Incorporating Structural Penalties in Multi-label Conformal Prediction

Kostas Katsios K.KATSIOS@ALBOURNE.COM
Harris Papadopoulos H.PAPADOPOULOS@FREDERICK.AC.CY
Computational Intelligence Research Lab.,
Frederick University, Nicosia, Cyprus

Machine Learning Research Group,
Albourne Partners (Cyprus) Ltd, Nicosia, Cyprus

Editor: Khuong An Nguyen, Zhiyuan Luo, Harris Papadopoulos, Tuwe Löfström, Lars Carlsson and Henrik Boström

Abstract

We propose two structural penalties for the Label-Powerset Split Conformal Prediction framework in multi-label learning. Building on our previously proposed Mahalanobis nonconformity measure, we add penalties that favour label-sets similar to previously observed ones in terms of Hamming distance and cardinality. The resulting nonconformity measure steers prediction regions toward label-sets that are both plausible and compact. Experiments on three public datasets (Emotions, PlantPseAAC, Yeast) show an average of 30% reduction in prediction region size for Emotions, 82% for PlantPseAAC and 39% for Yeast, compared to the Mahalanobis baseline.

Keywords: Mahalanobis, Multi-label, Classification, Conformal, Prediction, Power-set, Hamming, Cardinality, Reduction

1. Introduction

This work focuses on the reliable quantification of uncertainty in multi-label learning through the Conformal Prediction (CP) framework. Specifically, we propose a Split (or Inductive) Conformal Prediction (SCP) approach using the Label Power-set (LP) technique, as introduced by Papadopoulos (2014), which can be combined with any classifier that produces a score for each class. The proposed method calculates the nonconformity and p-value of each possible label-set and provides prediction regions of label-sets with guaranteed $1 - \varepsilon$ coverage for any significance level ε.

This study extends our previous work (Katsios and Papadopoulos, 2024) by proposing an approach to reduce prediction region sizes based on a Mahalanobis nonconformity measure. Inspired by the penalty term proposed by (Papadopoulos, 2014), our method involves processing of the proper-training label-sets and penalizing the Mahalanobis nonconformity scores. The Mahalanobis distance is defined through a covariance matrix, which in our case is derived from the proper-training data, taking into account correlations between error vectors. We extend the Mahalanobis nonconformity measure by adding two penalty terms, one based on the normalized Hamming distance and the other on the cardinality of each proper-training label-set.

As a preprocessing step, we calculate the normalized Hamming distance between each possible label-set and all proper-training label-sets, assigning to each possible label-set its

minimum normalized Hamming distance value. Similarly, we determine the frequency of each possible cardinality in the proper-training data and compute the difference of each frequency from unity. These values serve as penalty terms added to the original nonconformity score. The resulting nonconformity measure significantly reduces the size of prediction regions compared to the original Mahalanobis-based approach, as presented in our experimental results.

The rest of this paper is structured as follows. Section 2 provides an overview of the Split Conformal Prediction framework, the multi-label classification setting and previous work on CP for multi-label classification. Section 3 introduces our proposed method. Section 4 presents experimental results and compares our method with a previously proposed approach based on the Mahalanobis nonconformity measure. Finally, Section 5 concludes the paper and outlines directions for future work.

2. Technical background

2.1. Split Conformal Prediction

Split Conformal Prediction (SCP), also known as Inductive Conformal Prediction (ICP), is a computationally efficient framework for constructing prediction sets with guaranteed coverage (Papadopoulos et al., 2002a,b). Unlike the traditional full conformal prediction approach (Vovk et al., 2005), which requires retraining the model for each test instance, SCP avoids this by splitting the available data into two disjoint subsets.

For classification tasks, given a dataset $Z = \left\{(x_i, \psi_i) \ : \ x_i \in \mathbb{R}^s, \ \psi_i \in \Psi\right\}$ for $i = 1, ..., n$, where s is the number of attributes, $\mathbb{R}^s$ is the space of features and $\Psi = \{\mathcal{Y}_1, \ldots, \mathcal{Y}_d\}$ the set of possible classes. SCP divides the available data as follows:

- Proper-training Set: $Z_{tr} = \{(x_1, \psi_1), ..., (x_q, \psi_q)\}$, where $q \leq n$.
- Calibration Set: $Z_{cal} = \{(x_{q+1}, \psi_{q+1}), ..., (x_n, \psi_n)\}$.

The proper-training set is used to train the predictive model, called *underlying model*. The calibration set is utilized to calculate *nonconformity scores*, which measure how unusual or nonconforming a data point is with respect to the model's predictions. The nonconformity scores are produced by a *nonconformity measure*,

$$A : Z_{tr} \times \left\{(x_i, \psi_i)\right\} \rightarrow \mathbb{R} \tag{1}$$

that quantifies the dispute between the model's prediction for input x_i and the actual observed values. Once the model is trained on the proper-training set, it is used to compute nonconformity scores for the calibration set,

$$a_i^{\psi_i} = A\Big(\left\{(x_1, \psi_1), ..., (x_q, \psi_q)\right\}, (x_i, \psi_i)\Big), \ i = q+1, ..., n. \tag{2}$$

SCP assumes all possible classifications $\mathcal{Y}_c \in \{\mathcal{Y}_1, \ldots, \mathcal{Y}_d\}$ for the test instance x_{n+1} and calculates their nonconformity scores by applying function A to the trained model and each test pair $(\boldsymbol{x}_{n+1}, \mathcal{Y}_c)$,

$$a_{n+1}^{\mathcal{Y}_c} = A\Big(\left\{(x_1, \psi_1), ..., (x_q, \psi_q)\right\}, (x_{n+1}, \mathcal{Y}_c)\Big). \tag{3}$$

To quantify the plausibility of each candidate label, SCP computes a conformal p-value,

$$p(\mathcal{Y}_c) = \frac{\left|i = q+1, ..., n \ : \ a_i^{\psi_i} \geq a_{n+1}^{\mathcal{Y}_c}\right| + 1}{n - q + 1}, \tag{4}$$

which is a valid p-value of the null hypothesis that $\mathcal{Y}_c$ is the true label of x_{n+1}. The prediction region at significance level $\varepsilon \in [0, 1]$ is then defined as

$$\Gamma^{\varepsilon}_{x_{n+1}} = \left\{\mathcal{Y}_c \ : \ p(\mathcal{Y}_c) > \varepsilon\right\}. \tag{5}$$

This ensures that, with probability at least $(1 - \varepsilon)$, the prediction region $\Gamma^{\varepsilon}_{x_{n+1}}$ will cover the true label of x_{n+1},

$$\mathbb{P}\Big(\psi_{n+1} \in \Gamma^{\varepsilon}_{x_{n+1}}\Big) > 1 - \varepsilon. \tag{6}$$

In fact, SCP satisfies this coverage guarantee under the assumption of data exchangeability, regardless of the underlying data distribution or model.

