A hierarchical ensemble method for DAG-structured taxonomies

Peter N. Robinson[1,4], Marco Frasca[2], Sebastian Köhler[1],
Marco Notaro[3], Matteo Re[2], and Giorgio Valentini[2]

[1] Institut fur Medizinische Genetik und Humangenetik,
Charité - Universitatsmedizin Berlin, Germany
{peter.robinson,sebastian.koehler}@charite.de
[2] AnacletoLab - DI, Dipartimento di Informatica,
Università degli Studi di Milano, Italy {valentini,frasca,re}@di.unimi.it
[3] Dipartimento di Bioscienze,
Università degli Studi di Milano, Italy marco.notaro@studenti.unimi.it
[4] Institute for Bioinformatics, Department of Mathematics and Computer Science,
Freie Universität Berlin, Germany

Abstract. Structured taxonomies characterize several real world problems, ranging from text categorization, to video annotation and protein function prediction. In this context "flat" learning methods may introduce inconsistent predictions, while structured output-aware learning methods can improve the accuracy of the predictions by exploiting the hierarchical relationships between classes. We propose a novel hierarchical ensemble method able to provide theoretically guaranteed consistent predictions for any Directed Acyclic Graph (DAG)-structured taxonomy, and consequently also for any taxonomy structured according to a tree. Results with a complex real-world DAG-structured taxonomy involving about one thousand classes and twenty thousand of examples show that the proposed hierarchical ensemble approach significantly improves flat methods, especially in terms of precision/recall curves.

Keywords: Hierarchical ensemble classification methods, DAG-structured prediction, multi-label classification.

1 Introduction

Structured output classification consists in the prediction of multiple labels that are hierarchically correlated according to a pre-defined data structure, e.g. a tree or a directed acyclic graph (DAG). In this context "flat" classification methods, that predict labels independently of each other, can in principle be applied, but may introduce significant inconsistencies in the classification, due to the violation of the *true path rule* (also known as the *annotation propagation rule*) that governs the hierarchical relationships between classes [1, 2]. According to this rule, a positive prediction for a class and a negative prediction for its parent classes are not allowed, since this violates the inclusion relationship between them. Therefore, a positive prediction for a class implies positive predictions for all of the ancestors of the class, and a negative prediction implies negative predictions for all of the class's descendants to avoid violating the true path rule. Moreover, flat methods do not take into account the hierarchical relationships between classes, thus loosing important a priori knowledge about the constraints of the hierarchical labeling.

To properly handle these problems, several structured output-aware learning methods have been proposed to exploit the a priori known relationships between labels. A first general approach is based on the kernelization of both the input and the output space, through the introduction of a joint kernel that computes the "compatibility" of a given input-output pair [4], or through other related techniques based on large margin methods for structured and interdependent output variables [5, 3]. A recent work showed also that structured output methods can be enhanced by combining them through relatively simple ensemble techniques [6].

A second general approach is based on ensemble methods able to exploit the hierarchical relationships between classes [7]. More precisely, hierarchical ensemble methods, in their more general form, adopt a two-step learning strategy. In the first step each base learner separately or interacting with connected base learners learns a specific class. In most cases this yields a set of independent classification problems, where each base learning machine is trained to learn a specific class, independently of the other base learners. In the second step the predictions provided by the trained classifiers are combined by considering the hierarchical relationships between the base classifiers modeled according to the hierarchy of the classes.

Most of the proposed hierarchical ensemble methods focused on tree-structured taxonomies [8, 7, 9, 10] and the ones specific for DAGs [1, 11] showed that it is difficult to improve upon flat predictions.

We propose a novel ensemble learning strategy that exploits the DAG structure of the taxonomy through a double flow of information between the base learners associated to each class/node of the hierarchy: after separately learning each class with a specific classifier, predictions are first combined from bottom to top to enhance sensitivity, and successively from top to bottom to improve the precision of the predictions.

