A Mid-Semester Project Report

On

STUDY OF MACHINE LEARNING ALGORITHMS IN AUTISM SPECTRAL DISORDER

BY

MANVI GUPTA 2014B4A70980H

TARUNEE MEESALA 2014B4A70918H

Under the supervision of

DR.PTV PRAVEEN KUMAR

SUBMITTED IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS OF

MATH F266: STUDY PROJECT

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI(RAJASTHAN) HYDERABAD CAMPUS

(MARCH 2018)

Acknowledgements

We would like to thank Dr. PTV Praveen Kumar, for guiding us throughout this project. His guidance and meticulous suggestions helped us gain greater insight in this field.
We would also like to thank Mr. D.M.Arjun, the founder of SMILES Foundation for helping us in understanding Autism better and for providing a platform to obtain an improved data-set for the same. Finally, we would like to thank, BITS Pilani Hyderabad, for providing us this opportunity to gain exposure to the field of Autism, and help contribute to its research.

Birla Institute of Technology and Science-Pilani,
Hyderabad Campus

Certificate

This is to certify that the project report entitled ”STUDY OF MACHINE LEARNING ALGORITHMS IN AUTISM SPECTRAL DISORDER” submitted by
MANVI GUPTA 2014B4A70980H
TARUNEE MEESALA 2014B4A70918H
in partial fulfillment of the requirements of the course MATH F266, Study Project Course,embodies the work done by them under my supervision and guidance.

Date: 19/03/2018

DR.PTV PRAVEEN KUMAR

Abstract

The following report aims to discuss the current state of the art in the research area of the *Autism Spectral Disorder* in terms of *prescriptive and predictive analytics.* The public datasets available for predictive analysis from the *UCI Data Repository* were used to test the predictive power of the *Logistic Regression* algorithm for classification.

Keywords: Logistic Regression, Autism Spectral Disorder, prescriptive analytics,predictive analytics

Contents

1 Introduction

Advancements in medical technology and Science has forced the scientific community to change the way diseases/disorders are treated in society. With large computing power at our disposal, cognitive analysis of diseases is preferred over manual analysis, and is yielding great results in the medical domain. We attempt to study this change in paradigm in the treatment of the *Autism Spectral Disorder*.
Autism is a developmental disorder characterized by troubles with social interaction and communication, and by restricted and repetitive behavior.Parents usually notice signs in the first two or three years of their child's life.These signs often develop gradually, though some children with autism reach their developmental milestones at a normal pace and then worsen.
Autism may be studied from the purview of genetic sciences and also from the perspective of prescriptive therapy. We are aiming to study it from the point of view of the latter.
The disorder is caused by a combination of genetic and environmental factors, although its exact cause is still unknown. The following sections summarize the various paradigms in treating the disorder and provides a novel approach using data science, for predicting its onset in a child.

2 State of the Art: Autism Spectral Disorder

Currently, 1 percent of the world's population is diagnosed with a form of the *Autism Spectral Disorder*. 3.5 Million Americans are plagued by this disorder and they spend 236-263 Billion Dollars each year for its treatment. Sadly,there is no medicine or surgical cure for the disorder.
The disorder manifests itself in different forms in each child, and its expression is unique to every child, and time of day. This increases the complexity of finding a generic solution to treat it.

2.1 Life Journey in Autistic Children

The Life Cycle of an Autistic Child can be classified into 5 categories:

- **Early Intervention:** This phase is when the onset of the disorder is detected in children between the age of 6 months to 2 years. Behavioral changes and therapy may help the child lead a normal life.

- **Medical Advise:** For a child between the age group of 2-4 years, slight medication such as antidepressants, stimulants may be used to control erratic behavior. However, their side effects must be weighed against their benefits.
- Therapy Services: These services involve the speech and education therapy. Adapting different mechanisms to teach the child between the age group of 3-5 years after understanding their defective learning styles, might pus such children towards leading a normal life in future.
- **Technology Assisted Devices:** For children between the ages of 10-18 years, vocational training is required. Their behavioral problems may be dealt with using technology, to keep their surroundings informed, and prevent any accidents. However, their behavior cannot be repaired through cognitive study.
- **Proper Diet:** For children beyond the age of 20, no further changes in their behavior may be made. Their problems may only be curbed by making changes to their diet, such as *GFCF Diet*, which consists of gluten and caesin free meals.

Thus, the main changes may be made to the child's life style between the ages 3-8 years. These changes may be induced via creative teaching pedagogy, and better understanding of the retention power of such children. Early detection of the disorder, also improves chances of normalcy in the patient.

