Homework 3

Subhankar Ghosh

Question 1

(a)

We will fit a *Linear*, *Polynomial* and a *Radial* kernel SVM on our digits data from ElemStatLearn package.

Five fold cross validation has been used to pick the best model

- Linear Kernel : c = 0.0078125 gives the best training accuracy of 99.07%
- Polynomial Kernel : The model with degree = 2, scale = 0.01 and c = 0.5 gives the best training accuracy of 99.53%
- Radial kernel : The model with sigma = 0.125 and c = 2 gives the best training accuracy of 79.09%

(b)

Performance of different kernel SVM

Model_Name	Test_Accuracy
Linear SVM	0.9529412
Poly SVM	0.9500000
Radial SVM	0.1666667

(c)

Advantages of kernels used: - *Linear:* Faster to train than other kernels. Performs especially well when the seperation boundary is linear. Works very well with multiple features. Performs very well when number of features is large. - *Polynomial:* WOrks very well when some polynomial space of the features have a linear seperation boundary. Allows learning of non linear data models. Helps take care of interaction features as well. - *Radial:* Performs very well with non linear data, and when number of features is not very high. It can project features space into a higher dimensional feature space.

Disadvantages of kernels used: - *Linear:* Cannot be used if the decision boundary is non linear which is mostly the case in real life data. - *Polynomial:* Can be very slow as the degree of the polynomial increases. Can suffer from the problem of numerical instability. - *Radial:* Can be very slow and does not perform very well if the number of features in high.

Is cross-validation effective on this analysis? Is there any potential pitfall that you could think of?

Cross validation is effective in tuning the best parameter. But one potential pitfall can be the amount of time it takes to build each model and validate each of the many models it has to train.

Question 2

(a) My own linear discriminate analysis (LDA) code gave an accuracy of **0.82** on the test data.

The LDA from the MASS package also gave an accuracy of **0.82** on the test data.

(b) My own linear discriminate analysis (LDA) code gave an accuracy of **0.835** on the test data with an $\alpha = 0.05$.

