SMAI PROJECT REPORT

Team No : 44

Team Name : Random\_team\_1

Project ID : 26

Project Title : Implement MULTICLASS SVM WITH DIFFERENT KERNELS FROM SCRATCH

Team Members :

1.2018801010 - Karnati Venkata Kartheek

2.2018900061 - Shashikant Ghangare

GitHub Link:

<https://github.com/shashikant-ghangare/SMAI-Spring2019-Project>

Introduction (Problem statement, Motivation, Overview [Input - Method - Output] diagram wherever appropriate)

Approach (es) considered

Work Performed (Experiments, software programming, etc.)

**Introduction**

Multiclass SVM is usually implemented using One vs One or One vs Rest fashion, here in this work an alternative approach is used, optimization formulation is such that at a go it decides the scores for all the classes and assigns highest score class to the test sample.

**Problem statement**

In this project, we implement simultaneous multiclass SVM with different kernels using CVXOPT and compare their performance (space, time requirements and accuracy) with SCIKIT Learn implementation.

**Overview**

**Math behind the SVM and multiclass SVM**

In Soft margin binary SVM

The optimization function is

min L = ½ w’w + C ∑in ξi

such that yif(**x**i) ≥ 1 - ξi, for all i(1 to n)

ξi ≥ 0

Here C is the hyperparameter and ξI is the slack variable associated with ith  point.

In the Soft Margin, Multiclass SVM

The optimization function is

min L = ½ ∑t wt’wt + C ∑i ∑m(≠yi) ξim

such that for ever point i(1 to n),

Wyi’ **x**i - Wm’ **x**i ≥ 1 - ξim, for m(≠yi) = 1,2 ..k

Here C is the hyperparameter and ξimis the slack variable associated with ith  point and mth class.

n is the number of points in the given data and k is the number of classes present in the data.

Here **x**i is a **d+1** dimensional vector, Here each point xi is appended with 1 as new attribute to account for intercept. and wk is a **d+1** dimensional vector. the vectors Wk are concatenated to become a single vector

W =[ w1 w2 w3 …… wk]

