Fully Associative Ensemble Learning for Hierarchical Multi-Label Classification

Lingfeng Zhang
lzhang34@uh.edu
Shishir K. Shah
sshah@central.uh.edu
Ioannis A. Kakadiaris
ioannisk@uh.edu

Computational Biomedicine Lab
Department of Computer Science
University of Houston
Houston, TX, USA

Abstract

In contrast to traditional flat classification problems (*e.g.*, binary or multi-class classification), Hierarchical Multi-label Classification (HMC) takes into account the structural information embedded in the class hierarchy. In this paper, we propose a local hierarchical ensemble framework, *Fully Associative Ensemble Learning* (FAEL). We model the relationship between each node's global prediction and the local predictions of all the nodes as a multi-variable regression problem. The simplest version of our model leads to a ridge regression problem. It can be extended using the kernel trick, which explores the complex correlation between global and local prediction. In addition, we introduce a binary constraint model to restrict the optimal weight matrix learning. The proposed models have been applied to image annotation and gene function prediction datasets. The experimental results indicate that our models achieve better performance when compared with other baseline methods.

1 Introduction

Hierarchical Multi-label Classification (HMC) is a variant of classification where each sample has more than one label and all these labels are organized hierarchically in a tree or Direct Acyclic Graph (DAG). In reality, HMC can be applied to many different domains [4, 6, 15]. In web page classification, one website with the label "football" could be labeled with a high level label "sport". In image annotation, an image tagged as "outdoor" might have other low level concept labels, like "beach" or "garden". In gene function prediction, a gene can be simultaneously labeled as "metabolism" and "catalytic or binding activities" by the biological process hierarchy and the molecular function hierarchy, respectively.

A rich source of hierarchical information in tree and DAG structures is helpful to improve classification performance. Based on how this information is used, previous HMC approaches can be divided into global (big-bang) or local [16]. Global approaches learn a single model for the whole class hierarchy. Many classic machine learning algorithms have been extended to global approaches. Wang *et al*. [21] used association rules for hierarchical document categorization. Vens *et al*. [19] introduced a modified version of decision tree for HMC. Based on a predictive cluster tree, Dimitrovski *et al*. [6] proposed the cluster-HMC algorithm for medical image annotation. Global approaches enjoy smaller model size because

they build one model for the whole hierarchy. However, they ignore the local modularity, which is an essential advantage of HMC. Local approaches first build local classifiers on each node of the class hierarchy. Then, hierarchical information is aggregated across the local prediction results of all the local classifiers to obtain the global prediction results for all the nodes. We hereafter refer to "local prediction result" and "global prediction result" as "local prediction" and "global prediction", respectively. Dumais and Chen [7] applied a multiplicative threshold to update local prediction. The posterior probability is computed based on the parent-child relationship. Barutcuoglu and DeCoro [2] proposed a Bayesian aggregation model for image shape classification. The main idea is to obtain the most probable consistent set of global predictions. Valentini [18] presented the True Path Rule (TPR) ensembles. In that method, positive local predictions of child nodes affect their parent nodes and negative local predictions of non-leaf nodes affect their descendant nodes.

Previous local approaches suffer from three drawbacks. First, most of them focus only on the parent-child relationship. Other relationships in the hierarchy (*e.g.*, ancestor-descendant, siblings) are ignored. Second, their models are sensitive to local prediction. The error of one local node will propagate to other nodes in the hierarchy. Third, most local methods assume that the local structural constraint between two nodes will be reflected in their local predictions. However, this assumption might be shaken by different choices of features, local classification models and positive-negative sample selection rules [9, 17]. In such situations, previous methods would fail to integrate valid structural information into local prediction.

In this paper, we propose a novel local HMC framework, Fully Associative Ensemble Learning (FAEL). We call it "fully associative ensemble" because in our model the global prediction of each node takes into account the relationships between the current node and all the other nodes. Specifically, a multi-variable regression model is built to minimize the empirical loss between the global predictions of all the training samples and their corresponding true label observations.

Our contributions are: we (i) developed a novel local hierarchical ensemble framework, in which all the structural relationships in the class hierarchy are used to obtain global prediction; (ii) introduced empirical loss minimization into HMC, so that the learned model can capture the most useful information from historical data; and (iii) proposed kernel and binary constraint HMC models.

