Computer Science and Information Systems 15(1):187–209 https://doi.org/10.2298/CSIS170115043S

HMC-ReliefF: Feature Ranking for Hierarchical Multi-label Classification

Ivica Slavkov[1,2], Jana Karcheska[3], Dragi Kocev[4,5], and Sašo Džeroski[4,5]

[1] Centre for Genomic Regulation (CRG), Barcelona, Spain
[2] Universitat Pompeu Fabra (UPF), Barcelona, Spain
ivica.slavkov@crg.eu
[3] University Ss. Cyril and Methodius, Skopje, Macedonia
j.karcheska@gmail.com
[4] Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
[5] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
{dragi.kocev,saso.dzeroski}@ijs.si

Abstract. In machine learning, the growing complexity of the available data poses an increased challenge for its analysis. The rising complexity is both in terms of the data becoming more high-dimensional as well as the data having a more intricate structure. This emphasizes the need for developing machine learning algorithms that are able to tackle both the high-dimensionality and the complex structure of the data. Our work in this paper focuses on the development and analysis of the HMC-ReliefF algorithm, which is a feature relevance (ranking) algorithm for the task of Hierarchical Multi-label Classification (HMC). The basis of the algorithm is the RReliefF algorithm for regression that is adapted for hierarchical multi-label target variables. We perform an extensive experimental investigation of the HMC-ReliefF algorithm on several datasets from the domains of image annotation and functional genomics. We analyse the algorithm's performance in terms of accuracy in a filter-like setting and also in terms of ranking stability for various parameter values. The results show that the HMC-ReliefF can successfully detect relevant features from the data that can be further used for constructing accurate predictive models. Additionally, the stability analysis helps to determine the preferred parameter values for obtaining not just accurate, but also a stable algorithm output.

Keywords: feature selection, feature ranking, structured data, hierarchical multi-label classification, ReliefF.

1. Introduction

The current trend in machine learning is that the data available for analysis is becoming increasingly more complex. The complexity arises both from the data being high-dimensional and from the data being more structured. On one hand, high-dimensional data presents specific challenges for many machine learning algorithms, especially with the stability of the produced results [13]. On the other hand, mining complex data and extracting knowledge from it has been identified as one of the most challenging problems in machine learning [6, 21].

Various feature selection methods exist for dealing with the high-dimensionality of the data. They usually precede the induction of predictive models and can be classified

as filter, wrapper and embedded methods [12]. Filter methods [3] are the simplest ones and they usually involve a feature ranking algorithm that produces a list of relevant features. Wrapper methods [19] rely on classification algorithms to perform feature selection and are computationally expensive. Embedded methods [12] are basically classification algorithms that have the feature selection embedded in the model induction phase.

Learning in a supervised context, where the target is structured, has also attracted much attention. Several algorithms that were previously employed only for classification or regression purposes, have been extended to also work with structured targets. These include decision trees for hierarchical targets [40], support vector machines for multi-label and hierarchical multi-label problems [11], as well as tree ensembles that can be additionally employed for vectors of multiple targets [18].

Our work in this paper focuses on tackling the feature selection problem in the context of structured targets. We consider this a relevant problem in machine learning that relates to both of the previously discussed trends. So far, structured prediction has not been extensively researched in the context of feature ranking methods and we consider this a novel and interesting line of research to pursue.

More specifically, we focus on the ReliefF [28] algorithm for feature ranking. This algorithm is an intuitive, instance based algorithm and its theoretical properties have been extensively explored [28]. We extend ReliefF for a specific type of structured prediction problems, namely those from the Hierarchical Multi-label Classification (HMC) task [30]. The target that is predicted for these problems is defined with a hierarchy of classes and each instance in the dataset can be labelled with more than one class at a time. By definition, when an instance is labelled with one class it is also labelled with all of its parent classes according to the given hierarchy.

In practice, these types of problems appear in different domains, for example in biology for the task of gene function prediction or in image retrieval for the task of image annotation. For the task of gene function prediction, each gene can be annotated by multiple functions and the functions are organised into a tree-shaped hierarchy or a directed acyclic graph such as the Gene Ontology [2]. Thus, predicting the function of a gene from certain gene properties would have to take into account the multi-label annotation of each gene and also the hierarchical connections of these labels.

The work presented in this paper is based on an initial investigation in [32] and [33], which is extended along several dimensions. First, we compare the performance of the HMC-ReliefF method with a feature ranking method based on binary relevance [34] – typically considered as a baseline for comparison of feature ranking for (hierarchical) multi-label classification. Next, we perform stability analysis of the rankings produced with the proposed method, which allows us to gain insight both into the properties of the algorithm, as well as in the nature of the datasets under investigation. Finally, we provide an analysis of the computational complexity of the algorithm. Overall, this amounts to an extensive empirical analysis of the proposed HMC-ReliefF method.

In the remainder of this paper, we present the details of our work organised as follows. In Section 3, we define more formally the HMC setting and present the distance measures appropriate for this setting. Next, in Section 4, we discuss in depth the original RReliefF algorithm for regression and explain our HMC-ReliefF extension of the algorithm. We present our experimental design for evaluating the proposed HMC-ReliefF algorithm in

Section 5. We then discuss the obtained results in Section 6. Finally, in Section 7, we present our conclusions and discuss directions of possible further work.

2. Related Work

The task of hierarchical multi-label classification (HMC) has received wide attention from the research community [30]. Several methods for solving the task of HMC have been proposed and applied to real-life problems. However, the task of feature selection or feature ranking has not received much attention mainly due to the complexity of the task. Most of the related work comes from two other machine learning tasks: multi-label classification and hierarchical text categorization.

The few available methods for feature ranking for multi-label classification are mainly based on the problem transformation paradigm [34] thus they do not fully exploit the possible label dependencies. These methods are based on the binary relevance or label powerset approaches to multi-label classification. In the binary relevance methodology, a separate dataset is created for each label and then a simple feature ranking algorithm is executed for each label separately. The feature rankings are then aggregated, typically using averaging, into a single feature ranking.

The basis of the label powerset methods is to combine entire label sets into atomic (single) labels to form a single-label problem (i.e., single-class classification problem) [37, 9]. For the single-label problem, the set of possible single labels represents all distinct label subsets from the original multi-label representation. In this way, label powerset based methods directly take into account the label correlations. However, the space of possible label subsets can be very large. To resolve this issue, Read [26] has developed a pruned problem transformation method, that selects only the transformed labels that occur more than a predefined number of times. Tsoumakas et al. [37] use the label powerset transformed dataset to calculate simple χ^2 statistic thus producing a ranking of the features. Doquire and Verleysen [9] use the pruned, problem transformed dataset to calculate mutual information for performing feature selection and they show that this method outperforms the χ^2-based feature ranking.

Notwithstanding, Spolaôr et al. in [35] and [36] have proposed extensions of ReliefF towards the task of MLC by redefining the distance function. More precisely, they propose to use normalized Hamming loss or Jaccard dissimilarity as distance functions between two examples. The results from the evaluated on synthetic datasets show that the proposed method yields competitive results with other problem transformation methods (such as the one discussed above). The main differences between their approach and the approach proposed here are the tasks being addressed: (1) we address the task of HMC, while they address the task of MLC and (2) we consider the task of feature ranking, while they address the task of feature selection.

In the task of hierarchical text categorization, the existing methods generally use a binary relevance approach to the task of feature ranking for HMC. Moreover, in hierarchical text categorization, the task of feature selection (i.e., dimensionality reduction) has received most of the attention, while the task of feature ranking in this context has not been treated thus far. The majority of the methods performs binary relevance feature selection and then utilize the hierarchy to propagate the 'good' features from the bottom labels to the root label [24]. Here, we briefly list several such methods.

Wibowo and Williams [42] perform feature selection at each node of the hierarchy and then construct an architecture of complex classifiers, i.e., construct a complex classifier for each hierarchy node (label). Second, Wang et al. [41] define term-label contribution criterion and then for each term (i.e., feature) aggregate its value over all labels. Next, Jia et al. [15] construct separate feature space for each group of siblings in the classification hierarchy using feature selection with category analysis [41]. This improves the performance compared to a single feature space. Furthermore, Vateekul [39] constructs a decision tree for each label separately and then used the features that were used by the trees are selected as the most important. Finally, Esuli et al. [10] proposed a boosting-like method that uses feature selection at each hierarchy node and then employs Ada boost on the selected features.

