Functional Annotation of Genes Using Hierarchical Text Categorization

Svetlana Kiritchenko[1], **Stan Matwin**[1,3], **and A. Fazel Famili**[2]

[1]University of Ottawa, Canada, {svkir, stan}@site.uottawa.ca
[2]National Research Council Canada, Fazel.Famili@nrc-cnrc.gc.ca
[3]Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

ABSTRACT

This paper addresses the task of functional annotation of genes from biomedical literature. We view this task as a hierarchical text categorization problem with Gene Ontology as a class hierarchy. We present a novel global hierarchical learning approach that takes into account the semantics of a class hierarchy. This algorithm with AdaBoost as the underlying learning procedure significantly outperforms the corresponding "flat" approach, i.e. the approach that does not consider any hierarchical information. In addition, we propose a novel hierarchical evaluation measure that gives credit to partially correct classification and discriminates errors by both distance and depth in a class hierarchy.

1. INTRODUCTION

In many genomics studies one of the major steps is the gene expression analysis using high-throughput DNA microarrays. Traditionally, most computational research on analyzing gene expression data has focused on working with microarray data alone, using statistical or data mining tools. However, raw gene expression data are very hard to analyze even for an experienced scientist. On the other hand, there exists a wealth of information pertaining to the function and behavior of genes, described in papers and reports. This information could potentially be useful in the analysis of gene expression, if we had a way of combining it synergistically with the knowledge acquired from the microarray data experiments. Specifically, our research is aimed at providing molecular biologists with known functional information on genes used in the experiments in order to make microarray results and their analysis more biologically meaningful. At the same time, this functional information can also be used to partially validate the new findings.

Even though many genes for well-studied organisms, such as Escherichia coli or Saccharomyces cerevisiae, have been already annotated in specialized databases (EcoCyc, SGD), information on many other genes currently can be found only in scientific publications. Public databases are created and curated manually; thus, they cannot keep up with an overwhelming number of new discoveries published on a daily basis. Furthermore, these databases often use different vocabularies to describe gene functionality, which raises an additional challenge for integrating the results. Consequently, genomics databases are not always adequate to find the requisite information. Therefore, we need to apply text mining and categorization techniques to retrieve up-to-date information from biomedical literature and translate it into a standardized vocabulary to help life scientists in their everyday activities. Moreover, the same process can be used as a tool to assist in updating and curating databases.

In this work, we view the functional annotation task as a text categorization task where we classify biomedical articles describing the functionality of a given gene into one or several functional classes from Gene Ontology (GO) [1]. Since GO is not just a flat set of categories, but a hierarchy by its nature, we must turn to the hierarchical text categorization framework to realize the goal of this research. An immediate observation is that unlike the most used, "flat" text classification framework, the area of hierarchical classification has received little attention. We would like to fill in this gap and bring the benefits of hierarchical text categorization to genomics in general and gene function identification in particular.

2. FUNCTIONAL ANNOTATION USING TEXT CATEGORIZATION

We propose a system to classify genes/gene products into Gene Ontology codes based on the classification of documents from the Medline library that describe the genes. The purpose of this task is to retrieve the known functionality of a group of genes from the literature and translate it into a controlled vocabulary. Our system realizes a statistical approach to the problem, which does not require immense effort from domain experts. For training, we collect data making use of gene annotations available from genomics databases, such as SGD, MGD, etc. (for more details see [2]).

For the past few years, the functional annotation from texts has been the focus of several studies. Raychaudhuri et al. [4] proposed a straightforward technique of

applying standard machine learning algorithms to this problem and showed promising results. Last year, the BioCreative competition had a similar task (task 2)[1], where a number of NLP as well as machine learning systems participated. However, none of the mentioned works, except [6], addressed this problem as a hierarchical one. We believe that hierarchical techniques are more appropriate for these settings. They can explore the semantics of a class hierarchy improving the performance of learning systems. At the same time, they allow a trade off between classification precision and the required level of details on gene functionality.

