

# Multi-label Classification under Uncertainty: A Tree-based Conformal Prediction Approach

**Chhavi Tyagi**

*Department of Mathematical Sciences  
New Jersey Institute of Technology  
Newark, NJ 07102, USA*

CT364@NJIT.EDU

**Wenge Guo**

*Department of Mathematical Sciences  
New Jersey Institute of Technology  
Newark, NJ 07102, USA*

WENGE.GUO@NJIT.EDU

**Editors:** Harris Papadopoulos, Khuong An Nguyen, Henrik Boström and Lars Carlsson

## Abstract

Multi-label classification is a common challenge in various machine learning applications, where a single data instance can be associated with multiple classes simultaneously. The current paper proposes a novel tree-based method for multi-label classification using conformal prediction and multiple hypothesis testing. The proposed method employs hierarchical clustering with labelsets to develop a hierarchical tree, which is then formulated as a multiple-testing problem with a hierarchical structure. The split-conformal prediction method is used to obtain marginal conformal  $p$ -values for each tested hypothesis, and two *hierarchical testing procedures* are developed based on marginal conformal  $p$ -values, including a hierarchical Bonferroni procedure and its modification for controlling the family-wise error rate. The prediction sets are thus formed based on the testing outcomes of these two procedures. We establish a theoretical guarantee of valid coverage for the prediction sets through proven family-wise error rate control of those two procedures. We demonstrate the effectiveness of our method in a simulation study and two real data analysis compared to other conformal methods for multi-label classification.

**Keywords:** Multi-label Classification, Conformal Prediction, Hierarchical Tree, Multiple-Testing, Family-wise Error Rate

## 1. Introduction

Most machine learning research on classification deals with  $2n$  instances as  $\{(X_j, Y_j)\}_{j=1}^{2n}$  with features  $X_j \in \mathbb{R}^d$  and a response variable  $Y_j \in \mathcal{Y} = \{1, \dots, K\}$ . Each feature vector  $X_j$  is associated with a single response  $Y_j$ . As opposed to this standard setting, in multi-label classification, each instance can belong to multiple classes,  $Y_j = (Y_j^1, \dots, Y_j^c)$  is the response vector for  $j^{\text{th}}$  observation with  $c$  labels. There are many real-world problems in which such a setting is natural. For instance, in medical diagnosis, a patient may be suffering from diabetes and prostate cancer simultaneously. Another example is the problem of semantic scene classification, where multiple class labels can describe a natural scene (e.g., mountain, sea, tree, sun etc.)

Many multi-label classification methods have been developed like Binary Relevance (BR) (Boutell et al., 2004), classifier chains (Read et al., 2009) and Label Powerset (LP)

([Tsoumakas and Katakis, 2007](#); [Boutell et al., 2004](#)) using machine learning algorithms like logistic regression, SVM etc. Given the growing utilization of machine learning methods and the heightened risk of making erroneous decisions, it is crucial to develop techniques for accurately estimating the uncertain nature of their predictions. For example, in the field of medical imaging, a multi-label classifier may be employed to recognize various conditions in a single scan, such as identifying the presence of a tumor and its level of malignancy. If the classifier makes an incorrect prediction, it could lead to misdiagnosis and subsequent incorrect treatment, which can have serious consequences for the patient.

Therefore, it is essential for the classifier to provide not only an estimation of the most probable outcome but also a quantification of uncertainty that is actionable, such as a set of predictions that can be proven to contain the true diagnosis with a high level of confidence (for example, 90%). Conformal prediction (CP) is a popular method developed for uncertainty quantification. Importantly, conformal prediction is a distribution-free framework. This means that it does not rely on any assumptions about the distribution of the data or the model being used. Only a few works have been done to accommodate statistical guarantees using conformal prediction for multi-label classification. Among the existing approaches are the binary relevance multi-label conformal predictor (BR-MLCP) ([Lambrou and Papadopoulos, 2016](#)) and the label powerset multi-label conformal predictor (PS-MLCP1) ([Papadopoulos, 2014](#)). PS-MLCP1 has been applied to text classification tasks ([Paisios et al., 2019](#)) and a modified version has been developed by [Lysimachos et al. \(2022\)](#) to handle large number of labels. These procedures do not accommodate dependencies among the multiple response variables and are computationally inefficient for large number of response variables. Additionally, in multi-label classification, the response variables (i.e., the class labels) can often have missing information, i.e., with  $c$  labels and  $2^c$  possible labelsets, not all labelsets are present in the data. This can have significant implications for the accuracy and reliability of the predictions made by existing methods. In this paper, we develop a tree-based conformal prediction method for multi-label classification using multiple hypothesis testing. The method is not only able to have quantification of uncertainty with statistical guarantees but also takes into account of any missing information in the response variables. Since it uses a tree-based approach, the method is computationally efficient for large number of labels and considers label dependence. Thus, we have created a classification approach that delivers precise prediction sets with statistical coverage guarantee.

The rest of the paper is organized as follows. With conformal prediction framework and related works discussed in section 2, we provide notations and describe our proposed tree-based procedure (*TB*) in section 3. The statistical properties of the procedure are analyzed in Section 4. In section 5, we discuss our proposed procedures developed to account for missing information. Numerical findings from the simulation study and real data analysis are presented in section 6. We conclude the paper in section 7 with some remarks on the present work and brief discussions on some future research topics. We defer the proofs of the propositions in section 4 to the appendix A.

## 2. Background and related work

### 2.1. Conformal Prediction

Conformal prediction (CP) is a statistical learning framework that provides a prediction set  $\hat{C}(X_{2n+1})$ , instead of point predictions with statistical guarantees for a given test instance  $X_{2n+1}$ .

Given dataset  $\mathcal{D} \equiv \{(X_i, Y_i)\}_{i=1}^{2n} \equiv \{Z_i\}_{i=1}^{2n}$  with  $X_i \in \mathcal{X}$ ,  $\mathcal{X} \subset \mathbb{R}^d$  is input of real valued attributes and  $Y_i \in \mathcal{Y} = \{0, 1\}$ , where  $(X_{2n+1}, Y_{2n+1})$  and  $\{(X_i, Y_i)\}_{i=1}^{2n}$  are drawn exchangeably from  $P_{XY}$ . In general,  $\mathcal{Y}$  can take on different forms, including multi-class outcomes with  $\mathcal{Y} = \{1, 2, \dots, K\}$  or continuous outcomes with  $\mathcal{Y} \in \mathbb{R}$ s. The goal of CP is to construct a prediction set  $\hat{C}(X_{2n+1})$  that contains  $Y_{2n+1}$  with probability at least  $(1 - \alpha)$ , where  $\alpha$  is user-defined significance level such that  $\alpha \in (0, 1)$ .

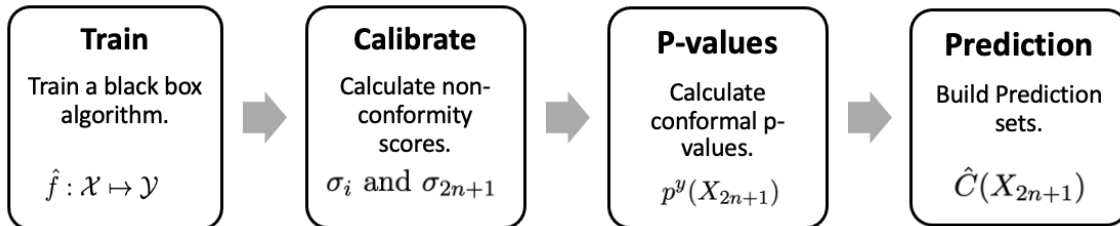


Figure 1: Steps of Conformal Prediction.

Figure 1 summarizes the main steps involved in the split-conformal prediction framework, starting from training the model to the final prediction set. The initial step involves splitting the data into two parts, i.e., proper training  $\mathcal{D}_{tr} \equiv \{(X_i, Y_i)\}_{i=1}^n$  and calibration  $\mathcal{D}_{cal} \equiv \{(X_i, Y_i)\}_{i=n+1}^{2n}$ . The first step involves training the model using a machine learning algorithm mapping features  $X$  to a real attribute  $Y$ . The next step, calibration is a crucial component of the conformal prediction framework as it defines a function (called as *non-conformity score*) that measures how well the true response value  $y$  conforms with the predictions of our fitted model  $\hat{f}$ . Given the score function  $\sigma : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ , the non-conformity score for observations in the calibration set is given by:

$$\sigma_i = \sigma(X_i, Y_i), \quad i = n + 1, \dots, 2n \quad (1)$$

For a given test observation,  $X_{2n+1}$ , we compute the non-conformity score for each  $y \in \mathcal{Y}$  as

$$\sigma_{2n+1}^y = \sigma(X_{2n+1}, y) \quad (2)$$

In our specific case, we use the non-conformity score as 1-predicted probability i.e.,

$$\sigma_i = 1 - \hat{f}(Y_i|X_i), \quad i = n + 1, \dots, 2n \quad (3)$$

$$\sigma_{2n+1}^y = 1 - \hat{f}(y|X_{2n+1}), \quad \text{for each } y \in \mathcal{Y} \quad (4)$$

