Defining a unified scale for a test battery for Parkinson patients

Computer Engineering

Yanfang Ye (860523-T201)

May. 24th, 2010

Computer Engineering

Programme	Reg number	Extent
Aritificial Intelligence	E3982D	15 ECTS
Name of student	Year-Month-Day	
YANFANG YE	2010-5-24	
Supervisor	Examiner	
Mr. Westin Jerker	Professor Mark Doupherty	
Company/Department	Supervisor at the Company/Department	
Title		
Defining a unified scale for a test battery for Parkinson patients		
Keywords		
Parkinson Disease, fuzzy controller, pharmacokinetic-pharmacodynamic model, levodopa, infusion, dose titration		

Abtract

Defining a unified scale for a test battery for Parkinson patients is always an uncertain problem in real world. The various researches have been done to study this field. In this thesis, a new model has been built with a subset of relevant predictors. We transform the target class from spiral ratings to tremor rating scal, and several kinds of techniques of attribute selection have been used to determine the subset for building the model. Stepwise regression, chisquared and gainratio attribute evaluator in Weka have been used to evaluate features. Part of the data have been used to train, but we find it is hard to make comprehensive analysis by using training set. In order to continue the analysis, we combine data from table GSR, PR and RA as merging set which is used to select attributes. Considering the distribution of data affect the selected attribute, we make the target distributes much more balance(more or less the same instances for each class). Integrating the results of every situation, we determine the selected attributes. They are first_pca_r, first_pca_drdt, answer value of question 7, question 4, question 6 and question 5, percent of question 10, speed of question 11 and dyskinesia of question 2. The night selected attriubtes have been used in the new model. Several classification methods have been tested, such as KNN, naive bays, j48 and neural network, and J48 performs the best. The new model performs well on new data.

Dalarna University
Röda vägen 3
S-781 88 Borlänge
Sweden

Tel: +46 (0)23 77 80 00
Fax: +46 (0)23 7780 80
www.du.se

Acknowledgement

I sincerely thank my supervisor Mr. Jerker Westin and sub-supervisor Dino. Thanks for their suggestions and guidance in my thesis work.

At the same time I want to acknowledge my family and my dear friends YuMei Liu and Sitao Feng who give me suggestion and totally support during the thesis period.

Dalarna University
Röda vägen 3
S-781 88 Borlänge
Sweden

Tel: +46 (0)23 77 80 00
Fax: +46 (0)23 7780 80
www.du.se

CONTENTS

LIST OF FIGURES **6**

1. INTRODUCTION **7**

1.1. GENERAL INTRODUCTION 7

1.2. OBJECTIVE 8

2. BACKGROUND **8**

2.1. DATA ACQUISITION 8

2.1.1 A Web Application for Follow-up Results *8*

2.1.2. Method for Assessing Drawing Impairment *10*

2.1.3. Defining a test score for status assessment during motor fluctuations in PD *10*

2.1.4. Colecting training and testing sets *11*

2.2. RELATED METHODS 12

2.2.1. Evaluating Methods *12*

2.2.1.A Stepwise Regression 12

2.2.1.B CfsSubsetEval + BestFirst/ExhaustiveSearch/GreedyStepwise 12

2.2.1.C ChiSquaredAttributeEval 13

2.2.1.D GainRatioAttributeEval 13

2.2.2 Classification Methods *13*

2.2.2.A J48 13

2.2.2.B NaiveBayes 13

2.2.2.C KNN 14

2.2.2.D Nural Network 14

3. METHODOLOGY **15**

3.1. DATA PRE-PROCESSING 15

3.1.1. Transformation *15*

3.1.2. File Conversion *15*

3.2. ATTRIBUTE SELECTION 15

3.2.1 Weka Build-In Methods *16*

3.2.1.A General selection 16

3.2.1.B. Optimise Operations(Increasing Instances) 17

3.2.2. Stepwise Regression *19*

3.3. CLASSIFICATION METHODS 23
3.3.1. J48(10 cross validation) *25*
3.3.2. Naïve Bays(10 cross validation) *26*
3.3.3. KNN(10 cross validation) *27*
3.4. NEURAL NETWORK 28

4.RESULTS AND ANALYSIS 30

4.1. RESULTS OF SELECTED ATTRIBUTES 30
4.2. RESULTS OF VARIOUS CLASSIFIERS 33
4.2.1. Neural Network *33*
4.2.2. J48(10 cross validation) *34*
4.2.3. NaiveBayes(10 cross validation) *35*
4.2.4. KNN(10 cross validation) *36*
4.3. RESULT OF TESTING 36

5. CONCLUSION 38

6. FUTURE WORK 39

REFERENCES 40

APPENDIX 41

List of Figures

Figure 1 System Description……10
Figure 2 Results of CfsSubsetEval……16
Figure 3 Results of Gainratio and Chisquared attribute evaluator……17
Figure 4 Results of CfsSubsetEval(merging set)……18
Figure 5 Result of After Increasing instances……18
Figure 6 Basic Variables……19
Figure 7 Methods Variables……20
Figure 8 State for (pcar,pcadrdt and q7)……20
Figure 9 Active Session Window……21
Figure 10 Result of Stepwise regression……21
Figure 11 Result of pcadrdtf……23
Figure 12 Result of q2nor……23
Figure 13 Result of different subsets……23
Figure 14 State of q10speed……24
Figure 15 Histogram of Attribute……24
Figure 16 Options of J48……25
Figure 17 Options of Naïve Bayes……26
Figure 18 Options of KNN……27
Figure 19 Options of nearest neighbour search algorithm in KNN……27
Figure 20 Options of ditance function in KNN……28
Figure 21 Stucture of NN……28
Figure 22 Results Table of selection(training set)……30
Figure 23 Results Table of selection by stepwise regression(merging set)……30
Figure 24 One of subsets……31
Figure 25 Results table of Chisquared and Gainratio……31
Figure 26 Result of 3 PC……32
Figure 27 Result of 4 PC……33
Figure 28 Result of 5 PC……33
Figure 29 Result of J48……34
Figure 30 Result of NaiveBayes……34
Figure 31 Result of KNN……35
Figure 32 Load test set……35
Figure 33 Result of testing……36

Dalarna University
Röda vägen 3
S-781 88 Borlänge
Sweden

Tel: +46 (0)23 77 80 00
Fax: +46 (0)23 7780 80
www.du.se

1. Introduction

1.1. General Introduction

Parkinson's disease (or PD) is a disorder. It is caused by the gradual loss of cells in a small part of the brain which called the substantia nigra medically. It is a neurological illness named after Dr. James Parkinson.The loss or the death of these cells produces a reduction in a vital chemical called "dopamine," which causes symptoms that may include shaking of hands, slowing down of movement, stiffness, and loss of balance. Other symptoms may include loss of facial expression, reduction in speech volume and clarity, difficulty swallowing, change in size of handwriting, dry skin, constipation, urinary difficulties, and depression. Because Parkinson's disease is a progressive disorder, these symptoms worsen with time.[1]

Although the illness most often affects older individuals, particularly those over the age of 55, Parkinson's disease may also affect people in their 30's and 40's. PD appears to be slightly more common in men than in women. Various studies have suggested that PD may be more common in certain ethnic groups or in certain regions of the world, but these results are hard to interpret in light of regional and ethnic variations in mortality, perceptions of illness, and access to health care.[1]

In order to understand a certain problem better we use data mining. Data mining course working in real world data, and extracting hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

1.2. Objective

Firstly, the target class should be formed by transforming/converting spiral ratings to Tremor Rating Scale (TRS) based on the drawing impairment and cause.

Then various techniques of attribute selection should be applied to the dataset in order to determine a subset of relevant predictors for building the model. Both types of classifiers, linear and nonlinear should be tested.

With regard to the data sets in training and testing, we should use part of the data for training and part for testing of performance.

Finally, optimize and evaluate the resulting systems in terms of the design parameters, then determine which method is most successful in the end and what the performance is?

2. Background

2.1. Data Acquisition

2.1.1 A Web Application for Follow-up Results

Before constructing and evaluating the prediction model, collecting orignal data is necessary and very important. The testing system should be able to summarize the variable series of the PD symptom during the testing periods and present them in a useful manner. The test data are collected on original sets which have been given.

Therefore, a proposed system has been given to collect original data---A web application for follow-up of results from a mobile device test battery for Parkinson's disease patients. The test battery consisting of self-assessment and motor tests which

concludes tapping and spiral drawing was developed for a hand computer with touch screen in a telemedicine setting. This test battery was used in a clinical trial by 65 patients on 9991 test occations, four tests per day during in all 362 week-long test periods, with advanced Parkinson's disease at nine clinics around Sweeden[8].

Test results from the hand unit are received and processed into scores for different dimendions of the symptom state and an overall condition of the patient during a test period. A web application presents summaries of the test results graphically(figure 1). This web application was demonstrated to fifteen study nurses who had used the test battery in the clinical trial[8].

