Linear SVM based automatic classification of biomedical publications

Shashank Khanna
Advisors: Dr. Peipei Ping, Vincent Kyi

Abstract

This paper aims to address the problem of creating a searchable database of biomedical software tools. Given that there was no central repository where software tools are published, a linear SVM based technique was developed to automatically identify a biomedical software product given a research paper. Using text classification techniques an accuracy of 88.8±0.7% was achieved for the same.

1 Introduction

Currently, Bioinformatics software tools are present in silos with no common endpoint which could serve as a one size fits all solution for researchers to access the software they need. This present condition, leads us to create a solution to collate all present bioinformatics software based resources and provide a common access point for them. In order to achieve that, all publications that could potentially contain software related to Bioinformatics, which is henceforth referred to as a tool in this paper, would need to be inspected, and identified as a tool or a not a tool. Given the scale of this problem, it is physically impossible to individually study every single publication. This leads the problem towards a machine learning solution.

However, given the constraints of the problem are tightly defined, it is possible to observe a subset of biomedical publications and manually classify them. Hence, I decided to adopt a supervised approach. I evaluated various supervised learning approaches namely: Decision Trees, which involve classifying a pattern through a sequence of questions where the next question depends on the answer to the current question (1); Neural Networks and Linear SVM and decided to pursue Linear SVM due to its performance metrics on the particular feature set relevant to our problem.

Among supervised techniques the Support Vector Machine (SVM) is a popular learning machine that learns from training data and classifies vectors in a feature space into one of two subgroups (2). The machine first converts the training data to a feature space and finds the optimal hyperplane that can divide the two sub-groups (3). Linear SVM is a sub-type of the SVM which is used when the training data is seperable by a hard-margin (linearly separable data).

2 Data

The intention behind this attempt was to mine all biomedical publications possible and be able to build a pipeline to enter the software tools into a searchable database. However, in the interest of building and verifying the accuracy of an initial proof-of-concept model, the scope was restricted to publications from four journals: Bioinformatics by Oxford Univeristy Press, Nature Biotechnology, PLOS Computational Biology and BMC Bioinformatics.

2.1 Modifying the Data for feature identification

The initial dataset was present as academic papers or articles in a pdf format. This was difficult to analyse programatically and hence was converted to a text format, which was then further processed into a serializable JSON format. Metadata was extracted from the original PDF version such as details regarding the journal, the authors and identification information. The first step towards feature identification was analyzing these JSON data files, for ideas that could potentially influence the probability of a particular publication being a tool. This was done by building a web application and manually classifying the initial training dataset into a tool or a non-tool through the application. Once the training dataset was available, the next step was to identify the potential characteristics of the individual file that could correlate it to being a tool. This was analysed through two main techniques.

2.1.1 Intuitive Features

The first technique was determining features intuitively and then verifying their accuracy through the feedback from the classifier. It was hypothesized that the probability of a particular product being a tool should increase if it has a connection to a source code repository in the publication. Also other conceivable features that were decided on were: the number of links, the mention and frequency of coding languages throughout the paper and the mention of operating systems in the paper. The validity of the features could then be tested by considering all possible combinations of the hypothesized features for the supervised classifier and selecting the combination that leads to the best possible accuracy statistics.

2.1.2 Feature Extraction based on word frequencies

The second technique involved looking at word frequencies in individual documents. This was tested for the both the abstracts of the given publications in the test data and also entire text output. Therefore, the separated data, tools and non tools were individually analyzed.

The ten most frequent words from each of the abstracts of tools and non-tools were first collected. These selected words were then processed such that each "frequent" word had a single frequency representation per abstract regarless of how many times it occurred. The 10 most frequent words that were from this entire collection of the tool set and non-tool set were then selected ignoring certain common english language and academic language words which were considered as stopwords. For this particular dataset, it resulted in the following words having the respective frequencies as shown in the table below.

Tools		Non-Tools	
Word	Frequency	Word	Frequency
data	207	pubmed	283
available	133	pmid	283
sequence	132	medline	225

method	119	indexed	225
use	90	doi	207
protein	83	nat	122
supplementary	81	nbt	119
gene	70	method	112
analysis	60	comment	109
result	54	use	53

Table 1: Frequency of common words for tools and non tools

This data was further analysed in order to remove words that are frequent to both tools and non-tools, i.e. for this example the word ‘use’ was not considered as it had a “frequent" occurance in 90 tools but also in 53 non-tools out of our dataset.