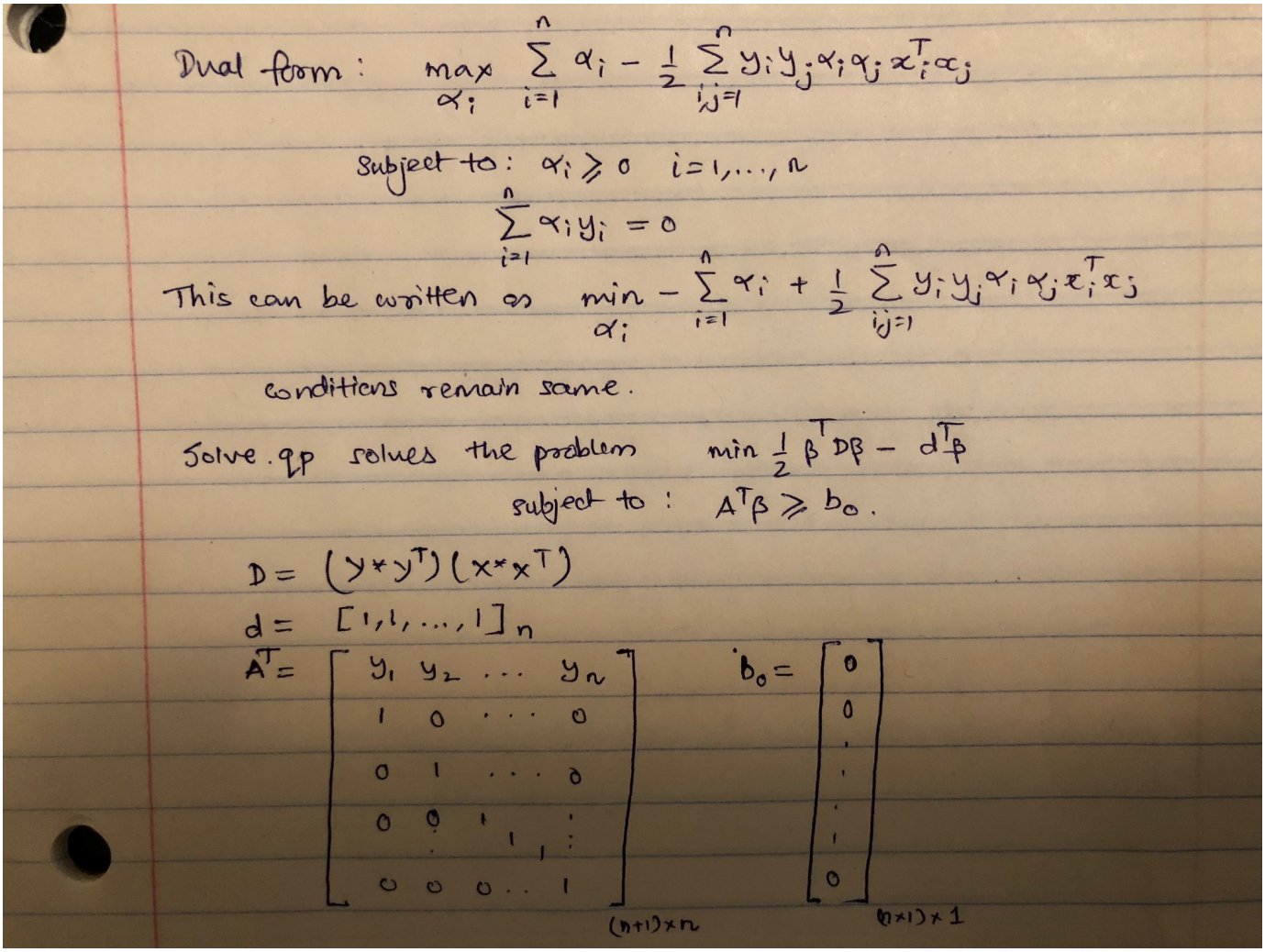
(c) Regularized QDA has an advantage over LDA because in this case we do not need to study the structure of the data in order to decide if we should apply LDA or QDA. It automatically decides the appropriate value of
α thus deciding whether a reduction to a particular form (LDA or QDA) is required or not.

QDA is unlikely to produce satisfactory results unless the ratio of the class sample sizes is large relative to the number of variables posed in the problem, and it has no advantages compared to LDA except when the class covariance matrices are quite different. In situations where the class covariance matrices are similar, LDA is likely to give improved results compared to QDA, because a single covariance matrix can be employed in the classification rule, thus reducing the number of estimates which need to be computed. LDA should therefore be able to produce better results, when the sample sizes are smaller.

Regularized QDA always gives results equivalent to or better than LDA and QDA since it automatically decides the appropriate value of
α thus deciding whether a reduction to a particular form (LDA or QDA) is required or not. The only downside to Regularized QDA is that it takes a lot of time.

Question 3

(a)



