

Classification

Artificial Intelligence

1 A

Task 1 - Data

The data that was given is not in an appropriate format for LibSVM. The first task is to convert the data sets into the format specified in the LibSVM 'README' file located in its root directory.

The source code for this task is located in **appendix A**.

The result was tested with the checkdata.py using the command line "*python checkdata.py fileName*" and the results were "*no error*".

Task 2 - Normalisation

The data that was converted is raw data, it has not been normalised (scaled). Normalising the data is very important when using SVMs as it stops attributes with a large numeric range from dominating the classification model.

This task required the use of svm-scale.exe in the command prompt, with the need of the un-normalised data and the new file as parameters, as shown in **appendix B**.

Task 3 - Grid Search

When using the C-SVC SVM with the Gaussian radial basis kernel there are two tuneable parameters, C (cost) and γ (gamma). To achieve the highest classification rate possible it is very important to search for an optimal pair of these values. LibSVM makes this process very simple by including a Python script which carries out a grid search, a systematic search for optimal SVM parameters.

This task is to carry out a grid search on the training data set.

At first a general idea for this task was made by using the default values for C and γ using the command prompt - *python grid.py normalisedFile*, results seen in **appendix C**.

Using the values of C and γ , and working out the log2 of each a finer search could be made using a longer command prompt - *python grid.py -log2c beg, end, step - log2g beg, end, step normalisedFile*, as seen in **appendix D**.

The values of the first grid search for C was 2 to power 9, so for the finer search a +1/-1 of the power value was used as the beginning and end of the search.

Task 4 - Classification

This task involves using LibSVM's 'svm-train' and 'svm-predict', both command line applications.

With the normalised training set as the input file, 'svm-train' can be used with the suitable parameter values discovered for c and γ during task 3 (seen at the bottom of **appendix E**). When the classification model is built the use of 'svm-predict' will be needed to classify the normalised testing set.

The command line of *svm-train.exe - g 0.0473661427034 - c 362.038671968 train.dat* was used first which outputted a .model file to be used by 'svm-predict'.

'svm-predict' used this file in conjunction with the normalised training set to create a predicted file, using the command line *svm-predict.exe test.dat train.dat.model train.dat*

As seen in **appendix F**.

SVM	Number of misclassified instances	Accuracy rate %
Non-linear	47	60.8333

Task 5 - Classification Analysis

By using the output file of 'svm-predict' (generated during task 4), it is possible to map the classifications made back to the original instances. The order of the predictions in the output file is the same as the order of the instances in the testing data set.

SVM	Number of misclassified instances	Accuracy rate %
Non-linear	47	80.83333333

Label	Actual label	Predicted label
Defective	60	29
Non-defective	60	91

Could for this can be found in **appendix G** as well as the command prompt view in **appendix H**.

As it can be seen by the output there was a total of 47 misclassified instances which was the same as task 4 but the only difference is that the accuracy between task 4 and 5 has a 20% difference.

This is because the features used for this test did not separate the classes were well.

Task 6 - Linear Classification

When using the C-SVC SVM with the linear kernel there is only one tunable parameter, C (cost). In this experiment, it is required to use LibSVM's 'svm-train' to train models using C = 10, C = 100, and C = 1000, in turn.

SVM	C	γ	Accuracy rate %
Linear	10	-	60.8333
Linear	100	-	60.8333
Linear	1000	-	61.6667

A look at the command prompt can be found in **appendix I**.

As shown by the table C = 1000 gives the best accuracy which is total different from task 4, as task 4 has the same accuracy as C = 10 and C = 100. Without a look at the full accuracy rate(full decimal) it is hard to determine if the accuracy is increasing with cost with just these three C values.

Appendices

Appendix A:

```
1. import csv
2. import sys
3.
4. inputFile = sys.argv[1] #input file
5. outputFile = sys.argv[2] # output file
6.
7. with open(inputFile, 'rb') as csvFile:
8.     writeToFile = [] #list
9.     myReader = csv.reader(csvFile, delimiter=',')#read from csv file
10.    for row in myReader:
11.        if row[len(row) - 1]== '1': # check character at end of line
12.            newLine = "+" + row[len(row)- 1] # add new line with '+1' at front
13.        else:
14.            newLine = "" + row[len(row) - 1]# add new line with '-1' at front
15.        del row[len(row) - 1]# delete it
16.        count = 1
17.        for v in row:
18.            newLine = newLine + " " + str(count) + ":" + v # add contents to send line
19.            count += 1
20.        newLine = newLine + "\n"
21.        writeToFile.append(newLine)
22. csvFile.close()# close input file
23.
24. del writeToFile[0]# delete title
25.
26. newFile = open(outputFile, "w")# create new file
27. for w in writeToFile:
28.     newFile.write(w)#add contents of the list to the new file
29.
30. newFile.close()# close output file
```

