Non-Parametric Predictive Inference for solving Multi-Label Classification

S. Moral-García[a,*], Carlos J. Mantas[a], Javier G. Castellano[a], Joaquín Abellán[a]

[a]*Department of Computer Science and Artificial Intelligence*
University of Granada, Granada, Spain

Abstract

Decision Trees (DTs) have been adapted to Multi-Label Classification (MLC). These adaptations are known as Multi-Label Decision Trees (ML-DT). In this research, a new ML-DT based on the Nonparametric Predictive Inference Model on Multinomial data (NPI-M) is proposed. NPI-M is an imprecise probabilities model that provides good results when it is applied to DTs in standard classification. Unlike other models based on imprecise probabilities, NPI-M is a nonparametric approach and it does not make unjustified assumptions before observing data. It is shown that the new ML-DTs based on NPI-M is more robust to noise than the ML-DT based on precise probabilities. As the intrinsic noise in MLC might be higher than in traditional classification, it is expected that the new ML-DT based on the NPI-M outperforms the already existing ML-DT. This fact is validated with an exhaustive experimentation carried out in this work on different MLC datasets with several levels of added noise. In it, many MLC evaluation metrics are employed in order to measure the performance of the algorithms. The experimental analysis shows that the proposed ML-DT based on NPI-M obtains better results than the ML-DT that uses precise probabilities, especially when we work on data with noise.

Keywords:

[*] Corresponding Author

Email addresses: `seramoral@decsai.ugr.es` (S. Moral-García), `cmantas@decsai.ugr.es` (Carlos J. Mantas), `fjgc@decsai.ugr.es` (Javier G. Castellano), `jabellan@decsai.ugr.es` (Joaquín Abellán)

Multi-Label Classification, Multi-Label Decision Tree, NPI-M, Multi-Label Credal Decision Tree, noise

1. Introduction

The *Multi-Label Classification* (MLC) is a task within the Machine Learning field whose goal is to predict the set of labels that are associated with an instance. Nowadays, it is applied to many domains like *text categorization*, [1, 2], *biology* [3, 4] or *image recognition* [5]. In these domains, it is more suitable to consider that an example has associated multiple labels, instead of assuming that an instance only has a single value of the class variable, as in traditional supervised classification.

Many approaches to solving the MLC problem have been developed so far. A summary of most of them can be seen in [6]. Basically, there are two kinds of algorithms for MLC. On the one hand, *problem transformation methods* convert the MLC problem into multiple well-known classification tasks. On the other hand, the *algorithm adaptation methods* modify the already existing algorithms for traditional classification to solve the MLC problem.

Within the second group of MLC algorithms, quite simple, interpretable and transparent models such as the Decision Trees (DT) have been adapted to MLC. Specifically, in [7], the well-known C4.5 algorithm [8] was modified to solve the MLC task. It is called Multi-Label Decision Tree (ML-DT).

The adaptation of DT to MLC proposed so far, called ML-DT, uses precise probabilities theory in the split criterion. For traditional supervised classification, new DTs based on imprecise probabilities theory have been developed. They are called Credal Decision Trees (CDT) [9]. Basically, in the building process, they utilize uncertainty measures on credal sets (closed and convex sets of probability distributions). CDTs have been shown to have quite good performance [10, 11, 12], especially when the data contains class noise. Besides, it has been shown, via exhaustive experimentations carried out in [13, 14], that the version of C4.5 proposed, based on imprecise probabilities, called Credal C4.5 (CC4.5), outperforms C4.5 when class noise is introduced in the data.

The CDTs mentioned in the previous paragraph use the Imprecise Dirichlet Model (IDM) [15], a formal model of imprecise probabilities based on probability intervals. The IDM verifies several principles that have been claimed to be desirable for inference [15]. The most remarkable of them is the *representation invariance principle*, which establishes that inferences on future events should be independent of the arrangement and labeling of the sample space. However, IDM assumes prior knowledge about the data, which is not very desirable. It is explained in more detail in [15]. Moreover, the performance of the IDM has a strong dependence on a parameter, called s, when it is used in classification [16].

For the previous reasons, a Non-Parametric Model for Predictive Inference on Multinomial data (NPI-M) is proposed in [17]. This model does not make unjustified assumptions before observing data. Furthermore, NPI-M is a nonparametric approach.

The NPI-M has been shown to be more suitable than the IDM to be applied to CDTs for classification, as shown in [18]. Actually, the CDTs based on IDM depend strongly on the parameter s and CDTs with NPI-M always have an equivalent performance to CDTs based on IDM with the best s value.

Summarising, in classification, CDTs based on NPI-M are more robust to high levels of noise than DTs based on precise probabilities. In addition, in MLC the intrinsic noise might be higher than in standard classification. This is due to an obvious reason: The probability that one instance has an error in one label in MLC, where we have multiple tags, is higher than the probability that one example has an error in the class value in standard classification. For these reasons, in this research, a new adaptation of Decision Trees for MLC based on the NPI-M, called Multi-Label Credal Decision Tree (ML-CDT), is proposed.

An extensive experimental study is carried out in this work on different MLC datasets with several levels of added noise. In order to measure the performance of the algorithms, many evaluation metrics for MLC are utilized. This experimentation shows that, in general, ML-CDT has better performance than ML-DT, being the improvement more significative if there is more noise introduced in the data. Therefore, it is concluded that it is very suitable to apply the NPI-M to the adaptations of DTs to MLC, especially when the data contains noise.

The rest of this paper is structured as follows: Section 2 describes the necessary previous knowledge: The Multi-Label Classification paradigm, the Multi-Label Decision Tree, the probability intervals theory and the Nonparametric Predictive Inference Model on Multinomial data. The new adaptation of Decision Tree to Multi-Label Classification based on the NPI-M is exposed in Section 3. The experimental analysis carried out in this work is detailed in Section 4. Finally, Section 5 is devoted to the concluding remarks and future work.

2. Background

2.1. Multi-Label Classification

2.1.1. Paradigm

The Multi-Label Classification (MLC) assumes that multiple labels are associated with an instance, unlike traditional supervised classification, where each example only belongs to a single state of the class variable. Hence, the MLC task aims to predict the set of labels corresponding to an instance, described through a set of attributes.

Formally, in MLC, it is started from a d-dimensional space attribute $\mathbf{X} = \big(X^1, X^2, \ldots, X^d\big) \subseteq \mathbb{R}^d$ and a set of q labels $\mathbf{L} = \{l_1, l_2, \ldots, l_q\}$. Normally, $q > 1$.

As in traditional classification, in MLC, it is learned a model from a training set $\mathcal{D} = \{(\mathbf{x_i}, \mathbf{L_i}), 1 \leq i \leq m\}$ of m examples. It allows to predict the set of labels corresponding to an instance. For the i-th instance, $\mathbf{x_i}$ is its set of attributes and $\mathbf{L_i} \subseteq \mathbf{L}$ is its labels set, $1 \leq i \leq m$. The label l_j is said to be relevant for the instance $\mathbf{x_i}$ if the example has associated it. Else, it is said that the label l_j is irrelevant for $\mathbf{x_i}$.

The quality of the model, measured in terms of its complexity and its predictive capacity, is indicated via a criterion $\mathbf{Q}$.

The criterion $\mathbf{Q}$ is tried to be maximized by finding a function $h : \mathbf{X} \rightarrow 2^{\mathbf{L}}$ that, for each instance, gives its set of labels predicted as relevant for the example.

Alternatively, in many cases, the goal is to find a real-valuated function $f : \mathbf{X} \times \mathbf{L} \rightarrow \mathbb{R}$ that tries to maximize $\mathbf{Q}$. It is interpreted as follows: $f(\mathbf{x}, l_j)$ indicates, for an instance described by the set of attributes $\mathbf{x}$, the posterior probability that the label l_j is relevant for $\mathbf{x}$.

For an instance $\mathbf{x}$, a ranking function $rank\_f_{\mathbf{x}} : \mathbf{L} \rightarrow \{1, 2, \ldots, q\}$ is derived from f. It is implicitly defined verifying that if $f(\mathbf{x}, l_i) > f(\mathbf{x}, l_j)$ then $rank\_f_{\mathbf{x}}(l_i) < rank\_f_{\mathbf{x}}(l_j)$, where $l_i, l_j \in \mathbf{L}$.

It can be considered a threshold function $t : \mathbf{X} \rightarrow \mathbb{R}$. This function lets obtain the set of labels associated with $\mathbf{x}$ given f. In fact, the set of labels that are relevant for $\mathbf{x}$ can be extracted in the following way:

$$h(\mathbf{x}) = \{l_j \in \mathbf{L} \mid f(\mathbf{x}, l_j) > t(\mathbf{x}), 1 \leq j \leq q\}\,. \tag{1}$$

Basically, there are three options to calibrate the threshold:

- In some cases, a constant value, which is often equal to 0.5, is used for the function t. An example of this point can be seen in [7]. When it is disposed of all the new test examples, the constant can be fixed by minimizing the difference between the label cardinality, defined as the average number of labels per instance, in training and test sets. In [19] can be found an example of this issue.

- The second option is that the threshold function is induced from the training examples [20]. For instance, the function t is sometimes assumed to be linear. A more detailed explanation of this can be found in [21, 22].

- Finally, some algorithms, such as Calibrated Label Ranking (CLR) [23], start from the label ranking of the instances and employ their own mechanism for determining how many relevant labels has each example.

2.1.2. Main approaches to MLC

A large number of algorithms for MLC have been proposed so far. In [6] a summary of most of them is shown. Essentially, the MLC algorithms developed can be divided into two groups:

- **Problem transformation methods**: The algorithms that belong to this category transform the MLC problem into well-known classification scenarios. For instance, Binary Relevance [24] or Classifier Chains [19] consider a binary problem per label and after they combine the predictions in order to provide a MLC solution. Other algorithms, such as Calibrated Label Ranking [23] convert the MLC problem into the task of label ranking. On the other hand, the Random k-labelsets algorithm [25] transforms the MLC task into several multi-class classification problems, combining the solutions in order to provide a multi-label prediction.

- **Algorithm adaptation methods**: The algorithms of this kind tackle the MLC problem by adapting the already existing algorithms for traditional classification to MLC. For example, the K-Nearest Neighbour algorithm was adapted in [26]. In [7] it is proposed an adaptation of Decision Trees to MLC. Support Vector Machines were adapted to MLC in [27].

In this work we will focus on the adaptation of Decision Trees for Multi-Label Classification, which is explained in Section 2.2.

2.2. Multi-Label Decision Trees

As in DTs for traditional classification, in the adaptations of DTs to MLC, called Multi-Label Decision Trees (ML-DT) [7], each node represents an attribute or feature variable and, for each possible value of that attribute, there is an associated branch. In order to branch in each node, it is selected the attribute that provides the maximum information gain according to a split criterion. When entering a feature in a node does not provide more information on the class variable via that criterion, a leaf or terminal node is obtained. Whereas in DTs for traditional classification each terminal node is labeled with a class value, in ML-DT, each leaf is labeled with a set of labels.

In the building process, the main difference between DTs for traditional classification and ML-DTs resides in the split criterion.

The split criterion used in ML-DT considers the Shannon entropy [28] for each label $l_j \in \mathbf{L}$:

$$H(l_j) = -p_j \log_2 p_j - (1 - p_j) \log_2 (1 - p_j), \tag{2}$$

being p_j the probability that the label l_j is relevant for an instance, which is estimated via relative frequencies:

$$p_j = \sum_{i}^{m} \frac{[[l_j \in \mathbf{L_i}]]}{m}, \tag{3}$$

where $[[l_j \in \mathbf{L_i}]]$ is a function that takes the value 1 if the condition $l_j \in \mathbf{L_i}$ is satisfied and 0 otherwise.

Once the entropy for each label l_j, $1 \leq j \leq q$, is defined, it can be obtained the entropy of the label set $\mathbf{L}$ simply by summing the entropy of all the labels:

$$H(\mathbf{L}) = \sum_{j=1}^{q} H(l_j). \tag{4}$$

Now, it is possible to define the split criterion used in ML-DT. Essentially, for an attribute A whose set of possible values is $\{a_1, \ldots, a_n\}$, in a similar way than in DTs for traditional classification, it consists of the information gain of the label set achieved by dividing the data set along the feature A. It is defined as follows:

$$IG(\mathbf{L}, A) = H(\mathbf{L}) - \sum_{i=1}^{n} P(A = a_i) H(\mathbf{L} \mid A = a_i), \tag{5}$$

where $P(A = a_i)$ is estimated via relative frequencies and $H(\mathbf{L} \mid A = a_i)$ is the entropy of the label set in the partition of the data set in which $A = a_i$, $\forall i = 1, 2, \ldots, n$.

