Extending ReliefF for Hierarchical Multi-label Classification ⋆

Ivica Slavkov[1], Jana Karcheska[2], Dragi Kocev[1], Slobodan Kalajdziski[2], and Sašo Džeroski[1]

[1] Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
[2] Ss. Cyril and Methodius University, Faculty of Computer Science and Engineering, Skopje, Macedonia
{ivica.slavkov,dragi.kocev,saso.dzeroski}@ijs.si,
j.karceska@gmail.com,
slobodan.kalajdziski@finki.ukim.mk

Abstract. In the recent years, the data available for analysis in machine learning is becoming very high-dimensional and also structured in a more complex way. This emphasises 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 extending a feature ranking algorithm that can be used as a filter method for specific type of structured data. More specifically, we adapt the RReliefF algorithm for regression, for the task of hierarchical multi-label classification (HMC). We evaluate this algorithm experimentally in a filter-like setting by employing PCTs for HMCs as a classifier and we consider datasets from various domains. The results show that HMC-ReliefF can identify the relevant features present in the data and produces a ranking where they are among the top ranked.

Keywords: feature selection, feature ranking, feature relevance, structured data, hierarchical multi-label classification, 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 [8]. On the other, mining complex data and extracting knowledge from it has been identified as one of the most challenging problems in machine learning [4], [13].

For dealing with the high-dimensionality of the data various feature selection methods exist. They usually precede the induction of predictive models and can be classified as filter, wrapper and embedded methods [7]. Filter methods [2] are the simplest ones and they usually involve a feature ranking algorithm that produces a list of relevant

⋆ The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint first authors.

features. Wrapper methods [11] rely on classification algorithms to perform feature selection and are computationally expensive. Embedded methods [7] 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 [19], SVMs for multi-label and hierarchical multi-label problems [6], as well as tree ensembles that can be additionally employed for vectors of multiple targets [10].

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 data trends. So far, structured prediction has not been considered in the context of a feature ranking method and we consider this an novel and interesting line of work to pursue.

More specifically, we focus on the ReliefF [16] algorithm for feature ranking. This algorithm is instance based and it works in a very intuitive fashion, making it relatively easy to extend. Its theoretical properties have been extensively explored [16] and it is very successful in detecting relevant feautres in a dataset.

We extend ReliefF for a specific type of structured problems, namely those from the Hierarchical Multi-Label Classification (HMC) domain [17]. 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.

This type of problems appear in different domains, for example in biology for the task of gene function prediction. Namely, for this task, 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 [1]. Predicting the function of a certain gene, would have to take into account the multi-label annotation of each gene and also the hierarchical connections of these labels.

Considering this, we present the details of our work in the rest of this paper, organised as follows. In Section 2 we define more formally the HMC setting and present the distance measures appropriate for this setting. Then, in Section 3, we discuss in depth the original RReliefF algorithm for regression and explain our HMC-ReliefF extension of the algorithm. We present our experimental evaluation of the proposed HMC-ReliefF algorithm in Section 4. At the end, in Section 5, we present our conclusion and discuss further directions of work.

2 Hierarchical Multi-label Classification

As previously discussed, in our work we extend the ReliefF algorithm for the task of hierarchical multi-label classification (HMC). Hierarchical classification differs from traditional classification that 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 hierarchical multi-label classification (HMC) [19], [17].

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., $\forall X_i \in X, X_i = (x_{i_1}, x_{i_2}, ..., x_{i_D})$, where D is the size of the tuple (or number of descriptive variables),
- 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 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 ($N = |E|$)

Calculating the distance between two different instances of the target space S_1 and S_2, can be done in different ways. In [19] the hierarchy of labels is represented as a vector of binary values. The vector is created by traversing the tree or DAG that is representing the hierarchy in pre-order and assigning a 0 or 1 sequentially in the vector for a missing or present label respectively.

This representation allows two hierarchies of labels for two instances to be compared by simply comparing two binary vectors. In our HMC-ReliefF algorithm we use a weighted Euclidean distance measure given with the following equation:

$$d(v_1, v_2) = \sqrt{\sum_i w(c_i)(v_{1,i} - v_{2,i})^2}, \tag{1}$$

where v_1 and v_2 are the binary vector representation of S_1 and S_2 respectively.

