Online Hierarchical Multi–label Classification

Wenting Qi
Department of Computer Science
University at Albany, SUNY
Albany, New York, USA
wqi@albany.edu

Charalampos Chelmis*†
Department of Computer Science
University at Albany, SUNY
Albany, New York, USA
cchelmis@albany.edu

2023 IEEE International Conference on Big Data (BigData) | 979-8-3503-2445-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/BigData59044.2023.10386110

***Abstract*—Existing approaches for multi–label classification are trained offline, missing the opportunity to adapt to new data instances as they become available. To address this gap, an online multi–label classification method was proposed recently, to learn from data instances sequentially. In this work, we focus on multi–label classification tasks, in which the labels are organized in a hierarchy. We formulate online hierarchical multi–labeled classification as an online optimization task that jointly learns individual label predictors and a label threshold, and propose a novel hierarchy constraint to penalize predictions that are inconsistent with the label hierarchy structure. Experimental results on three benchmark datasets show that the proposed approach outperforms online multi–label classification methods, and achieves comparable to, or even better performance than offline hierarchical classification frameworks with respect to hierarchical evaluation metrics.**

***Index Terms*—Hierarchical classification, online learning, learning with constrains**

I. Introduction

Multi–label classification refers to the task of classifying data instances, each of which is associated with multiple labels [1]. Multi–label classification algorithms have been widely applied to many real–world application scenarios, including but not limited to, protein function classification [2] and semantic scene classification [3]. The majority of multi–label classification methods [4]–[7] are trained offline, given a training set. However, in the era of Big Data, many existing and emerging applications, such as genomic data analysis [8] and misinformation detection [9], often require classifiers to be trained both on large datasets, as well as in an online manner. For example, a news classifier system trained frequently on current news articles, may experience an increased number of news on a given topic in one day (e.g., election results), and a totally different set of topics the following day (e.g., sports).

To facilitate training a classifier with an incomplete set of training data instances, and to adapt the classification model as new data instances become available, online learning techniques have been proposed [10]. Different from traditional learning, online learning trains a model sequentially by training on data instances arising one by one at each learning epoch, with the goal of maximizing prediction accuracy.

Recently, a framework of adaptive label threshold for online multi–label classification (FALT) was proposed [11]. However, [11] is not applicable to scenarios where the labels themselves are organized in a hierarchy [12]. Such scenarios include, but are not limited to, image and video annotation [13], [14], text classification [15], [16], and genomics classification [17]–[19]. In hierarchical classification [12], algorithms are designed with respect to a particular label hierarchy structure. Specifically, labels are often arranged as a tree [20] or directed acyclic graph (DAG) [21], where each label is a node residing in a path of the hierarchy. One common hierarchical constraint [20], [22] is to allow a current non–root node (label) become a member of the prediction label set only if its parent's (similarly, ancestor's) label is also in the prediction label set. Unfortunately, naively incorporating such hard constraint into FALT, would make its objective function noncontinuous, which would in turn mean that a gradient based method could no longer be employed to derive a closed–form update.

This paper proposes a novel online hierarchical multi–label classification framework that learns a classifier as training data instances arrive one at a time, while at the same time ensuring that predicted labels conform to the hierarchy structure. Our main contributions can be summarized as follows:

- We propose a novel hierarchy constraint that captures the entire label hierarchy structure, as opposed to local parent–child relationships.
- We formulate online hierarchical multi–labeled classification as an online optimization task that facilitates the joint learning of individual label predictions that compute scores for each label. We incorporate our hierarchy constraint directly in the objective function to penalize predictions that are inconsistent with the label hierarchy structure.
- We derive a closed form update for the case of linear classifiers.
- We experimentally evaluate the effectiveness of our approach using three benchmark datasets.

This material is based upon work supported by the National Science Foundation under Grant No. ECCS–1737443.
*Corresponding author.
†Both authors contributed equally.

II. Related Work

A. Multi–label Classification

Most multi–label classification algorithms can be divided into two categories, namely *transformation–based methods* and *algorithm variant methods*. Transformation–based methods convert the original multi–label classification task into a set of binary classification problems (i.e., learn a binary classifier for each class [23]) or learn ensembles (e.g., [24], [25]). The main drawback of transformation–based methods is that fixing the label threshold for each binary classifier may degrade performance, since fixing the label threshold to a predefined value may not always be optimal [11]. Methods in algorithm variant category modify existing algorithms, such as supported vector machine (SVM) [26], k–nearest neighbors (KNN) [27], and neural networks [28] to apply on multi–label classification tasks. For instance, multi–label classification AdaBoost.MH [29] is a variant of the AdaBoost algorithm [30]. All such methods are designed for *offline* learning, and can therefore not adapt to new incoming data instances. To the best of our knowledge, only [11] supports online multi–label classification, but the method proposed there is not readily applicable to hierarchical multi–label classification.

B. Online Learning

Online learning aims to train a machine learning model on data that become available sequentially, such that the model can be improved step by step as new data arrive [10]. A family of online learning algorithms based on linear predictors have been proposed in [31], which can be utilized for multiple tasks, including binary and multi–class classification. Online learning algorithms for single–label classification tasks (i.e. every data instance has exactly one label) include kernel–based algorithms [32], and perceptron–based algorithms [33], [34]. Online classification algorithms for multi–label classification tasks (i.e. every data instance has multiple labels), include streaming multi–label random trees (SMART) [35], extreme learning machine for online multi–label learning [36], [37], and ensembles of the multi–label Hoeffding trees [38]. Similar to [11], these methods are restricted to vectors of labels that are assumed to be unrelated. Instead, the method proposed here is the first online multi–label classification method that encodes hierarchical label relations directly into the multi–label learning task.

C. Hierarchical Multi–label Classification

Hierarchical multi–label classification methods can be categorized into three main themes. The first theme comprises methods that separate the hierarchical multi–label classification task into independent binary classification tasks for each class in the hierarchy, and then leverages existing methods to learn a classifier for each class separately, completely disregarding the hierarchical constraint [39]. The second theme comprises methods that restrict the training set for a particular class to those data instances belonging to the parent class [40]. However, the parameters of the classifier are increased dramatically for complex hierarchical structures (e.g., hundreds of nodes located in the class hierarchy). The third theme learns a single multi–label classifier and considers hierarchical constraint in the prediction stage [39]. Unfortunately, such methods do not generalize because of restrictions on certain learning model structures. Instead, we propose a hierarchy constraint that can easily be incorporated into any multi–label classification algorithm.

D. Online Hierarchical Classification

Online hierarchical classification models, such as [41] aim to classify streaming data in a top–down manner by learning a classifier for each node. Unlike [41], which is only applicable to tree hierarchies, the approach proposed here can accommodate any hierarchy structure, while at the same time learning a global classifier for all classes at once, thus reducing the number of parameters to be learned. Finally, [42] facilitates incremental learning when the label hierarchy itself is to be learned and can potentially change as new data instances arrive. Although equally challenging, that problem is orthogonal to the one addressed here.

