Applications of Supervised Learning to Parkinson's Disease Discrimination from Dysphonia Measurements

Michael Walton
Department of Computer Science, Georgia Institute of Technology

September 18, 2015

Abstract

In this work we explore the application of a variety of supervised Machine Learning algorithms to a Parkinson's Disease (PD) speech pathology dataset. Our objective is the development of models capable of discriminating unseen Parkinsonian voice samples from normative (PD-negative) controls. Predictive classification models are principally assessed by comparing their performance as a function of training set sample size. Following from the methods employed in the associated study which accompanies this dataset, we also explore the impact of representing individual subjects' data using central tendency and dispersion measures across multiple samples. Ultimately, we have been able to derive models with comparable PD-discriminating performance to the originating study.

1 Introduction

Parkinson's disease (PD) is a neurodegeneritive central nervous system disorder characterized by progressive loss in motor control, speech and cognitive function. Common symptoms include tremor, difficulty speaking (dysphonia) and dementia. Most treatments for PD are effective only in the early motor stages of the disease which makes early detection crucial for improving patients' quality of life and long-term prognosis.

Today, the majority of PD cases are diagnosed using the Unified Parkinson's Disease Rating Scale (UPDRS). This behavioral test panel assigns a numeric value which encodes the progression of the disease. The observation tests must be administered by a qualified neurologist making PD detection a time consumptive and costly procedure. The desire for a rapid, low-cost pre-screening

diagnostic has motivated a number of studies which attempt to distinguish early and difficult to detect PD symptoms such subtle aperiodic fluctuations in voice recordings. Although a thorough description of these metrics is beyond the scope of this study, it suffices to understand that clinical speech pathology literature has identified vocal acoustic irregularities such as turbulent non-Gaussian noise in early stage PD patients. Determining if variability in these phonations can yeild a robust PD diagnostic remains a subject of active research.

One such study, conducted by Sakar et al. [5] gathered audio samples from 20 persons with Parkinson's (PWP) and 20 otherwise healthy adults. The researchers gathered a series of voice samples from participants and extracted a set of 26 features using various linear and time-frequency based metrics such as jitter, shimmer and pitch. In the present study we define our instance space as the 26 dimensional space of acoustic features sampled in [5]. The target hypothesis is a consistent function mapping elements from the instance space to a binary classification label denoting Parkinson's Disease (PD+) or cognitively normative (PD-). To derive a maximum a-posteriori hypothesis given the aforementioned dataset, we split the data into distinct training and testing subsets and minimize error of a collection of learning algorithms on the training set. The generalization error on the test set is also recorded and compared.

In [5] the researchers also propose a means of reducing the impact of inter-trial variability in samples taken from the same subject. This is accomplished by taking the mean and standard deviation of samples from a common participant such that m participant samples map to a single instance for each participant. In effect, this increases the dimensionality of the instance space by a factor of two, while reducing the number of available samples by a factor of m; making the number of samples equivalent to the number of participants. It is of interest in this study, and the Machine Learning course generally, to understand the relationship between the dimensionality of the input and number of training examples to generalization performance of a learning algorithm. To keep things interesting, we conjecture (in opposition to the results in [5]) that performing this preprocessing step may in fact hinder some algorithms' performance due to the curse of dimensionality in conjunction with a restriction of the number of samples available for training.

The supervised learning algorithms considered in the present study are k-Nearest Neighbor, Support Vector Machines, Artificial Neural Networks, Decision Trees and Boosting. The python modules `scipy, numpy,` and `scikit-learn` [3] [4] are used extensively throughout our implemen-

tation; the GPU-accelerated math compiler Theano [1] is used in conjunction with convenience wrappers `lasagne` and `nolearn` for ANNs; Keka [2] is utilized for pruned decision trees and boosted models.[6]

2 Methods

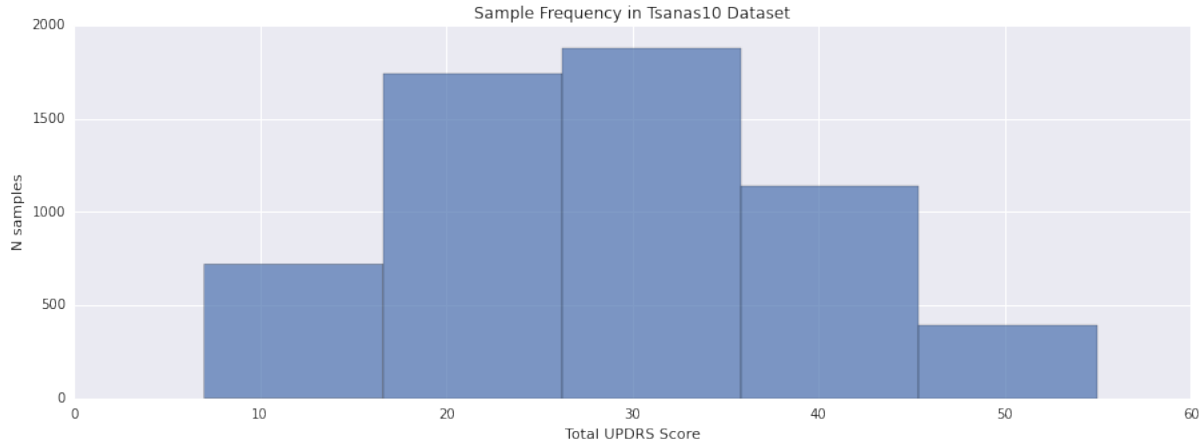
In the sections that follow, we will provide a brief overview of the dysphonia datasets used in this study and the assumptions made in working with these data. In addition, we will briefly discuss the implementation details for each considered algorithm and highlight any model-specific transformations of the data or optimization procedures.