The existing related approaches to feature ranking for HMC with problem transformation has two major drawbacks. First, the label dependencies and interconnections are not fully exploited. Second, while the feature subset sizes for the local classifiers are relatively small, the overall number of features remains quite large. Consequently, these methods are not scalable to domains with a large number of labels. To address these issues, we propose an algorithm adaptation method that is based on RReliefF. The proposed method is able to exploit the (hierarchical) dependencies between the labels and is scalable to domains with large number of labels.

3. Hierarchical Multi-label Classification

In our work we extend the ReliefF algorithm for the task of hierarchical multi-label classification (HMC). Hierarchical classification is a specific type of a classification task in which the classes are organised in a hierarchy. An example that belongs to a given class automatically belongs to all its super-classes (this is known as the *hierarchy constraint*). Furthermore, if an example can belong simultaneously to multiple classes that can follow multiple paths from the root class, then the task is called HMC [40, 30].

We formally define the hierarchical multi-label classification setting as follows:

- A description space X that consists of tuples of values of primitive data types (discrete or continuous), i.e., $\forall X_i \in X, X_i = (x_{i_1}, x_{i_2}, ..., x_{i_f})$, where f is the size of the tuple (or number of descriptive variables/features),
- a target space S, defined with a class hierarchy $(C, \leq_h)$, where C is a set of classes and $\leq_h$ is a partial order (e.g., structured as a rooted tree) representing the superclass relationship ($\forall\, c_1, c_2 \in C : c_1 \leq_h c_2$ if and only if c_1 is a superclass of c_2),
- a set of examples E, where each example is a pair of a tuple and a set, from the descriptive and target space respectively, and each set satisfies the hierarchy constraint, i.e., $E = \{(X_i, S_i) | X_i \in X, S_i \subseteq C, c \in S_i \Rightarrow \forall c' \leq_h c : c' \in S_i, 1 \leq i \leq N\}$ and N is the number of examples in E ($N = |E|$).

Two toy examples of classes organised in hierarchies can be seen in Figure 1. The first hierarchy in Figure 1(a) consists of five classes $\{c_1, c_2, c_3, c_{2.1}, c_{2.2}\}$, organised in a tree-like structure. The other hierarchy in Figure 1(c), contains six classes ($c_1 - c_6$) and they are organised in a directed acyclic graph (DAG), where each class can have multiple parents.

Calculating the distance between two different instances of the target space S_1 and S_2, can be done in different ways. The different distances include: weighted Euclidean distance for HMC [40], Jaccard distance (also known as Union-intersection distance/score) [14], simGIC (Similarity for Graph Information Content) [25] and ImageCLEF (evaluation score of the ImageCLEF image annotation task) [5]. An experimental evaluation comparing these distances in the context of HMC [1] has shown that learning predictive models that use the different distances, does not produce statistically significant differences in predictive performance.

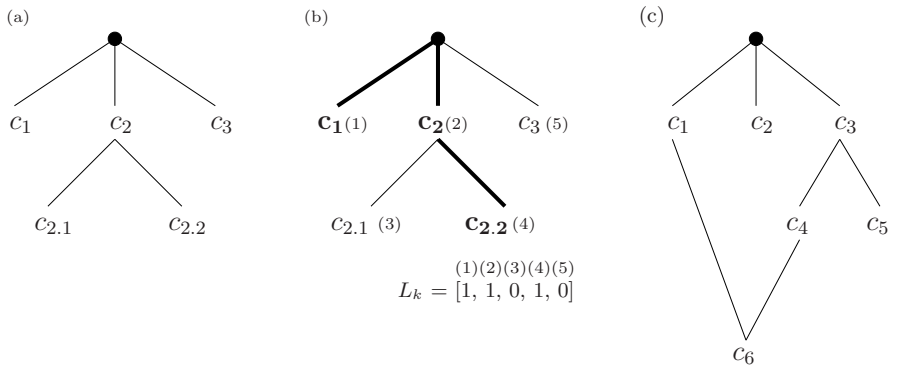


Fig. 1. Toy examples of hierarchies structured as a tree and a DAG.(a) Class label names contain information about the position in the hierarchy, e.g., $c_{2.1}$ is a subclass of c_2. (b) The set of classes $S_1 = \{c_1, c_2, c_{2.2}\}$, shown in bold in the hierarchy, represented as a vector (L_k). (c) A class hierarchy structured as a DAG. The class c_6 has two parents: c_1 and c_4.

In our work, we chose to extend the RReliefF algorithm (the ReliefF algortihm for regression [28]) by using a weighted Euclidean distance for HMC [40]. With this weighted Euclidean distance, the hierarchical aspect is incorporated by relating the class weight with the depth of the class within the hierarchy. Extending RReliefF with this distance is the most straightforward choice, considering that the original algorithm uses the Euclidean distance for calculating the distance for the target variable.

Before calculating the distance between two instances of the hierarchy, they are first represented as a vector of binary values [40]. The vector is created by traversing the tree or DAG that is representing the hierarchy in pre-order and assigning a 1 or 0 sequentially in the vector for a present or absent label respectively. For example, consider an instance of the toy class hierarchy S_1, given in boldface in Figure 1(b). This particular instance consists of three classes, namely $\{c_1, c_2, c_{2.2}\}$ and its corresponding vector representation would be $L_1 = [1, 1, 0, 1, 0]$.

If we additionally consider another instance S_2, labelled just with class $\{c_2\}$, with a vector representation $L_2 = [0, 1, 0, 0, 0]$, then the distance between S_1 and S_2 would be obtained by simply comparing the two binary vectors. In our HMC-ReliefF algorithm we use a weighted Euclidean distance measure given with the following equation:

$$d(L_1, L_2) = \sqrt{\sum_i w(c_i)(L_{1,i} - L_{2,i})^2}, \tag{1}$$

The weighting function $w(c)$ allows for the hierarchical structure of the classes to be taken into account by making the value dependent on the depth of the hierarchy:

$$w(c) = w_0^{depth(c)}, 0 < w_0 < 1. \tag{2}$$

This scheme ensures that the differences higher in the hierarchy have larger influence on the total distance.

For the specific case of comparing S_1 and S_2, the distance is calculated as follows:

$$d(S_1, S_2) = d([1, 1, 0, 1, 0], [0, 1, 0, 0, 0]) = \sqrt{w_0 + w_0^2}.$$

where $w(c_1) = w_0$ and $w(c_3) = w_0^2$.

If the hierarchy is represented with a DAG, this scheme needs to be modified. In this case, more than one path from the root to a given class may exist and thus a node can have different depths. This problem is solved with the following recursive equation:

$$w(c) = w_0 \cdot avg(w(parent_j(c))). \tag{3}$$

By using this weighting function, the weight of the different possible parents is averaged. This is recommended as a good way to take into account multiple inheritance which occurs in DAGs [40].

4. HMC-ReliefF Algorithm

Algorithms from the Relief family are instance-based methods for estimating feature relevance. The original Relief algorithm is formulated for binary classification problems [17]. The algorithm was extended to deal with multi-class problems and the extension was named ReliefF [20]. Later, it was also adapted for regression problems and named RReliefF [27].

In general, the feature relevance value assigned by the Relief algorithm to a feature F is an approximation of the following difference of probabilities [20]:

$$\begin{aligned} W[F] = P(&\text{diff. value of } F | \text{nearest inst. from diff. class}) - \\ &P(\text{diff. value of } F | \text{nearest inst. from same class}) \end{aligned} \tag{4}$$

In the case of classification, the basic intuition behind the ReliefF algorithm is to estimate the relevance of a feature according to how well it distinguishes between neighbouring instances. If the feature has different values for neighbouring instances that are of different class (nearest miss), then it is awarded a higher relevance values. However, if the values of the class for the neighbouring instances are the same (nearest hit), then the relevance value is decreased.

Although the hierarchical multi-label setting is a classification one, extending the ReliefF algorithm is not a good idea. Namely, if we simply treat two instances annotated by

different parts of the hierarchy in a simple hit/miss scenario, we would simply translate the HMC problem to a multi-class one, therefore ignoring both the hierarchical and the multi-label aspect. Having in mind that the definition of the HMC distance in Section 3 is actually weighted Euclidean distance, it is more suited to be included in the RReliefF algorithm, originally designed for regression.

