MODELING THE CLASSIFICATION OF PARKINSON'S DISEASE USING ABSTRACTED SPEECH DATA FROM A CASE-CONTROL STUDY

Joseph Kim JSKIM98@LIVE.UNC.EDU
Tao Tao TTAO@CS.UNC.EDU
Jacob Laudeman LAUDEMAN@LIVE.UNC.EDU
Ian Ferguson IANFERGU@LIVE.UNC.EDU

Abstract

Prior work on machine learning models to predict Parkinson's disease in patients has shown great potential as a clinical tool. Here, we present additional possible models to be used as classifiers, including decision trees, AdaBoost boosted decision tree, gradient boosted decision tree, support vector machines, random forest, extremely random forest, and voting ensemble classifier. The data examined included 756 audio recordings from 252 individuals, some healthy, and some diagnosed with Parkinson's disease. Our results show that AdaBoost boosted decision tree offers the best potential accuracy while also demonstrating the viability of alternative models.

1. Introduction

As a neurodegenerative disease, Parkinson's disease (PD) afflicts patients with a variety of symptoms. Because the disease affects the muscles of the face, it can also impede a patient's ability to produce clear speech. Therefore, tests that can accurately assess an individual's enunciation can allow for the early detection of PD. We build from work carried out by Sakar et al. to explore different classification methods in order to classify individuals as either PD patients or healthy (Sakar et al., 2019). They demonstrated a novel use of tunable Q factor wavelet transforms (TQWTs) for analyzing audio signals. We further explore various classification algorithms using these features extracted from audio recordings collected from both healthy and PD patients.

2. Methods

Our methodology is influenced by the pathological nature of PD. Unlike many other illnesses, research has found that the symptoms of PD are multi-modal and complex, usually reflecting themselves over a range of physical, neurological, and psychological characteristics of the patient. So far, the cause of PD is unknown except for some preliminary speculations (Kalia & Lang, 2015). The mysterious nature of PD makes specific feature and architecture engineering a difficult challenge.

Currently, deep neural networks provide the best generality and learning power for a vast domain of practical problems and enjoy numerous results in areas such as image analysis and natural language processing (Goodfellow et al., 2016). However, to obtain and train a satisfiable neural network model that not just yields high accuracy but are also interpretable, sophisticated data transformation architecture such as convolutions are required based on careful analysis of the inherent structures of the data sets (Lecun et al., 2015), which are not available in the case of PD due to the sophisticated nature of the problem. There are also a few issues related to over-fitting regarding the use of neural networks (Sakar & Kursun, 2010).

We found that it is still possible to obtain a decent model based on a handful of techniques, particularly harnessing the techniques of meta learning (Schaul & Schmidhuber, 2010), as we will demonstrate in the following sections. Meta learning is a terminology that broadly refers to statistical learning methods that are relatively generic and methodical as opposed to methods that are tailored to solving specific problems such as those in natural language processing. The most significant advantage of these methods is that they can be applied without complex data engineering efforts. This allows us to conduct a flexible range of experiments and perform interesting analyses based on the performance of these models.

University of North Carolina at Chapel Hill, COMP 562 Course Project.

2.1. Data

The first step in our research consists of choosing a data set that best suits our purposes. Here we cite the data sources from recent research which not only includes traditional baseline features in the PD analysis community but also adds results derived from newly conducted experiments on speech signals of a sample of PD patients (Sakar et al., 2019).

The data we selected examined 252 individuals, including 188 PD patients and a control group of 64 otherwise healthy individuals. For each individual, three audio recordings were sampled from three separate repetitions of the vowel /a/. These data were further processed via a tunable Q-factor wavelet transform in order to extract additional features. Four other sets of features were also included: baseline features, time-frequency features, Mel Frequency Cepstral Coefficients (MFCCs), and vocal fold features. The baseline features consist primarily of features already popularly used in PD research and classification problems, while the time-frequency features included the speech signal intensity and frequency range. The MFFCs provide a means to mimic the human ear and its effects on speech. In particular, it is well-suited for analyzing deviations in the movement of the tongue and lips that are associated with PD. Lastly, the vocal fold features measure the noise generated by the folds of the human vocal cords.

2.2. Classification Methods

To solve this classification problem, we looked at seven different methods: (1) decision tree; (2) AdaBoost boosted decision tree; (3) gradient boosted decision tree; (4) support vector machines; (5) random forest; (6) extremely random forest; and (7) voting ensemble classifier. For each classification method, we ran cross-validation on our test data to tune our hyperparameters before creating our final models.

