Using Credal C4.5 for Calibrated Label Ranking in Multi-Label Classification

Serafín Moral-García SERAMORAL@DECSAI.UGR.ES
Javier G. Castellano FJGC@DECSAI.UGR.ES
Carlos J. Mantas CMANTAS@DECSAI.UGR.ES
Joaquín Abellán JABELLAN@DECSAI.UGR.ES
Department of Computer Science and Artificial Intelligence, University of Granada, Spain

Abstract

The Multi-Label Classification (MLC) task aims to predict the set of labels that correspond to an instance. It differs from traditional classification, which assumes that each instance has associated a single value of a class variable. Within MLC, the Calibrated Label Ranking algorithm (CLR) considers a binary classification problem for each pair of labels to determine a label ranking for a given instance, exploiting in this way correlations between pairs of labels. Moreover, CLR mitigates the class imbalance problem that frequently appears in MLC motivated by the fact that, in MLC, there are usually very few instances that have associated a certain label. For solving the binary classification problems, a traditional classification algorithm is needed. The C4.5 algorithm, based on Decision Trees, has been widely employed in this domain. In this work, we show that the Credal C4.5 method, a version of C4.5 recently proposed that uses imprecise probabilities, is more suitable than C4.5 for solving the binary classification problems in CLR. An exhaustive experimental analysis carried out in this research shows that Credal C4.5 performs better than C4.5 when both algorithms are employed in CLR, being the improvement more notable as there is more noise in the labels.

Keywords: Multi-Label Classification, Calibrated Label Ranking, base classifier, imprecise probabilities, C4.5, Credal C4.5, noise

1. Introduction

In Machine Learning, the Multi-Label Classification task (MLC) aims to predict the set of labels that correspond to an instance. It differs from traditional classification, where each instance has associated a single value of a class variable. MLC is suitable to be used in many fields such as *text categorization* [16, 23], *image recognition* [20], or *biology* [7, 6].

The simplest approach to MLC might be the Binary Relevance method (BR) [8]. It considers a binary classification problem per label, in which the class variable indicates whether an instance has associated the corresponding label or not. Despite its simplicity, this method has achieved good results in practice [13]. Nevertheless, this method assumes that all the labels are independent, which might not be realistic. The Classifier Chain algorithm (CC) [22] considers if the instances have associated the previous labels according to an established label order. In this way, CC exploits some correlations between labels. However, it must be remarked that the label order strongly influences the performance of CC.

Furthermore, in MLC, there are usually very few instances that have associated a certain label. Hence, in BR, the binary classification problems often have a class-imbalance problem. In consequence, it might be difficult for the binary classifiers to predict that an instance has associated a certain label. The same happens in CC.

The Calibrated Label Ranking algorithm (CLR), proposed by Fürnkranz et al. [11], considers a binary classification problem for each pair of labels to determine, for a given instance, a ranking of labels. Such a classification problem considers, from the original training set, the instances that have associated one of the two labels and not the other one. The class variable indicates which of the labels is associated with each instance. Thus, CLR exploits correlations between pairs of labels and mitigates the class imbalance problem that often appears in BR and CC.

In CLR, a traditional classification algorithm is employed for solving each one of the binary classification problems, which is known as the *base classifier*. Within traditional classification, Decision Trees (DTs) are known to be very simple, transparent, and interpretable models. One of the most utilized traditional classification methods based on DTs is the C4.5 algorithm [21]. This method utilizes uncertainty measures based on classical Information Theory to build the tree.

DTs based on imprecise probabilities were proposed by Abellán and Moral [5]. They are called Credal Decision Trees (CDTs). Such DTs use a building process based on uncertainty measures on closed and convex sets of probability distributions, also called credal sets. CDTs have obtained satisfactory performance, especially when data contains class noise [4, 3, 2]. Within CDTs, Mantas and Abellán [14] proposed a new version of C4.5 based on imprecise probabilities: the Credal C4.5 algorithm (CC4.5). Mantas and Abellán [14], Mantas et al. [15], showed via

exhaustive experimentation that C4.5 and CC4.5 perform equivalently when there is no class noise and that CC4.5 significantly outperforms C4.5 with class noise in the data.

Real datasets may have intrinsic noise, i.e despite the fact that they have not been manipulated, they might contain errors. As pointed by Moral-García et al. [18], Moral-García et al. [19], the intrinsic noise in MLC tends to be higher than in traditional classification. BR and CC have obtained better results with CC4.5 as the base classifier than with C4.5, especially when there is noise in the labels (For more details, see [17] and [18], respectively). Also, the new adaptation of DTs to MLC based on imprecise probabilities, proposed by Moral-García et al. [19], performs significantly better than the existing one based on classical Probability Theory, especially with noise in the labels.

Summarizing, CLR is useful to exploit pairwise label correlations in MLC and mitigates the class-imbalance problem that usually appears in this domain, the intrinsic noise in MLC may be higher than in traditional classification, and the CC4.5 algorithm has provided good results with noise in the data. For these reasons, it is worth analyzing the use of CC4.5 as the base classifier of CLR. Thus, the performance of CC4.5 as the base classifier of CLR is studied in this research, checking whether it improves C4.5.

An exhaustive experimental analysis is carried out in this work with several MLC datasets, noise levels, and many MLC evaluation metrics to compare the performance of CC4.5 versus C4.5 as the base classifiers of CLR. Such an experimental analysis reveals that, in general, CC4.5 obtains better results than C4.5 when both algorithms are used to solve the binary classification problems in CLR, being the improvement more notable as there is more noise in the labels.