2.2. Multi-label learning

Multi-label learning is a branch of supervised learning in which each instance can be associated with multiple labels. Therefore $\psi_i \subseteq \mathcal{L}$, where $\mathcal{L} = \{\lambda_1, \ldots, \lambda_d\}$ denotes the set of d individual labels. This setting arises naturally in a wide range of real-world applications, including text categorization (see e.g. Haralabopoulos et al. (2020)), image annotation (see e.g. Rückert et al. (2024)), bioinformatics (see e.g. Maxwell et al. (2017)), and music classification (see e.g. Zhong et al. (2023)), where an object may belong to multiple categories.

Approaches to multi-label learning can be broadly divided into two primary categories: Problem Transformation (PT) methods and Algorithm Adaptation (AA) methods (see (Tsoumakas and Katakis, 2007)). Problem Transformation techniques convert the multi-label problem into one or more single-label or binary classification problems, allowing the use of standard machine learning algorithms. This enables the application of conventional supervised learning algorithms without requiring substantial algorithmic changes. In contrast, Algorithm Adaptation methods involve modifying existing learning algorithms to handle multi-label data.

Among the two, Problem Transformation (PT) methods are particularly popular due to their simplicity, scalability, and flexibility. The most commonly used PT methods include:

- Instance Reproduction (IR): The Instance Reproduction method tackles multi-label learning by generating multiple binary instances from each multi-label instance, one for every label in the label-set. In this transformation, each copy of the instance is paired with a single positive label while keeping the input features unchanged. The application of the Instance Reproduction transformation reformulates the multi-label dataset Z into a dataset

$$Z_{IR} = \bigcup_{i=1}^{n} \bigcup_{j=1}^{d} \left\{\left(x_i, \lambda_j\right)\right\}, \quad \text{for all } \lambda_j \in \psi_i. \tag{7}$$

  This approach is particularly well-suited for handling highly unbalanced datasets and supports label-specific modeling, allowing classifiers to focus on distinguishing each individual label's characteristics.

- Binary Relevance (BR): This method decomposes the multi-label problem into independent binary classification tasks as many as the number of labels, one for each label $\lambda_j \in \mathcal{L}$. Each label is mapped to a binary value by setting $\ell_j = 1$ if label λ_j is associated with instance x_i and $\ell_j = 0$ otherwise. Based on this encoding, a separate binary classification dataset is constructed for each label

$$Z_{BR}^{j} = \big\{(\boldsymbol{x}_i, \ell_j)\big\}_{i\in\mathbb{N}}, \quad \text{for } j = 1, \ldots, d. \tag{8}$$

  Although BR is computationally efficient and easy to implement, it assumes label independence, which limits its ability to capture correlations among labels.

- Label Power-set (LP): LP treats each unique combination of labels as a distinct class in a multi-class classification problem. This approach inherently models label dependencies but suffers from poor generalization in the presence of rare or unseen label-sets, leading to scalability challenges when the number of possible label combinations grows exponentially. The full set of possible label combinations is given by the power-set $\Psi = \mathcal{P}(\mathcal{L})$.

2.3. Multi-label Split Conformal Prediction

The three Problem Transformation (PT) strategies - Instance Reproduction (IR), Binary Relevance (BR), and Label Powerset (LP) - integrate differently with SCP, particularly in how they split the dataset and form prediction regions.

A central property of Conformal Prediction (CP) is its ability to control the prediction error at a user-defined significance level $\varepsilon \in [0, 1]$. However, the type of validity guarantee depends on the transformation applied. Below, we describe the integration of each transformation strategy within the SCP framework:

- **Instance Reproduction within SCP (IR-SCP):** The IR-SCP method was introduced by Wang et al. (2014). The original multi-label dataset Z is transformed into an expanded binary dataset Z_{IR}. The resulting dataset is split into a proper-training set and a calibration set:

$$Z_{IR} = Z_{tr} \cup Z_{cal}. \tag{9}$$

  A classifier is trained on Z_{tr} to predict the relevance of each label independently. For each label $\lambda_j \in \mathcal{L}$, nonconformity scores are computed on the calibration set Z_{cal}, and p-values are calculated for the testing pair (x_{n+1}, λ_j).

  The prediction region includes all labels with p-values exceeding the significance level:

$$\Gamma_{x_{n+1}}^{\varepsilon} = \big\{\lambda_j \in \mathcal{L} \ : p_j(\lambda_j) > \varepsilon\big\} \cup \Big\{\arg max_{j=1,\cdots,d}\big(p_j(\lambda_j)\big)\Big\}, \quad \text{for } j = i, \ldots, d. \tag{10}$$

  The error per label is defined as

$$err_{n+1}^{\lambda_j} = \begin{cases} 1, & \lambda_j \notin \Gamma_{x_{n+1}}^{\varepsilon},\ \lambda_j \in \psi_{n+1} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

  where ψ_{n+1} is the true label-set of instance x_{n+1}. And the overall error,

$$err_{n+1} = \frac{1}{|\psi_{n+1}|} \sum_{\lambda_j \in t_{n+1}} err_{n+1}^{\lambda_j}. \tag{12}$$

The IR-SCP guarantees coverage of individual labels as opposed to label-sets and the prediction is valid if

$$\mathbb{P}\{err_{n+1} \leq \varepsilon\} \leq \varepsilon. \tag{13}$$

- **Binary Relevance within SCP (BR-SCP):** The BR-SCP approach, described in Wang et al. (2015), reformulates the dataset separately for each label $\lambda_j \in \mathcal{L}$, resulting in d binary datasets. For each label, the dataset is split into a proper-training and calibration set:

$$Z^j_{BR} = Z^j_{tr} \cup Z^j_{cal} \tag{14}$$

A separate binary conformal predictor is trained for each label using Z^j_{tr}, and calibration scores are computed on Z^j_{cal}. For a test instance x_{n+1}, p-values $p_j(\lambda_j)$ are computed independently.

The prediction region is defined as the Cartesian product of the individual predictors,

$$\Gamma^{\varepsilon}_{x_{n+1}} = \Gamma^{\varepsilon}_1 \times \cdots \times \Gamma^{\varepsilon}_d, \tag{15}$$

where each per-label conformal predictor is defined as

$$\Gamma^{\varepsilon}_j = \left\{\lambda_j \ : \ p_j(\lambda_j) > \frac{\varepsilon}{d}\right\} \cup \left\{\arg max_{j=1,\cdots,d}\big(p_j(\lambda_j)\big)\right\}, \quad \text{for } j = i, \ldots, d, \tag{16}$$

To satisfy overall validity, one can apply the Bonferroni correction, adjusting the per label significance to $\frac{\varepsilon}{d}$, ensuring the overall minimum probability is $1 - \varepsilon$,

$$\mathbb{P}\Big(\psi_{n+1} \in \Gamma^{\varepsilon}_{x_{n+1}}\Big) > 1 - \varepsilon, \tag{17}$$

where ψ_{n+1} is the true label-set of x_{n+1}.

Furthermore, Lambrou and Papadopoulos (2016) proposed an adjustment to the per label significance to,

$$\frac{\varepsilon(h+1)}{d}, \tag{18}$$

which ensures that the prediction region has at most ε probability to occur a Hamming loss greater than h. This refinement enables relaxing the required guarantee so as to obtain tighter prediction regions.