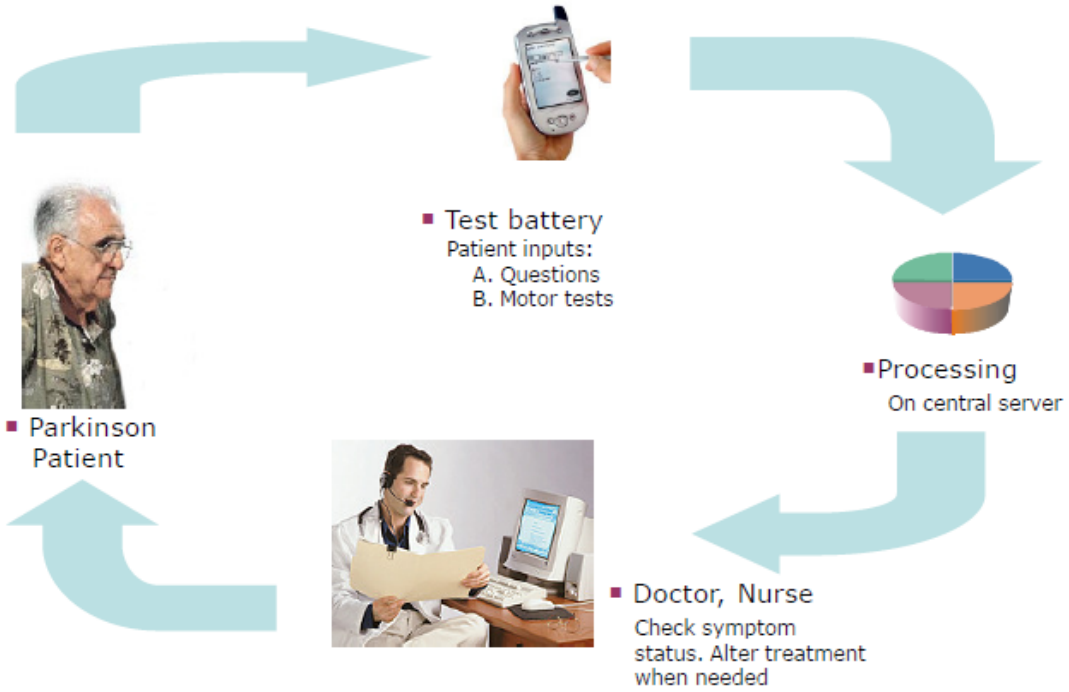


Figure 1 System Description

The web-based system that delivers decision support information to the treating clinical staff for assessing PD symptoms in their patients baed in the test battery data. Generally speaking, the responses from nurses were positive. They claimed that the test results shown in the system were consistent with their own clinical observations. They could follow complications, changes and trends within their patients.[8].

In conclusion, the system is able to summarize the various time series of moter test results and self assessments during test periods and present them in a useful manner. This could provide a basis for evaluation and adjustment of individual treatment[8].

2.1.2. Method for Assessing Drawing Impairment

One of the objective in this thesis is to transform spiral ratings to Tremor Rating Scale (TRS) based on the drawing impairment and cause. Now, a successful computer method for assessing drawing impairment in Parkinson's Disease has been proposed.

A test battery, consisting of self-assessments and motor tests(tapping and spiral drawing) was developed for a hand computer with touch screen in a telemedicine setting. The test battery was used in 8062 test occasions by 62 patients with advanced Parkinson's disease(PD). On each test occasion, 3 spirals were drawn. A computer method, using wavelt transforms and principal component analysis processed the spirals to generate a 'spiral score'. Two PD specialists(AJ, DN) rated drawing impairment in the 3 spiral drawings from 3 random test occasions per patient, using a modification of the Brain & Findley 10-category scale. Gold standard rating was defined as the mean of the two PD specialists' assessments.[8]

The results for this assessing method: the 95% confidence interval for prediction errors was below ± 2, which is similar to the differences of opinion between human raters. Spearman rank correlations were as follows: Between raters, 0.83; computer method to rater AJ, 0.86; and to rater DN, 0.85, and to mean rating, 0.89. The results show the computer method could successfully assess PD-related drawing impairments, well comparable to trained taters. We foresee potential in applying this method for remote monitoring of upper limb function in clinical practice and trials.[8]

2.1.3. Defining a test score for status assessment during motor fluctuations in PD

In his part we expect to define an overall test score, summarizing time series of self-assessed and motor test data from a test battery for patients with advanced PD.

Assessments and tests were carried out four times per day in 65 patients with advance

PD during 1-6 weekly test periods each. For most test periods, UPDRS ratings were avaliable. The information content of a test period with the test battery could be described by six dimensions, ‘off’, ‘dyskinesia’, ‘walking’, ‘satisfaction’, ‘spiral’, and ‘tapping’. Each dimension was defined as the first principal component of the level(mean), and fluctuation(standard deviation) for the questions or tests that this dimension is based on. To obtain weights for an overall test score, linear regression of the component regression of the component dimensions vs. the simutaneous UPDRS ratings was performed. To assess the internal consisternсy of the test battery, Cronbach’s Alpha for the six dimensions was calculated.[8]

In results we find weights in overall test score were(%): spirals, 41, tapping, 24, satisfied, 19, dyskinetic, 10, walking, 5.4 and off, 0.1. Internal consistency for the dimensions was 0.81. Obviously, spirals were assigned highest weight in overall score, reflecting the high weight of motor function in total UPDRS. Internal consistency among the dimensions in the overall score was strong, implying they are all measuring aspects of a common characteristic.[8]

2.1.4. Colecting training and testing sets

Base on the system for self assessments and motor tests, the original tables have been given. According to a series of summaries, the training set and testing set can be obtained.
Training data is from GSR; testing data is from RA.

Inputs features(18 in all):
Answer values of q1, q3, q4, q5, q6, q7, percent and speed of q8, q9,q11, speed of q10, Off, normal, dyskinesia of q2, firstPCA_r and firstPCA_drdt.

Output:
The test battery can be classified 2 classes: disability and cause. Disability can be classified from 0 to 10. Cause is between 1 and 3. Now, in order to predict the patients’ statement, disability and cause should be combined together.
Defining disability 0, 1, 2, 3, 4 as 1 meaning mild, diability 5, 6, 7 as 2 meaning moderate and disability 8, 9, 10 as 3 meaning severe. Defining cause 3 as 1 and cause 1, 2 as -1. The target is the product of disability and cause.

2.2. Related Methods

2.2.1. Evaluating Methods

2.2.1.A Stepwise Regression

Stepwise regression removes and adds variables to the regression model for the purpose of identifying a useful subset of the predictors. Minitab provides three commonly used procedures: standard stepwise regression (adds and removes variables), forward selection (adds variables), and backward elimination (removes variables).

In this thesis, we use standard stepwise regression to evaluate our data. We evaluate attributes one by one with mainly comparing their P-Values to estimate their significance individually.

P-value: P-value for each coefficient tests the null hypothesis that the coefficient is equal to zero (no effect). Therefore, low p-values suggest the predictor is a meaningful addition to your model. [10]

The next values are used to select model which is the best fit:
S: S is measured in the units of the response variable and represents the standard distance data values fall from the regression line, or the standard deviation of the residuals. For a given study, the better the equation predicts the response, the lower the value of S.[10]
R-sq: The proportion of the variation in the response data explained by the model. For the value of R-sq , the larger the better .[10]

2.2.1.B CfsSubsetEval + BestFirst/ExhaustiveSearch/GreedyStepwise

NAME: weka.attributeSelection.CfsSubsetEval
Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them.[9]

Subsets of features that are highly correlated with the class while having low intercorrelation are preferred.

NAME: weka.attributeSelection.BestFist

Searches the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility. Setting the number of consecutive non-improving nodes allowed controls the level of backtracking done. Best first may start with the empty set of attributes and search forward, or start with the full set of attributes and search backward, or start at any point and search in both directions (by considering all possible single attribute additions and deletions at a given point).[9]

NAME: weka.attributeSelection.ExhaustiveSearch
Performs an exhaustive search through the space of attribute subsets starting from the empty set of attrubutes. Reports the best subset found.[9]

NAME: weka.attributeSelection.GreedyStepwise
Performs a greedy forward or backward search through the space of attribute subsets. May start with no/all attributes or from an arbitrary point in the space. Stops when the addition/deletion of any remaining attributes results in a decrease in evaluation. Can also produce a ranked list of attributes by traversing the space from one side to the other and recording the order that attributes are selected.[9]

2.2.1.C ChiSquaredAttributeEval

NAME: weka.attributeSelection.ChiSquaredAttributeEval
Evaluates the worth of an attribute by computing the value of the chi-squared statistic with respect to the class.[9]

2.2.1.D GainRatioAttributeEval

NAME: weka.attributeSelection.GainRatioAttributeEval
Evaluates the worth of an attribute by measuring the gain ratio with respect to the class. GainR(Class, Attribute) = (H(Class) - H(Class | Attribute)) / H(Attribute).[9]

2.2.2 Classification Methods

2.2.2.A J48

NAME: weka.classifiers.trees.J48
Class for generating a pruned or unpruned C4.5 decision tree.[9]