This paper is arranged as follows: The Multi-Label Classification paradigm is explained in Section 2. Section 3 describes the Calibrated Label Ranking method. The Credal C4.5 algorithm is exposed in Section 4. Section 5 explains the Calibrated Label Ranking algorithm with Credal C4.5 as the base classifier. The experimental analysis carried out in this work is detailed in Section 6. Section 7 concludes the paper and provides ideas for future research.

2. Multi-Label Classification

The Multi-Label Classification task (MLC) starts from a d-dimensional attribute space $\mathscr{X} \subseteq \mathbb{R}^d$ and a label set $\mathscr{Y} = \{y_1, y_2, \ldots, y_q\}$, where $q > 1$.

As in traditional classification, in MLC, a model is learned from a dataset of N instances $\mathscr{D} = \{(\mathbf{x_i}, \mathbf{Y_i}), \quad i = 1, \ldots, N\}$. For the i-th instance, $\mathbf{x_i} \in \mathscr{X}$ is its attribute vector (d-dimensional) and $\mathbf{Y_i} \subseteq \mathscr{Y}$ is its label set. If $y_j \in \mathbf{Y_i}$, i.e if the instance $\mathbf{x_i}$ has associated the label y_j, y_j is said to be relevant for $\mathbf{x_i}$. Else, y_j is said to be irrelevant for $\mathbf{x_i}$, $\quad \forall i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, q$.

The model learned from the training set let us predict the set of relevant labels for an instance. It is given by a function $h : \mathscr{X} \to 2^{\mathscr{Y}}$ that, for an instance described via an attribute vector $\mathbf{x} \in \mathscr{X}$, returns the set of labels that are predicted to be associated with $\mathbf{x}$.

In many situations, the learned model consists of a real-valued function $f : \mathscr{X} \times \mathscr{Y} \to \mathbb{R}$ that, for a given instance $\mathbf{x} \in \mathscr{X}$, and a label $y_j \in \mathscr{Y}$, with $1 \leq j \leq q$, predicts the posterior probability that $\mathbf{x}$ has associated y_j. The real-valued function f, for a given instance $\mathbf{x} \in \mathscr{X}$, gives rise to a ranking function $rank\_f_{\mathbf{x}} : \mathscr{Y} \to \{1, 2, \ldots, q\}$. It is implicitly determined verifying that $rank\_f_{\mathbf{x}}(y_k) > rank\_f_{\mathbf{x}}(y_j)$ $\forall y_j, y_k \in \mathscr{Y}$ such that $f(\mathbf{x}, y_k) < f(\mathbf{x}, y_j)$.

3. Calibrated Label Ranking

The Calibrated Label Ranking algorithm, proposed by Fürnkranz et al. [11], transforms the MLC task into a label ranking problem, where the score for each label is determined through comparisons between that label and the remaining ones.

Specifically, for each pair of labels y_j, y_k, with $1 \leq j < k \leq q$, a binary classification model is learned. For that model, the instances for which one of the labels is relevant and the other one is irrelevant are selected. For each one of these instances, its attribute set is considered, as well as the relevance of the two labels for it. Formally:

$$\mathscr{D}_{jk} = \left\{ (\mathbf{x_i}, \phi(\mathbf{Y_i}, y_j)) \mid \phi(\mathbf{Y_i}, y_j) \neq \phi(\mathbf{Y_i}, y_k), 1 \leq i \leq N \right\}, \tag{1}$$

where

$$\phi(\mathbf{Y_i}, y_j) = \left\{ \begin{array}{lll} 1 & if & y_j \in \mathbf{Y_i} \\ 0 & if & y_j \notin \mathbf{Y_i} \end{array} \right\}, \quad \forall j = 1, 2, \ldots, q. \tag{2}$$

Using this training set, a binary classifier $h_{jk} : \mathscr{X} \to \{0, 1\}$ is induced via a traditional classification algorithm $\mathscr{B}$. This classifier, for a given instance, gives a prediction about the relative relevance of y_j versus y_k. The algorithm $\mathscr{B}$ is known as the *base classifier*.

When a new instance $\mathbf{x} \in \mathscr{X}$ is wanted to be classified, the relative relevances are predicted on the learned classifiers. For each label y_j, the number of votes in the classifiers corresponding to its comparisons with the rest of the labels is counted:

$$\mathscr{C}(\mathbf{x}, y_j) = \sum_{k=1}^{j-1} \left[\left[h_{kj}(\mathbf{x}) = 0 \right] \right] + \sum_{k=j+1}^{q} \left[\left[h_{jk}(\mathbf{x}) = 1 \right] \right], \tag{3}$$

where the function $[[cond]]$, being *cond* a logical condition, takes the value 1 if the condition holds and 0 otherwise.