The third step showcases to calculate conformal  $p$ -values as given in equation (5), which provides a way to quantify the confidence in the predictions

$$p^y(X_{2n+1}) = \frac{\sum_{i=n+1}^{2n} \{\mathbb{I}(\sigma_i < \sigma_{2n+1}^y) + U_{2n+1} \cdot \mathbb{I}(\sigma_i = \sigma_{2n+1}^y)\}}{n} \quad (5)$$

where  $U_{2n+1} \sim \text{Unif}(0, 1)$  is random variable independent of  $\sigma_i$ 's introduced to break ties. The last panel illustrates to calculate prediction sets using the conformal  $p$ -values provided by equation (6)

$$\hat{C}(X_{2n+1}) = \{y \in \mathcal{Y} : p^y(X_{2n+1}) \geq \alpha\} \quad (6)$$

The prediction set constructed using the conformal  $p$ -values in equation (5) is subject to the guarantee of *marginal coverage*, as expressed in Equation (7), which states that the probability of the true label of a new instance falling within the prediction set is at least  $1 - \alpha$ .

$$\mathbb{P}(Y_{2n+1} \in \hat{C}(X_{2n+1})) \geq 1 - \alpha \quad (7)$$

## 2.2. Related work

Conformal prediction is a statistical framework that was first proposed by Vovk et al. (1999, 2005); Shafer and Vovk (2008). The literature on conformal prediction has identified two main types of conformal prediction methods, namely full conformal prediction (transductive) and split-conformal prediction (inductive) Vovk et al. (2005), which is introduced in section 2.1. Full conformal prediction involves using all the training observations to calculate the conformal  $p$ -values, which can become computationally expensive. On the other hand, split-conformal prediction is a hold-out method where part of the data is reserved for training and the remaining are used for calibration, which may result in less accurate estimates of  $\hat{f}$  than if all the data had been used for estimation but much computationally efficient. The framework has been widely used in regression problems Romano et al. (2019); Johansson et al. (2014); Izbicki et al. (2019); Gupta et al. (2022); Lei and Wasserman (2014) and classification problems Balasubramanian et al. (2014); Vovk et al. (2003); Lei (2014); Sadinle et al. (2019); Romano et al. (2020). A detailed introduction to the conformal prediction framework is provided in Angelopoulos and Bates (2022). Our work primarily builds on the use of split-conformal prediction (Vovk et al., 2005). Other types of conformal prediction methods include cross-conformal prediction Vovk (2015) and CV+/Jackknife+ Barber et al. (2021).

The most relevant works in the context of multi-label classification problems using conformal prediction include instance-based approach (Wang et al., 2014), Binary Relevance multi-label conformal predictor (BR-MLCP) (Lambrou and Papadopoulos, 2016) and Power set multi-label conformal predictor (PS-MLCP1) (Papadopoulos, 2014). PS-MLCP1 has been applied to text classification (Paisios et al., 2019) and an improved version of PS-MLCP1 has been developed by Lysimachos et al. (2022) to handle large number of labels. BR-MLCP uses a one-against-all method to decompose the multi-label dataset into multiple single-label binary classification problems. However, this method does not consider the dependency among the labels. PS-MLCP1 efficiently accounts for the dependency among the labels but does not consider any missing information on the labels and also runs into computational issues for large number of labels. Wang et al. (2015) compares instance-based

MLCP, BR-MLCP and PS-MLCP methods in terms of prediction efficiency. [Cauchois et al. \(2021\)](#) developed a tree-structured method to consider the interaction among the labels. Another framework that addresses multi-label classification problems using conformal prediction is risk control introduced by [Bates et al. \(2021\)](#); [Angelopoulos et al. \(2022\)](#). [Bates et al. \(2021\)](#) proposed a method for constructing distribution-free prediction sets that control the risk of incorrect predictions by introducing a new loss function that directly controls the risk of incorrect predictions. Specifically, their approach minimizes the expected loss over the prediction sets, subject to the constraint that the coverage probability is at least as large as the specified level of significance. [Angelopoulos et al. \(2022\)](#) proposed a new method for controlling the risk of prediction sets, which is highly relevant for multi-label classification problems in the field of conformal prediction. Their approach, called Conformal Risk Control (CRC), uses a novel loss function to optimize the trade-off between coverage and size of the prediction sets.

In contrast to traditional conformal prediction methods, CRC aims to minimize the overall risk of incorrect predictions by taking into account the cost of making errors.

There are also several statistical extensions to conformal prediction, including the ideas of non-exchangeability like time series by [Xu and Xie \(2021\)](#), outlier testing by [Bates et al. \(2022\)](#), covariate shift by [Tibshirani et al. \(2019\)](#), and image classifiers by [Angelopoulos et al. \(2021\)](#).

### 3. Proposed Tree-based multi-label Conformal Prediction Method (*TB*)

#### 3.1. Notations

Consider the data set  $\mathcal{D} = \{(X_j, Y_j)\}_{j=1}^{2n} = \{Z_j\}_{j=1}^{2n}$ , with  $X_j \in \mathcal{X}, \mathcal{X} \subset \mathbb{R}^d$  is input of real valued attributes and  $Y_j = (Y_j^1, \dots, Y_j^c)$  is the response vector for  $j^{th}$  observation. Each  $Y_j^i \in \{0, 1\}$  denotes the label of the  $i^{th}$  class in the  $j^{th}$  observation, which is labeled as 1 if the instance belongs to the  $i^{th}$  class and 0 otherwise. Let  $(X_{2n+1}, Y_{2n+1})$  be the test observation. The goal is to provide prediction set  $\hat{C}(X_{2n+1})$  for unobserved  $Y_{2n+1}$  based on a given test instance  $X_{2n+1}$  to show the statistical guarantee on the prediction, given a significance level  $\alpha \in (0, 1)$ .

#### 3.2. Procedure

Our proposed procedure consists of five main steps. Firstly, a hierarchical tree is developed based on all possible labelsets obtained from  $c$  labels. Secondly, we formulate a hypothesis for each node of the tree. Since we have multiple nodes on the tree, it is a multiple hypothesis testing (MHT) problem with hierarchical structure. Thirdly, we compute conformal  $p$ -values for each tested hypothesis using the split-conformal prediction method and any machine learning algorithm. Fourthly, we develop two types of *hierarchical testing procedures* based on conformal  $p$ -values for controlling family-wise error rate (FWER). Finally, we form the prediction set from the test outcomes of the previous step by leveraging split-conformal prediction.

## 3.2.1. HIERARCHICAL TREE

In this section, we present a method for constructing a hierarchical tree for multi-label classification by utilizing hierarchical clustering on the binary representation of all possible labelsets, denoted by  $s_i \in \{0, 1\}^c, i = 1, \dots, 2^c$ . The use of hierarchical clustering is motivated by the need to handle a large number of labels in multi-label classification and to reduce the number of labelsets used for classification while preserving an elegant and manageable tree structure.

To build the hierarchical tree, we need to define a distance metric and linkage criteria between the labelsets. For our purposes, we use the Hamming distance (Tamasauskas et al., 2012), which is a widely used metric in the binary setting, to calculate the dissimilarity,  $d_{ij}$ , between each pair of labelsets  $s_i$  and  $s_j$ , for  $i, j = 1, 2, \dots, l$ . Given two binary vectors  $s_1$  and  $s_2$  with  $c$  elements each, the Hamming distance between them is defined as:

$$d_H(s_1, s_2) = \sum_{i=1}^c \mathbb{I}\{s_{1i} \neq s_{2i}\} \quad (8)$$

where  $s_j = (s_{j1}, \dots, s_{jc})$ , for  $j = 1, 2$ . Thus, the Hamming distance counts the number of mismatches between two binary vectors.

To calculate the distance among the clusters of labelsets, we use the complete linkage method (Tamasauskas et al., 2012). In complete linkage, the distance between any two clusters  $C_K$  and  $C_L$  is defined as the maximum distance between two observations that belong to different clusters,

$$D_{KL} = \max_{s \in C_K, t \in C_L} d(s, t). \quad (9)$$

The advantage of using complete linkage is that it is less sensitive to noise and outliers compared to other linkage methods such as single linkage or centroid linkage. The steps for constructing a hierarchical tree are outlined in Algorithm 1.

---

**Algorithm 1:** Hierarchical Clustering
 

---

**Input:** Labelsets  $s_i \in \{0, 1\}^c, i = 1, \dots, 2^c$

**Output:** Hierarchical Tree

*Cluster size*  $\leftarrow 2^c$ .

*// Each labelset is a cluster.*

Initialize a hierarchical tree  $T$  with  $2^c$  leaves, one for each labelset.

Compute pairwise distances ( $d_{ij}$ ) between labelsets  $s_i, s_j$  as defined in equation (8) and store in a matrix.

**while** *Cluster size*  $> 1$  **do**

    Find the pair of nodes with the smallest distance from the matrix formed.

    Merge the two nodes into a new parent node, and add edges to connect the parent node to the two children.

*// Reduce number of clusters by 1.*

    Compute the distance between the parent node and all other nodes in  $T$  from equation (9).