For example, consider the toy class hierarchy shown in Figure 1(a,b), and two data examples: (X_1, S_1) and (X_2, S_2) that belong to the classes $S_1 = \{c_1, c_2, c_{2.2}\}$ (boldface in Figure 1(b)) and $S_2 = \{c_2\}$, respectively. We use a vector representation with consecutive components representing membership of class c_1, c_2, $c_{2.1}$, $c_{2.2}$ and c_3, in that order (preorder traversal of the tree of class labels). The distance is then calculated as follows:

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

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 bigger influence on the total distance.

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 exists and with a different depth. This problem is solved with the following recursive equation:

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

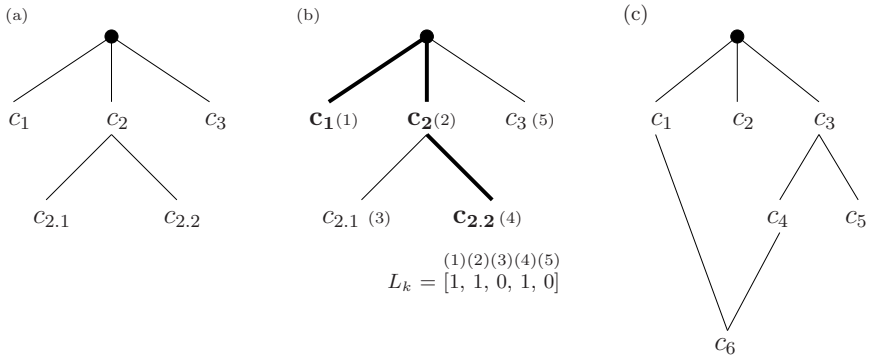


Fig. 1: Toy examples of hierarchies structured as a tree and a DAG.(a) Class label names contain information about the position in the hierarchy, e.g., $c_{2.1}$ is a subclass of c_2. (b) The set of classes $S_1 = \{c_1, c_2, c_{2.2}\}$, shown in bold in the hierarchy, represented as a vector (L_k).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

By using this weighting function, the weight of the different possible parents is averaged and in [19] this is recommended as a good way to take into account multiple inheritance which occurs in DAGs.

3 HMC-ReliefF Algorithm

The Relief family of algorithms are instance based methods for estimating the feature relevance. The original Relief algorithm was proposed in [9] and is limited to two-class classification problems. The algorithm was extended in [12] to deal with multi-class problems. The extension was named ReliefF. Later, it was also adapted for regression problems [15] and named RReliefF.

In general, the feature relevance value awarded by the Relief algorithm is an approximation of the following difference of probabilities [12]:

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

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

Algorithm 1 Pseudocode for the RReliefF algorithm, taken from [16].

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

1: set all N_{dC},$N_{dF}[F]$,$N_{dC\&dF}[F]$,$W[F]$ to 0
2: **for** $i = 1$ to m **do**
3: randomly select and instance R_i
4: select k instances I_j nearest to R_i
5: **for** $j = 1$ to m **do**
6: $N_{dC} = N_{dC} + diff(\tau(\cdot), R_i, I_j) \cdot d(i, j)$
7: **for** $F = 1$ to f **do**
8: $N_{dF}[F] = N_{dF}[F] + diff(F, R_i, I_j) \cdot d(i, j)$
9: $N_{dC\&dF}[F] + diff(\tau(\cdot), R_i, I_j) \cdot diff(F, R_i, I_j) \cdot d(i, j)$
10: **end for**
11: **end for**
12: **end for**
13: **for** $F = 1$ to f **do**
14: $W[F] = N_{dC\&dF}[F]/N_{dC} - (N_{dF}[F] - N_{dC\&dF}[F])/(m - N_{dC})$
15: **end for**

translate the HMC problem to a multi- class one, therefore ignoring the hierarchical aspect. Having in mind that the definitions of the HMC distances in Section 2 are actually weighted Euclidean, they are more suited to be included in the RReliefF algorithm, originally designed for regression.