The rest of this paper is organized as follows: in Section 2 we discuss related work. Section 3 describes the proposed FAEL models. The experimental design, results and analysis are presented in Section 4. Section 5 concludes the paper.

2 Related Work

This work is inspired by both top-down and bottom-up local models. The top-down models propagate predictions from high level nodes to the bottom [7, 13]. In contrast, the bottom-up models propagate predictions from the bottom to the whole hierarchy [3, 10]. As a state-of-the-art method, the TPR ensemble integrates both top-down and bottom-up rules [18]. The global prediction of each parent node is updated by the positive local predictions of its child nodes. Then, a top-down rule is applied to synchronize the obtained global predictions. In contrast to TPR, our model incorporates all pairs of hierarchical relationships and attempts to learn a fully associative weight matrix from training data. Take the "human" sub-hierarchy from the extended IAPR TC-12 image dataset [8] for example, Figure 1 depicts the merits of our model and shows the contributions of different nodes on each local prediction. The weight matrix computed indicates that each local node influences the nodes of the same path

Figure 1: (Top) The "human" sub-hierarchy. (Bottom) The weight matrix W^* learned from B-FAEL. For the non-leaf nodes 1 and 5 in (Top), using TPR, the global predictions are decided by their local prediction and the local predictions (those above threshold 0.5) of the child nodes. Using our model, they are made by the local predictions of all the fourteen non-root nodes. In (Bottom), we can observe that the nodes in the same path give positive weight, the other nodes give negative weight. For the remaining twelve leaf nodes, TPR uses local prediction as global prediction directly. In (Bottom), except for $W^*_{1,10}$, and $W^*_{7,10}$, the nodes in the same path give positive or zero weights, the other nodes give negative or zero weights. These observations are consistent with the fact that each image region is annotated by the labels of one continuous path from the root to the bottom gradually and exclusively.

positively while nodes not directly connected in the hierarchy provide a negative influence. Since the weight matrix of our model is learned based on all the training samples, we can minimize the influence of outlier examples of each node. The learning model also helps to avoid the error propagation problem, because all the global predictions are obtained simultaneously.

The proposed framework also inherits features from Multi-Task Learning (MTL) methods [5, 11, 22, 23]. Our model is close to the MTLs with tree or graph structures, where pre-defined structural information is extracted to fit the learning model [12, 14]. Similar to these MTLs, our hierarchical ensemble model can use various loss functions and regularization terms. One major difference lies in the features used in the model. In the MTLs, the features are shared consistently over all the tasks and they must be the same for each task. In our model, local predictions of all the nodes are used as features. Therefore, each local classifier can be built by completely different features.

3 Fully Associative Ensemble Learning

Let $\mathcal{S} = \{s_1, s_2, ..., s_n\}$ represent a hierarchical multi-label training set, which consists of n samples. Its hierarchical label set is denoted by $\mathcal{C} = \{c_1, c_2, ..., c_l\}$. There are l labels in total, and each label corresponds to one unique node in hierarchy $\mathcal{H}$. The training label matrix is defined as a binary matrix $Y = \{y_{ij}\}$, with size $n \times l$. If the i^{th} sample has the j^{th} label, $y_{ij} = 1$, otherwise $y_{ij} = 0$. As a local approach, local classifiers $\mathcal{F} = \{f_1, f_2, ..., f_l\}$ are built on each node. The local predictions of $\mathcal{S}$ are denoted by matrix $X = \{x_{ij}\}$, where x_{ij} represents the prediction of the i^{th} sample on the j^{th} label. A probabilistic classifier is used as the local learner, so we have $x_{ij} \in [0,1]$. Similarly, we represent the global prediction matrix by $\widehat{Y} = \{\widehat{y}_{ij}\}$ with size $n \times l$. In our model, global prediction is achieved based on local prediction and hierarchical information. To take all the node-to-node relationships into account, we define $W = \{w_{ij}\}$ as a weight matrix, where w_{ij} represents the weight of the i^{th} label's local prediction to the j^{th} label's global prediction. Thus, each label's global prediction is a weighted sum of the local predictions of all the nodes in $\mathcal{H}$. The global prediction matrix $\widehat{Y}$ is computed as: $\widehat{Y} = XW$.