**end**

---

In Figure 2, we illustrate the hierarchical tree for the case of 3 labels. In this tree, each node represents a subset of the labelset space. We use the terms “parent” and “children” to describe the relationships between nodes in the tree. The parent of a node is the immediate ancestor in the tree, and its children are the immediate descendants. For example, if we select node  $S_{21}$  from the tree, its parent would be  $S_{11}$  which contains the labelset  $\{0, 1, 2, 3\}$ , and its children would be  $S_{31}$  and  $S_{32}$  which contain the labelsets  $\{0, 1\}$  and  $\{2, 3\}$  respectively. This terminology describes the labelsets contained within the parent and children nodes in decimal form (i.e., base 10). Specifically, the parent node contains a larger subset of labelset than its children nodes. The depth of a hierarchical tree is defined as the number of layers in the tree, where the root node is at layer 0 and the leaf nodes are at the maximum layer. Each layer consists of nodes with the same depth. As an illustration, in Figure 2, node  $S_{11}$  and  $S_{12}$  are located at depth 1, and all 8 leaf nodes are at the maximum depth of 3. Nodes at depth 0 do not have parents and nodes at the maximum depth of the tree (i.e., leaf nodes) do not have children.

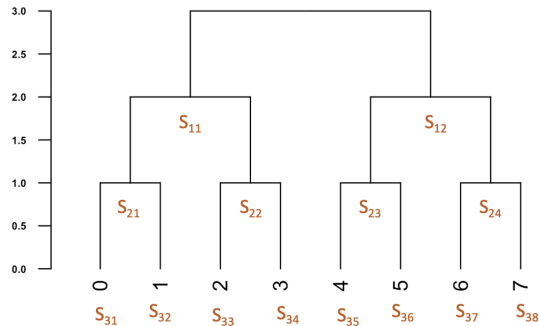


Figure 2: Hierarchical tree with  $c = 3$  labels.

**Remark 1** In the hierarchical tree, the union of all nodes in a layer covers the entire response space,  $Y$ , i.e., for each layer  $i = 1, \dots, c$ , we have  $\cup_{k=1}^{n_i} S_{ik} = Y$ , where  $n_i$  is the number of nodes in  $i^{\text{th}}$  layer. Additionally, for each pair of distinct nodes  $S_{ik}$  and  $S_{ik'}$  in the same layer  $i$ , their intersection is empty, i.e.,  $S_{ik} \cap S_{ik'} = \phi$ .

### 3.2.2. MULTIPLE HYPOTHESIS TESTING PROBLEM

In this section, we formulate the multi-label classification problem as a multiple hypothesis testing (MHT) problem based on the hierarchical tree established in Section 3.2.1. For each node in the tree, we construct a hypothesis. Since we have multiple nodes in the tree, it becomes a multiple hypothesis testing (MHT) problem. The hypotheses are constructed such that under the null hypothesis, the test label belongs to the subset of labelsets corresponding to that node; under the alternative hypothesis, the test label does not belong to the subset of labelsets corresponding to that node.

We construct the hypothesis layer by layer, starting from the second layer (i.e., depth 1) and proceeding until the leaves of the tree (i.e., maximum depth). For each layer  $i$ ,  $i = 1, \dots, c$ , assume  $n_i$  as the number of nodes in each layer. The structure for constructing these hypotheses is discussed below.

$i^{\text{th}}$  Layer: Layer  $i$  corresponds to the depth  $i$  on the tree in Figure 2. Then the hypotheses are as follows:

$$\begin{aligned} H_{i1} : Y_{2n+1} \in \mathcal{S}_{i1} \text{ versus } H'_{i1} : Y_{2n+1} \notin \mathcal{S}_{i1} \\ H_{i2} : Y_{2n+1} \in \mathcal{S}_{i2} \text{ versus } H'_{i2} : Y_{2n+1} \notin \mathcal{S}_{i2} \\ \vdots \\ H_{in_i} : Y_{2n+1} \in \mathcal{S}_{in_i} \text{ versus } H'_{in_i} : Y_{2n+1} \notin \mathcal{S}_{in_i} \end{aligned}$$

This procedure is repeated until we reach the tree's leaf layer, i.e.,  $i = c$ .

*Mathematical properties of the constructed hypotheses:*

1. Due to the structural relationship among the nodes of the tree, in each layer  $i$ , exactly one hypothesis is true.
2. If a children hypothesis is true, then its parent hypothesis is also true. Conversely, if a parent hypothesis is false, then its children's hypothesis are automatically false.

### 3.2.3. CONFORMAL P-VALUES

This section describes the method to compute the conformal  $p$ -values for the hypotheses constructed in Section 3.2.2. First, the data is split into training, calibration, tuning and test data as  $\mathcal{D}_{tr}$ ,  $\mathcal{D}_{cal}$ ,  $\mathcal{D}_{tune}$  and  $\mathcal{D}_{test}$  respectively using split-conformal prediction. We use an equal splitting between  $\mathcal{D}_{tr}$  and  $\mathcal{D}_{cal}$  for simplicity, but it is flexible and can be adjusted as needed. Then,  $\mathcal{D}_{tr}$  and  $\mathcal{D}_{cal}$  are transformed from  $(X_j, Y_j)$  to  $(\tilde{X}_j, \tilde{Y}_j)$  for each layer  $i$  as described below.

$$\tilde{X}_j = X_j, \text{ and } \tilde{Y}_j = k, \text{ if } Y_j \in S_{ik}, \tag{10}$$

where  $k = 1, 2, \dots, n_i$ . Thus, each layer is transformed as a multi-class classification problem with transformed data as  $\tilde{\mathcal{D}}_{tr}$  and  $\tilde{\mathcal{D}}_{cal}$ . Algorithm 2 describes the general procedure to calculate conformal  $p$ -values for each node on the tree using split-conformal prediction, which includes tuning set,  $\mathcal{D}_{tune}$  used later in Algorithm 4 for tuning parameter  $\lambda^*$ .

In addition to the procedure described above, there are alternative approaches to computing conformal  $p$ -values. One such approach involves transforming the problem into multiple multi-class classification problems for each layer. For instance, if nodes in a given layer share the same parent, a multi-class classification transformation is applied to those nodes. This alternative approach provides an alternate way to compute conformal  $p$ -values and may interest researchers seeking to explore different methods for solving this problem.

### 3.2.4. CONTROLLING FWER

In multiple-testing problems, control of some overall error rate such as the family-wise error rate (FWER) and the false discovery rate (FDR) (Hochberg and Tamhane, 1987), is crucial.

We use a *hierarchical testing procedure* to test the hypotheses formulated in section 3.2.2. Tree-based constructed hypotheses  $H_{ik}$  are tested layer by layer with critical value  $\alpha_i$  for  $i^{\text{th}}$  layer and  $k^{\text{th}}$  node,  $i = 1, \dots, c$  and  $k = 1, \dots, n_i$  such that  $\sum_{i=1}^c \alpha_i \leq \alpha$ . For each layer  $i$ , if  $p^{i,k} < \alpha_i$ , we reject the hypothesis and also reject all its children hypotheses and move to the next layer until the leaf layer. Our proposed procedure focuses on controlling



*FWER*, which is defined as the probability of not making any Type I error. It implies we do not allow making any Type I error. We consider the following two specific *hierarchical testing procedures* for controlling the family-wise error rate (FWER).

1. **Procedure 1:** In this *hierarchical testing procedure*, the significance level for each layer is set to  $\alpha_i = \alpha/c$ . This method is a Bonferroni procedure for hierarchical testing. It is known to be relatively less powerful.
2. **Procedure 2:** In this *hierarchical testing procedure*, the significance level for each layer is set to  $\alpha_i = \lambda\alpha$ , where  $\lambda$  is the tuning parameter. We use Algorithm 4 to perform hyperparameter tuning using binary search and find the optimal value of  $\lambda$ , denoted as  $\lambda^*$  s.t.  $\alpha_i = \lambda^*\alpha$ , which helps to overcome the conservativeness of the first method.

Algorithm 3 outlines the steps involved in implementing the FWER control procedure.

---

**Algorithm 2:** Conformal  $p$ -values

---

**Input:**  $\mathcal{D} = \{(X_j, Y_j)\}_{j=1}^{2n}$ , test observation  $X_{2n+1}$ , algorithm  $\mathcal{B}$ .

**Output:** Conformal  $p$ -values:  $p_{tune}^{i,k}(X_l)$  and  $p_{test}^{i,y}(X_{2n+1})$ .

Randomly split the data into 3 equal disjoint subsets,  $\mathcal{D}_{tr}$ ,  $\mathcal{D}_{cal}$  and  $\mathcal{D}_{tune}$  with  $n_1, n_2, n_3$  observations in each set respectively s.t.  $n_1 + n_2 + n_3 = 2n$ .

**forall**  $i \in \{1, 2, \dots, c\}$  **do**

Obtain  $\tilde{\mathcal{D}}_{tr}$  and  $\tilde{\mathcal{D}}_{cal}$  for each layer as in equation (10).

Train  $\mathcal{B}$  on all samples in  $\tilde{\mathcal{D}}_{tr}$ ,  $\hat{f} : \mathcal{B}(\{(X_j, Y_j)\}_{j \in \tilde{\mathcal{D}}_{tr}})$

**forall**  $k \in \{1, 2, \dots, n_i\}$  **do**

Calculate the non-conformity scores for  $i^{th}$  layer,  $k^{th}$  node, and  $j^{th}$  observation on  $\tilde{\mathcal{D}}_{cal}$  as  $\sigma_{cal}^{ik}(X_j), j = 1, \dots, n_2$  using equation (3).

Calculate the non-conformity scores for  $i^{th}$  layer,  $k^{th}$  node, and  $l^{th}$  observation on  $\tilde{\mathcal{D}}_{tune}$  as  $\sigma_{tune}^{ik}(X_l), l = 1, \dots, n_3$  using equation (3).