For continuous attributes, it is considered the binary discretization that gives the maximum information gain. More formally, if A is a continuous attribute, let $a_1, a_2, \ldots, a_N$ be the values that it takes for each one of the instances. For each of them, a_j, a discrete attribute A^j is considered, which takes for an instance the value 0 if the value of A for the example is lower than a_j and 1 otherwise, $\forall j = 1, \ldots, N$. Each discretized attribute A^j has an associated conditionated entropy $H(\mathbf{L} \mid A^j) = P(A^j = 0)H(\mathbf{L} \mid A^j = 0) + P(A^j = 1)H(\mathbf{L} \mid A^j = 1)$. Thus, it is considered the value a_j such that the corresponding discrete attribute A^j has the minimum conditional entropy associated $H(\mathbf{L} \mid A^j)$, $1 \leq j \leq N$.

Similarly to DTs for traditional classification, in ML-DT, when an unseen instance $\mathbf{x}$ is required to be classified, it is followed a path from the root node to a leaf using its attribute values. Let N be the total number of instances in that terminal node and n_j the number of instances that have associated the label l_j in the leaf, $\forall j = 1, 2, \ldots, q$. Then, the predicted posterior probability f that the label l_j is relevant for the example is simply obtained via relative frequencies:

$$f(\mathbf{x}, l_j) = \frac{n_j}{N}, \forall j = 1, 2, \ldots, q. \tag{6}$$

The predicted label set for the instance $\mathbf{x}$ consists of those labels for which the predicted posterior probability of being relevant for $\mathbf{x}$ is greater or equal than $\frac{1}{2}$:

$$h(\mathbf{x}) = \left\{ l_j \mid f(\mathbf{x}, l_j) \geq \frac{1}{2}, \quad 1 \leq j \leq q \right\}. \tag{7}$$

2.3. Probability intervals

The probability intervals theory [29] is a particular case of the more general lower and upper probability theory [30]. Let us suppose that we have a variable X whose set of possible values is $\{x_1, \dots, x_K\}$. In the probability intervals theory, the bounds $\{[l(x_i), u(x_i)]\}$, $\forall i = 1, \dots, K$ determine the lower and upper probabilities P_* and P^* of each event. Clearly, $P_*(x_i) = l(x_i)$ and $P^*(x_i) = u(x_i)$, $\forall i = 1, \dots, K$. It must also be verified that:

$$\sum_{i=1}^{K} P_*(x_i) \leq 1,$$

$$\sum_{i=1}^{K} P^*(x_i) \geq 1.$$

Each given set of probability intervals $I = \{[l(x_i), u(x_i)]\}$ gives rise to a credal set, $\mathcal{P}(I)$, defined in the following way:

$$\mathcal{P}(I) = \left\{ p \mid \sum_{i=1}^{K} p(x_i) = 1, p(x_i) \in [l(x_i), u(x_i)], \forall i = 1, \dots, K \right\}. \quad (8)$$

It is possible that, for a set of probability intervals I, there are some values for a particular interval that cannot be part of any probability distribution in $\mathcal{P}(I)$. In these cases the intervals are unnecessarily wide. In order to avoid this point, the concept of *reachable intervals* is used [29]:

Definition 1. *A set of probability intervals $I = \{[l(x_i), u(x_i)]\}$, $i = 1, \dots, K$ is said to be reachable if, $\forall v_i \in \{l(x_i), u(x_i)\}$ there is a probability distribution $p \in \mathcal{P}(I)$ verifying that $p(x_i) = v_i$, $\forall i = 1, \dots, K$*

The following result, proved in [29], can be used in order to check if a set of probability intervals is reachable.

Proposition 1. *A given interval set $I = \{[l(x_i), u(x_i)]\}$, $i = 1, \dots, K$ is reachable if and only if it verifies that, $\forall i = 1, \dots, K$:*

$$\sum_{j=1, j \neq i}^{K} l(x_j) + u(x_i) \leq 1,$$

$$\sum_{j=1, j \neq i}^{K} u(x_j) + l(x_i) \geq 1.$$

If the probability intervals are reachable, the upper and lower probabilities can be extracted from I following this result:

Proposition 2. *With the above notation, if I is a reachable set of probability intervals, the lower and upper probabilities are determined by:*

$$P_*(A) = \max\left\{\sum_{x \in A} l(x), 1 - \sum_{x \notin A} u(x)\right\},$$

$$P^*(A) = \min\left\{\sum_{x \in A} u(x), 1 - \sum_{x \notin A} l(x)\right\}.$$

$\forall A \subseteq X$.

2.4. Non-Parametric Predictive Inference Model for Multinomial data

The Non-Parametric Predictive Inference Model for Multinomial data (NPI-M) is proposed in [17, 31]. It is based on a variation of Hill's assumption $A_{(n)}$ [32, 33], that relates to predictive inference involving n real-valued data observations $Y_i = y_i$, $i = 1, \dots, n$. These observed data correspond to observations associated with a latent variable which creates n intervals in a circular way, represented as $I_j = (y_j, y_{j+1})$, $\forall j = 1, \dots, n-1$ and $I_n = (y_n, y_1)$. According to the circular-$A_{(n)}$ assumption, the next observation will fall into any of these intervals with equal probability, $P(Y_{n+1} \in I_j) = \frac{1}{n}$, $\forall j = 1, \dots, n$.

Let us suppose that there are K distinct categories $c_1, \dots, c_K$ altogether and that k of these categories $c_1, \dots, c_k$ have been already observed (obviously, $k \leq K$). Suppose that, among the n observations, n_j belong to the category c_j, $\forall j = 1, \dots, k$, and that $\sum_{i=1}^{k} n_i = n$. We suppose in this research that we know the value of K.

The concept that underlies NPI-M consists of a latent-variable *probability wheel* representation of the data. On this representation, each one of our observations is represented by a line from the center of the wheel to its boundary. Thus, the wheel is partitioned into n slices with equal size. From the circular-$A_{(n)}$ assumption it is deduced that the probability that the next observation is in any given slice is $\frac{1}{n}$. Then, it must be decided which category of these slices should represent. [17, 31] assumes that each category can only be represented by one single sector of the wheel. Consequently, two or more lines representing the same category must always be positioned next to each other on the wheel. Two cases are distinguished: If two lines that represent the same category border to a slice, that slice must be assigned to this category. However, when a slice is bordered by two lines which represent distinct categories, it might be assigned to one of the two categories associated with the slice's bordering lines, or to any category that has been not observed yet.

If $J \subseteq \{1, \ldots, K\}$, a general event of interest can be represented as follows:

$$E = Y_{n+1} \in \cup_{j \in J} \, c_j. \tag{9}$$

Let OJ be the index-set for the categories belonging to E that have been already observed:

$$OJ = J \cap \{1, \ldots, k\}. \tag{10}$$

Similarly, let us call UJ to the set of indices for the categories in E that have not been observed yet:

$$UJ = J \cap \{k+1, \ldots, K\}. \tag{11}$$

Let us consider $r = |OJ|$ and $l = |UJ|$.

The NPI-M lower probability for the general event E is determined by building a configuration of the probability wheel which provides the minimum number of slices that are assigned to E. With the above notation, the lower probability for E based on the observations is given by [31]:

$$P_*(E) = \frac{n_J - \min(K - r - l, r)}{n}, \tag{12}$$

where $n_J = \sum_{j \in J} n_j$.

Analogously, the upper probability for the event E is obtained by assigning the maximum possible number of slices to E. It is extracted as follows:

$$P^*(E) = \frac{n_J + \min(r + l, k - r)}{n}. \tag{13}$$

For a singleton event $\{Y_{n+1} \in c_i\}$ we have the following lower and upper probabilities:

$$P_*(Y_{n+1} \in c) = \max\left\{0, \frac{n_i - 1}{n}\right\}, P^*(Y_{n+1} \in c) = \min\left\{\frac{n_i + 1}{n}, 1\right\}. \quad (14)$$

Then, it is obtained the following set of probability intervals for singleton events:

$$\mathcal{I} = \left\{[l_i, u_i], l_i = \max\left\{0, \frac{n_i - 1}{n}\right\}, u_i = \min\left\{\frac{n_i + 1}{n}, 1\right\}, \forall i = 1, 2, \ldots, K\right\}. \quad (15)$$

The two following results are proved in [34]:

Proposition 3. *$\mathcal{I}$ is a set of reachable probability intervals.*

Proposition 4. *The set of lower and upper probabilities generated by $\mathcal{I}$ is the same as the produced by the NPI-M lower and upper probabilities produced by (14).*

Therefore, applying the NPI-M to a set of n observations, the lower and upper probabilities of any event can be obtained only by using those of the singleton events. This set of lower and upper probabilities associated with the singleton events gives rise to a reachable set of probability intervals, and, in consequence, to a credal set. However, not all the distributions within this set are compatible with the NPI-M, as it is shown with an example in [34]. In fact, the set of probability distributions that are compatible with the NPI-M is a strict subset of the credal set generated by the lower and upper probabilities of the singleton events.

Considering all the probability distributions of the credal set derived from $\mathcal{I}$ which are compatible with the set of lower and upper probabilities obtained from NPI-M, it is derived an approximate model, called A-NPI-M [34]. This model uses the convex hull of the set of distributions compatible with the NPI-M. Hence, it corresponds to the structure defined by the singleton probabilities. Hence, with the A-NPI-M, the exact model is simplified. Furthermore, with the approximate model, it is avoided to consider a difficult set of constraints. The A-NPI-M represents a standard credal set via a reachable set of probability intervals. For more details, see [34]. When both NPI-M and A-NPI-M are employed in DTs for traditional classification the results are almost identical, as it is shown in [18]. Thus, in this work, it is used the A-NPI-M. However, for the sake of simplicity, we say that we utilize the NPI-M.

3. Multi-Label Credal Decision Tree

The Multi-Label Credal Decision Tree (ML-CDT), proposed in this work, starts from the ML-DT, explained in Section 2.2. There are two main differences between both algorithms: the split criterion and the method used to assign the posterior probabilities that the labels are relevant for new instances.

Let us suppose that N is the total number of instances in a certain node and that n_j, (respectively, n'_j) is the number of instances in that node for which the label l_j is relevant, (respectively, irrelevant) $\forall j = 1, 2, \ldots, q$.

Let us consider, for each label l_j, $1 \leq j \leq q$, the probability intervals associated with the NPI-M corresponding to the events that the label l_j is relevant and irrelevant for an instance:

$$I_{l_j} = \left[\max\left\{0, \frac{n_j - 1}{N}\right\}, \min\left\{\frac{n_j + 1}{N}, 1\right\}\right], \tag{16}$$

$$I'_{l_j} = \left[\max\left\{0, \frac{n'_j - 1}{N}\right\}, \min\left\{\frac{n'_j + 1}{N}, 1\right\}\right]. \tag{17}$$

For each label l_j, $1 \leq j \leq q$, these probability intervals, as shown in Section 2.3, give rise to this credal set:

$$\mathcal{P}(l_j) = \left\{p \mid p(l_j) \in I_{l_j}, (1 - p(l_j)) \in I'_{l_j}\right\}. \tag{18}$$

The split criterion used in ML-CDT considers the maximum of entropy in the credal set $\mathcal{P}(l_j)$, $\forall j = 1, 2, \ldots, q$.

$$H^*(\mathcal{P}(l_j)) = \max_{p \in \mathcal{P}(l_j)} H(p). \tag{19}$$

The algorithm to calculate the distribution that provides the maximum value of entropy $H^*(l_j)$, $\forall j = 1, \ldots, q$ is given in the Figure 1. It is based on the algorithm for A-NPI-M proposed in [34].[1].

H^* is a well-established measure that verifies good properties [35].

[1] In [34] it is shown that the set of probabilities associated with probability intervals resulting from the NPI-M and the credal set corresponding to the A-NPI-M give rise to quite similar values of the maximum of entropy. For this reason and the sake of simplicity, in this research we use the algorithm associated with the A-NPI-M

```
Procedure **Distribution of maximum of entropy in** $\mathcal{P}(\mathbf{l_j})$(training set $\mathcal{D}$ of $N$ instances, $n_j$ number of instances in $\mathcal{D}$ that has associated the label $l_j$, $n'_j$ number of instances for which $l_j$ is irrelevant)

1. If $\left|n_j - n'_j\right| \leq 2$ then
        1.1 $\hat{p}(l_j) = \frac{1}{2}$
        1.2 $\hat{p'}(l_j) = \frac{1}{2}$
2. Else If $n_j < n'_j$ then
        2.1 $\hat{p}(l_j) = \frac{n_j+1}{N}$
        2.2 $\hat{p'}(l_j) = \frac{n'_j-1}{N}$
3. Else
        3.1 $\hat{p}(l_j) = \frac{n_j-1}{N}$
        3.2 $\hat{p'}(l_j) = \frac{n'_j+1}{N}$
4. Return $\hat{p}$.
```

Figure 1: Pseudo-code of the calculation of the distribution that produces the maximum of entropy for l_j

The ML-CDT takes into account the sum of the maximum of entropies in the corresponding credal sets overall labels l_j, $1 \leq j \leq q$:

$$H^*(\mathbf{L}) = \sum_{j=1}^{q} H^*(\mathcal{P}(l_j)). \tag{20}$$

The split criterion used in ML-CDT is similar to the one used in ML-DT. However, the first one is based on the maximum of entropy in the corresponding credal set for each label. Therefore, if A is an attribute and $\{a_1, \ldots, a_n\}$ are its possible values, the split criterion for ML-CDT is called Imprecise Information Gain (IIG), and it is defined by the following formula:

$$IIG(\mathbf{L}, A) = H^*(\mathbf{L}) - \sum_{i=1}^{n} P(A = a_i) H^*(\mathbf{L} \mid A = a_i), \tag{21}$$

where $H^*(\mathbf{L} \mid A = a_i)$ is the maximum of entropy $H^*(\mathbf{L})$ (on the corresponding credal set) in the partition of the dataset composed by those instances that verify that $A = a_i$, $\forall i = 1, \ldots, n$.