III. Preliminaries

A. Problem Definition

Let D denote the training set, and $(\mathbf{x}, Y)$ denote each training data instance in D, where $\mathbf{x} \in \mathbb{R}^d$ and $Y \subseteq \mathcal{Y}$, and $1 \leq |\mathcal{Y}| \leq L$, with L being the total number of labels. Among the label space $\mathcal{Y}$, labels are organized into the hierarchy $\mathcal{H}$. We use i as the label index, so that $\mathcal{Y}[i]$ denotes the membership of $\mathbf{x}$ to the i–th label in $\mathcal{H}$, and $\mathcal{H}[i]$ is the i–th label in the hierarchy. Without loss of generality, for a tree hierarchy, nodes are indexed as $1, 2, 3, ..., L$ in a top to bottom manner (i.e., 1 is the root, 2 indicates its leftmost child, and so forth). Additionally, let $pa(i)$ denote node i's parent in $\mathcal{H}$. Finally, let an arbitrary sequence $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_T, y_T)$ denote the data instances arriving sequentially, one at a time (i.e., $(\mathbf{x}_t, y_t)$ for any learning round $1 \leq t \leq T$). At each round t, $\mathbf{W}_t = [\mathbf{w}_t^1, ..., \mathbf{w}_t^i, ..., \mathbf{w}_t^L, \mathbf{w}_t^{L+1}] \in \mathbb{R}^{d \times (L+1)}$ denotes the current multi–label classifier comprising L label predictors that compute scores for each label of $\mathbf{x}_t$, and an additional predictor for determining the threshold to be used for assigning labels to $\mathbf{x}_t$. Without loss of generality, we drop the round variable t, and refer to $\mathbf{x}_t$ simply as $\mathbf{x}$ to avoid confusion. Label $i \in \mathcal{Y}$ is considered to be relevant for $\mathbf{x}$ if $\mathbf{x}^T\mathbf{w}^i > \mathbf{x}^T\mathbf{w}^{L+1}$. Thus, there are two true label sets for each $\mathbf{x}$: one relevant label set Y and one irrelevant label set $\tilde{Y}$ (i.e., $\tilde{Y} = \mathcal{Y} - Y$). $\hat{Y}$ denotes the predicted label set which is $\{\hat{Y} : i \in \mathcal{Y} : \mathbf{x}^T\mathbf{w}^i > \mathbf{x}^T\mathbf{w}^{L+1}\}$. After that, the ground truth Y_t is revealed, and the differences (if any) between Y_t and $\hat{Y}_t$ are used to adapt $\mathbf{W_t}$ into a new model $\mathbf{W_t}$, which is expected to perform better in the next round, $t + 1$. Table I summarizes the notation used hereafter.

Given $\mathcal{D}$ and $\mathcal{H}$, the goal is to train a hierarchical multi–label classification model as well as label threshold in an online manner, which can be used to predict the hierarchical categories of unseen data instances.

TABLE I
EXPLANATION OF MAIN SYMBOLS.

Symbol	Description
D	Training set
t	Online learning round index
L	Total number of labels. The index of each label is denoted as i
$\mathbf{x}_t$	The arrived data instance at t-th learning round
$\mathbf{w}_t^i$	Learning weight for label i
$\mathbf{W}_t$	Linear classifier at t-th learning round
$\mathcal{Y}$	Label space
Y_t	True relevant label set for data instance x_t
$\bar{Y}_t$	True irrelevant label set for data instance x_t
$\hat{Y}_t$	Predicted label set for data instance x_t
$\hat{\mathbf{Y}}_\mathbf{t}$	Prediction result for data instance x_t in the vector form
H	Hierarchical relationship matrix with m and n denoting row and column index separately
h	The depth of the hierarchical label structure
$\hat{p}_t$	Valid hierarchical edges inside of the prediction result $\hat{Y}_t$
$pa(i)$	The parent label of i
α_t	Learning step size

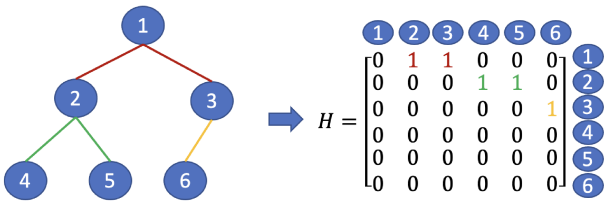


Fig. 1. Toy hierarchy and its corresponding matrix H.

IV. HIERARCHY CONSTRAINT

A. Definition

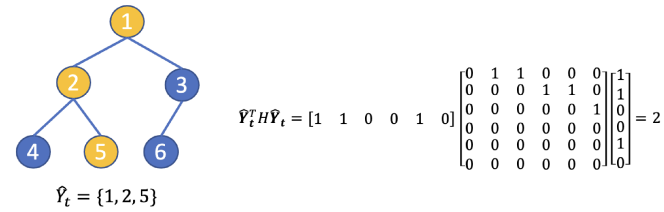
We design a hierarchy matrix $H \in \mathbb{R}^{L\times L}$ to represent the hierarchical label structure $\mathcal{H}$. Specifically, H records all links between each non–root node to its parents ($< i, pa(i) >$). For instance, in the toy example shown in Figure 1, the link between node 2 to 4 is a valid edge according to the hierarchy. Matrix H is upper triangular, with 0 in the diagonal. Indices $m \in [1, ..., L]$ and $n \in [1, ..., L]$ refer to the rows and columns of H, respectively. Each element H_{mn} is 1 if label m is the parent of label n in $\mathcal{H}$, otherwise 0.

The corresponding matrix H for the toy hierarchy in Figure 1 is shown for reference. For convenience we use h to denote the depth of the hierarchical label structure (2 in the toy example).

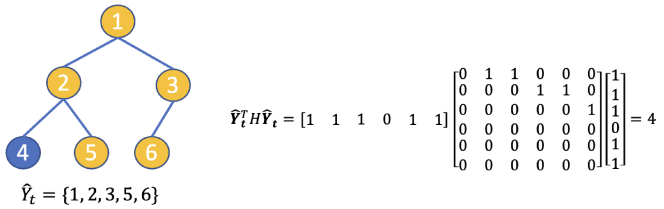
We calculate the number of edges inside $\hat{Y}$ that are valid with respect to $\mathcal{H}$ as:

$$\hat{p} = \hat{\mathbf{Y}}^T H \hat{\mathbf{Y}}, \tag{1}$$

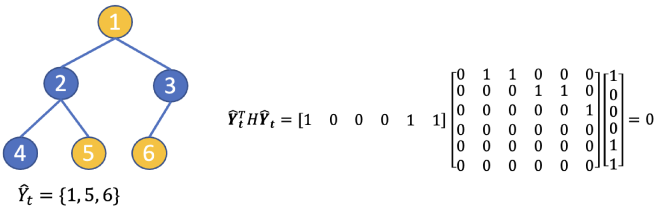
where $\hat{\mathbf{Y}} \in \mathbb{R}^{L\times 1}$ denotes the vector representation of $\hat{Y}$, with entry i being 1 if $i \in \hat{Y}$, otherwise 0. For a single path, if all predicted labels in $\hat{\mathbf{Y}}$ follow the label hierarchical structure, the number of valid edges with respect to the hierarchy is $||\hat{\mathbf{Y}}|| - 1$. Therefore, we wish each prediction output $\hat{Y}$ to satisfy the hierarchy constraint $\hat{\mathbf{Y}}^T H \hat{\mathbf{Y}} = ||\hat{\mathbf{Y}}|| - 1$.



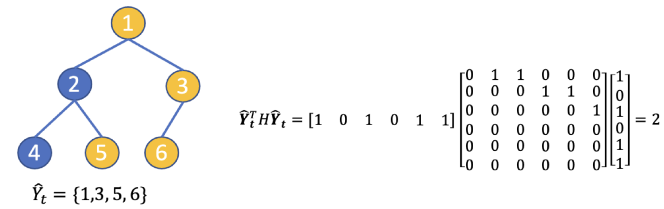
(a) Example 1



(b) Example 2



(c) Example 3



(d) Example 4

Fig. 2. Toy example of four possible predictions, to illustrate $\hat{p}_t = \hat{\mathbf{Y}}_\mathbf{t}^T H \hat{\mathbf{Y}}_\mathbf{t}$.