3.1 The Basic Model

The simplest way to estimate the weight matrix W is by minimizing the squared loss between the global prediction matrix $\widehat{Y}$ with the true label matrix Y. To reduce the variance of w_{ij}, we penalize the Frobenius norm of W and obtain the following objective function:

$$\min_W \|Y - XW\|_F^2 + \lambda_1 \|W\|_F^2, \tag{1}$$

where the first term measures the empirical loss of the training set, the second term controls the generalization error, and λ_1 is a regularization parameter. The above function is known as ridge regression. Taking derivatives w.r.t. W and setting to zero, we have:

$$W = \left(X^T X + \lambda_1 I_l\right)^{-1} X^T Y, \tag{2}$$

where I_l represents the $l \times l$ identity matrix. Thus, we obtain an analytical solution for the basic FAEL model.

3.2 The Kernel Model

To capture the complex correlation between global and local prediction, we can generalize the above basic model using the kernel trick. Let Φ represent the map applied to each example's local prediction vector $\mathbf{x}_i$. A kernel function is induced by $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$. By replacing the term X in (1), we obtain:

$$\min_{W_k} \|Y - \Phi W_k\|_F^2 + \lambda_1 \|W_k\|_F^2. \tag{3}$$

After several matrix manipulations [1], the solution of W_k becomes:

$$W_k = \left(\Phi^T \Phi + \lambda_1 I_l\right)^{-1} \Phi^T Y = \Phi^T \left(\Phi \Phi^T + \lambda_1 I_n\right)^{-1} Y, \tag{4}$$

where I_n represents the $n \times n$ identity matrix. For a given testing example s^t and its local prediction $\mathbf{x}^t$, the global prediction $\widehat{\mathbf{y}}^t$ is obtained by $\widehat{\mathbf{y}}^t = \mathbf{x}^t W$. For a kernel version, we obtain:

$$\widehat{\mathbf{y}}_k^t = \Phi(\mathbf{x}^t)W_k = \Phi(\mathbf{x}^t)\Phi^T\left(\Phi\Phi^T + \lambda_1 I_n\right)^{-1} Y = K(\mathbf{x}^t,\mathbf{x})\left(K(\mathbf{x},\mathbf{x}) + \lambda_1 I_n\right)^{-1} Y, \quad (5)$$

where $K(\mathbf{x}^t,\mathbf{x}) = [k(\mathbf{x}^t,\mathbf{x}^1),k(\mathbf{x}^t,\mathbf{x}^2),...,k(\mathbf{x}^t,\mathbf{x}^n)]$ and $K(\mathbf{x},\mathbf{x}) = \{k(\mathbf{x}_i,\mathbf{x}_j)\}$ are both kernel computations.

One potential drawback of the above kernel model is its scalability. During the training phase, the complexity of computing and storing $K(\mathbf{x},\mathbf{x})$ is significant even for moderate size problems. Therefore, we adopt a simple sample-selection technique to reduce the kernel complexity of large-scale datasets. First, we sort the training set based on the difference between each sample's true label observations and the global predictions obtained from the basic model. Then, the top $n_k (n_k \ll n)$ samples with smallest differences are selected to build the kernel model, which reduces the kernel complexity from $O(n \times n)$ to $O(n_k \times n_k)$. Moreover, sample selection is also helpful to exclude outliers.

3.3 The Binary Constraint Model

Another limitation of the basic model is that the weights between different nodes are considered independently. To make full use of the hierarchical relationships between different nodes, we introduce a regularization term to the optimization function in (1).

