Operations Research II Project

Distance Weighted Discrimination Method for Parkinson's for Automatic Classification of Rehabilitative Speech Treatment for Parkinson's Patients

Nicol Lo

1. Introduction

High Dimension Low Sample Size (HDLSS) statistical analysis is an emerging area of significance in multivariate analysis and machine learning, where the dimension of the data d is often much larger than that of its sample size n ($d >> n$). In particular, classification of HDLSS data is especially relevant for in health sciences research, such as genetic micro-array and medical imaging analysis, where obtaining a correct diagnosis is often the main goal.

Commonly used classification methods include statistical ones such as logistic regression and linear discriminant analysis, as well as machine learning-based ones such as the support vector machine (SVM) method.

However, these methods are not immune to the curse of dimensionality. Moreover, many classification methods cannot be employed when there are more variables than samples. Therefore before classification methods are applied, HDLSS data are often pre-processed using feature selection, the process of selecting a subset of variables relevant to model construction, thus reducing dimensionality.

While this can avoid issues such as 'overfitting' and increases interpretability of the results, it can ignore interdependence between variables commonly seen in biological data. As a result, there are classification methods specifically designed for HDLSS analysis. The Distance Weighted Discrimination (DWD) method is one of them. Also formulated as an optimization problem, it is developed based on SVM and is solvable using an interior-point method call Second-Order Cone Programming.

The goal of this project is to try to develop a binary classifier algorithm for a HDLSS Parkinson's Disease speech dataset analyzed in a 2013 research paper. Without using any feature selection methods, we will test out both DWD and SVM to compare classification performance between a method optimized for HDLSS analysis and one that is not. Since the algorithm in the original paper involves both feature selection and SVM method, we can investigate whether feature selection influences classification performance for this dataset.

2. Problem Description

Background

Parkinson's disease is degenerative neurological disorder primarily characterized by the progressive deterioration of motor functions. A result of losing motor control over the muscles producing speech such as lips and tongue, many Parkinson's patients experience significant vocal impairment and speech difficulties. Typical symptoms include reduced volume, monotonous voice, hoarseness and imprecise articulation. Interestingly, the degree of PD-induced vocal deficiencies can be assessed with running speech, or sustained vowel phonations.

The sustained vowel "ahh…" (or /a/) has been sufficient for voice assessment applications to assess the degree of PD vocal impairment.

Lee Silverman Voice Treatment (LSVT) Companion is a computerized speech treatment program developed to allow Parkinson's patients to be engaged in a treatment session independently. We would like to develop an algorithm such that using data collected during treatment sessions, we can provide instantaneous feedback on whether the subject's speech is considered "acceptable" or "unacceptable".

In total we have 126 samples obtained from 14 subjects with Parkinson's Disease (8 males and 6 females), who had a mean age of 65 and standard deviation of 6.5 years. For each sample, we are given data on 310 dysphonia measures, speech information extracted by speech signal processing algorithms, and whether they are considered "acceptable" by clinician. Note that only 42, or a third of the samples are considered acceptable, and the rest considered "unacceptable".

Methodology Comparison

Tsanas et al. 's model involves selecting a ranked feature subset with the weighted feature selection algorithm LOGO (fit locally, think globally), then SVM with a Gaussian radial basis kernel as a binary classifier. Using the top 8 selected features consistently generates a mean prediction accuracy rate of around 90% with 10-fold cross validation (113 samples, 13 test cases).

In this project, we will apply SVM and DWD on the dataset without any feature selection, and compare their 10-fold cross validation performance with that from Tsanas et al's model. Since the algorithm in the original paper involves both feature selection and SVM method, it will also be interesting to see how feature selection influences classification performance. We would also like to investigate how the size of the training sample affect HDLSS data classification performance for the two methods.

3. Classification methods *

For our proposes we will only discuss 1) binary classification and 2) linear discrimination methods, as linear kernels tend to be comparable to non-linear ones in high-dimensional settings.

