True Path Rule Hierarchical Ensembles

Giorgio Valentini

DSI, Dipartimento di Scienze dell' Informazione,
Università degli Studi di Milano,
Via Comelico 39, 20135 Milano, Italia.
valentini@dsi.unimi.it

Abstract. Hierarchical classification problems gained increasing attention within the machine learning community, and several methods for hierarchically structured taxonomies have been recently proposed, with applications ranging from classification of web documents to bioinformatics. In this paper we propose a novel ensemble algorithm for multilabel, multi-path, tree-structured hierarchical classification problems based on the true path rule borrowed from the Gene Ontology. Local base classifiers, each specialized to recognize a single class of the hierarchy, exchange information between them to achieve a global "consensus" ensemble decision. A two-way asymmetric flow of information crosses the tree-structured ensemble: positive predictions for a node influence its ancestors, while negative predictions influence its offsprings. The resulting *True Path Rule* hierarchical ensemble is applied to the prediction of gene function in the yeast, using the FunCat taxonomy and biomolecular data obtained from high-throughput biotechnologies.

1 Introduction

Several interesting real-world classification problems are characterized by hierarchical relationships between classes [1, 2, 3]. These problems come from different fields, ranging from textual classification of web content [1, 2], to gene function prediction in bioinformatics [3, 4], and share the common property that a certain general class may be further specified by more refined classes at different levels of an overall hierarchy. For instance, in the *FunCat* taxonomy [5] the general class "metabolism" has several child classes, such as "amino acid metabolism", "C-compound and carbohydrate metabolism", "lipid and fatty acid metabolism" and others that provide more detailed specifications and subdivisions of the parent class. Moreover each child class, e.g. "amino acid metabolism", can be further subdivided in "metabolism of the aspartate family", "metabolism of the cysteine - aromatic group" and so on, thus resulting in a complex hierarchy divided at multiple levels.

Several hierarchical algorithms have been proposed in the literature, with different characteristics and purposes, considering for instance methods restricted to multilabels with single and no partial paths [1, 6], or other methods extended to multiple and also partial paths [2, 7]. Nevertheless, algorithms that explicitly take into account the relationships between the classes of the structured hierarchy received much less attention.

In particular in this paper we propose a hierarchical ensemble algorithm, by which classifications of positive examples in child nodes influence the prediction of the parent node in a recursive way, while negative predictions in a node influence the prediction in the descendant nodes. This general behaviour is a consequence of the *true path rule*, a term borrowed from the Gene Ontology [8]: according to this rule, if an example belongs to a class, it belongs to all its ancestors, and if does not belong to a class it does not belong to all its offsprings.

In the next section the main motivations and characteristics of the proposed ensemble algorithm are presented and discussed. Then in Sect. 3 we test the proposed method on a complex hierarchical gene function prediction problem, using the FunCat taxonomy and bio-molecular data obtained from public databases, and discuss some drawbacks and possible enhancements of the proposed hierarchical ensemble approach. The conclusions end the paper.

2 An Ensemble Algorithm Based on the True Path Rule

2.1 Definitions and Notation

We consider a multiclass multilabel classification problem where the classes are structured according to a given hierarchy.

More precisely, an example x can be assigned to 1 or more classes of the set $\Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}$. The assignments are coded through a vector of multilabels $\mathbf{y} =< y_1, y_2, \ldots, y_m > \in \{0,1\}^m$, by which if x belongs to class ω_j, then $y_j = 1$, otherwise $y_j = 0$.

The classes are structured according to a hierarchy and can be represented by a directed graph, where nodes correspond to classes, and arcs to relationships between classes. Considering that each node corresponds to a class, the node corresponding to the class ω_i may be simply denoted by i. We denote by child(i) the set of children nodes of i, while par(i) represents the set of the parents of node i. Moreover $y_{child(i)}$ denotes the labels of the children classes of node i and analogously $y_{par(i)}$ denotes the labels of the parent classes of i.