The real-valued function $f : \mathbf{X} \times \mathbf{Y} \to \mathbb{R}$ is obtained via a normalization of the number of votes determined via

Equation (3):

$$f(\mathbf{x},y_j)=\frac{\mathscr{C}(\mathbf{x},y_j)}{q}. \tag{4}$$

Once reached this point, we have a label ranking for $\mathbf{x}$. The next step consists of distinguishing between relevant and irrelevant labels for $\mathbf{x}$. For this purpose, the CLR algorithm considers q additional binary classifiers $\{h_j, \quad j=1,2,\ldots,q\}$ using the same traditional classification algorithm $\mathscr{B}$. For each one of them h_j, $1\leq j\leq q$, the original training set is considered, taking for each instance its attribute set and the relevance of y_j for it, as the Binary Relevance method (BR) [8]. Formally:

$$\mathscr{D}_j=\left\{(\mathbf{x}_i,\phi(\mathbf{Y_i},y_j), \quad i=1,2,\ldots,N\right\}. \tag{5}$$

For each label, the number of votes given by Equation (3) is incremented in one value if BR predicts that the label is relevant for the instance. Thus, the final number of votes for each label is given by:

$$\mathscr{C}^*(\mathbf{x},y_j)=\mathscr{C}(\mathbf{x},y_j)+h_j(\mathbf{x}), \quad \forall j=1,2,\ldots,q. \tag{6}$$

The number of labels that BR predicts as irrelevant for $\mathbf{x}$ is also considered:

$$\mathscr{C}'^*(\mathbf{x})=\sum_{j=1}^{q}\left[[h_j(\mathbf{x})=0]\right]. \tag{7}$$

The set of labels predicted as relevant by CLR for $\mathbf{x}$ is composed of those labels for which the final number of votes, determined via Equation (6), is higher than the number of labels that BR predicts as irrelevant for $\mathbf{x}$:

$$h(\mathbf{x})=\left\{y_j \mid \mathscr{C}^*(\mathbf{x},y_j)>\mathscr{C}'^*(\mathbf{x}), \quad 1\leq j\leq q\right\}. \tag{8}$$

4. Credal C4.5

The C4.5 algorithm [21] is a well-known traditional classification method based on Decision Trees. A new version of this algorithm, called Credal C4.5 (CC4.5), was proposed by Mantas and Abellán [14]. Both algorithms principally differ in the split criterion.

Let C be the class variable and $\{c_1,\ldots,c_K\}$ its set of possible values. Let $\mathscr{D}$ be the dataset in a certain node and X an attribute whose set of possible values is $\{x_1,\ldots,x_n\}$.

The split criterion utilized in C4.5 considers the Shannon entropy [24] of the class variable C, defined as follows:

$$S(C)=-\sum_{j=1}^{K}P(C=c_j)\log_2 P(C=c_j), \tag{9}$$

being $P(C=c_j)$ the probability that C takes the c_j value, estimated via relative frequencies, $\forall j=1,2,\ldots,K$.

The basis of the split criterion employed in C4.5 is the Information Gain (IG), defined by:

$$IG(C,X)=S(C)-\sum_{i=1}^{n}P(X=x_i)S(C|X=x_i), \tag{10}$$

where $P(X=x_i)$ is the probability that $X=x_i$, estimated through relative frequencies, and $S(C|X=x_i)$ is the Shannon entropy of C on the partition of $\mathscr{D}$ composed of those instances for which $X=x_i$, $\quad \forall i=1,2,\ldots,n$.

The split criterion of C4.5 is the Information Gain Ratio (IGR), which derives from IG by normalizing by the attribute entropy ($S(X)$):

$$IGR(C,X)=\frac{IG(C,X)}{S(X)}. \tag{11}$$

The CC4.5 algorithm uses a split criterion similar to IGR. However, unlike C4.5, CC4.5 employs imprecise probabilities for the split criterion.

Specifically, CC4.5 is based on the Imprecise Dirichlet Model (IDM) [26], a formal imprecise probabilities model. According to the IDM, the probability that C takes its possible value c_j belongs to the following interval:

$$P(C=c_j)\in\mathscr{I}_j=\left[\frac{n(c_j)}{N+s},\frac{n(c_j)+s}{N+s}\right], \quad \forall j=1,2,\ldots,K, \tag{12}$$

where N is the size of the dataset, $n(c_j)$ is the number of instances in the dataset that satisfy $C=c_j$, $\quad \forall j=1,2,\ldots K$, and $s>0$ is a given parameter of the model.

The probability intervals determined by Equation (12), $\mathscr{I}_j$, $j=1,2,\ldots,K$, give rise to the following closed and convex set of probability distributions, also called credal set, on C [1]:

$$\mathscr{P}^{\mathscr{D}}(C)=\left\{p\in\mathscr{P}(C)\mid p(c_j)\in\mathscr{I}_j, \quad \forall j=1,\ldots,K\right\}, \tag{13}$$

where $\mathscr{P}(C)$ is the set of all probability distributions on C.

The basis of the split criterion used in CC4.5 is the maximum of entropy on the IDM credal set on C:

$$S^*(\mathscr{P}^{\mathscr{D}}(C))=max\left\{S(p)\mid p\in\mathscr{P}^{\mathscr{D}}(C)\right\}. \tag{14}$$

The maximum of entropy is a well-established total uncertainty measure on credal sets [12]. In [14], the algorithm that allows obtaining the probability distribution that attains $S^*(\mathscr{P}(C))$ can be found.

Now, the Imprecise Information Gain (IIG) [5] is defined in the following way:

$$IIG(C,X)=S^*(\mathscr{P}^{\mathscr{D}}(C))-\sum_{i=1}^{n}p^{\mathscr{D}}(x_i)S^*(\mathscr{P}^{\mathscr{D}}(C\mid X=x_i)), \tag{15}$$

being $\mathscr{P}^{\mathscr{D}}(C\mid X=x_i)$ the IDM credal set on C on the partition of $\mathscr{D}$ composed of those instances for which $X=x_i$, $\quad \forall i=1,2,\ldots,n$, and $p^{\mathscr{D}}$ the probability distribution that reaches the maximum of entropy on the IDM credal set on X.