We provide theoretical guarantees that the proposed True Path Rule (*TPR-DAG*) hierarchical ensembles obey the true path rule in DAGs. Moreover we experimentally show that our approach can consistently improve flat predictions in a complex task involving human gene - phenotype associations, where classes are DAG-structured according to the Human Phenotype Ontology (HPO) [12].

2 True Path Rule (*TPR-DAG*) hierarchical ensembles for DAG structured taxonomies

TPR-DAG requires a first phase in which any class is learned by a dedicated base learner: in principle any base learner can be used to score each example.

After this learning phase, the second phase modifies these "flat" predictions to provide the *TPR* ensemble predictions. This second phase is divided into two steps:

1. *Bottom-up step.* For each example the DAG is visited from bottom to top to propagate "positive" predictions across the hierarchy. The aim of this step is to enhance the sensitivity of the predictions.
2. *Top-down step.* Starting from the root, and traversing the DAG toward the bottom, "negative predictions" are propagated toward the children. The aim of this step is to enhance the precision of the predictions.

This method builds on the previously proposed *TPR* ensemble method that can be safely applied only to tree-structured taxonomies [13, 9]. The main difference with respect to the original tree-version consists in the fact that the per-level traversal is now performed through two completely distinct steps: a bottom-up per level visit of the graph followed by a top-down visit, while in the original tree-version the per-level traversal is performed in an "interleaved" fashion (that is the bottom-up and top-down traversal are alternated at each level [9]). Moreover the level of a class is defined in terms of its maximum distance from the root, since in a DAG we may have multiple paths from each node to the root. These two items (bottom-up and top-down separation and levels defined in terms of the maximum distance from the root) assure the true path rule consistency of the predictions, i.e. the requirement that the score of a parent or an ancestor node must be larger or equal than that of its children or descendants.

In the next subsections, after introducing some basic notations and definitions, we describe in detail the bottom-up and top-down steps of the *TPR-DAG* algorithm, as well its consistency properties.

2.1 Basic notation and definitions

Let $G =< V, E >$ denote a Directed Acyclic Graph (DAG) with vertices $V = \{1, 2, \ldots, |V|\}$ and edges $e = (i, j) \in E, i, j \in V$, where nodes $i \in V$ represent classes of the taxonomy and a direct edge $(i, j) \in E$ the hierarchical relationship between i and j: i is the parent class and j is the child class. The set of children of a node i is denoted $child(i)$, the set of its parents $par(i)$, the set of its ancestors $anc(i)$ and the set of its descendants $desc(i)$.

A "flat continuous" classifier $f_i : X \rightarrow [0, 1]$ associated with each node $i \in V$ provides scores $\hat{y}_i \in [0, 1]$ that can be interpreted as the likelihood or probability for a given example $x \in X$ of belonging to a given class i. The set of $|V|$ flat classifiers provides a multi-label score $\hat{\boldsymbol{y}} \in [0, 1]^{|V|}$:

$$\hat{\boldsymbol{y}} =< \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|V|} > \tag{1}$$

We say that a multi-label scoring $\boldsymbol{y}$ is consistent if it obeys the *true path rule*:

$$\boldsymbol{y} \text{ is consistent } \iff \forall i \in V, j \in par(i) \Rightarrow y_j \geq y_i \tag{2}$$

2.2 Bottom-up step

The basic *TPR-DAG* adopts a per-level bottom-up traversal of the DAG, starting from the nodes most distant (in the sense of the maximum distance) from the root. More precisely, if $p(r,i)$ represents a path from the root node r and a node $i \in V$, $l\left(p(r,i)\right)$ the length of a path p, $\mathcal{L} = \{0, 1, \ldots, \xi\}$ the set of observed levels, with ξ the maximum node level, then $\psi : V \longrightarrow \mathcal{L}$ is a level function which assigns each node $i \in V$ to its level $\psi(i)$:

$$\psi(i) = \max_{p} l\left(p(r,i)\right) \tag{3}$$

At each level the flat predictions $\hat{y}_i$ are changed to $\tilde{y}_i$ taking into account the "positive" predictions of its children:

$$\tilde{y}_i := \frac{1}{1 + |\phi_i|}(\hat{y}_i + \sum_{j \in \phi_i} \tilde{y}_j) \tag{4}$$

where ϕ_i are the "positive" children of i. The main goal of the bottom-up step consists in improving the sensitivity (recall) of the predictions. This is accomplished by allowing only the "positive" children (that is the nodes for which a relatively large score has been achieved) to transmit their scores to their parents. In this context a key issue is the selection of the positive children ϕ_i, and different strategies to select them can be applied:

1. *Threshold Free (TPR-TF) strategy.* A simple solution consists in choosing those children that can increment the score of the node i (that is positive nodes are those that achieve a higher score than that of their parent):

$$\phi_i := \{j \in \mathit{child}(i) | \tilde{y}_j > \hat{y}_i\} \tag{5}$$

2. *Thresholded (TPR-T) strategy.*
In this case we set a threshold to select the positive children. We can a priori select a given threshold $\bar{t}\ \forall i \in V$, or we can select the threshold to maximize some performance metric estimated on the available data, e.g. the F-score or the AUC. The corresponding set of positives $\forall i \in V$ is:

$$\phi_i := \{j \in \mathit{child}(i) | \tilde{y}_j > \bar{t}\} \tag{6}$$

For instance $\bar{t}$ can be selected from a set of $t \in (0,1)$ through cross-validation techniques.

Moreover we can also balance the weight $w \in [0,1]$ between the prediction of the classifier associated with the node i and that of its "positive" children ϕ_i, through their convex combination. In this way, analogously to the "tree" version of the weighted *TPR* ensemble method [14] we can obtain the "weighted" version *TPR-W* of the *TPR-DAG* algorithm:

$$\tilde{y}_i := w\hat{y}_i + \frac{(1-w)}{|\phi_i|}\sum_{j \in \phi_i} \tilde{y}_j \tag{7}$$

Independently of the variants of the basic *TPR-DAG* ensemble method, predictions are bottom-up propagated, thus moving positive predictions towards the parents and recursively towards the ancestors of each node.

2.3 Top-down step

The successive top-down step modifies the "bottom-up" scores computed in the previous bottom-up step (Sect. 2.2) by running in the opposite direction from the top to the bottom of the DAG. The main goal of this step consists in propagating "negative" predictions towards the children and recursively toward the descendants of each node, in order to provide consistent and more precise predictions. It adopts this simple rule by per-level visiting the nodes from top to bottom:

$$\bar{y}_i := \begin{cases} \tilde{y}_i & \text{if} \quad i \in root(G) \\ \min_{j \in par(i)} \bar{y}_j & \text{if} \quad \tilde{y}_i > \min_{j \in par(i)} \bar{y}_j \\ \tilde{y}_i & \text{otherwise} \end{cases} \tag{8}$$

The $\tilde{y}_i$ scores are those computed in the bottom-up step, while $\bar{y}_i$ are the final scores computer by the *TPR* ensemble.

The top-down step assures the hierarchical consistency of the predictions of the *TPR*, as stated by the following theorem:

Theorem 1. *Given a DAG $G =< V, E >$, a level function ψ that assigns to each node its maximum path length from the root, a set of predictions $\tilde{\boldsymbol{y}} =< \tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{|V|} >$ generated by the bottom-up step of the* TPR *algorithm for each class associated with its corresponding node $i \in \{1, \ldots, |V|\}$, the top-down step of the* TPR *algorithm assures that for the set of ensemble predictions $\bar{\boldsymbol{y}} =< \bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{|V|} >$ the following property holds:*

$$\forall i \in V,\ j \in par(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$$

The proof can be obtained by applying (8) to each node according to a per-level visit of the DAG, where levels are defined in terms of the maximum path length from the root (3), and by observing that each node is visited only once by the top-down step of the algorithm (details are omitted for lack of space).

