Support Vector Machine for Alphabet recognition

Table of Contents

Name: Ventrapragada Sai Shravani

PRN:17070123120

Batch: Entc(2017-21) G-3

Aim:

Implementation of Support Vector Machine for Alphabet recognition.

Theory:

The worth of a classifier is not in how well it seperates the training data.

SVMS try to find all such data that correctly classify the training data and among all such lines pick the one that has the greatest distance to the points closest to it.

For data that isn't linearly sepereable we can project data to a space where it is almost linealy seperable.

An important aspect of SVMs is that all the mathematical machinery that it uses, the exact projection of the number of dimensions doesn't show up. You can write all of it in terms of dot products between various data points represented as vectors.

```
clc;
clear all;
close all;
```

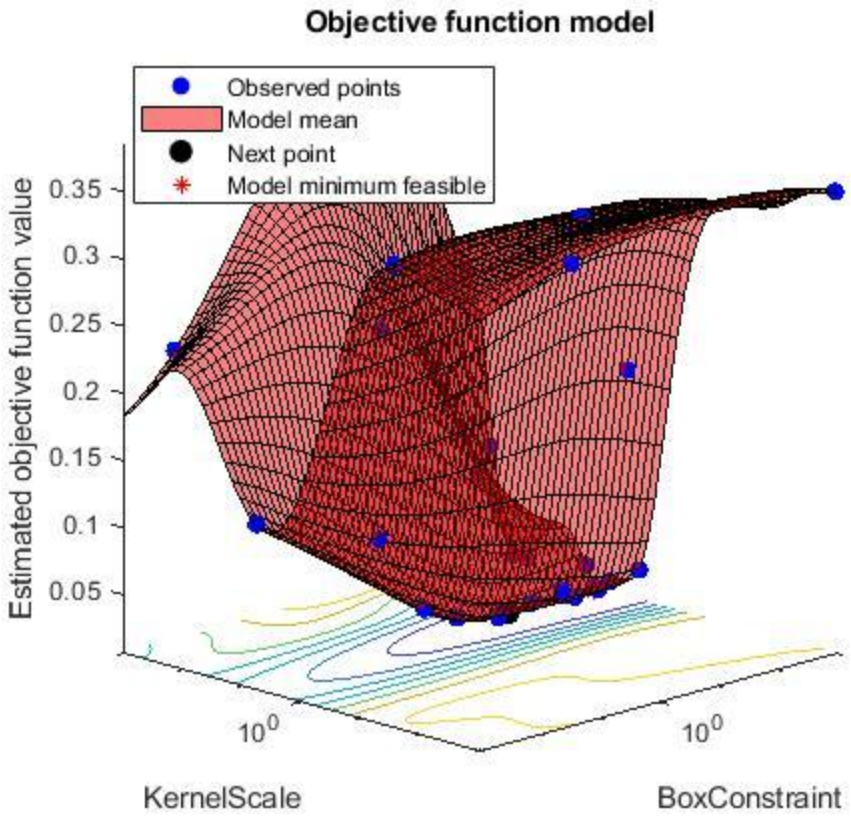
SVM for Alphabet recognition

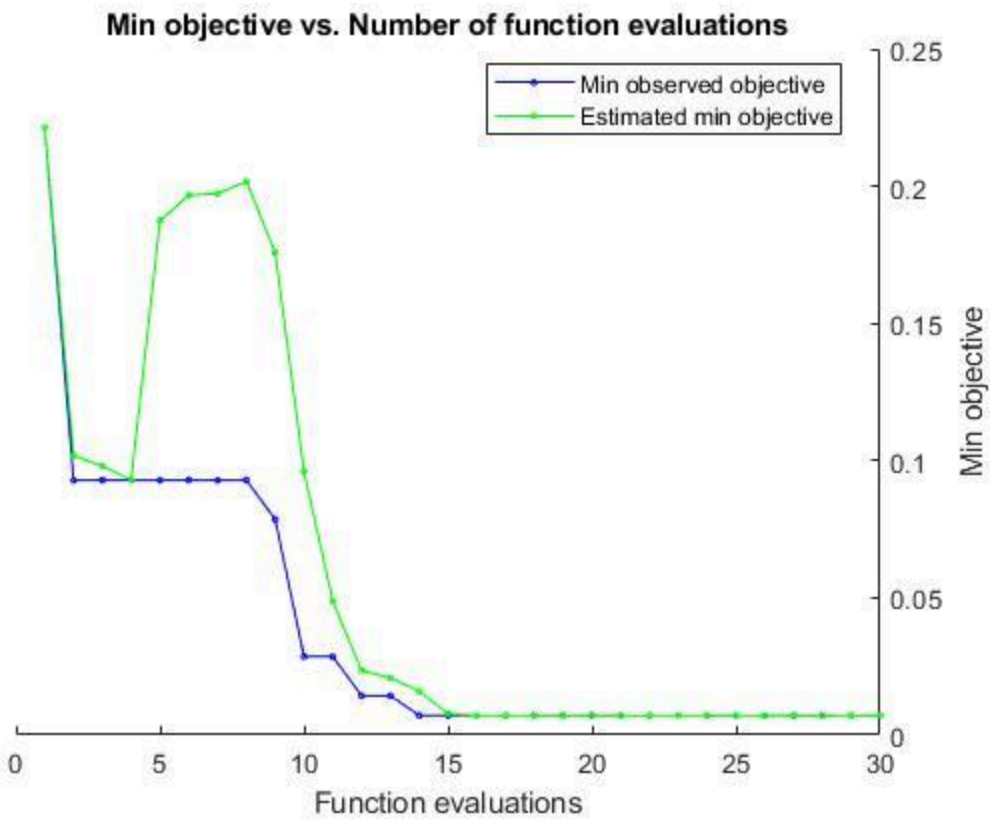
```
X = readmatrix('Trainingcsv2.csv'); %Input 24 features of A
Y = readmatrix('Character.csv'); %Alphabet a and b character as target
svmModel =
 fitcsvm(X,Y,'Standardize',true,'KernelFunction','RBF','OptimizeHyperparameters','
improvement-plus'))

|
=====================================================================
```

```
| Iter | Eval   | Objective   | Objective   | BestSoFar   | BestSoFar
  | BoxConstraint|  KernelScale |
|      | result |             | runtime     | (observed)  | (estim.)
  |              |             |
|
=========================================================================================
|    1 | Best   |     0.22143 |      1.1986 |     0.22143 |
 0.22143 |       1.7209 |       81.323 |
|    2 | Best   |    0.092857 |      0.3154 |    0.092857 |
 0.10179 |       808.19 |       552.18 |
|    3 | Accept |        0.35 |     0.14193 |    0.092857 |
 0.097978 |       1.2004 |     0.023942 |
|    4 | Accept |     0.34286 |     0.11183 |    0.092857 |
 0.09289 |    0.0021476 |     0.059109 |
|    5 | Accept |     0.22143 |      0.3469 |    0.092857 |
 0.18739 |       608.72 |       999.08 |
|    6 | Accept |     0.22143 |    0.084021 |    0.092857 |
 0.19672 |       188.76 |       663.28 |
|    7 | Accept |        0.35 |    0.076315 |    0.092857 |
 0.19727 |     0.034167 |    0.0010008 |
|    8 | Accept |     0.22143 |    0.070651 |    0.092857 |