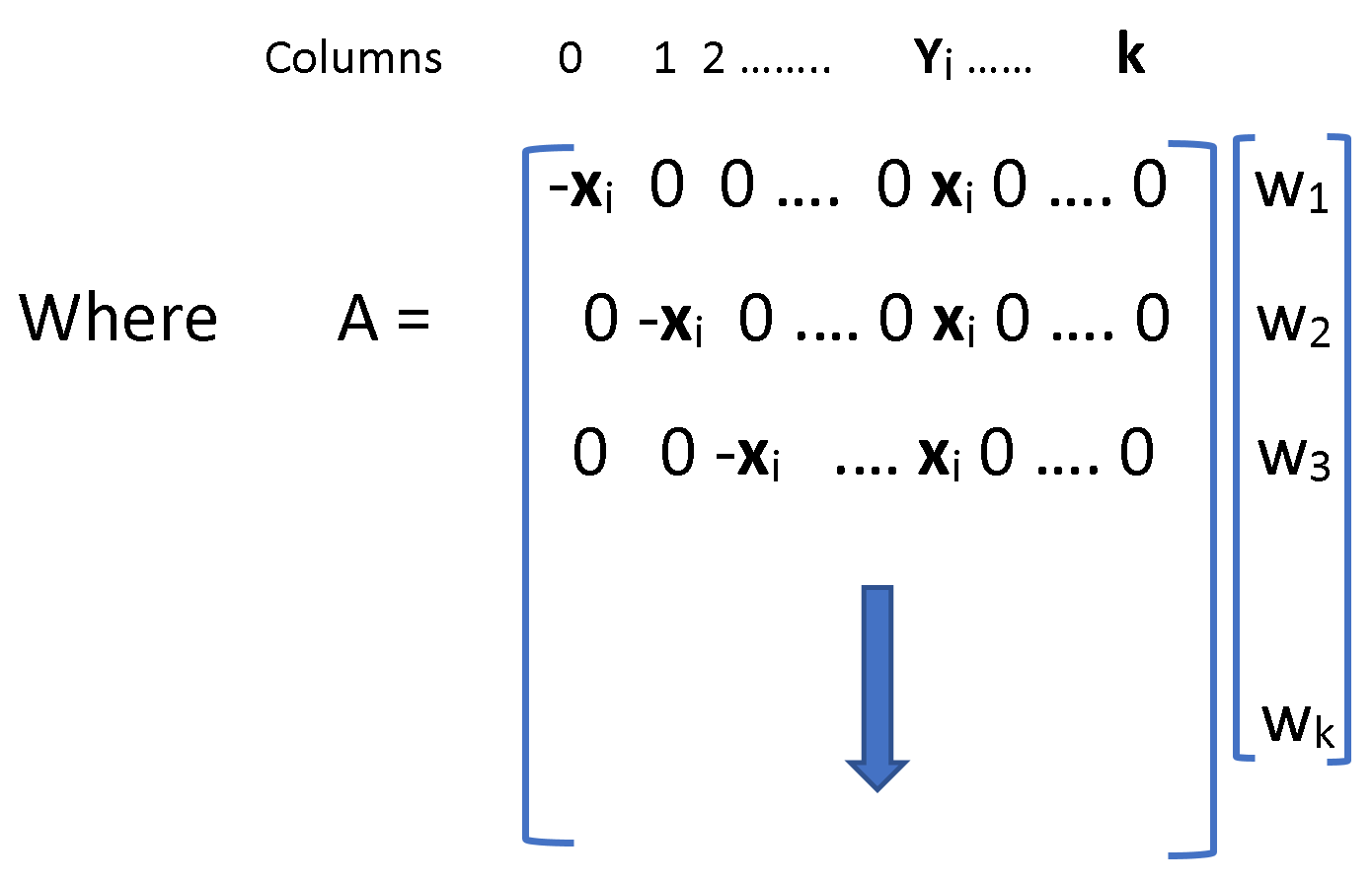
Thus, W is of dimension **(d+1) \*k**

So, the vectorized optimization function with respect to concatenated W is

L = ½ W’W + C 1’ ξ

The constraints Wyi’ **x**i - Wm’ **x**i ≥ 1 - ξim, for m(≠yi) = 1,2 ..k are converted into the form

A\*W ≥ 1- ξ



Here the Dimensions of A = [n(k-1),k(d+1)]

Because for each point there a k-1 constraints and thus n(k-1)

In each row, there are k vectors of dimensions (d+1) thus dimensions k(d+1).

