Extending the tracking of Parkinson’s symptoms through telemonitored speech samples

Donald Dunagan, for the course CSCI 6380, 2 August 2018, *edited 14 October 2018*

**Abstract**

The tracking of Parkinson’s Disease symptoms can be cumbersome due to the necessity of in-person clinic visits and highly specialized medical staff. A logistically simpler option for the tracking of symptom progression, measured on the Unified Parkinson’s Disease Rating Scale (UPDRS), is through the use of remote telemonitoring. From sustained vowel phonations, speech disorder measurements can be extracted, which, when used as features for a machine learning scheme, produce clinically useful UPDRS estimates. While the original work produces estimates within 7.5 UPDRS points of a clinician’s evaluation, the models developed in this paper: k-nearest neighbors, classification and regression trees, random forests, and boosted classification and regression trees, demonstrate improved accuracy, the greatest of which is within 6 UPDRS points.

**1. Introduction**

It is through neurological connections that information is garnered from the outside world and through these same connections that muscles are controlled in the body. Unfortunately, neurodegenerative disorders such as Parkinson’s Disease (PD), Alzheimer’s, and Epilepsy reduce a patient’s capacity, often significantly, for both, through the degradation of neurological pathways. After Alzheimer’s, PD is the most common neurodegenerative disorder [1] with estimates of more than one million people being affected just in North America [2]. Additionally, twenty percent of people with PD end up never being diagnosed [3]. With age as the most dominant risk-factor for the onset of PD and the fact that after the age of fifty, the risk of onset increases greatly [4], the number of people with PD is expected to increase as the worldwide population ages [5]. Current drug treatments are incapable of curing or reversing the effects of PD [6], however, PD specific speech therapies such as the Lee Silverman Voice Treatment have the ability to increase a patient’s quality of life through improvements in facial expression, speech, and breathing and swallowing ability [7].

As part of their treatment, PD patients undergo subjective physical and vocal tests administered by a clinician with the Unified Parkinson’s Disease Rating Scale (UPDRS) being used as a metric to measure the advancement of PD symptoms. The UPDRS is scaled from 0-176 with 176 being total disability. An overall UPDRS value is composed of three subsections: (1) Mentation, Behavior, and Mood, (2) Activities of daily living, and (3) Motor. While symptoms of PD can include tremor, muscular stiffness, and cognitive difficulty, this paper is most concerned with the vocal impairments associated with PD. Specific traits found in the speech of advanced PD patients include dysphonia (difficulty speaking) and hypophonia (quiet speech volume) [8]. Speech difficulty falls under the Motor subsection of the UPDRS evaluation with Motor being scored from 0-108 and 108 signifying significant impairment.

A new medical technology, telemonitoring, is a possible option for PD patients which can not only reduce inconvenient clinic visits, but offer a potentially more accurate assessment of the patient’s speech than that of a clinician [9]. A task which can be easily integrated into a telemonitoring system and has been shown to be effective in PD symptom monitoring is a sustained vowel phonation, which can be analyzed by signal processing algorithms for PD features [10]. While previous studies [10], [11] have demonstrated that the speech of people with PD can accurately be distinguished from that of healthy individuals, Tsanas *et al.* [9] extend this idea to assigning a UPDRS value to a PD patient based on the severity of their voice symptoms. This is done by applying speech signal processing algorithms to a voice sample to extract dysphonia features which are then used in different regression techniques to map speech signal properties to a UPDRS estimation.

This paper applies machine learning algorithms in the context of regression which have not been applied to this data set in attempt to improve upon the results presented by Tsanas *et al.* [9] as well as others. Following sections will discuss the data used and its origin, the results of prior work on this dataset, the machine learning schemes applied in this paper, the results of these schemes, and concluding remarks.

**2. Data and Previous Work**

The dataset used for the analysis in this paper, The Parkinson’s Telemonitoring Dataset [9], comes out of the work done by Goetz *et al.* [12]. In collaboration with Intel Corporation and supervised by ten medical centers in the United States, 52 patients with PD underwent a six-month study in which they used Intel Corporation’s At-Home Testing Device (AHTD) to record their PD symptoms. One facet of each AHTD session was sustained vowel phonations in which the patient would produce a given sound at a prescribed intensity for as long as possible.