The details of the RReliefF algorithm are given in pseudocode 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 4 of each feature in the following way.

First, we introduce the notation:

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

and then by using the Bayes rule, we have:

$$W[F] = \frac{P_{diffC|diffF} P_{diffF}}{P_{diffC}} - \frac{(1 - P_{diffC|diffF}) P_{diffF}}{1 - P_{diffC}} \tag{5}$$

The probabilities are estimated from N_{dC}, $N_{dF}[F]$ and $N_{dC\&dF}[F]$, where each of them is calculated as described in lines 6,8 and 9 from 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 editing the way that $diff(\tau(\cdot), R_i, I_j)$ is calculated. From Section 2 and Equation 1 we obtain:

$$diff(\tau(\cdot), R_i, I_j) = diff(S_i, S_j) = \sqrt{\sum_k w(c_k)(v_{i,k} - v_{j,k})^2} \tag{6}$$

where S_i and S_j are the target descriptions of R_i and I_j correspondingly, while $v_{i,k}$ and $v_{j,k}$ 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. We name this extension HMC-ReliefF and we test its properties on various datasets in the following text.

4 Experiments

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. This means, that in the random ranking, the distribution of the relevant features is expected to be uniform throughout the list.

Having this in mind, we employ a stepwise filter-like procedure [18] to evaluate our HMC-ReliefF algorithm. The idea is that starting from the ranked list of features, we construct classifiers for different number 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 $\mathscr{D}$, then the output would be a feature ranking $\mathbf{R}$, namely:

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

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

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

where:

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

If we assume that we can induce and evaluate a predictive model $\mathscr{M}(R_i, F_t)$, 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.

Algorithm 2 Stepwise evaluation of the top-k ranked features

Input: Feature Ranking, $\mathbf{R} = \{F_{r1}, \ldots, F_{rn}\}$; Target Feature, F_t
Output: FFA Curve, FFA, where $|FFA| = n$

```
R_S ⇐ ∅
for k = 1 to n do
    R_S ⇐ R_S ∪ feature(R, i)
    FFA[i] = qual(M(R_S, F_t))
end for
    return FFA
```

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), $|\mathscr{H}|$ is the number of classes in the hierarchy, $\mathscr{H}_d$ is the maximal depth of the classes in the hierarchy, $\overline{\mathscr{L}}$ is the average number of labels per example, and $\overline{\mathscr{L}_L}$ is the average number of leaf labels per example. Note that the values for $\mathscr{H}_d$ 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}	$\|D\|/\|C\|$	$\|\mathscr{H}\|$	$\mathscr{H}_d$	$\overline{\mathscr{L}}$	$\overline{\mathscr{L}_L}$
ImCLEF07D[5]	10000	0/80	46	3.0	3.0	1.0
ImCLEF07A[5]	10000	0/80	96	3.0	3.0	1.0
Reuters [14]	3000	0/47236	100	4.0	3.20	1.20
SCOP-GO [3]	6507	0/2003	523	5.5	6.26	0.95
SCOP-FUN [3]	2055	0/2003	250	4.0	3.42	0.95

For each step k of the filtering, 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 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.

4.1 Experimental Setup

In this section, we give the details of the specific experimental setup that we consider.

From the description of the HMC-ReliefF algorithm, given in Algorithm 1, there are two basic parameters on which the produced results depend. Those are the number of random instances that are chosen m and the number of nearest neighbours k that are used to calculate the feature relevance values. Therefore, we decided to explore a reasonable set of values of these parameters in order to evaluate the algorithm performance. More specifically, for the number of random instances m and the number of nearest neighbours k, we consider the following parameters:

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

As a baseline for our comparisons we use a set of 50 random rankings for each different dataset. For each of these rankings we perform the previously described procedure in Section 4 and we 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.

As a predictive model which we induce and evaluate, we use random forests of the so-called predictive clustering trees for hierarchical multi-label classification (PCT-HMCs)[19], [10]. The specific parameters that we used for the random forests of PCTs were 100 trees and a feature subset size of 10% of the dataset. In the HMC context,

there are various error measures that can be considered. We use the area of a variant of precision-recall curve, namely the Pooled Area Under the Precision-Recall Curve ($AU(\overline{PRC})$), details given in [19]. For estimating the $AU(\overline{PRC})$, we perform a ten-fold cross validation.