From Theorem 1 it is easy to prove that the consistency of the predictions holds for all the ancestors of a given node $i \in V$:

Corollary 1. *Given a DAG $G =< V, E >$, the level function ψ, a set of flat predictions $\hat{\boldsymbol{y}} =< \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|V|} >$ for each class associated with each node $i \in \{1, \ldots, |V|\}$, the* TPR *algorithm assures that for the set of ensemble predictions $\bar{\boldsymbol{y}} =< \bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{|V|} >$ the following property holds:*

$$\forall i \in V,\ j \in anc(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$$

The proof can be easily obtained from Theorem 1 by "reductio ad absurdum".

The function ψ that computes the maximum distance of each node from the root (eq. 3) can be implemented through a straightforward variant of the classical Bellman-Ford algorithm [15]: by recalling that it finds the shortest paths from a source node to all the other nodes of a weighted digraph, it is sufficient to invert the sign of each edge weight to obtain the maximum distance (longest path) from the root. The complexity of the Bellman-Ford algorithm is cubic with respect the number of vertices, but recalling that the function ψ must be computed only once for a given hierarchical task, this complexity could be acceptable for most low and medium-sized DAGs. For big DAGs a variant of the classical topological sort algorithm for graphs can be applied instead: by exploiting the topological ordering of the nodes, the maximum distance from the root can be easily computed with time complexity $\mathcal{O}(|V| + |E|)$, that is in quadratic time for dense graph and in linear time for sparse DAGs with respect to the number of vertices.

2.4 The overall *TPR-DAG* algorithm

Fig. 1 shows the high-level pseudo-code of the *TPR-DAG* algorithm. The first four rows compute the maximum distance of each node from the root, using the Bellman-Ford algorithm. Note that the with a certain abuse of notation $E' := \{e' | e \in E, e' = -e\}$ indicates a new set E' of edges having weights with opposite sign with respect to the original set of edges E. The block B (rows 5-12) performs a bottom-up visit of the graph and updates the predictions $\tilde{y}_i$ of the *TPR* ensemble according to eq. 4 and one of the positive selection strategies described in Section 2.2. Note that this step propagates the "positive" predictions from bottom to top of the DAG, but does not assure their true path rule consistency. This is accomplished by the third block (rows $13 - 24$) that simply executes a hierarchical top-down step, according to the procedures described in Section 2.3.

It is easy to verify that complexity of the *TPR* algorithm is $\mathcal{O}(|V|)$ for both the B and C blocks when the DAG is sparse, while the complexity of block A depends on the selected algorithm: by choosing the variant of the Bellman-Ford algorithm the complexity is $\mathcal{O}(|V|^3)$, while by applying the variant of the topological sort algorithm the complexity is $\mathcal{O}(|V|+|E|)$. Note that block A must be executed only once for all the examples, while blocks B and C must be iterated for each example whose DAG-structured multi-label should be predicted.

3 Experimental set-up

We applied the proposed hierarchical ensemble methods to the prediction of Human Phenotype Ontology (HPO) terms associated with Mendelian disease genes [16]. The HPO aims at providing a standardized categorization of the abnormalities associated with human diseases and the semantic relationships between them. More precisely, HPO classes (terms) describe human phenotypic abnormalities and are structured according to a DAG, where children terms can be interpreted as subclasses of their parents. The experiments presented in this manuscript are based on the September 2013 HPO release ($10,099$ terms and $13,382$ between-term relationships). We downloaded from the same HPO release all the available annotations (gene-term associations), resulting in set of 2759 genes having at least 1 annotation. In our experiments we included a set of