To these phonations, Tsanas *et al.* [9] applied signal processing algorithms and extracted 16 dysphonia measures. Data from 10 of the study participants was dropped, 8 due to inadequate test data and 2 who left the study early, resulting in 5875 instances from 42 patients. Each instance is a speech recording of a sustained vowel phonation of the phoneme /a/, “ahh.” The dataset contains 22 attributes: subject number, age, gender, time from recruitment for the study, Motor UPDRS, total UPDRS, and 16 extracted dysphonia measures.

During the six-month trial period, clinical evaluations of UPDRS were taken initially as a baseline, at three months into the study, and at six months at the conclusion. The voice samples, however, were taken on a weekly basis. Because of the lack of weekly UPDRS values, weekly intermediary UPDRS values were interpolated linearly, passing through a patient’s initial, three month, and final UPDRS evaluations. A linear progression of PD symptoms is well supported [13], particularly over a time span of a year or less [14]. The mean baseline motor UPDRS was: 19.42 ± 8.12, at three months: 21.69 ± 9.18, and at six months: 29.57 ± 9.17. The mean baseline total UPDRS was: 26.39 ± 10.80, at three months: 29.36 ± 1.82, and at six months: 29.57 ± 11.92.

Tsanas *et al.* [9] used three linear models and one nonlinear model to predict UPDRS values based on dysphonia measures. The linear models included classical least squares (LS), iteratively re-weighted least squares (IRLS), to reduce the influence of any large, outlying values, and least absolute shrinkage and selection operator (LASSO). The LASSO was used not only in accordance with the general principle of parsimony, but also in anticipation of the curse of dimensionality due to the sixteen input features. It has been shown that dysphonia measures are highly correlated [10], in which LASSO helps due to its capacity to effectively reduce the coefficients of some features towards zero. The nonlinear model used was a classification and regression tree (CART), specifically, a pruned binary regression tree. The testing mean absolute errors (MAE) for motor UPDRS and total UPDRS following 1,000 repetitions of out of sample testing were 6.7 and 8.5 points for LS, 6.8 ± 0.17 and 8.47 ± 0.27 points for IRLS, and 6.8 and 8.6 points for the LASSO, respectively. The CART model performed better than the linear models with a testing MAE of 5.95 ± 0.19 points for motor UPDRS and 7.52 ± 0.25 points for total UPDRS after 1,000 testing repetitions. These results are extended in Tsanas *et al.* [15] when LS is applied to a LASSO selected subset of the 13 classical measures and the log transformations of those 13 measures. After 100 repetitions of out of sample testing, the testing MAE was 6.57 ± 0.16 points for motor UPDRS and 8.38 ± 0.23 points for total UPDRS. Incorporating the log transformed features into the LR regression model reduced error compared to the model which did not. However, even with the improvement, LR did not perform as well as the CART model in Tsanas *et al.* [9].

Additional regression techniques were applied by Eskidere *et al.* [16]: support vector machines (SVM), least squares support vector machines (LS-SVM), multilayer perceptron neural networks (MLP), and general regression neural networks (ANN). The testing MAEs for motor UPDRS and total UPDRS following 100 repetitions of out of sample testing were 5.46 ± 0.14 and 7.02 ± 0.18 points for SVM, 4.87 ± 0.11 and 6.18 ± 0.16 points for LS-SVM, 5.61 ± 0.16 and 7.19 ± 0.22 points for MLP, and 6.36 ± 0.14 and 8.03 ± 0.19 points for ANN, respectively.

Another study to perform regression through the use of neural networks was Hlavica *et al.* [17]. Specifically, feed-forward multilayer neural networks (ANN) and adaptive network-based fuzzy inference systems (ANFIS) were applied to the problem. The reported testing motor UPDRS MAEs were 5.33 points and 5.9 points for the ANN and ANFIS, respectively. Although the authors conclude that these systems perform similarly to other models in the literature, namely those presented by Eskidere *et al.* [16], they argue that neural networks are better candidates for PD assessment systems, particularly in embedded devices, as they are faster to train and require less memory than the LS-SVM implemented by Eskidere *et al.* [16] which performed better than the ANN and ANFIS in this paper.