2.2.2.B NaiveBayes

NAME: weka.classifiers.bayes.NaiveBayes
A Bayes classifier is a simple probabilistic classifier based on applying Bayes'

theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.[9]

2.2.2.C KNN

NAME: weka.classifiers.lazy.Ibk

The k-nearest neighbors algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. k-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor. For our case in weka 3,6,0 version we find it by the name IBK.[9]

2.2.2.D Nural Network

A *neural network* (NN). in the case of artificial neurons called *artificial neural network* (ANN) or *simulated neural network* (SNN), is an interconnected group of natural or artificial neurons that uses a mathematical or computational model for information processing based on a connectionistic approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network. In more practical terms neural networks are non-linear statistical data modeling or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.[12]

NN has several learnings, and in this thesis we use supervised learning to test data.
In supervised learning, we are given a set of example pairs and the aim is to find a function *f* in the allowed class of functions that matches the examples. In other words, we wish to infer how the mapping implied by the data and the cost function is related to the mismatch between our mapping and the data.[12]

3. Methodology

3.1. Data Pre-processing

3.1.1. Transformation

• Answervalue_7(q7)
– Multiply (-1) and add 4
• Pergoodtime features (each one of q2)
– Subtract mean, divide by standard deviation
• Tapdatasummary features (percent and speed of q8 to q11)
---percent of q10 has been deleted
– Subtract mean, divide by standard deviation
• Wavelet features(pcar, pcadrdt)
– Pcar add 18, Subtract mean, divide by standard deviation
– Pcadrdt add 4, Subtract mean, divide by standard deviation
• Answervalue q1 to q7
– Subtract mean, divide by standard deviation

3.1.2. File Conversion

– The given database was in Microsoft access DBMS tool. After collecting data, importing the selected data in EXCEL and transformation. Then save them as .CSV format. Using JAVA command to convert the .CSV format to .arff format which can be used in WEKA.

3.2. Attribute Selection

From the given tables we have collected data to prepare the training set and testing set. These data sets both have a cluster of attributes. Actually speaking, there are 19 features for the sets. Some of them may be important for our results, but others are totally redundancy features.

In this thesis we should select all the important attributes to build a new model, and ignore the invalid features and these negatived features.

3.2.1 Weka Build-In Methods

3.2.1.A General selection

There are lots of methods in WEKA to select attributes. Choosing CfsSubsetEval as attribute evaluator, and setting search method as BestFirst, ExhaustiveSearch or GreedyStepwise.
The used set is training set.

The three combinations have the same result:

```
Selected attributes: 1,2,11 : 3
                     pcar
                     pcadrdt
                     q7
```

Figure 2 Results of CfsSubsetEval

We also apply the Chisquared and Gainratio attribute evaluator to select attributes, and the results are the same:

Ranked attributes:		Ranked attributes:	
189.246	1 pcar	0.6	11 q7
138.519	2 pcadrdt	0.435	2 pcadrdt
127.593	11 q7	0.374	1 pcar
0	7 q3	0	7 q3
0	8 q4	0	8 q4
0	4 q2dys	0	4 q2dys
0	3 q1	0	3 q1
0	6 q2off	0	6 q2off
0	5 q2nor	0	5 q2nor
0	16 q10per	0	16 q10per
0	15 q9per	0	15 q9per
0	18 q11per	0	18 q11per
0	17 q11speed	0	17 q11speed
0	10 q6	0	10 q6
0	9 q5	0	9 q5
0	14 q9speed	0	14 q9speed
0	12 q8speed	0	12 q8speed
0	13 q8per	0	13 q8per

Figure 3: Results of Chisquared attribute evaluator and Gainratio

These results are very limit. Only three of features are displayed.

So far we have used several kinds of Weka build-in approaches to select attributes, but almostly all the methods gave the same result. Only there of all the 18 features have been selected, and do not give other features' significance level. These selection processes can not achieve the goal of analysing the structure of the given sets.
Generally speaking, we can not determine the selected attributes at present. There are two problems:

1. Different kinds of attributes selection have been used, but the results are the same: q7, pcar and pcadrdt. The analysis encounters difficulties. Maybe the resson is the number of instances is not great enough.
2. Prediction the patient statement of PD is a practical issue, so the given 3 selected features are not good enough to describe or summarise the symptoms of patients. We should select more useful features.

3.2.1.B. Optimise Operations(Increasing Instances)

According to the proposed problem, we should increase the number of instance. One suggested way is to combine the instances of GSR, PR with RA. The training set is

from GSR. The testing set is from RA. We collect data from PR, and do transformation at the same as what we do on training set and testing set. Then combining the three sets together called merging set.
The merging set is the new set used to select attributes.

We use the same Weka build-in methods to select features.
Choosing CfsSubsetEval as attribute evaluator, and setting search method as BestFirst, ExhaustiveSearch or GreedyStepwise.

```
Selected attributes: 1,2,4,11 : 4
                     pcar
                     pcadrdt
                     q2dys
                     q7
```

Figure 4 Result of CfsSubsetEval(merging set)

Q2dys becomes one of the selected atrributes. The selected attributes set is bigger than we use training set to select features.

In the same way, we pply the Chisquared and Gainratio attribute evaluator to select attributes with the merging se.

```
Ranked attributes:
430.5349   1 pcar
331.0103  11 q7
281.0699   2 pcadrdt
108.4945  16 q10per
106.046   18 q11per
 65.6353  13 q8per
 64.6395   9 q5
 44.5747   4 q2dys
 42.4495  10 q6
 39.9198  15 q9per
 38.3334   8 q4
 36.727    7 q3
 29.4989   3 q1
  0       17 q11speed
  0       14 q9speed
  0       12 q8speed
  0        5 q2nor
  0        6 q2off
```

```
Ranked attributes:
 0.3739   1 pcar
 0.3722   9 q5
 0.3122   2 pcadrdt
 0.2307  11 q7
 0.2296  18 q11per
 0.2225  16 q10per
 0.1258  13 q8per
 0.0909  10 q6
 0.0836   4 q2dys
 0.079    8 q4
 0.0778   3 q1
 0.0769  15 q9per
 0.0673   7 q3
 0       14 q9speed
 0        6 q2off
 0       17 q11speed
 0       12 q8speed
 0        5 q2nor
```

Figure 5: Result of Chisquared and Gainratio(merging set)

From the above results, we see the results of Chisquared evaluation and Gainratio evaluation gave lists of significance for different features, which are ordered by their significance from the most important to less important ones. These results indicate the improving operation of increasing instances is successful.

Although the lists gave information of most of attributes' significance, we can not determine which ones should be selected..

3.2.2. Stepwise Regression

In order to select important attributes, we need to analyse the sturcture of these given data sets, so we take more analysis in MINITAB.

From the above selection, we combine all the results of the Weka build-in methods, and get the result: pcar, pcadrdt and q7 are selected attributes at present. The rest features are candidates which we will check their significance in the future analysis.

Since we have chosen attributes by many methods in WEKA, the results are not as good as we expected. Stepwise regression in MINITAB is an useful appoaches to select features one by one.

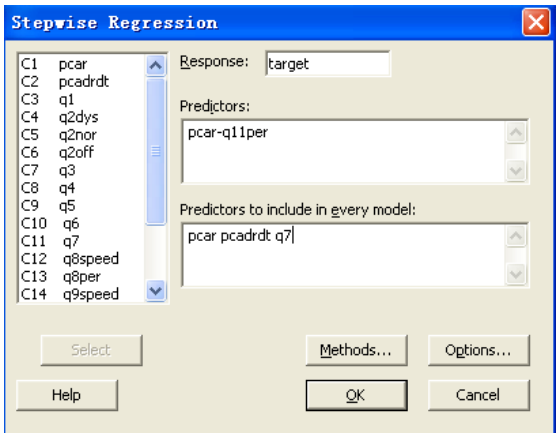


Figure 6: Basic Variables

The figure 6 shows the baisc variables before running stepwise regession. The response is target. As we want to see each feature's significance in the problem of

prediction PD patients statement,we select all the 18 features in predictors. Since we have select attributes (pcar, pcadrdt and q7) in WEKA, we select them in predictors to include in every model. Then we set the variables of methods.

