithm. The weighting factors of PW and PW-cb are normalized as per Equation 24. The columns are the same as in Table 3. The proposed variants of BCE consistently improve propensity scored metrics on all the datasets. For most of the datasets, the decrease in vanilla metrics is small.

the resulting values are not necessarily confined to the interval [0, 1], and thus renormalized versions (29) are being used [3, 6, 18]. This is fine for model comparison, but means that the reported metric is not an unbiased estimate of the true original metric. The second reason is empirical. Since the propensity model (3) implies that more weight is given to tail labels, the propensity scored metrics implicitly value the results in the tail more strongly, which is desirable for many applications.

Due to the shortcomings of PS metrics outlined above, and for consistency with previous results, we evaluate our algorithm with both PS and vanilla metrics. Our criterion for a successful algorithm is that it provides improved PS metric while at the same time only incurring minor decreases in vanilla metrics. We report, in addition to the individual metrics, also a summary over all vanilla metrics and their propensity scored counterparts as the last two columns. Since the different metrics may have significantly different scales, we do not report the average of their absolute values, but instead the mean of the relative changes to the non-propensity scored variation of the algorithm. Thus, these summaries are calculated over the set of metrics $\mathcal{M} = \{\text{P@}\{1,3,5\}, \text{nDCG@}\{1,3,5\}\}$ as

$$\text{mean(method)} = \frac{1}{6} \sum_{m \in \mathcal{M}} \frac{m(\text{method}) - m(\text{base})}{m(\text{base})}, \qquad (30)$$

with the analogous formulation for the propensity scored variation. Here "base" refers to the method without propensity weighting.

4 EXPERIMENTAL RESULTS

In this section, we discuss the results of applying the proposed variants of reweighted losses to the baselines. The goal is to improve propensity scored metrics, while vanilla metrics should not drop significantly. It should be noted that, since there is no raw text data available for WikiLSHTC-325K, the results of deep models are not presented for this dataset.

DiSMEC Results. The results for different variations of the `DiSMEC` algorithm are presented in Table 3. The main findings are:

- We can see that the variant based on propensity-weighting (PW based on equation (16)) improves the PS-metric results across all datasets (between 3.9% on Eurlex and 11.5% on AmazonCat), while not having much negative impact on the vanilla metrics (-1.5% on Amazon-670k up to +16.75% on Wikipedia-500k).
- Further improvements can be achieved on most datasets by choosing class-balanced weighting (PW-cb as given in Equation 21). For instance, except for Wikipedia-500K dataset, the relative improvement over `DiSMEC` range from 5.71% on Eurlex to 17.99% on WikiLSHTC dataset.

AttentionXML Results. The results for the propensity weighted variants of BCE loss used in `AttentionXML` are shown in Table 4. The main findings are:

- When applied to the standard `AttentionXML` architecture, the proposed variants of the BCE loss achieve significant

improvements over the baseline for the propensity scored variants of precision and nDCG. The corresponding changes are quite significant for Wikipedia-31K dataset, with an average increase of approximately 23% for propensity scored metrics.
- While on one dataset (Wikipedia-31K) the propensity weighted BCE falls behind the ordinary one in terms of PS metrics, PW-cb, which further rebalances the loss function, surpasses the ordinary BCE on all the datasets.

APLC-XLNet Results. Table 5 presents a comparison of the proposed variants of BCE with the ordinary one in `APLC-XLNet`. The main findings of the results are listed below:

- On all the datasets, the proposed methods consistently improve PS metrics, ranging from 1.79% on Wikipedia-500K to 23.52% on Wikipedia-31K.
- The same as the two other models, the improvements in propensity scored metrics comes at a slight degradation on vanilla metrics. In this regard, PW performs significantly better than the rebalanced variant.

These results show that adapting the loss function to take into account missing labels improves the top-k classification (in terms of PS metrics) for all three investigated models and across a wide range of datasets. Unfortunately, there is no clear trend as to whether class-balancing further improves the results. In some instances it does quite substantially, whereas in others it leads to worse results. When applying these methods to new datasets, it is therefore recommendable to test both approaches and see which one performs better. It may be noted that the normalization in Equation 24 is crucial for the deep models as the prediction performance without it yielded mixed results.