Appendix B:

```
cmd
C:\WINDOWS\system32\cmd.exe
C:\Users\User\Desktop\libsvm-master\windows\svm-scale.exe -s param testingSeta.dat
1 1:-0.965517 2:-0.833333 3:-1 4:-0.909091 5:-0.833333 6:-0.882353 7:-1 8:-0.709091 9:-0.672515 10:-0.632959 11:-0.578947 12:-0.185185 13:-0.751553
1 1:-1 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.854545 9:-0.808433 10:-0.917603 11:-0.630411 12:-0.62963 13:-0.459032
1 1:-1 2:-0.75 3:-1 4:-1 5:-0.75 6:-0.647059 7:-1 8:-0.654545 9:-0.672515 10:-0.662021 11:-0.368421 12:-0.259259 13:-0.73913
1 1:-1 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-1 9:-0.976608 10:-0.985019 11:-0.894737 12:-0.925926 13:-0.987578
1 1:-1 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.963636 9:-0.976608 10:-0.955056 11:-0.929825 12:-0.62963 13:-0.975155
1 1:-1 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.981818 9:-0.988304 10:-0.985019 11:-0.929825 12:-0.851852 13:-1
1 1:-0.931034 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.890909 9:-0.94152 10:-0.917603 11:-0.859649 12:-0.703704 13:-0.900621
1 1:-0.931034 2:-0.833333 3:-1 4:-1 5:-0.916667 6:-0.882353 7:-1 8:-0.6 9:-0.789474 10:-0.670412 11:-0.54386 12:-0.185185 13:-0.781863
1 1:-0.931034 2:-0.833333 3:-1 4:-1 5:-0.833333 6:-0.764706 7:-1 8:-0.781818 9:-0.824561 10:-0.850187 11:-0.719298 12:-0.185185 13:-0.826087
1 1:-1 2:-0.833333 3:-1 4:-1 5:-0.916667 6:-0.882353 7:-1 8:-0.727273 9:-0.883041 10:-0.820225 11:-0.754386 12:-0.111111 13:-0.813665
1 1:-0.206897 3:-1 4:-1 6:-0.294118 8:-1 9:-0.438596 10:-0.790262 11:-0.333333 12:-0.925926 13:-1
1 1:-0.965517 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.963636 9:-0.964912 10:-0.955056 11:-0.859649 12:-0.62963 13:-0.962733
1 1:-0.862069 2:-1 3:-1 4:-1 5:-0.545455 6:-1 6:-1 7:-1 8:-0.872727 9:-0.812865 10:-0.859649 11:-0.614035 12:-0.62963 13:-0.714286
1 1:-1 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.945455 9:-0.964912 10:-0.947566 11:-0.859649 12:-0.555556 13:-0.937888
1 1:-0.689552 2:-0.5 3:-1 4:-0.727273 5:-0.5 6:-0.294118 7:-1 8:-0.438596 9:-0.391813 10:-0.161049 11:-0.492982 12:-0.037037 13:-0.291925
1 1:-0.896552 2:-0.666667 3:-1 4:-1 5:-0.666667 6:-0.647059 7:-0.8 8:-0.4 9:-0.578947 10:-0.580524 11:-0.403509 12:-0.111111 13:-0.552795
1 1:-1 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.981818 9:-0.976608 10:-0.970037 11:-0.894737 12:-0.851852 13:-0.987578
1 1:-0.896552 2:-0.916667 3:-1 4:-1 5:-0.818182 9:-0.916667 6:-0.882353 7:-1 8:-0.709091 9:-0.847953 10:-0.835206 11:-0.578947 12:-0.407407 13:-0.73913
1 1:-0.896552 2:-0.916667 3:-1 4:-1 5:-0.916667 6:-0.882353 7:-1 8:-0.8 9:-0.883041 10:-0.872659 11:-0.719298 12:-0.407407 13:-0.801242
1 1:-0.310345 2:-0.333333 3:-1 4:-0.5 9:-0.333333 6:-0.294118 7:-0.4 8:-0.236364 9:-0.169591 10:-0.220974 11:-0.333333 12:-0.481481 13:-0.279503
1 1:-0.62069 2:-0.583333 3:-0.4 4:-0.909091 5:-0.583333 6:-0.647059 7:-0.6 8:-0.218182 9:-0.345029 10:-0.450712 11:-0.0877193 12:-0.111111 13:-0.130435
1 1:-0.965517 2:-0.916667 3:-1 4:-1 5:-0.916667 6:-0.833333 7:-1 8:-0.872727 9:-0.918129 10:-0.910112 11:-0.859649 12:-0.481481 13:-0.900621
1 1:-0.931034 2:-0.666667 3:-1 4:-1 5:-0.666667 6:-0.647059 7:-0.8 8:-0.345455 9:-0.532164 10:-0.475965 11:-0.368421 12:-0.037037 13:-0.52795
1 1:-0.689655 2:-0.333333 3:-1 4:-1 5:-0.636364 5:-0.333333 6:-0.647059 7:-0.1 8:-0.961818 9:-0.754386 10:-0.842697 11:-0.859649 12:-0.703704 13:-0.57764
1 1:-1 2:-0.916667 3:-1 4:-1 5:-0.916667 6:-0.882353 7:-1 8:-0.945455 9:-0.988304 10:-0.947566 11:-0.964912 12:-0.555556 13:-0.962733
1 1:-1 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.909091 9:-0.883041 10:-0.917603 11:-0.614035 12:-0.925926 13:-0.875776
1 1:-0.793103 2:-0.25 3:-1 4:-0.363636 5:-0.25 6:-0.0588235 7:-1 8:-0.0545455 9:-0.146199 10:-0.076617 11:-0.192982 12:-0.111111 13:-0.192547