Figure 2(a) shows one possible prediction result, where $\hat{Y}_t = \{1, 2, 5\}$, and $\hat{p} = 2$ due to two valid edges: $< 1, 2 >$ and $< 2, 5 >$. Figure 2(b) shows a different prediction outcome $\hat{Y}_t = \{1, 2, 3, 5, 6\}$ with four valid edges (i.e., $\hat{p} = 4$), namely $< 1, 2 >$, $< 1, 3 >$, $< 2, 5 >$ and $< 3, 6 >$.

This example illustrates another important property of the proposed hierarchy constraint. Specifically, H is not limited to single–path hierarchical classification tasks. Instead, it can accommodate multiple paths, as illustrated in Figure 2(b). Figure 2(c) shows an example of an undesirable prediction ($\hat{Y}_t = \{1, 5, 6\}$). The prediction is undesirable because it has 0 valid edges with respect to $\mathcal{H}$. Finally, Figure 2(d) shows a prediction output (i.e., $\hat{Y}_t = \{1, 3, 5, 6\}$) that has only two valid edges according to $\mathcal{H}$.

B. Generality

Hierarchical multi–label classification types can be compactly described as a $3-$tuple $< \Upsilon, \Psi, \Phi >$, where Υ denotes

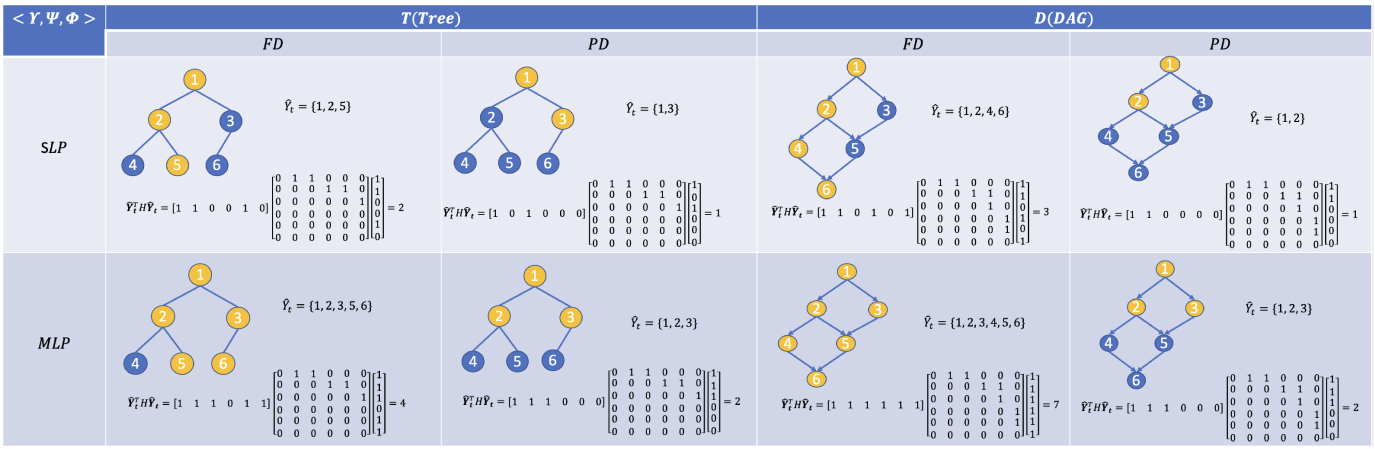
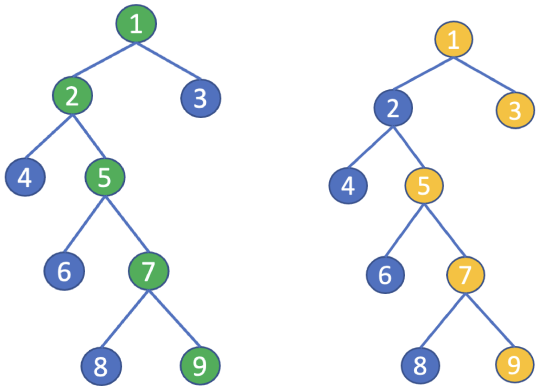


Fig. 3. Examples to illustrate the broad applicability of the proposed constraint in diverse hierarchical multi–label classification settings. Yellow denotes predicted labels for data instances x_t. Full Depth, Partial Depth, Single Label Path, Multiple Label Paths are abbreviated as FD, PD, SLP, and MLP.



(a) Ground truth (b) Possible prediction result

Fig. 4. Illustrative example of the top–down hierarchical constraint.

the type of graph (i.e., $T(Tree)$ or $D(DAG)$) representing the hierarchy of classes. Ψ (i.e., SLP[1], MLP[2]) indicates whether labels for each data instance are allowed to be associated with single or multiple paths in the hierarchy, and Φ (i.e., FD[3], PD[4]) indicates whether a data instance can have full or partial depth of labeling [12]. Figure 3 shows examples of such hierarchical multi–label classification settings, and illustrates the ability of the proposed hierarchy constraint to encompass all such cases.

In addition to generalizing other settings, the proposed hierarchy constraint overcomes the limitation of traditional top–down [43] approaches. Specifically, top–down methods require upper–level predictions to be accurate to avoid propagating errors down to the lower levels. Figure 4 illustrates the limitation of top–down hierarchical constraints. Specifically, Figure 4(a) shows the ground truth whereas Figure 4(b) shows a possible prediction result. According to the top–down constraint, only class 1 (i.e., node 1) is considered to be correctly predicted, even though classes $5, 7, 9$ seem to be labeled correctly as compared to the ground truth (see Figure 4(a)). This is because top–down methods enforce hierarchy constraints after making classification predictions, and once a mistake is detected (i.e., at class 2), the remainder of the predictions are considered to be wrong for training purposes. Instead, the number of valid edges with respect to proposed hierarchy constraint is 3 (i.e., $< 1, 3 >, < 5, 7 >, < 7, 9 >$) out of a total of 4 valid edges (i.e., $< 1, 2 >, < 2, 5 >, < 5, 7 >, < 7, 9 >$). Note that edge $< 1, 3 >$ is valid with respect to the hierarchy, but should be treated as a classification mistake. Section V discusses how to obtain correct classifications in addition to enforcing valid paths according to the hierarchy.

[1]Single Path of Labels.

[2]Multiple Paths of Labels.

[3]Full Depth Labeling: Every instance is labeled with classes at all levels [12].

[4]Multiple Paths of Labels: The value of the class label at some level is unknown [12].

V. Online Hierarchical Multi–Label Classification (OHMC)

We propose to solve the online hierarchical multi–labeled classification problem by solving the following objective:

$$\begin{aligned} \mathbf{W}_{t+1} = & \underset{\mathbf{W}\in\mathbb{R}^{d\times(L+1)}}{\arg\min} \frac{1}{2}||\mathbf{W}-\mathbf{W}_t||_F^2 \\ s.t. \quad & \mathbf{x}_t^T\mathbf{w}^i - \mathbf{x}_t^T\mathbf{w}^{L+1} \geq 1, \forall i \in Y_t \\ & \mathbf{x}_t^T\mathbf{w}^{L+1} - \mathbf{x}_t^T\mathbf{w}^i \geq 1, \forall i \in \tilde{Y}_t \\ & \hat{\mathbf{Y}}_\mathbf{t}^T H \hat{\mathbf{Y}}_\mathbf{t} = ||\hat{\mathbf{Y}}|| - 1 \end{aligned} \tag{2}$$

where $||\cdot||_F$ denotes Frobenius norm of a matrix, and $||\cdot||$ denotes the $L1$ norm. With each new round, the goal is to improve $\mathbf{W}_t$ to $\mathbf{W}_{t+1}$ to achieve better classification accuracy (i.e., separating the relevant and irrelevant sets with high confidence). The hierarchy constraint (i.e., last term in Eq. 2) is used to penalize predictions that violate the label hierarchy structure.