3. HIERARCHICAL TEXT CATEGORIZATION

3.1 Hierarchical consistency

We begin the description of the hierarchical categorization techniques employed to address the functional annotation task with the introduction of a new notion of hierarchical consistency. Hierarchical consistency takes into account the semantics of a class hierarchy (such as GO) and is intended to make the classification results more comprehensible for end users. Since hierarchies are mostly designed in the way that lower level categories are specialization of higher level categories, which is represented by transitive relations, such as "is-a" and "part-of", we can assume that an instance belonging to a category also belongs to all ancestor nodes of that category. Therefore, we would like a classifier explicitly assign all the relevant labels, including the ancestor labels, to a given instance. In this way, the assigned labels would clearly indicate the position of an instance in a category hierarchy. Thus, we expect any hierarchical classification algorithm to produce labeling consistent with a given class hierarchy.

Definition (Hierarchical consistency). A label set C_i assigned to an instance d_i is called consistent with a given hierarchy if C_i forms a connected "proper" subgraph of the hierarchy graph rooted in the top node, i.e. if $c_k \in C_i$ and $c_j \in Ancestors(c_k)$, then $c_j \in C_i$.

We assume that every instance belongs to the root of a class hierarchy; therefore, from now on we will always exclude the root node from any ancestor set since including it does not provide any additional information on the instance.

3.2 Hierarchical global learning algorithm

Hierarchical classification methods can be divided in two types: *local* (or top-down level-based) and *global* (or big-bang). A *local* approach builds separate classifiers for each internal node of a hierarchy. A local classifier usually proceeds in a top-down fashion first picking the most relevant categories of the top level and then recursively making the choice among the low-level categories, children of the relevant top-level categories. This method naturally produces consistent labeling, since we classify an instance into a category only if we have already classified it into the parent category at the previous classification step. In a *global* approach only one classifier is built to discriminate all categories in a hierarchy simultaneously. It is similar to the "flat" approach except it somehow takes into account the relationships between the categories in a hierarchy. In many situations, one classifier produced by a global approach is easier to maintain and to interpret by end users than a bunch of classifiers produced by a local method. Moreover, a global approach is capable of avoiding uninformed decisions on categories with a very small number of training instances while a local method is basically forced to make a decision at every level of a hierarchy as far as leaf classes. Unlike the local approach, a global learning algorithm has to be specifically designed to produce consistent classification.

We propose such a hierarchical global approach to learn classifiers that produce consistent labeling on unseen instances. We use this approach in combination with a state-of-the-art learning algorithm AdaBoost.MH [5], a boosting method designed for multi-class multi-label problems. The new hierarchical method is simple and effective and can be applied to any categorization task with a class hierarchy represented as a directed acyclic graph (DAG). The main idea of the algorithm is to transform an initial (possibly single-label) task into a multi-label task by expanding the label set of each training example with the corresponding ancestor labels. This data modification forces a learning algorithm to focus on high level categories by providing a large number of training examples for those categories. The correct classification of unseen instances into high level categories is very important in hierarchical categorization since high level categories define the most general functional classes for genes.

Overall, the algorithm consists of three steps:

1. Transformation of training data making them consistent with a given class hierarchy;
2. Application of a regular learning algorithm on a multi-label dataset;
3. Re-labeling of inconsistently classified test instances.

On the first step, we replace each example (d_i, C_i), $d_i \in D$, $C_i \subseteq C$, with $(d_i, \hat{C}_i)$, where $\hat{C}_i = \{\bigcup_{c_k \in C_i} Ancestors(c_k)\}$. Then, we apply a regular learning algorithm, in our case AdaBoost.MH, on the modified multi-label dataset. Since we train a classifier on the consistent data, we expect that most test instances would be labeled consistently as well. However, it is not guaranteed. Some of the test instances can end up with inconsistent labels. This happens if we assign some class

[1]http://www.pdg.cnb.uam.es/BioLINK/workshop_BioCreative_04/handout/index.html

A to an instance, but do not assign one of its ancestor classes. For such instances we need to do the third post-processing step. At this step we re-label the instances in a consistent manner by considering the confidence in the predictions for class A and all its ancestor classes. For example, if the average of these confidences is high enough, then we would label the instance with class A and all its ancestor classes; if it is not, we do not assign class A to the instance.

3.3 Hierarchical evaluation measure

Most researchers evaluate hierarchical classification systems based on standard "flat" measures: accuracy/ error and precision/recall. However, these measures are not suitable for hierarchical categorization since they do not differentiate among different kinds of misclassification errors. A widely-used hierarchical measure based on the notion of distance overcomes this problem. However, it has some drawbacks. First, it is not easily extendable to DAG hierarchies (where multiple paths between two categories can exist) and multi-label tasks. Second, it does not change with depth. Misclassification into a sibling category of a top level node and misclassification into a sibling of the node 10-level deep are considered the same type of error (distance of 2). However, an error at the 10th level seems a lot less harmful than an error at the top level.