We can form the Legrangian

L = ½ W’W + C 1’ ξ - α’ (A\*W-(1- ξ)) – β‘ ξ

L = ½ W’W + C 1’ ξ - α’ (A\*W) + α’1- α’ξ – β‘ ξ

Differentiating w.r.to W = 0

½ 2W’ = α’A

W = A’ α ------ 1

Differentiating w.r.to ξ = 0

C \*1’ - α’ – β’ = 0

C \*1’ = α’ + β’ ------ 2

i.e 0 ≤ α’≤ C

Substituting 1 , 2 back into legrangian, we get

L = ½ W’W + C 1’ ξ - α’ (A\*W) + α’1- α’ξ – β‘ ξ

L = ½ α’ A’ A α + (α’ + β’) ξ - α’ A A’ α + α’1- α’ξ – β‘ ξ

Maximize L = α’1- ½ α’ A’ A α

Or

Minimize L = ½ α’ A’ A α – 1’α

Such that 0 ≤ α’≤ C or α’≤ C and -α’≤ 0

This is the dual form of the optimization problem

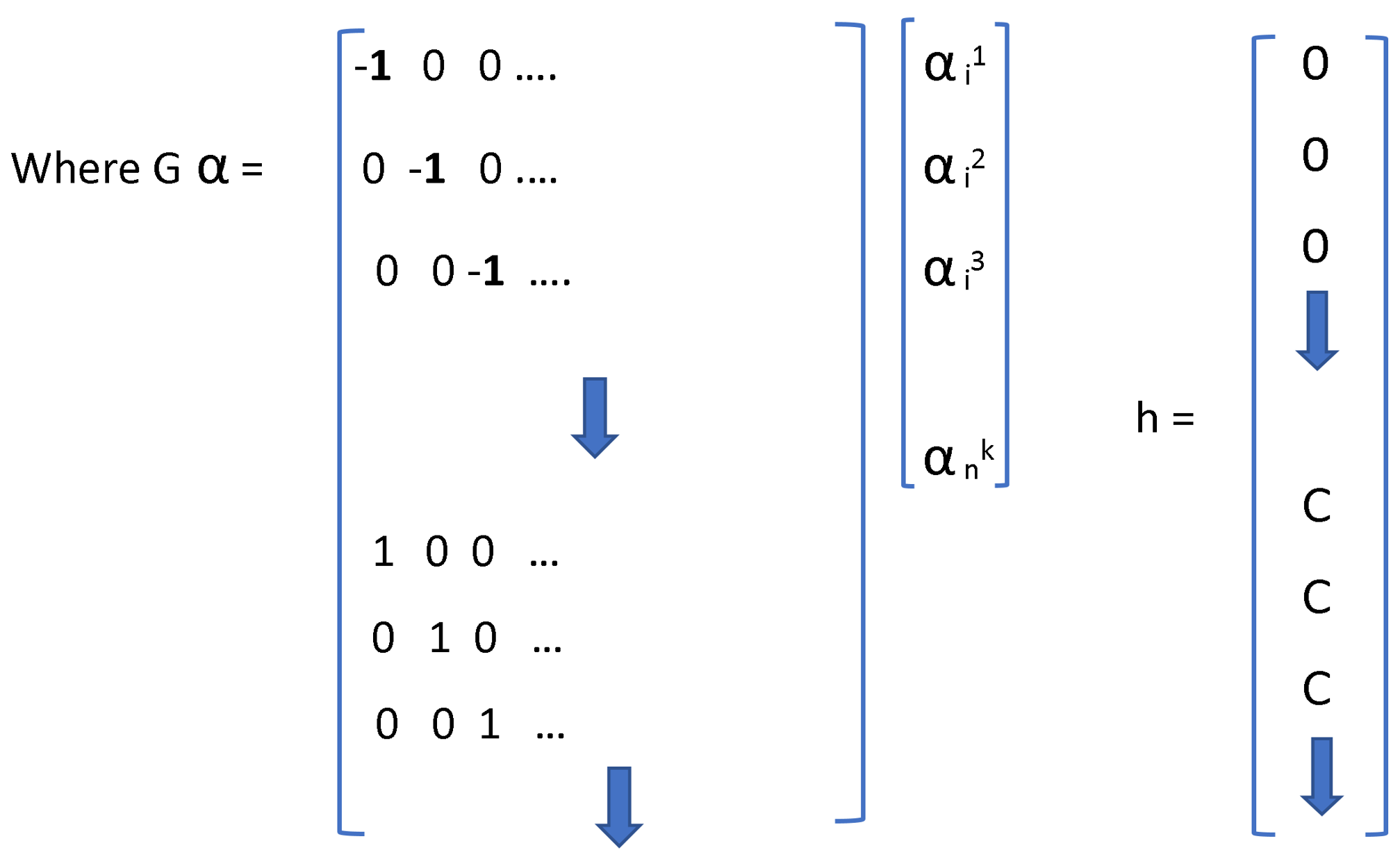
Optimization format accepted by CVXOPT is