3 Results

A combination of these techniques were used to define features for the classifier, which primarily involved the link count, the use of the intuitively identified word presence and set of words created through frequency analysis. This was done by first representing the feature set in a JSON format which contained the respective intuitive features and word to frequency mapping and converting it to a format consumeable by a classifier. The given data was then divided into a training set, a validity set and a test set. The data was trained using the training set and the validity set was used for optimizing the magin for the separation of the hyperplane in our SVM classifier.

The data was ran through different classifiers which resulted in the results as shown in the graph below. It was first ran on a Decision Tree, and then on a Neural Network with one hidden layer. For the case of the Linear SVM, multiple values of C, which informs the regularization paramter, namely 0.1, 1, 10 and 100 were tested using the validity set. The most optimal C value, which in this particular case was 0.1 was then used for the test data.

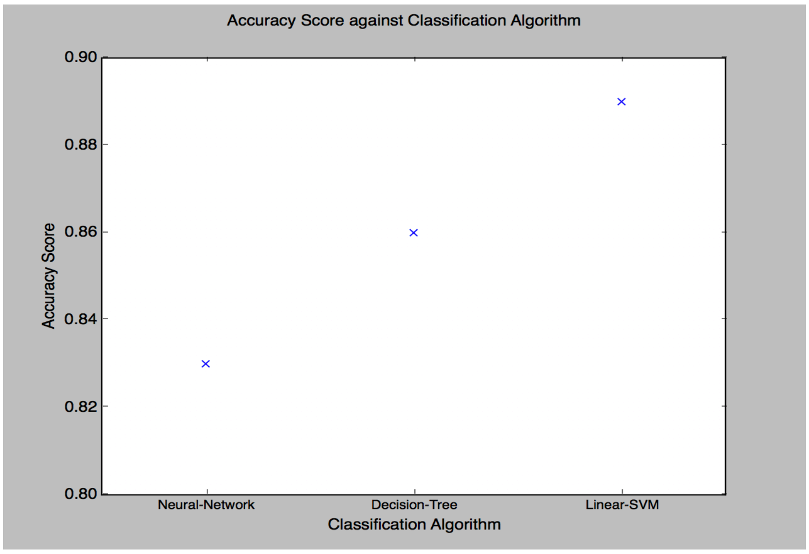

Figure 1 Comparing accuracy scores of supervised learning classifiers

4 Analysis

Linear SVM was intially chosen as the focal point for this classification as it has been proven to result in a better accuracy when the ratio of the feature set is large relative to the size of the training data which was the case in this particular text classification problem. From the results of this experiment, the Linear SVM provided an accuracy of 88.89% upon being trained on a dataset of 1212 publications out of which 481 were manually identified as tools and the remaining were non-tools. Out of these publications, 20% of them were randomly selected to be part of the test dataset, 10% were a part of the validity set and the remaining 70% were used to train the data. This was the highest among all other classifiers, with the remaining being 86.1% for the decision tree and 83.4% for the neural network.

5 Conclusion

5.1 Alternate techniques to create a feature set

This particular model was trained by analysing word frequencies in the abstracts of the given publication. Another potential technique to create the feature set can be by examining the word frequencies in the entire publication and comparing that to the other publications that are considered tools. This can be done by exploring the term frequency inverse document frequency or the tf-idf weight, which is a measure to evaluate how important a word is to a document in a collection. This could theoretically lead to a higher accuracy in classifying tools from non-tools.

5.2 Future implications

The pipeline designed as a part of this project can be used to create a central repository for all biomedical software publications. Given that all biomedical publications can be accessed, this classifier becomes an integral part of the decision making process behind the publications that are included in this repository. As a result, biomedical researchers no longer need to manually curate the software necessary for their research, they can instead rely on a common searchable database.

6 Bibliography

1. **MSU.** Decision Trees. *MSU Computer Science.* [Online] January 2, 2017. http://www.cse.msu.edu/~cse802/DecisionTrees.pdf.
2. *Support-Vector Networks* . **Cortes, C. & Vapnik, V.** 273, 1995, Machine Learning, Vol. 20. 1022627411411.