Given  $(X_l, Y_l) \in \mathcal{D}_{tune}$ , conformal  $p$ -values for  $i^{th}$  layer,  $k^{th}$  node and  $l^{th}$  observation is,

$$p_{tune}^{i,k}(X_l) = \frac{\sum_{j=1}^{n_2} \{\mathbb{I}(\sigma_{cal}^{ik}(X_j) < \sigma_{tune}^{ik}(X_l)) + U_{n_2+1} \cdot \mathbb{I}(\sigma_{cal}^{ik}(X_j) = \sigma_{tune}^{ik}(X_l))\}}{n_2 + 1}, \quad (11)$$

Given  $X_{2n+1} \in \mathcal{D}_{test}$  and  $y = k$ , we form  $(X_{2n+1}, y)$ , compute the non-conformity score  $\sigma_{test}^{i,y}(X_{2n+1})$  from equation (4) and corresponding conformal  $p$ -value as

$$p_{test}^{i,y}(X_{2n+1}) = \frac{\sum_{j=1}^{n_2} \{\mathbb{I}(\sigma_{cal}^{ik}(X_j) < \sigma_{test}^{i,y}(X_{2n+1})) + U_{n_2+1} \cdot \mathbb{I}(\sigma_{cal}^{ik}(X_j) = \sigma_{test}^{i,y}(X_{2n+1}))\}}{n_2 + 1} \quad (12)$$

**end**

**end**

---

**Remark 2** In Algorithm 3, to utilize the first procedure of FWER control, we replace  $\tilde{\alpha}_i$  with  $\alpha_i = \alpha/c$  and there is no need for  $\mathcal{D}_{tune}$ . If the second procedure is being employed, we substitute  $\tilde{\alpha}_i$  with  $\alpha_i = \lambda^* \alpha$ .

---

**Algorithm 3:** FWER controlling procedure for MHT

---

**Input:**  $\alpha \in (0, 1)$ ,  $\tilde{\alpha}_i$ ,  $i = 1, 2, \dots, c$  s.t.  $\sum_{i=1}^c \tilde{\alpha}_i \leq \alpha$ .

**Output:**  $\mathcal{S}(\alpha)$ : Set of accepted hypotheses in the leaf layer.

**while** layer,  $i < c$  **do**

**for**  $k \in \{1, \dots, n_i\}$  **do**

        For  $y = k$ , compute conformal  $p$ -values for each node  $y$  in the  $i^{th}$  layer as  $p^{i,y}(X_{2n+1})$ , using Algorithm 1.

        Test  $H_{ik}$ , in layer  $i$  at  $\tilde{\alpha}_i$ .

        Reject  $H_{ik}$  and all its children's hypotheses if  $p^{i,y}(X_{2n+1}) < \tilde{\alpha}_i$ .

**end**

**end**

For layer  $i = c$  and  $k = 1, \dots, n_i$ , Reject  $H_{ik}$  if  $p^{i,y}(X_{2n+1}) < \tilde{\alpha}_i$ .

---

### 3.2.5. PREDICTION SET

To construct the prediction set, we utilize the output of Algorithm 3, which provides  $\mathcal{S}(\alpha)$  as the set of accepted hypotheses in the leaf layer. The prediction set includes all labels for which the corresponding leaf hypotheses are not rejected at level  $\alpha$ . The prediction set for  $\mathcal{D}_{test} = X_{2n+1}$  is given by:

$$\hat{C}_{test}(X_{2n+1}, \alpha) = \{y : \text{Labelset } y \text{ for which leaf hypothesis } \in \mathcal{S}(\alpha)\}, \quad (14)$$

**Split-Conformal Prediction Set:** Algorithm 5 outlines our proposed method, Tree-Based Multi-Label Conformal Prediction (*TB*) with split-conformal method. For simplicity, we divide the data into three equal subsets, namely  $\mathcal{D}_{tr}$ ,  $\mathcal{D}_{cal}$  and  $\mathcal{D}_{tune}$ . However, in practice, we can change the splitting ratio for better results.

---

**Algorithm 5:** Split-Conformal  $\lambda^*$  TB

---

**Input:**  $\mathcal{D} = (X_j, Y_j)_{j=1}^{2n}$ ,  $\mathcal{D}_{test} = X_{2n+1}$ , algorithm  $\mathcal{B}$ , number of labels  $c$ .

**Output:** Prediction set,  $\hat{C}_{test}(X_{2n+1}, \tilde{\alpha})$  for the unobserved label  $Y_{2n+1}$ .

Build a hierarchical tree using Algorithm 1.

Randomly split the data into 3 equal subsets,  $\mathcal{D}_{tr}$ ,  $\mathcal{D}_{cal}$  and  $\mathcal{D}_{tune}$ .

Calculate the conformal  $p$ -values  $\tilde{p}_{tune}^{i,k}(X_j)$  on the nodes of the tree for the labeled data in  $\mathcal{D}_{tune}$  using Algorithm 2.

Perform multiple hypothesis testing (MHT) on the conformal  $p$ -values,  $\tilde{p}_{tune}^{i,k}(X_j)$  using Algorithm 3.

Find the tuning parameter  $\lambda^*$  using  $\tilde{p}_{tune}^{i,k}(X_j)$  in Algorithm 4.

Calculate the conformal  $p$ -values  $\tilde{p}_{test}^{i,y}(X_{2n+1})$  on the test data in  $\mathcal{D}_{test}$  using Algorithm 2.

Perform multiple hypothesis testing (MHT) on the conformal  $p$ -values  $\tilde{p}_{test}^{i,y}(X_{2n+1})$  with  $\lambda^*$  using Algorithm 3.

Build the prediction set,  $\hat{C}_{test}(X_{2n+1}, \tilde{\alpha})$  for  $\mathcal{D}_{test} = X_{2n+1}$  using equation (14).

---

---

**Algorithm 4:** Hyper-parameter tuning  $\lambda^*$ 


---

**Input:**  $\alpha_f = 0, \alpha_c = \alpha, \alpha^* = (\alpha_f + \alpha_c)/2, \alpha \in (0, 1)$ .

**Output:**  $\lambda^*$ 
**while** ( $\alpha^* > 0$  and  $\alpha^* \leq \alpha$ ) **do**

 Compute prediction set for each observation  $(X_j, Y_j) \in \mathcal{D}_{tune}, j = 1, \dots, n_3$  is given by

$$\hat{C}_{tune}(X_j, \alpha^*) = \{y : \text{Labelset } y \text{ for which leaf hypothesis } \in \mathcal{S}(\alpha^*)\} \quad (13)$$

 // This is done using output of Algorithm 3 where  $X_{2n+1}$  is replaced by  $X_j \in \mathcal{D}_{tune}$ .

 Calculate the coverage on  $\hat{C}_{tune}(X_j, \alpha^*)$  as

$$\mathcal{C}(\alpha^*) = \frac{1}{n_3} \sum_{j=1}^{n_3} \mathbb{I}\{Y_j \in \hat{C}_{tune}(X_j, \alpha^*)\}$$

**if**  $\mathcal{C}(\alpha^*) - (1 - (1 + 1/n)(\alpha - 1/n)) < 0$  **then**

 |  $\alpha_c \leftarrow \alpha^*; \alpha^* = (\alpha_f + \alpha_c)/2$ 
**end**
**if**  $\mathcal{C}(\alpha^*) - (1 - (1 + 1/n)(\alpha - 1/n)) > 1/n$  **then**

 |  $\alpha_f \leftarrow \alpha^*; \alpha^* = (\alpha_f + \alpha_c)/2$ 
**end**
**if**  $0 \leq \mathcal{C}(\alpha^*) - (1 - (1 + 1/n)(\alpha - 1/n)) \leq 1/n$  **then**

 | **break**
**end**
**end**
 $\lambda^* = \alpha^*/\alpha$ 


---

**Remark 3** In Algorithm 5, if we use procedure 1 of FWER control, then  $\mathcal{D}_{tune}$  and calculation of  $\lambda^*$  is omitted.

#### 4. Statistical Property

For any non-conformity score function, given the non-conformity scores of calibration data  $(X_i, Y_i)_{i=1}^n$  as  $\sigma_i, i = 1, \dots, n$  and for test data  $(X_{n+1}, Y_{n+1})$ , where  $Y_{n+1}$  is unknown as  $\sigma_{n+1}$ , we define smoothed conformal  $p$ -value for test observation as  $p(X_{n+1}, Y_{n+1})$  defined below:

$$p(X_{n+1}, Y_{n+1}) = \frac{\sum_{i=1}^n \{\mathbb{I}(\sigma_i < \sigma_{n+1}^{Y_{n+1}}) + U_{n+1} \cdot \mathbb{I}(\sigma_i = \sigma_{n+1}^{Y_{n+1}})\}}{n + 1}, \quad (15)$$

where  $U_{n+1} \sim Unif(0, 1)$  is random variable independent of  $\sigma_i$ 's introduced to break ties. For the sake of simplicity, we will use the notation  $\sigma_{n+1}$  to represent  $\sigma_{n+1}^{Y_{n+1}}$  throughout the remaining proofs in Section 4 and appendix A.

