Development of Robot Movement Recognition System and Autism Diagnosis System

EE 660 Course Project

Project Type: Design systems based on real-world data

Number of student authors: 1

Performed by: Samwoo Seong

Email:samwoose@usc.edu

Date:12/01/2019

1. Abstract

(1) Wall Following Robot Navigation System

As we face golden era of robotics, path planning and navigation in robotics become more important. I try to build a system that can recognize what movement a wall following robot made based on the data samples collected by 24 ultrasound sensors using machine learning algorithms. In this task, 3 learning algorithms are used such as Logistic Regression, Lasso Regression, and Random Forest. Since dataset has nonlinear behaviors, it turns out Random Forest model tends to work properly.

(2) Autism Diagnosis System

In medical field, autism diagnosis becomes a hot issue nowadays. I want to build a system that can diagnose autism based on personal information and answers to screening test using machine learning techniques. For the task, 3 learning methods are used such as Logistic Regression, Lasso Regression and Random Forest. It turns out the answers to screening test data play crucial rules for this classification problem.

2. Introduction

2.1. Problem Type, Statement and Goals

(1) First dataset (Wall Following Robot Navigation dataset)

The goal of developing the Wall Following Robot Movement Recognition system is that builds a system that can tell us which movements the robot made amongst slight-right-turn, move-forward, sharp-right-turn and slight-left-turn based on data from 24 ultrasound sensors arranged circular around its waist [1]. This problem can be considered as a classification problem, which classes are slight-right-turn(class1), move-forward(class2), sharp-right-turn(class3), and slight-left-turn(class4).

It is interesting because we can track down the path of the robot only based on data from ultrasound sensors not based on image from camera with the robot movement recognition system. It means it requires relatively lighter computation power compared to a system that uses camera images for the given task.

The problem can be challenging because data from the sensors have nonlinear behaviors, and data set is unbalanced (i.e., the number of samples of class2&3 are much larger than the number of samples of class1 or 2.) It leads us to some amount of preprocessing requirement.

(2) Second dataset (Autism Adult dataset)

The goal of developing the Autism Diagnosis system is that builds a system that can tell if a person has autism or not based on their personal information and answer to questions [2]. We can think of this problem as binary classification problem, which classes are person with autism (Class1), and person without autism (Class2).

It is important because diagnosing autism can be expensive if an expert is involved. With this system, we will be able to tell if someone has autism or not by only answering some questions and their personal information.

The problem can be non-trivial since data set has a lot of categorical features. It is also an unbalanced and it doesn't have a lot of data samples. Plus, it has missing values. All those characteristics of the data sets requires some amount of preprocessing.

2.2. Our Prior and Related Work (Mandatory)

"Prior and Related Work - None"

2.3. Overview of Our Approach

In this project, Logistic Regression with L2 norm regularization, Lasso Regression as known as Logistic Regression with L1 norm regularization, and Random Forest learning algorithms have been used. To decide which model will be used in training process, Cross-Validation technique is performed. Since data sets are unbalanced, training process is performed on both balanced dataset and unbalanced dataset. Then, accuracy (or error) on training/test set, confusion matrix, F-1 score are chosen as performance metrics to select final model for the systems.

3. Implementation

3.1. Data Set

(1) First dataset (Wall Following Robot Navigation dataset)

The data were collected as the SCITOS G5 robot navigates through the room following the wall in a clockwise direction, for 4 rounds, using 24 ultrasound sensors arranged circularly around its waist [1]. Data type of each input variable is real number only because it is collected from the same type of sensor but different locations.