A classifier $D : X \rightarrow \{0,1\}^m$ computes the multilabel associated to each example $x \in X$, and $d_i(x) \in \{0,1\}$ is the label predicted by the classifier for class ω_i. For the sake of simplicity if there is no ambiguity we represent $d_i(x)$ simply by d_i.

2.2 The True Path Rule

The proposed algorithm is inspired by the "true path rule" that characterizes the hierarchy of the gene functional classes of both the *Gene Ontology (GO)* [8] and *FunCat* [5] taxonomies:

> "If the child term describes the gene product, then all its parent terms must also apply to that gene product"

This means that if a gene is annotated with a specific functional term (functional class), then it is annotated with all the "parent" classes, and with all its ancestors in a recursive way. On the contrary if a gene is not annotated to a a class, it cannot be annotated to its offsprings. (Fig. 1).

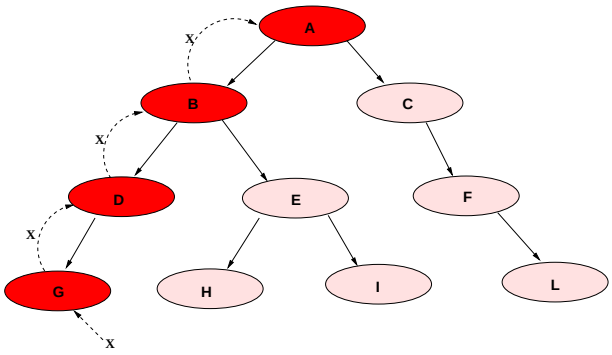

Fig. 1: True path rule: if example x belongs to class G then it belongs also to class D, B and A. On the contrary if an example x does not belong to class C it cannot belong to class F or L.

From the "true path rule", for a given example x, considering the parents of a given node i, the following rules can be immediately deduced:

$$\begin{cases} y_i = 1 \Rightarrow y_{par(i)} = 1 \\ y_i = 0 \nRightarrow y_{par(i)} = 0 \end{cases} \tag{1}$$

As a consequence a classifier that respects the true path rule needs to obey the following rules:

$$\begin{cases} d_i = 1 \Rightarrow d_{par(i)} = 1 \\ d_i = 0 \nRightarrow d_{par(i)} = 0 \end{cases} \tag{2}$$

On the other hand, considering the children of a given node i, the following rules can be immediately deduced:

$$\begin{cases} y_i = 1 \nRightarrow y_{child(i)} = 1 \\ y_i = 0 \Rightarrow y_{child(i)} = 0 \end{cases} \tag{3}$$

and a classifier that respects the true path rule needs to obey the following rules:

$$\begin{cases} d_i = 1 \nRightarrow d_{child(i)} = 1 \\ d_i = 0 \Rightarrow d_{child(i)} = 0 \end{cases} \tag{4}$$

From eq. 1 and 3 we can observe an asymmetry in the rules that govern the assignments of positive and negative labels. Indeed we have a propagation of positive labels from bottom to top of the hierarchy (eq. 1), and a propagation of negative labels from top to bottom (eq. 3). On the contrary negative labels cannot propagate from bottom to top, and positive predictions cannot propagate from top to bottom.

2.3 The Main Ideas behind the Algorithm

We can design a hierarchical classifier that uses the predictions made at each node by local "base" classifiers and puts together their decisions to realize an ensemble that obeys the "true path rule". More precisely the basic ideas behind the *true path rule ensemble algorithm* are the following:

1. A set of base classifiers associated to each class/node of the graph provides a local decision about the assignment of a given example to a given node.
2. Positive decisions for a node influence the decisions made by the parent nodes in a recursive way (that is a positive decision influences the parent and may propagate from bottom to top across the graph). On the contrary negative decisions do no affect decisions of the parent node (that is they do not propagate from bottom to up, eq. 2).
3. If the classifier takes a negative prediction for a given node (taking into account the local decision of its descendants), it in turns set to negative all its descendants, to preserve the consistency of the hierarchy according to the true path rule. On the contrary positive decisions do not influence decisions of child nodes (eq. 4).