In a regression setting, the target space is continuous and the concept of nearest hit/miss does not apply. Therefore, the feature relevance $W[F]$ is reformulated as the difference between the following probabilities:

$$\begin{aligned} W[F] = P(\text{diff. value of } F|\text{nearest inst. with diff. prediction}) - \\ P(\text{diff. value of } F|\text{nearest inst. with same prediction}) \end{aligned} \tag{5}$$

Additionally, if we introduce the following probabilities:

$$P_{diffF}(\text{diff. value of } F|\text{nearest instance})$$

and

$$P_{diffC}(\text{diff. prediction}|\text{nearest instance}),$$

as well as the conditional probability:

$$P_{diffC|diffF}(\text{diff. prediction}|\text{diff. value of } F \text{ and nearest instances}).$$

Finally, by using the Bayes rule, we obtain:

$$W[F] = \frac{P_{diffC|diffF} P_{diffF}}{P_{diffC}} - \frac{(1 - P_{diffC|diffF}) P_{diffF}}{1 - P_{diffC}} \tag{6}$$

The details of the RReliefF algorithm are given in pseudo-code form in Algorithm 1. The algorithm begins by selecting a random instance (R_i) and finding the k nearest instances I_j to it. From these instances, it then approximates the relevance $W[F]$ from Equation 6 of each feature by calculating N_{dC}, $N_{dF}[F]$ and $N_{dC\&dF}[F]$, described in lines 6,8 and 9 of Algorithm 1. The estimations of these values is based on the distance calculation in the feature space, $diff(F, R_i, I_j)$, (lines 8 and 9) and in the target space, $diff(\tau(\cdot), R_i, I_j)$, (lines 6 and 9).

Our original purpose is to extend the RReliefF algorithm for hierarchical multi-label classification problems. Considering that the HMC refers to the target space, we extend the RReliefF algorithm by changing the way that $diff(\tau(\cdot), R_i, I_j)$, from lines 6 and 9, is calculated. From Section 3 and Equation 1 we obtain:

$$diff(\tau(\cdot), R_i, I_j) = diff(S_i, S_j) = \sqrt{\sum_k w(c_k)(L_{i,k} - L_{j,k})^2} \tag{7}$$

where S_i and S_j are the target descriptions of R_i and I_j correspondingly, while $L_{i,k}$ and $L_{j,k}$ are their binary representations. In this way, by changing the way the distance is calculated, the original RReliefF algorithm is extended to work for HMC problems and we name this extension HMC-ReliefF.

Algorithm 1 Pseudocode for the RReliefF algorithm, taken from [28].

Input: for each training instance a vector of feature values $\mathbf{x}$ and predicted value $\tau(\mathbf{x})$
Output: the vector W of estimations of the relevance of features

set all N_{dC},$N_{dF}[F]$,$N_{dC\&dF}[F]$,$W[F]$ to 0
for $i = 1$ to m **do**
    randomly select an instance R_i
    select k instances I_j nearest to R_i
    for $j = 1$ to m **do**
        $N_{dC} = N_{dC} + \mathit{diff}(\tau(\cdot), R_i, I_j) \cdot d(i, j)$
        for $F = 1$ to f **do**
            $N_{dF}[F] = N_{dF}[F] + \mathit{diff}(F, R_i, I_j) \cdot d(i, j)$
            $N_{dC\&dF}[F] = N_{dC\&dF}[F] + \mathit{diff}(\tau(\cdot), R_i, I_j) \cdot \mathit{diff}(F, R_i, I_j) \cdot d(i, j)$
        end for
    end for
end for
for $F = 1$ to f **do**
    $W[F] = N_{dC\&dF}[F]/N_{dC} - (N_{dF}[F] - N_{dC\&dF}[F])/(m - N_{dC})$
end for

5. Experiments

In our experimental work, we investigate the HMC-ReliefF algorithm from two aspects: the algorithm's ability to find relevant features and the stability of the feature rankings it outputs. For both aspects, we employ a stepwise filter-like procedure [31], with which we examine various top-k subsets of the feature rankings allowing for a gradual analysis of the HMC-ReliefF properties. We examine the algorithm's ability to correctly place relevant feature on top of the ranking by iteratively constructing predictive models for various top-k ranked features, described in Section 5.1. For estimating the stability, we use the Canberra distance between the top-k ranked features produced for different algorithm parameters, described in Section 5.2. We give the specific details of the whole experimental work in Section 5.3.

5.1. Forward Feature Addition

Our experimental evaluation of the HMC-ReliefF is based on the intuition of what is the expected output of a good feature ranking algorithm. Namely, a "good" feature ranking algorithm would output the relevant features on top of the ranked list of features. A "bad" ranking algorithm would not necessarily be the one that gives an inverse ranking according to relevance, but the one that outputs a random ranking. In a random ranking, the expected distribution of relevant features should be uniform throughout the list.

Having this in mind, we employ a stepwise filter-like procedure [31] to evaluate our HMC-ReliefF algorithm. The idea is that starting from the ranked list of features, we construct classifiers for different numbers of top-k ranked features. If there are relevant features on top of the feature ranking, then we can construct a classifier that has a good predictive performance. If the ranking is random then the number of relevant features in the top-k ranked features is expected to be smaller.

Formally, if we have a feature ranking algorithm r that we use on a dataset $\mathcal{D}$, then the output would be a feature ranking $\mathbf{R}$, namely:

$$r(\mathcal{D}) \rightarrow \mathbf{R}.$$

The feature ranking $\mathbf{R}$ is defined as an ordered list of features F ($|F| = f$), more specifically:

$$\mathbf{R} = (F_{r1}, \ldots, F_{rj}, \ldots, F_{rf})$$

where:

$$rank(F_{r1}) \leq \cdots \leq rank(F_{rj}) \leq \cdots \leq rank(F_{rf})$$

If we assume that we can induce and evaluate a predictive model $\mathcal{M}(R_i, F_t)$, where $R_i \subseteq \mathbf{R}$ and F_t is a target feature, then our whole evaluation procedure can be described as in Algorithm 2.

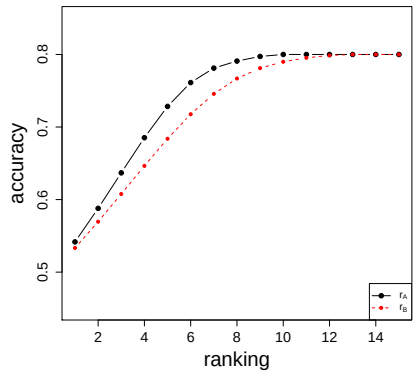


Fig. 2. FFA curves comparison example

For each step k of the filtering, i.e., for each subset of top-k ranked feature subsets, we induce a classification model and evaluate its performance. This process of generating feature sets from the feature ranking is performed in a forward manner, by adding more and more of the top ranked features, which we name *forward feature addition* (FFA). At

Algorithm 2 Stepwise evaluation of the top-k ranked features

Input: Feature Ranking, $\mathbf{R} = \{F_{r1}, \ldots, F_{rf}\}$; Target Feature, F_t
Output: FFA Curve, FFA, where $|FFA| = f$
  $\mathbf{R}_S \Leftarrow \emptyset$
  for $k = 1$ to f **do**
    $\mathbf{R}_S \Leftarrow \mathbf{R}_S \cup feature(\mathbf{R}, i)$
    $FFA[i] = eval(\mathcal{M}(\mathbf{R}_S, F_t))$
  end for
    return FFA

the end, we obtain a vector of model quality estimates that we can plot as a curve, thus obtaining a *FFA curve* that we use to estimate the performance of the feature ranking algorithm.

In order to say that the FFA curve of a certain feature ranking algorithm r_a is better than that of another ranking algorithm r_b, the model quality estimates of the ranking r_a must be larger than those of the models from the ranking r_b. Visually, this would mean that the FFA curve of the algorithm would be above the FFA curve of the random ranking, illustrated in Figure 2.

