Extension of Binary SVM into Multi Class Recognition

Mrutyunjaya Lenka and Sandeep Gunda
Arizona State University

1. Introduction

Support vector machines (SVMs) are a set of supervised learning methods for classification, regression and outlier detection. The advantage of support vector machines lie in their effectiveness in high dimensional space. SVMs deliver state-of-the-art performance in real-world applications such as text categorization, hand-written character recognition, image classification, biosequences analysis, etc., and are now established as one of the standard tools for machine learning and data mining. In this project we implement a binary SVM for a given data set and extend the approach to multi class recognition using quadratic programming model. Section 2 outlines the phases in which the entire project was carried out. An analysis of the different multi class extension approaches and their results is presented in the following section.

2. Implementation

Step – 0: Binary SVM

- This step calculates the binary classifier parameters from the training data extracted from HearDataSet.mat.
- The primal cost function of the binary SVM was solved by the MATLAB quadratic programming function. One of the categories was labeled as 1 and the other was labeled as -1.
- The classifier was formed by the parameters obtained in the above step. The label of the test samples were calculated and compared to the observed labels.
- The accuracy obtained was 85%.
- The cost of misclassification was incremented from 1 to 50 in steps of 1. The above procedure was repeated for each increment of the cost.
  Instruction: Please run Step0.m to verify the results.

Step – 1: One-vs-all

- This step trains a multi class SVM for a 4 category Vehicle dataset which contains samples labeled as bus, saab, opel and van.
- This process starts by choosing one of the 4 categories and assigning the label corresponding to it as 1. The rest of the 3 categories are assigned a label value -1.
- The parameters weight and bias were calculated and thus by repeating the above process 4 classifiers were formed.

- The training samples were fed to the classifiers and the predicted labels were compared to the observed ones and correctly classified samples were counted for each of the classifier.
- The above process was repeated for a cost increment from 1 to 50 in steps of 1.
- The stable accuracy obtained was 80%.
  Instruction: Please run Step1.m to verify the results.

Step – 2: One-vs-one

- This step trains a multi class SVM for a 4 category Vehicle dataset which contains samples labeled as bus, saab, opel and van.
- The process starts by choosing a pair of categories. One of the chosen categories was labeled as 1 and the other was -1. The rest of the categories and their corresponding training samples were removed from the training data.
- The parameters weight and bias were calculated for each pair and thus by repeating the above process 6 classifiers were formed.
- The training samples were fed to the classifiers and the predicted labels were compared to the observed ones and correctly classified samples were counted for each of the classifier.
- The above process was repeated for a cost increment from 1 to 50 in steps of 1.
- The stable accuracy obtained was 84%.
  Instruction: Please run Step2.m to verify the results.

Step – 3: Error-correcting-coding:

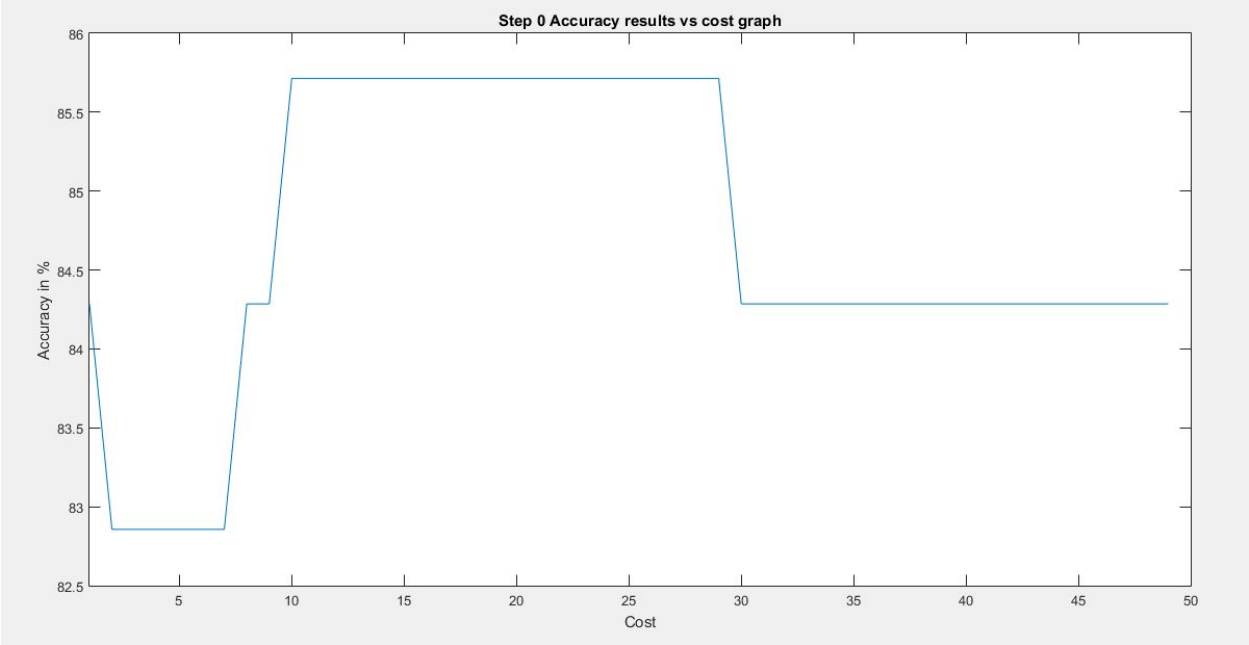
- This step trains a multi class SVM for a 4 category Vehicle dataset using error correcting coding.
- In this step we choose error correcting matrix mentioned in project description. There are 7 columns and each column defines which of categories should be 1,-1
- The parameters weight and bias were calculated for each pair and thus by repeating the above process 7 classifiers were formed.
- The training samples were fed to the classifiers and the predicted labels were compared to the observed ones using the minimum distance method and correctly classified samples were counted for each of the classifier.
- The above process was repeated for a cost increment from 1 to 50 in steps of 1.
- The stable accuracy obtained was 68%.
  Instruction: Please run Step3.m to verify the results.