For continuous attributes, ML-CDT makes a discretization similar to the one made in ML-DT: it is considered the binary discretization that gives the maximum value of IIG. More formally, if A is a continuous attribute, let $a_1, a_2, \ldots, a_N$ be the values that it takes for each one of the instances. For each of them, a_j, a discrete attribute A^j is considered, which takes, for an instance, the value 0 if the value of A for the example is lower than a_j and 1 otherwise, $\forall j = 1, \ldots, N$. Each discretized attribute A^j has an associated conditionated maximum of entropy over the corresponding credal set $H^*(\mathbf{L} \mid A^j) = P(A^j = 0)H^*(\mathbf{L} \mid A^j = 0) + P(A^j = 1)H^*(\mathbf{L} \mid A^j = 1)$. Thus, it is considered the value a_j such that the corresponding discrete attribute A^j has the minimum value of $H^*(\mathbf{L} \mid A^j)$, $1 \leq j \leq N$.

If it is wanted to classify an unseen instance $\mathbf{x}$, a path from the root to a leaf node is followed. When the terminal node is reached, following the above notation for the total number of examples and the total number of instances for which each label l_j is relevant or irrelevant, $1 \leq j \leq q$, the predicted posterior probability that l_j is relevant for $\mathbf{x}$ is the one that provides the maximum of entropy in the corresponding credal set $\mathcal{P}(l_j)$, i.e:

$$f(\mathbf{x}, l_j) = \arg \max_{p \in \mathcal{P}(l_j)} H(p). \tag{22}$$

As in ML-DT, the labels predicted as relevant are those for which the predicted posterior probability is greater or equal than $\frac{1}{2}$:

$$h(\mathbf{x}) = \left\{ l_j \mid f(\mathbf{x}, l_j) \geq \frac{1}{2}, \quad 1 \leq j \leq q \right\}. \tag{23}$$

3.1. Differences between ML-DT and ML-CDT

3.1.1. Size of the training set

It is easy to observe that if the value of N is higher, i.e, if the training set is larger, then the intervals $\left[\max\left(\frac{n_j-1}{N}, 0\right), \min\left(\frac{n_j+1}{N}, 1\right)\right]$ and $\left[\max\left(\frac{n'_j-1}{N}, 0\right), \min\left(\frac{n'_j+1}{N}, 1\right)\right]$ are narrower. In consequence, the corresponding credal set has fewer probability distributions that are really different from the one obtained by relative frequencies. Hence, in upper levels of the tree, when N is often pretty large, IG and IIG, defined respectively, in (5) and (21), provide similar values and, consequently, ML-DT and ML-CDT have similar behavior. However, in lower levels of the tree, when the size of the training set is usually small, the intervals are narrower. This, the associated credal set might contain many probability distributions which are probably very distinct from the one utilized to calculate the Shannon Entropy H. In this way, in these cases, IG and IIG might give no similar values since H and H^* might be quite different. Therefore, ML-DT and ML-CDT behave similarly at upper levels of the tree but they can be very distinct at lower levels.

3.1.2. Stop criterion

For a variable A whose possible values are $\{a_1, \ldots, a_n\}$, the value of $IIG(\mathbf{L}, A)$ can be negative, unlike $IG(\mathbf{L}, A)$. The reason is that, according to [34], the imprecise information gain for a label $l \in \mathbf{L}$, $H^*(l) - \sum_{i=1}^{n} P(A = a_i)H^*(l \mid A = a_i)$ can be negative, unlike the information gain $H(l) - \sum_{i=1}^{n} P(A = a_i)H(l \mid A = a_i)$. Hence, ML-CDT avoids to select attributes that worsen the information of the label set. Therefore, overfitting in ML-CDT should be lower than in ML-DT because the branching of the tree often stops before in ML-CDT than in ML-DT.

3.1.3. Predicitions at leaf nodes

As it can be easily observed, for both ML-DT and ML-CDT, in a leaf node, a label l_j is predicted as relevant for an instance if and only if $n_j \geq n'_j$. Therefore, the decision rule at leaf nodes is the same for both algorithms. Regarding the posterior probabilities for the labels, the ones predicted by ML-DT are the ones estimated via relative frequencies, whereas the posterior probabilities predicted by ML-CDT are the ones that give rise to the maximum of entropy on the corresponding credal sets. Thus, as we show below, the predicted posterior probabilities by ML-CDT are less sensitive to noise than the ones predicted by ML-DT.

3.1.4. Data with noise

It can be shown that, for a label $l \in \mathbf{L}$, the Shannon entropy H is more sensitive to noise than the maximum of entropy H^* over the corresponding credal set. We illustrate this issue with a simple example below.

Let us suppose that we have a dataset $\mathcal{D}$ of size N. Let us denotate n_1 as the number of instances that have associated the label l and n'_1 the number of instances for which l is irrelevant. Let us suppose that $n'_1 > n_1 + 3$ and that $n_1 > 2$. Let $\mathcal{D}_n$ be a noisy dataset derived from $\mathcal{D}$ by changing the value of the label l for one instance for which l is irrelevant (in the noisy dataset the instance has associated the label). With these assumptions, we have the following result.

Proposition 5. $H^{\mathcal{D}_n}(l) - H^{\mathcal{D}}(l) \geq H^*(\mathcal{P}^{\mathcal{D}_n}(l)) - H^*(\mathcal{P}^{\mathcal{D}}(l))$

Proof:. We have that:

$$H^{\mathcal{D}}(l) = -\frac{n'_1}{N}\log_2\frac{n'_1}{N} - -\frac{n_1}{N}\log_2\frac{n_1}{N}$$

$$H^{\mathcal{D}_n}(l) = -\frac{n'_1 - 1}{N}\log_2\frac{n'_1 - 1}{N} - \frac{n_1 + 1}{N}\log_2\frac{n_1 + 1}{N}$$

Hence:

$$H^{\mathcal{D}_n}(l) - H^{\mathcal{D}}(l) = -\frac{n'_1 - 1}{N}\times\log_2(n'_1 - 1) - \frac{n_1 + 1}{N}\times\log_2(n_1 + 1) + \frac{n'_1}{N}\log_2(n'_1)$$

$$+\frac{n_1}{N}\log_2(n_1) + \frac{\log_2(N)}{N}\times[n'_1 - 1 + n_1 + 1 - n'_1 - n_1] =$$
$$\frac{-(n'_1 - 1)\log_2(n'_1 - 1) - (n_1 + 1)\log_2(n_1 + 1) + n'_1\log_2(n'_1) + n_1\log_2(n_1)}{N}$$

Now, it is easy to observe that the distribution that produces the maximum of entropy in $\mathcal{P}^{\mathcal{D}}(l)$ is $\hat{p}(l) = \frac{n_1+1}{N}$ and $\hat{p'}(l) = \frac{n'_1-1}{N}$. It is also easily checkable that the distribution that gives rise to the maximum of entropy in $\mathcal{P}^{\mathcal{D}_n}(l)$ is $\hat{p}_n(l) = \frac{n_1+2}{N}$ and $\hat{p'}_n(l) = \frac{n'_1-2}{N}$.

Thus:

$$H^*(\mathcal{P}^{\mathcal{D}_n}(l)) - H^*(\mathcal{P}^{\mathcal{D}}(l)) = \frac{-(n'_1 - 2)\log_2(n'_1 - 2) - (n'_1 + 2)\log_2(n_1 + 2)}{N}$$
$$+\frac{(n'_1 - 1)\log_2(n'_1 - 1) + (n_1 + 1)\log_2(n_1 + 1)}{N}$$

Therefore,

$$H^{\mathcal{D}_n}(l) - H^{\mathcal{D}}(l) \geq H^*(\mathcal{P}^{\mathcal{D}_n}(l)) - H^*(\mathcal{P}^{\mathcal{D}}(l)) \Leftrightarrow$$

$$-(n'_1 - 1)\log_2(n'_1 - 1) - (n_1 + 1)\log_2(n_1 + 1) + n'_1\log_2(n'_1) + n_1\log_2(n_1) \geq$$

$$-(n_1'-2)\log_2(n_1'-2)-(n_1'+2)\log_2(n_1+2)$$
$$+(n_1'-1)\log_2(n_1'-1)+(n_1+1)\log_2(n_1+1)\Leftrightarrow$$
$$n_1'\log_2(n_1')+n_1\log_2(n_1)+(n_1'-2)\log_2(n_1'-2)+(n_1+2)\log_2(n_1+2)\geq$$

$$2(n_1'-1)\log_2(n_1'-1)+2(n_1+1)\log_2(n_1+1)$$

Since the logarithm function is convex:

$$n_1'\log_2(n_1')+(n_1'-2)\log_2(n_1'-2)\geq 2(n_1'-1)\log_2(n_1'-1),$$

$$(n_1+2)\log_2(n_1+2)+n_1\log_2(n_1)\geq 2(n_1+1)\log_2(n_1+1);$$

it is quite easy to check that our hypothesis holds.

□

Remark that the main difference between the split criteria of ML-CDT and ML-DT is that the first one uses H^*, whereas the second one employs H. Hence, from the previous proposition it is followed that, when a label is modified for an instance of the dataset $\mathcal{D}$, the split criterion used in ML-CDT is less sensitive to the change than the one used in ML-DT. We show below an example of this point, which is very based on one given in [14]. In this example, the superscript $\mathcal{D}$ (respectively, $\mathcal{D}_n$) indicates that the corresponding measure is calculated over the dataset $\mathcal{D}$ (respectively, $\mathcal{D}_n$).

Example 1. *Let $\mathcal{D}$ be a dataset of size $N = 15$. Let us suppose that for 5 instances a certain label l is irrelevant and the other 10 instances has associated the label l. Let us also suppose that we have two binary attributes A_1 and A_2. For each one of the possible values of the attributes, the instances are arranged as follows:*

$A_1 = 0 \rightarrow (n_1 = 4, n_1' = 5)$
$A_1 = 1 \rightarrow (n_1 = 6, n_1' = 0)$
$A_2 = 0 \rightarrow (n_1 = 1, n_1' = 5)$
$A_2 = 1 \rightarrow (n_1 = 9, n_1' = 0)$

We have that:

$$H^{\mathcal{D}}(l) = -\frac{10}{15}\log_2(\frac{10}{15}) - \frac{5}{15}\log_2(\frac{5}{15}) = 0.918$$
$$H^{\mathcal{D}}(l \mid A_1 = 0) = -\frac{4}{9}\log_2(\frac{4}{9}) - \frac{5}{9}\log_2(\frac{5}{9}) = 0.991$$

$$H^{\mathcal{D}}(l \mid A_1 = 1) = 0$$

Hence, the information gain of A_1 associated with l is:

$$IG^{\mathcal{D}}(l, A_1) = H^{\mathcal{D}}(l) - P(A_1 = 0)H^{\mathcal{D}}(l \mid A_1 = 0) - P(A_1 = 1)H^{\mathcal{D}}(l \mid A_1 = 1) =$$

$$0.918 - 0.991 \times 0.6 = 0.3237$$

Regarding A_2:

$$H^{\mathcal{D}}(l \mid A_2 = 0) = -\frac{1}{6}\log_2(\frac{1}{6}) - \frac{5}{6}\log_2(\frac{5}{6}) = 0.65$$

$$H^{\mathcal{D}}(l \mid A_2 = 1) = 0$$

Information gain of A_2 corresponding to l:

$$IG^{\mathcal{D}}(l, A_2) = H^{\mathcal{D}}(l) - P(A_2 = 0)H^{\mathcal{D}}(l \mid A_2 = 0) - P(A_2 = 1)H^{\mathcal{D}}(l \mid A_2 = 1) =$$

$$0.918 - 0.65 \times 0.4 = 0.6583$$

Imprecise information gain of A_1 associated with l:

$$H^*(\mathcal{P}^{\mathcal{D}}(l)) = 0.971$$

$$H^*(\mathcal{P}^{\mathcal{D}}(l \mid A_1 = 0)) = 1$$

$$H^*(\mathcal{P}^{\mathcal{D}}(l \mid A_1 = 1)) = 0.65$$

$$IIG^{\mathcal{D}}(l, A_1) = H^*(\mathcal{P}^{\mathcal{D}}(l)) - P(A_1 = 0)H^*(\mathcal{P}^{\mathcal{D}}(l \mid A_1 = 0)) -$$

$$P(A_1 = 1)H^*(\mathcal{P}^{\mathcal{D}}(l \mid A_1 = 1)) = 0.971 - 0.6 - 0.26 \times 0.4 = 0.111$$

Imprecise information gain of A_2 associated with l:

$$H^*(\mathcal{P}^{\mathcal{D}}(l \mid A_2 = 0)) = 0.65$$

$$H^*(\mathcal{P}^{\mathcal{D}}(l \mid A_2 = 1)) = 0.5033$$

$$IIG^{\mathcal{D}}(l, A_2) = H^*(\mathcal{P}^{\mathcal{D}}(l)) - P(A_2 = 0)H^*(\mathcal{P}^{\mathcal{D}}(l \mid A_2 = 0)) -$$

$$P(A_2 = 1)H^*(\mathcal{P}^{\mathcal{D}}(l \mid A_2 = 1)) = 0.971 - 0.4 \times 0.65 - 0.5033 \times 0.6 = 0.409$$

As $IG^{\mathcal{D}}(l, A_1) < IG^{\mathcal{D}}(l, A_2)$ and $IIG^{\mathcal{D}}(l, A_1) < IIG^{\mathcal{D}}(l, A_2)$, if we only had the label l, the attribute A_2 would be selected in order to branch the tree in ML-DT, as well as in ML-CDT.