Fig. 1. Hierarchical True Path Rule algorithm for DAGs (TPR-DAG)

```
Input:
- $G =< V, E >$
- $V = \{1, 2, \ldots, |V|\}$, 1 is the *root* node
- $\hat{\boldsymbol{y}} =< \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|V|} >, \quad \hat{y}_i \in [0, 1]$
begin algorithm
    A. Compute $\forall i \in V$ the max distance from $root(G)$:
        $E' := \{e' | e \in E,\ e' = -e\}$
        $G' :=< V, E' >$
        $dist :=$ Bellman.Ford$(G', root(G'))$
    B. Per-level bottom-up visit of $G$:
        for each $d$ from $\max(dist)$ to 0 do
            $N_d := \{i | dist(i) = d\}$
            for each $i \in N_d$ do
                Select $\phi_i$ according to a *positive selection strategy*
                $\tilde{y}_i := \frac{1}{1+|\phi_i|}(\hat{y}_i + \sum_{j \in \phi_i} \tilde{y}_j)$
            end for
        end for
    C. Per-level top-down visit of $G$:
        $\bar{y}_1 := \tilde{y}_1$
        for each $d$ from 1 to $\max(dist)$ do
            $N_d := \{i | dist(i) = d\}$
            for each $i \in N_d$ do
                $x := \min_{j \in par(i)} \bar{y}_j$
                if $(x < \tilde{y}_i)$
                    $\bar{y}_i := x$
                else
                    $\bar{y}_i := \tilde{y}_i$
            end for
        end for
end algorithm
Output:
- $\bar{\boldsymbol{y}} =< \bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{|V|} >$
```

20257 human genes, and hence more than 17000 genes had no HPO annotations. After pruning HPO terms having less than 50 annotations we obtained a final set of 911 HPO terms and $1,095$ between-term relationships that were used in our experiments.

A collection of feature vectors containing functional and biomolecular signatures describing the products of $20,257$ human genes was constructed starting from different publicly available biological databases (Table 1). Then the binary feature vectors were used to construct $n = 8$ gene networks (one for each data source listed in Table 1) by computing the Jaccard similarity between each possible pair of feature vectors associated to the genes.

Table 1. Data sources used in the experiments

Database	**Content**	**Web site**
InterPro	functional family, domains, functional sites	`www.ebi.ac.uk/interpro`
Pfam	functional family, domains	`pfam.xfam.org`
PRINTS	protein fingerprints, conserved motifs	`www.bioinf.manchester.ac.uk`
PROSITE	domains, families, functional sites	`prosite.expasy.org`
SMART	modular architectures	`smart.embl-heidelberg.de`
SUPFAM	structural and functional annotation	`supfam.cs.bris.ac.uk`
Gene Ontology	biological processes, cellular components and molecular functions	`geneontology.org`
OMIM	genetic diseases	`www.omim.org`
FI net (Wu et al.)	integrated network with expert-curated and non-curated sources of information	
HumanNet (Lee et al.)	integrated network with multi-species data	

We then combined the n gene networks by simply averaging the edge weights w_{ij}^{d} of each network $d \in \{1, n\}$ [17]:

$$\bar{w}_{ij} = \frac{1}{n} \sum_{d=1}^{n} w_{ij}^{d} \tag{9}$$

In order to construct a more informative gene network we performed the integration by adding two more functional gene networks (FI and HumanNet) taken from the literature [18, 19], thus obtaining a final integration of 10 biomolecular networks (Table 1).

To process and provide flat scores for the considered 911 HPO terms using the above networked data we applied two semi-supervised methods: a) the classical semi-supervised label propagation method (*LP*) based on Gaussian Fields and Harmonic Functions [20]; b) the kernelized score functions (*RANKS*) semi-supervised network-based learning method recently successfully applied to both gene disease prioritization [17], and drug repositioning [21]. *RANKS* implements both local and global learning strategies by embedding in a "local" score function a graph kernel that takes into account the "global" topology of the network. In our experiments we applied *RANKS* with the average score function and the *1-step random walk kernel* [22].