The hierarchical structure can be viewed as a set of "binary constraints" among all the nodes. Here, we only focus on the "parent-child" constraints and the "ancestor-descendent" constraints. Let $\mathcal{R} = \{r_i(c_p,c_q)\}$ denote the binary constraint set of hierarchy $\mathcal{H}$. Each member $r_i(c_p,c_q)$ meets either $c_p =\uparrow c_q$ or $c_p =\Uparrow c_q$, where " $\uparrow$ " and " $\Uparrow$ " represent the "parent-child" constraint and the "ancestor-descendent" constraint, respectively [16]. The size of $\mathcal{R}$ depends on the structure of $\mathcal{H}$. Its maximum is $l \times (l-1)/2$, which is equal to the number of all the possible constraints. In this case, there is only one path from the root node to the single leaf node in the hierarchy. Now, we introduce a weight restriction to each pair of nodes in $\mathcal{R}$. Define coefficient $m_{pq} \in \mathbb{R}^+$ for the i^{th} pair $r_i(c_p,c_q)$, so that:

$$w_{pk} = m_{pq} * w_{qk}. \quad (6)$$

The intuition behind this definition is that high-level nodes should give larger weights than low-level nodes. For the global prediction of node k, the weight of node p is m_{pq} times the weight of node q. The value of m_{pq} is set by:

$$m_{pq} = \begin{cases} \mu & c_p =\uparrow c_q \\ \mu * (e_{pq}+1) & c_p =\Uparrow c_q \end{cases}, \quad (7)$$

where μ is a positive constant and e_{pq} represents the number of nodes between p and q. Thus, the coefficient of an "ancestor-descendent" constraint is larger than that of a "parent-child" constraint. Specifically, it is decided by the depth difference of the two corresponding nodes in the hierarchy. All the restrictions over the hierarchy are summarized as:

$$\sum_{r_i(c_p,c_q)}^{|\mathcal{R}|} \sum_{k=1}^{l} \left(w_{pk} - m_{pq} * w_{qk}\right)^2. \quad (8)$$

To convert the above equations into a matrix version, we introduce a sparse matrix $M = [\mathbf{m}_1,\mathbf{m}_2,...,\mathbf{m}_{|R|}]^T$, in which the i^{th} row $\mathbf{m}_i$ corresponds to the i^{th} pair in $\mathcal{R}$. Each row in M has only two non-zero entries. The p^{th} entry is 1 and the q^{th} entry is $-m_{pq}$, all the other entries are zero. Thus, we obtain the regularization term of the binary constraint model:

$$\sum_{r_i(c_p,c_q)}^{|\mathcal{R}|}\sum_{k=1}^{l}\left(w_{pk}-m_{pq}*w_{qk}\right)^2=\|MW_b\|_F^2. \tag{9}$$

Adding this term to (1), the optimization function becomes:

$$\min_W \|Y-XW_b\|_F^2+\lambda_1\|W_b\|_F^2+\lambda_2\|MW_b\|_F^2. \tag{10}$$

Taking the derivative w.r.t. W_b, setting to zero, and merging similar terms, we obtain:

$$(X^TX+\lambda_1 I_l+\lambda_2 M^TM)W_b=X^TY. \tag{11}$$

The analytical solution of the binary constraint model is given by:

$$W_b=\left(X^TX+\lambda_1 I_l+\lambda_2 M^TM\right)^{-1}X^TY. \tag{12}$$

The analytical solution ensures a low computational complexity for this model. In practice, we can also choose a small number of rows from M to build the regularization term and focus on a more specific constraint set.

3.4 Hierarchical Prediction

After we get the global predictions of all the nodes, the next step is to set thresholds for the global prediction of each node, and assign proper labels for each testing sample. In the TPR model, the author uses 0.5 as the threshold of all the nodes, which ignores the distribution difference of positive and negative samples. Here, the threshold is learned to separate them averagely. Let $\mathbf{d}=\{d_1,d_2,...,d_l\}$ denote the threshold set of global prediction, where d_i corresponds to node i. Let $\mathcal{S}_i^+$ and $\mathcal{S}_i^-$ represent the positive and negative training sets of node i, respectively. Their global predictions are computed as $\widehat{Y}_i^+$ and $\widehat{Y}_i^-$. We define threshold d_i as the midpoint of the averaged positive and negative global predictions of node i:

$$d_i=0.5*\left(\frac{1}{|\mathcal{S}_i^+|}\sum_j\widehat{y}_{ji}^+-\frac{1}{|\mathcal{S}_i^-|}\sum_j\widehat{y}_{ji}^-\right) \tag{13}$$

where $\widehat{y}_{ji}^+$ and $\widehat{y}_{ji}^-$ represent the global prediction of the j^{th} sample in $\mathcal{S}_i^+$ and $\mathcal{S}_i^-$, respectively.