Define two classes with class label +1 and -1, where +1 denotes "acceptable" and -1 as "unacceptable" speech samples. Let (x_i, y_i) , $i \in \{1, \dots, n\}$ be our training data set, where each observation x_i is a vector of dimension d, and y_i a class label with $y_i \in \{-1, +1\}$. y is then the n-vector of y_i's.

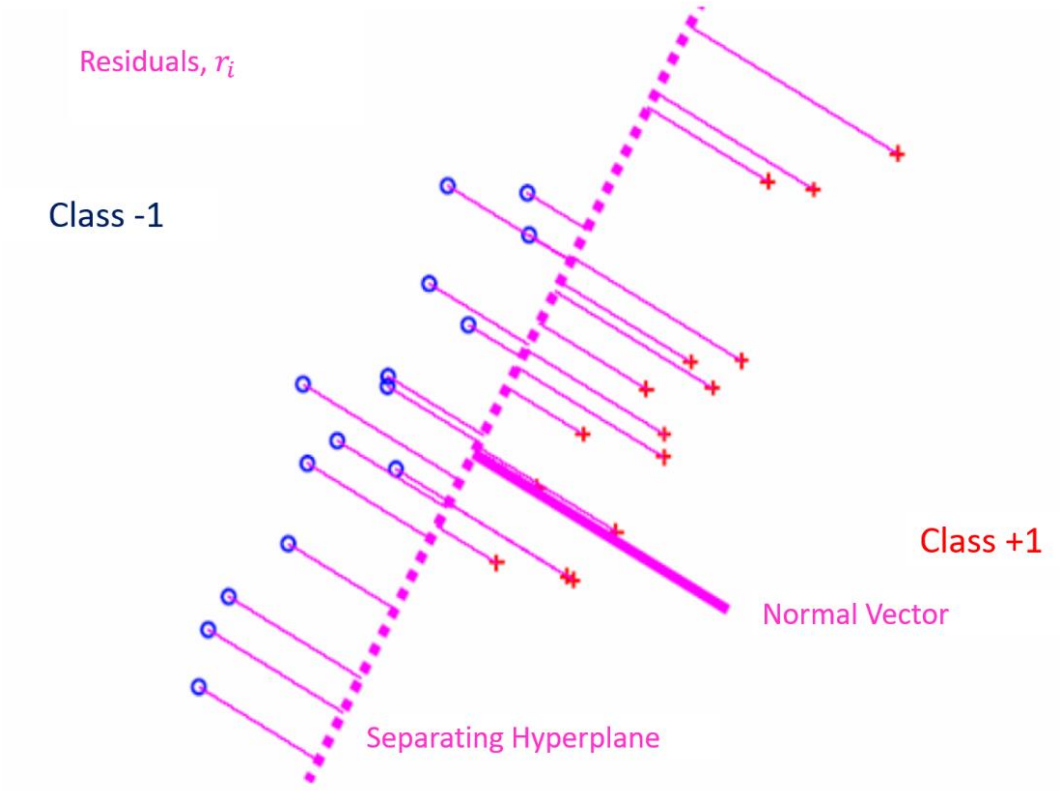
Let X be the $d \times n$ matrix with n columns of x_i's, and Y be the $n \times n$ diagonal matrix whose diagonal components are components of y. If we think of our data as a d-dimensional dataspace and $x_i's$ as data points in the space, we want to find a separating hyperplane to keep data of the same class on the same side of the plane.

The figure below illustrates a dataset with d=2, n = 30 with 15 points in each class, where our hyperplane is a line separating the two classes in a two-dimensional space. The hyperplane is defined by the normal vector $w \in \mathbb{R}^d$ and the position vector $\beta \in \mathbb{R}$.

Define the residual of an observation x_i as $\overline{r_i} = y_i(x_i'w + \beta)$, which is the distance of observation i to the hyperplane. Note that r_i is positive when it lies on the side of the plane of its class, and is negative when it is misclassified by the defined hyperplane. We would like to choose w and beta such that all rs are positive. As such, $r = YX'w + \beta y$ in matrix notation.