The decision of the ensemble classifier is thus the result of the local predictions made by the base classifiers associated to each node modified in order to take into account positive predictions that comes from the bottom of the graph and negative predictions that comes from the top of the graph.

We propose an algorithm for tree-structured graphs that scans the tree from bottom to top through a per level traversal of the tree. Base classifiers estimate local probabilities $\hat{p}_i(x)$ that a given example x belongs to class ω_i, and the ensemble corrects the local probabilities to estimate the "consensus" probability $p_i(x)$. More precisely, given the local estimates of the probabilities $\hat{p}_j(x)$ made by the base classifiers across the tree T of the m classes, the probability that an example x belongs to class ω_i is:

$$p_i(x) = P(\omega_i | x, T, \hat{p}_j(x), 1 \leq j \leq m) \tag{5}$$

2.4 The Hierarchical Ensemble Algorithm

The algorithms starts to train the m base learners (one for each node/class of the hierarchy); each trained classifiers computes an estimate of the local probabilities $\hat{p}_j(x)$. The core of the algorithm is represented by the evaluation phase, where the ensemble provides an estimate of the "consensus" global probability $p_i(x)$. A detailed representation of the evaluation phase of the algorithm is given in Algorithm 1. In the algorithm there are two main for loops: the external for (from row 1 to 26) handles a per level bottom-up traversal of the tree, while the internal (from row 2 to 25) scans the nodes at each level. If a node is a leaf (row 3), then the consensus probability p_i is equal to the local probability $\hat{p}_i(x)$. Note that a positive decision is taken if $p_i(x)$ is larger than a threshold t (row 5): a natural choice for t is 0.5. If a node is not a leaf (row 10), at first the

Algorithm 1 True Path Rule (TPR) hierarchical ensemble

Input:
- a test example x
- tree T of the m hierarchical classes
- set of m classifiers (one for each node) each predicting $\hat{p}_i(x)$, $1 \leq i \leq m$

1: **for all** levels k of T from bottom to top **do**
2: **for all** nodes i at level k **do**
3: **if** i is a leaf **then**
4: $p_i(x) \leftarrow \hat{p}_i(x)$
5: **if** $p_i(x) > t$ **then**
6: $d_i(x) \leftarrow 1$
7: **else**
8: $d_i(x) \leftarrow 0$
9: **end if**
10: **else**
11: $\phi(x) \leftarrow \{j | j \in \text{child}(i), d_j(x) = 1\}$
12: $p_i(x) \leftarrow \frac{1}{1+|\phi(x)|} \left(\hat{p}_i(x) + \sum_{j \in \phi(x)} p_j(x)\right)$
13: **if** $p_i(x) > t$ **then**
14: $d_i(x) \leftarrow 1$
15: **else**
16: $d_i(x) \leftarrow 0$
17: **for all** $j \in \text{subtree}(i)$ **do**
18: $d_j(x) \leftarrow 0$
19: **if** $p_j(x) > t$ **then**
20: $p_j(x) \leftarrow t$
21: **end if**
22: **end for**
23: **end if**
24: **end if**
25: **end for**
26: **end for**

Output:
- the ensemble decisions $d_i(x) = \begin{cases} 1 & \text{if } x \text{ belongs to node } i \\ 0 & \text{otherwise} \end{cases}$
- the probabilities $p_i(x)$ that x belongs to the node $i \in T$

set $\phi(x)$ collects all the children nodes for which we have a positive prediction, and the consensus probability p_i of the ensemble is computed by considering both the local estimate of the probability $\hat{p}_i$ and the probabilities computed by the children nodes for which a positive decision has been taken (row 12). Note that in case of a negative decision for the node i, all the classes belonging to the subtree rooted at i are set to negative (rows 17-18). The algorithm provides both the multilabels associated to the example x and the probabilities p_i that a given example belongs to the class i, $1 \leq i \leq m$.

