Computer Science Department

Early-stage detection of Parkinson's disease with Tappy

Course: Introduction to AI (CS156)
Course Instructor: Yulia Newton, Ph.D.

David Thach

Introduction/Background

Parkinson's disease is a progressive nervous system disorder that affects movement, commonly causing stiffness or slowness of movement. It affects nearly 6 million people, resulting in a range of both motor and non-motor symptoms.[1] Symptoms of this disease may be reflected in many activities in everyday life, potentially including typing. The goal of this project is to predict whether a particular person may have Parkinson's by analyzing their keystroke data (positional changes and press/release delays).

Dataset description

The dependent variable being determined is whether or not a particular person has Parkinson's disease. This is a strictly binary categorical variable, as they can only either test positive or negative. Tappy records the following independent variables, which are anonymized for use in the dataset, along with their respective unique user IDs, date, and timestamp:

Ind. Var.	*Description*	*Type*
Hand	L-hand or R-hand key pressed	Categorical (2 states)
Direction	Previous key to current key (LL, LR, RL, RR)	Categorical (4 states, 9 w/ space)
Latency time	Time between pressing the previous key and press of current key (in ms)	Numeric (floating point value)
Flight time	Time between release of previous key and press of current key (in ms)	Numeric (floating point value)
Hold time	Time between press and release for current key (in ms)	Numeric (floating point value)

These keystrokes are recorded at random and are not stored in consecutive order. This means that the keystrokes can be analyzed independently of each other or in respect to the previous stroke, as denoted in the "direction" field. This results in each user having hundreds to thousands of records each, which accumulates into a dataframe with dim size (9.0e6 entries x 8 columns). To reduce the size, the date and timestamp columns are then removed, with no effect on the dependent variable. To further reduce the size, participants with fewer than 1000 keystrokes are removed, bringing the total number of usable participants down from 227 to 155. This also benefits the model, since it reduces small sample bias - few keystrokes would not be fully representative of a person's ability, as small variations would be magnified . This removes roughly 10k entries from the dataframe, with dim size (8.9e6 entries x 6 columns).

	UserKey	Hand	Hold time	Direction	Latency time	Flight time
0	0EA27ICBLF	L	101.6	LL	234.4	156.3

8996101 rows × 6 columns

	HT Mean (L)	HT Mean (R)	HT STD (L)	HT STD (R)	HT Skew (L)	HT Skew (R)	HT Kurt (L)	HT Kurt (R)	Mean difference	LT Mean (LR)	...
0	77.749454	17.598336	1.585134	11.522029	79.306669	24.609195	4.130419	44.944694	1.557215	277.610541	

155 rows × 28 columns

```
Number of unique users: 155
```

At its current form, the dataframe holds too much unnecessary data for the model to draw any sort of conclusion. Each individual keystroke is not crucial to determining whether a participant has the disease, so we can simplify it by extracting important features from the independent variables, such that each feature has a single statistic that relates to the data for the user as a whole. With the hold time and latency time of each key, it would be best to split the features into those that depend on the interaction with an individual keystroke, and those that relate to when and where the previous keystroke was made.

For each feature, mean, standard deviation, skewness, kurtosis will be stored in the dataframe to minimize the impact of the loss of data and fully encompass the scope of the range of actions over time. In addition, since asymmetrical movement is a significant indicator of Parkinson's disease, distinctions relating to that (differences between key holds with both hands and speed in transitioning from one side of the keyboard to the other) will have their own features. All features are denoted in the following table, with 27 in total:

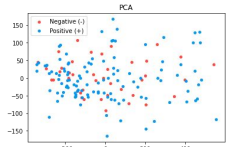
Hold time features (individual keys)	
left hand keys (mean, stdev, skew, kurtosis)	4
right hand keys (mean, stdev, skew, kurtosis)	4
difference between L-R (key press asymmetry)	1
Latency time features (n-key pairs)	
LR, RL keys (mean, stdev, skew, kurtosis)	8
LL, RR keys (mean, stdev, skew, kurtosis)	8
difference between mean LR-RL (opposite hand asymmetry)	1
difference between mean (same hand asymmetry)	1
Total: Hold features (**9**) + Latency features (**18**)	**27**

Methodology

Algorithm

Poly support vector classification (SVC) was selected for this project. It is effective at taking advantage of the large feature space of the dataset, utilizing kernels to find a hyperplane that best separates class instances.

Initially, I tried linear SVC to check if the data was linearly separable after feature selection. However, after performing principal component analysis (PCA) to reduce the dimensionality of the dataset and to maximize potential variance, no clear separation was made present.