1 1:-1 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.927273 9:-0.94152 10:-0.910112 11:-0.820561 12:-0.481481 13:-0.950311
1 1:-1 2:-1 3:-1 4:-1 5:-0.909091 5:-0.61 6:-1 7:-1 8:-1 9:-0.976608 10:-0.970037 11:-0.894737 12:-0.851852 13:-0.975155
1 1:-1 2:-1 3:-1 4:-1 5:-0.916667 3:-1 4:-1 5:-0.916667 6:-1 7:-1 8:-0.981818 9:-0.94152 10:-0.932584 11:-0.789474 12:-0.407407 13:-0.987578
1 1:-0.862069 2:-0.583333 3:-1 4:-1 5:-0.583333 6:-0.882353 7:-0.5 8:-0.727273 9:-0.730994 10:-0.670412 11:-0.614035 12:-0.333333 13:-0.763975
1 1:-0.827586 2:-0.416667 3:-1 4:-1 5:-0.818182 5:-0.416667 6:-0.176471 7:-0.5 8:-0.363636 9:-0.415205 10:-0.340824 11:-0.0526316 12:-0.111111 13:-0.478261
1 1:-0.965517 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.909091 9:-0.883041 10:-0.910112 11:-0.789474 12:-0.851852 13:-0.925466
1 1:-0.758621 2:-0.75 3:-1 4:-0.727273 5:-0.75 6:-0.764706 7:-0.8 8:-0.363636 9:-0.22807 10:-0.376277 11:-0.368421 12:-0.333333 13:-0.36646
1 1:-0.709103 2:-0.833333 3:-0.6 4:-0.909091 5:-0.833333 6:-0.764706 7:-1 8:-0.709091 9:-0.824561 10:-0.797753 11:-0.684211 12:-0.407407 13:-0.73913
1 1:-0.965517 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.909091 9:-0.883041 10:-0.842697 11:-0.754386 12:-0.481481 13:-0.900621
1 1:-0.965517 2:-0.75 3:-1 4:-1 5:-0.75 6:-0.647059 7:-1 8:-0.490909 9:-0.684211 10:-0.677903 11:-0.614035 12:-0.185185 13:-0.639752
1 1:-0.793103 2:-0.666667 3:-1 4:-1 5:-0.727273 5:-0.666667 6:-0.764706 7:-1 8:-0.254545 9:-0.356725 10:-0.370787 11:-0.0526316 12:-0.037037 13:-0.36646
1 1:-0.965517 2:-0.833333 3:-1 4:-1 5:-0.833333 6:-0.764706 7:-0.8 8:-0.854545 9:-0.871345 10:-0.805243 11:-0.754386 12:-0.259259 13:-0.875776
1 1:-0.896552 2:-0.666667 3:-1 4:-1 5:-0.666667 6:-0.666667 6:-0.529412 7:-1 8:-0.290909 9:-0.48538 10:-0.483146 11:-0.0877193 12:-0.037037 13:-0.428571
1 1:-1 2:-0.916667 3:-1 4:-1 5:-0.916667 6:-0.882353 7:-1 8:-0.727273 9:-0.918129 10:-0.872659 11:-0.824561 12:-0.481481 13:-0.913043
1 1:-1 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-1 9:-0.975528 11:-0.964912 10:-0.777778 13:-1
1 1:-1 2:-1 3:-1 4:-0.909091 5:-1 6:-1 7:-1 8:-0.909091 9:-0.929825 10:-0.955056 11:-0.754386 12:-0.703704 13:-0.925466
1 1:-0.965517 2:-0.75 3:-1 4:-1 5:-0.75 6:-0.647059 7:-0.7 8:-0.890909 9:-0.836257 10:-0.842697 11:-0.719298 12:-0.259259 13:-0.913043
1 1:-0.965517 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.872727 9:-0.812865 10:-0.835206 11:-0.54386 12:-0.333333 13:-0.900621
1 1:-1 2:-1 3:-1 4:-0.727273 5:-1 6:-1 7:-1 8:-0.436364 9:-1 10:-1 11:-0.894737 12:-1 13:-0.751553
1 1:-1 2:-1 3:-1 4:-0.909091 5:-1 6:-1 7:-1 8:-0.763636 9:-0.824561 10:-0.932584 11:-0.54386 12:-0.777778 13:-0.801242
1 1:-0.551724 2:-0.75 3:-1 4:-1 5:-0.75 6:-0.647059 7:-1 8:-0.690909 9:-0.710298 10:-0.685393 11:-0.719298 12:-0.259259 13:-0.627329
1 1:-0.827586 2:-0.75 3:-1 4:-0.818182 5:-0.75 6:-0.647059 7:-1 8:-0.345455 9:-0.637427 10:-0.505543 11:-0.473684 12:-0.185185 13:-0.465839
1 1:-0.931034 2:-0.916667 3:-1 4:-1 5:-0.916667 6:-0.882353 7:-1 8:-0.781818 9:-0.918129 10:-0.842697 11:-0.789474 12:-0.333333 13:-0.826087
1 1:-0.896552 2:-0.916667 3:-1 4:-1 5:-0.909091 5:-0.916667 6:-0.882353 7:-1 8:-0.745455 9:-0.766082 10:-0.752009 11:-0.500772 12:-0.259259 13:-0.751553
1 1:-0.931034 2:-0.833333 3:-1 4:-1 5:-0.833333 6:-0.764706 7:-1 8:-0.781818 9:-0.836257 10:-0.775281 11:-0.719298 12:-0.185185 13:-0.776398
1 1:-0.931034 2:-1 3:-1 4:-0.909091 5:-1 6:-1 7:-1 8:-0.8 9:-0.719298 10:-0.713556 11:-0.403509 12:-0.407407 13:-0.73913
1 1:-0.793103 2:-0.75 3:-1 4:-1 5:-0.75 6:-0.647059 7:-1 8:-0.327273 9:-0.578947 10:-0.305768 11:-0.500772 12:-0.259259 13:-0.453416