Table 1: Features and their descriptions [1]

Name	Type	Range/Cardinality	Description
US 1	Real	Unkown	Ultrasound sensor reading at the front of the robot
US 2~12	Real	Unkown	Ultrasound sensor reading
US 13	Real	Unkown	Ultrasound sensor reading situated at the back of the robot
US 14~24	Real	Unkown	Ultrasound sensor reading

(2) Second dataset (Autism Adult dataset)

The data contains Autistic Spectrum Disorder Screening data for adult. Data types of each input variable are integer, binary, and categorical.[2]

Table 2: Features and their descriptions [2]

Attribute	Type	Range/Cardinality	Description
Age	Number	Unknown	Age in years
Gender	Category	2	Male or Female
Ethnicity	Category	Unknown	List of common ethnicities in test format
Born with jaundice	Boolean (Yes or no)	2	Whether the case was born with jaundice
Family member with PDD	Boolean (Yes or no)	2	Whether any immediate family member has a PDD
Who is completing the test	Category	More than 5	Parent, self, caregiver, medical staff, clinician, etc.
Country of residence	Category	Unknown	List of countries in test format
Used the screening app before	Boolean (Yes or no)	2	Whether the user has used a screening app
Screening Method Type	Integer (0,1,2,3)	4	The type of screening methods chosen based on age category (0=toddler, 1 = child, 2 = adolescent, 3 = adult)
Question 1~10 Answer	Binary (0,1)	2	The answer code of the questions based on the

			screening method used
Screening Score	Integer	Unknown	The final score obtained based on the scoring algorithm of the screening method used. This was computed in an automated manner

3.2. Preprocessing, Feature Extraction, Dimensionality Adjustment

(1) First dataset (Wall Following Robot Navigation dataset)

Since all features have the same unit, I decide not to standardize features. It has 4 categorical classes. Therefore, I convert these to numerical classes such as 0,1,2, and 3. I use both unbalanced dataset and balanced dataset. To create the balanced dataset, I used a function called "balancedDataSetCreater", which draws the same number of samples of dominant classes as the number of samples of minority class at random. In this case, the number of samples for each class in the balanced is 328.

Table 3: Summary of preprocessing for the first dataset

Feature Standardization?	No.
Categorical Class?	Yes, Categorical Classes are converted to numeric classes.
Unbalanced?	Yes, create a balanced dataset with self-developed function.

(2) Second dataset (Autism Adult dataset)

Since it has categorical features, I recast all of these features using one-hot encoding (sklearn.preprocessing.OneHotEncoder used). I decide not to standardize integer valued features because their range is similar to each other. For the categorical classes, I convert them to numeric classes such as

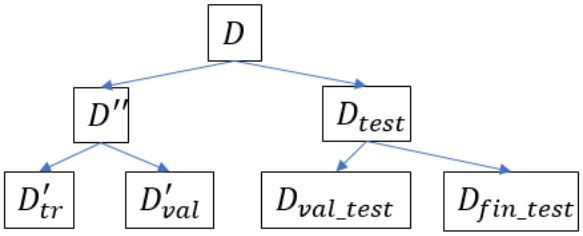
0 and 1. It is also an unbalanced dataset. Therefore, I apply the same strategy to construct a balanced dataset as I do in the first dataset.

Table 4: Summary of preprocessing for the second dataset

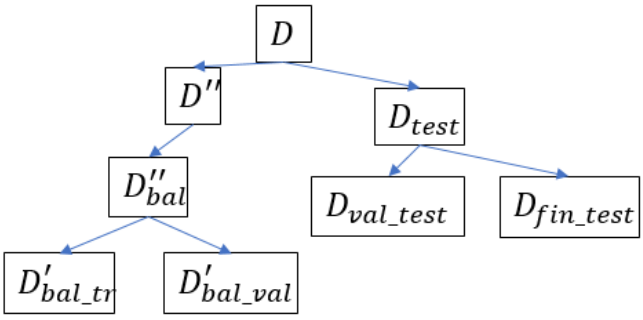
Categorical Features?	Yes, I apply one-hot-encoding technique.
Feature Standardization?	No.
Categorical Class?	Yes, it is converted to numeric classes.
Unbalanced?	Yes, create a balanced dataset with self-developed function.