Distance Weighted Discrimination

Figure 1. Toy sample illustrating Distance Weighted Discrimination[1]
Class +1 data shown as red plus signs, and Class -1 data shown as blue circles. The separating hyperplane is shown as the thick dashed line, with the corresponding normal vector shown as the thick solid line. The residuals, r_i, are the thin Lines..



[1] Modified from Cime, Addy M. Bolivar, J.S. Marron *Comparison of Binary Discrimination Methods for High Dimension Low Sample Size Data.* Available from http://www.stat.rice.edu/~jrojo/PASI/PASI2011/Dr.RojoPasi2011/prgacad/13.pdf

Distance Weighted Discrimination (DWD in the rest of the text) is an optimization method designed to minimize the "data piling" issue with various classifiers in HDLSS statistical analysis.

DWD aims to maximize the distance of *every* observation to the separating hyperplane by *minimizing the sum of the inverse of every residual*. If our data is truly linearly separable, all r_is are positive, and our problem can be formulated as follow:

$$\min_{r,w,\beta} \sum_i \frac{1}{r_i}$$

w.r.t. $r = YX'w + \beta y \geq e$ (hyperplane correctly separates every observation)
$\|w\| \leq 1$

Often our data is not linearly separable (i.e. there does not exist a hyperplane than divides our data without misclassifying at least one x). In this case we would need to modify the problem above because misclassified points would have negative residuals. We would need to introduce an error vector $\xi \in \mathbb{R}^n_+$ as our slack variable, such that $\xi_i = 0$ when observation xi lies on the proper side of the hyperplane and $\xi_i = 1$ when it lies on the "wrong" side. We also need to introduce a penalty factor $C > 0$ that we need to define ourselves. Our refined residuals are now $r = YX'w + \beta y + \xi$, and our new problem is now:

$$\min_{r,w,\beta,\xi} \sum_i \frac{1}{r_i} + Ce'\xi$$

w.r.t. $\bar{r} = YX'w + \beta y + \xi \geq e$ (penalized residuals greater or equal to 0)
$\|w\| \leq 1$
$\xi \geq 0$ (slack variable allows for errors)

Note that with DWD no data points are excluded; it takes all data into consideration but gives more significance to those closer to the hyperplane.

Support Vector Machine

Figure 2 shows a toy sample of SVM. SVM only considers selected points closest to the hyperplane called 'support vectors', indicated by the black boxes in Figure 2. Define two hyperplanes parallel to the separating hyperplane that intersect the support vectors, shown as the two black dashed lines, and define the distance between these hyperplanes as the 'margin' δ. SVM aims to maximize δ. In the truly linearly separable case, the problem is formulated as:

$$\max \ \delta$$

w.r.t. $YX'w + \beta y \geq \delta e$ (All observations correctly separated; the closest distance to an observation of another class is at least δ)