At each learning round t, the known information includes the current linear learner $\mathbf{W}_t$ and the newly arrived data

instance $\mathbf{x}_t$. $\hat{\mathbf{Y}}_\mathbf{t} \in \mathbb{R}^{L\times 1}$ is computed as a function of $\mathbf{W}_t$ and $\mathbf{x}_t$. Specifically, we leverage the sigmoid function (i.e., $Sigmoid(x) = S(x) = \frac{1}{1+e^{-\lambda x}}$) to simulate the decision process (i.e., $\hat{Y}_t^i = 1$ if $\mathbf{x}_t^T\mathbf{w}^i > \mathbf{x}_t^T\mathbf{w}^{L+1}$), where λ is a hyper–parameter that ensures $S(x)$ is close to 1 if $x > 0$, and close to 0 if $x < 0$. Therefore, $\hat{Y}_t^i = \frac{1}{1+e^{-\lambda x}}$, where $x = \mathbf{x}_t^T\mathbf{w}^i - \mathbf{x}_t^T\mathbf{w}^{L+1}$. By including the predicted labels into the objective function, we get:

$$\begin{aligned}(\mathbf{W}_{t+1}, \xi_{t+1}) = \arg\min_{\mathbf{W},\xi}\{&\frac{1}{2}||\mathbf{W}-\mathbf{W}_t||_F^2 \\ &+ \eta(\frac{1}{|Y_t|\sum_{i\in Y_t}\xi_i} + \frac{1}{|\tilde{Y}_t|\sum_{i\in\tilde{Y}_t}\xi_i})\} \\ s.t.\quad &\mathbf{x}_t^T\mathbf{w}^i - \mathbf{x}_t^T\mathbf{w}^{L+1} \geq 1-\xi_i, \forall i \in Y_t \\ &\mathbf{x}_t^T\mathbf{w}^{L+1} - \mathbf{x}_t^T\mathbf{w}^i \geq 1-\xi_i, \forall i \in \tilde{Y}_t \\ &\xi_i \geq 0, \forall i \in \{1,2,...,L\} \\ &\sum_{i=1}^{L}(\sum_{n=1}^{L} V_n H_{ni})V_i = h, \\ &V_i = \frac{1}{1+e^{-\lambda(\mathbf{x}_t^T\mathbf{w}^i - \mathbf{x}_t^T\mathbf{w}^{L+1})}}\end{aligned} \tag{3}$$

where hyper–parameter $\eta > 0$ controls the trade–off between the first regularization term and the slack variable term. Recall that h is the depth of the hierarchy structure, which for a single–path classification task on a tree hierarchy, is the maximum number of valid edges. For multi–path multi–label hierarchical tasks, the upper bound of possible valid edges can be changed to kh, where k denotes the maximum number of paths allowed. By incorporating the hierarchy constraint directly into the objective function, we get:

$$\begin{aligned}(\mathbf{W}_{t+1},\xi_{t+1}) = \arg\min_{\mathbf{W},\xi}\{&\frac{1}{2}||\mathbf{W}-\mathbf{W}_t||_F^2 \\ &+ \eta(\frac{1}{|Y_t|\sum_{i\in Y_t}\xi_i} + \frac{1}{|\tilde{Y}_t|\sum_{i\in Y_t}\xi_i}) \\ &+ \frac{1}{2}(\sum_{i=1}^{L}(\sum_{n=1}^{L} V_n H_{ni})V_i - h)^2\}, \\ s.t.\quad &\mathbf{x}_t^T\mathbf{w}^i - \mathbf{x}_t^T\mathbf{w}^{L+1} \geq 1-\xi_i, \forall i \in Y_t, \\ &\mathbf{x}_t^T\mathbf{w}^{L+1} - \mathbf{x}_t^T\mathbf{w}^i \geq 1-\xi_i, \forall i \in \tilde{Y}_t, \\ &\xi_i \geq 0, \forall i \in \{1,2,...,L\}.\end{aligned} \tag{4}$$

Equation (4) can be compactly expressed as:

$$\begin{aligned}\mathbf{W}_{t+1} = \arg\min_{\mathbf{W}}\{&\frac{1}{2\eta}||\mathbf{W}-\mathbf{W}_t||_F^2 + f_t(\mathbf{W}) \\ &+ \frac{1}{2}(\sum_{i=1}^{L}(\sum_{n=1}^{L} V_n H_{ni})V_i - h)^2\}\end{aligned} \tag{5}$$

where $f_t(\mathbf{W})$ is defined in [11] as:

$$\begin{aligned}f_t(\mathbf{W}) = &\frac{1}{|Y_t|}\sum_{i\in Y_t} max\{0, 1-(\mathbf{x}_t^T\mathbf{w}^i - \mathbf{x}_t^T\mathbf{w}^{L+1})\} \\ &+ \frac{1}{|\tilde{Y}_t|}\sum_{i\in\tilde{Y}_t} max\{0, 1-(\mathbf{x}_t^T\mathbf{w}^{L+1} - \mathbf{x}_t^T\mathbf{w}^i)\}.\end{aligned} \tag{6}$$

Given that $f_t(\mathbf{W})$ has been shown to be piecewise linear [11], a first–order approximation of $f_t(\mathbf{W})$ can be used, instead of directly optimizing it. Leveraging this approximation, the objective function in Equation (5) becomes:

$$\begin{aligned}\mathbf{W}_{t+1} = \arg\min_{\mathbf{W}}\{&\frac{1}{2\eta}||\mathbf{W}-\mathbf{W}_t||_F^2 + \sum_{i=1}^{L+1}(\nabla_t^i)^T\mathbf{w}^i \\ &+ \frac{1}{2}(\sum_{i=1}^{L}(\sum_{n=1}^{L} V_n H_{ni})V_i - h)^2\}.\end{aligned} \tag{7}$$

In this form, the objective function is differentiable and the first two terms (i.e., $\frac{1}{2\eta}||\mathbf{W}-\mathbf{W}_t||_F^2 + \sum_{i=1}^{L+1}(\nabla_t^i)^T\mathbf{w}^i$) are convex with respect to each $\mathbf{w}_t^i$. However, the last term (i.e., $\frac{1}{2}(\sum_{i=1}^{L}(\sum_{n=1}^{L} V_n H_{ni})V_i - h)^2$) is non–convex, as the sigmoid function is a non–convex function. We therefore use *Stochastic Gradient Descent (SGD)* [44] to update $\mathbf{w}_{t+1}^i$ as:

$$\mathbf{w}_{t+1}^i = \mathbf{w}_t^i - \alpha_t(\eta\nabla_t^i + h_w), \tag{8}$$

where α_t denotes the step size, and

$$\nabla_t^i = \begin{cases} -\frac{a_t^i}{|Y_t|}\mathbf{x}_t, \text{if } i \in Y_t \\ \frac{b_t^i}{|\tilde{Y}_t|}\mathbf{x}_t, \text{if } i \in \tilde{Y}_t \\ (\frac{a_t^i}{|Y_t|} - \frac{b_t^i}{|\tilde{Y}_t|})\mathbf{x}_t, \text{if } i = L+1 \end{cases} \tag{9}$$

