Project Report- CMPS142

Anirudh AV (aanumala@ucsc.edu)

1. Introduction

This article is focused on comparisons of 2 different classification algorithms based on their performance on a given problem. Section 3 describes how to reproduce all the results claimed in this report.

1.1. Parkinson's Disease (PD)

Parkinson's disease is a long term degenerate disorder of the central nervous system. Tremor is the major symptom that is indicative of Parkinson's disease. Frequency of a PD tremor is between 4Hz and 6Hz. A feature of the tremor is pill-rolling. The tendency of the index finger of the hand to get into contact with the thumb and perform together a circular movement. This motion was captured by a set of 24 sensor readings on a glove sleeve on the hand of a person suffering from PD.The different motions characteristic of a PD generated unique readings from the sleeve.

1.2. Essential Tremor (ET)

Essential tremor (ET, also referred to as benign tremor, familial tremor, or idiopathic tremor) is the most common movement disorder; its cause is unknown. It typically involves a tremor of the arms, hands or fingers. Similar to the PD case, a sleeve was used to extract the data-set about the subject and the same number of axis readings were generated for each subject.

1.3. About the Data

As mentioned above the data was generated from 24 different sensor readings from a sleeve. The intuition is that ET and PD have different signatures of tremors that are characteristic of the individual type of disease. An FFT was computed on the 24 different sensor readings and each reading gave rise to an amplitude-frequency value pair. Thus, now there were a total of 48 readings for each person. The data set used for this report was generated from 22 people who pretended to have ET and the same 22 people also pretended to have PD.

1.4. Interested Features

Although the PD tremor and the ET tremor is very representative of the age, weight and gender of the subject, this report does not consider any of those features as the data generated for the set was "simulated" by people pretending to have either PD or ET. Thus the tremor signatures will not be representative of the individual subject. The sole purpose of this report is to classify between the two diseases and identify the class only based on the type of motion signatures observed.

1.5. Setup

The Data has 48 features and 22 examples for each kind. The data is split (50/50) between training and test phases.

- The design matrix for the training is 'X', which is a 11 by 48 matrix.
- The set of labels 'y' is a vector of length 11 by 1 with values 0 for PD and 1 for ET.
- The matrix 'X_test' is similarly for the test data (11 by 48).
- The vector 'y_test' is the label vector for the test data (11 by 1).
- Since the data was generated sequentially in a particular order, the MATLAB function **randperm** was used to shuffle to the examples.

2. Approach

The report focuses mainly only approaches to classify the two diseases from the given sensor readings. In particular, it focuses on the performance of two different algorithms and tries to layout the reasons for the observations. The two algorithms are :-

- Logistic Regression with L1 regularization. LASSO.
- SVM with a soft-margin.

2.1. LASSO

The LASSO fits a linear model

$$\hat{y} = w0 + w1 * x1 + w2 * x2 .... wn * xn$$

2.1.1. Notation

- $\hat{y}$ is the estimate made by the model.
- w is the vector of the weights learned by the model.
- n is the number of features chosen for this model.
- X is the input example matrix.

2.1.2. LASSO mathematical definition

There are multiple approaches to solve the LASSO problem, the approach carried out in this article was to use a quadratic program. The LASSO problem is defined as :-

$$min_w \Sigma(y - \hat{y}) \tag{1}$$

$$\text{Subject to } |w| < s \tag{2}$$

Where s can be thought of as the 'regularization' parameter. When the value of s is large , the effect of the constraint becomes minimal and the problem can be treated as a general Least-squares problem. When the value of s is small, then the solution is a shrunken version of the least squares estimate.

