Machine Learning Coursework 3 Report

**Group B members:**

|  |  |
| --- | --- |
| Name | Student ID |
| Koo Wai Kun | 025254 |
| Darren Cheng Sum Jun | 025299 |
| Ong Kok Ting | 025288 |
| Loh Jin Xian | 016763 |

# Table of content

|  |
| --- |
| 1.0 Introduction  2.0 Comparison  3.0 Questions section  4.0 Conclusion |
|
|

# 1.0 Introduction

Machine learning can be categorized into 2 categories namely supervised learning and unsupervised learning. Different learning algorithms are chosen based on the structure and volume of the data and the use case of the issue at hand. One of the recent breakthroughs in the field of machine learning is the introduction of support vector machines invented by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963. Support vector machine (SVM) is a supervised learning technique from the field of machine learning applicable to both classification and regression problems.

SVM works by constructing hyperplanes in a multidimensional space that separates cases of different class labels. Unlike multilayer feed-forward networks, SVM is non-paramedic as it doesn't have a fixed size. It allows linearly separable data to be classified and for non-linearly separable data, the expensive calculations needed to separate the data can be sidestepped by applying kernel trick which is proposed by Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik in 1992. Compared to other other algorithms like neural networks, SVM has a higher speed and performance advantage with a limited number of samples (in thousands). There are many kernels in SVM, for examples linear, Gaussian and polynomial. In the following sections, performance comparisons and the number of support vectors needed are made between different kernels and other learning algorithms.

# 

# 2.0 Comparison

The tables below are the comparison between different outputs from both classification and regression problems.

**Number, N and percentage, % of Support vectors selected**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fold/k** | **Classification** | | | | **Regression** | | | |
| **Gaussian** | | **Polynomial** | | **Gaussian** | | **Polynomial** | |
| **N** | **%** | **N** | **%** | **N** | **%** | **N** | **%** |
| **1** | 127 | 94.1 | 41 | 30.4 | 4041 | 50.1 | 1452 | 18.0 |
| **2** | 122 | 90.4 | 44 | 32.6 | 4065 | 50.4 | 1420 | 17.6 |
| **3** | 123 | 91.1 | 39 | 28.9 | 4090 | 50.6 | 1442 | 17.9 |
| **4** | 124 | 91.9 | 44 | 32.6 | 4065 | 50.4 | 1424 | 17.7 |
| **5** | 127 | 94.1 | 44 | 32.6 | 4087 | 50.7 | 1441 | 17.9 |
| **6** | 124 | 91.9 | 38 | 28.2 | 4043 | 50.2 | 1445 | 17.9 |
| **7** | 130 | 96.3 | 46 | 34.1 | 4060 | 50.4 | 1438 | 17.8 |
| **8** | 126 | 93.3 | 43 | 31.9 | 4071 | 50.5 | 1435 | 17.8 |
| **9** | 127 | 94.1 | 38 | 28.2 | 4102 | 50.9 | 1453 | 18.0 |
| **10** | 125 | 92.6 | 44 | 32.6 | 4062 | 50.4 | 1445 | 17.9 |

*Table 2.1 Number and percentage of support vectors selected*

**Classification**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fold Number** | **SVM - linear** | **SVM - RBF** | **SVM - Polynomial** | **ANN** | **Decision Tree** |
| 1 | 100 | 100 | 57.1 | 80 | 80 |
| 2 | 100 | 100 | 100 | 66.7 | 80 |
| 3 | 57.1 | 100 | 94.1 | 82.4 | 40 |
| 4 | 50 | 100 | 90.9 | 85.7 | 80 |
| 5 | 71.4 | 75 | 66.7 | 66.7 | 85.7 |
| 6 | 90.9 | 100 | 80 | 92.3 | 76.9 |
| 7 | 100 | 100 | 85.7 | 100 | 80 |
| 8 | 88.9 | 72.73 | 90.9 | 90.9 | 100 |
| 9 | 85.7 | 90.91 | 85.7 | 66.7 | 75 |
| 10 | 75 | 100 | 76.9 | 90.9 | 75 |
| **Average** | **81.91** | **93.86** | **82.81** | **82.2** | **77.26** |

*Table 2.2 Comparison of f1 value between different training models for classification problems*

**Regression**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fold Number** | **SVM - linear** | **SVM - RBF** | **SVM - Polynomial** | **ANN** |
| 1 | 5.5750 | 0.7229 | 3.2081 | 1.1884 |
| 2 | 5.4728 | 0.9410 | 3.3012 | 0.8871 |
| 3 | 5.9080 | 0.8505 | 2.8234 | 1.7405 |
| 4 | 5.4633 | 0.7171 | 3.1558 | 0.9336 |
| 5 | 5.2224 | 0.9749 | 2.9994 | 1.0502 |
| 6 | 5.4037 | 0.6180 | 2.8123 | 0.9952 |
| 7 | 6.2894 | 0.5767 | 2.8283 | 1.0170 |
| 8 | 5.3575 | 1.0435 | 2.9740 | 3.2668 |
| 9 | 5.2470 | 0.8471 | 3.6835 | 1.6853 |
| 10 | 5.0041 | 0.5865 | 4.3132 | 1.1868 |
| **Average** | **5.4943** | **0.7878** | **3.2099** | **1.3951** |