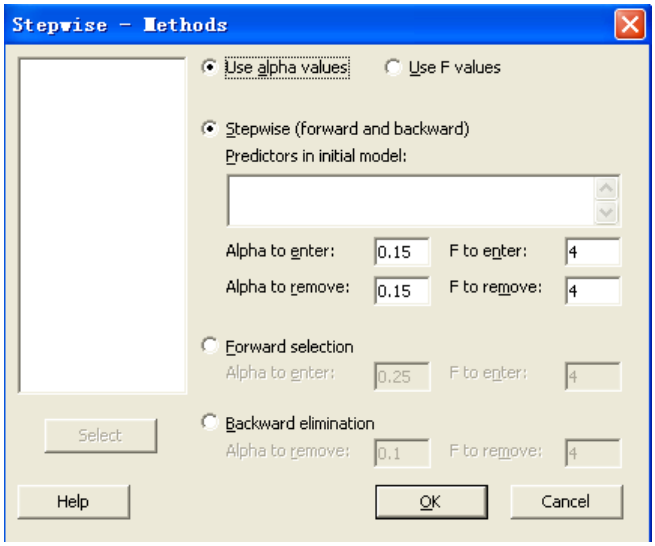


Figure 7: Methods Variables

In figure 7 we see the choosing method is forward and backward with using alpha values, and the value of alpha both to enter and to remove are setting 0.15. Actually, the 0.15 is not very important in this testing, for we want to see each feature's situation either to choose a fixed number features model.

pcar	0.11
T-Value	0.58
P-Value	0.566
pcadrdt	0.29
T-Value	1.48
P-Value	0.141
q7	1.118
T-Value	11.92
P-Value	0.000
S	1.25
R-Sq	46.00
R-Sq(adj)	45.09

Figure 8: State for (pcar, pcadrdt and q7) by training set

The testing attributes in figure 8 are the selected attributes in WEKA, the prediction percent for the future data is 46%(R-Sq). From the result we can see q7 is the most important attribute with P-Value equals to 0, and it has a great T-Value 11.92. The

second significant attribute is pcadrdt with P-Value and T-Value are 0.141 and 1.48 respectively. Pcar has bigger P-Values and smaller T-Values than pcadrdt and q7. From the the result we also know there are no other features' P-Value less than 0.15(because the alpha sets as 0.15 to enter in the suggested model).

The above result is based on training set which we got only three attributes by Weka build-in approaches, and they are q7, pcar and pcadrdt. After that we apply some optimize operation by using merging set, and we get more usefull information, so we will also use merging set to select attribute in MINITAB by stepwise regression.

Firstly, importing the merging set in the current worksheet.
Then, activating the commands session.

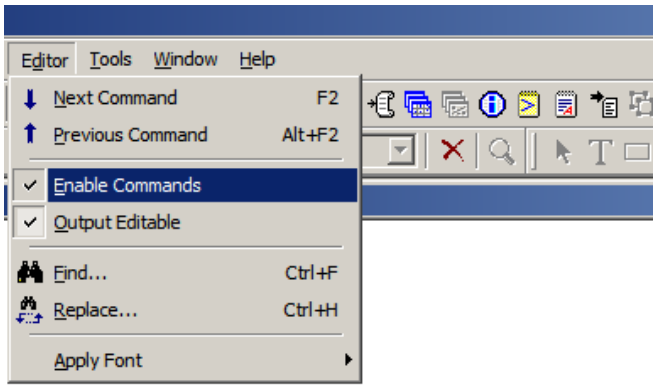


Figure 9 Active Session Window

Finally, setting the same method and the same value of alpha to entere and remove as using training set, but we do not set any predictors to indude in every model.

After application we get the result:

pcadrdt	0,610	0,616	0,526	0,507	0,453	0,467
T-Value	7,33	7,75	6,82	6,67	5,57	5,71
P-Value	0,000	0,000	0,000	0,000	0,000	0,000
q3		0,504	0,550	0,329	0,366	0,319
T-Value		6,34	7,21	3,51	3,83	3,20
P-Value		0,000	0,000	0,000	0,000	0,002
q4			-0,492	-0,545	-0,546	-0,548
T-Value			-6,34	-7,04	-7,08	-7,12
P-Value			0,000	0,000	0,000	0,000
q6				0,373	0,394	0,365
T-Value				3,92	4,13	3,75
P-Value				0,000	0,000	0,000
q5					-0,156	-0,152
T-Value					-1,85	-1,80
P-Value					0,066	0,073
q1						0,135
T-Value						1,57
P-Value						0,118
S	1,68	1,60	1,53	1,51	1,50	1,50
R-Sq	11,62	19,55	26,78	29,45	30,04	30,46
R-Sq(adj)	11,41	19,15	26,24	28,76	29,17	29,43

pcadrdt	0,458
T-Value	5,61
P-Value	0,000
q3	0,31
T-Value	3,09
P-Value	0,002
q4	-0,545
T-Value	-7,08
P-Value	0,000
q6	0,384
T-Value	3,93
P-Value	0,000
q5	-0,151
T-Value	-1,79
P-Value	0,074
q1	0,154
T-Value	1,77
P-Value	0,077
q11speed	-0,132
T-Value	-1,74
P-Value	0,082
S	1,50
R-Sq	30,98
R-Sq(adj)	29,78

Figure 10 Result of Stepwise Regession

This is the first time output. From the above results, we see all the p-value of the given features in each model are less than 0.15, and with the increasing number of attributes the R-Sq increases at the same time. The most right selected attributes set gives the highest pecent for the future. This set of selected attributes includes 7 attributes, but do not include q7 and pcar which we have thought they should be included in the selected attributes set, so we will analyse these uncertain attributes step by step by comparing the value of both R-square and P-value for each feature.

With that first result, we enter the rest features in the first selected attributes set.
We write in the session window:
----enter q2dys q2off q2nor q7 q8per q8speed q9per q9speed q10per q11per
----yes

Then we can get 12 subsets of selected attributes. Each subset composes by different attributes. The subset with all 18 attributes and the subseet with 7 attributes are the most members and the least members subsets. The result will give the process of decreasing 18 attributes to 7 attributes by deleting the less significance attributes. At the same time, the result also show each attribute's details, such as P-value and T-value.

Step	22	23	24	25	26	27
Constant	0,3504	0,3504	0,3504	0,3504	0,3504	0,3504
pcadrdt	0,32	0,31	0,32	0,31	0,32	0,32
T-Value	1,79	1,79	1,79	1,78	1,82	1,84
P-Value	0,075	0,075	0,074	0,076	0,069	0,067

Figure 11 Result of pcadrdt

In the above picture, the different step stands for different subset of selected attributes. We take attribute "pcadrdt" as an example. The P-value and T-value of "pcadrdt" are different in different subsets, but generally speaking, they are general the same in the subsets, but there are some attributes do not show this property.

q2nor	0,57	0,56	0,55	0,12	0,12	0,13
T-Value	0,45	0,45	0,44	0,92	0,90	0,93
P-Value	0,653	0,656	0,662	0,360	0,371	0,351

Figure 12 Result of q2nor

From the above result we see the P-value and T-value of "q2nor" show a great fluctuate in different subsets. This attribute's significance is unstable. Its significance also depends on some other attributes. When we depend the selected attributes, if the subset includes the attributes which affect the q2nor's significance, then we can choose q2nor, otherwise not.

S	1,51	1,51	1,50	1,50	1,50	1,50
R-Sq	31,72	31,72	31,70	31,68	31,66	31,62
R-Sq(adj)	28,58	28,76	28,93	29,09	29,24	29,38

Figure 13 Results of different subsets

This figure shows the quality for different subsets of selected attributes. We concentrate on the value of R-square. It indicates the prdiction quality of current attributes subset. At the same time we consider the R-sq(adj) which is a modified r square.

3.3. Classification Methods

As we have take various of approaches to analyse the data. Now we apply different classifier on the data.

We use training data to validate the performances of diverse classifiers.

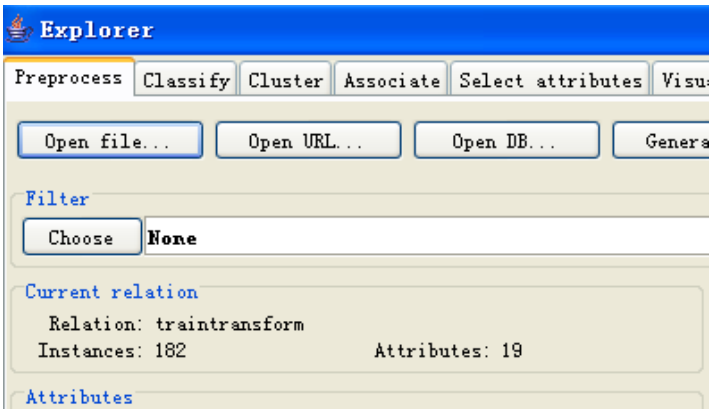


Figure 14: Load training set

Figure 14 shows the situation after loading training set in WEKA.