5.2. Stability of Feature Rankings

An important aspect of feature ranking algorithms is their stability or, more specifically, the stability of the ranked feature lists that they produce. The estimation of the stability of a feature ranking algorithm is intuitively similar to the analysis of stability of classification algorithms [38]. It is based on first inducing feature rankings, with the same algorithm, from different training instances and then comparing of these produced ranked lists.

In our experimental work, we use the Canberra distance as discussed by [16]. The Canberra distance [22, 23] is a weighted distance metric that puts bigger emphasis on the stability of the top ranked features. If we have two ranked lists R_A and R_B, where $\mathrm{R}(i)$ gives the rank of feature F_i, then the Canberra distance can be calculated as:

$$Ca(\mathrm{R}_A, \mathrm{R}_B) = \sum_{i=1}^{f} \frac{|\mathrm{R}_A(i) - \mathrm{R}_B(i)|}{\mathrm{R}_A(i) + \mathrm{R}_B(i)} \tag{8}$$

where f is the number of features.

In order for the distance to be applicable to partial rankings with $k < f$, the following adaptation is proposed:

$$Ca^{(k+1)}(\mathrm{R}_A, \mathrm{R}_B) = \sum_{i=1}^{f} \frac{|min\{\mathrm{R}_A(i), k+1\} - min\{\mathrm{R}_B(i), k+1\}|}{min\{\mathrm{R}_A(i), k+1\} + min\{\mathrm{R}_B(i), k+1\}} \tag{9}$$

where $Ca^{k+1} = Ca$.

Additionally, Jurman et al. [16] provide an analytical approximation of the expected Canberra distance between completely random rankings. The approximation is dependent only on the total number of features f and the size of the top-k subset that is considered. It is given by:

$$\hat{E}\{Ca^{(k+1)}\} = \frac{(k+1)(2f-k)}{f}\log(4) - \frac{2kf + 3f - k - k^2}{f} \tag{10}$$

For complete lists, the approximation becomes:

$$\hat{E}\{Ca^{(k+1)}\} = (\log(4) - 1)f + \log(4) - 2 \tag{11}$$

Finally, if the Canberra distance for partial rankings is normalized with the expected Canberra distance for each value of k, a normalized stability indicator is obtained, calculated as:

$$\hat{I} = \left\{ \left(k, \frac{Ca^{(k+1)}(\mathbf{R})}{\hat{E}\{Ca^{(k+1)}\}} \right) : 1 \leq k \leq f \right\} \tag{12}$$

With this adaptation, the stability indicator becomes independent of particular values for k and f, as it represents the relative change of distance between top-k lists w.r.t. the expected top-k distance. Therefore, the values of the stability index are directly comparable for datasets with different number of features f and between different top-k feature sets.

5.3. Experimental Setup

Here, we give a brief description of the datasets used in our experiments as well as the specific details used in our experimental setup. The general idea is to compare the feature rankings produced by the HMC-ReliefF algorithm with other baseline feature rankings, those produced by aggregated binary relevance and multiple random rankings. Also, we perform a stability analysis of the feature rankings output by the HMC-ReliefF algorithm.

Datasets Description We use datasets from two domains which have classes organised in a hierarchy. We use 5 datasets from 2 domains, more specifically: biology (SCOP-GO [4] and SCOP-FUN [4]) and image annotation/classification (Diatoms [8], ImCLEF07A [7] and ImCLEF07D [7]). The relevant properties that characterize each dataset are given in Table 1.

Table 1. Properties of the datasets with hierarchical targets; N_{tr} is the number of instances in the training dataset, D/C is the number of descriptive attributes (discrete/continuous), $|\mathcal{H}|$ is the number of classes in the hierarchy, $\mathcal{H}_d$ is the maximal depth of the classes in the hierarchy, $\overline{\mathcal{L}}$ is the average number of labels per example, and $\overline{\mathcal{L}_L}$ is the average number of leaf labels per example. Note that the values for $\mathcal{H}_d$ are not always a natural number because the hierarchy has a form of a DAG and the maximal depth of a node is calculated as the average of the depths of its parents.

Domain	N_{tr}	$\lvert D\rvert/\lvert C\rvert$	$\lvert\mathcal{H}\rvert$	$\mathcal{H}_d$	$\overline{\mathcal{L}}$	$\overline{\mathcal{L}_L}$
Diatoms	1098	0/200	107	2.0	1.98	0.98
ImCLEF07D	10006	0/80	46	3.0	3.0	1.0
ImCLEF07A	10006	0/80	96	3.0	3.0	1.0
SCOP-GO	9843	0/2003	572	5.5	6.26	0.95
SCOP-FUN	3097	0/2003	250	4.0	3.41	0.95

Both biological datasets concern the task of gene function prediction for *Arabidopsis Thaliana* by using Structural Classification of Proteins (SCOP) superfamily class predictions made by the Superfamily server [29]. The difference is in the type of the hierarchy of the gene functions: SCOP-GO has a hierarchy organised as a DAG (it uses Gene Ontology for the functional annotations), while SCOP-FUN has a tree-shaped hierarchy (it uses the FunCAT catalogue of gene functions).

Diatoms, ImCLEF07A and ImCLEF07D datasets concern the task of image annotation. For the Diatoms dataset the task is to correctly annotate microscopic images of

diatom organisms, while in the ImCLEF07A and ImCLEF07D datasets the task is to correctly annotate medical X-Ray images. In the three datasets, the images are described by extracting numeric features with state-of-the-art image descriptors, such as scale-invariant feature transform descriptors, local binary patterns, Fourier coefficients etc. For more details on all of the datasets, we refer the reader to the referenced literature.

HMC-ReliefF rankings In the HMC-ReliefF algorithm (Algorithm 1) and in Relief algorithms in general there are two basic parameters that influence the feature relevance estimation. These are the number of random instances m that are chosen and the number of nearest neighbours k that are used to calculate the feature relevance values. We explore a reasonable set of these parameter values for which we produce feature rankings for the datasets described in Section 5.3.

More specifically, we consider the following sets of parameters:

- $m = \{10, 50, 100, 250, 500\}$
- $k = \{10, 25, 50, 100\}$

Additionally, from the perspective of algorithm implementation, there is one parameter related to the number of random instances m. Namely, this is the seed s used for initializing the random generator that decides exactly which m data instances get selected. In order to provide a relevance estimate that is more statistically independent from the selected m random instances, we run the algorithm 10 times for each of the m and k values. Specifically, we use values of s ranging from 1 to 10.

In summary, for our experiments we produce feature rankings for each combination of values of m, k and s that we denote with $\mathrm{R}^{hmc}_{m,k,s}$. However, based on the stability results analysis from Section 6, for the further experimental work and comparisons, we use averaged rankings from the different values of the seed s, namely: $\mathrm{R}^{hmc}_{m,k} = average_s\left(\mathrm{R}^{hmc}_{m,k,s}\right)$.

Baseline Rankings The first baseline for comparison of the HMC-ReliefF rankings are the aggregated binary relevance rankings. In order to obtain these, we first split the original HMC datasets described in Section 5.3, into separate datasets containing a single binary class. Each binary class c_i is one of the classes from the original target hierarchy.

Next, for each of the newly produced binary datasets, we run the original ReliefF algorithm for binary classification and we produce binary relevance (BR) rankings. The BR rankings are produced for the same values of m and k as the HMC-ReliefF rankings, thus obtaining $\mathrm{R}^{br}_{m,k,c_i}$ rankings. For each dataset, these binary relevance rankings are then aggregated by averaging the feature rankings for all of the classes, namely: $\mathrm{R}^{br}_{m,k} = average_{c_i}\left(\mathrm{R}^{hmc}_{m,k,c_i}\right)$

As an additional baseline for our comparisons, we also use a set of 50 random rankings for each different dataset, R^{rand}_i. For each of these random rankings, we construct an FFA curve as described in Section 5.1 and generate a separate FFA curve. For the random rankings, we average the results of the 50 individual FFA curves, thus generating an expected FFA curve for a given dataset.

Stability and FFA curves We perform a stability analysis of the feature rankings produced by the HMC-ReliefF algorithm, just with respect to a single parameter, namely the variable random seed s. Computationally, it is expensive to estimate the feature relevance by setting the value of m equal to the number of instances in a dataset N and usually $m << N$. However, depending on the dataset in question the value of m should be set so that the relevance estimates are least variable w.r.t. to the seed s, i.e. the specific random m instances that are used.