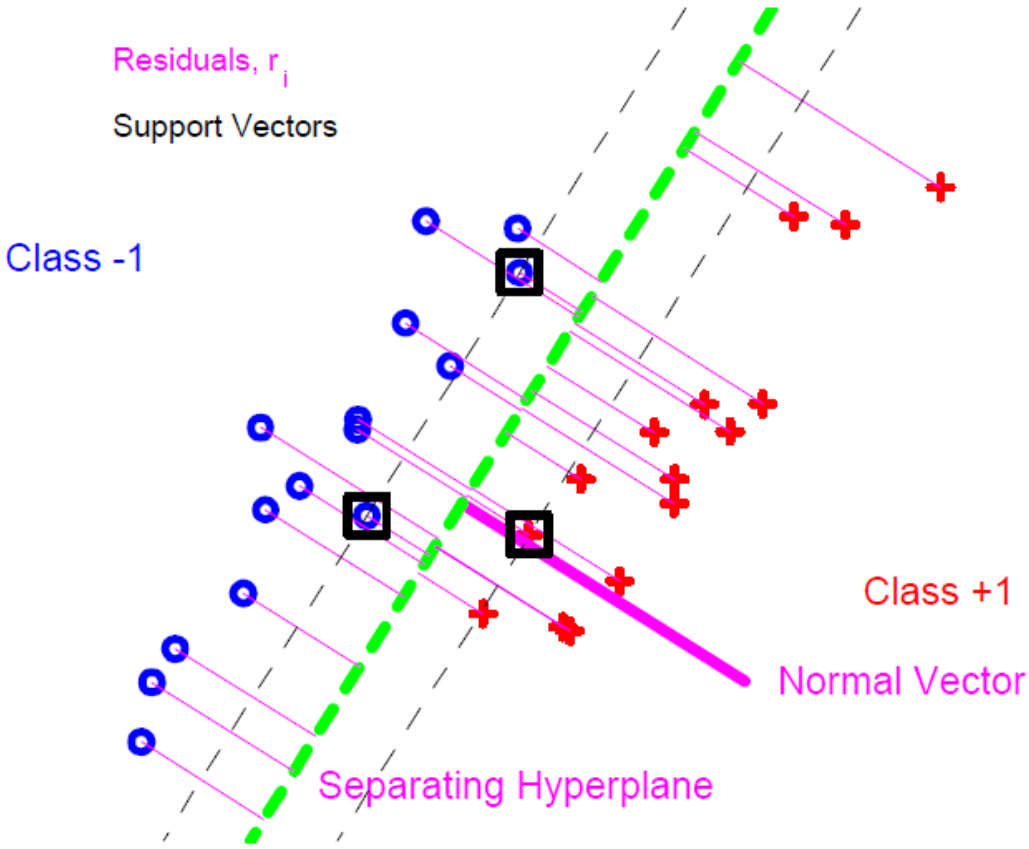
$$\|w\| \leq 1$$

This is the dual form of the optimization problem. Note the second constraint is quadratic. Since it is easier to implement quadratic problem with all linear constraints, SVM is often reformulate as:

$$\min_{w,\beta} \ \frac{1}{2}\|w\|$$

w.r.t. $YX'w + \beta y \geq e,$ (hyperplane correctly separates every observation)

Figure 2. Toy sample illustrating the Support Vector Machine Method. The separating hyperplane is shown as the thick dashed green line, the margin as the thin dashed black line. The support vectors are highlighted with black boxes



Details can be found in Marron's paper. We can modify it for the non-linearly separable case as:

$$\min_{w,\beta,\xi} \ \frac{1}{2}\|w\| + Ce'\xi$$

w.r.t. $\bar{r} = YX'w + \beta y + \xi \ \geq e,$
$\xi \geq 0$

4. Mathematical Model (Problem formulation)

To determine classification performance, we need to randomly select a subset of our data as our training sample (size n_{tr}) and the rest as our testing sample (size n_{tst}). We will fit our model on the training data, predict responses to our test data, and compare our predicted responses to the real responses.

Remember that the both methods require some tuning parameter C to start. Since there is no fix way of choosing C, for each iteration we will try out 10 values of C via 10-fold cross validation with our training data, and choose the one with the highest accuracy rate. We define $C = [2^{-5}, 2^{-\frac{17}{5}}, \dots, 2^{2}]$.

Notation

4.1 General Inputs

- d: Number of features/dysphonia measures: $d = 310$
- n: Number of total speech samples: $n = 126$
  - n_{tr}: Number of samples in the training set, where $n_{tr} \in \{10,20 \dots ,110,113\}$
  - n_{tst}: Number of samples in the testing set, where $n_{tst} = n - n_{tr}$
- $X_{i,j}$: The value of the i^{th} feature for of the j^{th} sample, where $X \in \mathbb{R}^{d \times n}$
- C^*: The penalty factor for misclassification chosen through 10-fold cross validation, where $C^* \in [2^{-5}, 2^{-\frac{17}{5}}, \dots, 2^{2}]$.
- ξ_j : Indicator of whether the j^{th} sample is misclassified, $\xi_j \in \{0,1\}$

4.2 Input Constraints

- Positivity: $n, n_{tr}, n_{tst}, d, C^* > 0$
- Residuals are properly penalized: $\bar{r} = YX'w + \beta y \ + \xi \geq e$
- (For DWD only) The separating hyperplane lies between the two classes in every dimension: $\|w\| \leq 1$