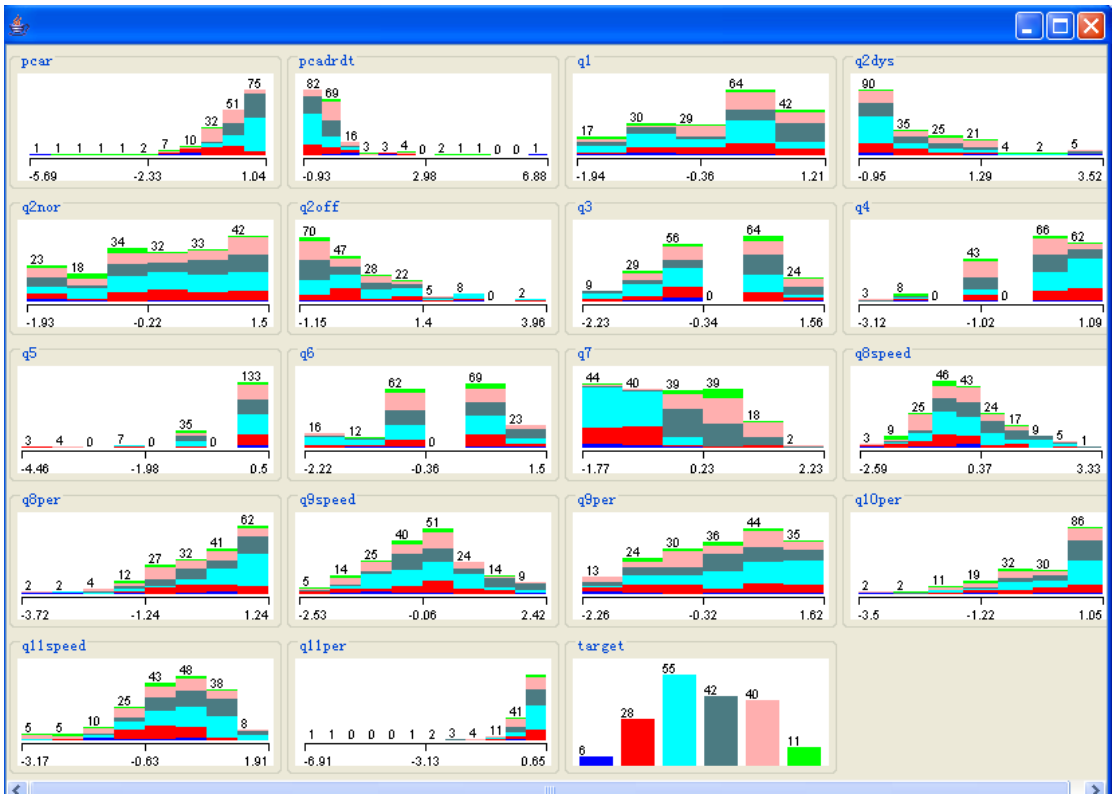


Figure 36: Histogram of attributes

After loading training set in WEKA, we can visualize histogram of each attribute. In this figure we can see details of each attribute.

For every classifier we choose 10-fold cross validation. In *10*-fold cross-validation, the original sample is randomly partitioned into *10* subsamples. Of the *10* subsamples, a single subsample is retained as the validation data for testing the model, and the remaining *9* subsamples are used as training data. The cross-validation process is then repeated *10* times (the folds), with each of the *10* subsamples used exactly once as the validation data. The *10* results from the folds then can be averaged (or otherwise combined) to produce a single estimation. The advantage of this method over repeated random subsampling is that all observations are used for both training and validation, and each observation is used for validation exactly once.

3.3.1. J48(10 cross validation)

J48 is used to classify for generating a pruned or unpruned C4.5 decision tree. It has several options, such as binarySplits -- Whether to use binary splits on nominal attributes when building the trees, confidenceFactor -- The confidence factor used for pruning (smaller values incur more pruning), numFolds -- Determines the amount of data used for reduced-error pruning. One fold is used for pruning, the rest for growing the tree and so on.[9]

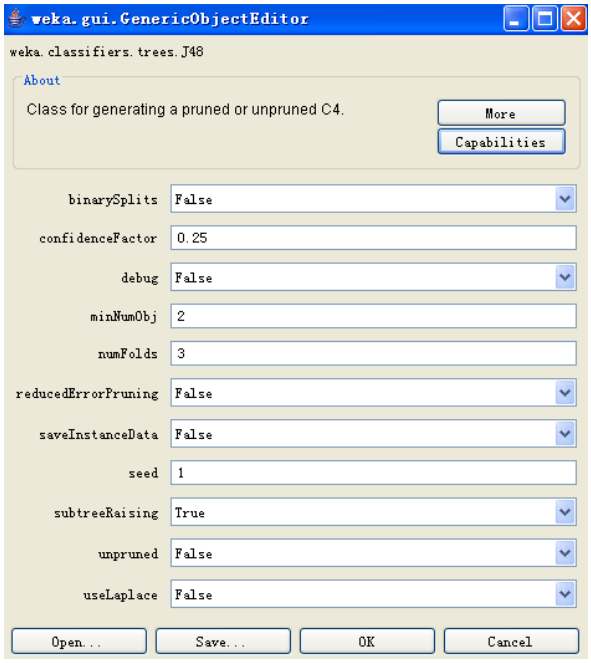


Figure 16 Options of J48

Figure 16 shows the values and types for different options of J48 in the training process.

For classifier J48, one of the most improtant criterion is the number of decision tree and number of leaves. The bigger of the number of decision tree and leaves, the worse the classifier is. In other words, we should try other classifier to test the current subset of seleected attributes, or we should select the subset again.

Generally speaking, the more attributes in the subset, the greater the number of decision tree and leaves are. However, when the selected attributes are more or less the same, the number of decision tree and leaves depends on the properties of the selected features. That is why we spend so much energy to analyse these attributes.

3.3.2. Naïve Bays(10 cross validation)

A Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.[9]

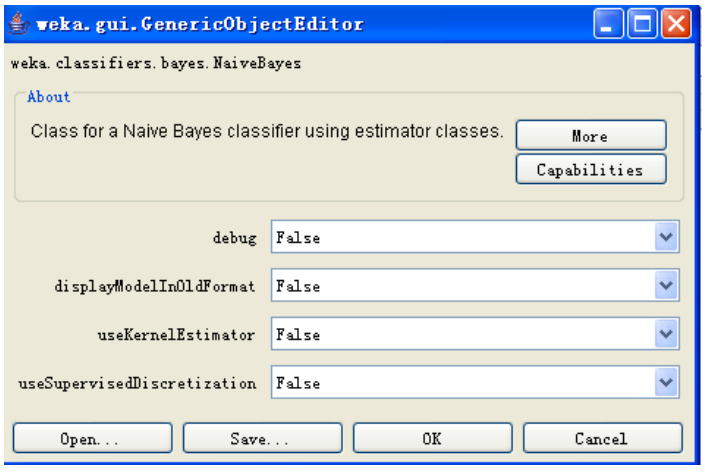


Figure 17 Options of Naïve Bayes

The above figure gives the different options in Naïve Bayes, and this situation in the above picture will be used in our testing.

3.3.3. KNN(10 cross validation)

The k-nearest neighbors algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. k-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor. For our case in weka 3,6,0 version we find it by the name IBK.[9]

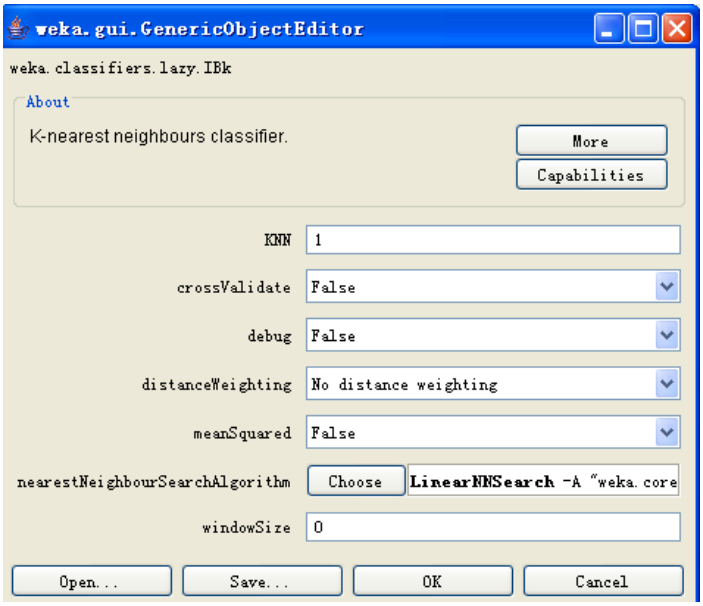


Figure 18 Options of KNN

We use the options in KNN mainly by default, and the option of nearestNeighbourSearchAlgorithm shows in the next.

