Categorization of Genomics Text Based on Decision Rules

Rocio Guillén
Computer Science Department, California State University San Marcos
email:{rguillen}@csusm.edu

Abstract

In this paper we present the approach, experiments and results for the categorization task in the Genomics Track. The approach we used is based on decision trees and decision rules for text categorization [1], [2], [3]. The features selected were the keywords and the contents of the glosref tags to induce rules. Rules were then matched with the contents of font-changes in the abstracts of the documents to determine whether to select or not the text.

1. Introduction

A decision tree (DT) text classifier is a tree in which internal nodes are labeled by terms, edges outcoming from these nodes are labeled by tests on the weight that the term has in the document, and leafs are labeled by categories.[4] A DT classifier categorizes a document D_i by recursively testing the weights that the terms labeling the internal nodes have in vector D_i, until a leaf node is reached; the label of this node is then assigned to D_i. Most DT classifiers use binary document representations, and thus consist of binary trees [5]. . Text categorization efforts based on experimental Decision Trees include Weiss et. al. [7], Dumais et. al. [8], and Lewis and Ringuette [9].

A possible method for learning a DT for category C_j consists in using a divide and conquer strategy, which is applied recursively. The method first checks whether all the training examples have the same label, i.e., C_j or $\bar{C}_j$. If this is not the case, it then selects a term T_k, that partitions the tree into classes of documents that have the same value for T_k. Lastly, each class is placed in a separate subtree. In the triage categorization task there are basically two classes to consider: 1) select for annotation or 2) not select for annotation. With only two classes, the main efforts were directed towards finding decision rules to classify the documents.

Decision rule classifiers for category C_j are built by an inductive rule learning method. The decision rule consists of a disjunctive normal form (DNF) rule, that is, of a conditional rule with a premise in DNF. The literals, which are keywords possibly negated, in the premise denote the presence or absence of the keyword in the document D_i, while the clause head denotes the decision to classify D_i in category C_j. DNF rules are similar to DTs in that they encode any Boolean function. An advantage of DNF rule classifiers is that they tend to be more compact classifiers than DTs.

Rule learning methods usually attempt to select from the set of rules that correctly classify all the training examples the best one according to some criterion. DNF rules are often built in a bottom-up manner. Initially, every training example D_i is considered a clause with terms $t_1, \ldots, t_n \rightarrow v_j$, v_j equals C_j or $\bar{C}_j$ depending on whether D_i is a positive or negative example of C_i. The set of clauses is already a DNF classifier for C_i, which scores high in terms of overfitting. A process of generalization follows to simplify the rule maximixing its compactness while at the same time preserving the "covering" property of the classifier. Lastly, the tree is pruned to trade correct classification of all training examples for more generality. DNF rule learners vary widely in terms of the methods, heuristics and criteria used for generalization and pruning.

2. Methodology

The methodology for the triage categorization task was built around the use of decision rules. The goal was to use automatic rule induction from the analysis of the training set. We accomplished this goal partially, since we did not implement a full DNF rule learner.

The first step was to produce a list of features from the documents categorized as positive examples in the training set. The features are single words in our case. The features selected are the $< GLOSREF >$ tag, which encloses a gene identifier (e.g., name, abbreviation of a gene name), and the $< kwd >$ tag, which encloses keywords. The set of values for each feature are single words obtained by parsing the documents and extracting the information within the tags to create a list of values per document. A document is represented as a list of such values, duplicate values are removed. Some

documents do not include keywords, but we observed that the set of keywords was usually contained in the set of "glosref"s. For instance, take document with artnum="jcb.200110108". The set of values for $GLOSREF$ is { c, Bax, Bak, Bim, Bid, Bad, Fas, FasL} and the set of values for KWD is { Bax, Bad, Bid, caspase, Fas}.