Minimize ½ α’P α+ q’ α

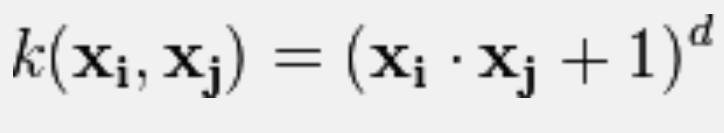
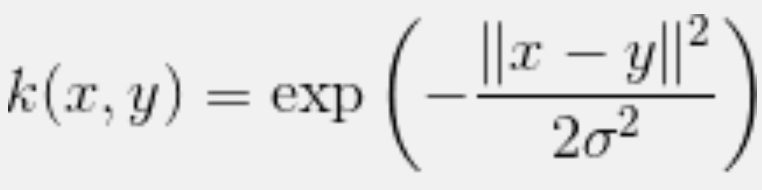
Subject to G α ≤ h

So in order to solve using CVXOPT

Here P = A’ A q = [-1 -1 -1 ….]



For the linear kernel just the product A’ A is enough but for the polynomial kernel and rbf kernel every dot product xi’xj involved in A’A calculation is replaced respectively by

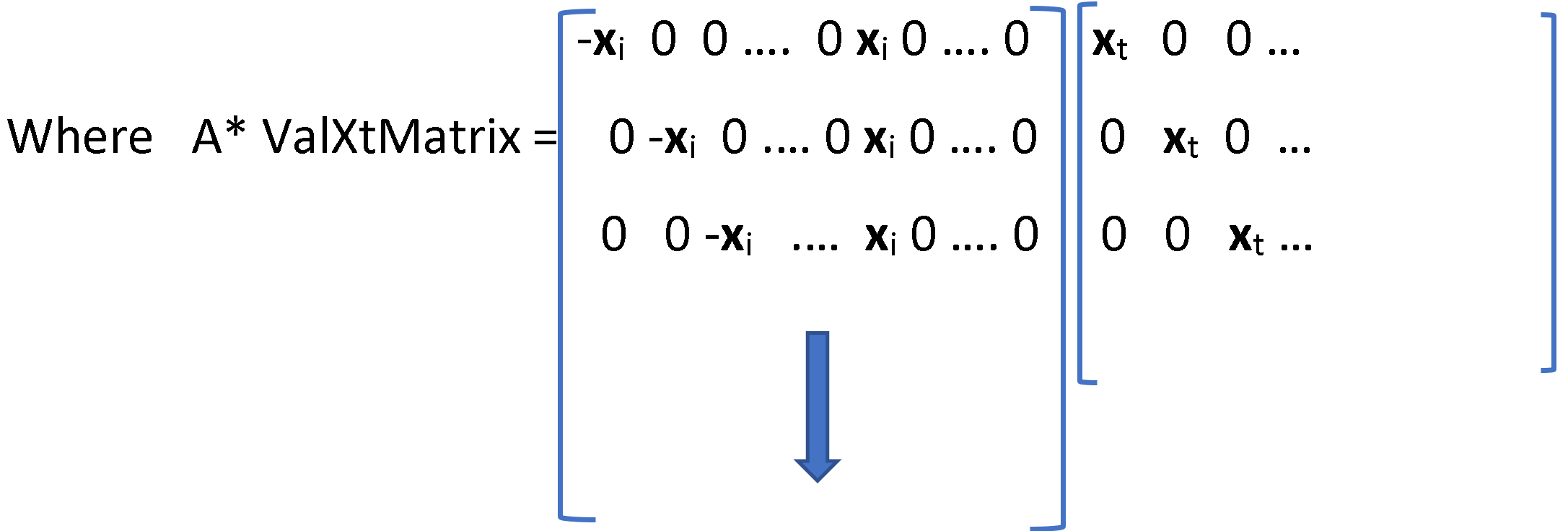
To predict the label for point **x**t we calculate

argmax ([W1’ **x**t , W2’ **x**t , …….., Wk’ **x**t])

[W1’ **x**t, W2’ **x**t , …….., Wk’ **x**t] = W’\*ValXtMatrix

= α’A\*ValXtMatrix

= α’kernel(A\*ValXtMatrix)