3.3. Dataset Methodology

(1) Dataset Usage for Unbalanced Dataset Approach

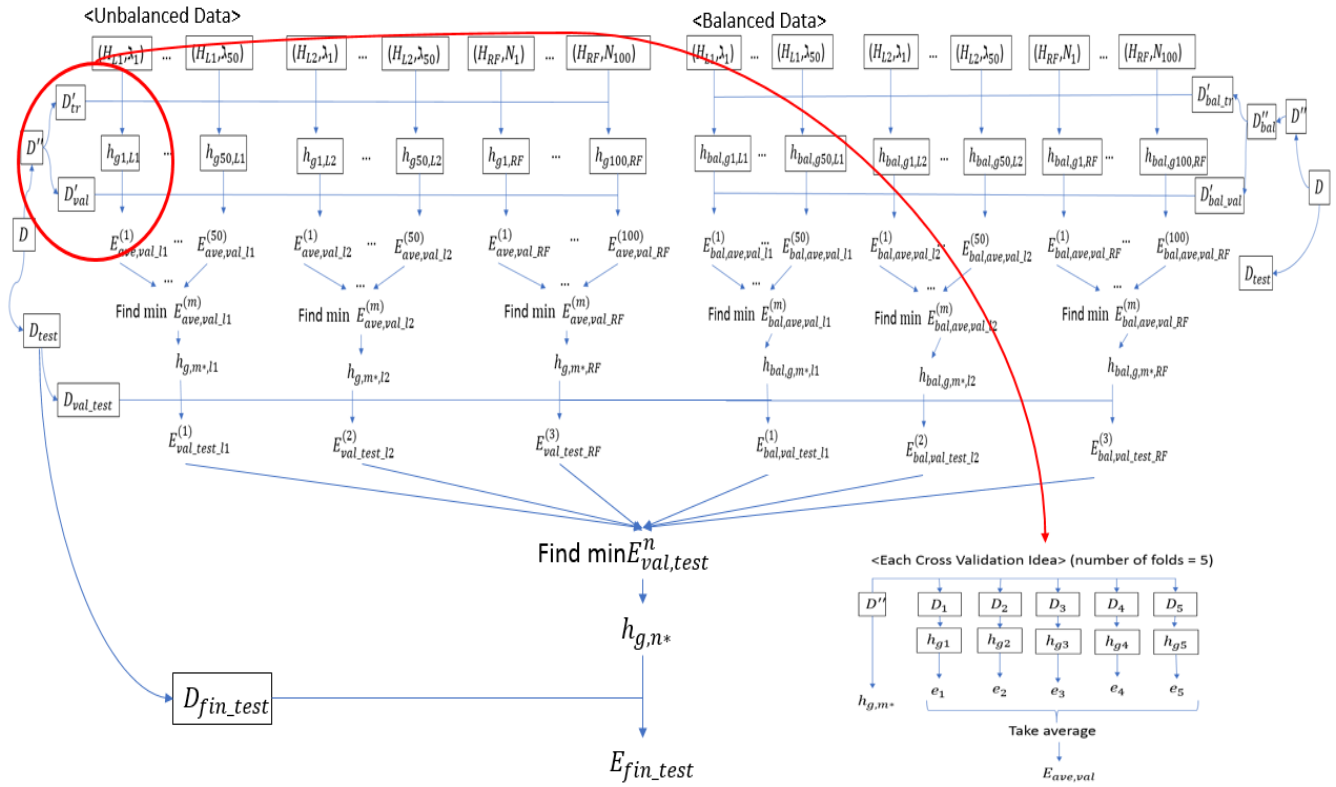


(2) Dataset Usage for Balanced Dataset Approach



(3) Flow Chat for Model Selection

(Please, zoom in this part to at least 190%)



(4) Table 5. Notation Summary

Notation	Description	Number of Samples	Purpose/Location
D	Whole Dataset	Wall: 5456 Autism: 704	After preprocessing
D''	Training dataset. It contains D'_{tr} &D'_{val}	Wall:4364 Autism:487	Used for training after CV (short for Cross Validation)/ At training session.
D'_{tr}	Training set in CV	Wall:3491 Autism:387	Used for training in CV
D'_{val}	Validation set in CV	Wall:873 Autism:100	Used for validation in CV
D''_{bal}	Balanced version of D''	Wall:1068 Autism:278	Used for training after CV (short for Cross Validation)/