The split criterion employed in CC4.5, called Imprecise Information Gain Ratio [14], is defined as follows:

$$IIGR(C,X)=\frac{IIG(C,X)}{S^*(\mathscr{P}^{\mathscr{D}}(X))}, \tag{16}$$

where $\mathscr{P}^{\mathscr{D}}(X)$ is the IDM credal set on X and $S^*\left(\mathscr{P}^{\mathscr{D}}(X)\right)$ is the maximum of entropy on $\mathscr{P}^{\mathscr{D}}(X)$.

The choice of the s parameter is an essential point. It is easy to observe that IDM intervals are wider as the s value is higher. Hence, the s parameter indicates how reliable are the data. Walley [26] does not give a definitive recommendation about the parameter. Nevertheless, it suggests two values: $s = 1$ and $s = 2$. In addition, the procedure to obtain the maximum of entropy with the IDM reaches its lowest computational cost when $s = 1$ [1]. For these reasons, we use the value $s = 1$ in this work.

5. Calibrated Label Ranking with Credal C4.5

As exposed in Section 3, the CLR method builds a binary classification problem for each pair of labels. For such a classification problem, those instances for which one of the two labels is relevant and the other one irrelevant are used. In this research, we use the CC4.5 algorithm for handling these binary classification problems. Additionally, for each label, CLR builds a binary classifier that predicts the relevance of the corresponding label for an instance, as the BR method. For building these classifiers, we also utilize CC4.5.

When a new instance is wanted to be classified, for each label, the number of favorable votes in the classifiers corresponding to the comparisons with the rest of the labels is counted. This gives rise to a label ranking, and the posterior probabilities of the relevance of the labels for the instance can be obtained from the number of votes for each label. To predict the set of relevant labels for the instance, the number of votes for each label is incremented in one if BR predicts that the instance has associated that label. Then, a label is predicted as relevant for the instance if, and only if, the number of votes for the label is higher than the number of labels that BR predicts as irrelevant for the instance.

Algorithm 1 summarizes the CLR procedure with CC4.5, where we use the same notation as in Section 3.

In this work, we study the use of CC4.5 as the base classifier of CLR, comparing it with C4.5. Hence, in Section 5.1, the advantages of CC4.5 over C4.5 are explained.

for $j = 1$ ***to*** $q-1$ **do**
  for $k = j+1$ ***to*** q **do**
    $\mathscr{D}_{jk} = \left\{(\mathbf{x_i}, \phi(\mathbf{Y_i}, y_j)) \mid \phi(\mathbf{Y_i}, y_j) \neq \phi(\mathbf{Y_i}, y_k)\right\}$,
    Build a binary classifier h_{jk} from $\mathscr{D}_{jk}$ using CC4.5.
  end
end
for $j = 1$ ***to*** q **do**
  $\mathscr{D}_j = \left\{(\mathbf{x}_i, \phi(\mathbf{Y_i}, y_j)), \quad i = 1, \ldots, N\right\}$,
  Build a binary classifier h_j from $\mathscr{D}_j$ using CC4.5.
end
When a new instance $\mathbf{x}$ is wanted to be classified:
for $j = 1$ ***to*** q **do**
  $\mathscr{C}(\mathbf{x}, y_j) = \sum_{k=1}^{j-1}\left[\left[h_{kj}(\mathbf{x}) = 0\right]\right] + \sum_{k=j+1}^{q}\left[\left[h_{jk}(\mathbf{x}) = 1\right]\right]$,
  $f(\mathbf{x}, y_j) = \frac{\mathscr{C}(\mathbf{x}, y_j)}{q}$,
  $\mathscr{C}^*(\mathbf{x}, y_j) = \mathscr{C}(\mathbf{x}, y_j) + h_j(\mathbf{x})$.
end
$\mathscr{C}'^*(\mathbf{x}) = \sum_{j=1}^{q}[[h_j(\mathbf{x}) = 0]]$,
$h(\mathbf{x}) = \left\{y_j \mid \mathscr{C}^*(\mathbf{x}, y_j) > \mathscr{C}'^*(\mathbf{x}), \quad 1 \leq j \leq q\right\}$.

Algorithm 1: Procedure of Calibrated Label Ranking with Credal C4.5.

5.1. C4.5 VS Credal C4.5

Regarding the differences between the behavior of C4.5 and Credal C4.5, the following points are remarkable:

- When $s = 0$, C4.5 and CC4.5 are identical. If $s > 0$, IDM credal sets are smaller as the number of instances in the dataset is higher. In this way, at the upper levels of the tree, where there are often many instances, S is quite close to S^* and, thus, IGR and IIGR provide similar values. In contrast, at the lower levels of the tree, where there are usually very few instances, IDM credal sets may contain many probability distributions that differ very much from the one that attains S. So, in these cases, the values of IGR and IIGR can be very different since S and S^* may have pretty different values. Therefore, C4.5 and CC4.5 have a similar behavior at the upper levels of the tree, while, at the lower levels, they might select different split attributes.

- The value of IIGR for an attribute can be negative, unlike IGR [14]. In consequence, in CC4.5, via IIGR, it is avoided to choose those attributes that worsen the uncertainty-based information about the class variable, which implies that CC4.5 may stop branching the tree before C4.5. Hence, with CC4.5, there might be less data overfitting than with C4.5.