4.3 Response variable

- y_i : Indicator variable for which class the i^{th} observation is assigned to for some $i \in \{1, \dots, n\}$, such that:

$$y_i := \begin{cases} +1 & \text{if "acceptable"; a clinician would allow persisting in speech treatment} \\ -1 & \text{if "unacceptable"; a clinician would not allow persisting in speech treatment} \end{cases}$$

Define $Y \in \mathbb{R}^{n \times n}$ such that $Y_{i,j} := \begin{cases} y_i & \text{if i=j} \\ 0 & \text{otherwise} \end{cases}$

5. Analysis

5.1 Data Pre-Processing

The original response variable Y with $y_i \in \{1, 2\}$ for the two classes has been transformed such that $y_i \in \{-1, 1\}$ as defined in section 4 for clarity. All data are then normalized to have a range from -1 to 1 using min-max normalization, with mins and max obtained from the training data to prevent our results be dominated by measures with wider ranges of possible values.

Another problem is that only 42, or about a third of our data are considered "acceptable". This means it is possible that our randomly selected X_{tr} contains only data from one class when size of the training sample is small (e.g. $n_{tr} = 10$), which makes classification impossible. To prevent this issue, the we developed a "random" data sampler that would regenerate samples when either class represent less than 25% of the training sample. This ensures the performance when using smaller training sample sizes is not due to a skewed ratio.

5.2 Algorithm

The algorithm is as follow:

1. **"Random" selection.** Randomly select sample of size n_{tr} as our training data X_{tr}, and rest as our testing data X_{tst}. Divide Y into Y_{tr} and Y_{tst} accordingly.
    a. Note that for our case $n_{tst} = n - n_{tr} = 126 - n_{tr}$.
    b. To prevent our selected sample to only contain
2. **Normalization**. Apply min-max normalization to both X_{tr} and X_{tst} using the mins and maxs of each feature in X_{tr} to obtain X_{tr}^N and X_{tst}^N.
3. **Fit model using DWD.**
    a. Use DWD method to fit a model M_{DWD} with X_{tr}^N, Y_{tr} and each of the 10 values of C we initialized.
    b. Using 10-fold cross validation, choose the C^* with the highest accuracy rate.
    c. Predict responses to X_{tst} using M and C^*, obtain predicted response Y'_{tst} .

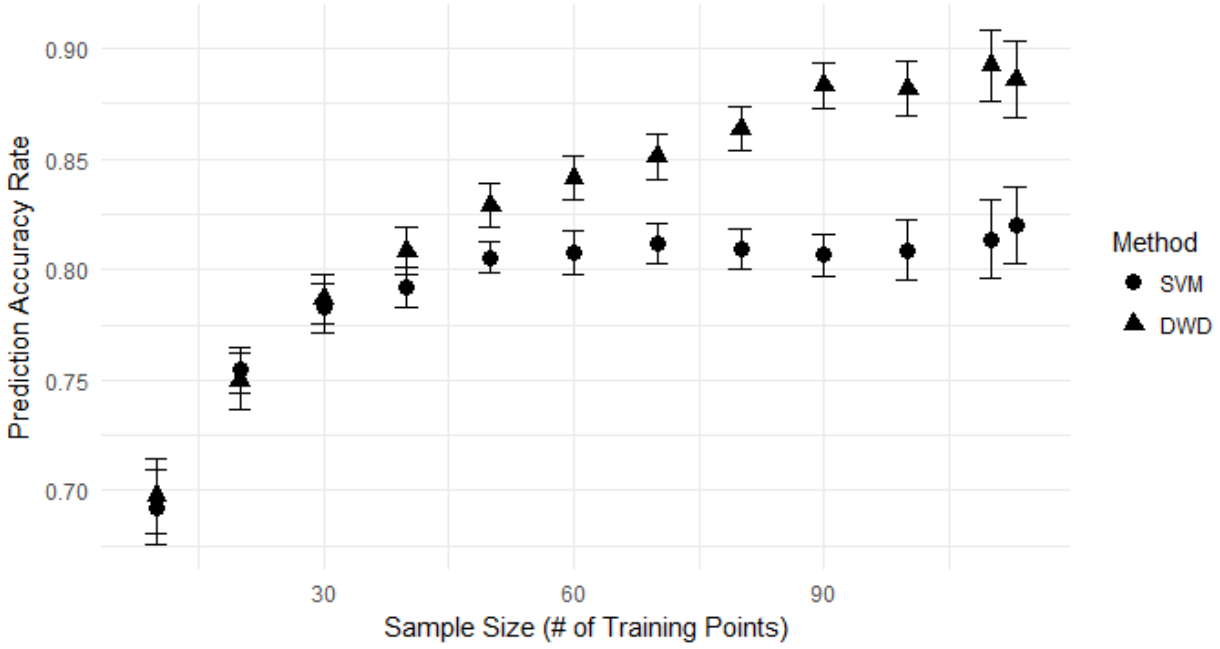
d. Calculate accuracy rate by comparing Y'_{tst} and Y_{tst}.