Let us suppose that noise is introduced in the dataset by changing the value of l for an instance that verifies that $A_1 = 0$ and $A_2 = 1$. The label in this noisy dataset is irrelevant for the instance, whereas in the clean one the instance had associated the label l. In this noisy dataset $\mathcal{D}_n$, the instances are arranged as follows:

$A_1 = 0 \rightarrow (n_1 = 3, n_1' = 6)$
$A_1 = 1 \rightarrow (n_1 = 6, n_1' = 0)$
$A_2 = 0 \rightarrow (n_1 = 1, n_1' = 5)$
$A_2 = 1 \rightarrow (n_1 = 8, n_1' = 1)$

Values of IG and IIG in this noisy dataset:

$$H^{\mathcal{D}_n}(l) = -\frac{9}{15}\log_2(\frac{9}{15}) - \frac{6}{15}\log_2(\frac{6}{15}) = 0.971$$

$$H^{\mathcal{D}_n}(l \mid A_1 = 0) = -\frac{3}{9}\log_2(\frac{3}{9}) - \frac{6}{9}\log_2(\frac{6}{9}) = 0.9183$$

$$H^{\mathcal{D}_n}(l \mid A_1 = 1) = 0$$

$$IG^{\mathcal{D}_n}(l, A_1) = H^{\mathcal{D}_n}(l) - P(A_1 = 0)H^{\mathcal{D}_n}(l \mid A_1 = 0) - P(A_1 = 1)H^{\mathcal{D}_n}(l \mid A_1 = 1) =$$

$$0.971 - 0.9183 \times 0.6 = 0.42$$

$$H^{\mathcal{D}_n}(l \mid A_2 = 0) = -\frac{1}{6}\log_2(\frac{1}{6}) - \frac{5}{6}\log_2(\frac{5}{6}) = 0.65$$

$$H^{\mathcal{D}_n}(l \mid A_2 = 1) = -\frac{8}{9}\log_2(\frac{8}{9}) - \frac{1}{9}\log_2(\frac{1}{9}) = 0.5033$$

$$IG^{\mathcal{D}_n}(l, A_2) = H^{\mathcal{D}_n}(l) - P(A_2 = 0)H^{\mathcal{D}_n}(l \mid A_2 = 0) - P(A_2 = 1)H^{\mathcal{D}_n}(l \mid A_2 = 1) =$$

$$0.971 - 0.65 * 0.4 - 0.5033 \times 0.6 = 0.409$$

$$H^*(\mathcal{P}^{\mathcal{D}_n}(l)) = 0.9968$$

$$H^*(\mathcal{P}^{\mathcal{D}_n}(l \mid A_1 = 0)) = 0.9911$$
$$H^*(\mathcal{P}^{\mathcal{D}_n}(l \mid A_1 = 1)) = 0.65$$

$$IIG^{\mathcal{D}_n}(l, A_1) = H^*(\mathcal{P}^{\mathcal{D}_n}(l)) - P(A_1 = 0)H^*(\mathcal{P}^{\mathcal{D}_n}(l \mid A_1 = 0)) -$$

$$P(A_1 = 1)H^*(\mathcal{P}^{\mathcal{D}_n}(l \mid A_1 = 1)) = 0.9968 - 0.6 \times 0.9911 - 0.65 \times 0.4 = 0.1421$$

$$H^*(\mathcal{P}^{\mathcal{D}_n}(l \mid A_2 = 0)) = 0.65$$
$$H^*(\mathcal{P}^{\mathcal{D}_n}(l \mid A_2 = 1)) = 0.5033$$

$$IIG^{\mathcal{D}_n}(l, A_2) = H^*(\mathcal{P}^{\mathcal{D}_n}(l)) - P(A_2 = 0)H^*(\mathcal{P}^{\mathcal{D}_n}(l \mid A_2 = 0)) -$$

$$P(A_2 = 1)H^*(\mathcal{P}^{\mathcal{D}_n}(l \mid A_2 = 1)) = 0.9968 - 0.4 \times 0.65 - 0.5033 \times 0.6 = 0.4055$$

For this noisy dataset $IG^{\mathcal{D}_n}(l, A_2) < IG^{\mathcal{D}_n}(l, A_1)$ *and* $IG^{\mathcal{D}_n}(l, A_1 < IG^{\mathcal{D}_n}(l, A_2)$. *Hence, for ML-DT, the attribute* A_1 *is now selected for splitting the dataset (supposing that* l *is the only label), unlike with the clean dataset. Nevertheless, for ML-CDT, the selected splitting attribute is the same that the one choosen with the clean dataset:* A_2

In this way, in the previous example, IIG is not affected by the noise in the label l, unlike IG. It illustrates the fact that the split criterion used in ML-CDT is less sensitive to noise than the one used in ML-DT. Therefore, it can be deduced than ML-CDT should be more robust to noise than ML-DT since the main difference between both algorithms resides in the split criterion. In this work, this issue is checked with an extensive experimental analysis.

3.2. Noise in MLC

In the real world, as it is known, datasets can contain intrinsic noise, i.e, they can contain noise despite not being manipulated. As we have said previously, datasets in MLC might have more intrinsic noise than datasets in standard classification. In this Section we argue it with more detail.

Let us suppose that, in traditional classification, p is the probability that one instance has a wrong class value. Let us suppose that, in MLC, one instance also has probability p of having associated incorrectly a label o vice-versa is also p.

In standard classification, one instance has the right value of the class variable with probability:

$$1 - p$$

Nevertheless, in MLC, the probability that one example has no mistake in any of the labels is:

$$(1 - p)^q$$

In the prior situation, let us consider the values of q and p are, respectively, $q = 10$ and $p = 0.05$. Then, the probability that one instance has the correct value of the class variable in traditional classification is:

$$1 - 0.05 = 0.95$$

On the other hand, the probability that the example has the right value for all the labels in MLC is:

$$(0.95)^{10} = 0.6$$

Hence, the probability that one example has an erroneous value for at least one label is 0.4. Consequently, although one instance has an incorrect value in one label with a quite small probability, it might be pretty notable that the example has an error in any label, as occurs in this scenario. Therefore, it is easily observable that in MLC the intrinsic noise can be notably higher than in standard classification. For this reason, it is suitable to use a classifier that is robust to noise in MLC.

Summarizing, the intrinsic noise in MLC is probably higher than in standard classification and ML-CDT, which is based on the NPI-M, should be less sensitive to noise than ML-DT, as argued in Section 3.1. For this reason, it is appropriate to apply the NPI-M to the adaptations of DTs to MLC.

4. Experimentation

4.1. Experimental setup

4.1.1. Datasets:

In this experimental analysis, we have utilized 12 MLC datasets. Nine of them were employed in an extensive experimentation carried out in [36], in which multiple MLC algorithms were compared. We have not used the other three datasets used in that research due to their excessive computational cost. Instead, another four datasets have been utilized, which can be downloaded from the web site of mulan `http://mulan.sourceforge.net/datasets.html`.

Table 1 shows the principal characteristics of each database. Specifically, it allows us to see, for each dataset, its number of instances, its number of attributes (continuous and discrete), its number of labels, its label cardinality, its label density, defined as the label cardinality divided by the number of labels, i.e, the proportion of labels which are relevant for an instance (on average) and its domain.

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

Table 1: Datasets employed in the experimental research. N is the size of the dataset, N_DA and N_CA are, respectively, is 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

As can be observed, the datasets are diverse in terms of label cardinality, number of discrete and continuous features, number of examples and labels.

Essentially, our datasets come from three domains:

- **Text categorization:** The datasets that cover this domain are *bibtex* [37], which contains information about meta data of bibtex items, *enron* [38], that contains data about emails of Enron seniors, and *medical* [39], a dataset whose instances correspond to documents with a summary of a patient symptom history.

- **Biology:** Two datasets are associated with this domain. The first of them is *genbase* [40], that contains data about proteins. The second dataset corresponding to biology is *yeast* [27], which covers information about functions of genes.

- **Multimedia:** Within this domain, *birds* [41] covers data about birds audios. *Cal500* [42] contains information about pieces of music. *Corel5k* [43] is a dataset whose examples correspond to Corel images which have been segmented by normalised cuts. *Emotions* [44] covers data about pieces of music labeled with emotions. *Flags* contains information about flags of the countries. *Mediamill* [45] covers data about concepts which appear in videos. Finally, *scene* [24] contains information about scenes, which can be annotated in the following six contexts:mountain, urban, beach, sunset, field and fall-folliage.

Thus, our datasets are also varied in terms of thematic. Actually, they cover the main domains of MLC. Therefore, it can be said that the datasets used in this experimentation are representative.

4.1.2. Procedure:

In this experimentation two algorithms have been used: ML-DT, which uses a standard split criterion based on precise probabilities, and ML-CDT, the adaptation proposed in this work of DTs to MLC that utilizes using imprecise probabilities and uncertainty measures on sets of probabilities resulting from the NPI-M in the split criterion.[2] We have implemented both algorithms in the mulan library [46]. For this purpose, we have used some of the structures provided in Weka [47].

In order to measure the performance of both algorithms 17 metrics have been used. Among them, *Subset Accuracy*, *Hamming Loss*, *Accuracy*, *Precision*, *Recall* and *F1* correspond to the category of example-based classification measures. Six of the metrics are based on label classification: *Micro Precision*, *Macro Precision*, *Micro Recall*, *Macro Recall*, *Micro F1* and *Macro F1*. With respect to ranking labels in instances, we have used 5 measures:

[2] It has not been considered in this research the ML-CDT that uses uncertainty measures on the credal sets resulting from the IDM because this model has a strong dependence of a parameter. In fact, with its standard value, it obtains poor results when it is used for MLC in the ML-CDT procedure.

Ranking Loss, *Coverage*, *Average Precision*, *One Error* and *Binary Cross-Entropy.* All of these measures were employed in an extensive experimentation with MLC methods carried out in [36], except for Binary Cross Entropy. However, this last metric is very appropriate to measure the quality of the estimated probabilities for binary variables, and it has been very utilized in the literature in these cases. In Appendix A it can be found a detailed explanation of the evaluation metrics used in this research.

In this experimental analysis, we have considered three noise levels for each label: 0%, 5%, and 10%. We have done the following procedure of CV of 5 folds for each dataset and each noise level: We have divided each dataset into 5 partitions and, for each one of them, we have carried out an iteration in which we have considered the corresponding partition for testing and the rest of the data for training. For each label, x% of the instances in the training set have been selected and the value of their label has been modified. Part of the functionality provided in mulan has been used to create the partitions. Both algorithms have used the same partitions for each dataset. In order to generate noise, we have used the Weka filters (specifically, *AddNoise*), with the parameters that are given by default in this software (except for the corresponding to the noise level, as it is obvious). The model has been learned through this noisy dataset and each one of the evaluation metrics is extracted by utilizing the test set.

4.1.3. Statistical Evaluation:

In our experimental study, for each metric and each noise level, there are two algorithms to compare: ML-DT and ML-CDT. In this way, following the indications of [48, 49], the Wilcoxon test [50] has been used in order to determine which classifier performs better than the other one and if the differences are statistically significant or not, with a level of significance of $\alpha = 0.05$. For the Wilcoxon test, we have used the R software [51].