- According to the examples shown and results proved by Mantas et al. [15], the maximum of entropy on IDM credal sets, S^*, is less sensitive to noise than the Shannon entropy S. In this way, C4.5 is less robust to noise than CC4.5 and, thus, CLR is more robust to label noise with CC4.5 than with C4.5.

Summarizing, CLR mitigates the class imbalance problem that frequently appears in MLC and allows exploiting correlations between pairs of labels; C4.5 is one of the most known traditional classification algorithms; in MLC, there might be more intrinsic noise than in traditional classification [18, 19], and CC4.5 is less sensitive to noise than C4.5. For these reasons, it is worth analyzing the performance of CLR using CC4.5 as the base classifier.

6. Experimentation

6.1. Experimental Settings

- **Datasets**:

  In our experimental research, twelve datasets have been used, which can be downloaded from the official website of Mulan [25][1]. Table 1 shows the main characteristics of each dataset: domain, number of instances, number of continuous and discrete attributes, number of labels, average number of labels per instance, i.e label cardinality, and label density (average proportion of labels that are relevant for an instance).

Table 1: Datasets employed in our experimentation. N is the size of the dataset, N_DA and N_CA are, respectively, the number of discrete and continuous attributes, N_L is the number of labels, L_C is the label cardinality, and L_D is the label density.

Dataset	Domain	N	N_CA	N_DA	N_L	L_C	L_D
bibtex	Text	7395	0	1836	159	2.4	0.015
birds	Multimedia	645	258	2	19	1.014	0.053
cal500	Multimedia	502	68	0	174	26.044	0.15
corel5k	Multimedia	5000	0	499	374	3.52	0.009
emotions	Multimedia	593	72	0	6	1.87	0.311
enron	Text	1702	0	1001	53	3.38	0.064
flags	Multimedia	194	10	9	7	3.392	0.485
genbase	Biology	662	0	1186	27	1.252	0.046
mediamill	Multimedia	43907	120	0	101	4.38	0.043
medical	Text	978	0	1449	45	1.24	0.028
scene	Multimedia	2407	294	0	6	1.07	0.179
yeast	Biology	2417	103	0	14	4.24	0.303

- **Evaluation Measures**:

  We employ the same notation as in Section 2. Let $\mathscr{D}_{Test} = \{(\mathbf{x_i}, \mathbf{Y_i}), \quad i = 1,2,\ldots,N_{Test}\}$ be the test set, being N_{Test} the number of test instances, $\mathbf{x_i} \in \mathscr{X}$ the attribute vector of the test i-th instance, and $\mathbf{Y_i} \subseteq \mathscr{Y}$ its label set, $\quad \forall i = 1,2,\ldots,N_{Test}$.

  In this experimentation, we use the following evaluation metrics:

  – **Hamming Loss**: It is the proportion of pairs of label-instance incorrectly classified:

$$Hamming\_Loss = \frac{1}{N_{Test}} \sum_{i=1}^{N_{Test}} \frac{1}{q} \sum_{i=1}^{q} |h(\mathbf{x_i}) \triangle \mathbf{Y_i}|, \tag{17}$$

  where $\triangle$ is the number of elements belonging to one set but not to the other one, i.e the symmetric difference between two sets.

  – **Subset Accuracy**: It is defined as the proportion of instances for which the predicted label set coincides with the set of labels associated with the instance:

$$Subset\_Accuracy = \frac{1}{N_{Test}} \sum_{i=1}^{N_{Test}} [[h(\mathbf{x_i}) = \mathbf{Y_i}]]. \tag{18}$$

  – **Accuracy**: It is defined as the average Jaccard similarity coefficient between the set of relevant labels for an instance and the predicted label set for it:

$$Accuracy = \frac{1}{N_{Test}} \sum_{i=1}^{N_{Test}} \frac{|h(\mathbf{x_i}) \cap \mathbf{Y_i}|}{|h(\mathbf{x_i}) \cup \mathbf{Y_i}|}. \tag{19}$$

  – **F1**: It is determined by:

$$F1 = \frac{1}{N_{Test}} \sum_{i=1}^{N_{Test}} \frac{2 \times |h(\mathbf{x_i}) \cap \mathbf{Y_i}|}{|h(\mathbf{x_i})| + |\mathbf{Y_i}|}. \tag{20}$$

  – **One Error**: It indicates the proportion of instances for which the label with the highest predicted posterior probability is not relevant:

$$1\_E = \frac{1}{N_{Test}} \sum_{i=1}^{N_{Test}} [[\arg \max_{j=1,\ldots,q} f(\mathbf{x_i}, y_j) \notin \mathbf{Y_i}]]. \tag{21}$$

  – **Coverage**: It measures the average number of labels that are necessary to go down the label ranking for covering all the ones associated with an instance:

$$Coverage = \frac{1}{N_{Test}} \sum_{i=1}^{N_{Test}} \max_{j=1,2,\ldots,q} rank\_f_{\mathbf{x_i}}(y_j) - 1. \tag{22}$$

  – **Ranking Loss**: It is the average proportion of pairs of relevant-irrelevant labels reversely ordered:

$$Ranking\_Loss = \frac{1}{N_{Test}} \sum_{i=1}^{N_{Test}}, \frac{|\mathbf{Z_i}|}{|\mathbf{Y_i}| \times |\overline{\mathbf{Y_i}}|}, \tag{23}$$

  being $\overline{\mathbf{Y_i}}$ is the complementary set of $\mathbf{Y_i}$, and $\mathbf{Z_i} = \{(y_n, y_m) \mid rank\_f_{\mathbf{x_i}}(y_m) > rank\_f_{\mathbf{x_i}}(y_n), y_m \in \mathbf{Y_i}, y_n \in \overline{\mathbf{Y_i}}\}, \quad \forall i = 1,2,\ldots,N_{Test}$.