- **Label Power-Set within SCP (LP-SCP):** The LP-SCP method, introduced by Papadopoulos (2014), directly uses the original multi-label dataset without transforming it. It splits the dataset as

$$Z = Z_{tr} \cup Z_{cal} \tag{19}$$

The LP strategy treats each unique label-set as a distinct class. A multi-class classifier is trained on Z_{tr}, and nonconformity scores are computed on Z_{cal}. For a test instance x_{n+1}, p-values are calculated for each candidate label-set $\mathcal{Y}_c \in \mathcal{P}(\mathcal{L})$.

The prediction region includes all label-sets whose p-values exceed the significance level:

$$\Gamma^{\varepsilon}_{x_{n+1}} = \left\{\mathcal{Y}_c \ : \ p(\mathcal{Y}_c) > \varepsilon\right\} \cup \left\{\arg max_{j=1,\cdots,d}\big(p_j(\lambda_j)\big)\right\}, \quad \text{for } j = i, \ldots, d. \tag{20}$$

For the prediction region $\Gamma^{\varepsilon}_{x_{n+1}} \subseteq \mathcal{P}(\mathcal{L})$, LP-SCP provides the same validity guarantee (17). This approach directly controls the probability of covering the full multi-label vector, at the cost of increased computational complexity and label sparsity.

Papadopoulos (2014) proposed a nonconformity measure for LP-SCP based on the sum of absolute differences between the predicted probabilities of the model and the multi-hot encoding of label-sets in $\mathcal{P}(\mathcal{L})$. This measure captures the degree of disagreement between the model's predicted probabilities and the label assignments. In addition to the base measure, the same work introduced a penalization condition that incorporates structural information from the training data. Specifically, a penalty term, scaled by a weight parameter, is added to the nonconformity score. This term penalizes label-sets of non resembled pairs of labels according the data in the proper-training set.

Building on this, Maltoudoglou et al. (2022) represented the relationship between probability vectors and label-sets as data points and used the Euclidean norm to compute nonconformity scores, enabling a more geometric interpretation. The authors also proposed a computationally efficient method for handling datasets with a large number of labels. This method reduces the number of candidate label-sets in the power-set by eliminating those that are guaranteed to have p-values below a given significance level. The approach calculates the change in nonconformity score for each label resulting from adding it to or removing it from the predicted label-set. These changes are sorted in descending order, and their cumulative sum is computed until it reaches a specified threshold corresponding to the significance level ε. The number of terms included in this cumulative sum determines a subset of the $\mathcal{P}(\mathcal{L})$ that is guaranteed to include all label-sets of the conformal prediction region.

The recent work of Tyagi and Guo (2023) presented another a method of reducing the possible label-sets of the power-set $\mathcal{P}(\mathcal{L})$. Particularly, they proposed a tree-based conformal prediction method that accounts for label dependencies and uncertainty. Their approach applies hierarchical clustering on label-sets, effectively capturing the relationships among labels. The method formulates the prediction task as a multiple hypothesis testing problem within this hierarchical structure. By applying split-conformal prediction, marginal p-values are obtained for each hypothesis. Two hierarchical testing procedures are then employed: a standard hierarchical Bonferroni procedure and a modified version designed to control the family-wise error rate (FWER).

In summary, while IR-SCP and BR-SCP provide scalable marginal guarantees, LP-SCP is the only framework that guarantees validity without relying on conservative approximations. A comprehensive comparison of the three SCP methods can be found in (Wang et al., 2015).

3. Multilabel SCP with Extended Mahalanobis Measure

Within the LP-SCP framework for Multi-label classification our recent work (Katsios and Papadopoulos, 2024) proposed the Mahalanobis nonconformity measure. This approach effectively integrates the geometric robustness of Mahalanobis distance with the formal calibration properties of conformal prediction and offers reliable uncertainty quantification in

multi-label classification, yielding prediction sets that adapt to the dependencies among labels and predicted probabilities. In this section we present an extension of the Mahalanobis nonconformity measure. Our work was inspired by (Papadopoulos, 2014), where nonconformity scores are penalized for candidate label-sets containing pairs of labels that were not observed in the proper-training set.

3.1. SCP with Mahalanobis Measure

We define nonconformity using the Mahalanobis distance, a measure that accounts for the correlation structure among label-wise prediction errors. Each input instance is represented by a vector $\boldsymbol{x}_i = (x_{i_1}, ..., x_{i_s})$, where $x_i \subseteq \mathbb{R}^s$ and s is the number of attributes. Moreover, we convert the label-sets ψ_i into multi-hot vectors $\mathbf{y}_i = (\mathsf{y}_{i_1}, ..., \mathsf{y}_{i_d})$, where $\mathsf{y}_{i_j} = 1$ if $\lambda_j \in \psi_i$ and 0 otherwise, for every $\lambda_j \in \mathcal{L}$ with $j = 1, ..., d$. In this way, we form the subspace $Y = \{0, 1\}^d$ of multi-hot vectors in correspondence with the set Ψ. Under this conversion, we denote the dataset with the multi-hot vectors as $\mathcal{Z} = \big\{(\boldsymbol{x}_i, \mathbf{y}_i) \ : \ x_i \in X, \ \mathsf{y}_i \in Y\big\}$ for $i = 1, ..., n$. For each pair $(\boldsymbol{x}_i, \mathbf{y}_i)$ we compute the *error vector* $\boldsymbol{r}_i = |\mathbf{y}_i - \boldsymbol{o}(\boldsymbol{x}_i)|$, where $\boldsymbol{o}(\boldsymbol{x}_i) = (o_{i_1}, ..., o_{i_d})$ denotes the underlying model's predicted label probabilities for instance $\boldsymbol{x}_i$ as a vector of $\mathbb{R}^s$, with $o_{i_k} \in [0, 1]$ for $k = 1, ..., d$. The Mahalanobis nonconformity scores are then calculated as

$$\alpha_{\boldsymbol{x}_i}^{\mathbf{y}_i} = \sqrt{\boldsymbol{r}_i^T \ \Sigma^{-1} \ \boldsymbol{r}_i} \ , \tag{21}$$

where Σ is the covariance matrix of error vectors obtained from the proper-training set, $\boldsymbol{r}_i$ for $i = q+1, ..., n$ is the error vector of calibration instances, while $\boldsymbol{r}_{n+1}$ is the error vector of the test instance assigned candidate label-set $(\boldsymbol{x}_{n+1}, \mathbf{y_c})$. Rather than computing p-values, one can alternatively determine a threshold value such that any label with a nonconformity score below this threshold is included in the prediction region. To do so, the calibration scores are sorted in ascending order and denoted as $a_{i'}^{desc}$ for $i' = 1, ..., n - q$, satisfying $a_1^{desc} < ... < a_{n-q}^{desc}$. The index of the threshold value, for any given significance level ε, is the minimum integer $i'_\varepsilon \in \{1, ..., n - q\}$. This is calculated by the following formula,

$$i'_\varepsilon = \lfloor \varepsilon(n - q + 1) \rfloor. \tag{22}$$

Given i'_ε, the prediction set for a new instance x_{n+1} at the ε significance level the threshold value is $a_{i'_\varepsilon}^{desc}$. The prediction region is written in the form,

$$\Gamma_{x_{n+1}}^{\varepsilon} = \big\{\boldsymbol{y}_c \in Y \ : \ a_{x_{n+1}}^{\boldsymbol{y}_c} \leq a_{i'_\varepsilon}^{desc}\big\}. \tag{23}$$

3.2. Structural Penalties: Hamming Distance and Cardinality

To further incorporate structural similarity with previously seen label-sets, we introduce a Hamming distance penalty (Hp) and Cardinality penalty (Cp), which quantify how dissimilar a candidate label vector is from the empirical distribution of label-sets in the proper-training data. These terms serve as regularizers that bias the nonconformity scores toward candidate label vectors that are closer to those encountered in the training data, thereby reducing the likelihood of selecting implausible label-sets in the prediction region.