3 Experimental Results

3.1 Hierarchical Classification of Functional Classes of Genes

We considered the functional classification of yeast genes for a large number of classes structured according to the *FunCat* (Functional Catalogue), a hierarchically tree-structured, controlled classification system enabling the functional description of proteins from any organism [5].

We selected only the genes annotated to FunCat (funcat-2.1 scheme), available from the MIPS web site (`http://mips.gsf.de/projects/funcat`), using the *Hcgene* R package [9]. We also removed the genes annotated only with the "99" FunCat class ("UNCLASSIFIED PROTEINS") and selected classes with at least 20 positive examples, in order to get a not too small set of positive examples for training. The resulting tree has a depth equal to 5 and includes about 200 functional classes. Different strategies can be chosen to select negative examples for each functional class [10, 9]. In this work negative examples for each class have been selected in such a way that they are not annotated for the class, but belong to the parent class (i.e. positive for the parent class). In this way only negative examples that are not too dissimilar to the positive ones are selected.

3.2 Data sets

We chose four different types of bio-molecular data obtained from high-throughput bio-technologies and available from public databases or from literature. The main characteristics of the data we used in our experiments are summarized in Tab. 1. Proteins are constituted by structured and functionally character-

Table 1: Data sets

Data set	n. examples	n. feat.	n.classes
Protein domain	3529	5724	211
Phylogenesis	2445	24	187
Gene expression	4532	250	230
PPI - BioGRID	4531	5367	232

ized regions usually referred as domains joined by unstructured regions named loops. To capture this source of functional information we considered the E-value assigned to each gene product by a collection of profile-HMMs, each of which trained on a specific domain family, using data from the *Pfam* (Protein families) database [11]. The E-values have been obtained by means of the HMMER software toolkit [12].

Phylogenetic data have been obtained through BLAST searches [13]: each feature corresponds to the negative logarithm of the lowest E-value reported by

BLAST version 2.0 in a search against a complete genome, with negative values (corresponding to E-values greater than 1) truncated to 0 [14].

We merged the gene expression experiments of Spellman et al. (gene expression measures relative to 77 conditions) [15] with the transcriptional responses of yeast to environmental stress (173 conditions) by Gasch et al. [16], thus obtaining real-valued vector data with 250 features.

Finally we downloaded protein-protein interaction (PPI) data from the *BioGRID* database, that collects PPI data from both high-throughput studies and conventional focused studies [17]. Data are binary: they represent the presence or absence of protein-protein interactions.

3.3 Experimental Setup

For each data set we evaluated the performance of three different ensembles: the *Flat* ensemble, that does not take into account the hierarchical structure of the data, the *Hierarchical Top-Down* and the proposed *True Path Rule (TPR)* Hierarchical Bottom-Up ensemble. The classical hierarchical Top-down algorithm classifies an example x, where $d_i(x)$ is the classifier decision at node i and $root(T)$ denotes the set of nodes at the first level of the tree T, in the following way:

$$y_i = \begin{cases} d_i(x) \text{ if } i \in root(T) \\ d_i(x) \text{ if } i \notin root(T) \wedge y_{par(i)} = 1 \\ 0 \quad\quad \text{if } i \notin root(T) \wedge y_{par(i)} = 0 \end{cases}$$

As base learners we used 2^{nd} and 3^{rd} degree polynomial SVMs. The probabilistic output of the SVMs composing TPR ensembles has been computed using the sigmoid fitting proposed in [18].

Considering the large unbalance between positive and negative examples available for each class, we evaluated the performance of the ensembles through the F-measure, i.e. the harmonic mean between precision and recall, by applying for each data set 5-fold cross-validation techniques. We performed a limited model selection for the base learners, by applying a grid search only to the first level nodes (classifiers) of the tree (the nodes closest to the root), and then we extended the resulting best model parameters to all the other classifiers of the tree.