  – **Average Precision**: It indicates the average proportion of labels with a higher predicted posterior probability than a relevant label.

  Formally, for each $i = 1,2,\ldots,N_{Test}$, and for each $y_j \in \mathbf{Y_i}$, $1 \leq j \leq q$, let us consider $\Lambda_{i,j} = \{y_k \mid rank\_f_{\mathbf{x_i}}(y_k) < rank\_f_{\mathbf{x_i}}(y_j), 1 \leq k \leq q\}$. Average Precision is defined in the following way:

$$Avg\_Prec = \frac{1}{N_{Test}} \sum_{i=1}^{N_{Test}} \frac{1}{|\mathbf{Y_i}|} \sum_{y_j \in \mathbf{Y_i}} \frac{|\Lambda_{i,j}|}{rank\_f_{\mathbf{x_i}}(y_j)}. \tag{24}$$

1. http://mulan.sourceforge.net/datasets-mlc.html

- **Algorithms and Parameters**:

  The implementation given in Mulan for CLR has been used in this experimentation. For this MLC algorithm, two base classifiers have been considered: C4.5 and Credal C4.5 (CC4.5). As argued in Section 4, the value of the s parameter for CC4.5 has been fixed to 1. We have employed the implementations available in the Weka software [28] for C4.5 and CC4.5. The rest of the parameters used for all the algorithms in this experimentation have been the ones given by default in the corresponding software.

- **Procedure**:

  In our experimental analysis, three noise levels have been considered: 0%, 5%, and 10%. For each noise level and each dataset, the following cross-validation procedure of 5 folds has been carried out: the dataset has been divided into five partitions and, for each one of them, an iteration has been done, in which the corresponding partition has been used for testing and the rest of data for training. For each label, the x% of the training instances (where x indicates the noise level) have been selected and the value of their label has been changed (if the label is relevant it has been changed to irrelevant and vice-versa). To create the partitions, we have used part of the functionality provided in Mulan. In each dataset, the same partitions have been used for C4.5 and CC4.5. The Weka filters have been employed for generating the noise, with the parameters given by default in this software (except for the corresponding noise level). The model has been learned using the noisy training set, and the evaluation metrics have been extracted with the test set.

- **Statistical Evaluation**:

  For each metric and noise level, there are two base classifiers of CLR to compare: C4.5 and CC4.5. Hence, following the recommendations given by Demšar [10], Charte et al. [9] for statistical comparisons between two algorithms, the Wilcoxon test [27] has been used with a level of significance of $\alpha = 0.05$ to check which base classifier performs better and whether the differences are statistically significant.

6.2. Results and Discussion

Tables 2, 3, and 4 show a summary of the results obtained by CLR with C4.5 and CC4.5 with 0%, 5%, and 10% of noise introduced in the labels, respectively. Specifically, for each evaluation measure, they show which base classifier performs better according to the Wilcoxon test and whether the differences are statistically significant. Also, they show, for each evaluation metric, in how many datasets a base classifier gets a better result than the other one. We do not show the complete results here due to the lack of space.

Table 2: Summary of the results obtained by CLR for each measure without noise in the labels. (•) indicates that the base classifier of the column significantly improves the other one. (-) means that the algorithm of the column improves the other one as the base classifier of CLR, but the results are statistically equivalent.

Metric	C4.5	CC4.5	Wins C4.5	Wins CC4.5
Hamming Loss		(-)	3	7
Subset Accuracy		(-)	4	6
Accuracy		(-)	5	7
F1		(-)	5	7
Coverage	(-)		6	5
Ranking Loss	(-)		7	4
Average Precision	(-)		6	5
One-Error		(-)	5	6

Table 3: Summary of the results obtained by CLR for each measure when there is a 5% of noise introduced in the labels. (•) indicates that the base classifier of the column significantly improves the other one. (-) means that the algorithm of the column improves the other one as the base classifier of CLR, but the results are statistically equivalent.

Metric	C4.5	CC4.5	Wins C4.5	Wins CC4.5
Hamming Loss		(-)	3	8
Subset Accuracy		(-)	3	7
Accuracy		(•)	3	8
F1		(-)	3	8
Coverage		(-)	4	8
Ranking Loss		(-)	4	8
Average Precision		(-)	3	9
One-Error		(-)	3	8

Table 4: Summary of the results obtained by CLR for each measure when there is a 10% of noise introduced in the labels. (•) indicates that the base classifier of the column significantly improves the other one. (-) means that the algorithm of the column improves the other one as the base classifier of CLR, but the results are statistically equivalent.