			At training session.
D'_{bal_tr}	Balanced version of D'_{tr}	Wall:854 Autism:222	Used for training in CV
D'_{bal_val}	Balanced version of D'_{val}	Wall:214 Autism:56	Used for validation in CV
D_{test}	Test set. It contains D_{val_test}&D_{fin_test}	Wall: 1092 Autism:122	At final model selection & final evaluation session
D_{val_test}	Validation test set	Wall:546 Autism:61	Used for final model selection
D_{fin_test}	Final evaluation test set	Wall:546 Autism:61	Used for final evaluation of the final model
$E^{(m)}_{ave,val\_model}$	Average error on D'_{val}		Used to choose $h_{g,m*,moel}$
$h_{g,m*,model}$	Best model by CV		After CV
$E^{(m)}_{bal,ave,val\_model}$	Average error on D'_{bal_val}		Used to chose $h_{bal,g,m*,moel}$
$h_{bal,g,m*,model}$	Best model by CV		After CV
$E^{(n)}_{val\_test\_model}$	Error on D_{val_test}		Used to choose final hypothesis, $h_{g,n*}$
$h_{g,n*}$	Final hypothesis		Used to findE_{fin_test}
E_{fin_test}	Final evaluation on D_{fin_test}		To measure performance of the final model

(5) Description of Cross Validation Implementation and Decision Making

(5-1) Logistic Regression with L1/L2 norm Learning Algorithm

To determine a hyper parameter C (=1/ג) which is related to power of regularization, I used Cross Validation with D''(or D''_{bal} in balanced dataset approach). Number of folds set to 5. Amongst Logistic Regression with L1(or L2) models with different values of C, I compare average errors($E^{(m)}_{ave,val\_model}$ or $E^{(m)}_{bal,ave,val\_model}$) and choose the best C that gives the best average accuracy during cross validation. This chosen hypothesis is called $h_{g,m*,l1\ or\ l2}$ in the flow chart (or $h_{bal,g,m*,l1\ or\ l2}$ in balanced dataset approach).

(5-2) Random Forest Learning Algorithm

A hyper parameter N (=number of estimators = number of trees) is selected by Cross Validation with D''(or D''_{bal} in balanced dataset approach). The used number of folds is 5.

Among Random Forest models with different values of N, I compare average errors($E^{(m)}_{ave,val\_model}$ or $E^{(m)}_{bal,ave,val\_model}$) to choose the best N that gives the highest average accuracy during Cross Validation. The chosen hypothesis is $h_{g,m*,RF}$ in the flow chart (or $h_{bal,g,m*,RF}$ in balanced dataset approach).

After choosing 6 models through Cross Validation, I used D_{val_test} to calculate accuracy and choose a hypothesis, $h_{g,n*}$ as final hypothesis that provides the lowest error, $E^{(n)}_{val\_test\_model}$ amongst 6 models. If there is any tie in terms of accuracy, break it by choosing a model trained by balanced dataset.

Then I evaluate E_{fin_test} on D_{fin_test} to measure the performance of the final model. Since I set aside D_{fin_test}, this data set has been used only once at the end of the process and not been used during any training and decision-making process.

3.4. Training Process

(1) Logistic Regression with L1/L2 Norm Learning Algorithm

Logistic regression learning algorithm uses probability of y = certain class given samples & weight vector, $p(y|x,w)$ for classification problem, where $p(y|x,w)$ can be modeled by Bernuli density, $p(y|x,w) = \mu^{[\![y=1]\!]}(1-\mu)^{[\![y=0]\!]}$, in which $\mu = sigm(w^T x)$. Now, to find the optimized w that provide the best likelihood $p(D|w)$, we can apply MLE method. By doing it, we can find our objective function $J(w,D) = \sum_{i=1}^{N} \ln\left(1 + e^{-\tilde{y}_i w^T x_i}\right) = NLL(w)$. To avoid overfitting, we can add a regularizer such as L1 norm and L2 norm.

Then the objective function $J(w,D) = NLL(w) + \lambda||w||_1^1 (or\ \lambda||w||_2^2)$. Since $\nabla_w J(w,D) = 0$ is not algebraically solvable, we use iterative optimization technique such as Stochastic Average Gradient Descent. I choose these learning methods because it has regularizers which are a good combat strategy against overfitting.