Hamming Distance Penalty. The Hamming distance penalty (Hp) quantifies how much a candidate label vector differs from the label-sets observed in the proper-training data. In the context of multi-label classification, the normalized Hamming distance (Hd) between two multi-hot vectors is defined as the proportion of components on which they differ from being equal. For two multi-hot vectors $\mathbf{y}_1 = (\mathsf{y}_{1_1}, \ldots, \mathsf{y}_{1_d})$ and $\mathbf{y}_2 = (\mathsf{y}_{2_1}, \ldots, \mathsf{y}_{2_d})$ the normalized Hamming distance is given by:

$$Hd = \frac{1}{d} \sum_{k=1}^{d} \mathbf{y}_{1_k} \oplus \mathbf{y}_{2_k}, \tag{24}$$

where $\oplus$ denotes the bitwise XOR operation, and d is the total number of labels. The value of Hd lies in the interval $[0, 1]$, with 0 indicating identical label vectors and 1 indicating complete dissimilarity.

We now define the Hamming distance penalty used in our framework:

Definition 1 *We define the Hamming distance penalty (Hp) for a possible label-set $\boldsymbol{y}_c \in Y$ as the minimum value of the normalized Hamming distances between the label-set $\boldsymbol{y}_c$ and every proper-training label-set $\mathbf{y}_i$,*

$$Hp_{y_c} = min_{i=1,\ldots,q} \left(\frac{1}{d} \sum_{k=1}^{d} \mathbf{y}_{i_k} \oplus \boldsymbol{y}_{c_k} \right). \tag{25}$$

Cardinality Penalty. To further guide the prediction model toward realistic label combinations, we introduce the Cardinality penalty, which penalizes candidate label-sets whose number of active labels deviates from the cardinality numbers observed in the proper-training set.

For the q multi-hot label vectors in the proper-training set $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_q\} \subseteq \{0, 1\}^d$, where each $\mathbf{y}_i = (\mathsf{y}_{i1}, \ldots, \mathsf{y}_{id})$ represents the labels for the i-th instance, and d is the total number of labels. The cardinality of a label vector $\mathbf{y}_i$ is defined as:

$$\mathrm{Card}(\mathbf{y}_i) = \sum_{k=1}^{d} \mathsf{y}_{i_k}. \tag{26}$$

The empirical frequency distribution $f_{card} : \{0, 1, \ldots, d\} \to [0, 1]$ of cardinalities is defined as

$$f_{card}(b) = \frac{1}{q} \sum_{i=1}^{q} \mathbb{I}\left[\mathrm{Card}(\boldsymbol{y}_i) = b\right], \quad \text{for } b = 0, \ldots, d, \tag{27}$$

where $\mathbb{I}[\cdot]$ is the indicator function that returns 1 if the condition is true and 0 otherwise. The calculation (27) gives the empirical probability that a randomly selected label vector from the proper-training set has cardinality b. By construction, the empirical distribution satisfies,

$$\sum_{b=0}^{d} f_{card}(b) = 1. \tag{28}$$

Using (27), we define the Cardinality penalty as follows:

Definition 2 *Let $Card(\boldsymbol{y}_c)$ denote the cardinality of a candidate label-set $\boldsymbol{y}_c$. We define the Cardinality penalty (Cp) for the $\boldsymbol{y}_c$ as*

$$Cp_{\boldsymbol{y}_c} = 1 - f_{card}\big(Card(\boldsymbol{y}_c)\big). \tag{29}$$

Both Hp and Cp are computed as preprocessing steps for all candidate vectors in Y, using Hamming distance and cardinality derived from the proper-training label vectors.

3.3. Extended Mahalanobis Measure

The Hp and Cp penalties are incorporated additively into the Mahalanobis nonconformity score to form an extended score that favors structurally plausible label-sets.

The new nonconformity measure is defined as follows:

Definition 3 *For an instance x and a multi-hot y, the extended Mahalanobis nonconformity measure is written in the form*

$$\mathrm{a}_{\boldsymbol{x}}^{\mathbf{y}} = \sqrt{\boldsymbol{r}^T\ \Sigma^{-1}\ \boldsymbol{r}} + \mu \cdot Hp_{\mathbf{y}} + \nu \cdot Cp_{\mathbf{y}}, \tag{30}$$

where $\boldsymbol{r} = |\mathbf{y} - \boldsymbol{o}(\boldsymbol{x})|$ is the error vector of the pair $(\boldsymbol{x}, \mathbf{y})$, $Hp_{\mathbf{y}}$ and $Cp_{\mathbf{y}}$ are the structural penalties associated with the multi-hot vector $\boldsymbol{y}$ and $\mu, \nu \in \mathbb{R}$.

The parameters μ and ν control the relative influence of the structural penalties Hp and Cp, respectively, in the extended nonconformity measure. Specifically, μ determines the degree to which the Hamming-based penalty affects the nonconformity score, thereby modulating the preference for label-sets that are closer to those observed in the proper-training data. Similarly, ν adjusts the impact of the cardinality penalty Cp, which biases predictions toward label-sets with more plausible label cardinalities.

The complete algorithm of LP-SCP with the extended Mahalanobis measure is given in Algorithm 1.

4. Experiments on Multi-label Datasets

In this section, we evaluate the performance of the LP-SCP framework using the proposed extension of the Mahalanobis nonconformity measure, comparing it against the standard Mahalanobis measure. The evaluation focuses on the effectiveness of incorporating structural penalties into the nonconformity score, particularly in terms of the resulting reduction to the size of prediction regions

4.1. Experimental setting

For experimenting, we employ three multi-label datasets, the Emotions, the PlantPseAAC and the Yeast dataset, with distinct properties. Table 1 provides detailed information on the datasets, including the number of instances, attributes, labels, average cardinality[1] and density[2]. Our experiments were performed following a 10-fold cross-validation process, which was repeated 10 times. The results were calculated as the average over all folds and repetitions.