2.1.3. LASSO as a quadratic programming problem

The **quadprog** function in MATLAB solves the problem

$$\frac{1}{2} x^T H x + f^T x$$

Subject to :-

$$Ax < b$$

In order to solve the lasso problem, equations (1) and (2) have to be fitted into the above two equations. That is done according to the steps shown below,

2.1.4. Transforming LASSO into a quadratic program

$$\Sigma(y-\hat{y})^2 \text{can be wrttein as}$$
$$(y-Xw)^T(y-Xw)$$
$$=(y^T-w^TX^T)(y-Xw)$$
$$=(y^Ty-y^TXw-w^TX^Ty-w^TX^TXw)$$
$$=y^Ty-2y^TXw-w^TX^TXw$$

The quadratic program in MATLAB has the following syntax:-

$$x=quadprog(H,f,A,b)$$

Where:-

$$\text{H} = \begin{vmatrix} X^TX & 0 \\ 0 & 0 \end{vmatrix}$$

$$f=[-2X^Tyy^Ty]$$

$$\text{A}=\begin{vmatrix} -1 & 0 & 0\ldots & \\ 0 & -1 & 0\ldots & \\ . & & & \\ . & & & \\ 0 & . & . & -1 \end{vmatrix}$$

$$b=\begin{vmatrix} s \\ s \\ . \\ . \\ s \end{vmatrix}$$

Feeding this into the quadratic program we get a vector q where the first n rows is the weight vector w.

2.1.5. Computing the estimates

Once the weight vector has been computed with the above formulation and fed into the quadratic program, the estimates are computed by

$$\hat{y}=Xw$$

Then the training accuracy and the test accuracy are computed :-

- Training accuracy - 100%
- Testing accuracy - 95.4%

The hyper parameter s is best chosen using cross-validation. The effect of the parameter s on the prediction power of the model is shown in the diagram below:-

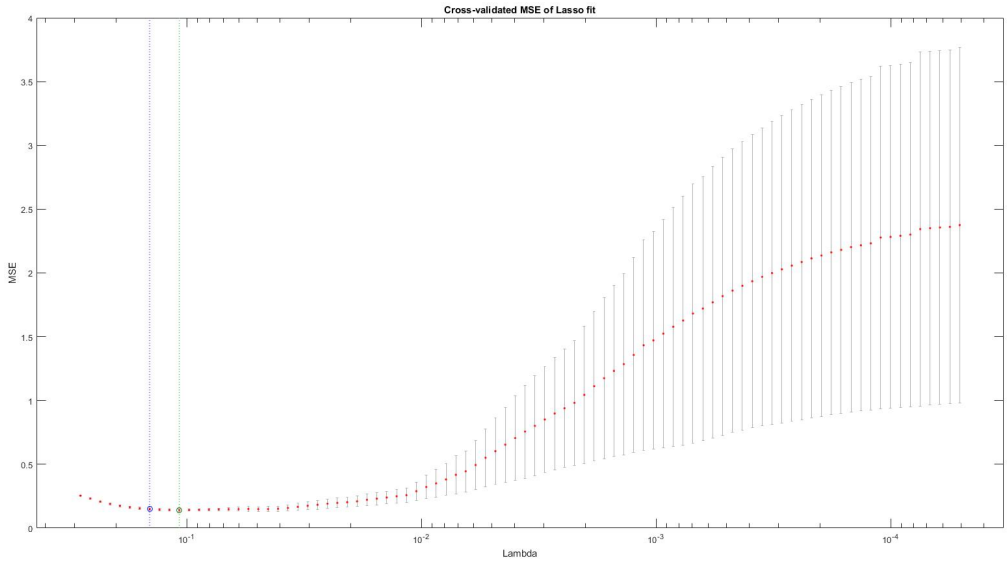


Figure 1: MSE with cross validation as a function of the regularization parameter.

2.1.6. Observation

The green line shows the s at which we observe minimum MSE and the blue line shows the MSE which is one std-deviation away from minimum MSE. MSE is mean squared error and lambda is the regularization parameter. The error increases as the lambda decreases beyond a certain value.

That is because , here lambda is a regularization parameter which helps prevent over-fitting. Once the value becomes really small, the effect of smoothing diminishes and the problem turns into a regular least squares problem with no tight bounds to stop it from over-fitting the data. The model has a lot of variance.

Thus with a really low Lambda value , the training error is generally low as it captures the exact behavior of the training data, but it does not perform very well on the test data.