In terms of parameter selection, I choose regularizer L1 and L2 to compare to each other. I use 'sag' and 'saga' (stochastic average gradient descent) for optimization method because it tends to calculate fast when dataset is large. For multiclass problem, I use 'ovr'(one vs rest) because it is

commonly used. The maximum iteration is 1000 because I want to iterate more than default value, 100.

We can analyze complexity of this model with lemma ($if\ H = \{h_1, \dots h_M\}, then\ d_{vc}(H) \leq log_2|H|$ [3]). For logistic regression, $|H| = \infty$ since possible values of w is any real value. Therefore, $d_{vc}(H) \leq log_2|H| = \infty$.

For Wall Following Robot Navigation dataset, it has total 5456 samples and 24 features which satisfies rule of thumb, $(3\sim10) * number\ of\ learning\ variables \ll number\ of\ samples$.

For Autism Adult dataset, it has total 704 samples and 97 features after applying one hot encode technique for categorical features. It also satisfies rule of thumb.

To avoid overfitting, I use regularizers (L1 norm and L2 norm). Since I have enough samples, I don't much worry about underfitting.

(2) Random Forest Learning Algorithm.

Random Forest Algorithm is one way to overcome cons of CART method that $var(x)$ regarding error bound can be high.

The reason why Random Forest Method can reduce value of $var(x)$ is that drawn d features in the method are d<D(number of all features) or d<<D. Therefore, amount of correlation between features gets smaller. In consequence, value of $var(x)$ becomes lower. After growing all trees, we can decide decision boundary by majority vote meaning a data point in certain region will be assigned to the class that is mostly assigned by all trees. In this project, bag size is 1 for cross validation since I use python.

I choose this method because it tends to handle well nonlinear behavior and I can use cross validation to avoid overfit in term of number of trees.

For parameter selection, I set bootstrap to 'True' to save time by not using all data points to build each tree. Also, I use cross validation to decide number of trees.

In terms of complexity of this method, we can reuse lemma ($if\ H = \{h_1, \dots h_M\}, then\ d_{vc}(H) \leq log_2|H|$ [3]), and think of each h_m as Classifier by CART. Therefore, we need to consider VC dimension of CART. To calculate this, we can use Exhaustive Search Algorithm [4].

I use Cross Validation to avoid possible overfitting by using large number of trees for this algorithm. I don't worry about underfitting for the same reason in logistic regression algorithm.

Table 6. Learning Methods, Parameters, and Hyper Parameters

Learning Method	Parameters	Hyper Parameter
Logistic Regression(L2)	-Penalty: 'l2' -Solver: 'sag' -Multiclass: 'ovr' -max_iter: 1000	Wall -$C(=\frac{1}{\lambda})$ for Unbalanced:494.17 -$C(=\frac{1}{\lambda})$ for balanced:152.64 Autism -$C(=\frac{1}{\lambda})$ for Unbalanced:193.06 -$C(=\frac{1}{\lambda})$ for balanced:308.88
Logistic Regression(L1)	-Penalty: 'l1' -Solver: 'saga' -Multiclass: 'ovr' -max_iter: 1000	Wall -$C(=\frac{1}{\lambda})$ for Unbalanced:7.19 -$C(=\frac{1}{\lambda})$ for balanced:308.88 Autism -$C(=\frac{1}{\lambda})$ for Unbalanced:193.06 -$C(=\frac{1}{\lambda})$ for balanced:9.10
Random Forest	-bootstrap: 'True'	Wall -N for Unbalanced:300 -N for balanced:370 Autism -N

		for Unbalanced:70 -*N* for balanced:100

3.5. Model Selection and Comparison of Results

I cover about model selection on subsection 3.3 (5) above.

