Pattern-based Customized Learning for Categorization Task in TREC Genomics Track *

Wai Lam Yiqiu Han Ki Chan
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
Shatin
Hong Kong
{wlam,yqhan,kchan}@se.cuhk.edu.hk

Abstract

Our group participated in the categorization task in the TREC Genomics Track 2005, where biological literatures have to be categorized into four types of information. Our approach to this problem adopts customized learning methods in model learning and document categorization. Our pattern-based learning approach can discover useful patterns for tackling categorization challenges.

1 Overview

The rapidly evolving amount of biological articles raises the needs of automatic knowledge management techniques for organizing and cooperating the articles. How to organize the articles and collaborate them with databases is one of the actively investigating issues. Our group participated in the categorization task in the TREC Genomics Track 2005. For the categorization task, it tackles the issue of categorizing biological literatures for facilitating the curation process of literatures to the corresponding databases or controlled vocabularies. Together with the Gene Ontology(GO) annotation, decisions on whether the documents are worth for curation have to be made on documents according to the four types of information.

*The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Nos: CUHK 4179/03E and CUHK4193/04E), the Direct Grant of the Faculty of Engineering, CUHK (Project Code: 2050363), and CUHK Strategic Grant (No: 4410001). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

Hence, the problem is formulated as a categorization problem, where documents are to be categorized into four categories, namely, GO annotation, tumor biology, embryologic gene expression, and alleles of mutant phenotypes.

As four types of decisions have to be made, this task can be formulated as four binary classification tasks, one for each type of information. Each of the binary classification is to decide whether a document is worth for curation in respect to that type of information, for example, the decision on whether a document is worth for curation for the Gene Ontology or not.

Feature engineering is a crucial issue towards the quality of classification results. Context organization is usually pervaded in textual documents for expressing concepts and relations. When an author writes, he or she would normally have an organization in mind to present their information. As an example, an author may express an idea or a concept within a certain context window and in which with some keynotes are closely adjoined together. Our aim is to decide whether a document contains valuable information with regards to gene products, hence, the key concept of the literatures is about gene products. As authors usually express ideas around these key concepts, all other features can be regarded as associated to the gene products with different extents. Hence, we investigate this context association conveyed in the biomedical literatures by utilizing the proximity relationship between the key concepts, target gene(s), and term features. Features appearing more distant to the gene tends to be less closely related with it. Therefore, we preprocess the articles for differentiating the gene names and locating all the variants of the gene names. Gene names are located in the articles through string matching with a rule-based expanded synonym list of genes from the MGI database of mouse, human, and rat. Although documents are mainly from the MGI database, synonyms from other species may also be adopted in expressing concepts. The documents are further preprocessed by removing stopwords and filtered terms with frequency less than 5. The features are then selected according to their information gain, and tf-idf weighting taking the proximity of features towards the gene names into consideration is used in our feature engineering.

Our learning approach adopts customized learning methods for model learning and document categorization. Two learning methods are investigated, namely, k-nearest neighbour (kNN) [4] and pattern-based learning. kNN is a lazy-learning approach, whose aim is to identify the most highly associated documents focusing on the documents to be classified [1].

A majority voting scheme is adopted so as to evaluate the strength of association between documents and categories. The number of neighbours and the threshold for our kNN approach are automatically tuned and applied on the categorization of testing data. The pattern-based learning approach attempts to discover useful patterns for categorization in a customized manner [5].

Our groups have submitted 3 runs to the triage task. The first run was conducted by kNN whereas the other two runs were conducted by pattern-based learning. Details of the pattern-based learning and the results will be presented in the following sections.

2 Pattern-based Customized Learning with Sparse Instances and Excessive Attributes

The pattern based learning approach we used is a simplified variant of our previous customized learning model, CSPL [2, 3]. This variant, called CSPL-A, is designed to handle learning problems with small number of training instances and large number of attributes. Because common instance-based pattern learning methods have great difficulties in dealing with sparse positive instances. Particularly, for problems with a large attribute dimension such as text categorization, the patterns generalized from the training instances under this circumstance usually have an extraordinarily large set of attribute values. Thus they have little practical usage due to the fact that they are too specific to be applied to unseen instances. In other words, the generalization in the learning process cannot work well.

CSPL-A extends the definition of useful patterns as a combination of attributes and instances, it takes the large number of attributes as an advantage for learning rather than conducts feature selection. Each attribute value is regarded as a small piece of information source and CSPL-A can utilize as many attributes as possible rather than to pre-prune features too early.