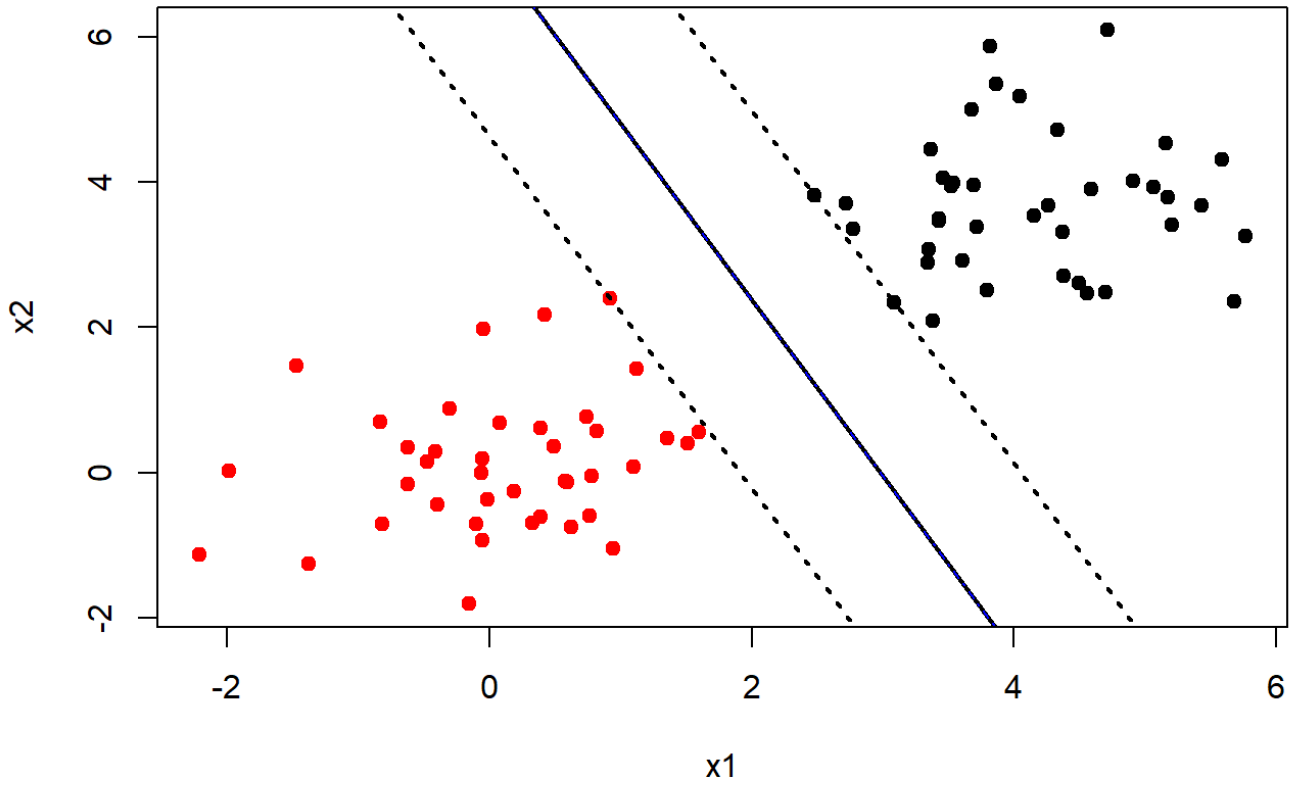
Formulation of Dual form

```
## [1] "Coefficients from my SVM"
```

```
## [1] "Beta0 = 2.78237298573782  Beta values = -0.933425983836912, -0.384981990665
094"
```