(1) Wall Following Robot Dataset

I choose Random Forest model trained by balanced dataset as my final model for the Robot Movement Recognition System based on error and tie break rule mentioned on subsection 3.3(5)

(2) Autism Adult Dataset

I choose Random Forest model trained by balanced dataset as my final model for the Autism Diagnosis System based on error and tie break rule mentioned on subsection 3.3(5)

Table 7. Results of 6 Models for Wall Following Robot Dataset

Model	Err_ave	Err_tr	Err_val_test
L1_Unbal	0.306	0.30	0.301
L2_Unbal	0.306	0.30	0.301
RF_Unbal	0.006	0.00	0.008
L1_Balanced	0.026	0.264	0.363
L2_Balanced	0.294	0.261	0.363
RF_Balanced	0.02	0.00	0.008

Table 8. Results of 6 models for Autism Adult Dataset

Model	Err_ave	Err_tr	Err_val_test
L1_Unbal	0.066	0.071	0.033
L2_Unbal	0.046	0.048	0.017
RF_Unbal	0.00	0.00	0.00
L1_Balanced	0.124	0.143	0.066
L2_Balanced	0.103	0.106	0.033
RF_Balanced	0.00	0.00	0.00

4. Final Results and Interpretation

(1) Wall Following Robot Dataset

(1-1) Interpretation

I use majority selection that always classifies a data point to major class as baseline. Based on this baseline, the error on D_{fin_test} is $0.608 (= 1 - \left(\frac{214}{83+214+205+44}\right))$. However, the error on D_{fin_test} with final model as known as Random Forest model trained by balanced dataset is 0.013. It indicates that there is a huge improvement in term of classification accuracy. Moreover, out of sample error can be found as follows.

$$E_{out}(h_{g,n*}) \leq E_{fin,test}(h_{g,n*}) + \sqrt{\frac{1}{2N_{fin,test}} \ln\left(\frac{2M}{\delta}\right)},$$

where, $\mathrm{M} = |\mathrm{H}| = 1, N_{fin,test} = 546, \delta = 0.05$

However, since number of D_{fin_test} is rather small, upper bound of out of sample error is rather high.

It is worth looking at confusion matrix and F-1 score when dealing with unbalanced dataset. As result, values on the diagonal of confusion matrix are very close to true number of each class labels, which means this model works well even though dataset is unbalanced. As a following consequence, F-1 score is very high. Random Forest learning algorithm-based model works very well because this model can handle nonlinear behaviors quite well (i.e., dataset has nonlinear behaviors [1]) compared to logistic regression and lasso regression. I want to see any difference between logistic regression and lasso regression. However, since all data are from the same type of ultrasound sensor, the importance of feature doesn't play a significant role in this problem. That is why performance of two regression methods are somewhat similar to each other.

Table 9. Summary

Baseline Error	Final Model Error	Out of Sample Error Upper Bound
0.608	0.013	0.0711 with probability 0.95

(1-2) Table 10. Parameter Values

Random Forest	-bootstrap: 'True'	N :370

(1-3) Table 11. Confusion Matrix

		Predicted			
		Class1	Class2	Class3	Class4
Actual	Class1	83	0	0	0
	Class2	3	207	2	
	Class3	0	0	205	
	Class4	0	0	0	44

(1-4) F-1 Score

0.9846

(2) Autism Adult Dataset

(2-1) Interpretation

I use majority selection that always classifies a data point to major class as baseline. Based on this baseline, the error on D_{fin_test} is $0.344 (= 1 - \left(\frac{40}{40+21}\right))$. However, the error on D_{fin_test} with final model as known as Random Forest model trained by balanced dataset is 0.00. It indicates that there is a huge improvement in term of classification accuracy. Moreover, out of sample error can be found as follows.

$E_{out}(h_{g,n*}) \leq E_{fin,test}(h_{g,n*}) + \sqrt{\frac{1}{2N_{fin,test}} \ln\left(\frac{2M}{\delta}\right)}$,

where, $\mathrm{M} = |\mathrm{H}| = 1, N_{fin,test}$ =61, $\delta = 0.05$

However, since number of D_{fin_test} is rather small, upper bound of out of sample error is rather high.

It is worth looking at confusion matrix and F-1 score when dealing with unbalanced dataset. As it shows, values of diagonal of the matrix are the same as the number of classes which means it perfectly classifies all data points from D_{fin_test}. As a following consequence, F-1 score is 1. In this problem, all three learning methods tend to have very high accuracy. I think one of features or few features are playing a crucial rule to decide whether or not a person is autism here. Therefore, if the feature (or features) is used in training, it can perform well.