1 1:-0.827586 2:-0.583333 3:-1 4:-1 5:-0.818182 5:-0.583333 6:-0.529412 7:-0.6 8:-0.290909 9:-0.368421 10:-0.348315 11:-0.0526316 12:-0.407407 13:-0.428571
1 1:-1 2:-0.965517 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.890909 10:-0.859649 10:-0.820225 11:-0.684211 12:-0.555556 13:-0.900621
1 1:-0.517241 2:-0.333333 3:-1 4:-1 5:-0.333333 6:-0.647059 7:-0.8 8:-0.454545 9:-0.438596 10:-0.355805 11:-0.578947 12:-0.62963 13:-0.167702
1 1:-0.965517 2:-0.916667 3:-1 4:-1 5:-0.916667 6:-0.882353 7:-1 8:-0.836364 9:-0.871345 10:-0.865169 11:-0.859649 12:-0.259259 13:-0.875776
1 1:-0.965517 2:-0.5 3:-1 4:-0.909091 5:-0.5 6:-0.882353 7:-0.4 8:-0.672727 9:-0.649123 10:-0.520599 11:-0.333333 12:-0.185185 13:-0.714286
1 1:-0.793103 2:-0.416667 3:-1 4:-0.909091 5:-0.416667 6:-0.176471 7:-0.6 8:-0.0545455 9:-0.426901 10:-0.393258 11:-0.0877193 12:-0.259259 13:-0.267801
1 1:-0.965517 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.981818 9:-0.988304 10:-0.970037 11:-0.929825 12:-0.703704 13:-0.975155
1 1:-1 2:-1 3:-1 4:-1 5:-0.636364 5:-1 6:-1 7:-1 8:-0.927273 9:-0.93216 10:-0.917603 11:-0.894737 12:-0.703704 13:-0.813665
1 1:-0.896552 2:-0.666667 3:-1 4:-1 5:-0.666667 6:-0.529412 7:-0.6 8:-0.527273 9:-0.625731 10:-0.602996 11:-0.614035 12:-0.037037 13:-0.614907
1 1:-0.965517 2:-0.708333 3:-1 4:-1 5:-0.666667 6:-0.529412 7:-1 8:-0.2 9:-0.602339 10:-0.475965 11:-0.263158 12:-0.185185 13:-0.403727
1 1:-0.931034 2:-0.666667 3:-1 4:-1 5:-0.666667 6:-0.647059 7:-0.8 8:-0.4 9:-0.520488 10:-0.535581 11:-0.333333 12:-0.111111 13:-0.552795
1 1:-0.896552 2:-0.666667 3:-1 4:-1 5:-0.666667 6:-0.529412 7:-0.7 8:-0.690909 9:-0.777778 10:-0.692884 11:-0.684211 12:-0.037037 13:-0.714286
1 1:-0.931034 2:-0.833333 3:-1 4:-1 5:-0.833333 6:-0.764706 7:-1 8:-0.781818 9:-0.859649 10:-0.885243 11:-0.789474 12:-0.407407 13:-0.826087
1 1:-0.965517 2:-0.916667 3:-1 4:-1 5:-0.916667 6:-0.882353 7:-1 8:-0.93364 9:-0.918129 10:-0.865169 11:-0.789474 12:-0.481481 13:-0.975776
1 1:-0.965517 2:-0.916667 3:-1 4:-1 5:-0.909091 5:-1 6:-1 7:-1 8:-0.836364 9:-0.929825 10:-0.857678 11:-0.789474 12:-0.481481 13:-0.850932
1 1:-1 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.981818 9:-0.964912 10:-0.962547 11:-0.859649 12:-0.703704 13:-0.975155
1 1:-1 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.963636 9:-0.988304 10:-0.917603 11:-0.929825 12:-0.407407 13:-0.975155
1 1:-1 2:-1 3:-1 4:-0.909091 5:-1 6:-1 7:-1 8:-0.981818 9:-0.976608 10:-0.947566 11:-0.894737 12:-0.62963 13:-0.975155
1 1:-0.965517 2:-0.916667 3:-1 4:-0.727273 5:-0.916667 6:-0.882353 7:-1 8:-0.890909 9:-0.94152 10:-0.932584 11:-0.859649 12:-0.555556 13:-0.863354
1 1:-0.655772 2:-0.166667 3:-1 4:-1 5:-0.166667 6:-0.647059 7:-0.2 8:-0.227273 9:-0.0175439 10:-0.485891 11:-0.263158 12:-0.481481 13:-0.0062118
1 1:-0.965517 2:-0.75 3:-1 4:-1 5:-0.75 6:-0.764706 7:-0.8 8:-0.527273 9:-0.719298 10:-0.604449 11:-0.500772 12:-0.037037 13:-0.664596
1 1:-0.689655 2:-0.841667 3:-1 5:-0.833333 6:-0.411765 7:-0.3 8:-0.963636 9:-0.54386 10:-0.468105 11:-0.684211 12:-0.481481 13:-0.590062
1 1:-0.931034 2:-0.833333 3:-1 4:-1 5:-0.909091 5:-0.833333 6:-0.882353 7:-1 8:-0.818182 9:-0.812865 10:-0.775281 11:-0.684211 12:-0.481481 13:-0.838509
1 1:-0.758621 2:-0.666667 3:-1 4:-1 5:-0.666667 6:-0.764706 7:-0.8 8:-0.436364 9:-0.403509 10:-0.505618 11:-0.263158 12:-0.333333 13:-0.52795
1 1:-0.931034 2:-0.916667 3:-1 4:-1 5:-0.916667 6:-0.882353 7:-1 8:-0.781818 9:-0.684211 10:-0.713556 11:-0.473684 12:-0.407407 13:-0.763975
1 1:-0.896552 2:-0.833333 3:-1 4:-1 5:-0.833333 6:-0.764706 7:-1 8:-0.490909 9:-0.567251 10:-0.490637 11:-0.403509 12:-0.185185 13:-0.614907
1 1:-0.931034 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.872727 9:-0.871345 10:-0.80764 11:-0.614035 12:-0.62963 13:-0.850932
```