1. Average Cardinality : measures the average number of labels associated with each instance
2. Density : cardinality divided by the number of labels

Algorithm 1: LP-SCP using Extended Mahalanobis Measure

Input:

- Classifier' s predicted probabilities for:
  - proper-training data $\boldsymbol{o}(\boldsymbol{x}_i)$, $i = 1, ..., q$
  - calibration data $\boldsymbol{o}(\boldsymbol{x}_i)$, $i = q+1, ..., n$
  - test instance $o(\boldsymbol{x}_{n+1})$
- Label-sets of:
  - proper-training data $\mathbf{y}_i$, $i = 1, ..., q$
  - calibration data $\mathbf{y}_i$, $i = q+1, ..., n$
- Parameters μ, ν
- Required significance level ε

1. Preprocessing on proper-training data:
   - Calculate the error vectors $\boldsymbol{r}_i = |\boldsymbol{o}(\boldsymbol{x}_i) - \mathbf{y_i}|$, $i = 1, ..., q$
   - Form the covariance matrix Σ using error vectors $\boldsymbol{r}_i$, $i = 1, ..., q$
   - Generate all binary vectors Y
   - For each $\boldsymbol{y}_c \in Y$ calculate $Hp_{\boldsymbol{y}_c}$ using (25)
   - For each $\boldsymbol{y}_c \in Y$ calculate $Cp_{\boldsymbol{y}_c}$ using (29)
2. Preprocessing on calibration data:
   - Calculate the calibration nonconformity scores $\mathrm{a}_{\boldsymbol{x}_i}^{\mathbf{y_i}}$, $i = q+1, ..., n$, using (30)
   - Sort calibration scores in descending order $\mathrm{a}_{i'}^{desc}$, $i' = 1, ..., n-q$
   - Calculate i'_ε using (22)
3. Calculate scores $\mathrm{a}_{\boldsymbol{x}_{n+1}}^{\boldsymbol{y}_c}$, for every possible label-set $\boldsymbol{y_c} \in Y$, using (30)

Output: Predicted set, $\Gamma^{\varepsilon}_{\boldsymbol{x}_{n+1}} = \left\{\boldsymbol{y}_c \in Y \ : \ \mathrm{a}_{\boldsymbol{x}_{n+1}}^{\boldsymbol{y}_c} \leq \mathrm{a}_{i'_\varepsilon}^{desc}\right\}$

Table 1: Dataset Characteristics

Dataset	Instances	Attributes	Labels	Average Cardinality	Density
Emotions	593	72	6	1.868	0.311
PlantPseAAC	978	452	12	1.078	0.089
Yeast	2417	103	14	4.237	0.302

For LP-SCP the training set of each fold was further divided into proper-training, validation and calibration sets. In particular, the partition of each dataset is given in Table 2.

Table 2: Dataset Partition

Dataset	Proper-training	Validation	Calibration	Test
Emotions	354	80	99	59
PlantPseAAC	704	132	176	97
Yeast	1522	327	653	241

In our experiments, we employ the XGBoost algorithm as the underlying classifier to address binary classification tasks for each label independently. To this end, we utilized the MultiOutputClassifier from scikit-learn library to facilitate simultaneous prediction of multiple labels in the dataset. For each label, an individual XGBoost model was trained with a logistic objective, incorporating techniques such as early stopping and validation on a per-label basis to prevent overfitting.

No hyperparameter tuning was performed, as the goal of this study was not to optimize classification performance, but rather to evaluate the effects of the proposed extensions to the nonconformity scores. A standard configuration of XGBoost is used to ensure stable and reasonably good baseline performance. Note that the proposed nonconformity measures are independent of the underlying model and can be applied with any suitable base classifier, depending on the characteristics of the dataset.

In table 3, we present the performance of the underlying XGBoost classifier (with calibration set included in the training set) on the three datasets in terms of four standard multi-label classification metrics.

Table 3: Underlying Classifier Evaluation with Standard Multi-label Metrics

	Emotions	PlantPseAAC	Yeast
Hamming loss	0.199	0.090	0.197
Accuracy	0.270	0.073	0.150
F1 Micro	0.651	0.135	0.642
F1 Macro	0.617	0.065	0.377

4.2. Evaluation Criteria

We evaluate the performance of the LP-SCP framework using two key metrics, the N-criterion (see (Vovk et al., 2016)), which assesses the size of the prediction regions, and the empirical validity, which measures the actual coverage achieved at a specified significance level.

To analyze the impact of structural penalties on the effectiveness of LP-SCP, we report the average size of the prediction regions using the N-criterion. This metric quantifies the mean number of label-sets included in the prediction region across all test instances for a given significance level ε. Formally, the N-criterion is defined as:

$$N(\varepsilon) = \frac{1}{g} \sum_{i=1}^{g} \left| \Gamma^{\varepsilon}_{\boldsymbol{x}_i} \right|, \tag{31}$$

where $\Gamma^{\varepsilon}_{\boldsymbol{x}_i}$ denotes the prediction region for the i-th test instance at significance level ε, and g is the total number of test instances. A smaller N-criterion indicates more statistically efficient predictions, as fewer candidate label-sets are retained.

To assess the validity of LP-SCP, we compute coverage, which represents the proportion of test instances for which the true label vector falls within the predicted region. This metric evaluates whether the conformal predictor satisfies its theoretical coverage guarantee. The empirical validity is given by

$$\hat{C}(\varepsilon) = \frac{1}{g} \sum_{i=1}^{g} \mathbb{I}\{\boldsymbol{y}_{n+i} \in \Gamma^{\varepsilon}_{\boldsymbol{x}_{n+i}}\}, \tag{32}$$

where $\boldsymbol{y}_{n+i}$ is the true multi-label vector for the i-th test instance, $\Gamma^{\varepsilon}_{\boldsymbol{x}_{n+i}}$ is the corresponding prediction region, and $\mathbb{I}\{\cdot\}$ is the indicator function. A conformal predictor is considered valid if $\hat{C}(\varepsilon)$ closely approximates the nominal coverage level $1 - \varepsilon$.

4.3. Individual Effect of Structural Penalties

This subsection presents experimental results evaluating the impact of the structural penalties Hp and Cp, each applied with a weight of 1. Their effectiveness is compared against the baseline Mahalanobis nonconformity measure, which does not incorporate any structural penalty. Figures 1, 2, and 3 display the N-criterion values for significance levels in range $[0.01, 0.1]$. The penalty configurations are denoted as weight pairs (μ, ν), where μ is the weight assigned to Hp and ν to C_p. Each line in the plots corresponds to one of the following configurations: (0,0) for the standard Mahalanobis measure, (1,0) for Mahalanobis with only the Hp penalty, and (0,1) for Mahalanobis with only the Cp penalty.

For the Emotions dataset (Figure 1), both structural penalties lead to a consistent reduction in the size of the prediction regions compared to the unpenalized Mahalanobis baseline (configuration (0,0)). The Hp penalty (configuration (1,0)) yields the most substantial reduction across all significance levels, followed closely by the Cp penalty (configuration (0,1)). This suggests that label dependencies, which are captured effectively by Hp, play a dominant role in this dataset's label structure.