5 APPLICATION TO OTHER ALGORITHMS

Apart from the linear and deep non-linear models for extreme classification discussed in section 4, we mention below other approaches for extreme classification in which the proposed loss functions could be applied.

(1) *Sparse linear models* : `(P)PD-Sparse` [39, 40] algorithms exploit the sparsity in the primal and dual problem combined with elastic net regularization. `PD-Sparse` uses multi-class hinge loss while `PPD-Sparse` uses hinge loss for one-vs-rest style binary classification. Though not directly applicable to the multi-class hinge loss case in `PD-Sparse`, weighting the positive part of the loss function by $2/p - 1$ as in Section 2.2 is applicable to the binary loss function in `PPD-Sparse`.
`ProXML` [3] uses squared hinge loss and improves tail-label detection by posing the learning problem as an instance of robust optimization. It proposes to guard against small perturbations in the feature composition of the instances of the same class, leading to ℓ_1 regularization. As a future work, the regularization can be combined with the loss function form of Equation 16.

(2) *Deep learning* : Deeper architectures on top of word-embeddings have also been explored in recent works. A convolutional network based approach, `XML-CNN`, for deep extreme multi-label classification was proposed in [24]. Motivated by the success of AttentionXML for deep extreme classification, `X-Bert`, an approach based on pre-trained Bert language model ([15]) has been presented in the work [9]. It is expected that the convex surrogates for the BCE loss proposed in this paper are applicable to the settings in `XML-CNN` and `X-Bert`.

(3) *Label-tree methods* : In label-tree based methods, the labels or training instances are hierarchically partitioned into different groups. For instance, `Parabel` [28] partitions the labels into two balanced groups using 2-means leading to a the construction of a label-tree. More flexible partitioning is introduced in `Bonsai` [22] via k-means clustering with potential imbalance among the k clusters. Linear classifiers by optimizing squared hinge loss in a one-vs-rest manner are learnt at the internal and leaf nodes of the label trees. Hence the same technique as used in the label tree of `AttentionXML` (described in Section 3) can be applied for the label tree-based methods including [35] & recently proposed `NapkinXC` [20].

(4) *Negative Sampling based methods* : The primary goal of these algorithms [4, 17, 31] is to avoid computing the loss over all the samples which do not belong a given label, and hence speed up training without any significant loss in prediction accuracy. In particular, since the Slice algorithm [17] uses fixed representations learnt from XML-CNN model to train the classifier in the last layer with squared hinge loss, and hence the formulation in Equation 16 is applicable.

Apart from the class of methods mentioned above, label-embedding approaches assume that, despite the large number of labels, the label matrix is effectively low rank and therefore project it to a low-dimensional sub-space [19, 33, 42] . In some of the works, it was argued that the low rank embedding may be insufficient for capturing the label diversity in XMC settings ([7, 36]), which has been questioned in the recent work [16]. The loss functions developed in this work apply to the setting in which the loss function decomposes over labels such as in [42]. On the other hand, it is not directly applicable for non-decomposable scenarios such as [7, 16].

6 CONCLUSION

In order to improve classification in settings with an extremely large and imbalanced set of labels which might go missing, we analyzed unbiased loss functions which decompose over the individual labels. These include the popular hinge- and squared-hinge-loss as well as Hamming and binary cross-entropy. Even though we can calculate unbiased estimates of many common loss functions used in XC (all that can be decomposed into binary losses), the resulting optimization problem is often ill-defined and thus impractical. However, the theory of reweighted surrogates provides a way to circumvent this problem, and allows for combination with other techniques used in XC to alleviate the imbalance problem.

For the deep methods, in order to stabilize the learning, an additional rescaling as given in Equation 24 has to be introduced. Thus, we get a set of methods that address both missing and imbalanced labels and work with both shallow and deep models. As our experiments showed, these can be applied in practice and provide a noticeable boost in performance across a wide range of datasets.