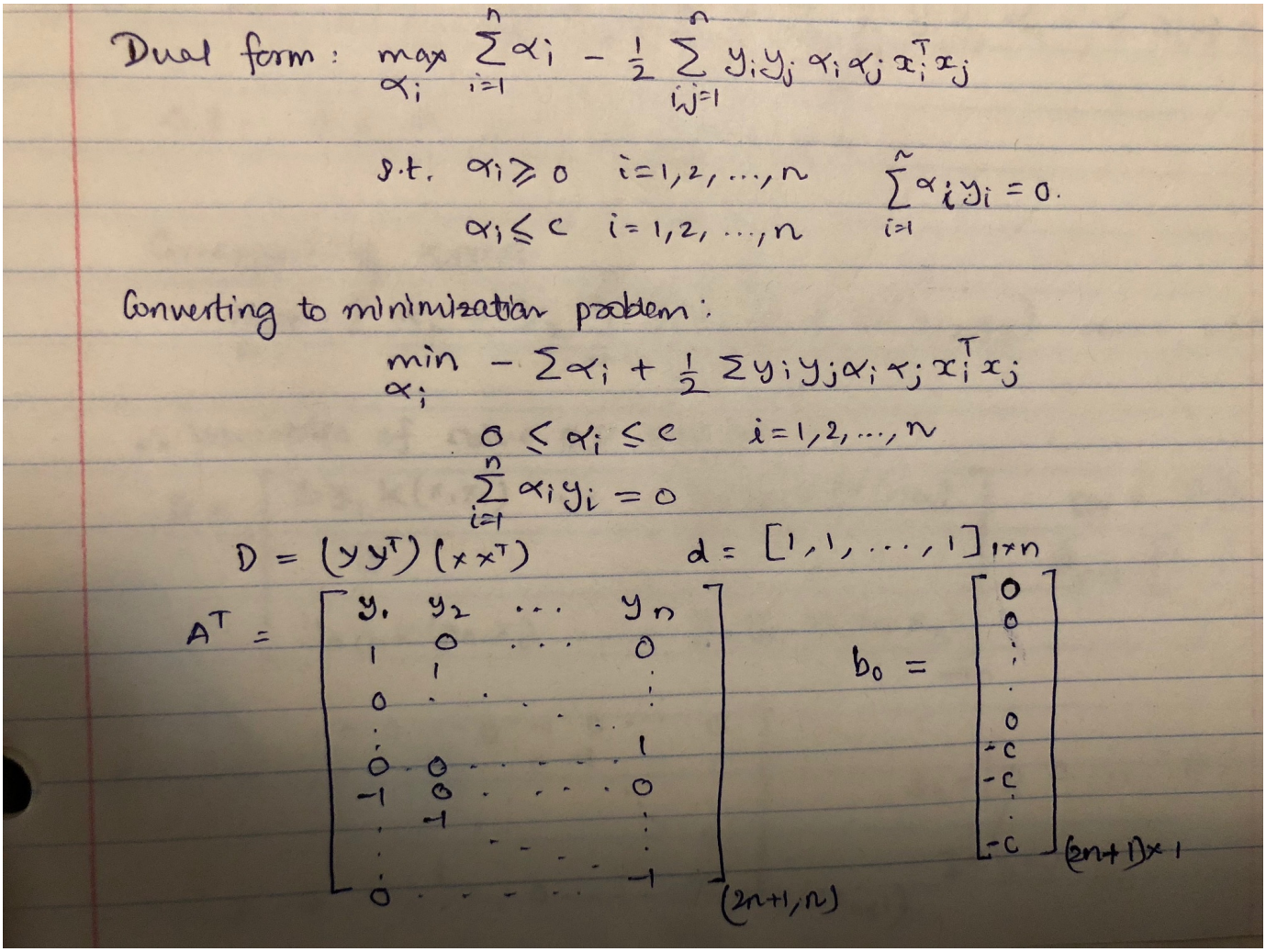
```
## [1] "Coefficients from package SVM"
```

```
## [1] "Beta0 = 2.78195312036252  Beta values = -0.933873966286481, -0.384495654907
918"
```