Appendix C

```
C:\WINDOWS\system32\cmd.exe

C:\Users\User\Desktop\libsvm-master\tools>python grid.py train.dat
gnuplot executable not found
[local] 5 -7 63.9474 (best c=32.0, g=0.0078125, rate=63.9474)
[local] -1 -7 59.2105 (best c=32.0, g=0.0078125, rate=63.9474)
[local] 5 -1 64.7368 (best c=32.0, g=0.5, rate=64.7368)
[local] -1 -1 62.6316 (best c=32.0, g=0.5, rate=64.7368)
[local] 11 -7 66.3158 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 11 -1 62.6316 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 5 -13 58.6842 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] -1 -13 54.4737 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 11 -13 63.9474 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] -3 -7 55.5263 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] -3 -1 60.2632 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] -3 -13 54.4737 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 5 1 62.6316 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] -1 1 63.1579 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 11 1 63.1579 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] -3 1 60.2632 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 9 -7 65.0 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 9 -1 61.0526 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 9 -13 61.5789 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 9 1 62.3684 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 5 -11 61.3158 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] -1 -11 54.4737 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 11 -11 65.0 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] -3 -11 54.4737 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 9 -11 64.2105 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 3 -7 61.5789 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 3 -1 64.2105 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 3 -13 54.4737 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 3 1 62.8947 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 3 -11 58.6842 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 5 -5 63.6842 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] -1 -5 60.5263 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 11 -5 65.2632 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] -3 -5 60.2632 (best c=2048.0, g=0.0078125, rate=66.3158)
[local] 9 -5 66.8421 (best c=512.0, g=0.03125, rate=66.8421)
[local] 3 -5 63.1579 (best c=512.0, g=0.03125, rate=66.8421)
[local] 15 -7 65.7895 (best c=512.0, g=0.03125, rate=66.8421)
[local] 15 -1 62.6316 (best c=512.0, g=0.03125, rate=66.8421)
[local] 15 -13 66.0526 (best c=512.0, g=0.03125, rate=66.8421)
[local] 15 1 63.6842 (best c=512.0, g=0.03125, rate=66.8421)
[local] 15 -11 66.0526 (best c=512.0, g=0.03125, rate=66.8421)
[local] 15 -5 63.9474 (best c=512.0, g=0.03125, rate=66.8421)
[local] 5 -15 54.4737 (best c=512.0, g=0.03125, rate=66.8421)
[local] -1 -15 54.4737 (best c=512.0, g=0.03125, rate=66.8421)
[local] 11 -15 61.5789 (best c=512.0, g=0.03125, rate=66.8421)
[local] -3 -15 54.4737 (best c=512.0, g=0.03125, rate=66.8421)
[local] 9 -15 61.3158 (best c=512.0, g=0.03125, rate=66.8421)
[local] 3 -15 54.4737 (best c=512.0, g=0.03125, rate=66.8421)
[local] 15 -15 65.0 (best c=512.0, g=0.03125, rate=66.8421)
[local] -5 -7 55.5263 (best c=512.0, g=0.03125, rate=66.8421)
[local] -5 -1 60.0 (best c=512.0, g=0.03125, rate=66.8421)
[local] -5 -13 54.4737 (best c=512.0, g=0.03125, rate=66.8421)
[local] -5 1 59.4737 (best c=512.0, g=0.03125, rate=66.8421)
[local] -5 -11 54.4737 (best c=512.0, g=0.03125, rate=66.8421)
[local] -5 -5 59.2105 (best c=512.0, g=0.03125, rate=66.8421)
[local] -5 -15 54.4737 (best c=512.0, g=0.03125, rate=66.8421)
[local] 5 3 63.9474 (best c=512.0, g=0.03125, rate=66.8421)
[local] -1 3 63.1579 (best c=512.0, g=0.03125, rate=66.8421)
[local] 11 3 65.0 (best c=512.0, g=0.03125, rate=66.8421)
[local] -3 3 61.5789 (best c=512.0, g=0.03125, rate=66.8421)
[local] 9 3 64.7368 (best c=512.0, g=0.03125, rate=66.8421)
[local] 3 3 65.0 (best c=512.0, g=0.03125, rate=66.8421)
[local] 15 3 64.7368 (best c=512.0, g=0.03125, rate=66.8421)
[local] -5 3 60.7895 (best c=512.0, g=0.03125, rate=66.8421)
[local] 7 -7 64.7368 (best c=512.0, g=0.03125, rate=66.8421)
[local] 7 -1 62.8947 (best c=512.0, g=0.03125, rate=66.8421)
[local] 7 -13 61.3158 (best c=512.0, g=0.03125, rate=66.8421)
[local] 7 1 64.7368 (best c=512.0, g=0.03125, rate=66.8421)
[local] 7 -11 61.5789 (best c=512.0, g=0.03125, rate=66.8421)
[local] 7 -5 65.2632 (best c=512.0, g=0.03125, rate=66.8421)
[local] 7 -15 58.6842 (best c=512.0, g=0.03125, rate=66.8421)
[local] 7 3 64.7368 (best c=512.0, g=0.03125, rate=66.8421)
[local] 5 -9 61.5789 (best c=512.0, g=0.03125, rate=66.8421)
[local] -1 -9 54.4737 (best c=512.0, g=0.03125, rate=66.8421)
[local] 11 -9 65.7895 (best c=512.0, g=0.03125, rate=66.8421)
[local] -3 -9 54.4737 (best c=512.0, g=0.03125, rate=66.8421)
[local] 9 -9 65.2632 (best c=512.0, g=0.03125, rate=66.8421)
[local] 3 -9 61.3158 (best c=512.0, g=0.03125, rate=66.8421)
[local] 15 -9 65.5263 (best c=512.0, g=0.03125, rate=66.8421)
[local] -5 -9 54.4737 (best c=512.0, g=0.03125, rate=66.8421)
[local] 7 -9 63.9474 (best c=512.0, g=0.03125, rate=66.8421)
```