For the stability analysis we use the Canberra distance described in Section 5.2. We calculate the stability of the rankings by measuring the rank distances between the rankings produced for fixed values of m and k (given previously) but a variable s. Namely, we calculate the stability indicator for each m and k as follows:

$$\hat{I}_{m,k,i} = \left\{ \frac{Ca^{(k+1)}(\mathrm{R}^{hmc}_{m,k,s})}{\hat{E}\{Ca^{(k+1)}\}} \right\} \tag{13}$$

where $i = \{1 \ldots f\}$ and $s = \{1 \ldots 10\}$.

For constructing the FFA curves, as a predictive model that we induce and evaluate, we use random forests of predictive clustering trees for hierarchical multi-label classification (PCT-HMCs)[40, 18]. The specific parameters that we used for the random forests of PCTs were 100 trees and a variable feature subset size of depending on the size of the considered top-k feature subsets. For estimating the PCT-HMCs performance, we split the datasets on train and test data.

In the HMC context, there are various error measures that can be considered. We use the area of a variant of a precision-recall curve, namely the Pooled Area Under the Precision-Recall Curve ($AU(\overline{PRC})$). For this measure, the precision ($\overline{Prec}$) and recall ($\overline{Rec}$) are micro averaged for all classes from the hierarchy as follows:

$$\overline{Prec} = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FP_i} \tag{14}$$

and

$$\overline{Rec} = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FN_i} \tag{15}$$

where i ranges over all classes. In this context, $\overline{Prec}$ corresponds to the proportion of predicted labels that are correct, while $\overline{Rec}$ corresponds to the proportion of labels in the data that are correctly predicted. By varying the threshold, we then obtain an average PR curve. The area under this curve is then the Pooled Area Under the Precision-Recall Curve ($AU(\overline{PRC})$). More details about the evaluation measures can be found in [40].

6. Results and Discussion

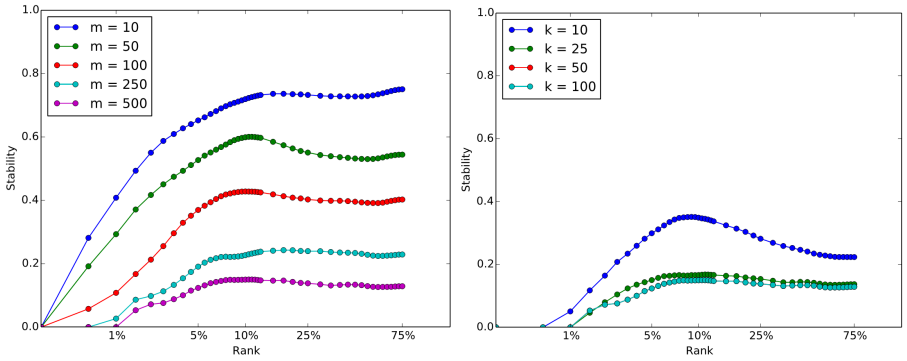
In this section, we present and discuss the results from the experimental evaluation of the proposed methods. We first give the results on the stability of the obtained rankings. We then compare our method with the random ranking and the baseline.

6.1. Stability

The stability of the HMC-ReliefF algorithm was calculated by using different random seeds for sampling the instances. In other words, we investigated the sensitivity of the algorithm of the choice of instances. We obtained two groups of results. The first group of results contains graphs for different feature rankings obtained with different sizes of the neighbourhood (k), while the second contains graphs for different number of sampled instances (m). The total number of graphs per dataset is 9 (4 for the different values of k and 5 for the different values of m).

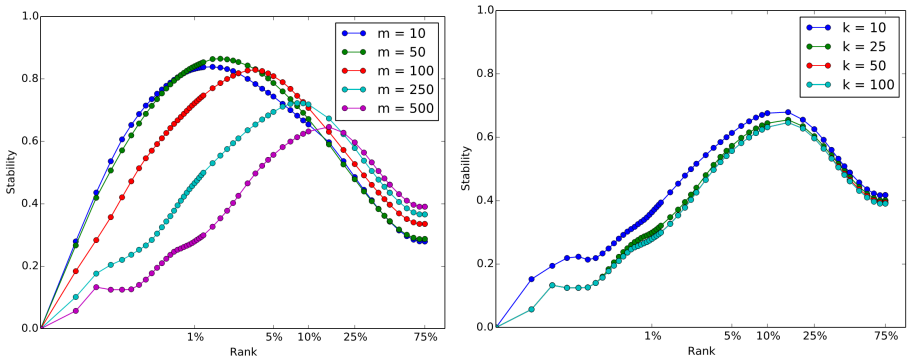
In Figures 3 and 4, we illustrate the stability of the rankings produced with HMC-ReliefF on the Diatoms and SCOP-GO, respectively. We discuss these four graphs in more detail. First, we focus on the stability of the rankings with different values for m with the value of k set to 100. For the Diatoms dataset these stability curves are given in Figure 3(a). By increasing the value of m, i.e., by increasing the number of sampled instances, the stability of the rankings clearly improves. This finding is also valid for the other values of k. Moreover, similar graphs are obtained for the ImageCLEF07A and ImageCLEF07D datasets. For the SCOP-GO dataset the stability curves for fixed m are given in Figure 4(a). Here, the stability of the ranking improves with the increase of the number of sampled instances mainly at the top of the ranking. Note that this is acceptable behaviour since we are more interested to get stable and relevant features on the top of the ranking. Similar graphs are obtained for the other values of k and for the SCOP-FUN dataset.

We next discuss the stability curves of the rankings with different neighbourhood sizes (i.e., varying k) and the number of sampled instances (m) set to 500. First, for the Diatoms dataset, the stability curves are given in Figure 3(b), while for the SCOP-GO dataset in Figure 4(b). For the curves in the both graphs, we can note that by increasing k the stability improves. The improvement is more noticeable for the Diatoms dataset than for the Scop_ara_GO dataset. Although, the rankings are not that much sensitive to the



(a) (b)

Fig. 3. The stability of the HMC-ReliefF algorithm for the Diatoms dataset. (a) stability of the ranking for a neighbourhood of 100 instances ($k = 100$) and a variable number of sampled instances (m) and (b) stability of the ranking for a 500 sampled instances ($m = 500$) and a variable size of the neighbourhood (k)



(a) (b)

Fig. 4. The stability of the HMC-ReliefF algorithm for the SCOP-GO dataset. (a) stability of the ranking for a neighbourhood of 100 instances ($k = 100$) and a variable number of sampled instances (m) and (b) stability of the ranking for a 500 sampled instances ($m = 500$) and a variable size of the neighbourhood (k)

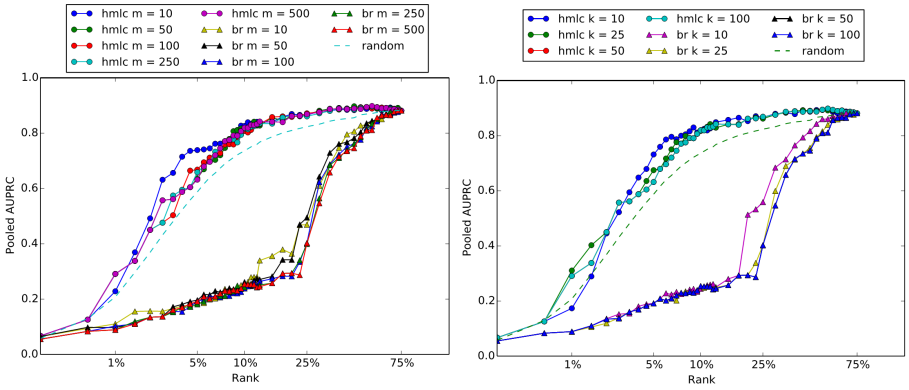
change of k, for the values of k larger than 10 (i.e., with $k \geq 25$, the rankings have similar stability curves). Similar conclusions are valid for the other values of m and for the other datasets.