2.2.1. Decision tree

Our first experiment involves using generic decision trees (Rokach & Maimon, 2008) as the foundation of our analysis. It is one of the most interpretable learning methods available with sound theoretical properties. The idea is to construct a complete binary tree with each node corresponding to a specific feature that acts as discriminators of the two partitions. Hence, the final tree represents a decision selecting model that predicts data.

We made the decision tree using scikitlearn's DecisionTreeClassifer (Pedregosa et al., 2011). We cross-validated on 1 maximum tree depth of 1 through 10, 15, and 20. Maximum tree depth determines how complex the decision tree is, so it's good to cross-validate on it to prevent overfitting. Upon cross-validation, the best maximum depth was 6, which will be used in our final decision tree.

A stronger version of the generic decision tree is made possible by the AdaBoost algorithm among a range of other boosting algorithms (Schapire & Freund, 2013). Boosting is an ensemble method that combines many weak classifiers to make one strong classifier to improve accuracy rates. We cross-validated on the number of estimators and learning rate. The number of estimators is the number of weak classifiers that are used in our ensemble, and the learning rate shrinks the contribution of each classifier. For the number of estimators, we cross-validated from 50 to 200 by increments of 50. For the learning rate, we cross-validated from 0.1 to 1 by increments of 0.1. Upon cross-validation, the best combination of parameters was 200 estimators, and the learning rate of 0.4, which will be used in our final AdaBoost boosted decision tree.

We also explored another boosting method called gradient boosting. Gradient boosting iteratively enhances the model by aggregating initial weak results with a revised model that minimizes a certain error objective. We cross-validated on the number of estimators and learning rates as well. Upon cross-validation, the best combination of parameters was 200 estimators, and a learning rate of 0.2, which will be used in our final gradient boosted decision tree.

2.2.2. Random forests

We further add variety to our methods by including random forests. As opposed to a single decision tree, random forests learn multiple instances of decision trees by randomizing the decision tree construction procedure and predict results based on a voting mechanism. This complements the fact that generic decision tree is usually not optimal and prone to overfitting.

We cross-validated on the maximum number of estimators from 50 to 200 by increments of 50 and found that the best hyperparameter was 50, which will be used in our final random forest model.

We made the extremely randomized forest using scikitlearn's ExtraTreesClassifier (Pedregosa et al., 2011). We cross-validated on the maximum number of estimators from 50 to 500 by increments of 50 and found that the best hyperparameter was 50, which will be used in our final extremely randomized forest model.

2.2.3. Support vector machines

Other than the set of decision tree-based methods we showcased above, Support vector machines (SVM) often prove to be very effective in binary classification problems and hence are included as well. Conceptually speaking, classic SVMs classify two categories of data by fitting a hyperplane across the high-dimensional data space that maximizes the margins.

We cross-validated on the regularization parameter C: 1, 10, 100, 1000, and 10000. As learned in class, C impacts the margin of the support vector machine. Upon cross-validation, the best parameter was 1000, which will be used in our final SVM.

2.2.4. Voting classifiers

Lastly, we used a voting classifier using scikitlearn's VotingClassifier module. This classifier classifies a sample by consulting different classification methods through a voting system. Hard voting classifies based on majority voting, and soft voting classifies based on weighted voting. We used all six previous classification methods in our voting classifier models and made both a soft voting classifier and a hard voting classifier.

3. Analysis

We make several observations based on the results obtained from the experiments.

3.1. Experimental results

The prediction accuracies, Recall, Precision, F-Measure, and Matthews Correlation Coefficient for each classification method are provided in the table below. Along with the individual classification methods, Hard and Soft sample voting were included to gain an understanding of the accuracy of all of the prediction methods combined.

3.2. Model evaluation

From Table 1, shown below, we can see that the most accurate classification method for predicting whether a subject has Parkinson's or not is the AdaBoost Boosted Decision Tree, with an accuracy of 0.899. Gradient boosting was only slightly inferior, with an accuracy of 0.888. Random Forest and Extremely Random Tree were also highly accurate. The Support Vector Machine and Decision Tree were the least accurate.

In the graph below, the ROC curves of the four best algorithms are compared. The curves for Gradient Boosting and AdaBoosting somewhat complement each other. AdaBoost reaches a slightly higher true positive rate at the beginning, which makes its curve stand out. It makes sense to select a model that will have a higher true positive rate with a slightly increased false-positive rate for a medical test. Meaning, you would rather correctly diagnose a few more people that need care in exchange for a few extra people getting unnecessary treatment.