Appendix D

```
C:\WINDOWS\system32\cmd.exe - python grid.py -log2c 8,10,0.1 -log2g -6,-4,0.1 train.dat
```

local	8.8	-4.7	67.6316	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.8	-5.2	66.8421	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.0	-4.1	65.0	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.5	-4.1	66.0526	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.0	-4.1	65.2632	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.6	-4.1	67.3684	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.3	-4.1	65.0	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.8	-4.1	65.2632	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.9	-5.0	66.3158	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.9	-5.0	66.3158	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.9	-4.4	65.2632	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.9	-5.8	67.1053	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.9	-4.7	65.2632	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.9	-5.2	66.0526	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.9	-4.1	64.4737	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.0	-5.9	66.0526	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.5	-5.9	65.2632	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.6	-5.9	66.5789	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.2	-5.9	65.0	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.3	-5.9	66.5789	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.8	-5.9	66.0526	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.1	-5.9	66.8421	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.9	-5.0	66.0526	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.1	-4.4	67.1053	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.1	-4.4	67.1053	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.1	-5.8	65.0	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.1	-4.7	66.3158	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.1	-5.2	65.2632	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.1	-4.1	67.1053	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.1	-5.9	64.7368	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.0	-4.8	67.6316	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.5	-4.8	66.8421	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.6	-4.8	65.7895	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.2	-4.8	66.3158	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.3	-4.8	66.8421	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.8	-4.8	67.3684	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.1	-4.8	66.3158	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	8.2	-5.0	67.3684	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.2	-4.4	65.7895	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.2	-4.7	66.3158	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.2	-4.8	66.3158	(best	=c362.038671968,	g=0.0473661427034,	rate=67.6316)
local	9.2	-5.2	66.5789	(best	=c36		