**Proposition 1 (Validity of conformal  $p$ -values)** *Given  $(n+1)$  observations,  $(X_i, Y_i)_{i=1}^{n+1}$  are exchangeable, then the smoothed conformal  $p$ -value defined in equation (15) is uniformly distributed on the interval  $(0, 1)$  i.e., for any given  $t \in [0, 1]$ , we have*

$$\mathbb{P}\{p(X_{n+1}, Y_{n+1}) \leq t\} = t$$

**Proposition 2** *Under the same exchangeability assumption as in Proposition 1, any hierarchical testing procedure defined in section 3.2.4 with critical values  $\alpha_i$  satisfying  $\sum_{i=1}^c \alpha_i \leq \alpha$  strongly controls FWER at level  $\alpha$ .*

**Remark 3** *By using the result of Proposition 2, procedure 1 described in section 3.2.4 with critical value  $\alpha_i = \alpha/c$  strongly controls FWER at level  $\alpha$ .*

To show the FWER control of procedure 2, we need to use the following lemma.

**Lemma 4** (*Romano et al., 2019*) *Suppose  $Z_1, \dots, Z_n$  are exchangeable random variables. For any  $\alpha \in (0, 1)$ ,*

$$\mathbb{P}\{Z_{n+1} \leq \hat{Q}_n((1 + \frac{1}{n})\alpha)\} \geq \alpha$$

*Moreover, if the random variable  $Z_1, \dots, Z_n$  are almost surely distinct, then also,*

$$\mathbb{P}\{Z_{n+1} \leq \hat{Q}_n((1 + \frac{1}{n})\alpha)\} \leq \alpha + \frac{1}{n}$$

**Proposition 5** *Under the same exchangeability assumption as in Proposition 1, procedure 2 defined in Section 3.2.4, ensures strong control of FWER at level  $\alpha$  i.e.,*

$$FWER \leq \alpha$$

*Moreover, the FWER is bounded from below by  $\alpha - 1/n$ , i.e.,*

$$FWER \geq \alpha - \frac{1}{n}$$

**Remark 6** *Proposition 5 shows procedure 2 is almost optimal in the sense that the corresponding FWER is almost equal to the pre-specified level  $\alpha$  if  $n$  is large enough.*

**Theorem 7 (Validity of Prediction Set)** *Under the same exchangeability assumption as in Proposition 1, the prediction sets obtained based on procedure 1 and 2 and equation (14) are both guaranteed to be marginally valid, i.e.,*

$$\mathbb{P}\{Y_{n+1} \in \hat{C}_{test}(X_{n+1}, \alpha)\} \geq 1 - \alpha.$$

*Moreover, the prediction set based on procedure 2 also satisfies*

$$\mathbb{P}\{Y_{n+1} \in \hat{C}_{test}(X_{n+1}, \alpha)\} \leq 1 - \alpha + \frac{1}{n}.$$

**Proof** *By using the same argument as in Proposition 5, the event of  $Y_{n+1} \in \hat{C}_{test}(X_{n+1}, \alpha^*)$  is equivalent to the only true null hypothesis is not rejected for each layer  $i$ , which in turn implies*

$$\mathbb{P}\{Y_{n+1} \in \hat{C}_{test}(X_{n+1}, \alpha)\} = 1 - FWER \tag{16}$$

By using proposition 2 and 5, we have marginal validity of prediction sets from both the procedures,

$$\mathbb{P}\{Y_{n+1} \in \hat{C}_{test}(X_{n+1}, \alpha)\} \geq 1 - \alpha. \quad (17)$$

From equation (26) of proposition 5, we get the upper bound of the prediction set for Procedure 2 as

$$\mathbb{P}\{Y_{n+1} \in \hat{C}_{test}(X_{n+1}, \alpha)\} \leq 1 - \alpha + \frac{1}{n}. \quad (18)$$

This completes the proof. ■

## 5. Missing Information

In practical applications, it is common to encounter situations where not all possible labelsets are present in the data. This missing information can pose a challenge when implementing our proposed Tree-Based Multi-label Conformal Prediction (*TB*) procedure described in section 3. To address this issue, we have developed two modified versions of our procedure that can handle missing labelsets in different ways.

1. The first approach referred to as *Tree-Based MLCP with present labelsets (TB1)*, involves building the hierarchical tree using only the labelsets that are present in the data. Next, we formulate the multiple hypothesis testing (MHT) problem based on the resulting hierarchical tree. Finally, we apply Algorithm 5 to build a prediction set for an unobserved test instance. Papadopoulos (2014) proposed Power-Set MLCP1 (PS-MLCP1) and a modified version for handling missing labelsets. We use the basic version proposed by Papadopoulos (2014) in our study for comparisons.
2. The second approach referred to as *Tree-Based MLCP with parent p-value labelsets (TB2)*, addresses the issue of missing labelsets by using the hierarchical tree of all possible labelsets and assigning parent *p*-value to the nodes with missing information. Finally, we apply Algorithm 5 to build a prediction set for an unobserved test instance. This approach provides a more structured way of making predictions on missing labelsets and takes into account the relationships between different labelsets in the hierarchical tree. Due to the logical relationship between parent and children from the hierarchical tree, we can assign the parent *p*-value to the nodes with missing labelsets (i.e., missing information). Hence, it can handle the missing information more efficiently than PS-MLCP2 (refer section 6 for more details), where we use PS-MLCP1 to the present labelsets and add all missing labelsets in the predictions set to account for missing information.

Overall, these two modified versions of our *TB* procedure provide more robust ways of handling missing labelsets in real-world applications. The choice of approach will depend on the specific context and nature of the missing labelsets, and both approaches have their respective strengths and limitations. *TB2* accounts for the missing information by assigning parent *p*-value to the nodes with missing information. However, this method is computationally intensive. On the other hand, *TB1* will be computationally efficient, but it does not account for missing information in the data. A trade-off between the length of

prediction sets and coverage rate arises when choosing between *TB1* and *TB2* methods. *TB1*, which employs present labelsets, yields shorter prediction sets with low coverage compared to *TB2*, which provides larger prediction sets with higher coverage.

## 6. Experiments

In this section, we evaluate the performance of our proposed method, *TB*, described in section 3, and compare it with two existing methods, BR-MLCP and PS-MLCP, with some modifications for fair comparisons. We use one simulated dataset and two real datasets to compare the methods based on their length of prediction set and marginal coverage. We evaluate the results for a range of significance levels, including  $\alpha = (0.02, 0.05, 0.08, 0.10, 0.12, 0.15, 0.20, 0.25, 0.30, 0.35)$ . For each task, we implement the following 5 methods for comparisons:

1. *Binary Relevance MLCP (BR-MLCP)*: With significance level  $= \alpha/c$  for  $c$  labels.
2. *Power Set MLCP with present labelsets (PS-MLCP1)*: In this scenario, Power set MLCP is applied to the data with present labelsets and missing labelsets are not included in the prediction set.
3. *Power Set MLCP with all labelsets (PS-MLCP2)*: In this scenario, Power set MLCP is applied to the data with present labelsets and missing labelsets are added in the prediction set.
4. *Tree Based MLCP with present labelsets (TB1)*: The following two versions are implemented with present labelsets:
  - (i) *TB1 fixed-alpha* method: Using procedure 1 of FWER control i.e.,  $\alpha_i = \alpha/c$  with Algorithm 5 (omitting tuning set and  $\lambda^*$ ).
  - (ii) *TB1 adaptive-alpha*: Using procedure 2 of FWER control i.e., using optimal  $\lambda^*$  with Algorithm 5.
5. *Tree Based MLCP with parent p-value labelsets (TB2)*: The following two versions are implemented with parent  $p$ -value labelsets:
  - (i) *TB2 fixed-alpha*: Using procedure 1 of FWER control i.e.,  $\alpha_i = \alpha/c$  with Algorithm 5 (omitting tuning set and  $\lambda^*$ ).
  - (ii) *TB2 adaptive-alpha*: Using procedure 2 of FWER control i.e., using optimal  $\lambda^*$  with Algorithm 5.

For all experiments, we use the Naive-Bayes algorithm as the machine learning algorithm, and the non-conformity score is computed as  $1 - \text{predicted probability}$ , as discussed in section 2.1. We perform 50 replications of the simulations and real data and present the average results.

### 6.1. Simulation

We generate  $n = 10,000$  data points  $(X_i, Y_i), i = 1, 2, \dots, n$ , and  $Y_i = (Y_i^1, \dots, Y_i^5)$  denotes the response vector for  $i^{th}$  observation with  $c = 5$  labels and each feature  $X = (X_1, X_2)$  is derived from Gaussian mixture model of two distributions as follows:

- $X_1 \sim 0.5 * \mathcal{N}(1, 0.3) + 0.5 * \mathcal{N}(0, 1)$
- $X_2 \sim 0.3 * \mathcal{N}(-1/2, 0.2) + 0.7 * \mathcal{N}(-\sqrt{3/2}, 0.4)$

We generate multiple labels (i.e., response variables) sequentially by  $Y^1 \sim \text{Bernoulli}(\pi_1)$ , where  $\pi_1 = \mathbb{P}(Y^1|X) = \frac{1}{1 + e^{-z_1}}$  and  $z_1 = X^T \beta + \epsilon_1$ , for a coefficient vector  $\beta = (\beta_0, \beta_1, \beta_2) = (2, 2.5, 2)$  and random error term,  $\epsilon_1 = -0.5X_1^3$ . For consecutive labels, i.e., for the  $j$ -th label,  $j > 1$  and  $j = 2, \dots, 5$ , we set  $z_j = X^T \beta + \sum_{k=1}^{j-1} w_{jk} Y^k + \epsilon_j$ ,  $w_{jk} = 0$  for  $k \geq j$ , and  $w_{jk} = 1.5$  for  $k < j$ , indicating that the value of  $(Y^1, \dots, Y^{j-1})$  has a direct effect on  $Y^j$ . This is done to introduce dependence among the labels. The total number of possible labelsets is  $2^c$ , out of which 16 labelsets are present in the data as illustrated from the leaf nodes in Figure 3. However, *TB2* (all labelsets) uses 32 possible labelsets to build the tree.