4 Results

We compared the generalization performance of *Flat* and *TPR* ensemble methods by using 5-fold cross-validation techniques, and considering separately the two different base learners (*RANKS* and *LP*, Section 3). We also compared the results of *TPR* ensemble methods with three heuristic hierarchical ensemble methods (i.e. *And*, *Or* and *Max*), originally proposed for the hierarchical prediction of Gene Ontology terms [1]. It is worth noting that in the same work [1] *isotonic regression*-based hierarchical methods achieved better results than the heuristic ensemble algorithms used in our experiments, but we did non use them due to their computational complexity, considering the relatively large size of the taxonomy and of the input data considered here.

Table 2. Average AUC, and precision at 10, 20 and 40% recall (P10R, P20R and P40R), using kernelized score functions as base learner. Flat stands for flat ensemble method, TPR-TF for True Path Rule Threshold-Free, Max for Hierarchical Maximum, And for Hierarchical And and Or for Hierarchical Or ensemble methods. Methods that are significantly better than all the others according to the Wilcoxon rank sum test ($\alpha = 10^{-5}$) are highlighted in bold.

	Flat	**TPR-TF**	**Max**	**And**	**Or**
AUC	0.8213	**0.8269**	0.8241	**0.8274**	0.8241
P10R	0.2969	**0.3427**	0.2908	0.2815	0.2994
P20R	0.2043	**0.2333**	0.2025	0.1903	0.2081
P40R	0.1054	**0.1225**	0.1071	0.0993	0.1095

4.1 Experimental results using kernelized score functions (*RANKS*) as base learner

By looking at the single 911 HPO terms (classes), in terms of AUC the *TPR-TF* ensemble achieves better results than *Flat* for 830 terms and worse results for 81 HPO terms. Table 2 shows that the average AUC across classes is only slightly larger for *TPR-TF* ensembles with respect to *Flat*, but the difference is statistically significant according to the Wilcoxon rank sum test. Also with respect to three heuristic hierarchical ensemble methods (*And*, *Or*, *Max*) *TPR-TF* achieve equal or significantly better results. More precisely the difference is statistically significant with respect to *Or* and *Max*, while no significant difference is registered with the *And* method.

Better results are obtained by the *TPR-TF* method in terms of the precision at fixed recall rates. Indeed the difference is statistically significant with respect to *Flat* and the three heuristic hierarchical ensemble methods, both at 10, 20 and 40% recall (Table 2). These results are confirmed also by the precision-recall curves (Fig. 2): the *TPR-TF* solid line marked with circles is consistently above all the other curves, showing that *TPR-TF* achieves on the average better results than all the other methods compared.

On the contrary, by comparing the different variants of the proposed *TPR* hierarchical ensemble methods, no statistically significant differences between them were identified (data not shown).

4.2 Experimental results using label propagation (*LP*) as base learner

We repeated the same experiments using this time the label propagation (*LP*) method to implement the flat base learners. Also with this base learner we achieved significantly better results with *TPR* ensemble methods with respect to the *Flat* approach, both in terms of the average AUC and average precision at fixed recall rates. Especially considering precision at fixed recall rates the *TPR* ensemble achieved significantly better results than those obtained with *Flat* and