Based on the learned thresholds, the output labels of each testing sample should be consistent with the hierarchical structure. All the labels with positive output can be linked into one or multiple continuous paths from the root to the bottom in hierarchy $\mathcal{H}$. Here we apply bottom-up strategy to synchronize the output labels. Given a testing sample s^t with global prediction $\widehat{\mathbf{y}}^t=[\widehat{y}_1^t,\widehat{y}_2^t,...,\widehat{y}_l^t]$, its final output $\mathbf{o}^t=[o_1^t,o_2^t,...,o_l^t]$ is decided by:

$$o_i^t=\begin{cases}1 & \widehat{y}_i^t>d_i\\ 1 & \widehat{y}_k^t>d_k, c_i=\uparrow c_k \text{ or } \Uparrow c_k\\ 0 & otherwise\end{cases}. \tag{14}$$

Note that from the above rule, we might obtain multiple valid paths as final output. It is appropriate for some applications such as gene function prediction where each gene can have more than one path in the "FunCat" hierarchy. However, in other applications such as image annotation, the ideal output is one path of the conceptual hierarchy that indicates the exact content of each image region. In this case, we average the global predictions on each continuous path and return the maximum path. The pseudo-code of the proposed framework is summarized in Algorithm 1.

Algorithm 1: The Fully Associative Ensemble Learning

Input: $\mathcal{S}^r = \{s_1^r, s_2^r, ..., s_n^r\}$, $\mathcal{C} = \{c_1, c_2, ..., c_l\}$, $\mathcal{H}$, $Y^r = \{y_{ij}^r\} \in \mathbb{R}^{n \times l}$ and $\mathcal{S}^t = \{s_1^t, s_2^t, ..., s_m^t\}$

Output: $\widehat{Y}^t = \{\widehat{y}_{ij}^t\} \in \mathbb{R}^{m \times l}$ and $O^t = \{o_{ij}^t\} \in \mathbb{R}^{m \times l}$

for $i \leftarrow 1$ ***to*** l **do**
  Select positive and negative examples for node i
  Build a local classifier f_i on node i
  Compute the local prediction of $\mathcal{S}^r$ on node i, $f_i(\mathcal{S}^r)$
Select binary constraint pairs and obtain M
Compute W with (2), (12) and (4)
Compute $\mathbf{d}$ for all the nodes with (13)
for $i \leftarrow 1$ ***to*** m **do**
  Compute the local prediction of s_i^t on each node, $\mathbf{x}_i^t = f(s_i^t)$
  Compute the global prediction of s_i^t with $\widehat{\mathbf{y}}_i^t = \mathbf{x}_i^t \times W$ and (5)
  Compute the final output with (14)
return $\{\widehat{Y}^t, O^t\}$;

Figure 2: Sample image with hierarchical annotations.

4 Experiments

4.1 Image Annotation

In this section, we present our evaluation of the proposed models on the extended IAPR TC-12 image collection [8]. In this dataset, every image is segmented into several regions and each region is annotated by a set of labels from a conceptual hierarchy. Figure 2 depicts a sample image and its corresponding labels. The whole conceptual hierarchy consists of 275 nodes which are located in six main branches: "animal", "landscape", "man-made", "human", "food" and "other". Considering their conceptual difference and hierarchy size, we build five separate sub-hierarchies with the first five main branches. Their detailed descriptions are shown in Table 1. The "other" branch is excluded because it has only six child nodes with the same depth.

Given the original features from the dataset, each region is viewed as a sample. In order to build three-fold cross-validation, we ignore the nodes that have less than ten samples. Based on [8], we use Random Forests as the basic classifier under the one-versus-all sample selection technique. The number of trees in Random Forests is set to 100. In our models, λ_1 is set to 0.5. We choose Gaussian kernel in the Kernel FAEL model (K-FAEL). The

Sub-hierarchy	Number of samples	Number of nodes	Depth of tree
animal	1,999	41	5
food	861	5	3
human	17,011	14	4
landscape	45,048	42	4
man-made	33,984	99	5

Table 1: The extended IAPR TC-12 sub-hierarchy descriptions.