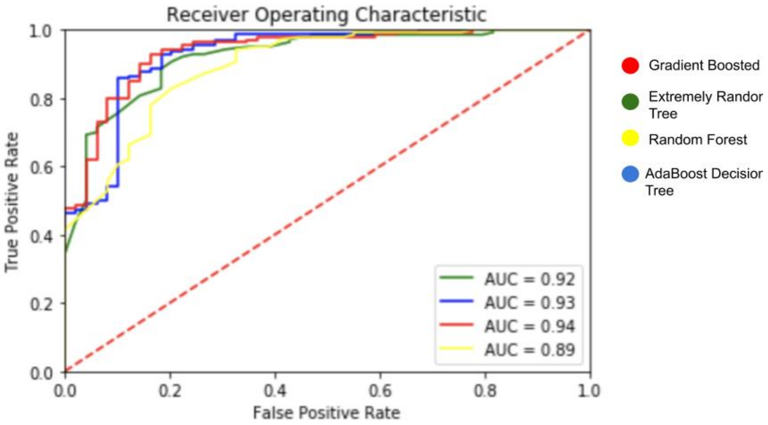

Table 2. ROC curves for four highest performing models

With an MCC of 0.728, it is confirmed that AdaBoost has the best ROC curve, as it results in the best trade-off of TPR and FPR. Due to this fact, and that it is the highest performing by all measures (except for Precision, which is slightly less than Random Forest), we can see that AdaBoost is all-around the best classification algorithm to predict PD.

3.3. Meta-analysis and future works

As a standard, effective model of binary classification, SVM scores significantly lower in accuracy than decision tree models. This, together with the fact that all the decision-tree-based models share relatively close results, indicates the fact that the decision tree model is an overall competent model for feature-based classification on PD. This makes sense as SVM classifies based on support vector positions and does not discriminate features. In this sense, the decision trees do a better job and signify the importance of feature sets in PD data sets.

On the other hand, the boosting algorithms we applied are designed to obtain a better fit of the data. While doing so, at each stage, proper cross-validation is used to minimize over-fitting. Hence, the boosted results reflect the actual power it has on the PD data set. As we can see, the boosting algorithms have a limited effect on improving the accuracy of the generic decision tree model by yielding less than 10% gains. However, since a perfect model would not benefit from boosting, it would be interesting to conduct further research and look at how feature engineering can be applied to decision trees to enhance its effectiveness.

4. Conclusion

Parkinson's disease threatens the health of the public demography with its detrimental effects and difficult nature. The latter determines the prematurity of complex engineering efforts in the data science of PD. Our study is a small step into the research by applying an ensemble of meta

Table 1. Prediction Accuracy and other measures for performance of correlation algorithms

Classification Method	Accuracy Score	Recall	Precision	F Measure	MCC
Decision Tree	0.825	0.51	0.735	0.602	0.509
Extremely Random Tree	0.868	0.571	0.875	0.691	0.639
Random Forest	0.868	0.551	0.900	0.684	0.635
AdaBoost Boosted Decision Tree	0.899	0.714	0.875	0.786	0.728
Gradient Boosting	0.888	0.653	0.889	0.753	0.697
Support Vector Machine	0.757	0.184	0.600	0.282	0.228
Hard Voting Ensemble	0.857	0.571	0.903	0.700	0.651
Soft Voting Ensemble	0.862	0.51	0.926	0.658	0.621

learning algorithms on a selection of data set that has the best feature-wise representations of the neurological and physical characteristics sampled by state-of-the-art instruments from a pool of Parkinson's disease patients. Based on the experimental results, we classified the effectiveness of the decision tree algorithms and suggested a few future research directions.

References

Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron. *Deep learning*. MIT press, 2016.

Kalia, Lorraine V and Lang, Anthony E. Parkinson's disease. *The Lancet*, 386(9996):896 – 912, 2015. ISSN 0140-6736. doi: https://doi.org/10.1016/S0140-6736(14)61393-3.

Lecun, Yann et al. Lenet-5. *convolutional neural networks*, 2015.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Rokach, Lior and Maimon, Oded Z. *Data mining with decision trees: theory and applications*, volume 69. World scientific, 2008.

Sakar, C Okan and Kursun, Olcay. Telediagnosis of parkinson's disease using measurements of dysphonia. *Journal of medical systems*, 34(4):591–599, 2010.

Sakar, C Okan, Serbes, Gorkem, Gunduz, Aysegul, Tunc, Hunkar C, Nizam, Hatice, Sakar, Betul Erdogdu, Tutuncu, Melih, Aydin, Tarkan, Isenkul, M Erdem, and Apaydin, Hulya. A comparative analysis of speech signal processing algorithms for parkinson's disease classification and the use of the tunable q-factor wavelet transform. *Appl. Soft Comput.*, 74:255–263, January 2019.

Schapire, Robert E and Freund, Yoav. Boosting: Foundations and algorithms. *Kybernetes*, 2013.

Schaul, Tom and Schmidhuber, Jürgen. Metalearning. *Scholarpedia*, 5(6):4650, 2010.