*Table 2.3 Comparison of root mean squared error (RMSE) between different models for regression problems*

From the results above shown, it is clear that SVM yields a better result for both classification and regression problem. For binary classification, SVM RBF gives the highest average F1 value for the 10 fold cross validation as compared to ANN and Decision tree. For regression problem, SVM RBF again gives the lowest root mean square error(RMSE) as compared to the one trained using Artificial Neural Network(ANN).

**T-test Comparison**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **ANN** | | **Decision tree** | |
| Statistically Significantly Different (Yes/No) | P-value | Statistically Significantly Different (Yes/No) | P-value |
| **SVM Linear** | No | 0.9643 | No | 0.7951 |
| **SVM RBF** | Yes | 0.0397 | No | 0.0791 |
| **SVM Polynomial** | No | 0.9175 | No | 0.8729 |
| **Decision tree** | No | 0.7864 |  |  |

*Table 2.4 T-Test comparison of binary classification dataset*

|  |  |  |
| --- | --- | --- |
|  | **ANN** | |
| Statistically Significantly different (Yes/No) | P-value |
| **SVM Linear** | Yes | 3.213e-16 |
| **SVM RBF** | Yes | 0.0029 |
| **SVM Polynomial** | Yes | 1.639e-09 |

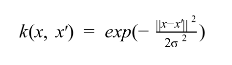
*Table 2.5 T-Test comparison of regression dataset*

All the comparisons from regression dataset are statistically significant. For binary classification dataset, all are statistically insignificant except the comparison between ANN and SVM RBF. Statistically insignificant means that ttest2 function returns 0 for the H value, which indicates a failure to reject the null hypothesis at the default 5% significance level. We use the pair using F1 measures for binary classification and RMSE for regression.

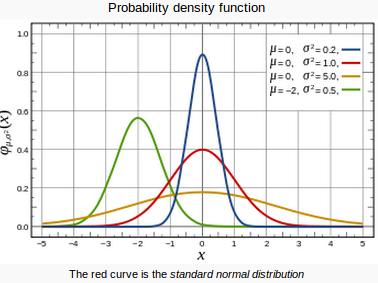
# 3.0 Questions section

3.1.1 What does the kernel parameter of the Gaussian RBF kernel signify (sigma)?

As shown in the kernel function below, sigma represents the kernel parameter.



The sigma value is often defined as the standard deviation in a Gaussian Distribution. Standard deviation determines the width for the Gaussian distribution as shown in the figure below.



In Gaussian kernel, sigma affects the decision boundary

defined by our SVM. The sigma value affects the decision boundary by determining the distance of points from the boundary. A larger sigma takes into account points further from the boundary, while a smaller sigma has a smaller reach.

3.1.2 What happens when you increase its value?

As the value of sigma increases, the width of the distribution increases. Points further from the boundary is taken into account. This leads to a more general classifier that is smoother and more linear. This prevents overfitting, however the rate of misclassification increases. A larger sigma tends to be of less variance and is more bias. The decision boundary from a large sigma value is also less flexible.

3.2.1 Explain what happens when a hard-margin SVM is fit to a dataset of two classes with overlapping features.

Hard-margin SVM is only defined when the data are linearly separable, i.e. we can draw a line where every data point in the first class is on one side and every point in the other class is on the other side of the line. Since the data is not linearly separable in feature space, it is impossible to find any hyperplane that satisfies it without using feature transformations.

3.2.2 What value do you need to set C (the slack variable hyper-parameter) to attain a hard-margin SVM?

We should set a large value for C as high C value severely penalize misclassified data points, and thus choosing a small margin with few support vectors, which is still the objective of using a hard-margin SVM.

# 

# 4.0 Conclusion

In conclusion, Support Vector Machine yields better results compared to Artificial Neural Network and Decision Tree in our tests. For binary classification problem, SVM RBF gives the best F1 score with 93.86% average while for the regression problem, SVM RBF again scored the lowest Root Mean Square Error (RMSE) among all other training models.

Training an SVM with Gaussian RBF requires finding the optimal hyperparameters. A suitable sigma value should be used in order to produce a flexible decision boundary that does not overfit. Using the inner-fold cross-validation method, we are able to acquire the optimal hyperparameters. We were able to get the best results for both classification and regression using SVM RBF, utilizing the hyperparameters obtained from the inner-fold cross-validation.