2.2 Blue SMILES Self-Assessment for Detection of Disorder

Currently, SMILES Foundation India has a mobile application that is capable of early detection of the disorder in children. The application generates a questionnaire to judge the IQ of the child in consideration, and the test is carried out in 4 areas:

- Social
- Academic

- Personal
- Occupational

If the average IQ scores exceeds 80 percent in all the aforementioned areas, then the child is considered *normal*.
The questionnaire is changed for different age groups, starting from new born to adult.
The scores are then sent to a local database for use in predictive analytics.
For infants, there is a special audio/visual application, that is capable of detecting facial expressions that might indicate the presence of the disorder.

Thus, right now, the main diagnosis of the disorder is based on behavior and no cause or mechanism. Hence, the quality of the questionnaire is critical in correct classification of the patient. As of now, there are 2 questionnaires available, for detection:

- Autism Diagnostic Interview- Revised (ADI-R)
- Childhood Autism Rating Scale (CARS)

Our aim is to device a better questionnaire in order to improve the quality of data to be analyzed by taking help of the experts in the field, such as SMILES Foundation. We are closely working with them to launch a application for use by the guardians of autistic patients, which will serve as a prescription for them, and at the same time be used for better prediction of the features of the disorder.

3 Logistic Regression to Classify Autistic Patients

Classification of Autistic patients is a *binary classification* problem. We have used *Logistic Regression* to model this prediction problem, as the number of available features in the model are reasonably lower than the number of observations. The dataset is not sparse, but in our further analysis, we might modify it to resemble a sparse dataset.

3.1 Mathematics behind Logistic Regression

Logistic Regression is a classification algorithm, and we have used it to model a 2 class problem.
In logistic regression, the target variable y_n is actually the class probabilities conditioned on the test instance. For a 2 class problem, it is defined as follows:

$$p(C_1|\phi) = y(\phi) = \sigma(w^T\phi) \text{ And } , p(C_2|\phi) = 1 - p(C_1|\phi) \tag{1}$$

where, σ is defined as the logistic function, $\sigma(w^T\phi) = \frac{1}{(}1 + e^{w^T\phi})$
Maximum Likelihood Estimate is used to find the optimal parameter w and the error function as follows:

Now, By the assumption of conditional independence of the training data points we have:

$$p(\mathbf{t}|\mathbf{w}) = \Pi_{n=1}^{N} y_n^{t_n}(1 - y_n)^{1-t_n} \tag{2}$$

where, $\mathbf{t} = (t_1, t_2, \ldots, t_n)^T$ and $\mathbf{w} = (w_1, w_2, \ldots, w_m)^T$
The error function is defined as the negative logarithm of the MLE:

$$E(\mathbf{w}) = -ln(p(\mathbf{t}|\mathbf{w})) = -\Sigma_{n=1}^{N} t_n ln(y_n) + (1 - t_n) ln(1 - y_n) \tag{3}$$

Taking the gradient of this error function with respect to w:

$$\nabla E(w) = \Sigma_{n=1}^{N}(y_n - t_n)\phi_n \tag{4}$$

Here we consider, ϕ to be the feature vector in the M-dimensional space.

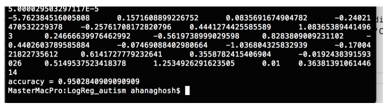
Now, we proceed to optimize $\mathbf{w}$ using *Gradient Descent* approach, as *Stochastic Gradient Descent* proves to be infeasible for such a small dataset.The following iteration is used for optimization:

$$w^{k+1} = w^k - \eta \nabla E(\mathbf{w}) \tag{5}$$

where η is the learning rate. The larger the learning rate, the slower the convergence of $\mathbf{w}$.

3.2 Features of the Dataset

Dataset Size:704 records, 21 features
The dataset comprised of 21 features, out of which 9 features were categorical. 2 of them were numeric and the rest were binary responses. The



```
5.000029503297117E-5
-5.762384516005808      0.1571608899226752      0.0835691674904782      -0.24021
470532229378    -0.25761708172820796    0.4441274425585589      1.08365389441496
3       0.24666639976462992     -0.5619738999029598     0.8283809009231102      -
0.4402603789585884      -0.07469088402980664    -1.036804325832939      -0.17004
21822735612     0.6141727779232641      0.3558782415406904      -0.0192438391593
026     0.5149537523418378      1.2534926291623505      0.01    0.36381391061446
14
accuracy = 0.9502840909090909
MasterMacPro:LogReg_autism ahanaghosh$
```