**3. Methods**

*3.1 Overview of Models*

The goal of this paper is to develop regression models which take as input the sixteen dysphonia measures found in the dataset [9] and predict linearly motor and total UPDRS values. Classification and regression trees (CART) are developed as a baseline comparison to those presented by Tsanas *et al.* [9] in addition to three models not present in the literature regarding this dataset: k-nearest neighbors (KNN), random forests (RF), and AdaBoosted classification and regression trees (ABCART).

CART [18] falls under the more general decision tree machine learning scheme and is a supervised framework capable of both regression as well as classification. The CART algorithm shares many similarities with C4.5 [19] in that information gain is used to split nodes and binary tress are constructed. CART, however, supports regression where as C4.5 only supports classification tasks. Scikit-Learn [20] uses an optimized CART algorithm. Although decision trees are white box models which can be understood and interpreted, they can easily be overfit to the training data and are susceptible to small variations in that training data.

KNN is a supervised learning algorithm capable of being used for both classification as well as regression [21]. It functions as an instance-based learner in that it calculates the closest specified number of neighbors to the query point in terms of training samples and then makes a prediction based on these either by a vote for classification or by taking the mean target value for regression. One option for KNN is to assign weights to neighbors in the neighborhood of the query point such that points that are closer to the query point impact the predicted label more than points which are farther away. When this is done, weights are assigned proportional to the inverse of the distance from the query point. Euclidian distance is the most common distance metric used, but other metrics can be specified.

Ensemble methods are a family of algorithms which work by combining the predications of many base models into one composite model which ideally performs better than any of its given parts. Two camps of ensemble methods are averaging methods, in which the results of the base estimators are averaged, and boosting methods, in which the base models are constructed iteratively with each iteration focusing on difficult to classify observations.

The composite model for RF [22] consists of bagged (bootstrap aggregated) [23] random trees (RT) which are complex predictors with low bias, but high variance. Specifically, a series of bootstrapped samples are taken from the training data and a random tree is constructed from each sample. The randomness in the decision tree is introduced in that at each node split, the tree is limited to a random subset of the possible available features. This increases the bias of a given tree, but with aggregation, the variance of the composite model decreases enough to outweigh this [24]. In general, with bagging, the greater the number of estimators in the composite model, the lower the testing error. However, in addition to requiring a greater amount of computational power, testing error will converge after a certain number of estimators.

In boosting methods, base estimators are sequentially built to reduce the bias of the composite model. Base estimators are often weak learners with high bias but low variance whose predictive capacity need only be better than chance (50%). A common base model for boosting is the decision stump (a decision tree of height 1). The AdaBoost (Adaptive Boosting) algorithm [25] fits weak learners to iteratively updated training data. During each boosting iteration, weights are applied to the training observations such that misclassified observations receive a higher weight and correctly classified observations receive a lower weight. Thus, as the iterative process continues, observations which are difficult to classify receive more and more attention and the composite model becomes more and more capable of correctly classifying the difficult observations. In Sci-kit Learn [20], regression utilizing AdaBoosting is implemented through AdaBoost.R2 [26].

*3.2 Model Development and Testing*

All models developed for this paper were done so using the Python programming language and the Scikit-Learn package [20]. For the development of each model, the data were first shuffled and then split 90-10 into training (5287) and testing (588) instances à la the analysis of Tsanas *et al.* [9]. Then, optimal relevant model parameters in regard to the training data were selected using a 10-fold CV search of a parameter grid. Note, this process was repeated twice for each scheme as optimal model parameters differed depending on whether motor UPDRS or total UPDRS was being targeted. To test each model, the data were shuffled and split 90-10 into training and testing instances, the model, with target tuned parameters, was fit using the training data, predictions were made using the testing data features, and the MAE between the actual and predicted values was recorded. The CART and KNN models underwent 1,000 testing repetitions in emulation of the CART testing in Tsanas *et al.* [9]. RF and ABCART underwent 100 testing repetitions in emulation of model testing in Tsanas *et al.* [15] and Eskidere *et al.* [16].