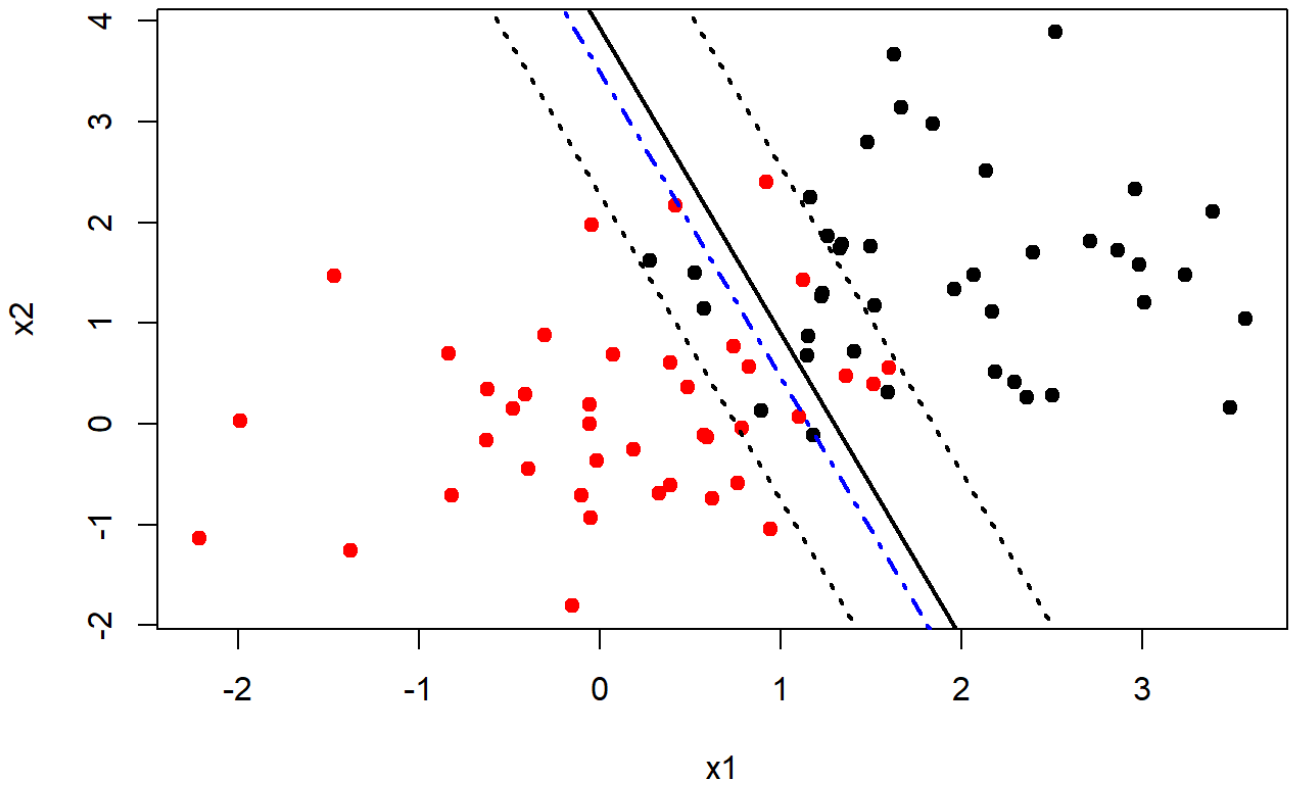
Comparison plot of my svm vs package. Blue:SVM Package, Black: My LDA

(b)



