A TEXT-MINING APPROACH FOR CLASSIFICATION OF GENOMIC FRAGMENTS

Vinay Gadia and Gail Rosen

vg482@drexel.edu, gailr@ece.drexel.edu

Department of Electrical and Computer Engineering
Drexel University, Philadelphia, PA 19104

ABSTRACT

Genome identification is an emerging area of interest due to the study of environmental DNA samples. We show that performance approaches 50% for classifying 500bp fragments when using 12mer features, but more importantly, the performance linearly increases for large N . Secondly, we determine that an inverted TF-IDF measure performs 16% better when only using 80% of the words, as opposed to taking the fullset (100%). This increase implies that while too sparse of a feature subset does not produce good results, a carefully selected set has the potential to improve genome classification over a random feature set. Computing even 80% of all possible features can result in a significant savings in computation. The Euclidean classifier and TF-IDF measures will pave the way for more discriminative classification techniques.

1. INTRODUCTION

As of January 2008, more than 260,000 named species are represented in GenBank, and new species are being added at the rate of over 1700 per month [1]. This makes an expanding database for genome analysis, including phylogeny (evolutionary tree) studies, comparative genomics, etc. There has never been more of a need to develop fast global tool for fragment identification [2, 3, 4]. Such a framework will provide biologists with the ability to identify the content of a sample in a high-throughput manner.

There have been previous attempts at classifying DNA based on text-mining measures. Typically, given a language each document is reduced to representation by a vector of frequencies of terms selected in an appropriate way. The next step relates to finding a suitable distance between the vectors, such as Manhattan, Euclidean, or Kullback-Leibler distances. In [5], [6], the authors explore how using these measures with text-mining approaches can be used to cluster 37 species.

Essentially, we have a linguistics problem where we have a set of words contained in several documents (genomes), and now we can use information-retrieval methods to determine a vector space model [7]. From the features, we wish to choose the appropriate features as well as classify the “documents” [8]. In our paper, we develop a classifier using frequencies of N -mer words and test over 635 microbial genomes. We have two types of vectors, the fragment vector and the database vector, and use the Euclidean distance as the measure between the two. We explore choosing the best features (or words) via the TF-IDF (term-frequency inverse-document frequency) measure, the inverse TF-IDF, and a random subset of words. We then evaluate the performance of our classifier over varying N and subset size of N -mers. For 100% (or a fullset) of words, N is varied over 3, 5, 6, 9, 10, 11, and 12. For the feature selection, the percentage of selected words is varied over 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100% of the words and is tested for $N = 3, 5, 6$, and 9.

2. METHODS

2.1. The Euclidean Classifier

Firstly, we compile the set of words used in both the fragment and the database. Secondly, We classify the fragment vector by calculating the distance from the fragment to each database vector through the Euclidean distance:

$$d(\mathbf{f}, \mathbf{d}_m) = \sum_{l=1}^L \left(\frac{f_l}{|\mathbf{f}|} - \frac{d_{ml}}{|\mathbf{d}_m|} \right)^2 \quad (1)$$

where l is the index of a particular word in the fragment and genome. There is a possibility that the l th word will not exist in m^{th} genome or in the fragment. In these cases, the frequency count will have a 0 in this position (for the fragment or genome it does not occur in). A flow diagram of the algorithm can be seen in Fig. 1.

The authors would like to thank Elaine Garbarine for Matlab tips.

Fig. 1. A flow diagram of the Euclidean classifier for a fragment word-probability vector against database word-probability vectors. The above example is for one fragment.

Fig. 2. The Euclidean classifier performance for 500bp fragments for $N = 3, 5, 6, 9, 10, 11$, and 12. Not until $N = 9$ does performance significantly increase. In fact, it is hypothesized that the Euclidean distance does not perform well with full usage of N mers but for $N > 9$, the probability of N -mer uniqueness increase performance.

2.2. Choosing Genomic Features via the TF-IDF Measure

We compute the TF-IDF as $TF - IDF_{ml} = tf_{ml} idf_l$ [8]. For the overall TF-IDF weight of a word, we sum over the rows and rearrange the equation to get:

$$TF - IDF_l = \sum_{m=1}^M \left(\frac{d_{ml}}{\sum_{l=1}^L d_{ml}} \log \left(\frac{M}{\sum_{m=1}^M d_{ml}} \right) \right) \quad (2)$$

We sort the TF-IDF weights from (2) to rank the top $P * 4^N$ % words corresponding to the TF-IDF weights, where P is a percentage-of-words-chosen threshold.

2.3. The Inverted TF-IDF and Random Word Selection

The inverted TF-IDF measure is defined as sorting the lowest-ranking TF-IDF score to the highest and choosing the top- $P * 4^N$ percentage of words (or the lowest TF-IDF weights). Both the TF-IDF and the inverted TF-IDF are compared against choosing a random set of words, to see if salient classification features can be obtained from the TF-IDF measure.

3. RESULTS

For our analysis, we used all completed microbial sequences in the NCBI Genbank as of February 2008, which totaled 635 distinct microbial strains. The 635 microbes belong to 470 distinct species and 260 distinct genera in this dataset.