4.2. Results and discussion

Tables 2, 3 and 4 show, respectively, a summary of the obtained results by each classifier for each metric with 0%, 5% and 10% of added noise. Specifically, for each metric, they allow us to see, according to the Wilcoxon test, which classifier is better than the other one and if the differences are significative or not. It is also shown the number of wins for each metric, which is defined as the number of datasets in which the corresponding algorithm performs better than the other one. In Appendix B, the complete results of this experimental study can be found.

Metric	ML-DT	ML-CDT	Wins ML-DT	Wins ML-CDT
Hamming Loss		(•)	0	12
Subset Accuracy		(-)	2	9
Accuracy	(-)		5	7
Precision		(-)	2	10
Recall	(•)		11	1
F1	(-)		5	7
Micro Precision		(•)	0	12
Macro Precision		(-)	5	7
Micro Recall	(•)		11	1
Macro Recall	(•)		10	2
Micro F1	(-)		5	7
Macro F1		(-)	4	8
Coverage		(•)	1	11
Ranking Loss		(•)	1	11
Average Precision		(-)	3	9
One Error		(-)	3	9
Binary Cross-Entropy		(•)	0	12

Table 2: Summary of the results obtained by ML-DT and ML-CDT for each metric when there is no noise introduced in the data. (•) indicates that the algorithm of the column performs significantly better than the other one. (-) means that the classifier of the column performs better than the other one but the results are statistically equivalent.

4.2.1. Example-based classification metrics:

As can be observed, in Hamming Loss the performance obtained by ML-CDT is always significantly better than the one obtained by ML-DT. It implies that, on average, the difference between the real and the predicted label sets for the examples are smaller for ML-CDT than for ML-DT for all the noise levels. The reason for this fact is that in MLC the intrinsic noise is probably higher than in traditional classification and, as we have argued in Section 3.1, ML-CDT performs better than ML-DT when the data contains noise. On the other hand, ML-DT always predicts more relevant labels correctly than ML-DT, due to the better results obtained in Recall. This is because ML-CDT stops branching the tree before than ML-DT and

Metric	ML-DT	ML-CDT	Wins ML-DT	Wins ML-CDT
Hamming Loss		(•)	0	12
Subset Accuracy		(•)	2	9
Accuracy		(-)	3	9
Precision		(•)	1	11
Recall	(•)		10	2
F1		(-)	3	9
Micro Precision		(•)	1	10
Macro Precision		(-)	2	10
Micro Recall	(•)		12	0
Macro Recall	(•)		12	0
Micro F1		(-)	3	9
Macro F1	(-)		7	5
Coverage		(•)	2	10
Ranking Loss		(•)	2	10
Average Precision		(-)	3	9
One Error		(-)	3	9
Binary Cross-Entropy		(•)	0	12

Table 3: Summary of the results obtained by ML-DT and ML-CDT for each metric when 5% of noise is introduced in the data. (•) indicates that the algorithm of the column performs significantly better than the other one. (-) means that the classifier of the column performs better than the other one but the results are statistically equivalent.

Metric	ML-DT	ML-CDT	Wins ML-DT	Wins ML-CDT
Hamming Loss		(●)	0	12
Subset Accuracy		(●)	0	11
Example-Based Accuracy		(●)	2	10
Example-Based Precision		(●)	1	11
Recall	(●)		11	1
F1		(●)	2	10
Micro Precision		(●)	1	11
Macro Precision		(●)	1	11
Micro Recall	(●)		11	1
Macro Recall	(●)		11	1
Micro F1		(●)	2	10
Macro F1	(-)		6	6
Coverage		(●)	2	10
Ranking Loss		(●)	2	10
Average Precision		(-)	4	8
One Error		(-)	4	8
Binary Cross-Entropy		(●)	0	12

Table 4: Summary of the results obtained by ML-DT and ML-CDT for each metric when 10% of noise is introduced in the data. (●) indicates that the algorithm of the column performs significantly better than the other one. (-) means that the classifier of the column performs better than the other one but the results are statistically equivalent.

the proportion of instances that have associated a certain label in MLC tends to be very low. Consequently, in a large number of cases, ML-CDT does not reach parts of the tree for which a label is correctly predicted as relevant. Nevertheless, in these parts of the tree the noise influences negatively.

When there is no noise introduced in the data, the results in the rest of the example-based classification metrics are statistically equivalent according to the Wilcoxon test. However, the number of wins of ML-CDT in Precision and Subset Accuracy is considerably higher than the wins of ML-DT in both of the previous metrics. Hence, ML-CDT predicts less irrelevant labels as relevant (Precision) than ML-DT and the first algorithm predicts correctly the entire relevant label set for more instances than the second one (Subset Accuracy). The reason why ML-CDT performs better than ML-DT in Precision is that, as we say in the previous paragraph, ML-CDT stops branching the tree before than ML-DT, avoiding reaching parts of the tree where the noise has a negative influence. Likewise, ML-CDT obtains better results than ML-DT in Subset Accuracy because ML-CDT outperforms ML-DT with noisy data and in MLC there might be intrinsic noise in the data.

With 5% of noise, the differences in Precision and Subset Accuracy are significative, favorable to ML-CDT. Besides, as without noise in the data, ML-CDT achieves significantly better performance in Hamming Loss than ML-DT, whereas ML-DT obtains significantly better results in Recall. The results obtained in Accuracy, which measures how the algorithm predicts the labels for the examples in general, and in F1, which consists of the harmonic mean between Precision and Recall, are statistically equivalent. Nevertheless, in both of the previous metrics, the number of wins of ML-CDT is notably higher than the number of wins of ML-DT (9 versus 3 in both metrics).

When the level of noise introduced in the data is 10%, ML-CDT obtains significantly better performance than ML-DT for all example-based classification metrics, except for Recall. In this last measure, the results are significantly favorable to ML-DT again.

In summary, the higher is the level of noise in the data, the higher is the improvement of ML-CDT over ML-DT in example-based classification metrics, except in Recall, where the results are always better for ML-DT. This is because ML-CDT is more robust to noise than ML-DT, although the first algorithm stops branching the tree before than the second one, avoiding to reach parts of the tree where some labels are predicted as relevant for the instances. This fact is more emphasized as there is more noise introduced in the data.

4.2.2. Label-based classification metrics:

Firstly, for all the noise levels, the results corresponding to Precision averaged over all pairs label-example (Micro Precision) are significantly better in ML-CDT than in ML-DT. On the other hand, ML-DT always achieves significantly better results than ML-CDT in the Recall averaged over all labels (Macro Recall) and in the Recall averaged over all the pairs label-instance (Micro Recall). These points are because, as we comment above, ML-CDT stops branching the tree before than ML-DT and, in consequence, reaches fewer parts of the tree where some labels for the instances are predicted as relevant and the noise influences negatively.

When there is no noise introduced in the data the results obtained in the rest of label-based classification metrics are statistically equivalent according to the Wilcoxon test.

The results are similar when there is a 5% of noise introduced in the data. However, remark that in Macro Precision, which is the Precision averaged on all labels, in 10 datasets the result is better for ML-CDT, whereas ML-DT obtains better performance according to this metric in only 2 datasets.

With 10% of noise, in Macro Precision, the results obtained by ML-CDT are significantly better than the ones obtained by ML-DT. The performance in terms of the harmonic mean between Micro Precision and Micro Recall (Micro F1) is significantly better with ML-CDT than with ML-DT. The results in Macro F1, the harmonic mean between Precision and Recall averaged over all labels, are statistically equivalent, as happens with 0 and 5% of noise.

The fact that the harmonic mean between Micro Precision and Micro Recall (Micro F1) is more favorable to ML-CDT as there is more noise introduced in the data is principally due to ML-CDT is more robust to noise than ML-DT, as we have argued in Section 3.1.

4.2.3. Example-based ranking metrics:

It is easy to check that, for all noise levels, the results obtained by ML-CDT in example-based ranking metrics are better than the ones obtained by ML-DT. Consequently, in general, ML-CDT predicts a higher posterior probability for relevant labels and lower for irrelevant ones than ML-DT.

More specifically, ML-CDT achieves significantly better performance in ordering fewer pairs of relevant and irrelevant labels reversely (Ranking Loss). Furthermore, average the number of steps that are required to go down to cover all the relevant labels for an instance is considerably lower with ML-CDT than with ML-DT, due to the significantly better results obtained in Coverage. The results obtained in Binary Cross-Entropy by ML-CDT are significantly better than the ones obtained by ML-DT. It implies that the predicted posterior probabilities are closer to one for the relevant labels and nearer to cero for labels that are not associated with the instances with ML-CDT than with ML-DT. In the other two example-based ranking metrics the results are statistically equivalent according to the Wilcoxon test. Nevertheless, in both measures, the number of wins of ML-CDT is notably higher than the number of wins of ML-DT.

The reasons for the previous points are that, as we have shown in Section 3.1, the building process of ML-CDT is less sensitive to noise than the one used in ML-DT. Furthermore, in the terminal nodes, the posterior probabilities estimated by ML-DT are the ones corresponding to relative frequencies, whereas the ones estimated by ML-CDT are the ones that give rise to the maximum of entropy in the corresponding credal sets. Therefore, the posterior probabilities predicted by ML-CDT are more robust to noise than the one predicted by ML-DT. Besides, it is convenient to remark that in leaf nodes there are uaually few instances and the difference between both algoritms is notable.

4.2.4. Summary:

Firstly, as we have commented before, the predictions about the relevant label sets of the instances are generally better with ML-CDT than with ML-DT. The higher is the level of noise introduced in the data, the more notable is this improvement. It happens because if the level of noise is higher, the predictions of the relevant labels are worsened, and ML-CDT is less sensitive to noise than ML-DT.

Secondly, ML-DT predicts more relevant labels correctly than ML-CDT, since the first algorithm always obtains better performance in the metrics associated with Recall. On the other hand, ML-CDT predicts less irrelevant labels as relevant than ML-DT, due to the better results in the measures corresponding to Precision. The performances obtained by both algorithms in the metrics associated with F1 (harmonic mean between Precision and Recall) are statistically equivalent without noise. However, when noise is introduced in the data, the results are more favorable to ML-CDT. The reasons for the previous fact have been commented above: in MLC for some labels, there are very few instances that have associated them. Besides, ML-CDT might stop the tree before than ML-DT. Thus, in many cases, ML-CDT does not reach parts of the tree where relevant labels are predicted correctly. However, in these parts of the tree, the noise has a quite negative influence and there, the irrelevant labels sometimes are predicted erroneously as relevant. Since ML-CDT considers that the data set is less reliable than ML-DT, as the noise is higher, the harmonic means between Precision and Recall are more favorable to ML-CDT.

Also, ML-CDT, for all noise levels, achieves better performance than ML-DT in predicting a greater posterior probability for the relevant labels and lower for the irrelevant ones, as commented in example-based ranking metrics. The results obtained in Binary Cross-Entropy allows us to deduce that the quality of the estimated posterior probabilities is better for ML-CDT than for ML-DT. The reasons are that the noise has a negative influence on the ranking of labels given by the adaptations of DTs to MLC and the building process of ML-CDT is more robust to noise than the one employed in ML-DT. Moreover, the posterior probabilities predicted by ML-CDT are the ones that give rise to the maximum of entropy in the corresponding credal sets, unlike the ones that utilize relative frequencies, as in ML-DT. As we have said before, in terminal nodes there are often few instances and the differences ML-CDT and ML-DT are notable. Hence, the predicted posterior probabilities are less sensitive to noise in the case of ML-CDT.

In general, the number of metrics for which ML-CDT obtains significantly better performance than ML-DT is always higher than the number of measures for which the results are significantly better with ML-DT. Moreover, the more is the level of noise in the data, the higher is the difference.

Therefore, it can be concluded that the use of the NPI-M supposes an improvement in the adaptations of DTs to MLC, being this improvement more notable as more noise is introduced in the data.

5. Conclusions and Future Work

In this work, we have proposed a new adaptation of Decision Trees to Multi-Label Classification based on imprecise probabilities. It uses uncertainty measures on credal sets resulting from the Non-Parametric Predictive Model for Inference in the split criterion and at the time of labeling the instances.

It has been shown that this new Multi-Label Decision Tree is more robust to noise than the already existing one, based on precise probabilities.

Furthermore, we have argued that in Multi-Label Classification the intrinsic noise is usually higher than in standard classification. Hence, the Non-Parametric Predictive Model for Inference is suitable to be applied to the adaptation of Decision Trees to Multi-Label Classification due to its performance on datasets with high levels of noise.

An extensive experimentation has been carried out in this work on different datasets with several levels of added noise. In it, we have employed many Multi-Label Classification metrics in order to compare the performance of the algorithms. This experimental analysis has shown that, in general, the new Multi-Label Decision Tree based on the Non-Parametric Predictive Model for Inference has a better performance than the already existing Multi-Label Decision Tree. This improvement is more notable as more noise is introduced in the data.