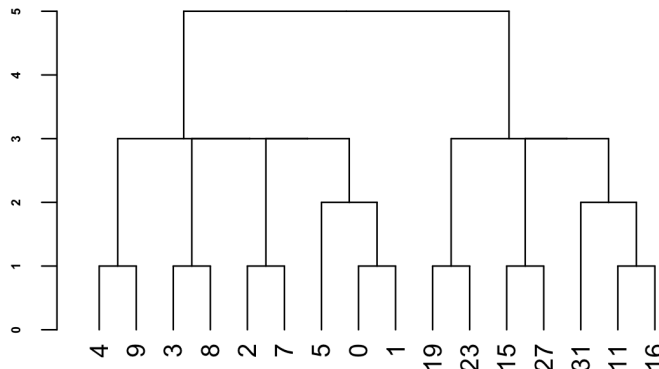


Figure 3: Hierarchical tree for TB-MLCP (present labelsets) where  $c = 5$ .

In the adaptive-alpha approach, we split the data into four parts: proper training, calibration, tuning and test, with a split ratio of 30:30:20:20. In the fixed-alpha approach, we split the data into three parts: proper training, calibration and test, with a split ratio of 20:60:20.

In Figure 4, we present a comparison of our proposed procedures (*TB1* and *TB2* with both fixed-alpha and adaptive-alpha) with existing methods including BR-MLCP, PS-MLCP1, and PS-MLCP2. The left panel of Figure 4 shows that all evaluated methods, including our proposed procedures, are able to achieve marginal coverage, indicating their ability to capture the true labels with high probability. *TB1* and *TB2* with fixed-alpha approach provide over-coverage, but with adaptive-alpha approaches, *TB1* and *TB2* provide nicely controlled coverages. The right panel of Figure 4 indicates that BR-MLCP and PS-MLCP2 tend to generate larger prediction sets, while our proposed tree-based approach (*TB2* with adaptive-alpha) provides shorter prediction sets than *TB2* with fixed-alpha approach and PS-MLCP2 method, providing a more efficient solution to the problem of missing

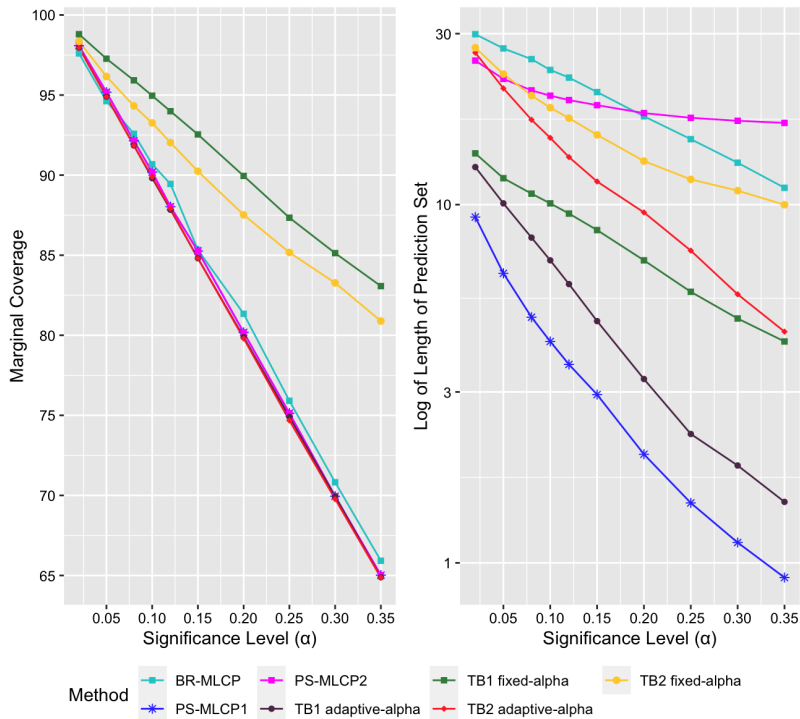


Figure 4: Comparison results of our-proposed procedure with BR-MLCP and PS-MLCP in terms of logarithm of the length of the prediction set and marginal coverage.

information than the PS-MLCP2 method. We also note that there is only a slight difference between the prediction set lengths of PS-MLCP1 and TB1 with adaptive-alpha. Also, the adaptive-alpha approach produces shorter prediction sets than the fixed-alpha approach in both *TB1* and *TB2* methods.

## 6.2. Real Data Example

The real datasets used in this study were obtained from the MULAN library.

### 6.2.1. SCENE CLASSIFICATION

The dataset pertains to the semantic classification of images into multiple labels, including beach, sunset, foliage, field, mountain, and urban, with a total of 6 labels. The dataset comprises 1,211 training and 1,196 test samples, each described by 294 features. Among the 64 possible labelsets in the data with  $c = 6$ , only 14 were present, and we further filtered and preprocessed the data to focus on 8 of these labelsets (that occurred more than 20 times). To train and evaluate our proposed methods with existing methods, we split training samples into proper training and calibration set as 40:60 split. If we require tuning set (i.e., for *TB1* and *TB2* with adaptive-alpha approach), then we split test samples into tuning and test with 50:50 split else, there is no split for test samples.



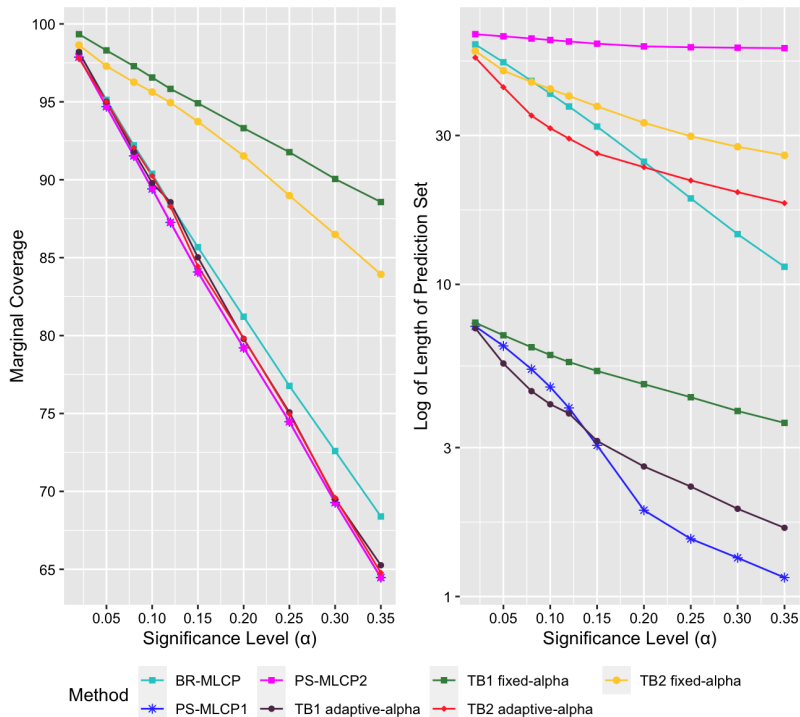


Figure 5: Comparison results of our-proposed procedure with BR-MLCP and PS-MLCP in terms of logarithm of the length of the prediction set and marginal coverage on scene dataset.

Figure 5 provides similar results as simulation studies. The left panel of the figure shows that all methods, including our proposed methods, provide marginal coverage guarantees with over coverage provided by the fixed-alpha approach. However, with adaptive-alpha approach, nice coverages are achieved for both *TB1* and *TB2* methods. The right panel of Figure 5 shows that *TB2* fixed-alpha and adaptive-alpha approach produces shorter sets compared to the PS-MLCP2 approach. Moreover, the set size produced by the *TB2* adaptive-alpha approach is smaller than *TB2* fixed-alpha approach. Meanwhile, PS-MLCP1 and *TB1* adaptive-alpha produce similar set sizes. *TB1* fixed-alpha set sizes are slightly larger than *TB1* adaptive-alpha. BR-MLCP produces larger set sizes than other methods.

### 6.2.2. YEAST

The dataset comprises 1,500 training and 917 test samples. Each gene is initially characterized by microarray expression data and phylogenetic profile, from which 103 features were extracted, and is linked to a subset of 14 functional classes (i.e.,  $c = 14$  labels). Among the 16,384 possible labelsets in the data with  $c = 14$ , only 16 were present. To train and evaluate our proposed methods, we have used the same setting as in section 6.2.1.