Overall, the calculated stability curves show that the HMC-ReliefF algorithm produces feature rankings with good stability. To begin with, the stability improves with the increase of the number of sampled instances. Moreover, by sampling 500 instances, the obtained rankings are quite stable. In other words, the algorithm is not much sensitive on the random sampling of instances, as long as we sample large enough number of them. Next, the algorithm is not much sensitive on the size of the neighbourhood: Already with 25 neighbours the obtained rankings are stable.

6.2. FFA curves

In this section, we compare the performance of HMC-ReliefF first with random rankings and then with the baseline feature ranking algorithm - binary relevance of ReliefF rankings. The FFA curves for selected values of the number of sampled instances m and the neighbourhood size k are given in Figures 5, 6, 7 and 8. The graphs on the left-hand side in the Figures (sub-figures (a)) represent the FFA curves for a fixed value of m, while the value of k is varied. Correspondingly, the graphs on the right-hand side (sub-figures (b)) contain FFA curves for a fixed value of k, while the value of m is varied.

Overall, it can be observed that all of the FFA curves of the HMC-ReliefF algorithm are most of the time above the FFA curves of the random rankings. This means that at the top of the rankings produced by HMC-ReliefF, for different settings of m and k, relevant features can be found. It also means that this is not by chance, as the $AU(\overline{PRC})$ of the produced models is larger than the expected value of a random ranking. However, there are differences in the obtained curves for the different datasets, which we will discuss in detail.



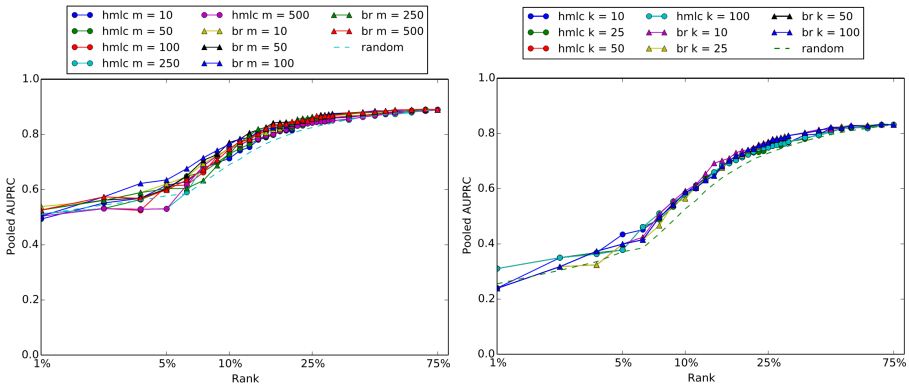
(a) (b)

Fig. 5. Comparison of FFA curves obtained from the HMC-ReliefF algorithm, random ranking and the BR ranking for the Diatoms dataset. (a) FFA curve for a ranking using a neighbourhood of 100 instances ($k = 100$) and a variable number of sampled instances (m) and (b) FFA curve for a ranking using 500 sampled instances ($m = 500$) and a variable size of the neighbourhood (k)

We first consider the FFA curves for the Diatoms dataset, given in Figure 5. It can be noticed that all of the FFA curves produced by HMC-ReliefF, are only slightly higher, i.e., are only slightly better, than the expected FFA curves of the random rankings. Also, there is no great variability of the FFA curves with respect to the different number of m and k. This is expected if we take into account this specific domain and the way the features are produced. Namely, most of the features are image descriptors, which are informative about the image and most of them are relevant. This can also be concluded if we observe just the expected FFA curve of the random rankings. Furthermore, the rankings obtained with HMC-ReliefF are better than the ones obtained with BR. The FFA curves for the BR rankings are even below the FFA curve for the random curves. This leads us to believe that the BR method produces an inverse ranking in this specific case, i.e., puts the more relevant features at the end of the ranking. Similarly as for the HMC-ReliefF, the FFA curves for the BR rankings showed no great variability with respect to the different number of m and k.

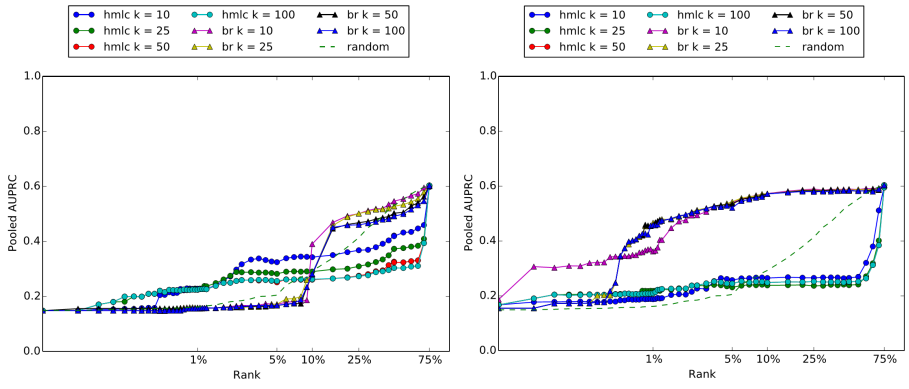
Second, we discuss the FFA curves for the ImageCLEF07A and ImageCLEF07D datasets, given in Figure 6. In this case, all of the FFA curves for the HMC-ReliefF, BR and the random ranking are very much close to each other. The FFA curves for the both methods show no variability with respect to the values of m and k. Similarly as for the Diatoms dataset, this behaviour is somewhat expected because the features here are image descriptors and are informative about the image.

Next, we consider the FFA curves for the SCOP-FUN dataset, given in Figure 7. For this dataset, we show only the FFA curves with varying k and fixed m, because they offer the most insightful information about the performance of the methods on this dataset. These curves require a more complex interpretation. First, the FFA curves of the HMC-ReliefF are above the FFA curves of the random ranking at the beginning of the ranking. After adding 10% of the features, the FFA curves of HMC-ReliefF goes below the FFA curve of the random ranking, i.e., seemingly irrelevant or redundant features are added.



(a) (b)

Fig. 6. Comparison of FFA curves obtained from the HMC-ReliefF algorithm, random ranking and the BR ranking for the ImageCLEF datasets. (a) FFA curve for a ranking on ImageCLEF07D using a neighbourhood of 100 instances ($k = 100$) and a variable number of sampled instances (m) and (b) FFA curve for a ranking on ImageCLEF07A using 10 sampled instances ($m = 10$) and a variable size of the neighbourhood (k)



(a) (b)

Fig. 7. Comparison of FFA curves obtained from the HMC-ReliefF algorithm, random ranking and the BR ranking for the SCOP-FUN dataset. (a) FFA curve for a ranking using 10 sampled instances ($m = 10$) and a variable size of the neighbourhood (k) and (a) FFA curve for a ranking using 500 sampled instances ($m = 500$) and a variable size of the neighbourhood (k)

This is until 75% of the features are added. After this point there is a jump in the number of relevant features that are added, as the $AU(\overline{PRC})$ values become larger. Upon a closer inspection of the dataset, we notice that all of the attributes are declared with type numeric, while their values are set either to 10 (for the majority of the instances) or to a small value (typically smaller than $1 \cdot 10^{-5}$). This effect coupled together with the sparsity of the annotation hierarchy (tree-shaped hierarchy of gene functions - FunCAT catalogue of gene functions) probably contributed to such a disturbance in the relevance estimates. Moreover, the ReliefF family of algorithms suffers from an underestimation of numeric attributes [28]. To alleviate this issue, the use of a ramp function was proposed when calculating the distance between the numerical attributes. In our implementation, a ramp function was also used, however different threshold parameters of this function were not explored. Robnik-Šikonja and Kononenko [28] noted that for different domains, different thresholds might be appropriate. Note that both HMC-ReliefF and BR ranking exhibit similar behaviour on this dataset.

Let us now focus on the FFA curves of the methods. First, the FFA curves of HMC-ReliefF are not sensitive to changes of m and k. On the other hand, the FFA curves for the BR ranking are sensitive to the changes of m and k: Best performance of BR ranking is obtained with a large number of sampled instances (a large m) and by considering small neighbourhoods (a small k). This is consistent with the analysis of ReliefF in [28] where it is stated that the values of m and k are often problem dependent and often smaller values might be better in order to preserve "locality" of the relevance estimations. BR ranking is able to profit on the small neighbourhood information and obtain good feature ranking. BR ranking yields better FFA curves than the HMC-ReliefF for a large value of m and small value of k (Figure 7(b)), while for a small value of m, HMC-ReliefF is better than BR ranking at the top of the ranking (Figure 7(a)).