As future work, the Non-Parametric Predictive Model for Inference can be also used in the ensembles of Decision Trees in Multi-Label Classification. It is expected that these future ensembles of Multi-Label Decision Tree with imprecise probabilities obtain better results than the already existing ones, which use precise probabilities.

Acknowledgments

This work has been supported by the Spanish "Ministerio de Economía y Competitividad" and by "Fondo Europeo de Desarrollo Regional" (FEDER) under Project TEC2015-69496-R.

Appendix A. Evaluation measures

In this Section we explain the evaluation metrics utilized in order to compare the performance of the MLC algorithms considered in the experiments. We suppose that it is disposed of a test set of size N $\{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$. The same notation of Section 2.1 is employed.

Example-based classification measures:

Example-based classification metrics [2, 52, 53] focus on the predictions made in the test examples.

- **Subset Accuracy**: It indicates the proportion of instances whose predicted label set coincides with its set of relevant labels:

$$Subset\_Accuracy(h) = \frac{1}{N}\sum_{i=1}^{N} I(h(\mathbf{x_i}) = \mathbf{L}_i), \tag{24}$$

being I an indicator function, which has the value 1 if the condition is verified and 0 otherwise.

- **Hamming Loss**:

  It indicates the number of times, on average, an example-label is classified incorrectly:

$$Hamming\_Loss(h) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{q}\left|h(\mathbf{x}_i)\triangle\mathbf{L}_i\right|, \tag{25}$$

  where $\triangle$ denotes the symmetric difference between two sets, which indicates how many elements belong to one set and not to the other one.

- **Accuracy**:

  It consists of the average Jaccard similarity coefficient between the set of labels predicted as relevant and the set of labels that are associated with an instance:

$$Accuracy(h) = \frac{1}{N}\sum_{i=1}^{N} \frac{|h(\mathbf{x}_i) \cap \mathbf{L}_i|}{|h(\mathbf{x}_i) \cup \mathbf{L}_i|}. \tag{26}$$

- **Precision**:

  It indicates the proportion of the labels predicted as relevant that are really relevant for the instances, on average:

$$Precision(h) = \frac{1}{N}\sum_{i=1}^{N} \frac{|h(\mathbf{x}_i) \cap \mathbf{L}_i|}{|h(\mathbf{x}_i)|}. \tag{27}$$

- **Recall**:

  It measures the proportion of relevant labels for the examples that are predicted as relevant, on average:

$$Recall(h) = \frac{1}{N} \sum_{i=1}^{N} \frac{|h(\mathbf{x}_i) \cap \mathbf{L}_i|}{|\mathbf{L}_i|}. \tag{28}$$

- **F1**:

  It is the harmonic mean between Example-based Precision and Example-based Recall:

  $$F1(h) = \frac{1}{N} \sum_{i=1}^{N} \frac{2 \times |h(\mathbf{x}_i) \cap \mathbf{L}_i|}{|h(\mathbf{x}_i)| + |\mathbf{L}_i|}. \tag{29}$$

Label based classification measures:

These measures [25] assume that each label is a binary class variable, whose value is 1 for an instance if the corresponding label if relevant for it and 0 else.

- **Macro Precision**:

  It is defined by the Precision averaged across all the labels:

  $$Macro\_Precision = \frac{1}{q} \sum_{j=1}^{q} \frac{tp_j}{tp_j + fp_j}, \tag{30}$$

  being fp_j and tp_j the number of false positives and true positives for the label j, respectively, $1 \leq j \leq q$.

- **Macro Recall**:

  It indicates the Recall averaged across all the labels:

  $$Macro\_Recall = \frac{1}{q} \sum_{j=1}^{q} \frac{tp_j}{tp_j + fn_j}, \tag{31}$$

  where, for the label l_j, fn_j is the corresponding number of false negatives.

- **Macro F1**:

  It consists of the harmonic mean between Recall and Precision, computed for each tag and averaged over all labels.

$$Macro\_F1 = \frac{1}{q}\sum_{j=1}^{q}\frac{2 \times r_j \times p_j}{r_j + p_j}, \tag{32}$$

  being p_j and r_j, respectively, the Precision and Recall for the jth label.

- **Micro Precision**:

  It is the average of the Precision over all the instance-label pairs

$$Micro\_Precision = \frac{\sum_{j=1}^{q} tp_j}{\sum_{j=1}^{q} tp_j + \sum_{j=1}^{q} fp_j}. \tag{33}$$

- **Micro Recall**:

  It indicates the average of the Recall averaged over all the instance-label pairs.

$$Micro\_Recall = \frac{\sum_{j=1}^{q} tp_j}{\sum_{j=1}^{q} tp_j + \sum_{j=1}^{q} fn_j}. \tag{34}$$

- **Micro F1**:

  It consists of the harmonic mean between Micro Precision and Micro Recall.

$$Micro\_F1 = \frac{2 \times Micro\_Precision \times Micro\_Recall}{Micro\_Rrecision + Micro\_Recall}. \tag{35}$$

Ranking based measures:

Ranking-based metrics [2, 52, 53, 25] focus on the real-valuated function returned by the multi-label classifier and on its related ranking function.

- **One Error**:

  It is the proportion of examples for which the top-ranked label is not associated with it. Formally:

$$One\_Error(f) = \frac{1}{N} \sum_{i=1}^{N} I(\arg\max_{l \in \mathbf{L}} f(\mathbf{x}_i, l) \notin \mathbf{L}_i). \tag{36}$$

- **Coverage**:

  It indicates the average number of steps that are required to do going down the rank of labels to cover all the labels that are associated with an instance.

$$Coverage(f) = \frac{1}{N} \sum_{i=1}^{N} \max_{l \in \mathbf{L}_i} \{rank\_f_{\mathbf{x}_i}(l)\} - 1. \tag{37}$$

- **Ranking Loss**:

  It consists of the average proportion of label pairs which are ordered in a reverse way for a particular example. Formally:

  Let $\mathbf{Z}_i = \{(\,l_n, l_m) \mid f(\mathbf{x}_i, l_m) \leq f(\mathbf{x}_i, l_n), l_m \in \mathbf{L}_i, l_n \in \overline{\mathbf{L}}_i\}$, being $\overline{\mathbf{L}}_i$ the complementary set of $\mathbf{L}_i$. This metric is given by:

$$Ranking\_Loss(f) = \frac{1}{N} \sum_{i=1}^{N} \frac{|\mathbf{Z}_i|}{|\mathbf{L}_i| \left|\overline{\mathbf{L}}_i\right|}. \tag{38}$$

- **Average precision**:

  It measures, on average, the proportion of labels which are higher punctuated than a relevant one.

  Let us consider $\Lambda_i = \left\{l' \mid rank\_f_{\mathbf{x}_i}(l') \leq rank\_f_{\mathbf{x}_i}(l), l' \in \mathbf{L}_i\right\}$. This measure is defined as follows:

$$Average\_Precision(f) = \frac{1}{N} \sum_{i=1}^{N} \frac{|\Lambda_i|}{rank\_f_{\mathbf{x}_i}(l)}. \tag{39}$$

- **Binary Cross-Entropy**:

  It indicates, on average, how far away from the real value of the labels of the instances (1 if the label is relevant for the example and 0 if the label is not associated with the instance) are the predicted posterior probabilities.

$$\textit{binary\_cross\_entropy}(f) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{q}\sum_{j=1}^{q}[[l_j \in \mathbf{L_i}]]\, log(f(\mathbf{x_i}, \mathbf{l_j})) +$$

$$[[l_j \notin \mathbf{L_i}]]\, log(1 - f(\mathbf{x_i}, \mathbf{l_j})).$$

The lower is the value of Coverage, Ranking Loss, One Error and Binary Cross-Entropy, the better is the performance. Nevertheless, with Average Precision, the opposite happens.

Appendix B. Complete experimental results

In this Section we show the complete results from the experimental study. Specifically, the results for each dataset used in the experiments, for each noise and for each metric are presented. In each case, the best result is marked in bold.

Table 5: Complete Subset Accuracy results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	**0.0368**	0.0031	**0.0147**	0.0072	0.0103	**0.0134**
birds	0.1767	**0.4574**	0.1054	**0.445**	0.1147	**0.4062**
CAL500	0	0	0	0	0	0
Corel5k	**0.0006**	0	**0.0006**	0	0.0004	**0.0018**
emotions	0.172	**0.241**	0.1179	**0.2006**	0.1347	**0.2056**
enron	0.01	**0.0147**	0.0065	**0.0182**	0.007	**0.0252**
flags	0.0721	**0.1555**	0.0309	**0.1555**	0.031	**0.1196**
mediamill	0.0517	**0.0551**	0.0323	**0.0559**	0.0285	**0.052**
medical	0.135	**0.2372**	0.0133	**0.1708**	0.002	**0.1779**
genbase	0	**0.6995**	0	**0.7177**	0	**0.5042**
scene	0.3257	**0.504**	0.2613	**0.4911**	0.2505	**0.4495**
yeast	0.0662	**0.1419**	0.0658	**0.1353**	0.0521	**0.1216**

References

[1] A. McCallum, Multi-label text classification with a mixture model trained by EM, in: AAAI'99 Workshop on Text Learning., 1999, pp. 1–7.

[2] R. E. Schapire, Y. Singer, Boostexter: A boosting-based system for text categorization, Machine Learning 39 (2) (2000) 135–168. doi:10.1023/A:1007649029923.

Table 6: Complete Hamming Loss results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	0.0488	**0.0142**	0.5688	**0.038**	0.6085	**0.0633**
birds	0.0945	**0.0537**	0.4505	**0.0721**	0.4605	**0.1324**
CAL500	0.263	**0.1371**	0.3843	**0.1871**	0.5457	**0.2106**
Corel5k	0.049	**0.0098**	0.4543	**0.0153**	0.6811	**0.0308**
emotions	0.2963	**0.2645**	0.3862	**0.3112**	0.3745	**0.3188**
enron	0.14	**0.0586**	0.6043	**0.1028**	0.5736	**0.1238**
flags	0.3513	**0.2906**	0.3829	**0.2987**	0.4151	**0.3068**
mediamill	0.0621	**0.0328**	0.3545	**0.0444**	0.3457	**0.0714**
medical	0.0759	**0.0212**	0.7932	**0.0396**	0.9193	**0.0698**
genbase	0.9247	**0.0161**	0.9536	**0.0399**	0.9536	**0.0517**
scene	0.1976	**0.1576**	0.297	**0.199**	0.3339	**0.2342**
yeast	0.3201	**0.2559**	0.3699	**0.2883**	0.4005	**0.3119**

Table 7: Complete Accuracy results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	**0.1824**	0.0482	**0.0742**	0.0522	**0.0575**	0.0572
birds	0.3162	**0.476**	0.1923	**0.4471**	0.2021	**0.4154**
CAL500	**0.2363**	0.1863	**0.2184**	0.1908	0.1871	**0.1979**
Corel5k	**0.0719**	0.0001	**0.0435**	0.0014	**0.0294**	0.0078
emotions	0.4606	**0.5063**	0.4009	**0.4657**	0.4159	**0.4659**
enron	**0.2645**	0.2306	0.1567	**0.238**	0.1636	**0.2266**
flags	0.5397	**0.577**	0.5176	**0.5776**	0.502	**0.5685**
mediamill	0.35	**0.3649**	0.2445	**0.3656**	0.2405	**0.3512**
medical	**0.3564**	0.3033	0.0879	**0.2316**	0.0435	**0.2402**
genbase	0.0475	**0.7597**	0.0464	**0.7806**	0.0464	**0.5409**
scene	0.4975	**0.5556**	0.4344	**0.5488**	0.4153	**0.5153**
yeast	0.4164	**0.4688**	0.4011	**0.4556**	0.378	**0.4431**

Table 8: Complete Precision results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	**0.2123**	0.144	0.086	**0.1448**	0.0667	**0.1458**
birds	0.3485	**0.5101**	0.2113	**0.4476**	0.2178	**0.416**
CAL500	0.2967	**0.6347**	0.2658	**0.5895**	0.2177	**0.5598**
Corel5k	**0.0802**	0.0001	**0.0483**	0.0033	**0.0323**	0.0152
emotions	0.536	**0.5836**	0.4531	**0.5362**	0.4833	**0.5398**
enron	0.308	**0.5829**	0.1796	**0.4828**	0.1878	**0.5212**
flags	0.5957	**0.6625**	0.5651	**0.6527**	0.5495	**0.6427**
mediamill	0.4225	**0.7032**	0.2912	**0.6873**	0.2869	**0.6791**
medical	0.3662	**0.3732**	0.0914	**0.2948**	0.044	**0.3024**
genbase	0.0475	**0.8254**	0.0464	**0.8285**	0.0464	**0.5637**
scene	0.5111	**0.5785**	0.4463	**0.5731**	0.427	**0.5359**
yeast	0.4963	**0.6054**	0.4728	**0.5873**	0.4475	**0.5629**