  Note: Please divide the training data and test data files into separate folders. Give the full path of the data folders when prompted during execution of each of the above steps.

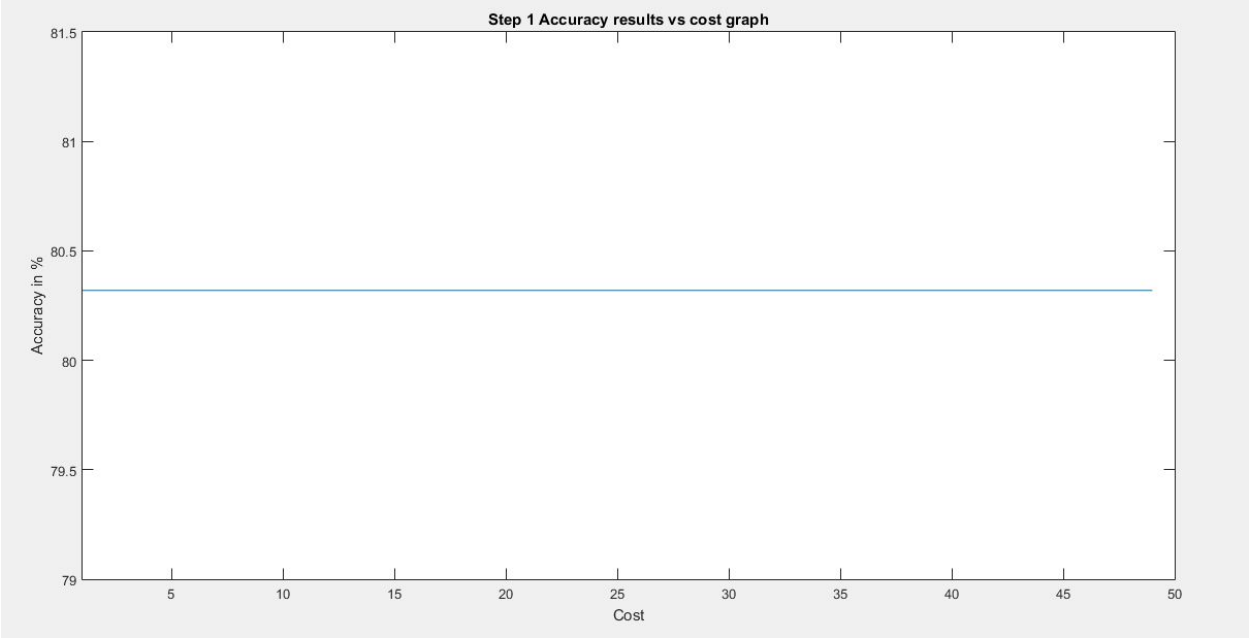
3. Results and Analysis

Below are graphs of accuracy vs cost values.

Graph for Step 0:

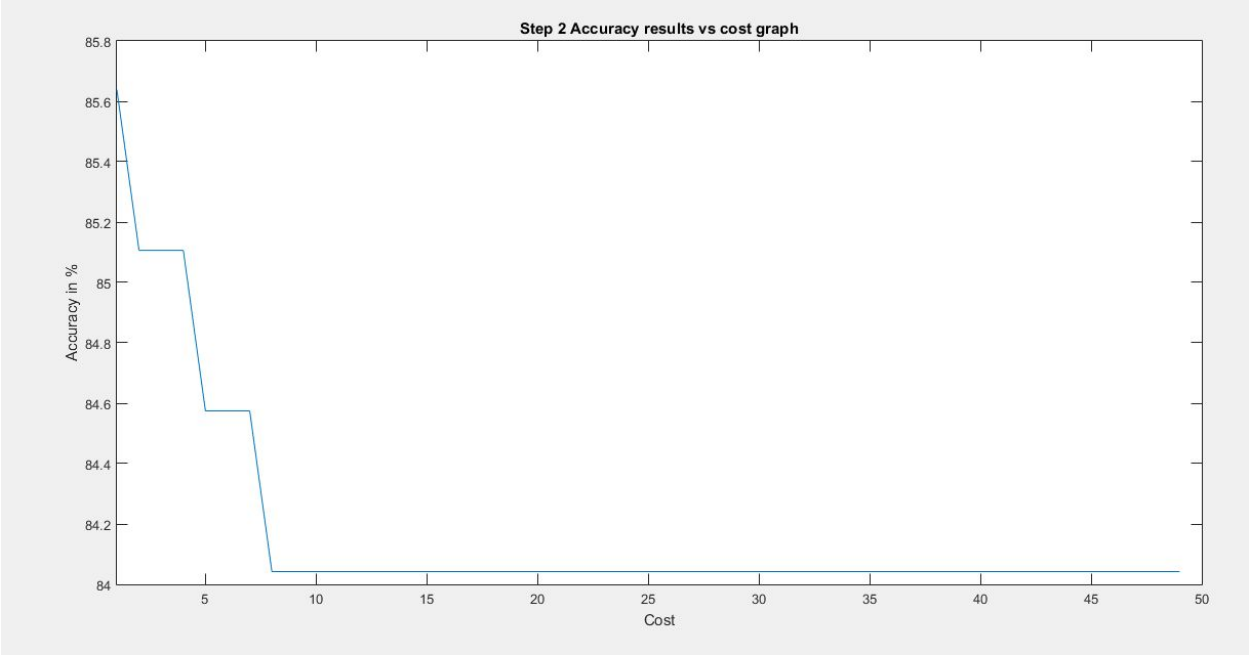


Graph for Step 1:

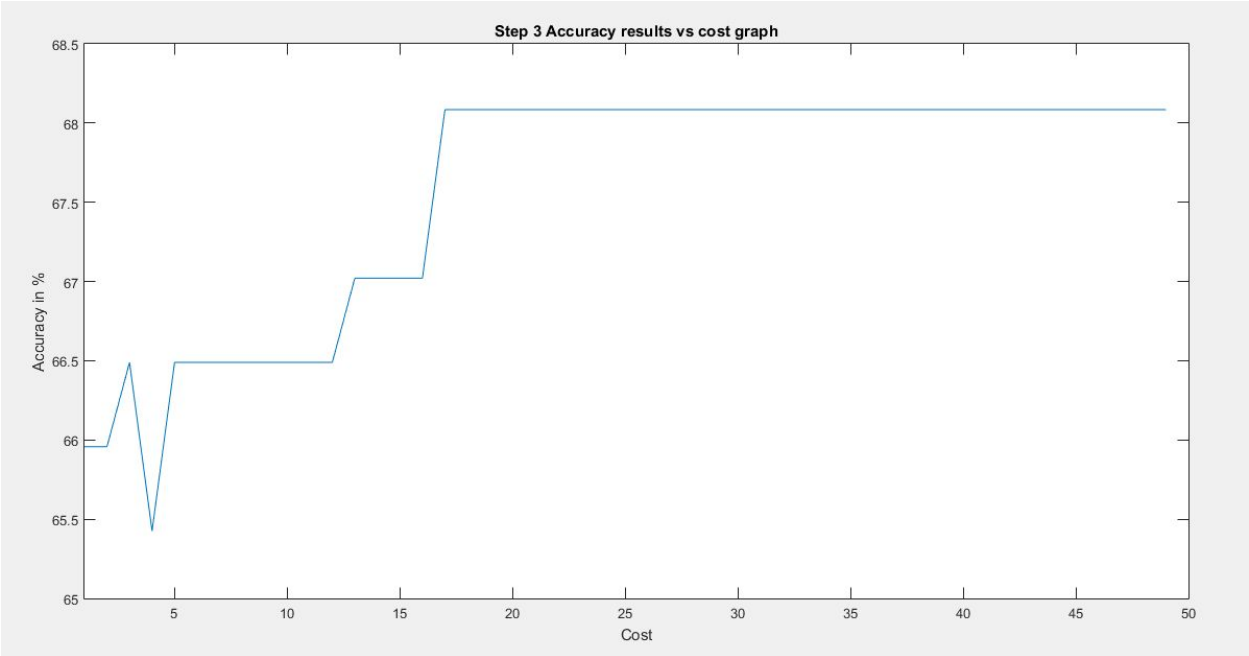


For Step 1 the no significant difference in accuracy is observed w.r.t the cost value

Graph for Step 2:



Graph for Step 3:



Step Number	Samples correctly classified / Total Samples	Accuracy
0	**59 /70**	**84.2 %**
1	**151/188**	**80.9 %**
2	**158/188**	**84.04 %**
3	**128/188**	**68.08%**

4. Observations:

- SVM is a powerful classifier which classified above 80% of data in almost all cases.
- The one-vs-one extension for multi class recognition gives the best result among all the methods. But the space and time requirement is higher compared to others.

5. Contributions:

Sandeep Gunda: Implemented step 1 and step 3.
Mrutyunjaya Lenka: Implemented step 0 and step 2.

6. References:

[1] S. Theodoridis and K. Koutroumbas, Pattern Recognition (4th Edition), Academic Press, 2008.

[2] K. Crammer and Y. Singer. On the Algorithmic Implementation of Multi-class SVMs, JMLR, 2001