with $a_t^i = \mathbb{1}[\mathbf{x}_t^T\mathbf{w}^i - \mathbf{x}_t^T\mathbf{w}^{L+1} < 1]$, $b_t^i = \mathbb{1}[\mathbf{x}_t^T\mathbf{w}^{L+1} - \mathbf{x}_t^T\mathbf{w}^i < 1]$, $a_t = \sum_{i\in Y_t} a_t^i$, and $b_t = \sum_{i\in\tilde{Y}_t} a_t^i$, and

$$\begin{aligned}h_w = &\frac{1}{1+e^{-\lambda(\mathbf{x}_t^T\mathbf{w}^i - \mathbf{x}_t^T\mathbf{w}^{L+1})}}\sum_{n=1}^{L}\frac{\lambda\mathbf{x}_t e^{-\lambda(\mathbf{x}_t^T\mathbf{w}^n - \mathbf{x}_t^T\mathbf{w}^{L+1})}H_{ni}}{(1+e^{-\lambda(\mathbf{x}_t^T\mathbf{w}^n - \mathbf{x}_t^T\mathbf{w}^{L+1})})^2} \\ &+ \frac{\lambda\mathbf{x}_t e^{-\lambda(\mathbf{x}_t^T\mathbf{w}^n - \mathbf{x}_t^T\mathbf{w}^{L+1})}}{(1+e^{-\lambda(\mathbf{x}_t^T\mathbf{w}^i - \mathbf{x}_t^T\mathbf{w}^{L+1})})^2}\sum_{n=1}^{L}\frac{H_{ni}}{1+e^{-\lambda(\mathbf{x}_t^T\mathbf{w}^n - \mathbf{x}_t^T\mathbf{w}^{L+1})}}.\end{aligned} \tag{10}$$

Equation (7) leads to an efficient implementation described in Algorithm 1.

A. Convergence Analysis

Here, we discuss the convergence of Algorithm 1. For simplicity, we denote Equation (8) as:

$$\mathbf{w}_{t+1}^i = \mathbf{w}_t^i - \alpha_t\nabla g(\mathbf{w}_t^i), \tag{11}$$

where $g(\mathbf{W})$ substitutes the objective function in (7). Specifically, at learning round $t+1$, by Taylor's theorem,

$$\begin{aligned}g(\mathbf{w}_{t+1}^i) &= g(\mathbf{w}_t^i - \alpha_t\nabla g(\mathbf{w}_t^i)) \\ &= g(\mathbf{w}_t) - \alpha_t\nabla g(\mathbf{w}_t^i)^T\nabla g(\mathbf{w}_t^i) \\ &\quad + \frac{\alpha_t^2}{2}\nabla g(\mathbf{w}_t^i)^T\nabla^2 g(\mathbf{w}_t^i)\nabla g(\mathbf{w}_t^i),\end{aligned} \tag{12}$$

where ∇^2 denotes the Hessian matrix. Now, we show that $-CI \preceq \nabla^2 g(\mathbf{w}_t^i) \preceq CI$, where $C \geq 0$, and I is the identity matrix. The sigmoid function (i.e., $S(x)$) is bounded. The

Algorithm 1 OHMC

Input: Dataset $\mathcal{D} = (x, \mathbf{Y}_t)$, hierarchical matrix H, label set $\mathcal{Y}$

1: Randomly initialize the linear weights $\mathbf{W}_1 = [\mathbf{w}_1^{(1)}, ..., \mathbf{w}_1^{(L+1)}]$
2: **repeat**
3: Observe incoming data instance $\mathbf{x}_t$
4: $\hat{Y}_t \leftarrow \emptyset$
5: **for** each $i \in \mathcal{Y}$ **do**
6: **if** $\mathbf{x}^T\mathbf{w}_t^i > \mathbf{x}^T\mathbf{w}_t^{L+1}$ **then**
7: Add i in $\hat{Y}_t$
8: **end if**
9: **end for**
10: Predict the relevant label set $\hat{Y}_t$,
11: Reveal true relevant and irrelevant label set Y_t and $\tilde{Y}_t$ (i.e., $\mathcal{Y} - Y_t$) for $\mathbf{x}_t$
12: Calculate h_w by Eq. (10) and ∇_t^i by Eq. (9)
13: $t \leftarrow t + 1$
14: Update $\mathbf{W}_t$ by Eq. (8)
15: **until** $t = T$
16: **return** $\mathbf{W}_t$

second order derivative of $S(x)$ is also bounded. The proof follows.

Proof 1:

$$\begin{aligned} S(x)^{'} &= (\frac{1}{1+e^{-\lambda x}})' = \frac{e^{-\lambda x}}{(1+e^{-\lambda x})^2} \\ &= \frac{1+e^{-\lambda x}-1}{(1+e^{-\lambda x})^2} = \frac{1}{1+e^{-\lambda x}} - \frac{1}{(1+e^{-\lambda x})^2} \\ &= S(x)(1-S(x)), \end{aligned} \tag{13}$$

and

$$\begin{aligned} S(x)^{''} &= (S(x)(1-(S(x))))' \\ &= (S(x) - S(x)^2)' \\ &= S(x)' - 2S(x)S(x)' \\ &= S(x)(1-S(x)) - 2S(x)S(x)(1-S(x)). \end{aligned} \tag{14}$$

The second order derivative of $S(x)$ comprises $S(x)$ itself, which is bounded in $[0, 1]$, meaning that the second order derivative of $S(x)$ is also bounded (i.e., $S(x)^{''}$ is bounded within $[1, -2]$).

Furthermore, the second order derivative of $\mathbf{x}_t^T\mathbf{w}_t^i - \mathbf{x}_t^T\mathbf{w}^{L+1}$ is bounded with respect to $\mathbf{w}_t^i$, as, without loss of generality, training data values can be assumed to be bounded[5]. Hence, $-CI \preceq \nabla^2 g(\mathbf{w}_t^i) \preceq CI$, and Equation (12) becomes:

$$\begin{aligned} g(\mathbf{w}_{t+1}^i) &\leq g(\mathbf{w}_t^i) - \alpha_t \nabla g(\mathbf{w}_t^i)^T \nabla g(\mathbf{w}_t^i) + \frac{\alpha_t^2 C}{2} \|\nabla g(\mathbf{w}_t^i)\|^2, \\ &= g(\mathbf{w}_t^i) - (\alpha_t - \frac{\alpha_t^2 C}{2}) \|\nabla g(\mathbf{w}_t^i)\|^2. \end{aligned} \tag{15}$$

[5]Feature values are often normalized or quantized during feature engineering, before training a machine learning model.

Setting $\alpha_t = \alpha$ that satisfies $1 - \frac{\alpha C}{2} > \frac{1}{2}$, we get:

$$g(\mathbf{w}_{t+1}^i) \leq g(\mathbf{w}_t^i) - \frac{\alpha_t}{2} \|\nabla g(\mathbf{w}_t^i)\|^2. \tag{16}$$

Summing up over all T learning rounds,

$$g(\mathbf{w}_T^i) \leq g(\mathbf{w}_t^i) - \sum_{t=0}^{T-1} \frac{\alpha_t}{2} \|\nabla g(\mathbf{w}_t^i)\|^2. \tag{17}$$

Let $\alpha_t = \frac{\alpha_0}{t+1}$. We get:

$$g(\mathbf{w}_T^i) \leq g(\mathbf{w}_t^i) - \sum_{t=0}^{T-1} \frac{\alpha_0}{2(t+1)} \|\nabla g(\mathbf{w}_t^i)\|^2. \tag{18}$$

Rearranging the above equation, we get:

$$\sum_{t=0}^{T-1} \frac{\alpha_0}{2(t+1)} \|\nabla g(\mathbf{w}_t^i)\|^2 \leq g(\mathbf{w}_t^i) - g(\mathbf{w}_T^i). \tag{19}$$

Let $z_T = \mathbf{w}_t$ with probability $\frac{1}{R_T(t+1)}$, where $R_T = \sum_{t=0}^{T-1} \frac{1}{t+1}$ [45]. Then,

$$\|\nabla g(z_T)\|^2 = \sum_{t=0}^{T-1} \frac{1}{R_T(t+1)} \|\nabla g(\mathbf{w}_t^i)\|^2, \tag{20}$$

which means there exists a constant Z that $\|\nabla g(z_T)\|^2 \leq \frac{Z}{logT}$ [45]. When $T \to \infty$, we have:

$$\|\nabla g(z_T)\|^2 \to 0. \tag{21}$$

Therefore, the SGD algorithm is expected to converge to a local minimum. This leads to the efficient algorithm, described in Algorithm 1.