In order to ensure this methodology was appropriate and to review the CART results of Tsanas *et al.* [9], the first models to be developed were CARTs. In Tsanas *et al.* [9], a “pruning parameter” is used in order to reduce the chance of overfitting. However, the Scikit-Learn [20] CART implementation does not have such a parameter. To avoid overfitting, parameters such as maximum tree depth, maximum number of features to be considered when splitting a node, and the maximum number of leaf nodes were tuned using parameter optimizing cross validated grid search, instead. The optimal parameter values follow. When targeting motor UPDRS, the maximum depth of the tree was 5, the maximum number of features considered when splitting a node was 12, and the maximum number of leaf nodes was 150. When targeting total UPDRS, the maximum depth of the tree was 7, the maximum number of features considered when splitting a node was 14, and the maximum number of leaf nodes was 40.

For both KNN models, the Minkowski metric power parameter was left at the default 2 which is equivalent to using the Euclidian distance measure. The other optimal parameter values follow. When targeting motor UPDRS, the number of neighbors used was 21 and neighbors were weighted inversely to their distance from the query point rather than being weighted equally so that closer neighbors were more influential than neighbors that were farther away. When targeting total UPDRS, the number of neighbors used was 19 and neighbors were weighted inversely to their distance from the query point.

For RF targeting motor UPDRS, the optimal maximum depth of the tree was 65, maximum number of leaf nodes 250, and maximum number of features to consider when splitting a node 16. For RF targeting total UPDRS, the optimal maximum depth of the tree was 20, maximum number of leaf nodes 650, and maximum number of features to consider when splitting a node 15. As would be expected, predictive capacity increased alongside the number of base estimators. However, due to computational costs, the number of base estimators was capped at 100 during the testing sequence.

When trying to tune the parameters of the ABCARTs, it became clear that the algorithm preferred to boost only a handful of times on a moderately strong learner rather than many times on a weak learner. As such, the boosting algorithm was simply given a default CART as the base estimator with the other parameters such as learning rate and maximum number of base estimators left at their default values of 1.0 and 50, respectively.

**4. Results**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Applied Frameworks and Respective MAEs** | | | | | |  |  |
|  |  |  |  |  |  | **Motor UPDRS** | | | **Total UPDRS** |
| **Tsanas et al. 2010** | | |  |  |  |  | |  |  |
| Least Squares (LS) | |  |  |  |  | 6.70 | |  | 8.50 |
| Iteratively Reweighted Lease Squares (IRLS) | | | | |  | 6.80 ± 0.17 | |  | 8.47 ± 0.27 |
| Least Absolute Shrinkage and Selection Operator (LASSO) | | | | | | 6.80 | |  | 8.60 |
| Classification and Regression Tree (CART) | | | |  |  | 5.95 ± 0.19 | | | 7.50 ± 0.25 |
|  |  |  |  |  |  |  | |  |  |
| **Tsanas et al. 2010** | |  |  |  |  |  | |  |  |
| Least Squares(LS) w/log transformed features | | | | |  | 6.57 ± 0.16 | | | 8.38 ± 0.23 |
|  |  |  |  |  |  |  | |  |  |
| **Eskidere et al. 2012** | |  |  |  |  |  | |  |  |
| Support Vector Machine (SVM) | | |  |  |  | 5.46 ± 0.14 | | | 7.02 ± 0.18 |
| Least Squares Support Vector Machine (LS-SVM) | | | | |  | 4.87 ± 0.11 | | | 6.18 ± 0.16 |
| Multilayer Perceptron Neural Network (MLP) | | | | |  | 5.61 ± 0.16 | | | 7.19 ± 0.22 |
| General Regression Neural Network (ANN) | | | | |  | 6.36 ± 0.14 | | | 8.03 ± 0.19 |
|  |  |  |  |  |  |  | |  |  |
| **Hlavica et al. 2016** | |  |  |  |  |  | |  |  |
| Feed-Forward Multilayer Neural Networks (ANN) | | | | |  | 5.33 |  | |  |
| Adaptive Network-Based Fuzzy Inference System (ANFIS) | | | | | | 5.90 | |  |  |
|  |  |  |  |  |  |  | |  |  |
| **Dunagan** | |  |  |  |  |  | |  |  |
| K-Nearest Neighbors (KNN) | | |  |  |  | 6.38 ± 0.16 | | | 8.06 ± 0.21 |
| Classification and Regression Tree (CART) | | | |  |  | 6.01 ± 0.18 | | | 7.63 ± 0.22 |
| Random Forest (RF) | |  |  |  |  | 5.17 ± 0.13 | | | 6.46 ± 0.21 |
| AdaBoosted Classification and Regression Tree (ABCART) | | | | | | 4.70 ± 0.16 | | | 5.95 ± 0.21 |