3.4 Results

Results of the comparison between Flat, Top-down and True Path Rule hierarchical ensembles are summarized in Tab. 2. The table reports the average F-measure across classes, using the same 0/1 loss for each class of the hierarchy. Data in bold denote results for an ensemble better than both the other two (at 0.05 significance level), according to the 5-fold cross-validated paired t-test [19]. True Path Rule ensembles achieve significantly better results with respect to both Flat and Hierarchical top-down ensembles: only with Gene expression data

Table 2: Average F-measure across FunCat classes: comparison between Flat, Top-down and TPR (true path rule) ensembles.

Data set	**Flat**	**Top-down**	**TPR**
Protein domain	0.0976	0.1246	**0.1590**
Phylogenetic	0.0204	0.0005	**0.0708**
Gene expression	0.0882	0.1139	0.1058
PPI - BioGRID	0.0396	0.0255	**0.1257**
Average across data	0.0614	0.0661	**0.1153**

Top-down ensembles perform better, even if the difference is not statistically significant.

Looking at Tab. 3 we can observe that the better results of TPR ensembles are due to a better balancing between precision and recall. Indeed on the average the higher recall is obtained by the Flat ensemble, while the higher average precision by the Top-down ensembles (Tab. 3). In both cases the recall and precision of the TPR ensemble is on the middle, but results in a larger F-measure. Nevertheless, for real applications to gene function prediction, the precision is actually too low to be useful in practice. Indeed in real applications an "in silico" prediction needs to be validated by "in vitro" biological functional validation, and we need a reasonably high precision to justify the more expensive biological validation.

Note that here we consider the average precision, recall, and F-measure across classes, and hence we may obtain an average F-measure that is lower of both the average precision and recall.

Table 3: Average Precision and Recall across FunCat classes: comparison between Flat, Top-down and TPR (true path rule) ensembles.

	Flat		**Top-down**		**TPR**	
Data set	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Protein domain	0.1133	0.3256	0.3370	0.0800	0.1488	0.2395
Phylogenetic	0.1288	0.2095	0.0103	0.0002	0.1050	0.0853
Gene expression	0.0669	0.3772	0.1518	0.0961	0.0757	0.2777
PPI - BioGRID	0.1462	0.2282	0.2235	0.0145	0.1862	0.1204
Average across data	0.1138	0.2851	0.1806	0.0477	0.1289	0.1807

Even if the average accuracy across classes is quite high (for both Hierarchical Top-down and TPR ensembles is larger than 90%, while for Flat is about 75%, data not shown), note that this results is not so significant, considering the large unbalance between positive and negative examples for most functional classes. On the contrary the F-measure is quite low: the average across data sets is only

0.1153 for TPR ensembles and this result is halved with both Flat and Top-down ensembles (Tab. 2).

These relatively poor results are due to the intrinsic complexity of the hierarchical multiclass multilabel classification of genes [20]. In many cases biomolecular data obtained through complex bio-technologies are affected by a relatively high degree of noise. Moreover, usually each data set can provide useful information only for a subset of classes, while for others may be substantially uninformative. It is well-known that by combining multiple sources of data we can substantially improve the results [14, 3], and we may expect substantial improvements by applying data fusion techniques with TPR ensembles.

Another important problem is the local model selection of each base learner. In the experiments, for computational complexity reasons, we applied a relatively moderate model selection strategy limited only to the first level of the tree hierarchy (16 nodes/classes out of more than 200). By applying a computational intensive model selection through internal cross validation we may expect a further improvement of the results of TPR ensembles.

4 Conclusions

In this work we presented a novel ensemble algorithm for multiclass multilabel hierarchical classification problems. The training phase is straightforward (even if computationally intensive), but the core of the algorithm is represented by the evaluation phase. At this stage the base classifiers associated to each node of the tree exchange information in an asymmetric way from bottom to top and top to bottom: positive predictions affect the decisions at "higher level" nodes (i.e. ancestor nodes), while negative predictions affect offsprings, according to the *true path rule* borrowed from the Gene Ontology.