Metric	C4.5	CC4.5	Wins C4.5	Wins CC4.5
Hamming Loss		(•)	2	10
Subset Accuracy		(•)	0	11
Accuracy		(•)	0	11
F1		(•)	1	11
Coverage		(-)	4	8
Ranking Loss		(•)	3	9
Average Precision		(•)	2	9
One-Error		(•)	2	9

We express the following comments about the obtained results:

- **Classification-based Metrics:**
  - When there is no label noise introduced in the data, the results are statistically equivalent according to the Wilcoxon test in all the classification-based metrics considered here. Nonetheless, in Hamming Loss, CLR performs better with CC4.5 than with C4.5 in seven datasets, while, in another three datasets, the opposite happens. Hence, without noise introduced in the labels, the predicted label sets differ less from the sets of relevant labels for the instances with CC4.5 than with C4.5, even though, in this case, the differences are not statistically significant.
  - With a 5% of noise introduced in the labels, the performance of CLR is better with CC4.5 than with C4.5 in all the classification-based metrics considered here. Indeed, in all of these metrics, the number of wins of CC4.5 is considerably higher than the number of wins of C4.5. Moreover, in Accuracy, the differences are statistically significant according to the Wilcoxon test. It implies that, when there is a 5% of label noise introduced in the data, CLR predicts the sets of relevant labels for the instances far better with CC4.5 than with C4.5.
  - The difference in the performance of CLR with C4.5 and CC4.5 in classification-based metrics is even more notable with a 10% of noise introduced in the labels. In this case, CLR performs significantly better with CC4.5 than with C4.5 in all the classification-based evaluation metrics, and the results obtained with CC4.5 are better than the ones achieved with C4.5 in almost all the datasets.
- **Ranking-based Metrics:**
  - As in classification-based evaluation metrics, when there is no noise introduced in the labels, the results obtained by CLR with C4.5 and CC4.5 are statistically equivalent according to the Wilcoxon test in all the ranking-based measures considered in this experimentation. Also, it can be observed that, in each metric, the number of wins (losses) of C4.5 is similar to the number of wins (losses) of CC4.5, except for Ranking Loss. In this evaluation measure, CLR obtains a better result with C4.5 in seven datasets and with CC4.5 in another three datasets. Thus, without noise introduced in the labels, CLR orders fewer pairs of relevant-irrelevant labels reversely using C4.5 as the base classifier, although the differences are not statistically significant.
  - When there is a 5% of noise introduced in the labels, there are also no statistically significant differences via the Wilcoxon test in any of the ranking-based measures utilized in this experimentation. Nevertheless, in all of these evaluation metrics, the number of datasets in which CC4.5 performs better than C4.5 as the base classifier of CLR is considerably higher than the number of datasets in which the opposite occurs. Therefore, with a 5% of noise introduced in the labels, CLR predicts much more appropriate posterior probabilities with CC4.5 than with C4.5.
  - With a 10% of noise introduced in the labels, the results obtained in ranking-based evaluation metrics are even more favorable to the use of CC4.5 as the base classifier of CLR. In fact, according to the Wilcoxon test, the performance of CC4.5 is significantly better than the one of C4.5 in all the ranking-based measures considered in this experimental research, except for Coverage, which indicates the average number of steps that are necessary to go down the label ranking to cover all the relevant labels for an instance. However, in this measure, CC4.5 obtains eight wins, whereas C4.5 performs better than CC4.5 in four datasets. In consequence, when a 10% of noise is introduced in the labels, CLR predicts much more suitable posterior probabilities of the relevance of the labels for the instances with CC4.5 than with C4.5.

Summary of the results: Without noise introduced in the labels, there are no statistically significant differences via the Wilcoxon test between the performance of C4.5 and CC4.5 as the base classifiers of CLR in any evaluation measure. However, in general, the number of wins of CC4.5 is considerably higher than the number of wins of C4.5. The results are more favorable to CC4.5 as there is more noise in the labels (more measures in which CC4.5 performs significantly better than C4.5 and more datasets in which the results are better with CC4.5 than with C4.5).

Thus, it can be stated that CLR performs better with CC4.5 as the base classifier than with C4.5, being the difference more notable as there is more noise in the labels. It is because, as argued in Section 5.1, CC4.5 is more suitable than C4.5 to handle noisy data, and the intrinsic noise in MLC tends to be higher than in traditional classification.

7. Conclusions and Future Work

In this research, the Calibrated Label Ranking algorithm for Multi-Label Classification has been considered. It considers a binary classification problem for each pair of labels to determine, for a given instance, a label ranking. Hence, it exploits correlations among pairs of labels. Furthermore, Calibrated Label Ranking mitigates the class imbalance problem that frequently appears in Multi-Label Classification because, in this field, there are often very few instances that have associated a certain label.

Since the C4.5 method is widely used in traditional classification and the Credal C4.5 algorithm outperforms C4.5 when there is noise in the data, in this work, we have analyzed the use of Credal C4.5 as the base classifier of Calibrated Label Ranking. It has been shown that Credal C4.5 is more appropriate than C4.5 to tackle the binary classification problems in Calibrated Label Ranking because C4.5 is more sensitive to noise than Credal C4.5, and the intrinsic noise in Multi-Label Classification may be higher than in traditional classification.

An exhaustive experimental analysis has been carried out in this work with several MLC datasets, noise levels, and many evaluation metrics for Multi-label Classification to compare the performance of Credal C4.5 and C4.5 as the base classifiers of Calibrated Label Ranking. Such an experimental analysis has highlighted that Credal C4.5 obtains better results than C4.5 without noise in the labels, although the differences are not statistically significant; Credal C4.5 performs significantly better than C4.5 in both classification-based and ranking-based evaluation metrics when there is noise in the labels, being the differences more significant as the noise level is higher. Therefore, the use of Credal C4.5 as the base classifier of Calibrated Label Ranking supposes an improvement over C4.5, especially with label noise in the data.