 0.20175 |    0.0073719 |       999.83 |
|    9 | Best   |    0.078571 |    0.082656 |    0.078571 |
 0.17584 |       289.78 |       287.94 |
|   10 | Best   |    0.028571 |     0.15683 |    0.028571 |
 0.095979 |       988.14 |       2.1247 |
|   11 | Accept |    0.028571 |     0.11534 |    0.028571 |
 0.048694 |       997.61 |       2.0664 |
|   12 | Best   |    0.014286 |     0.14155 |    0.014286 |
 0.023486 |       991.95 |       6.4437 |
|   13 | Accept |    0.021429 |    0.077597 |    0.014286 |
 0.020814 |       974.01 |       15.875 |
|   14 | Best   |   0.0071429 |    0.099031 |   0.0071429 |
 0.015983 |       978.15 |       11.135 |
|   15 | Accept |    0.021429 |    0.057766 |   0.0071429 |
 0.0077146 |        963.9 |       16.657 |
|   16 | Accept |   0.0071429 |     0.08571 |   0.0071429 |
 0.007125 |       838.46 |       8.7414 |
|   17 | Accept |    0.014286 |    0.091355 |   0.0071429 |
 0.0071273 |        996.7 |       159.08 |
|   18 | Accept |    0.028571 |    0.061611 |   0.0071429 |
 0.0070435 |       53.357 |       2.3115 |
|   19 | Accept |     0.19286 |    0.072752 |   0.0071429 |
 0.0070173 |       183.85 |      0.61875 |
|   20 | Accept |    0.014286 |    0.066321 |   0.0071429 |
 0.0070022 |       48.434 |       6.2679 |
|
=========================================================================================
| Iter | Eval   | Objective   | Objective   | BestSoFar   | BestSoFar
  | BoxConstraint|  KernelScale |
|      | result |             | runtime     | (observed)  | (estim.)
  |              |             |
|
=========================================================================================
```

```
|   21 | Accept |    0.014286 |    0.079586 |   0.0071429 |
 0.0070577 |       159.83 |       4.0769 |
|   22 | Accept |    0.035714 |    0.072777 |   0.0071429 |
 0.0071147 |      0.48548 |        4.261 |
|   23 | Accept |     0.12857 |    0.061395 |   0.0071429 |
 0.0071155 |    0.0010532 |       6.0815 |
|   24 | Accept |    0.014286 |    0.065892 |   0.0071429 |
 0.007154 |        8.426 |       4.0955 |
|   25 | Accept |    0.014286 |    0.085979 |   0.0071429 |
 0.007165 |       975.26 |       272.75 |
|   26 | Accept |   0.0071429 |    0.083459 |   0.0071429 |
 0.0070173 |       282.69 |       10.463 |
|   27 | Accept |     0.10714 |    0.060396 |   0.0071429 |
 0.0070226 |     0.030278 |       1.5097 |
|   28 | Accept |        0.35 |    0.063923 |   0.0071429 |