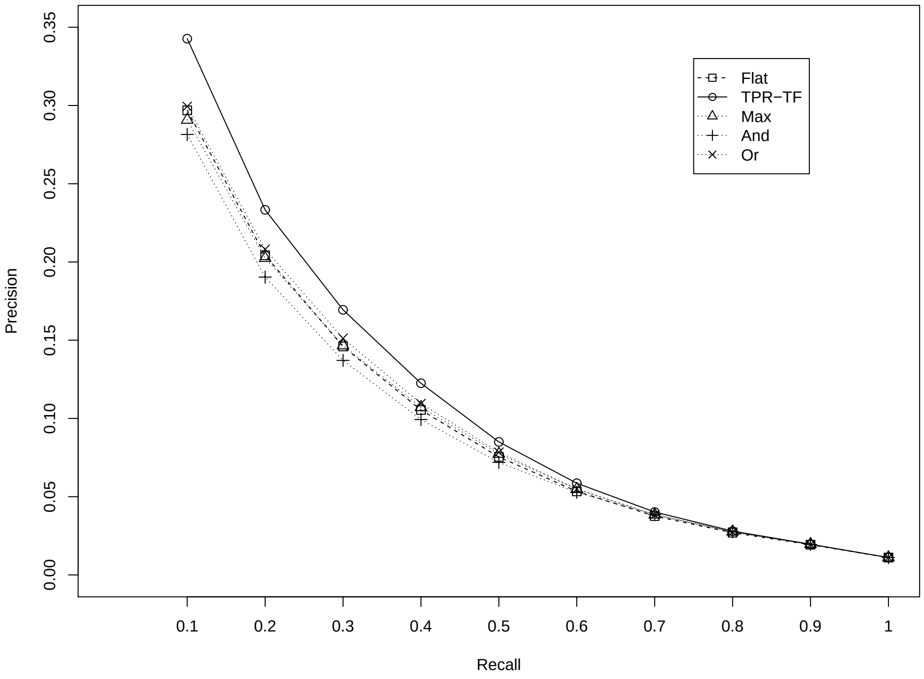

Fig. 2. Precision at different levels of recall, averaged across HPO terms (base learner: kernelized score functions)

the three heuristic hierarchical ensemble methods, according to the Wilcoxon rank sum test (Table 3).

It is worth noting that the absolute average AUC and precision values obtained with the *LP* base learner (Table 3) are significantly lower than those achieved with *RANKS* (Table 2), showing that *TPR* results, as well as those obtained by the other heuristic ensemble methods depend on the choice of the flat base learner. Nevertheless, *TPR* ensemble methods with *LP* base learners are able to achieve a relative precision improvement with respect to the *Flat* approach in the range between 15 and 40%, at least for recall rates between 0.1 and 0.4 (Table 3). Note that this is not the case for the three heuristic hierarchical ensemble methods (*And*, *Or*, *Max*), confirming previous results obtained in the context of gene function prediction problems [1].

Table 3. Average AUC, and precision at 10, 20 and 40% recall (P10R, P20R and P40R), using label propagation as base learner. TPR-T stands for True Path Rule ensembles with Threshold. Methods that are significantly better than all the others according to the Wilcoxon rank sum test ($\alpha = 10^{-5}$) are highlighted in bold.

	Flat	**TPR-T**	**Max**	**And**	**Or**
AUC	0.7883	**0.7967**	0.7869	**0.7974**	0.7923
P10R	0.0673	**0.0936**	0.0653	0.0704	0.0730
P20R	0.0568	**0.0709**	0.0549	0.0564	0.0606
P40R	0.0439	**0.0503**	0.0426	0.0444	0.0462

5 Conclusions

Several real-world problems ranging from text classification to protein function prediction are characterized by hierarchical multi-label classification tasks. In this context flat methods may provide inconsistent predictions and more in general are not able to exploit the hierarchical constraints between classes.

We theoretically guarantee that *TPR-DAG* ensembles provide predictions that obey the true path rule in DAG-structured taxonomies, and we show in a large experiment involving the DAG-structured Human Phenotype Ontology that our proposed hierarchical ensembles consistently outperform flat methods, independently of the base learner used.

We outline that the proposed hierarchical method is independent of the base learner used, even if learners providing scores or probabilities of belonging to a given class are better suited for the *TPR-DAG* ensembles. From this standpoint *TPR* methods can be conceived as a flexible tool that can be applied to any off-the-shelf flat method to improve its predictions for DAG-structured taxonomies. *TPR-DAG* can be also applied also to tree-structured taxonomies, since obviously trees are DAGs.

