Fundamentals of Statistical Learning and Pattern Recognition

CSE 569

Project Report on SVM Classification

Submitted To

Prof. Dr. Baoxin Li

Submitted By

Sai Vikhyath Kudhroli

ASU Id: 1225432689

Date: 12 November 2022

INTRODUCTION

Problem Statement: To understand the usage of libsvm package by performing classification. To train SVM models using different parameters for different tasks to obtain and compare classification accuracies.

Data Description: The dataset consists of 4786 training samples and 1833 testing samples stored in *"trainData.mat"* and *"testData.mat"*. The dataset consists of 50 class labels. Each sample is described by three different feature matrices.

Each feature matrix in training data is of the dimension 4786 x 1000 and in testing data is of the dimension 1833 x 1000.

LIBSVM

Libsvm is a library that is used for SVM classification and regression. Libsvm supports python interface to train SVM models.

Installation in Python: Libsvm can be installed using pip package installer.

Syntax: pip install -U libsvm-official

Usage of libsvm:

- svm-train - This method is used to train SVM model.

  model = svm_train(train_labels, train_data, options)

  where options is a string formed by the following

  -s: svm_type

  0: C-SVC used for multi-class classification. Default option.

  1: nu-SVC used for multi-class classification with nu parameter.

  2: one-class SVM

  3: epsilon-SVR for regression

  4: nu-SVR for regression

  -t: kernel_type

  0: linear

  1: polynomial

  2: radial basis function. Default option

  3: sigmoid

  4: precomputed kernel

  -d: degree in kernel function. 3 by default.

  -c: cost of C-SVC, epsilon-SVR, and nu-SVR. 1 by default.

  -n: nu of nu-SVC, one-class SVM, and nu-SVR. 0.5 by default.

  -p: epsilon in loss function of epsilon-SVR. 0.1 by default.

  -b: whether to train a model for probability estimates. 0 by default.

  -wi: weight*C as parameter C of class i for C-SVC. 1 by default

- svm-predict – This method is used to perform prediction of test data using trained models.

  label, accuracy, value = svm_predict(test_labels, test_data, trained_model, options)

  where only -b can be specified as options.

PROCEDURE

Task 0.1 - Classification by individual features: For this task, each of the feature matrices are trained using svm_train method using a linear kernel (-t 0) and a cost of 10 (-c 10). From which three SVM models are obtained, trained on X1, X2 and X3 respectively from training data. These models are then tested on the test data using svm_predict. From which accuracy of the models on test data is obtained.

Task 0.2 – Classification by individual features with probability estimates: For this task, each of the feature matrices are trained using svm_train method using a linear kernel (-t 0), a cost of 10 (-c 10) and setting the probability estimates option to 1 (-b 1). From which three SVM models are obtained, trained on X1, X2 and X3 respectively from training data. These models are then tested on the test data using svm_predict with the option probability estimates set to 1 (-b 1). From which the accuracy of the models on test data is obtained.

Task 1 – Feature combination by fusion of classifiers: For this task, the probability estimates obtained after predicting with the probability estimates option are combined and used for prediction of the labels.

Probability estimates on prediction using *"-b 1"* option are obtained in values attribute which are combined using the following formula.

$$p\left(\frac{\omega_i}{x}\right) = \frac{\sum_{k=1}^{3} p_k\left(\frac{\omega_i}{x}\right)}{3}$$

The final labels are computed as follows

$$w_{i^*} = argmax_i\left(p\left(\frac{w_i}{x}\right)\right)$$

Which means that the label for which the probability is maximum is decided as the label for that sample.

Then once the class labels are predicted, the accuracy of the model on test data is computed using the testing class labels.

Task 2 – Feature combination by simple concatenation: For this task, the feature matrices are concatenated horizontally to form the data matrix. Each training feature matrix is of dimension 4786 x 1000 and there are three feature matrices in the training data, so the training data matrix is of the dimension 4786 x 3000. And each testing feature matrix is of dimension 1833 x 1000 and upon concatenation, the testing data matrix is of the dimension, 1833 x 3000. So, the SVM model is trained using the training data matrix of the size 4786 x 3000 and is tested using the testing data matrix of the size, 1833 x 3000 and the classification accuracy of the model is obtained.

RESULTS AND OBSERVATIONS

Task 0.1 – Classification by individual features: The classification accuracies obtained are as follows

- Classification accuracy on test X1: 10.7807%
- Classification accuracy on test X1: 16.6755%
- Classification accuracy on test X1: 8.92193%

```
Accuracy = 10.7807% (203/1883) (classification)
Accuracy = 16.6755% (314/1883) (classification)
Accuracy = 8.92193% (168/1883) (classification)
```

Task 0.2 - Classification by individual features with probability estimates: The classification accuracies obtained are as follows

- Classification accuracy on test X1: 27.8279%
- Classification accuracy on test X2: 27.7748%
- Classification accuracy on test X3:27.0313%

```
Accuracy = 27.8279% (524/1883) (classification)
Accuracy = 27.7748% (523/1883) (classification)
Accuracy = 27.0313% (509/1883) (classification)
```

Task 1 – Feature combination by fusion of classifiers: The classification accuracy obtained is as follows

- Classification accuracy: 0.44344 or 44.344%

```
Accuracy = 44.3441317047265% (835/1883)
```

Task 2 – Feature combination by simple concatenation: The classification accuracy obtained is as follows

- Classification accuracy: 37.0685%

```
Accuracy = 37.0685% (698/1883) (classification)
```

CONCLUSION

Libsvm package can be used to perform multi-class classification and regression. There are parameter options defined in the package to specify the kernel to be used, initial costs to be used, whether to compute probability estimates or not, type of SVM to be performed and to specify the number of support vectors and others.

Performing classification using SVM on individual matrices had very low classification accuracies which improved when probability estimates were used in during training and testing. Classification accuracy was the best when the probability estimates were combined and argmax was used to compute the class labels. Classification accuracy when the feature matrices were combined horizontally into a single data matrix was better than performing classification on individual features.

REFERENCES

- Libsvm: Chih-Chung Chang and Chih-Jen Lin
    - https://www.csie.ntu.edu.tw/~cjlin/libsvm/
    - https://github.com/cjlin1/libsvm/tree/master/python
- Paper: Chih-Chung Chang and Chih-Jen Lin, LIBSVM: A Library for Support Vector Machines. ACM Transaction on Intelligent Systems and Technology.