As future work, the results obtained here motivate us to analyze the use of imprecise probabilities in other Multi-Label Classification algorithms, checking whether it improves the use of precise probabilities. Also, others imprecise probabilities methods to exploit label correlations in Multi-Label Classification could be developed.

References

[1] Joaquín Abellán. Uncertainty measures on probability intervals from the imprecise dirichlet model. *International Journal of General Systems*, 35(5):509–528, 2006. doi: 10.1080/03081070600687643.

[2] Joaquín Abellán. Ensembles of decision trees based on imprecise probabilities and uncertainty measures. *Information Fusion*, 14(4):423–430, 2013.

[3] Joaquín Abellán and Carlos J. Mantas. Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, 41(8):3825 – 3830, 2014. ISSN 0957-4174. doi: 10.1016/j.eswa.2013.12.003.

[4] Joaquín Abellán and Andrés R. Masegosa. An experimental study about simple decision trees for bagging ensemble on datasets with classification noise. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 5590 of *Lecture Notes in Computer Science*, pages 446–456. Springer, 2009. ISBN 978-3-642-02905-9. doi: 10.1007/978-3-642-02906-6_39.

[5] Joaquín Abellán and Serafín Moral. Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems*, 18(12): 1215–1225, 2003. ISSN 1098-111X. doi: 10.1002/int.10143.

[6] R. T. Alves, M. R. Delgado, and A. A. Freitas. Knowledge discovery with artificial immune systems for hierarchical multi-label classification of protein functions. In *International Conference on Fuzzy Systems*, pages 1–8, 2010. doi: 10.1109/FUZZY.2010.5584298.

[7] Zafer Barutcuoglu, Robert E. Schapire, and Olga G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006. doi: 10.1093/bioinformatics/btk048.

[8] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757 – 1771, 2004. ISSN 0031-3203. doi: 10.1016/j.patcog.2004.03.009.

[9] Francisco Charte, Antonio J. Rivera, David Charte, María J. del Jesus, and Francisco Herrera. Tips, guidelines and tools for managing multi-label datasets: The mldr.datasets r package and the cometa data repository. *Neurocomputing*, In Press, 2018. ISSN 0925-2312. doi: 10.1016/j.neucom.2018.02.011.

[10] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006. ISSN 1532-4435.

[11] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencorthogonal a, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 2008. doi: 10.1007/s10994-008-5064-8.

[12] George J. Klir. *Uncertainty and Information: Foundations of Generalized Information Theory*. John Wiley And Sons, Inc., 2005. ISBN 9780471755579. doi: 10.1002/0471755575.

[13] Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Džeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45(9):3084 – 3104, 2012. ISSN 0031-3203. doi: 10.1016/j.patcog.2012.03.004.

[14] Carlos J. Mantas and Joaquín Abellán. Credal-C4.5: Decision tree based on imprecise probabilities to classify noisy data. *Expert Systems with Applications*, 41(10):4625 – 4637, 2014. ISSN 0957-4174. doi: 10.1016/j.eswa.2014.01.017.

[15] Carlos J. Mantas, Joaquín Abellán, and Javier G. Castellano. Analysis of Credal-C4.5 for classification in noisy domains. *Expert Systems with Applications*, 61:314 – 326, 2016. ISSN 0957-4174. doi: 10.1016/j.eswa.2016.05.035.

[16] Andrew McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI'99 Workshop on Text Learning.*, pages 1–7, 1999.

[17] Serafín Moral-García, Carlos J Mantas, Javier G Castellano, and Joaqu'ın Abellán. Using credal-c4.5 with binary relevance for multi-label classification. *Journal of Intelligent & Fuzzy Systems*, 35(6):6501–6512, 2018. doi: 10.3233/JIFS-18746.

[18] Serafín Moral-García, Carlos J Mantas, Javier G Castellano, and Joaquín Abellán. Ensemble of classifier chains and credal c4.5 for solving multi-label classification. *Progress in Artificial Intelligence*, 8(2): 195–213, 2019. doi: 10.1007/s13748-018-00171-x.

[19] Serafín Moral-García, Carlos J. Mantas, Javier G. Castellano, and Joaquín Abellán. Non-parametric predictive inference for solving multi-label classification. *Applied Soft Computing*, 88:106011, 2020. ISSN 1568-4946. doi: 10.1016/j.asoc.2019.106011.

[20] Gulisong Nasierding and Abbas Kouzani. Image to Text Translation by Multi-Label Classification. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, volume 6216 of *Lecture Notes in Computer Science*, chapter 31, pages 247–254. Springer, 2010. ISBN 978-3-642-14931-3. doi: 10.1007/978-3-642-14932-0_31.

[21] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

[22] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333, 2011. ISSN 1573-0565. doi: 10.1007/s10994-011-5256-5.

[23] Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2):135–168, 2000. ISSN 1573-0565. doi: 10.1023/A:1007649029923.

[24] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948. ISSN 1538-7305. doi: 10.1002/j.1538-7305.1948.tb01338.x.

[25] Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, and Ioannis Vlahavas. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414, 2011.

[26] Peter Walley. Inferences from multinomial data; learning about a bag of marbles (with discussion). *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):3–57, 1996. ISSN 00359246. doi: 10.2307/2346164.

[27] Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, 1945. ISSN 00994987. doi: 10.2307/3001968.

[28] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, second edition, 2005. ISBN 0120884070.