For the PlantPseAAC dataset (Figure 2), the pattern differs. The Cp penalty (configuration (0,1)) significantly outperforms both the baseline and the Hp penalty (configuration

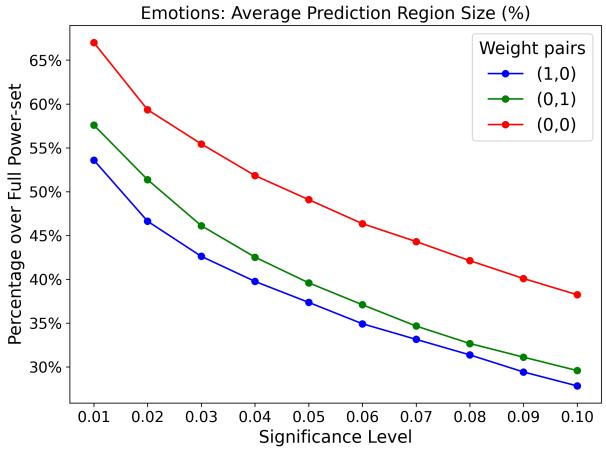


Figure 1: Individual effect of structural penalties per significance level on the Emotions dataset.

(1,0)), particularly at lower significance levels ($\varepsilon < 0.03$), where it results in highly compact prediction regions. This indicates that deviations in label cardinality represent a more informative structural indicator in this dataset. The Hp penalty also reduces region size compared to the baseline, but to a lesser degree and more uniformly across all levels.

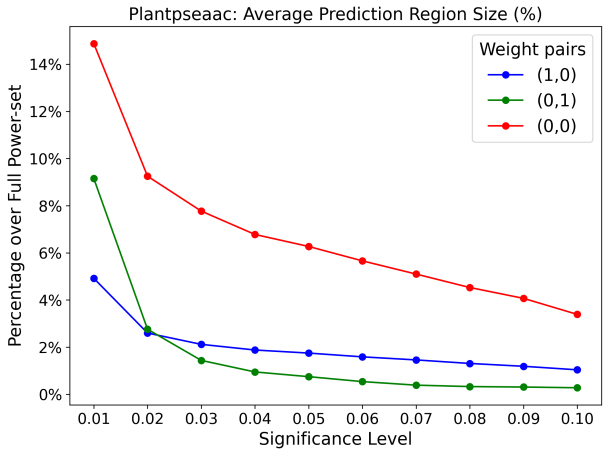


Figure 2: Individual Effect of structural penalties per significance level on the PlantPseAAC dataset.

In the case of the Yeast dataset (Figure 3), the Hp penalty (configuration (1,0)) again leads to a significant reduction in prediction region size compared to the baseline. This effect can be attributed to the Hamming distance penalty's tendency to enforce stricter

label agreement, thereby reducing the prediction regions. On the other hand, using only the Cp penalty (configuration (0,1)) produces slightly larger prediction regions.

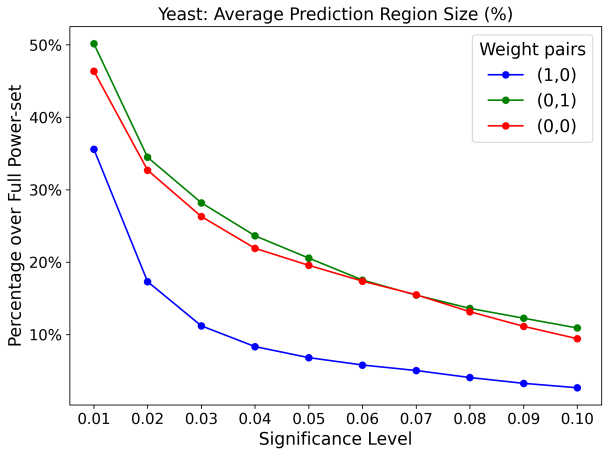


Figure 3: Individual Effect of structural penalties per significance level on the Yeast dataset.

Overall, the results demonstrate that the effectiveness of structural penalties is dataset-dependent. However, in all cases except for Cp in the Yeast dataset, they significantly reduce prediction region sizes. The small increase in the case of Cp for yeast may be due to the relatively high weight we used, as indicated by the results in the next section.

4.4. Combined Effect of Structural Penalties

In this subsection, we evaluate the combined effect of the structural penalties with different weight combinations on the resulting prediction region sizes. The weights for the hamming loss and cardinality penalties were varied across a grid search list for the weight parameter values of $\frac{1}{4}, \frac{1}{2}, 0, 1, 2$, and the resulting prediction region sizes were studied at significance levels ranging from 0.01 to 0.1. The focus is examining the overall improvement of the proposed nonconformity measure extension, as well as the effect of the penalty weights on different datasets.

Particularly, from all the obtained results, the top three combinations and the worst combination of penalty weights - producing the smallest and the largest average prediction region sizes, respectively - at significance level 0.01 were identified. To ease interpretation across datasets, the prediction region sizes were converted to percentages of the full power-set that they correspond to $\left(\frac{prediction\ region\ size}{2^d}\right.$, where d is the number of classes$\left.\right)$.

In Figures 4(a), 5(a) and 6(a), the prediction region size percentages for each of the top three and the worst weight combinations are plotted against significance level, together with those of the baseline Mahalanobis measure (weight configuration $(0, 0)$) for comparison. Additionally, Figures 4(b), 5(b) and 6(b) present the average and standard deviation of the prediction region size percentages for all 24 combinations (excluding (0,0)) across the grid search, again with those of the baseline Mahalanobis measure. Moreover, in Tables 4 - 6, we

present the percentage point improvement of the top three and worst penalty combinations, as well as the average percentage point improvement for all 24 combinations in the grid search over the standard Mahalanobis measure, at significance levels from 0.01 to 0.05.

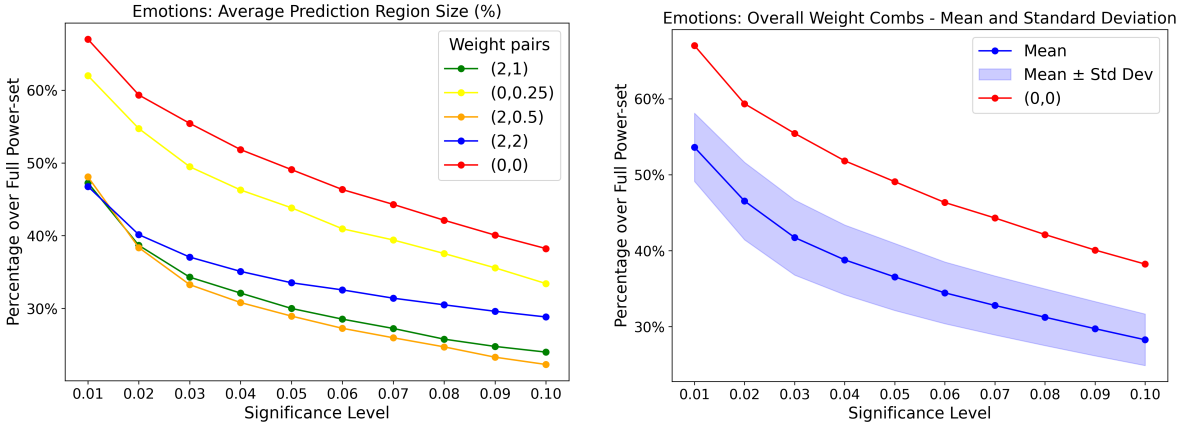
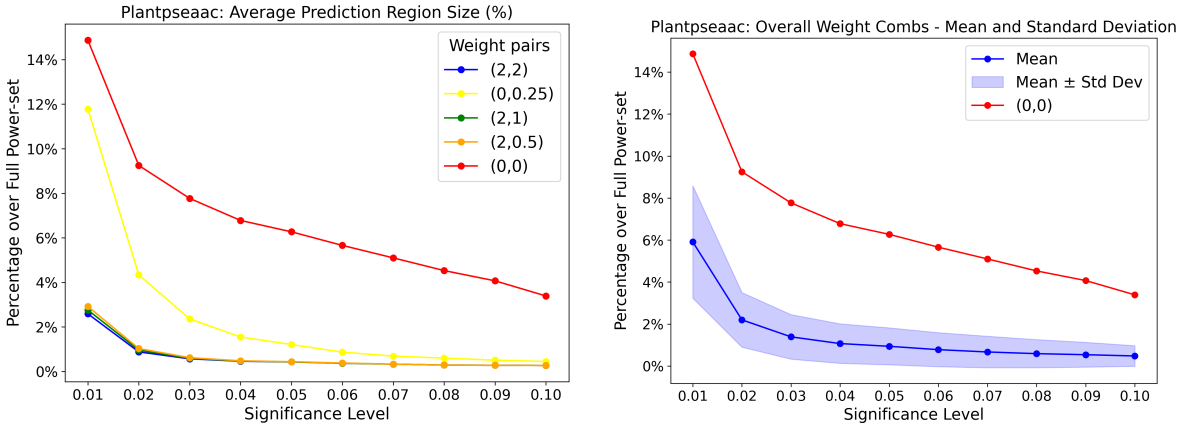