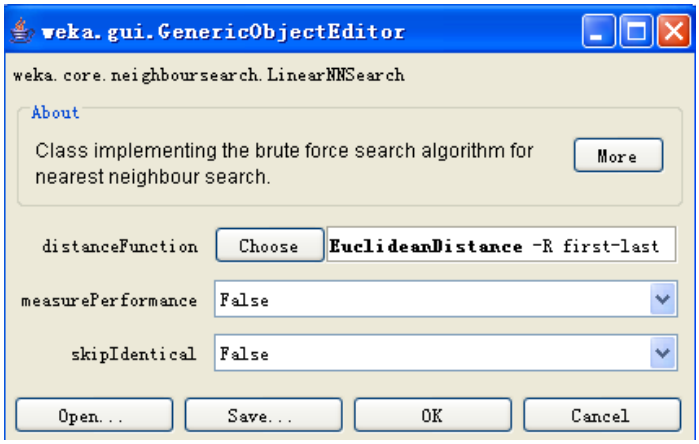


Figure 19 Options of nearest neighbor search Algorithm in KNN

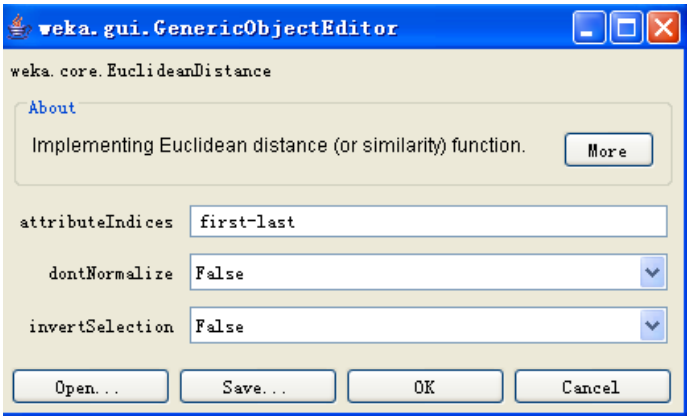


Figure 20 Options of distance function in KNN

3.4. Neural Network

Neural Network is one of the considered training methods. The using codes are written by hands which attaches on appendix. In this program process, we use principal components to reduce the dimentions of inputs.

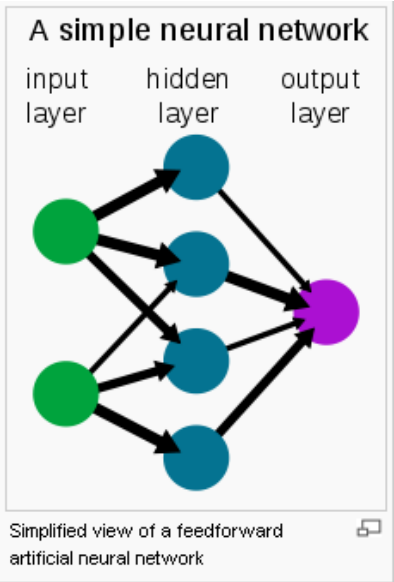


Figure 21 Structure of NN

In the neural network, we use principle components to reduce the data set firstly. Principle Components Analysis(PCA) involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The principle components are the eigenvectors with highest eigenvalues, and once the eigenvalues are computed, the principle components are ordered from the highest to the lowest. This gives the components in order of significance. The algorithm is listed below:

1. Get data and subtract mean.
2. Calculate the covariance matrix.
3. Calculate eigenvalues and eigenvectors.
4. Choosing components and forming a feature vector.
5. Deriving the new data set. FinalData=$(\text{FeatureVector})^{T}$ *$(\text{MeanAdjustedData})^{T}$.

In the training process, we use the the build-in function to build the network. We set the size of hidden layer as 4, and the size of output layer depends on the target. In this thesis, the size of output layer is 6.

For the results of the neural network, we plot the target outputs and the actual outputs in the same figure, so we can find it is easy to check whether we have classified our data correctly or not.

4.Results and Analysis

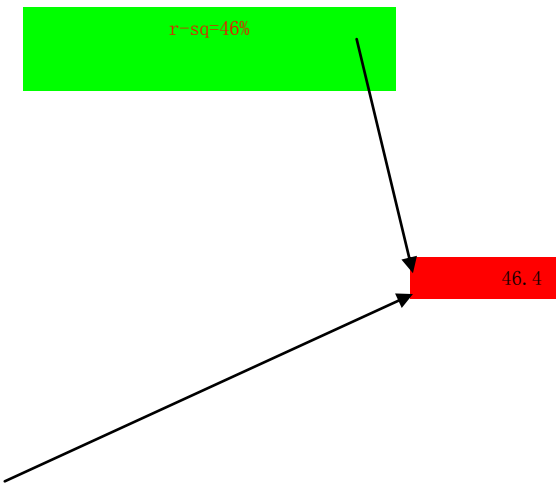
In this part we give the results of selecting attributes and the results by different classifiers.

4.1. Results of selected attributes

In the process of selected attributes, we use various validations.

Firstly, we use the training set applied in Weka by Weka build-in methods and stepwise regression.The next table summarys information from chisquared, gainratio and P-value tested by stepwise regression.

	chisquared	gainratio	stepwise(P-value)
pcar	189.246	0.374	0.566
pcadrdt	138.519	0.435	0.141
q7	127.593	0.6	0
q1	0	0	0.989
q2off	0	0	0.982
q2nor	0	0	0.919
q2dys	0	0	0.903
q3	0	0	0.845
q4	0	0	0.755
q5	0	0	0.74
q6	0	0	0.44
q8speed	0	0	0.944
q8per	0	0	0.255



q9speed	0	0	0.888
q9per	0	0	0.584
q10speed	0	0	0.768
q11per	0	0	0.689
q11speed	0	0	0.648

Figure 22 Results table of selection(training set)

From the above figure we see the results given by Weka build-in method show very limit message about the attribute. The P-value for most of the attributes are great which means they are not important. We also get the value of r-square for two subsets. Generally speaking, using training set to select attributes is not a good idea that is why we take improving operation.

pcadrdt	q3	q4	q6	q5	q1	q11speed	pcar	q7	q2dys	q2nor	q2off	q8speed	q8per	q9speed	q9per	q10per	q11per	R-sq
0.075	0.009	0	0	0.116	0.159	0.317	0.421	0.63	0.603	0.653	0.724	0.817	0.695	0.585	0.891	0.485	0.63	31.72
0.075	0.008	0	0	0.116	0.16	0.317	0.409	0.633	0.606	0.656	0.727	0.804	0.713	0.592		0.479	0.634	31.72
0.074	0.008	0	0	0.113	0.147	0.33	0.417	0.628	0.611	0.662	0.734		0.7	0.423		0.489	0.636	31.7
0.076	0.008	0	0	0.109	0.142	0.327	0.406	0.651	0.214	0.36			0.696	0.406		0.478	0.638	31.68
0.069	0.007	0	0	0.106	0.146	0.34	0.437	0.65	0.219	0.371				0.424		0.509	0.591	31.66
0.067	0.007	0	0	0.114	0.14	0.315	0.449		0.196	0.351				0.424		0.532	0.625	31.62
0.076	0.006	0	0	0.102	0.138	0.341	0.486		0.185	0.367				0.407		0.602		31.58
0.072	0.007	0	0	0.104	0.138	0.245	0.417		0.179	0.353				0.437				31.53
0.053	0.006	0	0	0.081	0.103	0.343	0.468		0.173	0.343								31.43
0	0.008	0	0	0.067	0.099	0.146			0.149	0.34								31.34
0	0.002	0	0	0.089	0.074	0.171			0.277									31.18
0	0.002	0	0	0.074	0.077	0.082												30.98

Figure 23 Results table of selection by stepwise regression(merging set)

Figure 23 shows results of all the 18 attributes by stepwise regression. Since we set the value of alpha to remove and enter as 0.15, the first six attributes are always included in the new subsets. From the above results, we see the first line in the table involves 18 attributes, and one of these attributes was deleted in the second line which is another subset. This operation continues until P-value of the rest attributes in the subset are all less than 0.15. In the results which attriubte will be deleted depends on the P-value of the attributes in the current subset. The attribute having the highest P-value will be deleted. The last column gives the percent of prediction for each subset. The subset of selected attributes should include significant attributes, so integrating the number of atrributes and r-square for each subset, we decide to choose

this subset:

pcadrdt	q3	q4	q6	q5	q1	q11speed	pcar	q7	q2dys	q2nor	q2off	q8speed	q8per	q9speed	q9per	q10per	q11per	R-sq
0.076	0.006	0	0	0.102	0.138	0.341	0.486		0.185	0.367				0.407		0.602		31.58

Figure 24 One of subsets

The above result is from stepwise regression, then we combine situations in Weka build-in methods.

	Chisquared	Gainratio
pcar	430.5349	0.3739
pcardrdt	281.0699	0.3122
q7	331.0103	0.2307
q11per	106.046	0.2296
q10per	108.4945	0.2225
q8per	65.6353	0.1258
q5	64.6395	0.3722
q2dys	44.57447	0.0836
q6	42.4495	0.0909
q9per	39.9198	0.0769
q4	38.3334	0.079
q3	36.727	0.0673
q1	29.4989	0.0778
q11speed	0	0
q2nor	0	0
q2off	0	0
q8speed	0	0
q9speed	0	0

Figure 25 Result table of Chisquared and Gainratio

Figure 25 lists the rank for chisquare and gainratio. Incorporating the figure and figure, the subset of selected attributes includes: pcar, pcadrdt, q7, q11speed, q10per, q6, q5, q4 and q2dys.

4.2. Results of various Classifiers

4.2.1. Neural Network

The codes is a function applying in Matlab. Calling the function and giving the conditional variables: input data set and target(training set), number of principle components.