ACKNOWLEDGMENTS

The authors wish to acknowledge CSC – IT Center for Science, Finland, as well as the Aalto Science-IT project, for computational resources.

REFERENCES

[1] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. 2013. Multi-Label Learning with Millions of Labels: Recommending Advertiser Bid Phrases for Web Pages. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) *(WWW '13)*. Association for Computing Machinery, New York, NY, USA, 13–24. https://doi.org/10.1145/2488388.2488391

[2] Rohit Babbar and Bernhard Schölkopf. 2017. DiSMEC: Distributed Sparse Machines for Extreme Multi-Label Classification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (Cambridge, United Kingdom) *(WSDM '17)*. Association for Computing Machinery, New York, NY, USA, 721–729. https://doi.org/10.1145/3018661.3018741

[3] Rohit Babbar and Bernhard Schölkopf. 2019. Data scarcity, robustness and extreme multi-label classification. *Machine Learning* 108, 8-9 (15 Sept. 2019), 1329–1351. https://doi.org/10.1007/s10994-019-05791-5

[4] Robert Bamler and Stephan Mandt. 2020. Extreme Classification via Adversarial Softmax Approximation. In *International Conference on Learning Representations*.

[5] K. Bhatia, K. Dahiya, H. Jain, A. Mittal, Y. Prabhu, and M. Varma. 2016. The extreme classification repository: Multi-label datasets and code. http://manikvarma.org/downloads/XC/XMLRepository.html

[6] Kush Bhatia, Kunal Dahiya, Himanshu Jain, Yashoteja Prabhu, and Manik Varma. 2016. The Extreme Classification Repository: Multi-label Datasets and Code. http://manikvarma.org/downloads/XC/XMLRepository.html.

[7] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. 2015. Sparse Local Embeddings for Extreme Multi-label Classification. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/35051070e572e47d2c26c241ab88307f-Paper.pdf

[8] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/621461af90cadfdaf0e8d4cc25129f91-Paper.pdf

[9] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit Dhillon. 2019. A modular deep learning approach for extreme multi-label text classification. *arXiv preprint arXiv:1905.02331* (2019).

[10] Yu-Ting Chou, Gang Niu, Hsuan-Tien Lin, and Masashi Sugiyama. 2020. Unbiased risk estimators can mislead: A case study of learning with complementary labels. In *International Conference on Machine Learning*. PMLR, 1929–1938.

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

[12] Ofer Dekel and Ohad Shamir. 2010. Multiclass-multilabel classification with more classes than examples. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 137–144.

[13] Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. 2010. What does classifying more than 10,000 image categories tell us?. In *ECCV*.

[14] Emily Denton, Jason Weston, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2015. User Conditional Hashtag Prediction for Images. In *KDD*.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[16] Chuan Guo, Ali Mousavi, Xiang Wu, Daniel N Holtmann-Rice, Satyen Kale, Sashank Reddi, and Sanjiv Kumar. 2019. Breaking the Glass Ceiling for Embedding-Based Classifiers for Large Output Spaces. In *Advances in Neural Information Processing Systems*. 4944–4954.

[17] Himanshu Jain, Venkatesh Balasubramanian, Bhanu Chunduri, and Manik Varma. 2019. Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches. In *WSDM*. 528–536.

[18] Himanshu Jain, Yashoteja Prabhu, and Manik Varma. 2016. Extreme Multi-label Loss Functions for Recommendation, Tagging, Ranking and Other Missing Label Applications. In *KDD*.

[19] Ankit Jalan and Purushottam Kar. 2019. Accelerating extreme classification via adaptive feature agglomeration. *arXiv preprint arXiv:1905.11769* (2019).

[20] Kalina Jasinska-Kobus, Marek Wydmuch, Krzysztof Dembczyński, Mikhail Kuznetsov, and Róbert Busa-Fekete. 2020. Probabilistic Label Trees for Extreme Multi-Label Classification. *CoRR* abs/2009.11218 (2020).