3.1. Euclidean classification using full-set of words

For genome classification, the Euclidean distance between the word-frequency fragment vector, $\mathbf{f}$, and each of the 635

Fig. 3. For $N = 6$, randomly selected words again do better than the measures up until 50% of the words. The inverted TF-IDF using 90% of words performs 4% (16% better) above the full-word performance, showing that by choosing a subset of words, classifier performance can be improved.

Fig. 4. For $N = 9$, randomly selected words perform better than the TF-IDF measure. The inverted TF-IDF using 80% of words performs 2% (10% better) above the full-word performance, showing that by choosing a subset of words, classifier performance can be improved.

genome word-frequency vectors in $\mathbf{d}$ is calculated. The genome resulting in the least Euclidean distance with the fragment is claimed to be the closest match and thus classified to a genome. Tests are performed for 500 bp fragments for $N = 3, 5, 6, 9, 10, 11$, and 12. The accuracy of the classifier increases as N grows (see Fig. 2).

3.2. Comparison of TF-IDF, inverted TF-IDF, and random word selection on the classifier

To determine the effectiveness of selecting only the “most important” words instead of all the words, we choose a varying % of TF-IDF, inverted TF-IDF, and random words for $N = 3, 5$, and 6. For our TF-IDF word selection, we pre-computed TF-IDF weights of the words for each Nmer using (2). Subsets of the words are chosen at subset intervals 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and compared against the fullset, 100% of the words, in the Euclidean classification. Only the fragment words occurring in the TF-IDF words are considered in the comparison vectors in (1), i.e. the vectors are shortened in size. For inverse TF-IDF word selection, the lowest TF-IDF-weighted words are used, and the Euclidean classifier is run for the newly obtained vectors. Both measures are compared against randomly selecting the words

from a possible 4^N combination of words. Figs. 3 and 4 illustrate how the different measures performed for varying N . In general, selecting under 50% words randomly performs better than the other tf-idf measures. Selecting more than 50% of the words, the inverted TF-IDF words performs the best, and optimal performance occurs at about 80% of the words. Thus, demonstrating that choosing an optimal subset of N -mers can improve classification performance.

4. DISCUSSION

In this paper, we explored 3 feature (or word) selection measures: 1) the TF-IDF selection, 2) the inverted TF-IDF selection, and 3) a random word selection. An interesting observation is that the words with high TF-IDF weights, do not perform as well as randomly chosen ones. But for small subsets of features used, TF-IDF intuitively outperformed the inverted TF-IDF; it is intuitive because the “term frequency – inverse document frequency” measure was meant to select a sparse subset of words that represents the important words in the database. Most importantly and suprisingly, the inverted TF-IDF (those with the lowest TF-IDF weights), outperform the randomly selected words in all cases when more than 50% of the words are chosen. Not only does it outperform the randomly chosen words, but it also causes the Euclidean classifier to perform better than using a fullset (100%) of words.

5. CONCLUSIONS

A Euclidean classifier was tested on 500 bp fragments for $N = 3, 5, 6, 9, 10, 11$, and 12-length features. This paper demonstrates the feasibility of the TF-IDF feature selection method, and that by taking a subset of words, a simple classifier can be improved. The fullword-set accuracy of the classifier increases for $N > 9$, reaching close to 50% accuracy for $N = 12$. The TF-IDF measure and an inverted TF-IDF measure for feature selection are compared to randomly selecting words. The TF-IDF is compared for word lengths of $N = 3, 5, 6$, and 9, and their effect on genome classification is shown. A promising result is that by taking 80% of the inverted TF-IDF measure words yielded a 4% performance increase (16% better) over full-word classification, demonstrating that by choosing the a word subset, a basic classifier performance can be improved.

6. REFERENCES

- [1] D. A. Benson and et al., “Genbank,” *Nucleic Acids Research*, vol. 36, 2008.
- [2] P. J. Turnbaugh and et al., “The human microbiome project,” *Nature*, vol. 449, pp. 804–810, 2007.
- [3] “Sequencing the microbial soup,” *Nature Structural and Molecular Biology*, vol. 15, no. 115, 2008.
- [4] J. Bohannon, “Confusing kinships,” *Science Magazine*, 2008.
- [5] Z. Volkovich and et al., “The method of n-grams in large-scale clustering of dna texts,” *Pattern Recognition*, 2005.
- [6] V. Kirzhner and et al., “Large-scale genome clustering across life based on a linguistic approach,” *Elsevier Journal of Biosystems*, 2005.
- [7] G. Salton, A. Wong, and C. S. Yang, “A vector space model for automatic indexing,” *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.
- [8] S. Weiss and et al., “Text mining,” in *Predictive Methods for Analyzing Unstructured Information*, Springer Science+Business Media, Ed., 2005.
- [9] Y. Fofanov, Y. Luo, C. Katili, and et al., “How independent are the appearances of n-mers in different genomes?,” *Bioinformatics*, 2004.