B. Complexity Analysis

We analyze the complexity of OHMC with resepct to each online learning round (i.e., steps: 4–10). The complexity of predicting label set $\hat{Y}$ of a data instance $x_t$is $\mathcal{O}(KL)$, where K denotes the total number of features in x_t (steps: 4–7). Updating the learning weights is $\mathcal{O}(KL)$ (steps: 8–9), where the complexity for calculating Eqs. 9, 10, and 8 are $\mathcal{O}(KL)$, $\mathcal{O}(KL)$, and $\mathcal{O}(K)$ respectively. Thus, the overall complexity of Algorithm 1 for a single round is $\mathcal{O}(KL)$.

VI. Experiments

A. Datasets

We conduct experiments on three publicly available datasets: **ImageCLEF07D** and **ImageCLEF07A** (X–ray images extracted from the 2007 ImageCLEF competition), and **WIPO** (World International Patent Organization (WIPO) dataset used to classify patent texts). Table II summarizes these datasets. Note that the hierarchy depth for each of the three datasets is 3, and the number of classes in each level of each dataset is provided for reference. Intuitively, the number of alternative options (i.e., labels) increases with depth. The average label cardinally per data instance for all three datasets is 3 due to single–path labels (i.e., each data instance can have up to 3 labels). Compared with the total number of classes, the number of true labels per data instance is significantly smaller (i.e., the true label distribution is very sparse), making classification particularly challenging.

TABLE II
DATASETS USED IN EXPERIMENTAL EVALUATION.

Dataset	Number of train data	Number of test data	Number of features	Number of classes	Number of classes per level	Cardinality[1]
ImageCLEF07A	$10,000$	$1,006$	80	96	$8/25/63$	3
ImageCLEF07D	$10,000$	$1,006$	80	46	$4/16/26$	3
WIPO	$1,352$	358	$74,435$	187	$7/20/160$	3

[1] The average number of class labels per data instance.

B. Experimental Setup

All experiments were conducted on an iMac running macOS Big Sur with 3.8 GHz 8–core intel Core i7 processor and 16 GB 2667 MHz DDR4 memory. The learning step α_t is set to 0.81 in OHMC. The cooresponding value of α_t is set to 1 in FALT and SALT, as suggested in [11]. Both OHMC, and the baselines (e.g., FALT, SALT, H–Ada.MH, and Global) are implemented in Python 3.8. The training and test sets are pre–split for all the three datasets.

C. Baselines

To demonstrate the effectiveness of OHMC, we compare it with baseline approaches for multi–label classification and hierarchical multi–label classification, as follows.

1) Multi–label Classification Baselines:

- **FALT** [11]: First order linear algorithm for online multi–label classification and adaptive label threshold selection.
- **SALT** [11]: Similar to FALT, but each element in $\mathbf{w}_t$ is updated with an adaptive learning rate.

2) Hierarchical Classification Baselines:

- **HM3** [46]: Kernel–based algorithm for hierarchical classification using maximum margin variable [47] that allows SVM–style objective functions to be optimized over hierarchical label outputs. We use HM3–$l_{\triangle}$ to represent HM3 method with symmetric loss, and HM3–l_{sub} and HM3–l_{sibl} for HM3 with re-weight prediction errors based on sibling and subtree, respectively [46]. HM3–l_{uni} refers to uniform weighting in conjunction with hierarchical loss. Experimental results come from [48].
- **Clus–HMC–Ens** [49]: Top–down decision tree based approach, where each node located in the tree corresponds to a classifier, which contains all the training examples belonging to that parent's node. Reported experimental results come from [48].
- **H–Ada.MH** [22]: Hierarchy–aware AdaBoost variant for hierarchical multiclass classification.
- **Global** [50]: A global approach that builds a single classifier to discriminate between all labels simultaneously.

D. Evaluation Metrics

Let $\hat{y}(t)$ be the set comprising the predicted labels for $\mathbf{x}_t$, and let $\hat{T}(t)$ be the set consisting of the true labels from of data $\mathbf{x}_t$. $|\bullet|$ refers to the number of elements in the corresponding set. N is the total number of test data instances.

- **Hierarchical Classification Metrics** [48]:
  - **Micro–averaged hierarchical precision** (higher is better): $hP^{\mu} = \frac{\sum_t^N |\hat{y}_t \cap \hat{T}_t|}{\sum_t^N |\hat{y}_t|}$.
  - **Micro–averaged hierarchical recall** (higher is better): $hR^{\mu} = \frac{\sum_t^N |\hat{y}_t \cap \hat{T}_t|}{\sum_t^N |\hat{T}_t|}$.
  - **Micro–averaged hierarchical F–measure** (higher is better): $hF^{\mu} = \frac{2 \times hP^{\mu} \times hR^{\mu}}{hP^{\mu} + hR^{\mu}}$.
- **Multi–label Classification Metrics** [11]:
  - **Precision** (higher is better): $P = \frac{1}{N} \sum_t^N \frac{|\hat{y}_t \cap \hat{T}_t|}{|\hat{y}_t|}$,
  - **Recall** (higher is better): $R = \frac{1}{N} \sum_t^N \frac{|\hat{y}_t \cap \hat{T}_t|}{|\hat{T}_t|}$, and
  - **F1–measure** (higher is better): $F1 = \frac{2 \times P \times R}{P + R}$.
  - **Hamming Loss** (lower is better): $HL = \frac{1}{N \times L} \sum_t^N |\hat{y}_t \triangle \hat{T}_t|$, where $\triangle$ refers to symmetric difference between two sets.
  - **Ranking Loss** (lower is better): $RL = \frac{1}{N} \sum_t^N \frac{\sum_{i \in Y_t, k \in \bar{Y}_t} \mathbb{1}[h(x_t,i) \leq h(x_t,k)]}{|\bar{Y}_t| \times |Y_t|}$, where $h(x_t, i)$ denotes the real value score assigned to label i.

TABLE III
PERFORMANCE COMPARISON WITH RESPECT TO MULTI–LABEL CLASSIFICATION METRICS.