Where **x**i ‘s in the first matrix are the row vectors and **x**t‘s in the second matrix are the column vectors

**Results**

**Visualization on some datasets, different kernels**

**Cross Validated Error rates comparison for 3 classifiers on different datasets**

Iris Dataset

Wine Dataset

Cancer Dataset

**Comparison w.r.to Training Time taken by 3 classifiers**

**Comparison w.r.to memory requirements of 3 classifiers**

**Comparison of Time taken while increasing number of points and number of classes**

**Failure cases**

When the number of classes and number of data points increase the matrices involved in optimization are of dimensions [n(k-1),k(d+1)], [n(k-1), n(k-1)] thus quickly scale up with n\*k and also takes time to constructed kernelized product of those matrices.

Discussion (Analysis of results, including failure cases, Future Directions)

A "**task assignment**" page detailing the tasks performed for this project, and each team member's percentage contribution to each task.

Acknowledge all codes that you obtained elsewhere

Columns 0 1 2 …….. **Y**i …… **k**

-**x**i  0 0 …. 0 **x**i 0 …. 0 **x**t  0 0 …

Where A\* ValXtMatrix = 0 -**x**i 0 .… 0 **x**i 0 …. 0 0 **x**t 0 …

0 0 -**x**i .… **x**i 0 …. 0 0 0 **x**t …

Columns 0 1 2 …….. **Y**i …… **k**

-**x**i  0 0 …. 0 **x**i 0 …. 0 w1

Where A = 0 -**x**i 0 .… 0 **x**i 0 …. 0 w2

0 0 -**x**i .… **x**i 0 …. 0 w3

wk

-**1** 0 0 …. α i1

Where G α = 0 -**1** 0 .… α i2

0 0 -**1** …. α i3

α nk

1. 0 0 …

0 1 0 …

0 0 1 …

Dual form of the changed optimization function is

0

0

0

h =

C

C

C

SIMPLE BINARY SVM (Implementation and results)

ONE VS ONE(OVA) AND ONE VS REST (OVR) SVM (Implementation and results)

SCIKIT LEARN IMPLEMENTATION OF ABOVE TWO (Implementation and results)

SIMULTANEOUS MULTICLASS SVM (Implementation and results)

TIME AND SPACE COMPLEXITY COMPARISION OF ALL METHODS

IMPLEMENTING KERNELS IN BOTH SIMPLE BINARY SVM AND MULTICLASS SVM

TRYING DIFFERENT KERNELS TO DIFFERENT NON - LINEARLY SEPERABLE DATA

A LIBRARY OF ALL THESE

Introduction (Problem statement, Motivation, Overview [Input - Method - Output] diagram wherever appropriate)

Approach (es) considered

Work Performed (Experiments, software programming, etc.)

Initially Binary SVM was implemented but later direct Multiclass SVM was implemented

Results

Successes

Failure cases

Discussion (Analysis of results, including failure cases, Future Directions)

Github Link

A "**task assignment**" page detailing the tasks performed for this project, and each team member's percentage contribution to each task.

Acknowledge all codes that you obtained elsewhere

x) = Xn i=1 αiφ(xi) Tφ(x) + b.

min L = ½ ∑k wk’wk + ∑ α k ( yk (w’xk + b) + sk -1) + α ∑ skPut ovo , ovr and smcsmv figure using some data

min L = ½ ∑k wk’wk + ∑ α k ( yk (w’xk + b) + sk -1) + α ∑ sk\\

Approach (es) considered

Simultaneous multiclass SVM

Work Performed (Experiments, software programming, etc.)

Implementing dimensionality reduction function

Implementing Binary SVM using cvxoptImplement(ovo,ovr and multiclass SVM using CVXOPT)Comparing them to scikit learn implementation

Results

Time and space requirements for the

Successes

Failure cases

Discussion (Analysis of results, including failure cases, Future Directions)

ask Assignment

tasks performed for this project

each team member's percentage contribution to each task

codes that were obtained elsewhere