Models	F-measure					Hierarchical F-measure				
	animal	food	human	landscape	man-made	animal	food	human	landscape	man-made
TD	0.129	0.345	0.233	0.264	0.067	0.319	0.375	0.605	0.501	0.179
TPR	0.138	0.345	0.234	0.274	0.073	0.327	0.375	0.605	0.504	0.186
TPR-w	0.140	0.345	0.234	0.274	0.075	0.329	0.375	0.605	0.504	0.189
FAEL	0.211	0.397	0.303	0.348	0.133	0.410	0.463	0.624	0.566	0.269
K-FAEL	0.266	**0.408**	0.310	0.331	0.147	0.436	**0.473**	**0.632**	**0.588**	0.307
B-FAEL	**0.290**	0.397	**0.350**	**0.388**	**0.199**	**0.489**	0.469	0.626	0.579	**0.382**

Table 2: F-measure and Hierarchical F-measure results on the image annotation dataset.

parameter σ is set to 0.05. Additional experiments not detailed in this paper indicate that our models are not sensitive to the choices of λ_1 and σ. In K-FAEL, we apply the sample selection technique to the training sets with more than 5,000 samples (n_k is set to 5,000). To test the performance of the Binary constraint FAEL model (B-FAEL), we set λ_2 to different values $\lambda_2 = \{0, 10, 20, ...100\}$, μ is set to 2. It is obvious that B-FAEL would degenerate to FAEL when λ_2 is equal to 0. We compare the proposed models with the Top-Down (TD) algorithm, TPR and weighted TPR (TPR-w) [18] under F-measure and Hierarchical F-measure. The results are summarized in Table 2 (B-FAEL with the best choice of λ_2). Figure 3 depicts the performance of B-FAEL with respect to λ_2.

From Table 2 we can observe that the proposed models perform better than other HMC algorithms. Under F-measure, B-FAEL achieves the best results on four sub-hierarchies. In the "food" hierarchy, the result of K-FAEL (0.408) is better than that of B-FAEL (0.397). As we know, the classic F-measure is designed for unstructured flat classification problems. Here, it evaluates the average prediction performance of all the nodes. By integrating structural information of prediction, Hierarchical F-measure is a more appropriate performance metric in HMC [18, 20]. Under Hierarchical F-measure, K-FAEL performs better on the two small size hierarchies ("food" and "human") and one moderate size hierarchy ("landscape"). B-FAEL performs better on the other moderate size hierarchy ("animal") and the large size hierarchy ("man-made"). The main reason is that B-FAEL can be limited by the number of

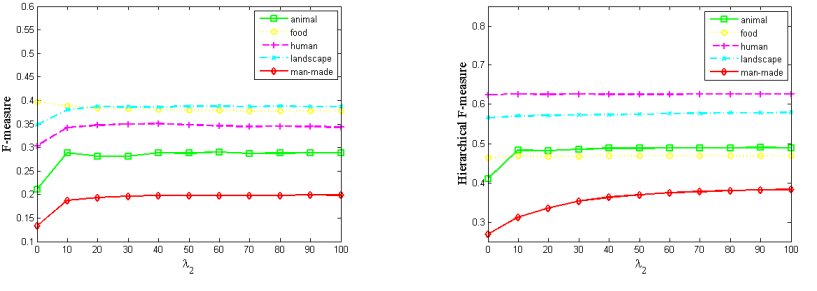

Figure 3: B-FAEL performance on the image annotation dataset.

Datasets	Description	A	B	C	D
Pfam-1	protein domain binary data from Pfam data	3,529	4,950	211	5
Pfam-2	protein domain log E data from Pfam data	3,529	5,724	211	5
Expr	gene Expression data	4,532	250	230	5
PPI-BG	PPI data from BioGRID	4,531	5,367	232	5
PPI-VM	PPI data from Von Mering experiments	2,338	2,559	177	5
SP-sim	Sequence Pairwise similarity data	3,527	6349	211	5

Table 3: The gene function dataset descriptions. Columns A, B, C, D represent number of samples, number of features, number of nodes and depth of tree, respectively.