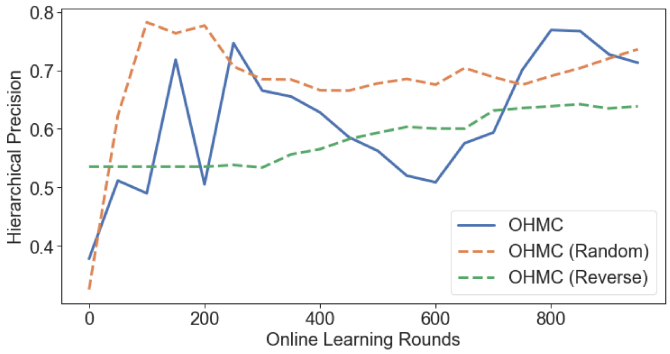
Dataset	Metric	OHMC	FALT	SALT
ImageCLEF07A	Psn	**0.54**	0.48	0.51
	Rcal	**0.55**	0.49	0.48
	F1	**0.54**	0.48	0.40
	HL	**3.56**	3.75	3.1
	RL	**0.13**	1.21	0.94
ImageCLEF07D	Psn	**0.76**	0.63	0.63
	Rcal	**0.69**	0.62	0.63
	F1	**0.73**	0.62	0.63
	HL	**2.13**	2.82	2.72
	RL	**0.08**	0.19	0.18
WIPO	Psn	**0.74**	0.68	0.68
	Rcal	**0.68**	0.68	0.68
	F1	**0.70**	0.68	0.68
	HL	**1.67**	1.85	1.86
	RL	**0.04**	0.15	0.15

E. Results

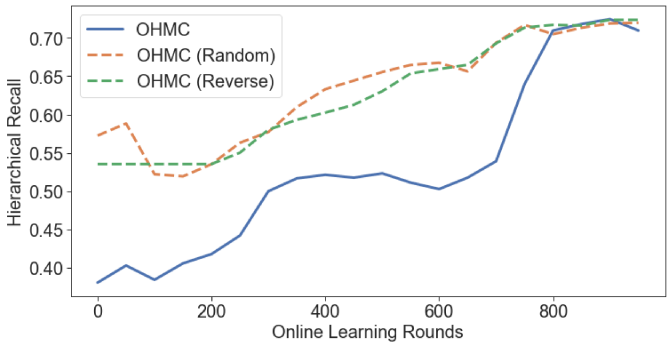
We begin by comparing OHMC with FALT and SALT with respect to multi–label classification metrics. Table III shows the evaluation results. As expected OHMC achieves the best performances in all cases, which illustrating the benefit of incorporating the hierarchical constraint directly into existing multi–label classification for adapting hierarchical multi–label classification task. The competitive advantage of OHMC becomes clearer in the ImageCLEF07D dataset, where OHMC achieves a 10% increase in F1 score as compared to FALT and SALT. Among the multi–label evaluation metrics, the RL value of OHMC is significantly lower than FALT and SALT. Average number of cases in which the real value prediction score $h(x_t, i)$ is smaller than $h(x_t, k)$ where i denotes the

TABLE IV
PERFORMANCE COMPARISON WITH HIERARCHICAL MULTI–LABEL CLASSIFICATION METHODS.

Dataset	Metric	OHMC	HM3–$l_{\triangle}$	HM3–l_{uni}	HM3–l_{sibl}	HM3–l_{sub}	Clus–HMC–Ens	H–Ada.MH	Gobal
ImageCLEF07A	hP^{μ}	0.57	0.51	0.53	0.46	0.41	0.70	**0.74**	0.67
	hR^{μ}	0.51	0.41	0.42	0.43	0.42	**0.71**	0.69	0.62
	hF^{μ}	0.52	0.45	0.42	0.43	0.46	**0.71**	**0.71**	0.64
ImageCLEF07D	hP^{μ}	0.74	0.47	0.45	0.48	0.52	**0.80**	0.73	0.66
	hR^{μ}	0.72	0.38	0.41	0.41	0.42	**0.81**	0.72	0.65
	hF^{μ}	0.72	0.41	0.42	0.43	0.46	**0.80**	0.72	0.65
WIPO	hP^{μ}	**0.69**	0.62	0.62	0.60	0.60	0.68	0.53	0.39
	hR^{μ}	**0.70**	0.40	0.38	0.50	0.50	0.68	0.49	0.24
	hF^{μ}	**0.69**	0.49	0.47	0.55	0.55	0.68	0.50	0.29



(a) Precision



(b) Recall

Fig. 5. Experiment results of WIPO for hP^{μ} and hR^{μ} with respect to OHMC (without disturbing the order of data instances), OHMC (Random), i.e., OHMC with data instances processed in random order), and OHMC (Reverse), i.e., OHMC with data instances appearing in reverse order.

true relevant label and k denotes the true irrelevant label. This suggests that OHMC is better able to distinguish between the relevant and irrelevant labels.

Next, we compare OHMC with hierarchical classification baselines (i.e., HM3–$l_{\triangle}$, HM3–l_{uni}, HM3–l_{sibl}, HM3–l_{sub}, Clus–HMC–Ens, H–Ada.MH, Gobal). Table IV presents the comparison results. In WIPO, the competitive advantage of OHMC becomes more prominent, as it outperforms all baselines. As for ImageCLEF07A and ImageCLEF07D, Clus–HMC–Ens and H–Ada.Mh perform better than OHMC. Our explanation for this result is threefold: (i) Clus–HMC–Ens leverages decision tree based learner, and trains each class as a separate classifier by pre–arranging data instances belonging to its parent's node, whereas OHMC considers all classes, training a single linear–based classifier without separating the training data instances (i.e., there are total 96 based classifiers in Clus–HMC–Ens compared to 1 in OHMC in ImageCLEF07A), (ii) hierarchical multi–label classification baselines are allowed to repeatedly train over each data instance in each learning epoch, whereas OHMC learns online, using each data instance once, and (iii) a nonlinear classifier may be needed to improve classification performance in image domain (e.g., ImageCLEF07A and ImageCLEF07D) where features are highly correlated. Nevertheless, OHMC achieves better results on ImageCLEF07A and ImageCLEF07D compared with SALT and FALT, which illustrates the benefit of proposed objective function in hierarchical classification tasks. Additionally, OHMC is able to better handle the large number of features and the complex hierarchical structure of WIPO.

Last but not least, we explore the impact (if any) of the order in which training data instances become available to OHMC. Specifically, we consider two additional scenarios, in which the order of training data instances is (i) randomized, and (ii) reversed with respect to the order of their appearance in the original WIPO dataset. These are denoted as OHMC(Random) and OHMC(Reverse), respectively. Figure 5 shows how hierarchical precision (hP^{μ}) and recall (hR^{μ}) evolve on the WIPO dataset as more data instances become available. Despite some variations due to different starting conditions (i.e., different data instances early on), OHMC seems to converge as more data instances becomes available. The results illustrate that although the order by which training data instances become available indeed has an impact on the evolution of $\mathbf{W}_t$, such impact is not significant from the overall training perspective.

In summary, the experimental results indicate that incorporating the hierarchy constraint directly into the online learning process is beneficial, particularly for datasets with complex hierarchical structure (e.g., WIPO).

VII. CONCLUSION

We presented OHMC, a new algorithm for online hierarchical multi–label classification. Specifically, we first introduced a new hierarchy constraint to describe the entire hierarchical label structure instead of local parent–child relationship. Next, we formulated the task of online hierarchical multi–labeled classification as an online optimization problem. We subsequently derived a closed form update for the case of

linear classifiers. Our experiments on three real–world datasets demonstrated the effectiveness of the proposed approach compared with both online multi–label classification algorithms and offline hierarchical classification frameworks.

In future work, we plan to extend our algorithm to non-linear classifiers so as to achieve better classification results, particularly in image datasets, such as ImageCLEF07A and ImageCLEF07D. We additionaly plan to regularize the potential deviation of predictor weights at successive training rounds to ensure temporal smoothness.

REFERENCES

[1] X. Shen, M. Boutell, J. Luo, and C. Brown, "Multilabel machine learning and its application to semantic scene classification," in *Storage and Retrieval Methods and Applications for Multimedia 2004*, vol. 5307. SPIE, 2003, pp. 188–199.