Table 9: Complete Recall results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	**0.4339**	0.0482	**0.7118**	0.0767	**0.7162**	0.1071
birds	0.4379	**0.476**	0.4568	**0.4592**	**0.4796**	0.4664
CAL500	**0.5338**	0.2069	**0.6257**	0.274	**0.7377**	0.3119
Corel5k	**0.3473**	0.0097	**0.6182**	0.0071	**0.7783**	0.0301
emotions	0.6638	**0.6834**	0.6714	**0.6746**	0.6641	**0.677**
enron	**0.6806**	0.245	0.8501	0.3244	**0.82**	0.318
flags	**0.831**	0.7527	**0.8398**	0.7744	**0.8412**	0.7822
mediamill	**0.6374**	0.3981	**0.7317**	0.4175	**0.7124**	0.4212
medical	**0.8604**	0.3033	**0.9651**	0.2503	**0.9864**	0.2879
genbase	**0.9957**	0.769	**1**	0.8366	**1**	0.5977
scene	**0.6647**	0.5999	**0.6859**	0.642	**0.6953**	0.6437
yeast	**0.6582**	0.6182	**0.6892**	0.633	**0.6745**	0.6443

Table 10: Complete F1 results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	**0.2527**	0.0699	0.1084	0.074	**0.087**	0.0788
birds	0.3658	**0.4845**	0.2363	**0.4487**	0.2461	**0.4209**
CAL500	**0.3736**	0.3089	**0.349**	0.3146	0.3086	**0.3246**
Corel5k	**0.1234**	0.0001	**0.0755**	0.002	**0.0518**	0.0104
emotions	0.5606	**0.5987**	0.5062	**0.5597**	0.5205	**0.5614**
enron	**0.3875**	0.3299	0.2402	**0.3316**	0.2481	**0.3186**
flags	0.6726	**0.6927**	0.6551	**0.6958**	0.6436	**0.6903**
mediamill	0.4676	**0.478**	0.3364	**0.4786**	0.3328	**0.4629**
medical	**0.4637**	0.326	0.1325	**0.253**	0.0751	**0.2622**
genbase	0.0896	**0.7819**	0.0876	**0.8049**	0.0876	**0.5561**
scene	0.5572	**0.5747**	0.5024	**0.5743**	0.4848	**0.5458**
yeast	0.5381	**0.5805**	0.5242	**0.569**	0.5013	**0.5587**

Table 11: Complete Micro Precision results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	0.1367	**1**	0.019	**0.0516**	0.0175	**0.0333**
birds	0.2989	**0.4239**	**0.0877**	0.0486	0.0837	0.077
CAL500	0.2915	**0.6347**	0.227	**0.3716**	0.1894	**0.3183**
Corel5k	0.0708	**0.0971**	0.0128	0.0128	0.0108	**0.0122**
emotions	0.519	**0.5621**	0.4248	**0.5005**	0.4357	**0.4926**
enron	0.2612	**0.5929**	0.0834	**0.2653**	0.0839	**0.218**
flags	0.6	**0.667**	0.5716	**0.6554**	0.5476	**0.6489**
mediamill	0.3719	**0.742**	0.0854	**0.4866**	0.0858	**0.2781**
medical	0.2495	**0.8171**	0.0324	**0.3874**	0.0289	**0.1443**
genbase	0.0475	**0.9484**	0.0464	**0.5566**	0.0464	**0.4768**
scene	0.4639	**0.5576**	0.3408	**0.4629**	0.3104	**0.403**
yeast	0.479	**0.5712**	0.4316	**0.5208**	0.4033	**0.4887**

Table 12: Complete Micro Recall results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	**0.4193**	0.06	**0.7032**	0.087	**0.7105**	0.1103
birds	**0.5614**	0.0578	**0.7028**	0.0209	**0.7414**	0.1125
CAL500	**0.5297**	0.1997	**0.6231**	0.2655	**0.7357**	0.3057
Corel5k	**0.3467**	0.0004	**0.6167**	0.0072	**0.778**	0.0296
emotions	0.6714	**0.6823**	**0.6776**	0.6748	0.6622	**0.6796**
enron	**0.6545**	0.2569	**0.8277**	0.3467	**0.7969**	0.3361
flags	**0.8309**	0.7897	**0.8392**	0.8013	**0.8398**	0.8051
mediamill	**0.6278**	0.3716	**0.7372**	0.3865	**0.7195**	0.3961
medical	**0.8514**	0.3	**0.959**	0.2622	**0.9858**	0.2935
genbase	**0.993**	0.7001	**1**	0.7813	**1**	0.5708
scene	**0.6558**	0.5888	**0.6829**	0.6298	**0.6916**	0.6354
yeast	**0.6559**	0.6193	**0.6874**	0.6345	**0.6712**	0.6431

Table 13: Complete Micro F1 results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	**0.2061**	0.1131	0.0369	**0.0647**	0.0341	**0.0506**
birds	**0.3872**	0.0985	**0.152**	0.0273	**0.1494**	0.0868
CAL500	**0.3759**	0.3036	**0.3299**	0.2985	0.2951	**0.3022**
Corel5k	**0.1175**	0.0011	**0.025**	0.0084	**0.0212**	0.0164
emotions	0.5853	**0.6162**	0.5215	**0.5746**	0.5251	**0.5701**
enron	**0.3734**	0.3573	0.1511	**0.2992**	0.1516	**0.2566**
flags	0.6959	**0.7228**	0.6798	**0.7206**	0.6622	**0.7171**
mediamill	0.4671	**0.4951**	0.153	**0.4301**	0.1532	**0.3259**
medical	0.3851	**0.4382**	0.0627	**0.2796**	0.0562	**0.1901**
genbase	0.0907	**0.7999**	0.0886	**0.6487**	0.0886	**0.5113**
scene	0.5431	**0.5722**	0.4534	**0.5324**	0.4276	**0.493**
yeast	0.5536	**0.5942**	0.5297	**0.5714**	0.5037	**0.5552**

Table 14: Complete Macro Precision results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	**0.1358**	0.0075	0.0186	**0.0218**	0.0173	**0.0201**
birds	**0.2839**	0.0578	**0.0824**	0.0421	**0.0794**	0.0705
CAL500	**0.1602**	0.0793	**0.1531**	0.1482	0.1495	**0.1502**
Corel5k	0.0486	**0.15**	**0.0103**	0.0094	0.0098	**0.0109**
emotions	0.5145	**0.5559**	0.4234	**0.4918**	0.4297	**0.4885**
enron	**0.1264**	0.1121	0.0688	**0.0865**	0.0687	**0.0835**
flags	0.5456	**0.6107**	0.5245	**0.6094**	0.5135	**0.6039**
mediamill	**0.1387**	0.0406	0.054	**0.0627**	0.0539	**0.0585**
medical	0.2508	**0.3601**	0.0318	**0.1205**	0.0287	**0.0489**
genbase	0.046	**0.5331**	0.0464	**0.2955**	0.0464	**0.2166**
scene	0.4998	**0.5591**	0.3537	**0.4635**	0.3145	**0.4028**
yeast	0.3646	**0.3984**	0.3439	**0.3725**	0.3311	**0.3592**

Table 15: Complete Macro Recall results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	**0.3353**	0.0068	**0.6641**	0.0326	**0.681**	0.0564
birds	**0.4845**	0.0537	**0.6172**	0.0163	**0.6796**	0.0977
CAL500	**0.2851**	0.0966	**0.4179**	0.1087	**0.5932**	0.1563
Corel5k	0.1235	**0.1467**	**0.4092**	0.0057	**0.5795**	0.0197
emotions	0.6593	**0.662**	**0.6674**	0.6585	0.6505	**0.675**
enron	**0.3187**	0.1153	**0.6498**	0.1202	**0.6047**	0.1366
flags	**0.7754**	0.7391	**0.7874**	0.7385	**0.7956**	0.7509
mediamill	**0.2987**	0.0306	**0.5157**	0.0469	**0.5125**	0.0798
medical	**0.4951**	0.3683	**0.6094**	0.1197	**0.6638**	0.0857
genbase	**0.7556**	0.541	**0.7852**	0.4268	**0.7852**	0.2915
scene	**0.6644**	0.6018	**0.6892**	0.6438	**0.697**	0.6478
yeast	**0.5103**	0.4218	**0.5547**	0.4492	**0.5488**	0.4702

Table 16: Complete Macro F1 results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	**0.1827**	0.007	**0.0353**	0.0246	**0.033**	0.0271
birds	**0.3345**	0.0536	**0.1356**	0.0209	**0.1354**	0.0716
CAL500	**0.1995**	0.0857	**0.2087**	0.1011	**0.214**	0.1284
Corel5k	0.0602	**0.1467**	**0.0189**	0.0046	**0.0183**	0.0092
emotions	0.5756	**0.5958**	0.5136	**0.557**	0.513	**0.5592**
enron	**0.1691**	0.1122	**0.1067**	0.0879	**0.1056**	0.0843
flags	0.6283	**0.6513**	0.617	**0.6465**	0.6079	**0.6567**
mediamill	**0.1844**	0.0328	**0.0796**	0.0453	**0.0792**	0.0531
medical	0.2942	**0.3637**	0.056	**0.1135**	0.0514	**0.0547**
genbase	0.0827	**0.5327**	0.0833	**0.3352**	0.0833	**0.2317**
scene	0.5648	**0.5761**	0.4641	**0.5353**	0.4314	**0.4944**
yeast	**0.42**	0.4036	**0.413**	0.4027	0.3996	**0.4016**

Table 17: Complete Ranking Loss results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	0.3007	**0.2113**	0.287	**0.2455**	0.3237	**0.2661**
birds	0.1559	**0.1473**	**0.1554**	0.1604	**0.1565**	0.1711
CAL500	0.3845	**0.1809**	0.3764	**0.2185**	0.3577	**0.2368**
Corel5k	0.4717	**0.1441**	0.3932	**0.1666**	0.3301	**0.1849**
emotions	0.2942	**0.2292**	0.3361	**0.2634**	0.3498	**0.2854**
enron	0.2193	**0.1031**	0.1813	**0.1366**	0.222	**0.1546**
flags	0.2961	**0.2556**	0.3357	**0.2754**	0.3557	**0.2682**
mediamill	0.1563	**0.0525**	0.1608	**0.0607**	0.206	**0.08**
medical	**0.0725**	0.0982	**0.1066**	0.1115	**0.1053**	0.1217
genbase	0.1557	**0.01**	0.1566	**0.0263**	0.1573	**0.0653**
scene	0.2114	**0.1797**	0.2433	**0.2009**	0.2606	**0.229**
yeast	0.3198	**0.231**	0.3337	**0.2555**	0.3677	**0.2799**

Table 18: Complete One Error results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	**0.6154**	0.7198	**0.7025**	0.7454	**0.7343**	0.7588
birds	**0.7349**	0.8016	**0.7612**	0.8326	**0.7783**	0.8419
CAL500	0.5397	**0.1155**	0.5538	**0.1852**	0.4703	**0.1992**
Corel5k	0.8132	**0.7764**	0.7824	**0.7692**	0.7936	**0.7664**
emotions	0.4049	**0.3559**	0.4672	**0.3912**	0.4673	**0.4182**
enron	0.3931	**0.376**	0.4383	**0.3849**	**0.4301**	0.4407
flags	0.3238	**0.2726**	0.3758	**0.2727**	0.3451	**0.2675**
mediamill	0.3268	**0.201**	0.3272	**0.2195**	0.3782	**0.2409**
medical	**0.18**	0.5296	**0.4775**	0.6042	**0.4949**	0.5757
genbase	0.7416	**0.1162**	0.7416	**0.1117**	0.7416	**0.322**
scene	0.4047	**0.3922**	0.447	**0.4126**	0.4578	**0.4371**
yeast	0.3492	**0.3045**	0.3765	**0.3446**	0.4357	**0.3674**

Table 19: Complete Coverage results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	70.8926	**47.5905**	64.9024	**54.8811**	71.3719	**59.6375**
birds	3.9674	**3.8248**	**3.831**	4.0822	**3.8868**	4.2729
CAL500	166.7654	**129.3176**	164.1748	**135.8787**	155.7104	**136.4531**
Corel5k	287.8468	**120.5674**	254.965	**136.1556**	220.9626	**146.9834**
emotions	2.4839	**2.0711**	2.6089	**2.2092**	2.7168	**2.3105**
enron	25.6047	**14.0697**	21.1962	**16.8145**	23.3036	**17.7864**
flags	4.1297	**3.9731**	4.3043	**4.0457**	4.459	**4.0811**
mediamill	41.3202	**17.8061**	38.2073	**19.125**	43.7239	**21.5757**
medical	**4.1273**	5.2019	**5.7065**	5.8991	**5.5625**	6.3209
genbase	4.7722	**0.5571**	4.7966	**1.0161**	4.8298	**2.13**
scene	1.1458	**0.9842**	1.3103	**1.0915**	1.3972	**1.2327**
yeast	8.6517	**7.2115**	8.6752	**7.4587**	9.0009	**7.6918**

Table 20: Complete Average Precision results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	**0.3485**	0.2473	**0.267**	0.2256	**0.2388**	0.2138
birds	**0.5235**	0.4427	**0.5024**	0.4062	**0.4817**	0.4055
CAL500	0.3326	**0.4981**	0.344	**0.4678**	0.3594	**0.4542**
Corel5k	0.1537	**0.2083**	0.1742	**0.2108**	0.1798	**0.2126**
emotions	0.697	**0.7473**	0.6624	**0.7214**	0.6571	**0.7041**
enron	0.5497	**0.5659**	0.5397	**0.5525**	**0.5235**	0.5159
flags	0.7705	**0.7829**	0.7402	**0.7695**	0.7257	**0.7715**
mediamill	0.5715	**0.6789**	0.5715	**0.6657**	0.5283	**0.6463**
medical	**0.8294**	0.5617	**0.5919**	0.5146	**0.5934**	0.5303
genbase	0.4316	**0.9268**	0.4341	**0.9233**	0.4321	**0.7629**
scene	0.7281	**0.747**	0.6953	**0.7292**	0.6849	**0.7061**
yeast	0.6442	**0.7036**	0.6308	**0.6808**	0.5994	**0.6627**

Table 21: Complete Binary Cross-Entropy results.