Finally, the best results were obtained for the SCOP-GO dataset, which we present in Figure 8. Both for a fixed m and k, the values of the FFA curves produced by HMC-ReliefF are much higher than those of the random rankings. Moreover, the FFA curves show that the ranking produced with HMC-ReliefF is better than the ranking produced with BR ranking: the FFA curves of the HMC-ReliefF are above the FFA curves of the BR ranking. Next, the FFA curves for the HMC-ReliefF rankings are not sensitive to the changes of m and k, while the FFA curves for the BR ranking seem to prefer a larger value of m.

7. Conclusions and Further Work

In this paper, we presented the HMC-ReliefF algorithm, which is an extension of the RReliefF algorithm for the task of Hierarchical Multi-label Classification. We believe that this is both an interesting and novel line of work, in the context of feature ranking algorithms. To the best of our knowledge, there has not been any work for feature ranking within the context of structured data. We specifically focused on the ReliefF algorithm, due to its success in both classification and regression settings. The specific type of structured problems that we considered (HMC), was motivated by the fact that this kind of data can be found in various domains including biology and image annotation.

We evaluated the HMC-ReliefF algorithm on datasets from two domains and with different properties of the hierarchies. We first evaluated the stability of the rankings pro-

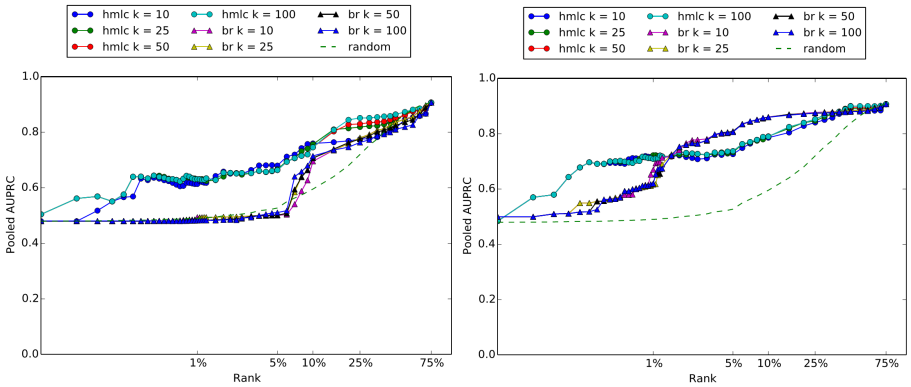


Fig. 8. Comparison of FFA curves obtained from the HMC-ReliefF algorithm, random ranking and the BR ranking for the SCOP-GO dataset. (a) FFA curve for a ranking using 10 sampled instances ($m = 10$) and a variable size of the neighbourhood (k) and (a) FFA curve for a ranking using 500 sampled instances ($m = 500$) and a variable size of the neighbourhood (k)

duced by the HMC-ReliefF algorithm. We then investigated if our algorithm is able to detect relevant features in a dataset and put them on top of the ranking. We consider this to be a minimum requirement of any feature ranking algorithm. Additionally, we also explored a reasonable set of parameter settings of HMC-ReliefF, which have influence on the feature relevance estimations. Finally, we compared the performance of HMC-ReliefF with a feature ranking algorithm based on binary relevance – a method typically used for solving the task of feature ranking for (hierarchical) multi-label classification.

The analysis of the stability of the produced rankings showed that the HMC-ReliefF algorithm produces feature rankings with good stability. The stability of the produced rankings improves with the increase of the number of sampled instances (the m parameter of the HMC-ReliefF algorithm). Conversely, the stability of the produced rankings is not much sensitive on the size of the neighbourhood (the k parameter of the algorithm).

Next, the results of our experiments showed that, for various datasets, the HMC-ReliefF algorithm performed well, as evaluated by a stepwise filter like approach of constructing FFA curves. This performance was compared to an expected FFA curve, obtained from a set of random rankings, and to the FFA curves of the baseline BR method.

The exploration of the various parameters of HMC-ReliefF showed the following. On one hand, the FFA curves for the HMC-ReliefF are not much sensitive to changes of the number of sampled instances and the neighbourhood size. On the other hand, the FFA curves for the BR method are sensitive on the parameter values and these values vary for different datasets. Next, on the majority of the datasets, the HMC-ReliefF rankings produced better FFA curves than the competing BR method. For some specific settings for m and k, the BR method was able to produce better FFA curves (thus better ranking) than HMC-ReliefF method.

For the image annotation datasets (Diatoms, ImageCLEF07A and ImageCLEF07D), the FFA curves of the HMC-ReliefF were just slightly above the FFA of the random rankings. This was due to the nature of the domain and due to the fact that most of the

features in the image annotation datasets were relevant. The FFA curve of the BR method for the Diatoms dataset was bellow the FFA curve of the random ranking. This leads us to believe that the BR method puts the more relevant features at the end of the ranking.

For the functional genomics datasets (SCOP-FUN and SCOP-GO), the results were more complex. First, the FFA curves of the HMC-ReliefF are above the FFA curves of the random ranking at the beginning of the ranking, while as features are being added the FFA curve drops bellow the random curve. This effect is due to some specific properties of the attributes, sparsity of the target hierarchy and the underestimation of the numeric attributes. Next, the BR method requires a larger value for m. For the SCOP-FUN dataset, the HMC-ReliefF is worse than the BR only for large values of m and small values of k, while for the SCOP-GO dataset HMC-ReliefF is better across the whole range of parameter values.

The directions for further work regarding our HMC-ReliefF algorithm are numerous. One major direction would be to define an artificial, controlled setting for investigating HMC problems in the context of feature ranking. Different types of hierarchies should be considered, which are also differently structured (balanced vs. unbalanced, different width, different depth), or differently populated by instances (sparse vs. non-sparse). Within this setting, the effects of the various parameters of HMC-ReliefF can be investigated and the advantages and limitations of the algorithm can be further explored. Another major direction is to consider different types of structured outputs, such as multi-label classification, multi-target classification and multi-target regression.

Acknowledgments. We would like to acknowledge the support of the European Commission through the project MAESTRA – Learning from Massive, Incompletely annotated, and Structured Data (Grant number ICT-2013-612944) and the Human Brain Project (Grant number 604102), and the support of the Spanish Ministry of Economy and Competitiveness, "Centro de Excelencia Severo Ochoa 2013-2017", SEV-2012-0208.

References

1. Aleksovski, D., Kocev, D., Džeroski, S.: Evaluation of distance measures for hierarchical multi-label classification in functional genomics. In: ECML/PKDD 2009 Workshop on Learning from Multi-Label Data. pp. 5–16 (2009)
2. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nature genetics 25(1), 25–29 (2000)
3. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. Artificial Intelligence 97, 245–271 (1997)
4. Clare, A.: Machine learning and data mining for yeast functional genomics. Ph.D. thesis, University of Wales Aberystwyth, Aberystwyth, Wales, UK (2003)
5. Deselaers, T., Deserno, T.M., Mller, H.: Automatic medical image annotation in ImageCLEF 2007: Overview, results, and discussion. Pattern Recognition Letters 29(15), 1988 – 1995 (2008)
6. Dietterich, T.G., Domingos, P., Getoor, L., Muggleton, S., Tadepalli, P.: Structured machine learning: the next ten years. Machine Learning 73(1), 3–23 (2008)

