### **KE5206 CI1, CA2 PROGRAMMING ASSIGNMENT SUBMISSION**

*Pratyush Mishra (A0178266R)*

*Apurv Garg (A0178205E)*

*Rajiv Hemanth (A0178387J)*

#### **ABSTRACT**

This document analyses the accuracy scores of a base approach to scikit-learn SVM, and proposes and tests a series of changes aimed at improving efficiency and accuracy

**1. INTRODUCTION**

This document takes for inspiration “A Practical Guide to Support Vector Classification” and categories and tests a series of approaches outlined in order to achieve higher accuracy and faster run-times of the basic SVM approach

The SVM basic approach is provided by the sklearn.svm package and is implemented in basic\_svm.py using python. The dataset is provided by <http://archive.ics.uci.edu/ml/datasets/> and the banking dataset hosted 20 attributes, 1 classification variable and 4119 instances. Since a larger dataset was expected to increase accuracy, the extended version of the dataset, with 41188 instances was used as well

The experimentation relied on parameter fine-tuning as well as cross validation, however the latter approach proved unsuccessful due to it being computationally expensive

**2. BASELINE APPROACH**

The baseline approach of this SVM implements sklearn.svm.SVC class in python. The following table contains the basic parameter that are set as default in this class and are assumed by us as the baseline approach.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Default** | **Potential** |
| C | 1 | A variation in the penalty parameter can, at the expense of accuracy, improve the efficiency of the script or vice versa increase accuracy by increasing the number of potential iterations and hence increase the cost of operation |
| Kernel | rbf | Changes in the kernel function are expected to be the largest contributing factor in terms of the runtime of the program. RBF will be used to benchmark the performance of the other kernels, namely, linear, sigmoid and polynomial |
| gamma | auto | gamma function is used in all kernel functions except linear. Auto gamma yields 1/n\_features as gamma |
| coef0 | 0.0 | Independent term in polynomial and sigmoidal kernel functions |
| probability | false | Determines whether to enable probability estimates. Can increase accuracy at the cost of processing time |
| Shrinking | True | Determines whether to use shrinking heuristics.  eg:-  step\_sizes = [self.max\_feature\_value \* 0.1,  self.max\_feature\_value \* 0.01,  self.max\_feature\_value \* 0.001,  ]    Can become increasingly costly to implement |
| tol | 1-e3 | Tolerance for stopping criterion. This is unlikely to have any major impact on the result unless altered dramatically |
| cache size | - | Kernel size in Mb. No effects expected |
| class weight | balanced | Factors the decision weight of a column. For this experiment, it is left balanced |
| verbose | False | No effect |
| max\_iter | -1 | Hard limit on iterations within solver, or -1 for no limit. Reducing will limit long runtimes but may not be accurate |
| decision\_function\_shape | ovr | Whether to return a one-vs-rest (‘ovr’) decision function of shape (n\_samples, n\_classes) as all other classifiers, or the original one-vs-one (‘ovo’) decision function of libsvm which has shape (n\_samples, n\_classes \* (n\_classes - 1) / 2).[[1]](#footnote-0) |
| random\_state | none | Modified to 6 in order to maintain a uniform split between training and testing data to remove variations caused by random seed |

**2.1. Results of basic approach**

The calculated accuracy over the given data set with R=6 according to svm class is: **87.1566%**

**3. PROPOSED APPROACH**

The proposed approach to increase the accuracy of the SVM is based on conducting a series of alterations to the conditions of the dataset and the classifier and their functional parameters and recording the benefits they bring to the control SVM as described in the baseline approach. The following alterations will be performed:

1. Covariance based column elimination
2. Change in penalty parameter
3. Change of Kernel function to Linear kernel function
4. Change of Kernel function to Polynomial kernel function
5. Change of Kernel function to Sigmoidal kernel function
6. Enabling probability estimates before calling the fit() function
7. Elimination of shrinking heuristics
8. Altering shape of decision function
9. Altering end tolerance
10. Extending the data set i.e using a larger instance of the data set

**4. EXPERIMENTAL RESULTS**

|  |  |
| --- | --- |
| Change | Accuracy |
| Base: | 87.1566 |
| Covar | 87.1566 |
| set a larger dataset | 89.0777 |
| set c=100 | 88.4709 |
| set c=0.1 | 89.1990 |
| set linear kernel | 89.1990 |
| set polynomial kernel | Failure |
| set sigmoid kernel | 89.1990 |
| Enable probability estimates | 89.4417 |
| Elimination of shrinking heuristics | 88.6717 |
| Altering shape of decision function | 89.199 |

**5. CONCLUSIONS**

The following tests increased accuracy:

1. Increasing the size of dataset
2. Setting a smaller penalty parameter
3. Enabling probability estimates
4. Enabling shrinking heuristics

The following tests decreased runtime:

(52892 base iterations):

1. C=0.1 : Used fewer iterations
2. Linear kernel used almost double iterations but was significantly faster
3. Sigmoid kernel was significantly faster and used only 80% of iterations used by rbf

An attempt to implement K-fold cross validation was met with failure due to the extremely expensive cost of K SVMs

We can therefore assert that implementing the following changes marked in green into our program will increase efficiency

Recorded efficiency will all changes inserted: **91.0177%**

**REFERENCES**

[1] A Practical Guide to Support Vector Classification, Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, Department of Computer Science, National Taiwan University, Taipei 106, Taiwan

[2] Improving Efficiency of SVM k-Fold Cross-Validation by Alpha Seeding, Zeyi Wen,1 Bin Li,2 Kotagiri Ramamohanarao,1 Jian Chen,2∗ Yawen Chen,2 Rui Zhang1 1{zeyi.wen, kotagiri, rui.zhang}@unimelb.edu.au

The University of Melbourne, Australia

2{gitlinux@gmail.com,ellachen@scut.edu.cn, elfairyhyuk@gmail.com}

South China University of Technology, China

1. [sklearn.svm.SVC documentation](http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html) [↑](#footnote-ref-0)