Recently, a new measure of semantic similarity specifically designed for Gene Ontology has been introduced [3]. It takes into account the specificity of a GO term, which is estimated through its probability of usage in gene annotations. The similarity of two terms is calculated as the minimum of the probabilities of their common ancestors. Since many pairs of terms would share the same ancestor nodes and, therefore, have the same semantic similarity, this measure has little discriminative power to be used as an evaluation measure.

To express the desired properties of a hierarchical evaluation measure (HM), we formulate the following requirements:

1. The measure gives credit to partially correct classification, e.g. misclassification into node I when the correct category is G (Figure 1) should be penalized less than misclassification into node D since I is in the same subgraph as G and D is not.

2. The measure punishes distant errors more heavily:

a) the measure gives higher evaluation for correctly classifying one level down comparing to staying at the parent node, e.g. classification into node E is better than classification into its parent C since E is closer to the correct category G;

b) the measure gives lower evaluation for incorrectly classifying one level down comparing to staying at the parent node, e.g. classification into node F is worse than classification into its parent C since F is farther away from G.

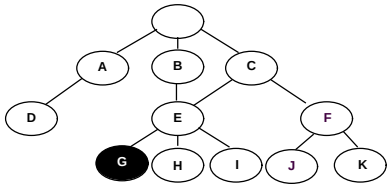

Figure 1: A sample DAG class hierarchy. The solid ellipse G represents the real category of an instance.

3. The measure punishes errors at higher levels of a hierarchy more heavily, e.g. misclassification into node I when the correct category is its sibling G is less severe than misclassification into node C when the correct category is its sibling A.

Seeing that previous measures do not satisfy all of the requirements, we propose a new hierarchical evaluation measure. The new measure is the pair precision and recall with the following addition: each example belongs not only to its class, but also to all ancestors of the class in a hierarchical graph, except the root (we exclude the root of the graph, since all examples belong to the root by default). We call the new measures hP (hierarchical precision) and hR (hierarchical recall).

Formally, in the multi-label settings, for any instance (d_i, C_i) classified into subset C'_i we extend sets C_i and C'_i with the corresponding ancestor labels: $\hat{C}_i = \{\bigcup_{c_k \in C_i} Ancestors(c_k)\}$, $\hat{C}'_i = \{\bigcup_{c_k \in C'_i} Ancestors(c_k)\}$. Then, we calculate (micro-averaged) hP and hR as follows:

$$hP = \frac{\sum_i |\hat{C}_i \cap \hat{C}'_i|}{\sum_i |\hat{C}'_i|} \qquad hR = \frac{\sum_i |\hat{C}_i \cap \hat{C}'_i|}{\sum_i |\hat{C}_i|}$$

For example, suppose an instance is classified into class F while it really belongs to class G (Figure 1). To calculate our hierarchical measure, we extend the set of real classes $C_i = \{G\}$ with all ancestors of class G: $\hat{C}_i = \{B, C, E, G\}$. We also extend the set of predicted classes $C'_i = \{F\}$ with all ancestors of class F: $\hat{C}'_i = \{C, F\}$. So, class C is the only correctly assigned label from the extended set: $|\hat{C}_i \cap \hat{C}'_i| = 1$. There are $|\hat{C}'_i| = 2$ assigned labels and $|\hat{C}_i| = 4$ real classes. Therefore, we get $hP = \frac{|\hat{C}_i \cap \hat{C}'_i|}{|\hat{C}'_i|} = \frac{1}{2}$ and $hR = \frac{|\hat{C}_i \cap \hat{C}'_i|}{|\hat{C}_i|} = \frac{1}{4}$.

We also can combine the two values hP and hR into one hF-measure:

$$hF_\beta = \frac{(\beta^2 + 1) \cdot hP \cdot hR}{(\beta^2 \cdot hP + hR)}, \beta \in [0, +\infty)$$

In our experiments we used $\beta = 1$, giving precision and recall equal weights.