For the experiments we use datasets from various domains, which have classes organized in a hierarchy. We use 5 datasets from 3 domains, more specifically: biology (*SCOP-GO* and *SCOP-FUN*), text classification (*Reuters*) and image annotation/classification (*ImCLEF07D, ImCLEF07A*). The relevant parameters that characterize each dataset are given in Table 1. Note that the SCOP-GO dataset has a hierarchy organized as a DAG, while the remaining datasets have tree-shaped hierarchies. For more details on the datasets, we refer the reader to the referenced literature.

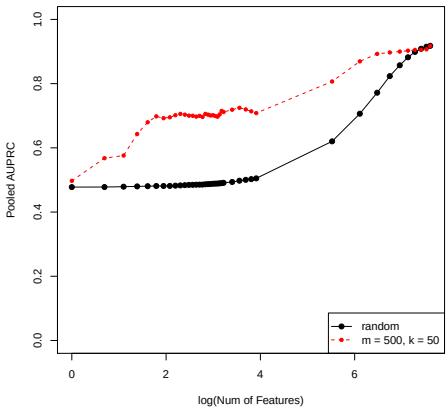
4.2 Results

In this section, we present the results from our experimental evaluation. In Figure 2, we give the results for the FFA curves for all of the datasets. For simplicity, we just compare the FFA curves for a single setting ($m = 500$, $k = 50$) and the expected FFA curves. An initial observation that can be made about all of the results is that the FFA curves of the HMC-ReliefF algorithm are most of the time above the FFA curves of the random rankings. This means that on 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.

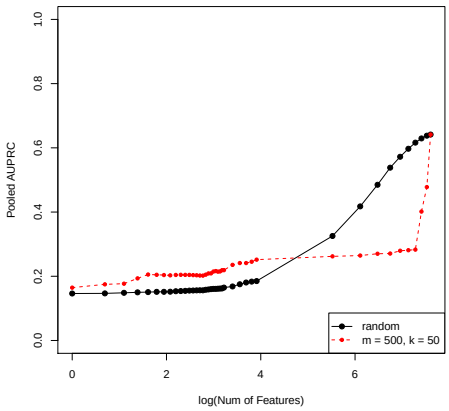
From the results, also some conclusions can be drawn about the behaviour of the HMC-ReliefF algorithm for different values of m and k. In Figure 3, we present a comparison of the FFA curves for a single dataset, namely *SCOP-GO*. The results are organised in four graphs, each showing FFA curves for a fixed k and different values for m, all compared to the expected FFA curve.

From all of the graphs, it can be observed that irrespectively of the number of neighbours k, a larger number of random instances m produces a better ranking. The best are obtained by setting the value of m to 250 and 500. This is also true for the *Reuters* dataset, while for the $ImCLEF07A$ and $ImCLEF07D$ datasets the results are not influenced by the change of the m and k parameters (results not shown).

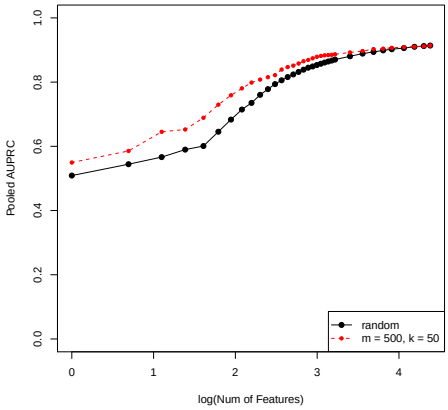
Opposite to the *SCOP-GO* dataset, the *SCOP-FUN* dataset seems to produce better results for a smaller k as well as a smaller m. This seems somewhat counter-intuitive considering that both of the datasets are from the same domain and share the same description space. However, we hypothesise that the different types of hierarchies considered (GO is a DAG and FUN is a tree) and their properties influenced these results.