Figure 4: Effect of different weighted combinations of structural penalties per significance level on the Emotions dataset comparing the base Mahalanobis measure $(0, 0)$ with:
4(a): the three best performing weight pairs $(2, 0.5)$, $(2, 1)$ $(2, 2)$ and the worst preforming weight pair $(0, 0.25)$,
4(b): average and standard deviation of all 24 combinations of the grid search.

Table 4: Improvement in Percentage Points on the Emotions dataset

	(0,0.25)	(2,0.5)	(2,1)	(2,2)	Overall Average
0.01	5.00	18.91	19.73	20.23	13.39
0.02	4.62	20.98	20.67	19.20	12.81
0.03	5.96	22.17	21.14	18.38	13.71
0.04	5.55	21.02	19.72	16.74	13.04
0.05	5.25	20.13	19.08	15.55	12.54

Overall, the combined effect of the structural penalties yields a consistent trend of reduced prediction region sizes across all three datasets. At the lowest significance level of 0.01, the application of penalties leads to notable reductions. For example, in the emotions dataset, the top-performing weight configuration $(2, 2)$ reduces the region size from 67% (baseline) to just 46.77%, marking a reduction of 20.23 percentage points. Similarly, in the Yeast dataset, $(2, 0)$ decreases the region size from 46.37% to 28%, while in the PlantPseAAC dataset, $(2, 2)$ achieves a reduction from 14.87% to 2.59%.



(a) (b)

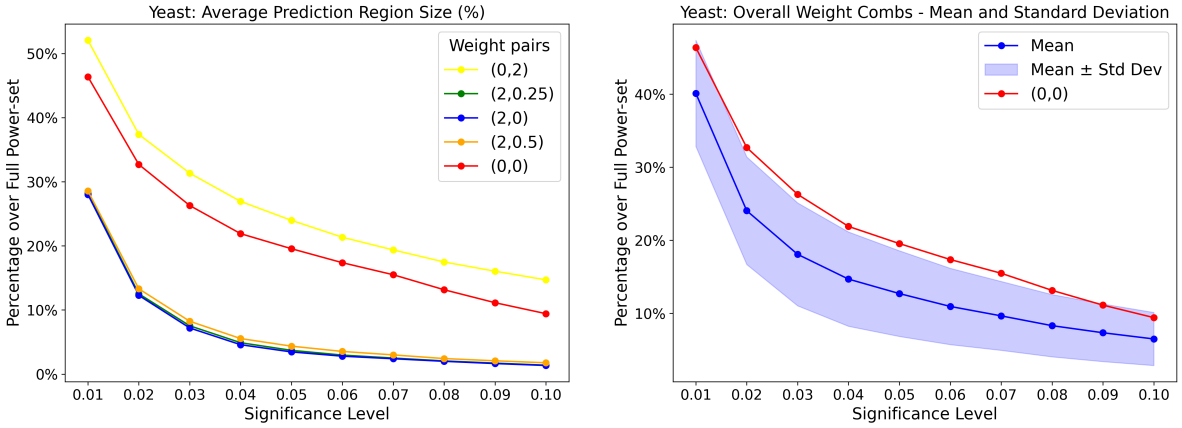
Figure 5: Effect of different weighted combinations of structural penalties per significance level on the PlantPseAAC dataset comparing the base Mahalanobis measure $(0, 0)$ with:
5(a): the three best performing weight pairs $(2, 0.5)$, $(2, 1)$ $(2, 2)$ and the worst preforming weight pair $(0, 0.25)$,
5(b): average and standard deviation of all 24 combinations of the grid search.

Table 5: Improvement in Percentage Points on the PlantPseAAC dataset

	(0,0.25)	(2,0.5)	(2,1)	(2,2)	Average Improvement
0.01	3.09	11.95	12.11	12.28	8.96
0.02	4.91	8.22	8.29	8.37	7.05
0.03	5.41	7.15	7.18	7.20	6.38
0.04	5.23	6.30	6.31	6.32	5.71
0.05	5.06	5.83	5.84	5.84	5.33

Both of the Figures 4 - 6 and the Tables 4 - 6 show that a rather significant improvement is achieved across significance levels of all datasets. Even the worst weight combinations reduce the prediction region sizes with the exception of the Yeast dataset. Note that this only happens when the weight of the Hp penalty is set to 0. Figures 4(b) - 6(b) and the last columns of Tables 4 - 6 show that a significant improvement cab be achieved even without an extensive fine tuning of the weights.

Regarding the influence of the penalty weights across all datasets, the maximum tested value of 2 for Hp consistently appears in the top-performing combinations, indicating that label-wise agreement via the Hamming-based penalty is highly effective. On the other hand, the effect of the cardinality-based penalty Cp is more dataset-dependent. For Emotions and PlantPseAAC, the best values range from 0.5 to 2, while for Yeast, lower values between



(a) (b)

Figure 6: Effect of different weighted combinations of structural penalties per significance level on the Yeast dataset comparing the base Mahalanobis measure $(0, 0)$ with:
6(a): the three best performing weight pairs $(2, 0)$, $(2, 0.25)$ $(2, 0.5)$ and the worst preforming weight pair $(0, 2)$,
6(b): average and standard deviation of all 24 combinations of the grid search.