For rule induction, the objective is to find sets of decision rules that distinguish one category of text from the others. The best rule set is selected, where "best" is a rule set that is both accurate and not very complex. Accuracy of rule sets can be measured effectively on large numbers of independent cases. Complexity can be measured in terms of numbers of rules, where smaller rule sets that are reasonably accurate are preferred to more complex sets of rules with slightly higher accuracy. We have created a set of decision rules incrementally by selecting a positive example from the training set, then making the conjunction of its corresponding feature value set. An example of a subset of rules is shown in Table 1.

Rule 1	gan **and** gigaxonin **and**MAP
Rule 2	schmidt **and** incisure **and** MUPP1
Rule 3	bax **and** bak bf and bim **and** bid **and** bad
Rule 4	paxillin **and** IL-3

Table 1. Rules

Not all feature values were used in the rules. Manual tuning of the rules was done to avoid complex and redundant rules as the number of rules increased. However, redundancy could not be completely avoided.

Some of the problems encountered were with abbreviations, abbreviations are a source of ambiguity that we did not have time to handle. We are currently working on word sense disambiguation to improve retrieval performance.

We ran experiments on the training set for each rule created, comparing the terms in the rule with the terms in the document labeled with font-tags to expedite the matching process. The output was a file containing the ARTNUM of each document and a tag with the rule number indicating that the rule fired, or "notag" indicating that the rule did not fire. Experiments were

run until all the positive examples were assigned a rule number. In many cases positive examples were assigned multiple rule numbers. As a side-effect many negative examples were categorized as positive. The total number of annotated positive examples was 367 and a matching rule(s) was found for each of them. The total number of negative examples was 5470 (includes non-annotated positive examples). The total number of negative examples for which a rule matched was 613. The total of training data was 5837. Results of these experiments are shown in Table 2.

	Positive	Negative
Positive	367	613
Negative	8	4849

Table 2: Totals for categorization task using training set

The results for the test data are shown in Table 3.

Counts:	tp=74 fp=860 fn=346
Precision:	0.0792
Recall:	0.1762
F-score:	0.1093
Utility Factor:	20
Raw Utility:	620
Max Utility:	8400
Normalized Utility:	0.0738

Table 3. Results for categorization task using test set

3. Conclusion

From the evaluation results generated for the triage task, we conclude that overfitting worked well to classify the training data, but not the test data. The set of rules was not general enough, no pruning or statistical techniques were used to further refine the rule set, and the negative examples were not properly represented to improve accuracy in the categorization. As a result, precision and recall were below the median.

References

[1] Apté C., and Weiss, S.M.: Data Mining with Decision Trees and Decision Rules. *Future generation Computer Systems*, 13:197-210, 1998.

[2] Apté C., Damerau F., and Weiss, S.M.: Automated Learning of Decision Rules for Text Categorization. *ACM Transactions on Information Systems*, 12(3):233-251, July 1994.

[3] Apté C., Damerau F., and Weiss, S.M.: Text Mining with Decision Trees and Decision Rules *Conference on Automated Learning Snd Disscovery*, Carnegie Me;;on University, June 1998.

[4] Mitchell, T.M.: *Machine Learning.* MacGraw Hill, ew York, NY, 1996.

[5] Sebastiani, F.: Machine Learning in Automated Text categorization *ACM Computing Surveys*, 34(1):1-47, March 2002.

[6] Yang, A.: A Comparative Study on Feature Selection for text categorization. In *Proceedings of the International Machine Learning Conference*, Morgan Kaufmann, 1997.

[7] Weiss, S.M., Apté, C., Damerau, F.J., Jihnson, D.E., Oles, F.J., Goetz, T., and Hampp,T.: Maximizing text-mining performance. In *IEEE Intelligent Systems*, 14(4):63-69, 1999.

[8] Dumais, S. T., Platt, J., Heckerman, D., and Sahami, M.: Inductive learning algorithms and representations for text categorization. In *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management (Bethesda, MD)*, 148-155, 1998.

[9] Lewis, D. D. and Ringuette, M.: A comparison of two learning algorithms for text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval (Las Vegas, NV).*, 81-93, 1994.