Appendix E

[illegible]

```
C:\Users\User\Desktop\libsvm-master\tools>
```


Appendix F

```
C:\WINDOWS\system32\cmd.exe

C:\Users\User\Desktop\libsvm-master\windows>svm-train.exe
Usage: svm-train [options] training_set_file [model_file]
options:
-s svm_type : set type of SVM (default 0)
  0 -- C-SVC          (multi-class classification)
  1 -- nu-SVC         (multi-class classification)
  2 -- one-class SVM   (multi-class classification)
  3 -- epsilon-SVR     (regression)
  4 -- nu-SVR          (regression)
-t kernel_type : set type of kernel function (default 2)
  0 -- linear: u'*v
  1 -- polynomial: (gamma*u'*v + coef0)^degree
  2 -- radial basis function: exp(-gamma*|u-v|^2)
  3 -- sigmoid: tanh(gamma*u'*v + coef0)
  4 -- precomputed kernel (kernel values in training_set_file)
-d degree : set degree in kernel function (default 3)
-g gamma : set gamma in kernel function (default 1/num_features)
-r coef0 : set coef0 in kernel function (default 0)
-c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)
-n nu : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)
-p epsilon : set the epsilon in loss function of epsilon-SVR (default 0.1)
-m cachesize : set cache memory size in MB (default 100)
-e epsilon : set tolerance of termination criterion (default 0.001)
-h shrinking : whether to use the shrinking heuristics, 0 or 1 (default 1)
-b probability_estimates : whether to train a SVC or SVR model for probability estimates, 0 or 1 (default 0)
-wi weight : set the parameter C of class i to weight*C, for C-SVC (default 1)
-v n: n-fold cross validation mode
-q : quiet mode (no outputs)

C:\Users\User\Desktop\libsvm-master\windows>svm-train.exe -g 0.0473661427034 -c 362.038671968
Usage: svm-train [options] training_set_file [model_file]
options:
-s svm_type : set type of SVM (default 0)
  0 -- C-SVC          (multi-class classification)
  1 -- nu-SVC         (multi-class classification)
  2 -- one-class SVM   (multi-class classification)
  3 -- epsilon-SVR     (regression)
  4 -- nu-SVR          (regression)
-t kernel_type : set type of kernel function (default 2)
  0 -- linear: u'*v
  1 -- polynomial: (gamma*u'*v + coef0)^degree
  2 -- radial basis function: exp(-gamma*|u-v|^2)
  3 -- sigmoid: tanh(gamma*u'*v + coef0)
  4 -- precomputed kernel (kernel values in training_set_file)
-d degree : set degree in kernel function (default 3)
-g gamma : set gamma in kernel function (default 1/num_features)
-r coef0 : set coef0 in kernel function (default 0)
-c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)
-n nu : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)
-p epsilon : set the epsilon in loss function of epsilon-SVR (default 0.1)
-m cachesize : set cache memory size in MB (default 100)
-e epsilon : set tolerance of termination criterion (default 0.001)
-h shrinking : whether to use the shrinking heuristics, 0 or 1 (default 1)
-b probability_estimates : whether to train a SVC or SVR model for probability estimates, 0 or 1 (default 0)
-wi weight : set the parameter C of class i to weight*C, for C-SVC (default 1)
-v n: n-fold cross validation mode
-q : quiet mode (no outputs)

C:\Users\User\Desktop\libsvm-master\windows>svm-train.exe -g 0.0473661427034 -c 362.038671968 train.dat
.....
optimization finished, #iter = 6891
nu = 0.679019
obj = -88940.469600, rho = -13.388583
nSV = 275, nBSV = 241
Total nSV = 275

C:\Users\User\Desktop\libsvm-master\windows>svm-predict.exe
Usage: svm-predict [options] test_file model_file output_file
options:
-b probability_estimates: whether to predict probability estimates, 0 or 1 (default 0); for one-class SVM only 0 is supported
-q : quiet mode (no outputs)

C:\Users\User\Desktop\libsvm-master\windows>svm-predict.exe test.dat train.dat.model train.dat
Accuracy = 60.8333% (73/120) (classification)

C:\Users\User\Desktop\libsvm-master\windows>svm-predict.exe test.dat train.dat.model trainpred.dat
Accuracy = 60.8333% (73/120) (classification)

C:\Users\User\Desktop\libsvm-master\windows>
```

Appendix G

```
1. testFile = "test.dat"
2. trainFile = "trainpred.dat"
3.
4. actual = []#list of acutal instances
5. predicted = []#list of predicted instances
6. #Combine files into a single list
7. with open (testFile) as textFile1:# open test file
8.     with open(trainFile) as textFile2:# open predicted file
9.         for e in range(0, 120): # look at all instances
10.             line = textFile1.readline().rstrip('\n')#read line from test
11.
12.             actual.append(int(line[0:2]))
13.             line = textFile2.readline().rstrip('\n')
14.             predicted.append(int(line[0:2]))
15. textFile1.close()
16. textFile2.close()
17.
18. #Analysis
19. incorrect = 0
20. incorrectly_predicted_nondefective = 0
21. incorrectly_predicted_defective = 0
22. act_defective = 0
23. act_nondefective = 0
24. pred_defective = 0
25. pred_nondefective = 0
26.
27. for e in range(0,120):
28.     original = actual[e]
29.     classification = predicted[e]
30.
31.     if original == 1 and classification == -1:
32.         incorrectly_predicted_nondefective +=1
33.
34.     if original == -1 and classification == 1:
35.         incorrectly_predicted_defective += 1
36.
37.     if original == 1:
38.         act_defective += 1
39.
40.     if original == -1:
41.         act_nondefective += 1
42.
43.     if classification == 1:
44.         pred_defective += 1
45.
46.     if classification == -1:
47.         pred_nondefective += 1
48.
49.
50. incorrect = incorrectly_predicted_nondefective + incorrectly_predicted_defective
51.
52. print "No of incorrect instances: " + str(incorrect)
53. print "No of instances that labeled as defective and incorrectly predicted as non-
54.     defective: " + str(incorrectly_predicted_nondefective)
55. print "No of instances that labeled as non-
56.     defective and incorrectly predicted as defective: " + str(incorrectly_predicted_defective)
57. print "No of actual defective: " + str(act_defective)
58. print "No of actual non-defective: " + str(act_nondefective)
59. print "No of predicted defective: " + str(pred_defective)
60. print "No of predicted non-defective: " + str(pred_nondefective)
```