Table 6: Improvement in Percentage Points on the Yeast dataset

	(0,2)	(2,0.5)	(2,0.25)	(2,0)	Average Improvement
0.01	-5.70	17.80	18.16	18.37	6.26
0.02	-4.69	19.39	20.17	20.42	8.64
0.03	-5.04	18.05	18.76	19.10	8.21
0.04	-5.03	16.36	17.01	17.32	7.21
0.05	-4.42	15.20	15.85	16.10	6.84

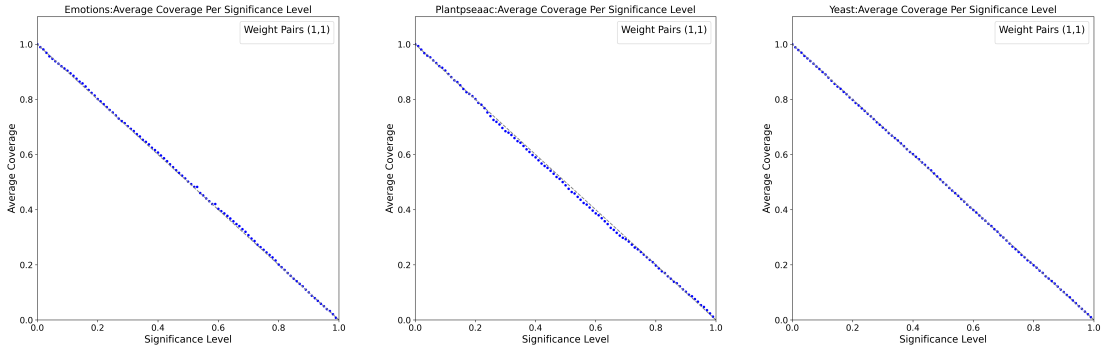
0 and 0.5 work better. In the PlantPseAAC dataset, where label cardinality distributions are more concentrated, higher weights such as 2 or 1 for Cp perform best, contributing significantly to the reduction. In contrast, for the Yeast dataset, higher values of Cp are less effective and can even worsen results - as shown by the weight configuration $(0, 2)$ in 6(a) - suggesting that cardinality penalties are not helpful when label cardinality is more varied. The weight value 0.5 for the Cp penalty strike a good balance for the three datasets.

Overall, the experimental results presented in this subsection show that both structural penalties, Hp and Cp, can contribute significantly to reducing prediction region sizes.

4.5. Empirical Validity

Figure 7 shows the coverage of the prediction regions across all significance levels for the three datasets, using the sampled weight pair (1,1). For all datasets, the coverage closely

follows the diagonal line, indicating a strong match between the nominal and empirical coverage rates. The same behaviour was repeated for all weights of combinations.



(a) Emotions (b) PlantPseAAC (c) Yeast

Figure 7: Coverage of same weight pair $(1, 1)$ for the three datasets.

5. Conclusions and Future Work

This study proposed an extended Mahalanobis nonconformity measure for multilabel classification under the LP-SCP framework, incorporating structural penalties based on Hamming distance and label cardinality deviation from proper-training label-sets. These penalties increase the nonconformity of unlikely label combinations according to the training data, and therefore focus prediction regions on the most likely label-sets. Our experimental results on three datasets demonstrated that the incorporation of structural penalties can lead to significantly more compact prediction regions.

Our examination of the individual penalties reveals that their impact is dataset-dependent. The Hamming-based penalty (Hp) has the strongest effect on the Emotions and Yeast datasets, consistently leading to improved performance when assigned higher weights. In contrast, the cardinality-based penalty (Cp) has the greatest influence on the PlantPseAAC dataset, where label cardinalities are more concentrated. When the penalties are combined, performance improves even further across all datasets. In particular, the best results are obtained when Hp is set to its highest tested value of 2, emphasizing the benefit of enforcing label-wise agreement. For Cp, optimal values vary, lower weights (e.g., 0.5) are more effective for Yeast, while higher weights (e.g., 1 or 2) perform better on PlantPseAAC and Emotions. In all cases, the combined approach yields performance improvements ranging from 15 to 20 percentage points for the Emotions dataset, 5 to 12 percentage points for the PlantPseAAC dataset, and 15 to 20 percentage points for the Yeast dataset, compared to the standard Mahalanobis-based nonconformity measure without penalties.

Our future plans include the examination of improvements to the penalty functions and especially Cp, and the inclusion of a regularization parameter for the covariance matrix. Further experimentation on different datasets is also important to test how well the method generalizes.

References

Giannis Haralabopoulos, Ioannis Anagnostopoulos, and Derek McAuley. Ensemble deep learning for multilabel binary classification of user-generated content. *Algorithms*, 13(4): 83, 2020.

Kostas Katsios and Harris Papadopoulos. Multi-label conformal prediction with a mahalanobis distance nonconformity measure. *Proceedings of Machine Learning Research*, 230: 1–14, 2024.

Antonis Lambrou and Harris Papadopoulos. Binary relevance multi-label conformal predictor. In *Conformal and Probabilistic Prediction with Applications*, pages 90–104. Springer, 2016.

Lysimachos Maltoudoglou, 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.

Andrew Maxwell, Runzhi Li, Bei Yang, Heng Weng, Aihua Ou, Huixiao Hong, Zhaoxian Zhou, Ping Gong, and Chaoyang Zhang. Deep learning architectures for multi-label classification of intelligent health risk prediction. *BMC bioinformatics*, 18:121–131, 2017.

Harris Papadopoulos. A cross-conformal predictor for multi-label classification. In *Artificial Intelligence Applications and Innovations: AIAI 2014 Workshops: CoPA, MHDW, IIVC, and MT4BD, Rhodes, Greece, September 19-21, 2014. Proceedings 10*, pages 241–250. Springer, 2014.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *Machine learning: ECML 2002: 13th European conference on machine learning Helsinki, Finland, August 19–23, 2002 proceedings 13*, pages 345–356. Springer, 2002a.

Harris Papadopoulos, Vladimir Vovk, and Alexander Gammerman. Qualified prediction for large data sets in the case of pattern recognition. In *ICMLA*, pages 159–163, 2002b.

Johannes Rückert, Louise Bloch, Raphael Brüngel, Ahmad Idrissi-Yaghir, Henning Schäfer, Cynthia S Schmidt, Sven Koitka, Obioma Pelka, Asma Ben Abacha, Alba G. Seco de Herrera, et al. Rocov2: Radiology objects in context version 2, an updated multimodal image dataset. *Scientific Data*, 11(1):688, 2024.

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

Chhavi Tyagi and Wenge Guo. Multi-label classification under uncertainty: A tree-based conformal prediction approach. In *Conformal and Probabilistic Prediction with Applications*, pages 488–512. PMLR, 2023.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Vladimir Vovk, Valentina Fedorova, Ilia Nouretdinov, and Alexander Gammerman. Criteria of efficiency for conformal prediction. In *Conformal and Probabilistic Prediction with Applications: 5th International Symposium, COPA 2016, Madrid, Spain, April 20-22, 2016, Proceedings 5*, pages 23–39. Springer, 2016.

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, Ilia 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, Egham, UK, April 20-23, 2015, Proceedings 3*, pages 241–250. Springer, 2015.

Zhi Zhong, Masato Hirano, Kazuki Shimada, Kazuya Tateishi, Shusuke Takahashi, and Yuki Mitsufuji. An attention-based approach to hierarchical multi-label music instrument classification. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.