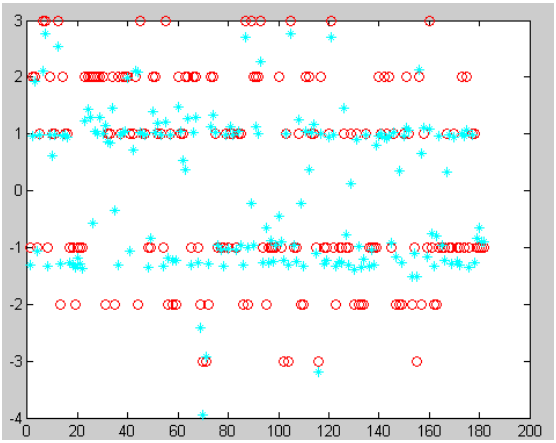


Figure 26: Result of 3 PC

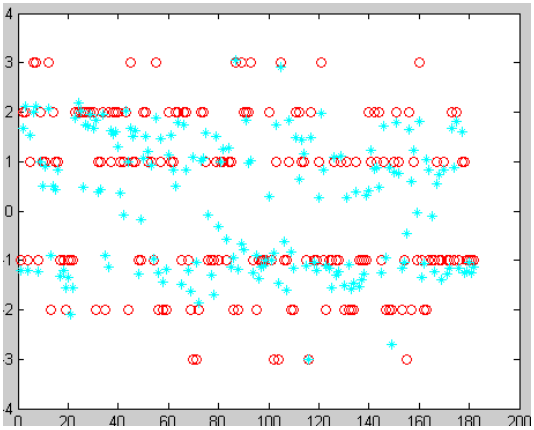


Figure 27: Result of 4 PC

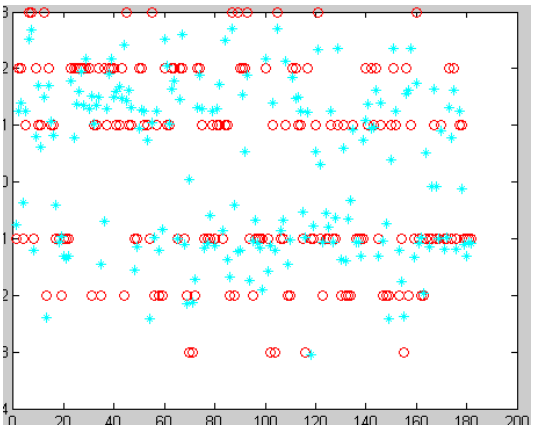


Figure 28: Result of 5 PC

From figure 26 to figure 28, we can compare the results of different principal components. It is not hard to find 3 principal components has a better result, but generally speaking Neural Network is not a good method to apply on this case. (Because we use internal training function, and the initial weights and bias of each neuron are chenged every time, the network is unstable).

4.2.2. J48(10 cross validation)

For Weka build-in methods, we use the selceted attributes(pcar, pcadrdt, q7, q11speed, q10per, q6, q5, q4 and q2dys).
Firstly we use J48 method to train the set:

```
Number of Leaves  :     19

Size of the tree :      37

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         119               65.3846 %
Incorrectly Classified Instances        63               34.6154 %
```

```
=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
  2  2  1  0  1  0 |  a = -3
  2 12  8  1  5  0 |  b = -2
  0  6 45  4  0  0 |  c = -1
  0  1  5 31  5  0 |  d = 1
  1  3  0 12 23  1 |  e = 2
  1  0  0  0  4  6 |  f = 3
```

Figure 29: Result of J48

The above figure shows the results of J48. The first one gives the number of leaves and the size of the pruned J48 trees. The following picture shows the precent of correctly classified and uncorrectly classified. According to the values, we can estimate the current classifier is good or not. The end one is the confusion matrix, which gives the details of classification by the current classifer. Although the confusion matrix, we can see which instances are correctly classified, and which ones are not, and how many of them are uncorrectly classified.

4.2.3. NaiveBayes(10 cross validation)

The next figure gives the result by Naïve Bayes.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         106               58.2418 %
Incorrectly Classified Instances        76               41.7582 %

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
  0  4  0  0  1  1 |  a = -3
  3  9 12  1  3  0 |  b = -2
  0  7 40  8  0  0 |  c = -1
  0  0  4 31  7  0 |  d = 1
  1  4  1 12 20  2 |  e = 2
  1  0  0  0  4  6 |  f = 3
```

Figure 30: Result of NaiveBayes

4.2.4. KNN(10 cross validation)

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          77               42.3077 %
Incorrectly Classified Instances       105               57.6923 %
               === Confusion Matrix ===

   a  b  c  d  e  f   <-- classified as
   0  2  1  0  2  1 |  a = -3
   0  7 14  3  4  0 |  b = -2
   0  9 40  5  1  0 |  c = -1
   0  3  7 15 17  0 |  d = 1
   0  3  1 20 12  4 |  e = 2
   0  1  0  2  5  3 |  f = 3
```

Figure 31: Result of KNN

We apply the training set by J48, NaiveBayes and KNN. From the results of them we find J48 obtain the highest correctly classified percent with 65%, and the other two training methods get lower percents. Comparing their confusion matrixes we find NaiveBayes and KNN can't classify -2 and -3 well. Actually, from the results of the above confusion matrixes, we can see there are instances belonging to -2 incorrectly classified to -1, and many instances belonging to -1 have been classified to -1. The same situation for class 1 and class 2. J48 is the chosen model with our selected attributes.

4.3. Result of Testing

We can use our build model to test new data. Supplying testing set as outside set, and uploading it:

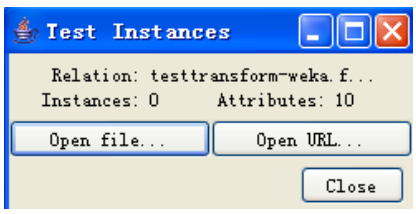


Figure 32: load test set

Now we can use J48 to test new data.

```
Number of Leaves  :     19

Size of the tree :      37
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          40               66.6667 %
Incorrectly Classified Instances        20               33.3333 %
=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
  0  0  0  0  0  0 |  a = -3
  0  3  3  2  5  0 |  b = -2
  0  0 15  1  1  0 |  c = -1
  0  0  4  5  1  0 |  d = 1
  0  0  1  1 13  0 |  e = 2
  0  0  0  0  1  4 |  f = 3
```

Figure 33: Result of testing

The above figure shows the results of testing new data set by the new model. From the confusion matrix we find the main problem is on the class -2. This is the problem what will spend much more energy on

5. CONCLUSION

In this thesis, firstly we apply the data in WEKA to select attributes. Although several attributes evaluators have been used to select features, the significance of mostly attributes are uncertain.

In order to gain a better model, improving operations have performed. Increasing the number of instances by combining the data from GSR, PR and RA tables. We get more useful message after this performances, and then we use another statistic tool-MINITAB. Stepwise regression in MINITAB gives P-Value and T-Value for each feature, so we get more understanding about these inputs.

According to stepwise regression, we have get more details of each attribues. Then we combine the analysis by setpwiese regression and Weka build-in methods to determine the selected attributes. They are: pcar, pcadrdt, q7, q6, q5, q4, q2dys, q10per and q11speed.

Classifier methods of J48, NaiveBayes and KNN are used to train the network. Comparing the results of them, J48 is the best with the highest pecent of correctly classified.

We test the new model by new testing data, and the result is 67% correctly classified percent, mostly of the instances have been correctly classified which indicates the built model works. However, there are problems in the new model which need much more researches.

6. Future Work

We has discussed the thesis purpose and how to solve problems to achieve the goal in discussion section. In the process of solving the series problem and analysing the results we find there are there aspects should take further consideration.

The first fact which we should consider more is the distribution of data. In other words, the data of traing set and testing set should fit to the real world. How many people are dyskinesia, how many ones are tremor and none in the same group of Parkinson disease. The more accurate distribution of traing set the more usefull of our new model. Because training set affect the selected attributes directly.

In this thesis we collect training set and testing set based on the given original tables. At the beginning we use training set to select attributes, but we find we can not get the expected result. So as to solve the problem we have two improving operations. One of them is combining GSR, PR and RA. We find the distribution of PR is not good. The percent of class 3 and -3 take up a great part and this situation in Parkinson disease is not true in real world.

The second is the number of instance. In order to summary the properties of the given data in data mining we always need a great of instance to analyse. The more instances we use, the more accurate conclusion we will get.

This thesis has proved this point. We use combining set which has more instance than training set to select attributes, and we get more useful information than we use training set.

The final aspect is to take more analysis about the given data. The more analysis methods have been used, the more side of the attribute will be display.

References

[1] Parkinson Disease information: http://www.parkinson-study-group.org/What%20is%20Parkinson%20Disease.html

[2] Wikipedia http://en.wikipedia.org/wiki/Weka_(machine_learning)

[3] MATLAB http://en.wikipedia.org/wiki/MATLAB

[4] MINITAB http://en.wikipedia.org/wiki/Minitab

[5] J Westin, M Dougherty, D Nyholm, T Groth. A home environment test battery for status assessment in patients with motor fluctuations. Comput Methods Programs Biomed. In press 2009; doi:10.1016/j.cmpb.2009.08.001.