Appendix H

```
C:\WINDOWS\system32\cmd.exe

C:\Users\User\Desktop\AI - Coursework 2>python Svm-Compare.py
Traceback (most recent call last):
  File "Svm-Compare.py", line 9, in <module>
    with open(testFile) as textFile1:
IOError: [Errno 2] No such file or directory: 'test.dat'

C:\Users\User\Desktop\AI - Coursework 2>python Svm-Compare.py
No of actual defective: 0
No of actual non-defective: 120
No of predicted defective: 120
No of predicted non-defective: 0

C:\Users\User\Desktop\AI - Coursework 2>python Svm-Compare.py
No of actual defective: 60
No of actual non-defective: 60
No of predicted defective: 29
No of predicted non-defective: 91

C:\Users\User\Desktop\AI - Coursework 2>python Svm-Compare.py
No of misclassified instances: 47
39 instances that labeled as defective and incorrectly predicted as non-defective
8 instances that labeled as non-defective and incorrectly predicted as defective

No of actual defective: 60
No of actual non-defective: 60
No of predicted defective: 29
No of predicted non-defective: 91

C:\Users\User\Desktop\AI - Coursework 2>python Svm-Compare.py
No of misclassified instances: 47
No of instances that labeled as defective and incorrectly predicted as non-defective: 39
No of instances that labeled as non-defective and incorrectly predicted as defective: 8
No of actual defective: 60
No of actual non-defective: 60
No of predicted defective: 29
No of predicted non-defective: 91

C:\Users\User\Desktop\AI - Coursework 2>python Svm-Compare.py
No of incorrect instances: 0
No of instances that labeled as defective and incorrectly predicted as non-defective: 39
No of instances that labeled as non-defective and incorrectly predicted as defective: 8
No of actual defective: 60
No of actual non-defective: 60
No of predicted defective: 29
No of predicted non-defective: 91

C:\Users\User\Desktop\AI - Coursework 2>python Svm-Compare.py
No of incorrect instances: 47
No of instances that labeled as defective and incorrectly predicted as non-defective: 39
No of instances that labeled as non-defective and incorrectly predicted as defective: 8
No of actual defective: 60
No of actual non-defective: 60
No of predicted defective: 29
No of predicted non-defective: 91

C:\Users\User\Desktop\AI - Coursework 2>_
```

Appendix I

```
C:\WINDOWS\system32\cmd.exe

C:\Users\User\Desktop\libsvm-master\windows>svm-predict.exe test.dat train.dat.model trainpred.dat
Accuracy = 60.8333% (73/120) (classification)

C:\Users\User\Desktop\libsvm-master\windows>svm-train.exe -c 10 linear10.dat
can't open input file linear10.dat

C:\Users\User\Desktop\libsvm-master\windows>svm-train.exe -c 10 linear10.dat
*
optimization finished, #iter = 353
nu = 0.761022
obj = -2786.493036, rho = -1.986552
nSV = 296, nBSV = 282
Total nSV = 296

C:\Users\User\Desktop\libsvm-master\windows>svm-train.exe -c 10 linear100.dat
*
optimization finished, #iter = 353
nu = 0.761022
obj = -2786.493036, rho = -1.986552
nSV = 296, nBSV = 282
Total nSV = 296

C:\Users\User\Desktop\libsvm-master\windows>svm-train.exe -c 100 linear100.dat
...*.
optimization finished, #iter = 1546
nu = 0.697054
obj = -25215.107082, rho = -5.370211
nSV = 279, nBSV = 256
Total nSV = 279

C:\Users\User\Desktop\libsvm-master\windows>svm-train.exe -c 1000 linear1000.dat
.....*.....*.....*.....*.....*
optimization finished, #iter = 18928
nu = 0.619781
obj = -226640.637343, rho = -12.777832
nSV = 263, nBSV = 217
Total nSV = 263

C:\Users\User\Desktop\libsvm-master\windows>svm-predict.exe test.dat linear10.dat.model linear10pred.dat
Accuracy = 60.8333% (73/120) (classification)

C:\Users\User\Desktop\libsvm-master\windows>svm-predict.exe test.dat linear100.dat.model linear100pred.dat
Accuracy = 60.8333% (73/120) (classification)

C:\Users\User\Desktop\libsvm-master\windows>svm-predict.exe test.dat linear1000.dat.model linear1000pred.dat
Accuracy = 61.6667% (74/120) (classification)

C:\Users\User\Desktop\libsvm-master\windows>
```