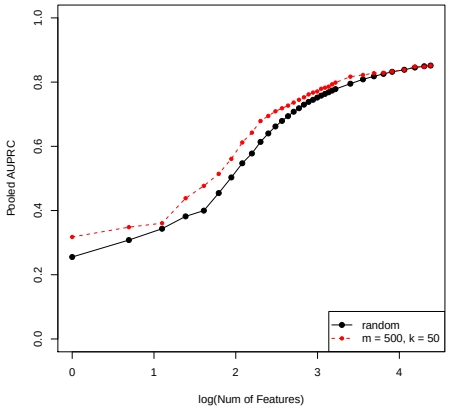
(a) FFA curves of **SCOP-GO**, **m = 500**, **k = 50**



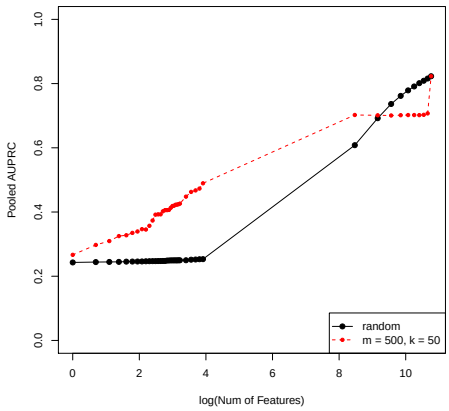
(b) FFA curves of **SCOP-FUN**, **m = 500**, **k = 50**



(c) FFA curves of **ImCLEF07D**, **m = 500**, **k = 50**

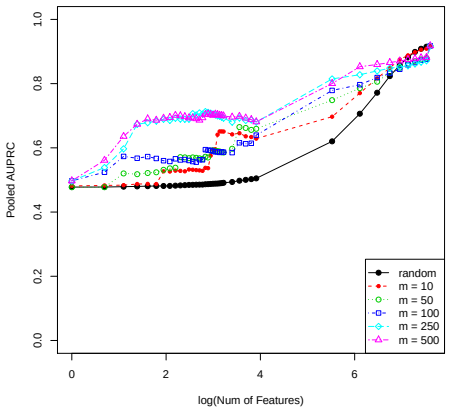


(d) FFA curves of **ImCLEF07A**, **m = 500**, **k = 50**

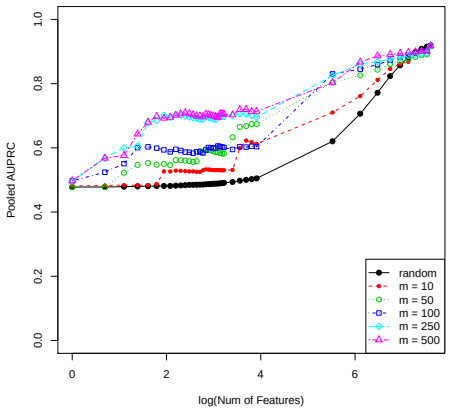


(e) FFA curves of **Reuters**, **m = 500**, **k = 50**

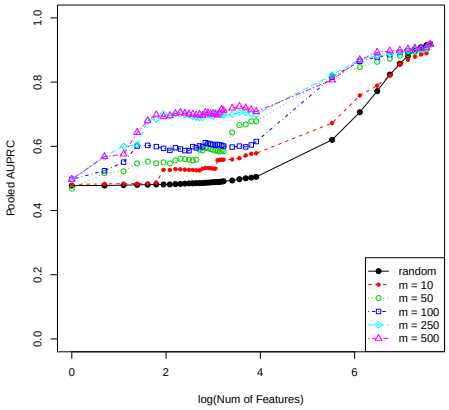
Fig. 2: Comparison of FFA curves of all datasets, for a fixed $m = 500$ and $k = 50$



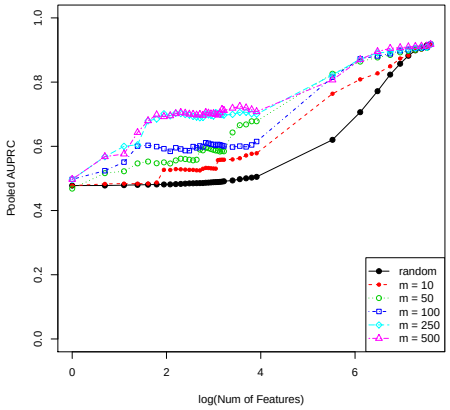
(a) FFA curves of **SCOP-GO**, **k = 10**