Even if this algorithm has been conceived for the prediction of the function of genes at genome-wide level, it is sufficiently general to be applied in other similar hierarchical problems in different fields and contexts.

The preliminary experimental results show that TPR ensembles are competitive with respect to both classical Flat and Hierarchical Top-down ensembles, and suggest also further directions to improve the basic TPR algorithm. For instance, considering that the decision for a class is influenced only by positive decisions of its offsprings, an ongoing research line consists in explicitly balancing the weigth of the local predictor with respect to that of its children: in this way we could tune the precision and the recall of the ensemble. Moreover, by introducing model selection strategies at each node and data fusion techniques to exploit multiple sources of biomolecular data, we may expect to substantially improve the overall performance of the ensemble.

Acknowledgments

The author would like to thank the anonymous reviewers for their comments, and gratefully acknowledges partial support by the PASCAL2 Network of Excellence under EC grant no. 216886. This publication only reflects the author's views.

References

[1] Dumais, S., Chen, H.: Hierarchical classification of web content. In: Proc. of the 23rd ACM Int. Conf. on Research and Development in Information Retrieval, ACM Press (2000) 256–263
[2] Rousu, J. et al.: Learning hierarchical multi-category text classification models. In: Proc. of the 22nd ICML, Omnipress (2005) 745–752
[3] Barutcuoglu, Z., Schapire, R., Troyanskaya, O.: Hierarchical multi-label prediction of gene function. Bioinformatics **22** (2006) 830–836
[4] Guan, Y. et al.: Predicting gene function in a hierarchical context with an ensemble of classifiers. Genome Biology **9** (2008)
[5] Ruepp, A. et al.: The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucl. Ac. Res. **32** (2004) 5539–5545
[6] Dekel, O., Keshet, J., Singer, Y.: Large margin hierarchical classification. In: Proc. of the 21st ICML, Omnipress (2004) 209–216
[7] Cesa-Bianchi, N., Gentile, C., Zaniboni, L.: Hierarchical classification: Combining Bayes with SVM. In: Proc. of the 23rd ICML, ACM Press (2006) 177–184
[8] The Gene Ontology Consortium: Gene ontology: tool for the unification of biology. Nature Genet. **25** (2000) 25–29
[9] Valentini, G., Cesa-Bianchi, N.: Hcgene: a software tool to support the hierarchical classification of genes. Bioinformatics **24** (2008) 729–731
[10] Ben-Hur, A., Noble, W.: Choosing negative examples for the prediction of protein-protein interactions. BMC Bioinformatics **7** (2006)
[11] Finn, R. et al.: The Pfam protein families database. Nucl. Ac. Res. **36** (2008) D281–D288
[12] Eddy, S.: Profile hidden markov models. Bioinformatics **14** (1998) 755–763
[13] Altschul, S., Gish, W., Miller, W., Myers, E., Lipman, D.: Basic local alignment search tool. Journal of Molecular Biology **215** (1990)
[14] Pavlidis, P., Weston, J., Cai, J., Noble, W.: Learning gene functional classification from multiple data. J. Comput. Biol. **9** (2002) 401–411
[15] Spellman, P., et al.: Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomices cerevisiae by microarray hybridization. Mol. Biol. Cell **9** (1998) 3273–3297
[16] Gasch, P., et al.: Genomic expression programs in the response of yeast cells to environmental changes. Mol.Biol.Cell **11** (2000) 4241–4257
[17] Stark, C. et al.: BioGRID: a general repository for interaction datasets. Nucl. Ac. Res. **34** (2006) D535–D539
[18] Lin, H., Lin, C., Weng, R.: A note on Platt's probabilistic outputs for support vector machines. Machine Learning **68** (2007) 267–276
[19] Dietterich, T.: Approximate statistical test for comparing supervised classification learning algorithms. Neural Computation **10** (1998) 1895–1924
[20] Pena-Castillo, L., et al.: A critical assessment of Mus musculus gene function prediction using integrated genomic evidence. Genome Biology **9** (2008)