Models	F-measure						Hierarchical F-measure					
	Pfam-1	Pfam-2	Expr	PPI-BG	PPI-VM	SP-sim	Pfam-1	Pfam-2	Expr	PPI-BG	PPI-VM	SP-sim
TD	**0.404**	0.206	0.062	0.269	0.359	0.249	0.412	0.341	0.117	0.323	0.398	0.425
TPR	0.362	0.156	0.070	0.234	0.261	0.131	0.308	0.268	0.170	0.267	0.280	0.226
TPR-w	**0.404**	0.220	0.077	0.295	0.356	0.254	0.413	0.370	0.178	0.349	0.400	**0.447**
FAEL	0.395	0.303	0.135	0.278	0.394	0.339	0.442	0.429	0.533	0.445	0.466	0.362
K-FAEL	0.398	**0.346**	**0.154**	**0.345**	**0.401**	**0.347**	0.443	0.454	0.515	0.451	0.474	0.376
B-FAEL	0.395	0.303	0.135	0.278	0.394	0.339	**0.457**	**0.498**	**0.596**	**0.543**	**0.477**	0.406

Table 4: F-measure and Hierarchical F-measure results on the gene function datasets.

constraints in small and moderate size hierarchies. In Figure 3, B-FAEL improves both F-measure and Hierarchical F-measure performance on four sub-hierarchies. As λ_2 increases, the performance first goes up and then becomes stable after reaching a peak. With a small hierarchy size of 5 nodes, the performance on the "food" hierarchy is basically unchanged.

4.2 Gene Function Prediction

Gene function prediction is another complex HMC problem. We use six yeast datasets integrated in [18]. Their descriptions are summarized in Table 3. To compare with the results of TD, TPR and TPR-w in [18], we use the same experimental settings. The results are summarized in Table 4. Figure 4 depicts the performance of B-FAEL with respect to λ_2.

As we can observe from Table 4, the proposed models obtain better or competitive performance in all the gene function datasets. Under flat measurement F-measure, K-FAEL achieves the best results in five data sets. In the Pfam-1 dataset, the result of K-FAEL (0.398) is very close to the best result (0.404) achieved by TD and TPR-w. Under Hierarchical F-measure, our models improve the performance on five datasets. In the SP-sim dataset, the results of our models (0.362, 0.376 and 0.406) are a little worse than that of TPR-w (0.447).

From Figure 4 we observe that, compared with FAEL and K-FAEL, the B-FAEL model

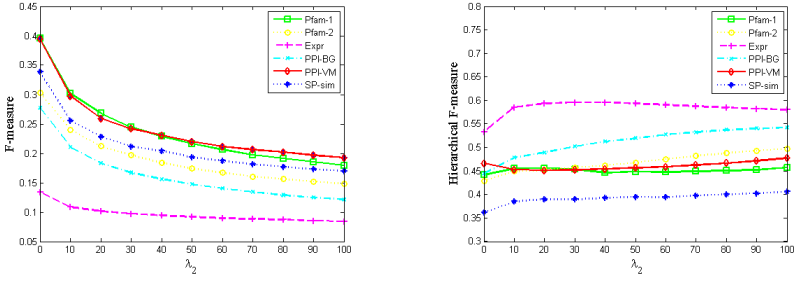

Figure 4: B-FAEL performance on the gene function datasets.

achieves better performance in Hierarchical F-measure. On the other hand, as λ_2 becomes larger, the F-measure performance of B-FAEL is worse than that of FAEL and K-FAEL. There are two reasons. First, the binary constraint model enforces the hierarchical consistency, which might weaken the independent discriminative ability of some nodes. Second, the "FunCat" hierarchy has large size and high complexity. With the given features, the binary constraint model cannot optimize both flat and hierarchical performance.

5 Conclusion

This paper introduces a novel HMC framework. We build a multi-variable regression model between the global and local predictions of all the nodes. The basic model is extended to the kernel model and the binary constraint model. Our work raises a number of questions that we plan to address in the future. For example, how to choose a limited number of hierarchical constraints to build an effective binary constraint model and how to determine a better threshold for global prediction.

Acknowledgements

This research was funded in part by the US Army Research Lab (W911NF-13-1-0127) and the UH Hugh Roy and Lillie Cranz Cullen Endowment Fund. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of the sponsors.