(b) FFA curves of **SCOP-GO**, **k = 25**



(c) FFA curves of **SCOP-GO**, **k = 50**



(d) FFA curves of **SCOP-GO**, **k = 100**

Fig. 3: Comparison of different FFA curves obtained by varying the number of *m* and *k* for the SCOP-GO dataset

5 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, with regards to 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, text mining and image annotation.

We evaluated the HMC-ReliefF algorithm on several datasets from different domains and with different properties of the hierarchies. We first investigated if our algorithm was able to detect relevant features in a dataset and put them on top of the ranking. This, we consider is a minimum requirement of any feature ranking algorithm. Additionally, we also explored a reasonable set of parameter settings of the HMC-ReliefF, which has influence on the feature relevance estimations.

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. The exploration of the various parameters of the HMC-ReliefF showed the following. For two of the datasets, a large value of m was preferred, irrespectively of the number of neighbours k. The FFA curves of the other two datasets (from the image annotation domain) remained stable for different values of the parameters. On the remaining dataset, the algorithm performed well for small values of m and k. These effects might have various causes, like the type and properties of the hierarchies considered, which at this time remain unexplored.

With this paper and the results presented we performed an initial investigation of the HMC-ReliefF algorithm. The directions of further work with respect to our HMC-ReliefF algorithm are numerous. One major direction of work, would be to define an artificial, controlled setting for investigating HMC problems. 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 also the advantages and limitations of the algorithm can be explored.

References

1. M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, 25(1):25–29, May 2000.
2. Avrim L. Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.
3. Amanda Clare. *Machine learning and data mining for yeast functional genomics*. PhD thesis, University of Wales Aberystwyth, Aberystwyth, Wales, UK, 2003.
4. Thomas G. Dietterich, Pedro Domingos, Lise Getoor, Stephen Muggleton, and Prasad Tadepalli. Structured machine learning: the next ten years. *Machine Learning*, 73(1):3–23, 2008.
5. Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sašo Džeroski. Hierchical annotation of medical images. In *Proceedings of the 11th International Multiconference - Information Society IS 2008*, pages 174–181. IJS, Ljubljana, 2008.

6. Thomas Gärtner and Shankar Vembu. On structured output training: hard cases and an efficient alternative. *Machine Learning*, 76:227–242, 2009.
7. Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March 2003.
8. Zengyou He and Weichuan Yu. Review article: Stable feature selection for biomarker discovery. *Comput. Biol. Chem.*, 34:215–225, August 2010.
9. Kenji Kira and Larry A. Rendell. A practical approach to feature selection. In *ML92: Proceedings of the ninth international workshop on Machine learning*, pages 249–256, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.
10. Dragi Kocev, Celine Vens, Jan Struyf, and Sašo Džeroski. Tree ensembles for predicting structured outputs. *Pattern Recognition*, 46(3):817–833, 2013.
11. Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97:273–324, 1997.
12. Igor Kononenko. Estimating attributes: Analysis and extensions of relief. In *European Conference on Machine Learning*, pages 171–182, 1994.
13. Hans-Peter Kriegel, Karsten Borgwardt, Peer Kröger, Alexey Pryakhin, Matthias Schubert, and Arthur Zimek. Future trends in data mining. *Data Mining and Knowledge Discovery*, 15:87–97, 2007.
14. David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.
15. Marko Robnik-Šikonja and Igor Kononenko. An adaptation of relief for attribute estimation in regression. In Douglas H. Fisher, editor, *ICML*, pages 296–304. Morgan Kaufmann, 1997.
16. Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Mach. Learn.*, 53:23–69, October 2003.
17. Carlos Silla and Alex Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72, 2011.
18. Ivica Slavkov. *An Evaluation Method for Feature Rankings*. PhD thesis, IPS Jožef Stefan, Ljubljana, Slovenia, 2012.
19. Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185–214, 2008.