Suppose the query instance, i.e., the unseen instance to be classified, is denoted by t, the kth training instance is denoted by d^k, then their ith attribute value is denoted by t_i and d_i^k. We use a binary C^j to indicate whether or not d^j belongs to the target class C. Then the likelihood score of the query instance belonging to the target class C is given by

CSPL-A as follows:

$$\Omega(A) = \sum_{j=1}^{m} \sum_{i=1}^{n} \frac{C^j d_i^j t_i \sum_{k=1}^{m} C^k d_i^k}{\sum_{k=1}^{m} d_i^k} \tag{1}$$

To decide whether or not to assign the target class label to t, CSPL-A uses a threshold ε to measure the score produced by Equation 1. ε can be determined by a tuning process, which preserves the training instance set D and prepares anther copy of D as query instance set. After all those "query instances" have been processed, CSPL-A obtains a list of scores for each training instance as well as their real class labels. Suppose the list is ranked in descending order, denoted by $\{\Omega^i\}_{i=1}^m$. The corresponding binary class label list is denoted by $\{C^i\}_{i=1}^m$. Then ε is selected to maximize its empirical classification accuracy among the training instances:

$$\varepsilon = \frac{\Omega^x + \Omega^{x+1}}{2} \tag{2}$$

where

$$\forall 1 \leq k \leq m, \quad (\sum_{i=1}^{k} C^i - \sum_{i=k}^{m} C^i) \leq (\sum_{i=1}^{x} C^i - \sum_{i=x}^{m} C^i) \tag{3}$$

ε can also be selected so as to maximize some other performance measures. In our experiments, we focused on maximizing the normalized utility measure as given in the official evaluation metric.

3 Experimental Results

The document triage task of TREC 2005 Genomics Track contains a total of 11,884 full-text documents, in which 5,837 and 6,047 documents are used for training and testing respectively. The ratio of positive instances for different categories varies from 5.6% to 0.5%. The overall number of terms is 113,203. There are four major types of information collected and catalogued by MGI are depicted as follows:

- Alleles of mutant phenotypes (A)
- Embryologic gene expression (E)
- GO annotations (G)
- Tumor biology (T)

Method	A	E	G	T
cuhkrun1	0.8228	0.8009	0.4635	0.8532
cuhkrun2	0.8443	0.8321	0.4293	0.3372
cuhkrun3	**0.8478**	**0.8321**	0.3045	**0.9028**
Median	0.7785	0.6548	0.4575	0.7610
Best	0.8710	0.8711	0.5870	0.9433

Table 1: Classification performance on the triage task of TREC 2005 Genomics Track, measured by the normalized utility.

We investigate this annotation problem using kNN and CSPL-A model. The performance on each subtask is measured by the normalized utility measure.

Table 1 summarizes the performance. cuhkrun1 is the performance of adopting kNN while cuhkrun2 and cuhkrun3 are that of using CSPL-A. All the three runs utilized the proximity based feature engineering. The parameters of kNN is tuned to be optimal on the training data set. A tuning process is conducted to select a small number of useful attributes from the large number of keywords. Only 200 attributes out of 12,000 are used for conducting kNN learning. cuhkrun1 and cuhkrun2 are the performance of kNN and CSPL-A on those 200 attributes, cuhkrun3 is the performance of CSPL-A without feature selection, because CSPL-A is able to fully exploit the attribute information for handling sparse positive examples. The performance of our CSPL-A is generally good with normalized utility well above the median.

The performance of CSPL-A on the largest GO category is not very good. The reason is that this category has the maximal number of positive instances among all categories. Moreover, if CSPL-A also uses selected set of attributes for this category, the performance will also be 0.4293. If kNN uses all attributes for learning, the performance will also degrade significantly for this category. On the other hand, CSPL-A demonstrates its prominent performance in handling sparse positive training instances with large number of attributes. It is clear that CSPL-A outperforms kNN on those 3 categories with sparse positive instances.

Moreover, it can be observed that most of our submitted runs in the four subtasks outperform the median of all the subtask runs. Hence, kNN demonstrates a promising performance over all four subtasks, while CSPL-A performs very well in the A, E, & T subtasks, having sparse positive instances, with normalized utility values of 0.85, 0.83 &

0.90 in comparing with median values of 0.7785, 0.6548 & 0.7610.

References

[1] Mitchell, T. 1997. Machine Learning. McGraw Hill.

[2] Y. Han and W. Lam. Query-driven support pattern discovery for classification learning. In *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM)*, pages 399–402, 2004.

[3] Y. Han and W. Lam. Lazy learning for classification based on query projections. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 227–238, 2005.

[4] B. V. Dasarathy. *Nearest neighbor (NN) norms: NN pattern classification techniques.* IEEE Computer Society Press, 1991.

[5] D. W. Aha. Editorial. *Artificial Intelligence Review, Special Issue on Lazy Learning*, 11:7–10, February 1997.