2.1 Data

The two datasets considered in this study originate from Parkinson's Disease clinical diagnostic literature; specifically, from a research domain interested in correlating vocal dysphonia metrics with the disease. In the first study, researchers collected voice recordings from 20 persons with Parkinson's (PWP) and 20 otherwise healthy adults. Multiple types of sound recordings were collected from each participant (including sustained vowels, words, numbers and short sentences). The raw audio samples were then processed into a set of 26 features using various linear and time-frequency based metrics such as jitter, shimmer and pitch. The input data $X \in \mathbb{R}^{n \times m}$ where n is the number of audio samples ($n = 1040$) and m is the dimensionality of the feature space ($m = 26$); target labels are contained in a binary vector $\boldsymbol{y} \in \mathbb{R}^n$. We therefore define our instance space as the 26 dimensional space of vocal dysphonia features sampled in [5]. The target hypothesis is a data-consistent function $h : x \in \mathbb{R}^m \mapsto y \in \{+, -\}$. The objective is some tbd. 'best' function which partitions the instance space by mapping each example $x \in X$ to a corresponding binary classification label $y \in \boldsymbol{y}$ denoting Parkinson's Disease (PD+) or cognitively normative (PD-).

The second dataset considered in our analysis contains similar dysphonia features to [5], however instances are associated with real-valued, continuous UPDRS scores. These data may be readily modeled as a regression problem; in order to render the data suitable for classification we generate proxy labels from the original distribution of UPDRS scores which represent the continuous scale as discrete 'stages' (the number of stages may be arbitrary, 5 has been selected so as to

maintain consistency with the Parkinson's literature). In the associated study, researchers gathered a total of 5,875 voice recordings from 42 PWP. The vocal dysphonia metrics used are largely overlapping with [5], however fewer metrics are considered. We define our target hypothesis as a consistent function $h : x \in \mathbb{R}^m \mapsto y \in [1, 5]$ where x, y, m are defined similarly to the previous study ($m = 16$).



Although the two datasets are very similar in their methods of data collection and overall approach, there are several key differences worth bearing in mind. The shape of these data is interesting in that: the Sakar study possesses less samples and a higher dimensional instance space when compared to Tsanas; it is possible that insufficient data and the curse of dimensionality play a significant factor in the generalization error of algorithms applied to Sakar. However, the sample distribution over classes in Sakar is perfectly uniform (20 in both PD+ and PD- classes), as opposed to the roughly gaussian distribution of UPDRS scores in Tsanas leading to imbalanced classes after partitioning the samples.

Taking these observations further, given the low occurrence of Parkinsons, the real-world performance of a model trained on uniformally distributed samples will be far lower than the estimated generalization error during training and validation. This is due to the fact that the a-priori distribution is only artificially uniform, when in fact the true probability of a randomly selected individual having Parkinson's is only about 0.00013. Though this is still a significant portion, it is difficult to make strong assertions about clinical significance and diagnostic value in this context. In the Tsanas study, however, participants are assumed to be PD+ and the learning task is simply the prediction of the progression of the disease from voice samples. In this context, it seems much more likely that the prior distribution of the collected samples more closely match the true

distribution of PWP.

2.2 Training

To derive maximum a-posteriori hypotheses given the aforementioned datasets, we apply the following preprocessing and training procedures which are roughly the same for all algorithms used:

- Separate the raw-data feature columns X from the label column $\boldsymbol{y}$
- Randomly shuffle and split the data into distinct training and testing sets (1/3 held out for testing)
- Scale the data to zero mean and unit variance using `StandardScaler` (the transformation is computed only from the training set, then applied to all data)
- Train each algorithm using stratified k-folds cross validation. In this method, the algorithm is iteratively evaluated on one of each of the k folds, after using the other $k-1$ folds for training. ($k = 5$ in Sakar experiments, $k = 3$ for Tsanas)[1]

For several algorithms we also applied an exhaustive, cross-validated parameter search using `GridSearchCV`. In this method, every combination of a specified parameter range is iteratively trained and evaluated using the cross-validation method previously described. The optimal parameter set is determined by the parameters that perform best over the average over the associated cross-validation runs.