7. Dimitrovski, I., Kocev, D., Loskovska, S., Džeroski, S.: Hierchical annotation of medical images. In: Proceedings of the 11th International Multiconference - Information Society IS 2008. pp. 174–181. IJS, Ljubljana (2008)
8. Dimitrovski, I., Kocev, D., Loskovska, S., Džeroski, S.: Hierarchical classification of diatom images using ensembles of predictive clustering trees. Ecological Informatics 7(1), 19–29 (2012)
9. Doquire, G., Verleysen, M.: Feature Selection for Multi-label Classification Problems. In: Advances in Computational Intelligence. pp. 9–16 (2011)
10. Esuli, A., Fagni, T., Sebastiani, F.: Treeboost.mh: A boosting algorithm for multi-label hierarchical text categorization. In: String Processing and Information Retrieval – LNCS 4209. pp. 13–24 (2006)
11. Gärtner, T., Vembu, S.: On structured output training: hard cases and an efficient alternative. Machine Learning 76, 227–242 (2009)
12. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (March 2003)
13. He, Z., Yu, W.: Stable feature selection for biomarker discovery. Computational Biology and Chemistry 34, 215–225 (2010)
14. Jaccard, P.: Étude comparative de la distribution florale dans une portion des alpes et des jura. Bulletin de la Société Vaudoise des Sciences Naturelles (37), 547–579 (1901)
15. Jia, M., Zheng, D., Yang, B., Chen, Q.: Hierarchical text categorization based on multiple feature selection and fusion of multiple classifiers approaches. In: 6th International Conference on Fuzzy Systems and Knowledge Discovery – FSKD '09. vol. 1, pp. 192–196 (2009)
16. Jurman, G., Merler, S., Barla, A., Paoli, S., Galea, A., Furlanello, C.: Algebraic stability indicators for ranked lists in molecular profiling. Bioinformatics 24, 258–264 (2008)
17. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: ML92: Proceedings of the 9th international workshop on Machine learning. pp. 249–256. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1992)
18. Kocev, D., Vens, C., Struyf, J., Džeroski, S.: Tree ensembles for predicting structured outputs. Pattern Recognition 46(3), 817–833 (2013)
19. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1-2), 273–324 (1997)
20. Kononenko, I.: Estimating attributes: Analysis and extensions of relief. In: European Conference on Machine Learning. pp. 171–182 (1994)
21. Kriegel, H.P., Borgwardt, K., Kröger, P., Pryakhin, A., Schubert, M., Zimek, A.: Future trends in data mining. Data Mining and Knowledge Discovery 15, 87–97 (2007)
22. Lance, G.N., Williams, W.T.: Computer programs for hierarchical polythetic classification ('similarity analyses'). The Computer Journal 9, 60–64 (May 1966)
23. Lance, G.N., Williams, W.T.: Mixed-Data Classificatory Programs I - Agglomerative Systems. Australian Computer Journal 1, 15–20 (1967)
24. Peng, X., Ming, Z., Wang, H.: Text learning and hierarchical feature selection in webpage classification. In: Advanced Data Mining and Applications – LNCS 5139. pp. 452–459 (2008)
25. Pesquita, C., Faria, D., Bastos, H., Falcao, A.O., Couto, F.: Evaluating go-based semantic similarity measures. In: BioOntologies SIG at ISMB/ECCB - 15th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB) (2007)
26. Read, J., Pfahringer, B., Holmes, G.: Multi-label Classification Using Ensembles of Pruned Sets. In: Proc. of the 8th IEEE International Conference on Data Mining. pp. 995–1000 (2008)
27. Robnik-Šikonja, M., Kononenko, I.: An adaptation of relief for attribute estimation in regression. In: Fisher, D.H. (ed.) ICML. pp. 296–304. Morgan Kaufmann (1997)
28. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of ReliefF and RReliefF. Machine Learning 53, 23–69 (2003)

29. Schietgat, L., Vens, C., Struyf, J., Blockeel, H., Kocev, D., Džeroski, S.: Predicting gene function using hierarchical multi-label decision tree ensembles. BMC Bioinformatics 11(2), 1–14 (2010)
30. Silla, C., Freitas, A.: A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery 22(1-2), 31–72 (2011)
31. Slavkov, I.: An Evaluation Method for Feature Rankings. Ph.D. thesis, IPS Jožef Stefan, Ljubljana, Slovenia (2012)
32. Slavkov, I., Karcheska, J., Kocev, D., Kalajdziski, S., Džeroski, S.: Extending ReliefF for hierarchical multi-label classification. In: Proc. of International Workshop on New Frontiers in Mining Complex Patterns. pp. 156–167 (2013)
33. Slavkov, I., Karcheska, J., Kocev, D., Kalajdziski, S., Džeroski, S.: ReliefF for Hierarchical Multi-label Classification. In: New Frontiers in Mining Complex Patterns – LNCS 8399. pp. 148–161. Springer International Publishing (2014)
34. Spolaôr, N., Cherman, E.A., Monard, M.C., Lee, H.D.: A comparison of multi-label feature selection methods using the problem transformation approach. Electronic Notes in Theoretical Computer Science 292, 135 – 151 (2013)
35. Spolaor, N., Cherman, E.A., Monard, M.C., Lee, H.D.: ReliefF for Multi-label Feature Selection. In: Proc. of the Brazilian Conference on Intelligent Systems. pp. 6–13 (2013)
36. Spolaor, N., Lee, H.D., Takaki, W.S.R., Wu, F.C.: Feature selection for multi-label learning: A systematic literature review and some experimental evaluations. International Journal of Computational Intelligence Systems 8, 3–15 (2015)
37. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data. In: Data Mining and Knowledge Discovery Handbook, pp. 667–685. Springer Berlin / Heidelberg (2010)
38. Turney, P.D.: Technical note: Bias and the quantification of stability. Machine Learning 20, 23–33 (1995)
39. Vateekul, P.: Hierarchical multi-label classification: Going beyond generalization trees. Ph.D. thesis, University of Miami, Coral Gables, Florida, US (2012)
40. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Machine Learning 73(2), 185–214 (2008)
41. Wang, Q., Guan, Y., Wang, X., Xu, Z.: A novel feature selection method based on category information analysis for class prejudging in text classification. International Journal of Computer Science and Network Security 6(1A), 113–119 (2006)
42. Wibowo, W., Williams, H.E.: Simple and accurate feature selection for hierarchical categorisation. In: Proceedings of the 2002 ACM Symposium on Document Engineering. pp. 111–118 (2002)

Ivica Slavkov graduated in Electrical Engineering at the University of Ss. Cyril and Methodius in Skopje, Macedonia. He worked as a researcher and obtained his PhD in computer science at the Jozef Stefan Institute in Ljubljana, Slovenia. His main focus of work was in the area of machine learning and data mining with applications in biology. Currently, he works as a postdoctoral researcher at the Centre of Genomic Regulation in Barcelona, Spain. He is working in the Systems Biology Programme, in the area of swarm robotics with focus on multi-agent modelling of morphogenesis.

Jana Karcheska is a senior software engineer at Netcetera, Skopje. She has more than 10 years of experience working mainly with web technologies. Her expertise includes working on complex safety critical and mission critical systems. Her other interests are machine learning, bioinformatics and functional programming.

Dragi Kocev is a researcher at the Department of Knowledge Technologies, JSI. He completed his PhD in 2011 at the Jozef Stefan International Postgraduate School in Ljubljana on the topic of learning ensemble models for predicting structured outputs. He was a visiting research fellow at the University of Bari, Italy in 2014/2015. His research interests are in the field of data mining and includes the study, development and application of data mining algorithms. His current research is aimed towards further development of efficient methods for learning from data with structured outputs (e.g., predicting multiple targets, hierarchical multi-label classification...) and their applications in machine vision, life sciences and ecological modelling. He has participated in several national Slovenian projects, the EU funded projects IQ and PHAGOSYS and is involved in the Human Brain Project. He was co-coordinator of the FP7 FET Open project MAESTRA.

Sašo Džeroski received his Ph.D. degree in computer science from the University of Ljubljana in 1995. He is a scientific councilor (senior researcher) at the Department of Knowledge Technologies, Jozef Stefan Institute, Ljubljana. He is also affiliated with the Centre of Excellence for Integrated Approaches in Chemistry and Biology of Proteins (scientific councilor) and the Jozef Stefan International Postgraduate School (full professor), both in Ljubljana, Slovenia. His research interests fall in the field of artificial inteligence (AI), focusing on the development of data mining and machine learning methods for a variety of tasks - including the prediction of structured outputs and the automated modeling of dynamic systems - and their applications to practical problems from science and engineering, e.g., environmental sciences (ecology) and life sciences (biomedicine). In 2008, he was elected fellow of the European AI Society for his ”Pioneering Work in the field of AI and Outstanding Service for the European AI community”. In 2015, he became a foreign member of the Macedonian Academy of Sciences and Arts. In 2016, he was elected a member of Academia Europaea.

Received: January 15, 2017; Accepted: September 12, 2017.