[21] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. 2020. Decoupling Representation and Classifier for Long-Tailed Recognition. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1gRTCVFvB

[22] Sujay Khandagale, Han Xiao, and Rohit Babbar. 2020. Bonsai: diverse and shallow trees for extreme multi-label classification. *Machine Learning* (2020), 1–21.

[23] Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. 2017. Positive-unlabeled learning with non-negative risk estimator. *arXiv preprint arXiv:1703.00593* (2017).

[24] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep Learning for Extreme Multi-label Text Classification. In *SIGIR*. ACM, 115–124.

[25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[26] Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. 2017. Cost-sensitive learning with noisy labels. *The Journal of Machine Learning Research* 18, 1 (2017), 5666–5698.

[27] Ioannis Partalas, Aris Kosmopoulos, Nicolas Baskiotis, Thierry Artieres, George Paliouras, Eric Gaussier, Ion Androutsopoulos, Massih-Reza Amini, and Patrick Galinari. 2015. Lshtc: A benchmark for large-scale text classification. *arXiv preprint arXiv:1503.08581* (2015).

[28] Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. 2018. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *Proceedings of the 2018 World Wide Web Conference*. 993–1002.

[29] Yashoteja Prabhu and Manik Varma. 2014. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*. ACM, 263–272.

[30] Filip Radlinski, Paul N Bennett, Ben Carterette, and Thorsten Joachims. 2009. Redundancy, diversity and interdependent document relevance. In *ACM SIGIR Forum*.

[31] Sashank J Reddi, Satyen Kale, Felix Yu, Daniel Holtmann-Rice, Jiecao Chen, and Sanjiv Kumar. 2019. Stochastic Negative Mining for Learning with Large Output Spaces. In *The 22nd International Conference on Artificial Intelligence and Statistics*. 1940–1949.

[32] Guy Shani and Asela Gunawardana. 2013. Tutorial on application-oriented evaluation of recommendation systems. *AI Communications* (2013).

[33] Yukihiro Tagami. 2017. AnnexML: Approximate Nearest Neighbor Search for Extreme Multi-label Classification. In *KDD*. ACM.

[34] Tong Wei and Yu-Feng Li. 2019. Does Tail Label Help for Large-Scale Multi-Label Learning? *IEEE Transactions on Neural Networks and Learning Systems* (2019).

[35] Marek Wydmuch, Kalina Jasinska, Mikhail Kuznetsov, Róbert Busa-Fekete, and Krzysztof Dembczynski. 2018. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In *Advances in Neural Information Processing Systems*. 6355–6366.

[36] Chang Xu, Dacheng Tao, and Chao Xu. 2016. Robust Extreme Multi-label Learning. In *KDD*.

[37] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *CoRR* abs/1906.08237 (2019). arXiv:1906.08237 http://arxiv.org/abs/1906.08237

[38] Hui Ye, Zhiyu Chen, Da-Han Wang, and Brian Davison. 2020. Pretrained Generalized Autoregressive Model with Adaptive Probabilistic Label Clusters for Extreme Multi-label Text Classification. In *International Conference on Machine Learning*. PMLR, 10809–10819.

[39] Ian EH Yen, Xiangru Huang, Wei Dai, Pradeep Ravikumar, Inderjit Dhillon, and Eric Xing. 2017. Ppdsparse: A parallel primal-dual sparse method for extreme classification. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 545–553.

[40] Ian En-Hsu Yen, Xiangru Huang, Pradeep Ravikumar, Kai Zhong, and Inderjit S. Dhillon. 2016. PD-Sparse : A Primal and Dual Sparse Approach to Extreme Multiclass and Multilabel Classification. In *ICML*.

[41] Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. AttentionXML: Label Tree-based Attention-Aware Deep Model for High-Performance Extreme Multi-Label Text Classification. In *Advances in Neural Information Processing Systems*. 5812–5822.

[42] Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit Dhillon. 2014. Large-scale Multi-label Learning with Missing Labels. In *Proceedings of the 31st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 32)*, Eric P. Xing and Tony Jebara (Eds.). PMLR, Bejing, China, 593–601. http://proceedings.mlr.press/v32/yu14.html