Yggdrasil:
Natural speech learning with random forests

Joseph D. Romano*[1] and Alexandre Yahi†[1]

[1]Departments of Biomedical Informatics, Systems Biology, and Medicine, Columbia Univeristy

April 25, 2016

1 Introduction

Natural language processing and speech recognition comprise one of the largest applications of machine learning and data mining techniques, spanning diverse industries such as economics, healthcare, information retrieval, and personal computing<REFs>.

2 Description of provided data

The original input we were provided with consisted of 126837 data records and 52 features, 36 of which were numeric (including binary 0/1) and 16 categorical. Each categorial feature had a variable number of potential categories. It is unclear what each feature vector represents, but the labels corresponding to some of the categorical feature vectors provide clues as to their meanings. For example, the categorical feature vector labeled "26" includes terms such as `vacknoweldge_acknowledge`, `vacknowledge_explain`, and `vacknowledge_clarify`, suggesting that it encodes the type of response given by one of the two parties holding the conversation. Other features seem to encode information about prepositional phrases, object comparisons, negation, and others. Some further information about the original encoding of the data is available in <ref>.

3 Learning approach

Our optimal learning approach for the binary speech classification problem involved two main steps:

1. Encoding all categorical feature vectors as sets of binary vectors via expansion
2. Training a random forest classifier and predicting over unlabeled data

We will discuss each of these below:

*jdr2160@cumc.columbia.edu

†ay2318@cumc.columbia.edu

3.1 Preprocessing and one-hot encoding of categorical features

One noteworthy characteristic of most random forest classifiers is that they are unaffected by scaling, centering, and other monotonic transformations, since all operations on the data are simply linear partitioning (although it is possible to design non-linear decision boundaries for partitioning the data, doing so is generally unnecessary with random forests).

3.2 Training and running random forest classifier

4 Results

5 Discussion

6 Conclusions