This reseach could be extended by exploring other base learners, including also supervised learners, and by comparing *TPR* with other hierarchical methods, including also structured output kernel methods. To test the effectiveness of *TPR-DAG* ensembles in different application domains, the hierarchical classification of web documents and the protein function prediction problem could be two significant real-world test-beds for future experiments.

References

[1] Obozinski, G. et al.: Consistent probabilistic output for protein function prediction. Genome Biology **9**(S6) (2008)

[2] Robinson, P.N., Bauer S.: Introduction to Bio-Ontologies. CRC Press, Boca Raton, FL (2011)

[3] Bakir, G.et al.: Predicting structured data. MIT Press, Cambridge, MA (2007)

[4] Lampert, C., Blaschko, M.: Structured prediction by joint kernel support estimation. Machine Learning **77** (2009) 249–269

[5] Tsochantaridis, I. et al.: Large margin methods for structured and interdependent output variables. JMLR **6** (2005) 1453–1484

[6] Cortes, C., Kuznetsov, V., Mohri, M.: Ensemble methods for structured prediction. In: Proc. of the 31st ICML, Beijing, China (2014)

[7] Silla, C., Freitas, A.: A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery **22**(1-2) (2011) 31–72

[8] Wang, H., She, X., Pan, W.: Large margin hierarchical classification with mutually exclusive class membership. JMLR **12** (2011) 2649–2676

[9] Valentini, G.: True Path Rule hierarchical ensembles for genome-wide gene function prediction. IEEE ACM Trans. on Comp. Biol. and Bioinf. **8**(3) (2011) 832–847

[10] Cesa-Bianchi, N., Re, M., Valentini, G.: Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. Machine Learning **88**(1) (2012) 209–241

[11] Schietgat, L. et al.: Predicting gene function using hierarchical multi-label decision tree ensembles. BMC Bioinformatics **11**(2) (2010)

[12] Robinson, P. et al.: The Human Phenotype Ontology: a tool for annotating and analyzing human hereditary disease. Am. J. Hum. Genet. **83** (2008) 610–615

[13] Valentini, G.: True path rule hierarchical ensembles. In Kittler, J., Benediktsson, J., Roli, F., eds.: MCS 2009, Reykjavik, Iceland. Volume 5519 of LNCS, Springer (2009) 232–241

[14] Re, M., Valentini, G.: An experimental comparison of Hierarchical Bayes and True Path Rule ensembles for protein function prediction. In: MCS 2010, Cairo, Egypt. Volume 5997 of LNCS, Springer (2010) 294–303

[15] Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. MIT Press, Boston (2009)

[16] Kohler, S., et al.: The human phenotype ontology project: linking molecular biology and disease through phenotype data. Nucleic Acids Research **42**((Database issue)) (2014) D966–74

[17] Valentini, G. et al.: An extensive analysis of disease-gene associations using network integration and fast kernel-based gene prioritization methods. Artificial Intelligence in Medicine **61**(2) (2014) 63–78

[18] Wu, G., Feng, X., Stein, L.: A human functional protein interaction network and its application to cancer data analysis. Genome Biol **11** (2010) R53

[19] Lee, I. et al.: Prioritizing candidate disease genes by network-based boosting of genome-wide association data. Genome Research **21**(7) (2011) 1109–1121

[20] Zhu, X., et al.: Semi-supervised learning with gaussian fields and harmonic functions. In: Proc. of the 20th ICML, Washintgton DC, USA (2003)

[21] Re, M., Valentini, G.: Network-based Drug Ranking and Repositioning with respect to DrugBank Therapeutic Categories. IEEE ACM Trans. on Comp. Biol. and Bioinf. **10**(6) (2013) 1359–1371

[22] Re, M., Valentini, G.: Cancer module genes ranking using kernelized score functions. BMC Bioinformatics **13**(Suppl 14/S3) (2012)