4. **Fit model using SVM.** Repeat Step 3, but use SVM instead when fitting the model.
5. Repeat Steps 1 to 4 for100 times.
6. Calculate mean classification performance and standard error of the mean for each value of n_{tr}.

6. Results

Using the algorithm detailed in section 5.2, we calculated the mean prediction accuracy rate and its standard error for each sample size to test our classifier performance as detailed in the table and graph below. Error bars indicates standard error of the mean

Figure 3. Comparison of prediction accuracy rate between DWD and SVM



When the training sample is very small ($n_{tst} \leq 30$) there is no noticeable difference in performance between DWD and SVM. With accuracy rates of around 69% at 10 samples and around 75% at 20 samples, both methods did surprisingly well with a very small training data set.

Interestingly, when sample size exceeds 40 classification performance of SVM is not affected by further increases in sample size, plateauing around 80%. In contrast, the performance of DWD continues to improve as number of training points grow, reaching 89.25% at 110 samples. DWD outperforms SVM significantly at larger sample sizes, with a 7.67% difference at 90 samples. Overall, DWD has a superior performance

Table 1. Classification Performance of DWD and SVM

# of Samples	DWD		SVM	
	Accuracy Rate (%)	**SE**	**Accuracy Rate (%)**	**SE**
10	69.74	0.883	69.22	0.846
20	74.92	0.852	75.44	0.524
30	78.66	0.64	78.25	0.552
40	80.85	0.574	79.19	0.467
50	82.92	0.541	80.55	0.352
60	84.15	0.496	80.77	0.488
70	85.11	0.524	81.2	0.457
80	86.37	0.499	80.96	0.452
90	88.31	0.526	80.64	0.494
100	88.19	0.609	80.88	0.677
110	89.25	0.8	81.38	0.87
113	88.62	0.864	82	0.883

7. Discussion

Even though our algorithm doesn't involve feature selection or other dimensionality reduction methods, our DWD model achieved an accuracy rate of 88.62% using 10-fold cross validation, comparable to the 90% obtained by Tsanas et al.'s two-step model (feature selection, then SVM for classification).

Considering that our DWD model does not involve a feature selection process, we can assume that it is easier to implement and more computationally efficient than Tsanas' model. However, models with less features are much more interpretable, and thus contains more useful information. By reducing the number of features considered, Tsanas' model highlights the 8 major features that distinguish 'acceptable' and 'unacceptable' sustained vowel phonations, a result that can be potentially be clinically-useful. With 300+ features, our DWD model cannot give us any information on individual features and their contributions to our prediction.

8. References

Marron, James Stephen, Michael J. Todd, and Jeongyoun Ahn. "Distance-weighted discrimination." *Journal of the American Statistical Association* 102.480 (2007): 1267-1271.

Tsanas, M.A. Little, C. Fox, L.O. Ramig: "Objective automatic assessment of rehabilitative speech treatment in Parkinson's disease", IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 22, pp. 181-190, January 2014

Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification." (2003): 1-16.