The initial error is also high, suggesting that , with a high Lambda value, the weights of the co-efficients are constrained to a grater extent and cannot capture the entire behavior of the data. This leads to under-fitting.

The plot below shoes the trace of the LASSO co-efficients as a function of the L1 norm of the prediction error for different values of Lambda:-

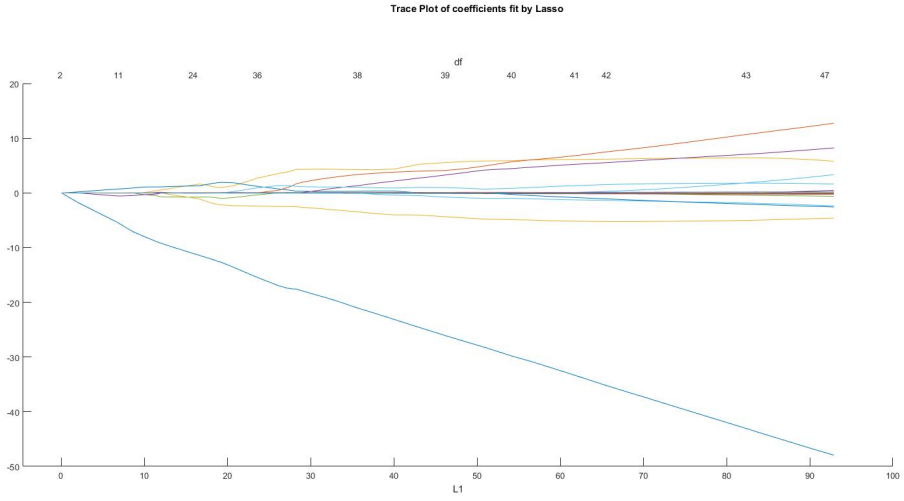


Figure 2: Trace of co-efficients with the L1 norm

2.2. SVM

The maximal margin classifier is an interesting method to use for this problem as it can be seen that the data is not that easily linearly separable in 2-D but as we move to the 48D higher dimensional space, we achieve a much higher accuracy. Kernerlization did not work very great with svm. There could be several reasons behind this:-

- We are more interested in sparsity here
- There is no need for shifting to a higher dimensional space, it's 48D already.
- The linear kernel that computes the inner product between 2 examples, it could be suggestive of the fact that the similarity between 2 examples is not really significant.

2.2.1. SVM formulation

The soft-margin svm is used in this case to allow for slack a variable. The soft-margin svm- can be described by :-

$$\frac{\beta}{2}||w||_2^2 + 1^T\varepsilon$$

$$\text{such that} \varepsilon \geq 1 - \Delta(y)(Xw + 1b)$$

$$\varepsilon \geq 0$$

2.2.2. Results

- Training accuracy:- 100%
- Testing accuracy :- 90%
- Testing accuracy with rbf kernel 77%

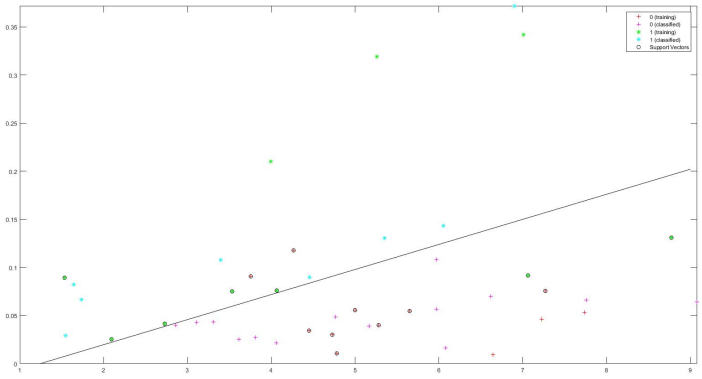


Figure 3: The data is more or less linearly separable in the x2-x3 plane, it provides a training accuracy of 81% and a testing accuracy of 77%