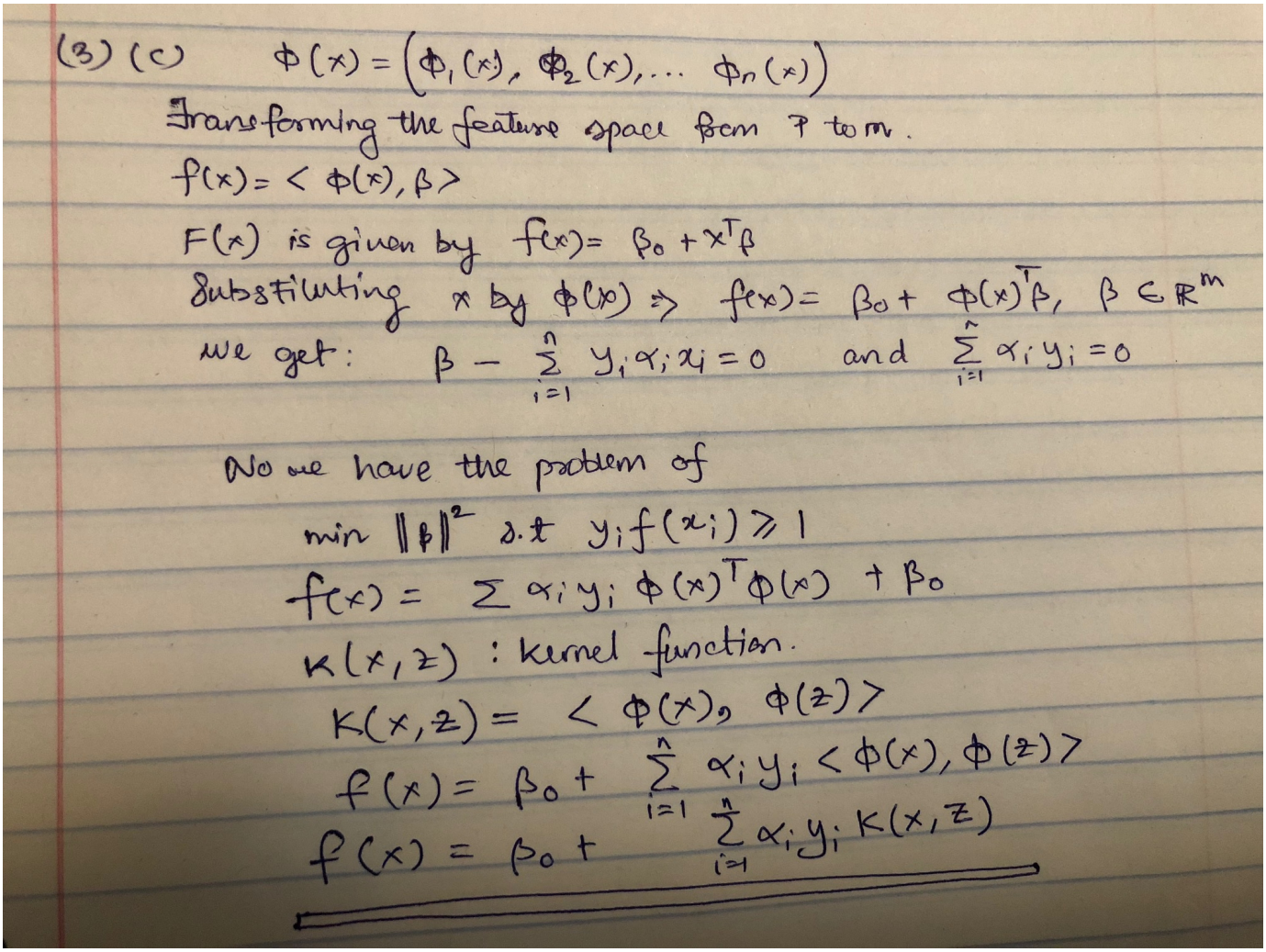
Formulation of dual form

Plot showing decision boundary we got from my SVM with that we get from the package



Comparison plot of my svm vs package. Blue:SVM Package, Black: My Regularized QDA

(c)



Formulation of new decision function

(d)

(D) Dual form: $\max_{\alpha_i} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \phi(x_i) \phi(x_j) \rangle$

s.t. $0 \leq \alpha_i \leq C$, $\sum_{i=1}^{n} \alpha_i y_i = 0$

Corresponding kernel

$\max_{\alpha_i} \sum \alpha_i - \frac{1}{2} \sum_{i=1}^{n} y_i y_j \alpha_i \alpha_j K(y_i, y_j)$ same constraint as above

∴ Variables of solve. QP will be :-

$$D = \begin{bmatrix} y_1 y_1 K(x_1, x_1) & \cdots & y_1 y_n K(x_1, x_n) \\ \vdots & & \\ y_n y_1 K(x_n, x_1) & \cdots & y_n y_n K(x_n, x_n) \end{bmatrix}_{n \times n}$$

$D_{ij} = y_i y_j K(x_i x_j)$

$d_{n \times 1} = [1 \ldots . 1]^T_{1 \times n}$.

$$A = \begin{bmatrix} y_1 & 1 & 0 & 0 \cdots 0 & -1 & 0 \cdots 0 \\ y_2 & 0 & 1 & & 0 & -1 & & \vdots \\ \vdots & \vdots & & \ddots & & & \ddots & 0 \\ y_n & 0 \ldots & & 1 & 0 \ldots & & & -1 \end{bmatrix}_{n \times (R_{n+1})}$$

$b_0 = [0 \ 0 \ldots . \ 0 \ -C \cdots -C]$

$\sum \alpha_i y_i = 0$

Processing math: 100% problem