2.3 Algorithms and Evaluation

In general, the performance measure we will assume is the F1 score. This metric defines accuracy as the harmonic mean of precision[2] and recall[3]. In the case of binary classification, given precision p,

[1] A smaller k was used on Tsanas data to ensure that each fold could preserve the label distribution of the complete set

[2] ratio of true positives to all predicted positives

[3] ratio of true positives to all actual positives

recall r and true positive, false positive and false negative rates denoted tp, fp and fn respectively; F1 is defined as

$$F1 = 2\frac{pr}{p+r} \text{ where } p = \frac{tp}{tp+fp},\ r = \frac{tp}{tp+fn} \tag{1}$$

This definition of accuracy was chosen because an algorithm that optimizes f1 treats precision and recall equally; that is, the learners' ability to correctly label the data is maximized while the occurrence of false positives is minimized. In the case of multi-class classification, as we have defined for the Tsanas dataset, individual scores are computed for each class using a one-vs-rest method and the total F1 score is defined by the average of the individual scores weighted by the frequency of true instances of each label.

Unless otherwise specified, we have attempted to keep the learning algorithm implementation as consistent with the course material as possible; though some liberties have been taken with certain modular components that are particular to our application or the `scikit-learn` library. In the implementation of k-NN, we used the Minkowski distance defined over normed vector spaces which is the generalization of both Euclidian and Manhattan distance. Specifically, given points $X = (x_1, x_2, ...x_n),\ Y = (y_1, y_2, ...y_n) \in \mathbb{R}^n$

$$\delta_m(X, Y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}} \tag{2}$$

We chose to use the Minkowski distance measure to allow straightforward manipulation of the power parameter p[4]. In the implementation of decision trees and the boosted version using the AdaBoost algorithm, the attribute selection criterion normally defined by entropy and corresponding information gain measures was replaced with the GINI impurity[5]; given data D divided into n subsets d_i by an attribute A we define:

$$gini_A(D) = \sum_{i=1}^{n} \frac{|d_i|}{|D|}(1 - \sum_{j=1}^{m} p(c_j|d_i)^2) \tag{3}$$

Which is the sum of the relative probabilities p of label c_j given the subset d_i taken over m classes, weighted by size of the subset in proportion to the total size of the data.

[4] p may take on any arbitrary value, when $p = 1$, the measure is equivalent to the Manhattan distance, when $p = 2$ the measure is equivalent to the Euclidean distance

[5] In practice, we observed little to no difference between GINI impurity and information gain with regards to attribute selection and overall performance.

In our neural network implementation, we elected to use the tangent hyperbolic function *tanh* in place of the generic sigmoid transfer function. This was based on its comparable non-linearity and less compute intensive calculation allowing larger networks and faster updates. Weight updates were performed with stochastic gradient descent using the weight update rule:

$$\omega_i^n \leftarrow \omega_i^{n-1} - \eta \frac{\partial \epsilon_n(\omega)}{\partial \omega_i} = \omega_i^{n-1} - \eta \frac{\partial y_\omega(x_n)}{\partial \omega_i}(y_\omega(x_n) - t_n) \tag{4}$$

We update the weight vector ω of layer i, on the n^th iteration by computing the old ω_i^{n-1} less the error ϵ gradient with respect to ω approximated given n scaled by some learning constant η. The second equation is an expansion where the error gradient is explicitly denoted in terms of the input vector x, target output t and network transformation $y_\omega(x)$ assuming parameters ω.

3 Results

3.1 Model Evaluation

3.2 Performance Comparison

4 Conclusion

SUPPLEMENTAL MATERIALS

Source Code: The algorithms discussed in the present study along with data-handling, post-hoc statistics and visualizations are implemented in the form of an ipython notebook (parkinsons.ipynb)

Dependencies Install Script: A python script which will fetch and install the python modules required by the ipython notebook (dependencies.py)

Parkinson's Dataset: Data set used in this study is contained in the folder pd-msr-data. This folder contains both raw audio samples (.wma) and extracted dysphonia measurements (.csv)

References

[1] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[2] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software. *SIGKDD Explor. Newsl.*, 11(1):10, nov 2009.

[3] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2015-09-15].

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

[5] B. E. Sakar, M.E. Isenkul, C. O. Sakar, A. Sertbas, F. Gurgen, S. Delil, H. Apaydin, and O. Kursun. Collection and analysis of a parkinson speech dataset with multiple types of sound recordings. *IEEE Journal of Biomedical and Health Informatics*, 17(4):828–834, jul 2013.

[6] A. Tsanas, M.A. Little, P.E. McSharry, and L.O. Ramig. Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests. *IEEE Transactions on Biomedical Engineering*, 57(4):884–893, apr 2010.