 0.0070227 |       990.56 |    0.0010525 |
|   29 | Accept |    0.014286 |    0.080514 |   0.0071429 |
 0.0071064 |       3.6558 |       8.9183 |
|   30 | Accept |   0.0071429 |    0.099756 |   0.0071429 |
 0.0072609 |       29.316 |       12.425 |


__________________________________________________________
Optimization completed.
MaxObjectiveEvaluations of 30 reached.
Total function evaluations: 30
Total elapsed time: 44.5987 seconds.
Total objective function evaluation time: 4.2078

Best observed feasible point:
    BoxConstraint    KernelScale
    _____________    ___________

       978.15          11.135

Observed objective function value = 0.0071429
Estimated objective function value = 0.0076405
Function evaluation time = 0.099031

Best estimated feasible point (according to models):
    BoxConstraint    KernelScale
    _____________    ___________

       838.46          8.7414

Estimated objective function value = 0.0072609
Estimated function evaluation time = 0.10061


svmModel =

  ClassificationSVM
                       ResponseName: 'Y'
              CategoricalPredictors: []
                         ClassNames: [0 1]
```

```
                    ScoreTransform: 'none'
                   NumObservations: 140
HyperparameterOptimizationResults: [1×1 BayesianOptimization]
                             Alpha: [32×1 double]
                              Bias: -1.1247
                   KernelParameters: [1×1 struct]
                                Mu: [1×24 double]
                             Sigma: [1×24 double]
                    BoxConstraints: [140×1 double]
                   ConvergenceInfo: [1×1 struct]
                   IsSupportVector: [140×1 logical]
                            Solver: 'SMO'
```





Conclusion:

In the above experiment we implement a MATLAB code for Support vector machine for clustering alphabet. I learnt Kernel function of Radial basis function Because support vector machines employing the kernel trick do not scale well to large numbers of training samples or large numbers of features in the input space, several approximations to the RBF kernel (and similar kernels) have been introduced.

We used fitcsvm to perform the SVM function, and also plotted the scatter plot using gscatter function. We used legend and axis and a few marker effects for better presentation.

Published with MATLAB® R2020a