[6] J Westin, M Memedi, S Ghiamati, D Nyholm, M Dougherty, T Groth. Defining a test score for status assessment during motor fluctuations in Parkinson's disease, Movement Disorders 2009;24(Suppl 1):389.

[7] Xiaowen Jiang. Master Thesis Report. E3840D, 2009

[8] Mevludin Memedi. Thesis Report.

[9] Specification of Software WEKA 3.5.8

[10] Specification of Software MINITAB 15

[11] I. Witten & E. Frank (2005) Data Mining: practical machine learning tools and techniques with Java Implementations.

[12] Wikipedia http://en.wikipedia.org/wiki/Neural_network

Appendix

Codes for Neural Network

```
function pca(t,n,target)%t is the original data set.and n is the number of features you
need
%UNTITLED2 Summary of this function goes here
%     Detailed explanation goes here
temp = t;
result = cov(temp);%calculate the covariance matrix of the temp' matrix
[vect value] = eig(result);% calculate the eignvector and the eignvalue
EigArray = eig(result);%calculate the eignvalue
eign=EigArray';
e_length = length(eign);%compute the length of the eignvalue vector

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%the following loops are to sort the eignvector and record the subscript
%corresponding to each eignvalue
for j=1:1:e_length
      val=-inf;
      for i=1:1:e_length
            if val<eign(i)
                  val=eign(i);
                  e_sub(j)=i;
                  i;
            end
      end
      eign(e_sub(j))=-inf;
end

featureVector=[vect(:,e_sub(1))];
% disp('feature is');
% featureVector
for m=2:1:n
      featureVector=[featureVector,vect(:,e_sub(m))];
```

```
end
%       disp('feature is');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%calculate the meanVector of the temp'
meanVector = temp;
for i=1:1:6
    mean = 1/182*sum(meanVector(:,i));
    for j=1:1:182
        meanVector(j,i)=meanVector(j,i)-mean;
    end
end

finalData=featureVector'*meanVector';%get the final data which is processed use PCA method


net = newff(finalData,target,5);%{'logsig','logsig','logsig'}
        net.trainParam.epochs = 50;
        net.trainParam.goal = 0.001;
        net.trainParam.lr = 0.1;
        net = train(net,finalData,target);
        simplefitOutputs = sim(net,finalData);
        plot(target,'ro')
        hold
        plot(simplefitOutputs,'c*')

end
```

Part of Training Data

pcar	pcadrdt	q1	q2dys	q2nor	q2off	q3	q4	q5	q6	q7	q8speed
0.610227	-0.37944	-1.94056	0.481702	-0.49043	0.177958	-0.33825	-1.01653	0.497355	-0.36278	-0.4321	-0.54175
-0.03729	0.093448	1.212851	-0.94661	1.502236	-1.15027	0.608847	-1.01653	0.497355	-0.36278	0.900829	-2.59466
-0.60901	0.714005	-1.94056	0.883413	-1.41804	1.0975	-0.33825	-1.01653	0.497355	-0.36278	1.567295	-1.06964
0.568695	-0.35441	-1.15221	-0.23245	-1.17755	2.017041	-1.28534	0.034654	-0.74262	-1.29271	-1.76504	0.22076
0.648927	-0.5659	1.212851	0.213894	0.608974	-1.15027	1.555944	0.034654	0.497355	0.567158	0.900829	0.22076
-1.58739	0.95081	-1.94056	0.437067	0.024918	-0.53724	0.608847	0.034654	0.497355	0.567158	0.900829	-1.42157
-2.5544	2.609694	-1.94056	0.883413	-1.34933	0.995328	1.555944	-2.06772	0.497355	-1.29271	0.900829	-1.42157
0.72917	-0.71655	-0.36385	-0.90197	0.299767	0.586643	0.608847	1.08584	0.497355	0.567158	-0.4321	-0.01386
-0.58441	-0.23434	1.212851	-0.94661	1.502236	-1.15027	1.555944	1.08584	0.497355	1.497093	0.234362	0.689997
0.327344	-0.42342	1.212851	0.035356	0.746398	-1.15027	0.608847	0.034654	0.497355	0.567158	0.234362	0.689997
0.585769	0.551865	1.212851	-0.94661	1.158673	-0.63941	1.555944	1.08584	0.497355	1.497093	0.234362	1.628469
-3.46138	3.675998	-1.94056	1.865375	-1.17755	-0.38398	-1.28534	-2.06772	-4.46256	-0.36278	0.900829	-0.77637
-0.49573	-0.45011	0.424498	-0.72343	0.712042	-0.23073	-0.33825	0.034654	-0.74262	0.567158	-1.09857	-2.36004
0.158335	0.174643	0.424498	-0.72343	1.330454	-1.15027	0.608847	1.08584	0.497355	0.567158	0.234362	-0.01386
0.302806	-0.05812	1.212851	-0.58953	0.127986	0.484472	0.608847	0.034654	0.497355	0.567158	0.234362	-0.30713
0.826944	-0.19823	1.212851	1.597568	-1.62418	0.586643	0.608847	1.08584	0.497355	1.497093	0.234362	2.15636
0.759529	-0.47534	0.424498	-0.8127	0.952536	-0.48616	-0.33825	1.08584	0.497355	-0.36278	-0.4321	0.748651

Dalarna University
Röda vägen 3
S-781 88 Borlänge
Sweden

Tel: +46 (0)23 77 80 00
Fax: +46 (0)23 7780 80
www.du.se

Part of Testing Data

pcar	pcadrdt	q1	q2dys	q2nor	q2off	q3	q4	q5	q6	q7	q8speed
0.641608	-0.58105	-1.89737	1.319313	-1.06424	0.108313	-1.15267	-1.17852	0.564774	-0.25268	1.70868	1.121118
0.206411	-0.1209	-0.31623	-0.75432	-1.06424	2.049504	-2.02811	-1.17852	-0.73855	-0.25268	-1.13912	0.42098
0.415662	-0.22017	-1.89737	0.552969	0.17273	-0.73568	-1.15267	-2.2499	-0.73855	-1.14449	-1.13912	-0.81772
1.062875	-0.98539	-1.89737	-0.75432	0.105867	0.572511	-1.15267	-0.10714	0.564774	-0.25268	-1.13912	-0.54844
-0.76465	-0.03983	-1.1068	0.237416	-0.93051	0.952309	-1.15267	-1.17852	-3.3452	-0.25268	-0.42717	-0.33301
-0.24067	-0.27182	1.264911	-0.93464	0.941656	-0.31369	0.598219	0.964244	0.564774	0.639131	-0.42717	0.367123
1.044973	-0.7169	0.474342	-0.84448	1.075383	-0.56688	-0.27722	0.964244	0.564774	-0.25268	-0.42717	0.690264
0.946039	-0.8407	-1.1068	-0.93464	0.039004	0.82571	-1.15267	0.964244	-2.04188	-1.14449	-1.85107	0.905691
-0.19767	0.336787	0.474342	3.573264	-1.69944	-1.19988	1.473662	-1.17852	0.564774	-0.25268	1.70868	-0.49458
-1.31227	-0.14294	0.474342	1.319313	-0.42904	-0.69348	1.473662	-0.10714	0.564774	0.639131	0.28478	-0.97929
0.682362	-0.46618	0.474342	0.50789	0.440183	-1.03108	-0.27722	0.964244	0.564774	0.639131	0.28478	2.252109
0.615075	-0.19227	-0.31623	-0.66416	-0.22845	0.910109	-1.15267	-0.10714	-0.73855	-2.0363	-1.85107	-0.11759
0.080327	-0.45797	0.474342	1.319313	-0.02786	-1.19988	1.473662	-1.17852	0.564774	0.639131	0.99673	1.875112
-0.20815	-0.18903	-0.31623	-0.7994	0.139298	0.572511	-1.15267	-0.10714	0.564774	0.639131	0.99673	-2.11029
0.703514	0.094363	0.474342	-0.93464	0.841362	-0.18709	0.598219	0.964244	0.564774	0.639131	-0.42717	0.09784
-0.25296	-0.18273	-1.1068	0.327574	-0.66306	0.530311	-0.27722	0.964244	0.564774	-1.14449	-1.13912	-1.19472
-0.08121	-0.34128	0.474342	1.048839	-0.86365	0.108313	0.598219	0.964244	0.564774	0.639131	-0.42717	-0.6023
0.376296	0.841909	0.474342	-0.93464	1.041951	-0.44028	-0.27722	-1.17852	0.564774	-0.25268	-0.42717	-0.17144
0.841705	-0.47114	0.474342	-0.93464	-1.46542	2.724701	-1.15267	0.964244	-2.04188	-0.25268	-1.13912	1.498115

Dalarna University
Röda vägen 3
S-781 88 Borlänge
Sweden

Tel: +46 (0)23 77 80 00
Fax: +46 (0)23 7780 80
www.du.se