The new hierarchical measure satisfies all three requirements for hierarchical evaluation measures listed above. In addition, the measure is easy to compute; it is based solely on a given hierarchy, so no parameter tuning is required. Furthermore, it is formulated for

Table 1: Comparison of "flat", hierarchical local, and hierarchical global AdaBoost. Numbers in bold are significantly better with 99% confidence.

dataset	depth	out-degree	boost. iter.	hF_1 measure		
				"flat"	local	global
medline_P	12	5.41	500	15.06	**59.27**	**59.31**
medline_F	10	10.29	500	8.78	**43.36**	38.17
medline_C	8	6.45	500	44.18	72.07	**73.35**

a general case of multi-label classification with a DAG class hierarchy.

4. RESULTS

We have composed 3 datasets for a task of predicting gene functions from biomedical literature. These datasets correspond to three aspects of Gene Ontology: biological process (P), molecular function (F), and cellular component (C). We used the Saccharomyces Genome Database (SGD) to collect training instances on yeast genes. All articles were pre-processed: stop words were removed, remaining words were stemmed and converted into binary attributes (a stem is present or not). "medline_P" dataset contains 3305 documents, 2793 attributes, and 1025 categories organized in a 12-level hierarchy. "medline_F" dataset contains 2468 documents, 2448 attributes, and 1078 categories organized in a 10-level hierarchy. "medline_C" dataset contains 2284 documents, 2957 attributes, and 331 categories organized in a 8-level hierarchy. Experiments were run on 10 random training/test splits (in proportion 2:1) for each dataset.

We compare the performance of the new hierarchical global AdaBoost with the corresponding "flat" approach as well as the hierarchical local method (Table 1). The "flat" algorithm, i.e. standard AdaBoost, does not take into account any hierarchical information. Evidently, both hierarchical approaches significantly outperform standard AdaBoost. The differences are more pronounced for larger hierarchies. On these biological data, where the number of classes is very large and the number of training instances per class is very small, the "flat" algorithm suffers a lot producing very poor results. At the same time, the hierarchical methods benefit from assembling more training data and therefore learning more accurate classifiers for high level categories, which are favored by the hierarchical evaluation measure.

Both hierarchical local and hierarchical global algorithms show comparable performances. The global approach explores all the categories simultaneously predicting only labels with high confidence scores. The local method, on the other hand, is forced to make classification decisions at each internal node of a hierarchy, in general, pushing all instances deep down. On the biological data, where instances can belong to intermediate nodes, this means additional errors for the local method. Increase in depth (d) of a hierarchy raises exponentially the number of classes ($\sim k^d$) and, as a result, the difficulty of the classification task for the global approach. Moreover, increase in out-degree (k) only slightly (linearly) complicates the task for the local method while adding a significant number of categories ($\sim k^{d-1}$) to the global method. This is reflected in the loss of the global algorithm on the highly "bushy" "medline_F" data.

5. FUTURE WORK

In this paper, we present a novel approach to automatic gene annotation from biomedical literature using hierarchical text categorization. This work will be continued by conducting experiments on other species (mice, human, etc.), by extending the training data with similar articles from Medline, and by including background knowledge, such as gene aliases, MeSH terms, etc.

Another direction for future research is incorporating assigned GO codes into gene clustering. One of the main challenges in the gene expression analysis is including background knowledge to produce more meaningful clusters of genes not only with similar expression profiles, but also with common functionalities. We can include the gene function information that we get at the classification step as such background knowledge in the clustering process.

6. REFERENCES

[1] M. Ashburner et al. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, 25(1):25–29, 2000.

[2] S. Kiritchenko, S. Matwin, and A. Famili. Hierarchical Text Categorization as a Tool of Associating Genes with Gene Ontology Codes. In *Proc. of the 2nd European Workshop on Data Mining and Text Mining for Bioinformatics*, pages 26–30, 2004.

[3] P. Lord, R. Stevens, A. Brass, and C. Goble. Investigating Semantic Similarity Measures Across the Gene Ontology: the Relationship between Sequence and Annotation. *Bioinformatics*, 19(10):1275–1283, 2003.

[4] S. Raychaudhuri, J. Chang, P. Sutphin, and R. Altman. Associating Genes with Gene Ontology Codes Using a Maximum Entropy Analysis of Biomedical Literature. *Genome Research*, 12:203–214, 2002.

[5] R. Schapire and Y. Singer. Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning*, 37:297–336, 1999.

[6] K. Verspoor et al. Protein Annotation as Term Categorization in the Gene Ontology. In *Proc. of BioCreative Workshop*, 2004.