References

[1] S. An, W. Liu, and S. Venkatesh. Face recognition using kernel ridge regression. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, Minneapolis, MN, USA, 2007.

[2] Z. Barutcuoglu and C. DeCoro. Hierarchical shape classification using bayesian aggregation. In *Proc. IEEE International Conference on Shape Modeling and Applications*, Matsushima, Japan, 2006.

[3] P. N. Bennett and N. Nguyen. Refined experts: Improving classification in large taxonomies. In *Proc. ACM/SIGIR International Conference on Research and Development in Information Retrieval*, pages 11–18, Boston, MA, USA, 2009.

[4] N. Cesa-Bianchi, M. Re, and G. Valentini. Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. *Machine Learning*, 88(1-2):209–241, 2012.

[5] A. Charuvaka and H. Rangwala. Multi-task learning for classifying proteins using dual hierarchies. In *Proc. IEEE International Conference on Data Mining*, pages 834–839, Brussels, Belgium, 2012.

[6] I. Dimitrovski, D. Kocev, S. Loskovska, and S. Džeroski. Hierarchical annotation of medical images. *Pattern Recognition*, 44(10):2436–2449, 2011.

[7] S. Dumais and H. Chen. Hierarchical classification of web content. In *Proc. ACM/SIGIR International Conference on Research and Development in Information Retrieval*, pages 256–263, Athens, Greece, 2000.

[8] H. J. Escalante, C. A. Hernández, J. A. Gonzalez, A. López-López, M. Montes, E. F. Morales, L. E. Sucar, L. Villaseñor, and M. Grubinger. The segmented and annotated IAPR TC-12 benchmark. *Computer Vision and Image Understanding*, 114(4):419–428, 2010.

[9] T. Fagni and F. Sebastiani. On the selection of negative examples for hierarchical text categorization. In *Proc. Language and Technology Conference*, pages 24–28, Poznań, Poland, 2007.

[10] Y. Guan, C. L. Myers, D. C. Hess, Z. Barutcuoglu, A. Caudy, and O. G. Troyanskaya. Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome Biology*, 9(Suppl 1):S3, 2008.

[11] L. Jacob, F. Bach, and J. P. Vert. Clustered multi-task learning: A convex formulation. In *Proc. Annual Conference on Neural Information Processing Systems*, pages 745–752, Vancouver, B.C., Canada, 2008.

[12] S. Ji, L. Yuan, Y. Li, Z. Zhou, S. Kumar, and J. Ye. Drosophila gene expression pattern annotation using sparse features and term-term interactions. In *Proc. ACM/SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 407–416, Paris, France, 2009.

[13] X. Jiang, N. Nariai, M. Steffen, S. Kasif, and E. Kolaczyk. Integration of relational and hierarchical network information for protein function prediction. *BMC Bioinformatics*, 9(1):350, 2008.

[14] S. Kim and E. P. Xing. Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eQTL mapping. *The Annals of Applied Statistics*, 6(3):1095–1117, 2012.

[15] T. Li, S. Zhu, and M. Ogihara. Hierarchical document classification using automatically generated hierarchy. *Journal of Intelligent Information Systems*, 29(2):211–230, 2007.

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

[17] C. N. Silla and A. A. Freitas. Novel top-down approaches for hierarchical classification and their application to automatic music genre classification. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, pages 3499–3504, San Antonio, Texas, USA, 2009.

[18] G. Valentini. True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8 (3):832–847, 2011.

[19] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185–214, 2008.

[20] K. Verspoor, J. Cohn, S. Mniszewski, and C. Joslyn. A categorization approach to automated ontological function annotation. *Protein Science*, 15(6):1544–1549, 2006.

[21] K. Wang, S. Zhou, and Y. He. Hierarchical classification of real life documents. In *Proc. SIAM International Conference on Data Mining*, pages 1–16, Chicago, IL, USA, 2001.

[22] J. Zhou, L. Yuan, J. Liu, and J. Ye. A multi-task learning formulation for predicting disease progression. In *Proc. ACM/SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 814–822, San Diego, CA, USA, 2011.

[23] J. Zhou, J. Liu, A. N. Vaibhav, and J. Ye. Modeling disease progression via multi-task learning. *NeuroImage*, 78(0):233–248, 2013.