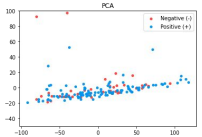

```
model = LinearSVC(multi_class='ovr', class_weight='balanced').fit(x_train, y_train)
```

With 27 distinct features and only 124 samples in the training set, the data was likely subject to the curse of dimensionality, where features become more difficult to cluster together into patterns. In addition, there is a high chance of overfitting the data, which may poorly impact the results of the test set.

Application to the project

Since the hold features were more closely related, I decided to remove the latency features from the dataframe, reducing the feature space by 66%. In addition, the use of poly SVC does not restrict the hyperplane to linearity, allowing for more robust separation of classes.

Applying PCA to the new dataset shows the following:




```
model = SVC(kernel='poly', class_weight='balanced').fit(x_train, y_train)
```

It form a linear relationship, since hold features are closely related together. Longer key hold times on one hand would correlate with longer key hold times on the other hand. Negative and positive training cases are marginally more separable. Thus, poly SVC may have an easier time distinguishing between the classes in the dataset.

Analysis and Results

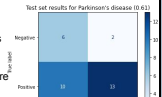
Due to the small sample size, a cross-validation approach would be most effective at making the most of the data. The data was split 80/20 into training and test data sets, resulting in 124 and 31 participants in each respective set. In addition, the model used 5-fold validation from the training set, where a validation fold will have about 24 participants each. The split is stratified to give each fold consistent proportions of those with and without Parkinson's, for a more balanced analysis.

The initial linear SVC model performed very poorly - insignificantly better than guessing at random. Some of the cross-validation folds scored below 0.5, with a mean of 0.57 across all folds and a performance of 0.61 on the test set.

```
Individual cross-validation accuracies: [0.44, 0.64, 0.56, 0.64, 0.58
Mean cross-validation accuracy: 0.573
```

```
Accuracy of linear SVC on test set: 0.61
```

While it was able to determine a sizable proportion of true positive cases where those tested with Parkinson's were accurately predicted as such, it also determined many false negatives, where roughly 33% of those with Parkinson's were predicted not to have it. The results are shown in the confusion matrix to the right:

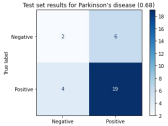


However, the poly SVC model with the reduced features dataset appeared to be more promising. All cross-validation folds were able to score above 0.5, with a mean of 0.69 across all folds. This performance is also reflected in the test set, where it achieved a performance of 0.68.

```
Individual cross-validation accuracies: [0.72, 0.76, 0.72, 0.52, 0.75]
Mean cross-validation accuracy: 0.694
```

```
Accuracy of poly SVC on test set: 0.68
```

The poly SVC model improved by determining significantly fewer false negatives. It was able to predict a majority of true positive cases, as shown in the confusion matrix below. However, it also predicted far fewer true negative cases, instead assuming that some of the negative cases were actually positive. While better than the linear SVC model, the poly SVC model appears to have played it safer and made frequent guesses that a participant was more likely positive, leading to an increase in false positive cases.



Summary/Conclusions

Overall, the poly SVC model serves as a decent indicator towards predicting whether a person has Parkinson's disease. However, it is far from perfect. One of the main limits in examining this dataset is its small sample size, which makes it difficult for the model to learn from the data and determine a separation boundary. In addition, the samples are skewed heavily towards those with Parkinson's, as most volunteers who contributed to this dataset were more likely to have it. This makes the model overpredict false positives, where it will most likely fail if presented with a test set with most people testing negative.

It was a difficult task pre-processing the data as well, since some of the fields were ridden with errors. To get the data into a useable format, I had to cast every column into the proper type, get rid of erroneous data, and simplify the data. If the data was cleaner, it would encourage more people to attempt a solution with this dataset.However, I learned a great deal from this project. I found it particularly insightful seeing how machine learning can tie into our everyday lives and have an application in biology. I would love to do more projects like this.

Key References

[1] Adams, Warwick R. "High-Accuracy Detection of Early Parkinson's Disease Using Multiple Characteristics of Finger Movement While Typing." PLOS ONE, Public Library of Science, 30 Nov. 2017, https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0188226.

[2] Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220.

Acknowledgements

I would like to thank Dr. Yulia Newton for her guidance in this project and for providing resources to further my understanding of class material. I would also like to thank Adams for his initial research conducted on machine learning in respect to Parkinson's disease, as well as PhysioNet for making the dataset publicly available.