The testing MAE for the linearly interpolated motor UPDRS was 6.38 ± 0.16 for KNN, 6.01 ± 0.18 for CART, 5.17 ± 0.13 for RF, and 4.70 ± 0.16 for ABCART. The testing MAE for the linearly interpolated total UPDRS was 8.06 ± 0.21 for KNN, 7.63 ± 0.22 for CART, 6.46 ± 0.21 for RF, and 5.95 ± 0.21 for ABCART.

The results for the CART models are in line with the errors for the CART models developed by Tsanas *et al.* [9]: 5.95 ± 0.19 for motor UPDRS and 7.50 ± 0.25 for total UPDRS. In addition to verifying the CART models of Tsanas *et al.* [9], this provides confidence in the model implementations provided by Scikit-Learn [20] as well as the model development and testing methodologies used in this paper. A note is that in this instance, tuning CART parameters such as maximum tree depth, maximum number of features to be considered when splitting a node, and the maximum number of leaf nodes using parameter optimizing cross validated grid search had a similar effect as tuning the “pruning” parameter used by Tsanas *et al.* [9] in reducing overfitting and minimizing testing MAE.

None of the models developed for this paper were linear models and this is for reason. In Tsanas *et al.* [9] and even in Tsanas *et al.* [15], with its linear model improvement, it is shown that UPDRS cannot as adequately be accounted for by a linear combination of dysphonia measures. More flexible models are needed in order to decrease absolute error [16] [17]. This can be seen in the KNN scheme. KNN was the worst performing model this paper applied to the dataset, but it still performed better than the best linear model [15].

Some studies have developed neural network style models [16] [17] as they can have high predictive capacity. On this dataset, however, more traditional, although still complex, models perform better. The three most accurate models were RF, LS-SVM [16], and ABCART, in that order.

**5. Discussion**

Telemonitoring as an emerging new technology and health monitoring tool. The recording of sustained vowel phonations through telemonitoring systems presents an opportunity to enrich the treatment of PD patients through greater ease and accuracy of symptom tracking. From these sustained phonations, measures of disordered speech can be extracted [9] which can then be mapped to UPDRS values through different machine learning schemes. While linear models provide fair estimates [9] [15], prediction ability is improved through the application of more flexible modeling [16].

This paper was primarily an exercise in exploring and applying machine learning schemes to a nontrivial dataset and more could be done than that was in this paper such as extending the scope of the literature review to find examples of regression performed on other datasets of speech samples and a more thorough initial exploration of the data before model development. Potentially, more accurate models could be developed with a better understanding of the data if one were to exclude features or perform principal component analysis. Another topic that could be investigated was the relatively low performance of the neural network models and why this was so.

References

[1] M. C. de Rijk *et al.*, “Prevalence of Parkinson’s disease in Europe: A collaborative study of population-based cohorts. Neurologic Diseases in the Elderly Research Group.,” *Neurology*, vol. 54, no. 11 Suppl 5, pp. S21-3, 2000.

[2] A. E. Lang and A. M. Lozano, “Parkinson’s disease. First of two parts.,” *N. Engl. J. Med.*, vol. 339, no. 15, pp. 1044–1053, 1998.

[3] A. Schrag, “How valid is the clinical diagnosis of Parkinson’s disease in the community?,” *J. Neurol. Neurosurg. Psychiatry*, vol. 73, no. 5, pp. 529–534, 2002.

[4] A. Elbaz *et al.*, “Risk tables for parkinsonism and Parkinson’s disease,” *J. Clin. Epidemiol.*, vol. 55, no. 1, pp. 25–31, 2002.

[5] S. K. Van Den Eeden, “Incidence of Parkinson’s Disease: Variation by Age, Gender, and Race/Ethnicity,” *Am. J. Epidemiol.*, vol. 157, no. >11, pp. 1015–1022, 2003.

[6] N. Singh, V. Pillay, and Y. E. Choonara, “Advances in the treatment of Parkinson’s disease,” *Prog. Neurobiol.*, vol. 81, no. 1, pp. 29–44, 2007.