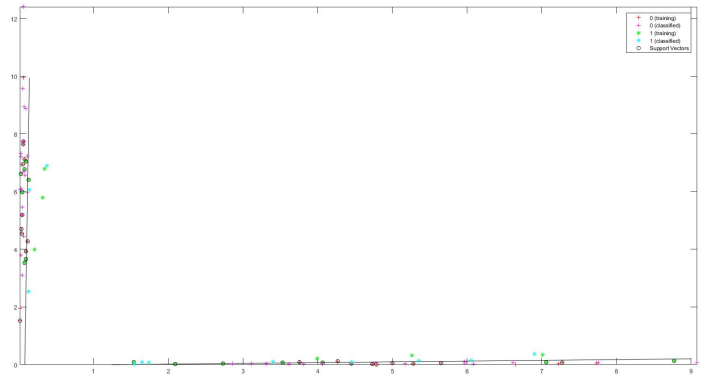
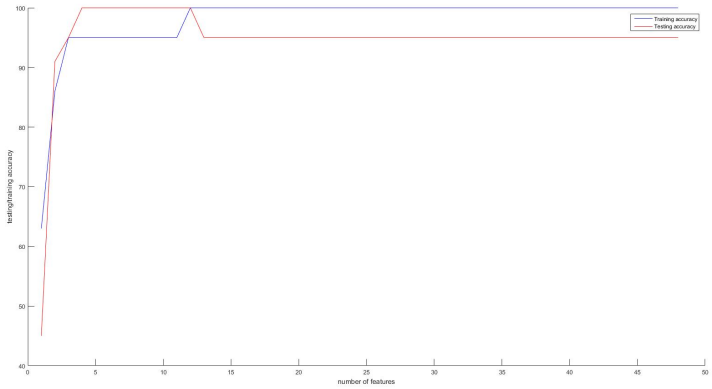


Figure 4: The data is not linearly separable in the x3-x4 plane and gives a training accuracy of 66% and a testing accuracy of 59%

The data rests in 48D and not easy to visualize. Therefore the we can view the interaction between 2 features in 2D space.

The behavior of the data as viewed in 2D in the Figures 3 and 4 suggests that the data is not exactly linearly separable in all the planes, thus increased dimensionality would help. Figure 5 shows the variations in testing and training accuracy with the number of features used.



2.2.3. SVM in MATLAB

The **svmtrain** function was used in MATLAB to train the svm model which used the following algorithm to optimize the given function and find the support vectors and the weights:-

$$c = \Sigma_i \alpha_i k(s_i, x) + b_i$$

where c is the class, k is the kernel function, b is the offset from the hyper-plane.

The **svmtrain** function uses the SMO implementation by default. The results obtained by the primal and dual soft-margin using the quadratic program are different.

2.2.4. Results using quadprog

- Training and Testing accuracy in dual form with all 3 kernels is 100%
- Training accuracy in primal form 45% testing accuracy in primal form is 54%

3. Reproducing the results

- To reproduce all the results mentioned above, just run the 'master_script.m' file.
- Make sure that the 'master_script.m' file is in the same directory as all the other .m files as there are many dependencies.
- Once you run the script, just follow the prompts in the command window to view the results.

4. Conclusion

Thus we have compared the performance of the svm algorithm with the LASSO. We have also compared the performances of the svm in primal and dual form, We have used 3 different kernel functions in the dual form. We have also shown the dependency of the prediction accuracy on the number of features selected. These are just a few implementations of the SVM and LASSO, many other implementations exits which have different trade-offs.

5. Acknowledgement

I would like to thank Prof. Farzaneh for her valuable feedback on my proposal and providing me with the resources and giving me the guidance to complete the project. A special thanks to Danny Eliahu for letting me use the data set that he collected.

6. References

- Lecture slides by Prof. Farzaneh Mirzazadeh. UCSC.
- Lectures by Prof Andrew Ng. Coursera.
- https://www.mathworks.com.
- http://statweb.stanford.edu/ tibs/lasso/simple.html.