Figure 1: Weights obtained from Logistic Regression on 21 cross 704 dataset

categorical variables were converted into numeric values using a hash map, and the rest were kept as is.
The entire dataset was normalized before the application of the algorithm. Among the 21 features, 10 of them were answers to a questionnaire on Autism created by the University of Auckland, New-Zealand. This is an indicator of the the fact that the predictive accuracy of our model can be improved further, if the *Questionnaire* is improved. This is one of the areas where we are taking assistance from SMILE Autism Foundation.
On running the algorithm, and obtaining the weight vector $\mathbf{w}$, we could also tell the relative importance of the features,in the prediction of the disorder. Figure 1 shows the weights obtained on running the model, with $\eta = 0.00001, \epsilon = 0.00005$ where, ϵ is the convergence threshold chosen for $\mathbf{w}$. The first weight corresponds to the bias of the model. It is used to improve the accuracy of prediction. The remaining weights are proportional to the relative importance of the feature in predicting the disorder in the person. Thus, we see that feature 19 has the most importance. it corresponds to the age description of the patient. The next important feature is the ethnicity of the patient. These factors could help in the prediction of the *epigenetic* factors that increase the risk of the disease.
An accuracy of **95 percent** was obtained on the training dataset, which was split into 80:20 ratio for training and testing of the model.
We have treated the *missing values* in our data-set as a separate attribute.

3.3 Future Improvements in Dataset and Model

- Currently we are in collaboration with the *SMILE Autism Foundation.* Through the collaboration we hope to obtain a revised questionnaire for detection of the disease in children. We plan on launching a mobile application which offers *prescriptive analytics* to the guardian of the patient, and at the same time builds improved data for *predictive* *analytics*.

- In the logistic regression algorithm, we are yet to use regularization on the weight vector, to reduce the variance of the model, and obtain better performance.
- We are in the process of writing a routine to carry out optimized *grid search* in the hyper-parameter space, to obtain best values of $\eta, and \lambda$, where η is the learning rate, and λ is the parameter corresponding to regularization of the model.
- As of now, we are directly using *hashing* to convert the categorical features into numeric constants. However, this is a major fallacy in the model, as we are implicitly assuming a hidden precedence relation between the attribute values. We will apply the **1 of K** encoding mechanism to deal with such features.
- We are also in the process of using visualization tools to understand the rates of convergence for the error function, for faster detection of local optima.We will also experiment with other optimization techniques such as *mini-batch* descent to improve the value of $\mathbf{w}$.

A JAVA Codes for Logistic Regression

The codes in this section were written in JAVA.