The results presented in Figure 6 demonstrate that all of the methods under consideration offer marginal coverage guarantees. *TB1* and *TB2* with adaptive-alpha approach

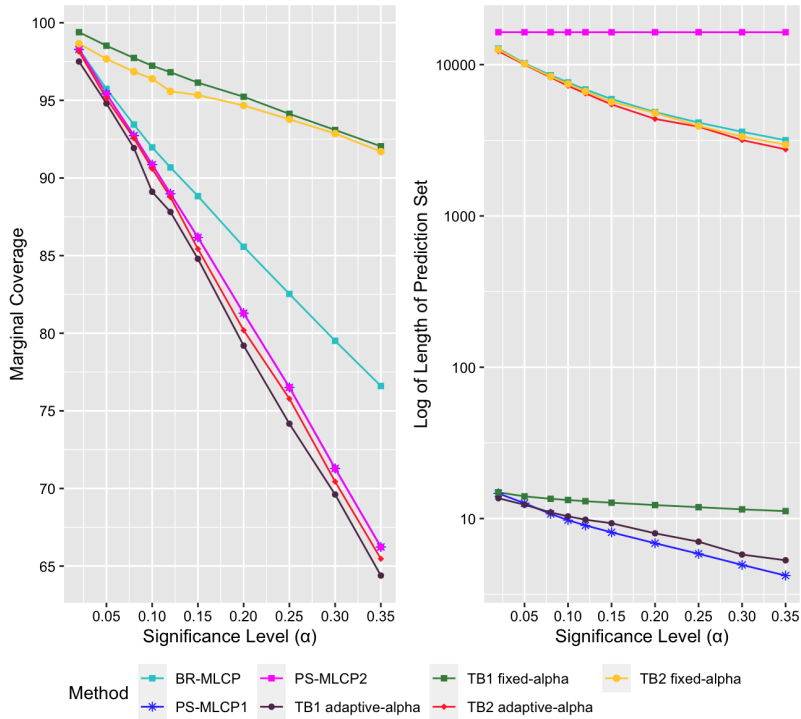


Figure 6: Comparison results of our-proposed procedure with BR-MLCP and PS-MLCP in terms of logarithm of the length of the prediction set and marginal coverage on yeast dataset.

provides nice coverage as compared to fixed-alpha approach with both methods. In terms of prediction set size, our proposed method *TB2* adaptive-alpha outperforms PS-MLCP2. Both PS-MLCP1 and *TB1* adaptive-alpha show similar results in terms of the length of prediction sets. On the other hand, the length of the prediction set for BR-MLCP is shorter than that of PS-MLCP2. *TB1* and *TB2* with adaptive-alpha approach produce shorter set sizes as compared to *TB1* and *TB2* with fixed-alpha approach.

### 6.2.3. TUNING PARAMETER

In this section, we verify that the hyperparameter  $\lambda^*$  is much larger than  $1/c$ , which implies that the adaptive-alpha approach is more powerful than the fixed-alpha approach in both *TB1* and *TB2* methods. This is because the adaptive-alpha approach uses higher critical value than the fixed-alpha approach. Figure 7 illustrates  $c\lambda^*$  as a function of  $\alpha$  and demonstrates that  $c\lambda^*$  is significantly greater than 1 for all three scenarios with  $c = 5, 6, 14$ . This finding clearly explains why the adaptive-alpha approach performs better than the fixed-alpha approach in our simulation study and real data analysis.

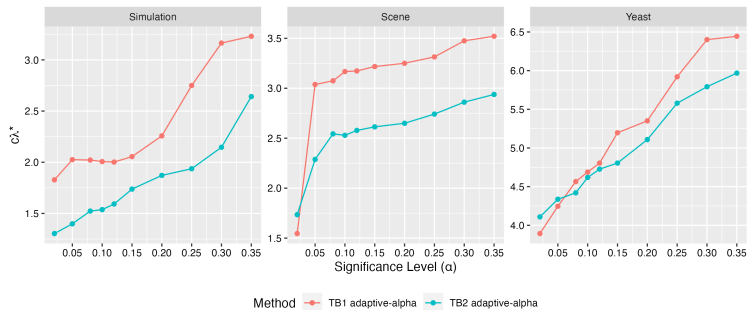


Figure 7: Tuning parameter  $\lambda^*$  behavior with *TB1* and *TB2* adaptive-alpha approaches in simulation, scene data and yeast data respectively.

## 7. Discussion

In this paper, we proposed a tree-based method for multi-label classification problems using conformal prediction and multiple-testing. We presented four variants of the method, *TB1* (present labelsets) and *TB2* (all labelsets), both with fixed-alpha and adaptive-alpha approaches. The adaptive-alpha approach necessitates dividing the data into four parts, making it more suitable for relatively large datasets where it may exhibit better performance. Our method allows the use of any base classifier and provides prediction sets with a pre-specified coverage rate, such as 90% while maintaining a small average length of the prediction set. We can tune the parameter  $\lambda$  for adaptive-alpha approaches to ensure the nice marginal coverage guarantee. Our simulation study and real data analysis demonstrate that *TB2* adaptive-alpha method outperforms other methods in terms of coverage and prediction set length and also accounts for missing label information. We anticipate that *TB2* adaptive-alpha method will be a useful tool for researchers and practitioners working on multi-label classification problems. However, some critical questions still need to be addressed in future work. Firstly, we will explore more desired conditional coverage guarantees. Secondly, when a large number of test samples are considered, we may need to control some appropriate overall error rate, such as false discovery rate or false coverage rate. Additionally, it is necessary to extend the method to handle a larger number of labels and explore its performance in other types of machine-learning problems.

## References

- Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511v6*, 2022.
- Anastasios N Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. *arXiv preprint arXiv:2208.02814*, 2022.

- Anastasios Nikolas Angelopoulos, Stephen Bates, Michael Jordan, and Jitendra Malik. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*, 2021.
- Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications*. Morgan Kaufmann Publishers Inc., 2014.
- Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486 – 507, 2021.
- Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021.
- Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p-values. *arXiv preprint arXiv:2104.08279v3*, 2022.
- Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.
- Maxime Cauchois, Suyash Gupta, and John C Duchi. Knowing what you know: valid and validated confidence sets in multiclass and multilabel prediction. *The Journal of Machine Learning Research*, 22(1):3681–3722, 2021.
- Chirag Gupta, Arun K. Kuchibhotla, and Aaditya Ramdas. Nested conformal prediction and quantile out-of-bag ensemble methods. *Pattern Recognition*, 127:108496, 2022.
- Yosef Hochberg and Ajit C Tamhane. *Multiple comparison procedures*. John Wiley & Sons, Inc., 1987.
- Rafael Izbicki, Gilson T Shimizu, and Rafael B Stern. Flexible distribution-free conditional predictive bands using density estimators. *arXiv preprint arXiv:1910.05575*, 2019.
- Ulf Johansson, Henrik Boström, Tuve Ljöfström, and Henrik Linusson. Regression conformal prediction with random forests. *Machine Learning*, 97:155–176, 2014.
- Antonis Lambrou and Harris Papadopoulos. Binary relevance multi-label conformal predictor. In *Conformal and Probabilistic Prediction with Applications: In Proceedings of the 5th International Symposium on Conformal and Probabilistic Prediction with Applications (COPA 2016), LNCS 9653*, pages 90–104. Springer, 2016.
- Jing Lei. Classification with confidence. *Biometrika*, 101(4):755–769, 2014.
- Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, 76(1): 71–96, 2014.
- Maltoudoglou Lysimachos, Andreas Paisios, Ladislav Lenc, Jiří Martínek, Pavel Král, and Harris Papadopoulos. Well-calibrated confidence measures for multi-label text classification with a large number of labels. *Pattern Recognition*, 122:108271, 2022.

- Andreas Paisios, Ladislav Lenc, Jiří Martínek, Pavel Král, and Harris Papadopoulos. A deep neural network conformal predictor for multi-label text classification. In *Proceedings of the 8th Symposium on Conformal and Probabilistic Prediction and Applications (COPA 2019)*, PMLR 105, pages 228–245, 2019.
- Harris Papadopoulos. A cross-conformal predictor for multi-label classification. In *Artificial Intelligence Applications and Innovations: AIAI 2014 Workshops: CoPA, MHDW, IIVC, and MT4BD, 2014. Proceedings 10*, pages 241–250. Springer, 2014.
- Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. In *Machine Learning and Knowledge Discovery in Databases*, pages 254–269. Springer Berlin Heidelberg, 2009.
- Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in Neural Information Processing Systems*, 32:3543–3553, 2019.
- Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591, 2020.
- Mauricio Sadinle, Jing Lei, and Larry Wasserman. Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 114(525): 223–234, 2019.
- Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3):371–421, 2008.
- Darius Tamasauskas, Virgilijus Sakalauskas, and Dalia Kriksciuniene. Evaluation framework of hierarchical clustering methods for binary data. In *2012 12th International Conference on Hybrid Intelligent Systems (HIS)*, pages 421–426, 2012.
- Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in Neural Information Processing Systems*, 32:2530–2540, 2019.
- Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.
- Vladimir Vovk. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74:9–28, 2015.
- Vladimir Vovk, David Lindsay, Ilia Nouretdinov, and Alex Gammerman. Mondrian confidence machine. *Technical Report*, 2003.
- Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.
- Volodya Vovk, Alexander Gammerman, and Craig Saunders. Machine-learning applications of algorithmic randomness. In *International Conference on Machine Learning*, pages 444–453, 1999.