[2] G. Yu, C. Domeniconi, H. Rangwala, G. Zhang, and Z. Yu, "Transductive multi-label ensemble classification for protein function prediction," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1077–1085.

[3] D. Senthilkumar and C. Akshayaa, "Efficient deep learning approach for multi-label semantic scene classification," in *International Conference on Image Processing and Capsule Networks*. Springer, 2020, pp. 397–410.

[4] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine learning*, vol. 39, no. 2, pp. 135–168, 2000.

[5] E. L. Mencía and J. Furnkranz, "Pairwise learning of multilabel classifications with perceptrons," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 2899–2906.

[6] C. H. Park and M. Lee, "On applying linear discriminant analysis for multi-labeled problems," *Pattern recognition letters*, vol. 29, no. 7, pp. 878–887, 2008.

[7] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in *2008 eighth IEEE international conference on data mining*. IEEE, 2008, pp. 995–1000.

[8] D. Ghosh and A. M. Chinnaiyan, "Classification and selection of biomarkers in genomic data using lasso," *Journal of Biomedicine and Biotechnology*, vol. 2005, no. 2, p. 147, 2005.

[9] S. R. Sahoo and B. B. Gupta, "Multiple features based approach for automatic fake news detection on social networks using deep learning," *Applied Soft Computing*, vol. 100, p. 106983, 2021.

[10] S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," *Neurocomputing*, vol. 459, pp. 249–289, 2021.

[11] T. Zhai, H. Tang, and H. Wang, "Adaptive label thresholding methods for online multi-label classification," *arXiv preprint arXiv:2112.02301*, 2021.

[12] C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1-2, pp. 31–72, 2011.

[13] J. Fan, H. Luo, Y. Gao, and R. Jain, "Incorporating concept ontology for hierarchical video classification, annotation, and visualization," *IEEE Transactions on Multimedia*, vol. 9, no. 5, pp. 939–957, 2007.

[14] I. Dimitrovski, D. Kocev, S. Loskovska, and S. Džeroski, "Hierarchical annotation of medical images," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2436–2449, 2011.

[15] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *Journal of machine learning research*, vol. 5, no. Apr, pp. 361–397, 2004.

[16] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes, "Hdltex: Hierarchical deep learning for text classification," in *2017 16th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2017, pp. 364–371.

[17] A. Freitas and A. Carvalho, "A tutorial on hierarchical classification with applications in bioinformatics," in *Research and trends in data mining technologies and applications*. IGI Global, 2007, pp. 175–208.

[18] N. Cesa-Bianchi and G. Valentini, "Hierarchical cost-sensitive algorithms for genome-wide gene function prediction," in *Machine Learning in Systems Biology*, 2009, pp. 14–29.

[19] A. Sokolov and A. Ben-Hur, "Hierarchical classification of gene ontology terms using the gostruct method," *Journal of bioinformatics and computational biology*, vol. 8, no. 02, pp. 357–376, 2010.

[20] W. Bi and J. T. Kwok, "Hierarchical multilabel classification with minimum bayes risk," in *2012 IEEE 12th International Conference on Data Mining*. IEEE, 2012, pp. 101–110.

[21] P. N. Robinson, M. Frasca, S. Köhler, M. Notaro, M. Re, and G. Valentini, "A hierarchical ensemble method for dag-structured taxonomies," in *International Workshop on Multiple Classifier Systems*. Springer, 2015, pp. 15–26.

[22] C. Chelmis and W. Qi, "Hierarchical multiclass adaboost," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 5063–5070.

[23] M.-L. Zhang, Y.-K. Li, X.-Y. Liu, and X. Geng, "Binary relevance for multi-label learning: an overview," *Frontiers of Computer Science*, vol. 12, no. 2, pp. 191–202, 2018.

[24] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE transactions on knowledge and data engineering*, vol. 23, no. 7, pp. 1079–1089, 2010.

[25] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *European conference on machine learning*. Springer, 2007, pp. 406–417.

[26] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," *Advances in neural information processing systems*, vol. 14, 2001.

[27] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[28] ——, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[29] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine learning*, vol. 37, no. 3, pp. 297–336, 1999.

[30] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[31] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, p. 551–585, dec 2006.

[32] S. Ding, B. Mirza, Z. Lin, J. Cao, X. Lai, T. V. Nguyen, and J. Sepulveda, "Kernel based online learning for imbalance multiclass classification," *Neurocomputing*, vol. 277, pp. 139–148, 2018.

[33] M. Herbster, "Learning additive models online with fast evaluating kernels," in *International Conference on Computational Learning Theory*. Springer, 2001, pp. 444–460.

[34] Y. Li and P. Long, "The relaxed online maximum margin algorithm," *Advances in neural information processing systems*, vol. 12, 1999.

[35] X. Kong and S. Y. Philip, "An ensemble-based approach to fast classification of multi-label data streams," in *7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*. IEEE, 2011, pp. 95–104.

[36] J. Du and C.-M. Vong, "Robust online multilabel learning under dynamic changes in data distribution with labels," *IEEE transactions on cybernetics*, vol. 50, no. 1, pp. 374–385, 2019.

[37] R. Venkatesan, M. J. Er, M. Dave, M. Pratama, and S. Wu, "A novel online multi-label classifier for high-speed streaming data applications," *Evolving Systems*, vol. 8, no. 4, pp. 303–315, 2017.

[38] J. Read, A. Bifet, G. Holmes, and B. Pfahringer, "Scalable and efficient multi-label classification for evolving data streams," *Machine Learning*, vol. 88, no. 1, pp. 243–272, 2012.

[39] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," *Machine learning*, vol. 73, no. 2, pp. 185–214, 2008.

[40] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Incremental algorithms for hierarchical classification," *Journal of Machine Learning Research*, vol. 7, no. Jan, pp. 31–54, 2006.

[41] A. R. S. Parmezan, V. Souza, and G. E. Batista, "Towards hierarchical classification of data streams," in *Iberoamerican Congress on Pattern Recognition*. Springer, 2018, pp. 314–322.

[42] J.-Y. Park and J.-H. Kim, "Incremental class learning for hierarchical classification," *IEEE transactions on cybernetics*, vol. 50, no. 1, pp. 178–189, 2018.

[43] D. Koller and M. Sahami, "Hierarchically classifying documents using very few words," Stanford InfoLab, Tech. Rep., 1997.
[44] N. Ketkar, "Stochastic gradient descent," in *Deep learning with Python*. Springer, 2017, pp. 113–132.
[45] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," *Advances in neural information processing systems*, vol. 26, 2013.
[46] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor, "Kernel-based learning of hierarchical multilabel classification models," *Journal of Machine Learning Research*, vol. 7, pp. 1601–1626, 2006.
[47] B. Taskar, C. Guestrin, and D. Koller, "Max-margin markov networks," *Advances in neural information processing systems*, vol. 16, 2003.
[48] J.-Y. Park and J.-H. Kim, "Online incremental hierarchical classification resonance network," *Pattern Recognition*, vol. 111, p. 107672, 2021.
[49] L. Schietgat, C. Vens, J. Struyf, H. Blockeel, D. Kocev, and S. Džeroski, "Predicting gene function using hierarchical multi-label decision tree ensembles," *BMC bioinformatics*, vol. 11, no. 1, pp. 1–14, 2010.
[50] C. N. Silla Jr and A. A. Freitas, "A global-model naive bayes approach to the hierarchical prediction of protein functions," in *2009 Ninth IEEE International Conference on Data Mining*. IEEE, 2009, pp. 992–997.