A.1 Logistic Regression Code

```java
import java.util.*;
import java.util.Arrays;
import java.io.*;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.nio.Buffer;
```

```
public class logReg_aut{
//major assumption basis function as identity funcion
//can be optimised by introducing it.
//gradient descent can be modified into stochastic also.
   //can array be returned??donno ...lite ...change if not compiled.

   //input values in vector X=<xn>
   public Vector data; //input points are stored here
   public int inputSize;
   public Double[] w; //array containing parameter values
      --->packages for dot product can be used later.
   public int attr; //number of attributes in input ----> specify
      in constructor(done in this code)or check while file reading
   public Vector<Double> target; //saves target values
   public double eta; //learning rate
   public double epsilon; //condition for termination
   public int cls;///not used here since binary classification
   //public HashMap<Integer,HashMap<String,Integer>> map;//for
      changing categorical attributes to numeric: not a great idea
   public HashMap<String,Integer> map;
   public HashMap<Integer,Integer> hm;//hashmap for attribute index
      and next hash value for string above
   public HashMap<String,Integer> most_freq;///for handling missing
      values


   //test variables
   public Vector test_data; //input points are stored here
   public Vector<Double> test_label; //saves target test label
      values
   public Vector<Double> pred_label; //computed test label values
   public int test_size;


   public logReg_aut(int numattr,Double[] param,double e,double ep)
   {
   //constructor for setting number of attribute values
   //intialisation of parameter array
```

```
    data = new Vector();
    inputSize = 0;
    attr = numattr; //setting number of attributes

    w = new Double[attr+1];
    for(int i=0;i<attr+1;i++){ //copying initialisations
       w[i]=param[i];
    }
    target = new Vector<Double>();
    eta = e;
    epsilon = ep;

    map = new HashMap<String,Integer>();
    hm = new HashMap<Integer,Integer>();
    most_freq = new HashMap<String,Integer>();

    //System.out.println(attr);

    //intialize test variables here
    test_data = new Vector();
    test_label = new Vector<Double>();
    pred_label = new Vector<Double>();
    test_size = 0;


 }

 public double sigmoid(Double[] xn,Double[] w1){//calculating
     sigmoid function

    //w1T*Xn
    double dotProduct = 0;
    for(int i=0;i<attr;i++){
       dotProduct = dotProduct + w1[i]*xn[i];
    }

    //exponent^(-w1T*Xn)
    double ex = Math.exp(-1*dotProduct);
    ex = ex +1;//ex^() + 1
    //inverse of exp
    double inv = Math.pow(ex,-1);

    return inv;

}

public Double[] gradient(Double[] w1){ //gradient function

    Double[] error= new Double[attr]; //array of size 4

    for(int i=0;i<attr;i++)
        error[i] = 0.0;

    //i is index for all data-inputs
    //j is index for error[j]
    //calculate error[i] for all data inputs.

    for(int i=0;i<attr;i++) //outer loop
    {
        for(int j=0;j<inputSize-1;j++) //inner loop
        {
            Double[] xn = new Double[attr];
            Vector<Double> x = new Vector<Double>();
            x = (Vector)data.elementAt(j);
            x.toArray(xn); //convert vector to array
            double yn = target.elementAt(j);
            error[i] = error[i] + ((sigmoid(xn,w1)-yn)*xn[i]);
        }
    }

    return error;

}

public double norm(Double[] w1,Double[] w){ //assuming euclidean
     norm <-- can be changed.

    double sum =0;
    for(int i=0;i<attr;i++){
      sum = sum + ((w[i]-w1[i])*(w[i]-w1[i]));
    }

    sum = Math.sqrt(sum);
    return sum;
  }

  public void gradientDescent(){//Gradient Descendent Function
  //iteratively calculate w function

    Double[] w1 = new Double[attr]; //array for saving new
       parameter valus
    for(int i=0;i<attr;i++){   //intialising new array to zero
      w1[i] = 0.0;
    }


    Double[] w2 = new Double[attr]; //array for saving new
       parameter valus
    for(int i=0;i<attr;i++){   //intialising new array to zero
      w2[i] = 0.0;
    }

    int iteration=1;
    while(norm(w1,w)>epsilon){ //condition for termination

      System.out.println(norm(w1,w));

      Double[] g = gradient(w1);
      for(int i=0;i<attr;i++){

        w1[i] = w[i] - eta*(g[i]);

      }//end of for

      for(int i=0;i<attr;i++){   //changing to new array
      w2[i] = w1[i];
      }

      iteration++;
```

```
        //w1=w and w=w2
        for(int i=0;i<attr;i++){   //changing to new array
        w1[i] = w[i];
        }
        for(int i=0;i<attr;i++){   //changing to new array
        w[i] = w2[i];
        }



    }//end of while
}

//handling missing values babe
//public void missing_values()


//key is attr value and i is the attr number
public double data_proc(String key,int i)
{
    if(key.equals("?")) return -1.0;///should be redirected
    double num;
    if(hm.containsKey(i))
    {

        int k = hm.get(i);
        if(map.containsKey(key))
        {
            num = map.get(key);
        }else
        {
            hm.put(i,k+1);
            map.put(key,k+1);
            num = k+1;
        }

    }else
    {
      hm.put(i,0);
      map.put(key,0);
      num = 0.0;
    }

    return num;

  }


  public void ReadFileBuffReader(){

    try {
      BufferedReader reader = new BufferedReader(new FileReader(
           "Autism-Adult-Data.arff"));
      String line = reader.readLine();

      while (line != null) {

        //System.out.println(line);


        //changes
        if(!(line.startsWith("1")||line.startsWith("0"))){
          line = reader.readLine();
          continue;}

        String[] tokens=line.split(",");

        inputSize++;

        //changes
        if(inputSize == 404) break;



        int i=0;
        double num =0;
        String tar = null;