Table 12. Summary

Baseline Error	Final Model Error	Out of Sample Error Upper Bound
0.344	0.00	0.1739 with probability 0.95

(2-2) Table 13. Final Model Parameter Values

Random Forest	-bootstrap: 'True'	N :100

(2-3) Table 14. Confusion Matrix

		Predicted	
		Class1	Class2
Actual	Class1	21	0
	Class2	3	40

(2-4) F-1 Score

1.0

5. Contributions of each team member

Team Member: Samwoo Seong

He has done all work in the project.

6. Summary and conclusions

(1) Wall Following Robot Dataset

Through solving this problem, I realize the capacity that learning algorithm can handle nonlinearity can make huge difference in terms of performance of the system. However, I must pay a lot of attention not to overfit model when I use the model that has high complexity.

(2) Autism Adult Dataset

As it shows, some key features can make machine learning task easier. However, in reality, collecting data of key features is not trivial and it can cost time and other resources. Also, it would be interesting to find the key feature by removing each feature and compare the results.

Overall, finding VC dimension was quiet challenging especially for Random Forest Learning method. It would be good to spend more time to try to figure out more precise upper bound of VC dimension for Random Forest for future work.

7. References

[1] “Wall-Following Robot Navigation Data Data Set”, [Online]. Available:

https://archive.ics.uci.edu/ml/datasets/Wall-Following+Robot+Navigation+Data

[2] “Autism Screening Adult Data Set” , [Online]. Available:

https://archive.ics.uci.edu/ml/datasets/Autism+Screening+Adult

[3] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin “Learning From Data A Short Course,” Problem 2.13, 2012

[4] Ozlem Aslan, Olcay, Taner Yildiz, and Ethem Alpaydin “Calculating the VC-Dimension of Decision Trees,” [Online]. Available:

https://pdfs.semanticscholar.org/d1cc/8a12ca7f2f5e2210da5ed91ba1fe9b47a46f.pdf

8. Appendix

Appendix I.

Instruction for Running Program

To run main_wall.py and main_autism.py properly, you will need to pay attention to the path in each script file in case it causes any error.

To reproduce results on report, please run main_wall.py and main_autism.py without changing anything .

Since preprocessing contains randomness when it preprocesses raw dataset and save to different dataset, please do not run this script. Otherwise, the datasets will be changed when you run main_wall.py and main_autism.py.

Appendix I I.

Extracted zip folder:

```
├──EE_660_Final_Report_Samwoo_F19.pdf
│
├──wall
├─── main_wall.py
├─── preprocessing_wall.py
├─── training_wall.py
├─── fin_RF_clf_unbal_model_wall.sav
├─── fin_RF_clf_bal_model_wall.sav'
├─── fin_LR_cla_L2_unbal_model_wall.sav
├─── fin_LR_cla_L2_bal_model_wall.sav
├─── fin_LR_cla_L1_unbal_model_wall.sav
├─── fin_LR_cla_L1_bal_model_wall.sav
├─── sensor_readings_24-Copy1.data
├─── data_wall
│    ├── data_fin_test.csv
│    ├── data_train.csv
│    ├── data_train_bal.csv
│    ├── data_val_test.csv
│    ├── data_wall.csv
├──autism
├─── main_autism.py
├─── preprocessing_ autism.py
├─── training_ autism.py
├─── Autism.csv
├─── fin_RF_clf_unbal_model_autism.sav
├─── fin_RF_clf_bal_model_autism.sav
├─── fin_LR_cla_L2_unbal_model_autism.sav
├─── fin_LR_cla_L2_bal_model_autism.sav
├─── fin_LR_cla_L1_unbal_model_autism.sav
├─── fin_LR_cla_L1_bal_model_autism.sav
├─── data_autism
```

```
|     ├── data_autism.csv
|     ├── data_fin_test_autism.csv
|     ├── data_train_autism.csv
|     ├── data_train_bal_autism.csv
|     ├── data_val_test_autism.csv
```