	0%		5%		10%	
Dataset	ML-DT	ML-CDT	ML-DT	ML-CDT	ML-DT	ML-CDT
bibtex	143.0411	**10.3074**	1665.8423	**16.7924**	1782.1928	**25.5771**
birds	33.0715	**3.2829**	157.6753	**3.9639**	161.1881	**5.1859**
CAL500	842.9059	**57.8521**	1231.7676	**64.6487**	1749.0878	**69.9017**
Corel5k	337.8832	**16.324**	3129.81	**31.0223**	4692.5358	**49.3165**
emotions	32.7453	**3.0986**	42.6874	**3.3595**	41.3939	**3.3887**
enron	136.6849	**8.3549**	589.9294	**11.0241**	559.9768	**13.4961**
flags	45.301	**3.7887**	49.3729	**3.9073**	53.5219	**4.0048**
mediamill	115.4692	**10.0265**	659.5249	**14.0049**	643.2242	**19.8996**
medical	62.9378	**4.4924**	657.5463	**5.6436**	762.0054	**7.8775**
genbase	459.8889	**1.3345**	474.2927	**2.7768**	474.2927	**4.7212**
scene	21.8344	**2.1346**	32.8226	**2.3008**	36.9019	**2.4948**
yeast	82.5386	**7.0919**	95.3947	**7.3346**	103.2745	**7.6661**

[3] Z. Barutcuoglu, R. E. Schapire, O. G. Troyanskaya, Hierarchical multi-label prediction of gene function, Bioinformatics 22 (7) (2006) 830–836. doi:10.1093/bioinformatics/btk048.

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

[5] G. Nasierding, A. Kouzani, Image to Text Translation by Multi-Label Classification, in: Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, Vol. 6216 of Lecture Notes in Computer Science, Springer, 2010, Ch. 31, pp. 247–254. doi:10.1007/978-3-642-14932-0_31.

[6] M. L. Zhang, Z. H. Zhou, A review on multi-label learning algorithms, IEEE Transactions on Knowledge and Data Engineering 26 (8) (2014) 1819–1837. doi:10.1109/TKDE.2013.39.

[7] A. Clare, R. D. King, Knowledge discovery in multi-label phenotype data, in: L. De Raedt, A. Siebes (Eds.), Principles of Data Mining and Knowledge Discovery, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 42–53.

[8] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[9] J. Abellán, S. Moral, Building classification trees using the total uncertainty criterion, International Journal of Intelligent Systems 18 (12) (2003) 1215–1225. doi:10.1002/int.10143.

[10] J. Abellán, A. 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, Vol. 5590 of Lecture Notes in Computer Science, Springer, 2009, pp. 446–456. doi:10.1007/978-3-642-02906-6_39.

[11] J. Abellán, C. J. Mantas, Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring, Expert Systems with Applications 41 (8) (2014) 3825 – 3830. doi:10.1016/j.eswa.2013.12.003.

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

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

[14] C. J. Mantas, J. Abellán, J. G. Castellano, Analysis of Credal-C4.5 for classification in noisy domains, Expert Systems with Applications 61 (2016) 314 – 326. doi:10.1016/j.eswa.2016.05.035.

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

[16] J. Abelln, C. J. Mantas, J. G. Castellano, Adaptativecc4.5: Credal c4.5 with a rough class noise estimator, Expert Systems with Applications 92 (2018) 363 – 379. doi:https://doi.org/10.1016/j.eswa.2017.09.057.

[17] F. P. A. Coolen, Learning from multinomial data: a nonparametric predictive alternative to the Imprecise Dirichlet Model, 2005.

[18] J. Abellán, R. M. Baker, F. P. Coolen, R. J. Crossman, A. R. Masegosa, Classification with decision trees from a nonparametric predictive inference perspective, Computational Statistics & Data Analysis 71 (2014) 789 – 802. doi:https://doi.org/10.1016/j.csda.2013.02.009.

[19] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Machine Learning 85 (3) (2011) 333. doi:10.1007/s10994-011-5256-5.

[20] M. Ioannou, G. Sakkas, G. Tsoumakas, I. Vlahavas, Obtaining bipartitions from score vectors for multi-label classification, in: 2010 22nd IEEE International Conference on Tools with Artificial Intelligence, Vol. 1, 2010, pp. 409–416. doi:10.1109/ICTAI.2010.65.

[21] M.-L. Zhang, Z.-H. Zhou, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Transactions on Knowledge and Data Engineering 18 (10) (2006) 1338–1351. doi:10.1109/TKDE.2006.162.

[22] A. Elisseeff, J. Weston, A kernel method for multi-labelled classification, in: In Advances in Neural Information Processing Systems 14, Vol. 14, 2001, pp. 681–687.

[23] J. Fürnkranz, E. Hüllermeier, E. Loza Mencorthogonal a, K. Brinker, Multilabel classification via calibrated label ranking, Machine Learning (2008). doi:10.1007/s10994-008-5064-8.

[24] M. R. Boutell, J. Luo, X. Shen, C. M. Brown, Learning multi-label scene classification, Pattern Recognition 37 (9) (2004) 1757 – 1771. doi:10.1016/j.patcog.2004.03.009.

[25] G. Tsoumakas, I. Vlahavas, Random k-labelsets: An ensemble method for multilabel classification, in: European Conference on Machine Learning, Springer, 2007, pp. 406–417. doi:10.1007/978-3-540-74958-5_38.

[26] M.-L. Zhang, Z.-H. Zhou, Ml-knn: A lazy learning approach to multi-label learning, Pattern Recognition 40 (7) (2007) 2038 – 2048. doi:https://doi.org/10.1016/j.patcog.2006.12.019.

[27] A. Elisseeff, J. Weston, A kernel method for multi-labelled classification, in: In Advances in Neural Information Processing Systems 14, Vol. 14, 2001, pp. 681–687. doi:10.1.1.29.432.

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

[29] L. M. De Campos, J. F. Huete, S. Moral, Probability intervals: a tool for uncertain reasoning, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 02 (02) (1994) 167–196. doi:10.1142/S0218488594000146.

[30] P. Suppes, The measurement of belief, Journal of the Royal Statistical Society. Series B (Methodological) 36 (2) (1974) 160–191.

[31] F. Coolen, T. Augustin, A nonparametric predictive alternative to the imprecise dirichlet model: The case of a known number of categories, International Journal of Approximate Reasoning 50 (2) (2009) 217 – 230, special Section on The Imprecise Dirichlet Model and Special Section on Bayesian Robustness (Issues in Imprecise Probability). doi:https://doi.org/10.1016/j.ijar.2008.03.011.

[32] B. M. Hill, Posterior distribution of percentiles: Bayes' theorem for sampling from a population, Journal of the American Statistical Association 63 (322) (1968) 677–691. doi:10.1080/01621459.1968.11009286.

[33] B. M. Hill, De finettis theorem, induction, and a (n) or bayesian nonparametric predictive inference (with discussion), Bayesian statistics 3 (1988) 211–241.

[34] J. Abellán, R. M. Baker, F. P. Coolen, Maximising entropy on the nonparametric predictive inference model for multinomial data, European Journal of Operational Research 212 (1) (2011) 112–122. doi:10.1016/j.ejor.2011.01.020.

[35] G. J. Klir, Uncertainty and Information: Foundations of Generalized Information Theory, John Wiley And Sons, Inc., 2005. doi:10.1002/0471755575.

[36] G. Madjarov, D. Kocev, D. Gjorgjevikj, S. Džeroski, An extensive experimental comparison of methods for multi-label learning, Pattern Recognition 45 (9) (2012) 3084 – 3104. doi:10.1016/j.patcog.2012.03.004.

[37] I. Katakis, G. Tsoumakas, I. Vlahavas, Multilabel Text Classification for Automated Tag Suggestion, in: Proceedings of the ECML/PKDD 2008 Discovery Challenge, 2008.

[38] B. Klimt, Y. Yang, The enron corpus: A new dataset for email classification research, in: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Machine Learning: ECML 2004, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 217–226.

[39] J. P. Pestian, C. Brew, P. Matykiewicz, D. J. Hovermale, N. Johnson, K. B. Cohen, W. Duch, A shared task involving multi-label classification of clinical free text, in: Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, Association for Computational Linguistics, 2007, pp. 97–104.

[40] S. Diplaris, G. Tsoumakas, P. A. Mitkas, I. Vlahavas, Protein classification with multiple algorithms, in: P. Bozanis, E. N. Houstis (Eds.), Advances in Informatics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 448–456.

[41] F. Briggs, Y. Huang, R. Raich, K. Eftaxias, Z. Lei, W. Cukierski, S. F. Hadley, A. Hadley, M. Betts, X. Z. Fern, J. Irvine, L. Neal, A. Thomas, G. Fodor, G. Tsoumakas, H. W. Ng, T. N. T. Nguyen, H. Huttunen, P. Ruusuvuori, T. Manninen, A. Diment, T. Virtanen, J. Marzat, J. Defretin, D. Callender, C. Hurlburt, K. Larrey, M. Milakov, The 9th annual mlsp competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment, in: 2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), 2013, pp. 1–8. doi:10.1109/MLSP.2013.6661934.

[42] D. Turnbull, L. Barrington, D. Torres, G. Lanckriet, Semantic annotation and retrieval of music and sound effects, IEEE Transactions on Audio, Speech, and Language Processing 16 (2) (2008) 467–476. doi:10.1109/TASL.2007.913750.

[43] P. Duygulu, K. Barnard, J. F. G. de Freitas, D. A. Forsyth, Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, in: A. Heyden, G. Sparr, M. Nielsen, P. Johansen (Eds.), Computer Vision — ECCV 2002, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 97–112.

[44] K. Trohidis, G. Tsoumakas, G. Kalliris, I. P. Vlahavas, Multi-label classification of music into emotions., in: ISMIR, Vol. 8, 2008, pp. 325–330.

[45] C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, A. W. M. Smeulders, The challenge problem for automated detection of 101 semantic concepts in multimedia, in: Proceedings of the 14th ACM International Conference on Multimedia, MM '06, ACM, 2006, pp. 421–430. doi:10.1145/1180639.1180727.

[46] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, I. Vlahavas, Mulan: A java library for multi-label learning, Journal of Machine Learning Research 12 (2011) 2411–2414.

[47] I. H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition, Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[48] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[49] F. Charte, A. J. Rivera, D. Charte, M. J. del Jesus, F. Herrera, Tips, guidelines and tools for managing multi-label datasets: The mldr.datasets r package and the cometa data repository, Neurocomputing In Press (2018). doi:10.1016/j.neucom.2018.02.011.

[50] F. Wilcoxon, Individual Comparisons by Ranking Methods, Biometrics Bulletin 1 (6) (1945) 80–83. doi:10.2307/3001968.

[51] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria (2013).
URL http://www.R-project.org/

[52] N. Ghamrawi, A. McCallum, Collective multi-label classification, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, CIKM '05, ACM, 2005, pp. 195–200. doi:10.1145/1099554.1099591.

[53] S. Godbole, S. Sarawagi, Discriminative methods for multi-labeled classification, in: Advances in Knowledge Discovery and Data Mining, Springer, 2004, pp. 22–30. doi:10.1007/978-3-540-24775-3_5.