    Vector<Double> attributes = new Vector<Double>();
        //array of attr values
    attributes.addElement(1.0);
    for(String key : tokens){

       try {
                   num = Double.parseDouble(key);
           } catch (NumberFormatException e){

                 //System.out.println(i+"**"+key);

                  num = data_proc(key,i);
           }


       //num = (Double)Double.parseDouble(key);

       if(i==attr)//for the last class labelled value
       {
          tar = key;
          break;

       }
       attributes.addElement(num);
       i++;
    }
    if(i==attr){//---->4 is same as attr
       if(tar.equals("NO"))
          num = 0.0;
       else
          num = 1.0;
       target.addElement(num);
    }

    data.addElement(attributes); //inserting each data line.

    // read next line
    line = reader.readLine();
 }
```

```
        //System.out.println(items_info);
        reader.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}


//reading test data

public void test_proc(){

    for(int i=0;i<test_size;i++)
    {

        Double[] xn = new Double[attr];
        Vector<Double> x = new Vector<Double>();
        x = (Vector)test_data.elementAt(i);
        x.toArray(xn); //convert vector to array
        double val = sigmoid(xn,w);
        //System.out.println("val = "+val);
        if(val>=0.5)
            pred_label.addElement(1.0);
        else
            pred_label.addElement(0.0);
    }

}

public void ReadTestFileBuffReader(){



    try {
        BufferedReader reader = new BufferedReader(new FileReader(
                "Autism-Adult-Data.arff"));
        String line = reader.readLine();
        int z=0;
```

```
while (line != null) {

   if(!(line.startsWith("1")||line.startsWith("0"))){
      line = reader.readLine();
      continue;}

   String[] tokens=line.split(",");

   z++;
   if(z<404)  continue;

   test_size++;

   //temp contains (VEctor) all the 5 attributes
   //the first 4 attributes are real numbers.
   //1.variance 2.skewness 3.curtosis 4.entropy 5.class(0/1)

   int i=0;
   double num =0;
   String tar = null;
   Vector<Double> attributes = new Vector<Double>();
       //array of attr values
   attributes.addElement(1.0);
   for(String key : tokens){
      try {
               num = Double.parseDouble(key);
          } catch (NumberFormatException e){

               //System.out.println(i+"**"+key);

                 num = data_proc(key,i);
          }

      if(i==attr)//for the last class labelled value
      {
         tar = key;
         break;

      }
      attributes.addElement(num);
```

```
               i++;
            }
            if(i==attr){//---->4 is same as attr
               if(tar.equals("NO"))
                  num = 0.0;
               else
                  num = 1.0;
               test_label.addElement(num);
            }

            test_data.addElement(attributes); //inserting each data
                line.

            // read next line
            line = reader.readLine();
         }


         //System.out.println(items_info);
         reader.close();
      } catch (IOException e) {
         e.printStackTrace();
      }
   }//end of test


   public static void main(String[] args) {

      String line = null;
      int a=0;

      //finding number of attributes
      try {
         BufferedReader reader = new BufferedReader(new FileReader(
               "Autism-Adult-Data.arff"));
         line = reader.readLine();
         //System.out.println(line);

         while (line != null) {
          //changes
          if(!(line.startsWith("1")||line.startsWith("0")))
          {
             line = reader.readLine();
             //System.out.println(line);
             continue;
          }else
          {
             //System.out.println(line);
             String[] tokens=line.split(",");

             int i=0;

             for(String key : tokens){

                i++;
                //System.out.println(i);
             }
             a=i;

             break;
          }
       }


       //System.out.println(items_info);
       reader.close();
    } catch (IOException e) {
       e.printStackTrace();
    }


    ///------> can be changed but carefully
    //System.out.println(a);

    Double[] w0 = new Double[a];

    for(int i=0;i<a;i++)
       w0[i] =0.01;
```

```
//logReg_aut(int numattr,double[] param,double eta,double
    epsilon)
logReg_aut lg= new logReg_aut(a-1,w0,0.00001,0.00005);

lg.ReadFileBuffReader();

//System.out.println(lg.target);

//prinitng hash values

/*for (int name: lg.hm.keySet()){

      //String key =name.toString();
      String value = lg.hm.get(name).toString();
      System.out.println(name + " " + value);


}
for (String name: lg.map.keySet()){

      String key =name.toString();
      String value = lg.map.get(name).toString();
      System.out.println(key + " " + value);


}*/
lg.attr = lg.attr+1;
lg.gradientDescent();

for(int i=0;i<lg.attr;i++)
   System.out.print(lg.w[i]+"\t");
System.out.println();

// for(int i=0;i<77;i++)
// {
//    System.out.println(lg.data.elementAt(i));
//    System.out.println(lg.target.elementAt(i));
// }
```

```
        lg.attr = lg.attr-1;
        lg.ReadTestFileBuffReader();
        lg.attr = lg.attr+1;
        lg.test_proc();


        lg.attr = lg.attr-1;
        int accuracy=0;
        for(int i=0;i<lg.test_size;i++)
        {
           double x = (Double)lg.pred_label.elementAt(i);
           double y = (Double)lg.test_label.elementAt(i);
           if(x == y)
              accuracy++;
        }
        double acc = (double)accuracy/lg.test_size;
        ///System.out.println("accuracy = "+accuracy+"total_size=
            "+lg.test_size);
        System.out.println("accuracy = "+acc);


   }
}
```

References

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

[2] http://www.worldautismsociety.org/

[3] C.Bishop *Pattern Recognition and Machine Learning*