Huazhen Wang, Xin Liu, Bing Lv, Fan Yang, and Yanzhu Hong. Reliable multi-label learning via conformal predictor and random forest for syndrome differentiation of chronic fatigue in traditional chinese medicine. *PloS One*, 9(6):e99565, 2014.

Huazhen Wang, Xin Liu, Iliia Nouretdinov, and Zhiyuan Luo. A comparison of three implementations of multi-label conformal prediction. In *Statistical Learning and Data Sciences: Third International Symposium, SLDS 2015, Proceedings 3*, pages 241–250. Springer, 2015.

Chen Xu and Yao Xie. Conformal prediction interval for dynamic time-series. In *International Conference on Machine Learning*, pages 11559–11569. PMLR, 2021.

## Appendix A. Proofs

### A.1. Proof of Proposition 1: Validity of conformal $p$ -values

**Proof** Suppose, for any given values of non-conformity scores,  $v_1, \dots, v_{n+1}$ , they can be rearranged as:

$$\tilde{v}_1 < \dots < \tilde{v}_k,$$

where each  $\tilde{v}_i$  is repeated  $n_i$  times such that  $\sum_{i=1}^k n_i = n + 1$ . Let  $E_v$  denote the event that the non-conformity scores  $\sigma_i$ ,  $i = 1, \dots, n + 1$  take on the specific values  $v_1, \dots, v_{n+1}$ . Mathematically,

$$E_v = \{(\sigma_1, \sigma_2, \dots, \sigma_{n+1}) : \sigma_i = v_i \text{ for some permutation of } v_1, \dots, v_{n+1}\}$$

Since  $\{X_i, Y_i\}_{i=1}^{n+1}$  are exchangeable,  $\sigma_i$ 's are also exchangeable, we have

$$\mathbb{P}\{\sigma_{n+1} = \tilde{v}_j | E_v\} = \frac{n_j}{n+1}, \text{ for } j = 1, \dots, k \quad (19)$$

Under the event of  $E_v$  and  $\sigma_{n+1} = \tilde{v}_j$ , we have from equation (15)

$$p(X_{n+1}, Y_{n+1}) = \frac{\sum_{i=1}^{j-1} n_i + U_{n+1} \cdot n_j}{n+1} \quad (20)$$

Thus, for any  $t \in [0, 1]$ , we have from equation (20)

$$\begin{aligned} \mathbb{P}\{p(X_{n+1}, Y_{n+1}) \leq t | E_v, \sigma_{n+1} = \tilde{v}_j\} &= \mathbb{P}\left\{\frac{\sum_{i=1}^{j-1} n_i + U_{n+1} \cdot n_j}{n+1} \leq t | E_v, \sigma_{n+1} = \tilde{v}_j\right\} \\ &= \begin{cases} \frac{(n+1)t - \sum_{i=1}^{j-1} n_i}{n_j} & \text{if } \frac{\sum_{i=1}^{j-1} n_i}{n+1} < t \leq \frac{\sum_{i=1}^j n_i}{n+1}, \\ 0 & \text{if } 0 < t \leq \frac{\sum_{i=1}^{j-1} n_i}{n+1}, \\ 1 & \text{if } \frac{\sum_{i=1}^j n_i}{n+1} < t \leq 1 \end{cases} \end{aligned}$$

for  $j = 1, \dots, k$ . Thus, for  $\frac{\sum_{i=1}^{j-1} n_i}{n+1} < t \leq \frac{\sum_{i=1}^j n_i}{n+1}$  and  $j = 1, 2, \dots, k$  we have

$$\begin{aligned} \mathbb{P}\{p(X_{n+1}, Y_{n+1}) \leq t | E_v\} &= \sum_{j=1}^k \mathbb{P}\{p(X_{n+1}, Y_{n+1}) \leq t | E_v, \sigma_{n+1} = \sigma_j\} \cdot \mathbb{P}\{\sigma_{n+1} = \sigma_j | E_v\} \\ &= \sum_{j=1}^k \frac{n_j}{n+1} \mathbb{P}\{p(X_{n+1}, Y_{n+1}) \leq t | E_v, \sigma_{n+1} = \sigma_j\}, \text{ using equation (19)} \\ &= \frac{n_j}{n+1} \left[ \frac{(n+1)t - \sum_{i=1}^{j-1} n_i}{n_j} \right] + \frac{\sum_{i=1}^{j-1} n_i}{n+1} \\ &= t - \frac{\sum_{i=1}^{j-1} n_i}{n+1} + \frac{\sum_{i=1}^{j-1} n_i}{n+1} \\ &= t \end{aligned}$$

That is,

$$p(X_{n+1}, Y_{n+1}) \mid E_v \sim U(0, 1) \quad (21)$$

Taking the expectation on both sides of equation (21), we have

$$p(X_{n+1}, Y_{n+1}) \sim U(0, 1)$$

This completes the proof. ■

## A.2. Proof of Proposition 2

**Proof** For the tree-based tested hypotheses formulated in section 3.2.2, there is only one true null hypothesis in each layer. Let  $H_i$  denote the true null hypotheses for  $i^{\text{th}}$  layer. Then, the FWER of the *hierarchical testing procedure* with critical value  $\alpha_i$  is given below.

$$FWER = \mathbb{P}\{\text{at least one } H_i \text{ is rejected for } i = 1, \dots, c\}$$

Based on the construction of tested hypotheses, true null hypotheses  $H_1, \dots, H_c$  have the parent-children relationship (i.e.,  $H_i$  is a parent of  $H_{i+1}$  for  $i = 1, \dots, c-1$ ), we get

$$\begin{aligned} FWER &= \sum_{i=1}^c \mathbb{P}\{H_i \text{ is rejected but } H_j \text{ are not rejected for } j = 1, \dots, i-1.\} \\ &\leq \sum_{i=1}^c \mathbb{P}\{p(X_{n+1}, Y_{n+1}) \leq \alpha_i\} \text{ (based on the definition of hierarchical testing)} \\ &= \sum_{i=1}^c \alpha_i, \text{ (by Proposition 1)} \\ &\leq \alpha, \end{aligned}$$

the desired result. ■

## A.3. Proof of Proposition 5

**Proof** Let  $\mathcal{D}_{tune}$  be the tuning dataset with  $|\mathcal{D}_{tune}| = n$ . By Algorithm 4, we have

$$\frac{1}{n} \sum_{j=1}^n \mathbb{I}\{Y_j \in \hat{C}_{tune}(X_j, \alpha^*)\} = 1 - \left(1 + \frac{1}{n}\right)\left(\alpha - \frac{1}{n}\right) \quad (22)$$

Note that for any observation  $(X_j, Y_j) \in \mathcal{D}_{tune}$ , the event of  $Y_j \in \hat{C}_{tune}(X_j, \alpha^*)$  is equivalent to the corresponding true null hypothesis is not rejected for each layer  $i$ , which in turn is equivalent to  $p^i(X_j, Y_j) \geq \alpha^*$ , where  $p^i(X_j, Y_j)$  denotes the smoothed conformal  $p$ -value for  $i^{\text{th}}$  layer and  $j^{\text{th}}$  observation as defined in equation (15). Define  $p(X_j, Y_j) =$



$\min_{i \in \{1, \dots, c\}} \{p^i(X_j, Y_j)\}$ . Thus,  $\mathbb{I}\{Y_j \in \hat{C}_{tune}(X_j, \alpha^*)\} = \mathbb{I}\{p(X_j, Y_j) \geq \alpha^*\}$ . Therefore, equation (22) is equivalent to

$$\frac{1}{n} \sum_{j=1}^n \mathbb{I}\{p(X_j, Y_j) \geq \alpha^*\} = 1 - (1 + \frac{1}{n})(\alpha - \frac{1}{n}) \quad (23)$$

Thus,  $\alpha^* = \hat{Q}_n((1 + \frac{1}{n})(\alpha - \frac{1}{n}))$ , the upper  $(1 - (1 + \frac{1}{n})(\alpha - \frac{1}{n}))$  quantile of  $\{p(X_j, Y_j)\}_{j=1}^n$ . Let  $Z_j = p(X_j, Y_j)$ ,  $j = 1, \dots, n$  and  $Z_{n+1} = \min_{i \in \{1, \dots, c\}} \{p^i(X_{n+1}, Y_{n+1})\}$ . Note that  $(X_i, Y_i)$ 's are exchangeable, so  $Z_1, \dots, Z_{n+1}$  are also exchangeable. By Lemma 4, we have

$$\mathbb{P}\{Z_{n+1} \leq \hat{Q}_n((1 + \frac{1}{n})(\alpha - \frac{1}{n}))\} \geq \alpha - \frac{1}{n}$$

By using the fact of  $\alpha^* = \hat{Q}_n((1 + \frac{1}{n})(\alpha - \frac{1}{n}))$ , we have

$$\mathbb{P}\{Z_{n+1} \leq \alpha^*\} \geq \alpha - \frac{1}{n} \quad (24)$$

By equation (15),  $Z_1, \dots, Z_{n+1}$  are almost surely distinct, then Lemma 4 also gives

$$\mathbb{P}\{Z_{n+1} \leq \alpha^*\} \leq \alpha \quad (25)$$

Note that FWER of procedure 2 is given by  $\mathbb{P}\{Z_{n+1} \leq \alpha^*\}$ . By equation (24) and (25), we have

$$\alpha - \frac{1}{n} \leq FWER \leq \alpha \quad (26)$$

This completes the proof. ■