[7] S. Sapir, J. L. Spielman, L. O. Ramig, B. H. Story, and C. Fox, “Effects of Intensive Voice Treatment (the Lee Silverman Voice Treatment [LSVT]) on Vowel Articulation in Dysarthric Individuals With Idiopathic Parkinson Disease: Acoustic and Perceptual Findings,” *J. Speech Lang. Hear. Res.*, 2007.

[8] A. K. Ho, R. Iansek, C. Marigliani, J. L. Bradshaw, and S. Gates, “Speech Impairment in a Large Sample of Patients with Parkinson’s Disease,” *Behav. Neurol.*, vol. 11, no. 3, pp. 131–137, 1999.

[9] A. Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig, “Accurate Telemonitoring of Parkinson’s Disease Progression by Noninvasive Speech Tests,” *IEEE Trans. Biomed. Eng.*, vol. 57, no. 4, pp. 884–893, Apr. 2010.

[10] M. A. Little, P. E. McSharry, E. J. Hunter, J. Spielman, and L. O. Ramig, “Suitability of dysphonia measurements for telemonitoring of Parkinson’s disease,” *IEEE Trans. Biomed. Eng.*, vol. 56, no. 4, pp. 1015–1022, 2009.

[11] B. Harel, M. Cannizzaro, and P. J. Snyder, “Variability in fundamental frequency during speech in prodromal and incipient Parkinson’s disease: A longitudinal case study,” *Brain Cogn.*, vol. 56, no. 1, pp. 24–29, 2004.

[12] C. G. Goetz *et al.*, “TESTING OBJECTIVE MEASURES OF MOTOR IMPAIRMENT IN EARLY PARKINSON’S DISEASE: FEASIBILITY STUDY OF AN AT-HOME TESTING DEVICE,” *Mov Disord*, vol. 24, no. 4, pp. 551–556, 2009.

[13] P. L. S. Chan and N. H. G. Holford, “Drug Treatment Effects on Disease Progression,” *Annu. Rev. Pharmacol. Toxicol.*, vol. 41, no. 1, pp. 625–659, 2001.

[14] W. M. M Schüpbach *et al.*, “Segmental progression of early untreated Parkinson’s disease: a novel approach to clinical rating,” 2009.

[15] A. Tsanas, M. A. Little, and O. Patrick E. McSharry Lorraine Ramig, “Enhanced Classical Dysphonia measures and sparse Regression for Tele Monitoring of Parkinson Disease Progression,” *Int. Conf. Acoust. Speech Signal Process.*, no. March, pp. 594–597, 2010.

[16] Ö. Eskidere, F. Ertaş, and C. Hanilçi, “A comparison of regression methods for remote tracking of Parkinson’s disease progression,” *Expert Syst. Appl.*, vol. 39, no. 5, pp. 5523–5528, 2012.

[17] J. Hlavica, M. Prauzek, T. Peterek, and P. Musilek, “Assessment of Parkinson’s disease progression using neural network and ANFIS models,” *Neural Netw. World*, vol. 26, no. 2, pp. 111–128, 2016.

[18] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. 1984.

[19] J. R. Quinlan, *C4.5: Programs for Machine Learning*. 1992.

[20] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, 2012.

[21] N. S. Altman, “An Introduction to Kernel and Nearest-Neighbour Nonparametric Regression,” *Am. Stat.*, vol. 46, no. 3, pp. 175–185, 1992.

[22] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[23] L. Breiman, “Bagging Predictors,” Kluwer Academic Publishers, 1996.

[24] L. Breiman, “Arcing classifier (with discussion and a rejoinder by the author),” *Ann. Stat.*, vol. 26, no. 3, pp. 801–849, 1998.

[25] Y. Freund and R. E. Schapire, “A desicion-theoretic generalization of on-line learning and an application to boosting,” vol. 55, 1997, pp. 23–37.

[26] H. Drucker, “Improving regressors using boosting techniques,” *14th Int. Conf. Mach. Learn.*, pp. 107–115, 1997.

[27] K. Pearson, “LIII. On lines and planes of closest fit to systems of points in space,” *Philos. Mag. Ser. 6*, vol. 2, no. 11, pp. 559–572, 1901.