**Introduction to Pattern Recognition CSE 4/555**

**Assignment 3**

**Submitted By: Abhinav Kumar**

**Ub Person #: 50247660**

Q1. (2 points) Write code to train a multi-class support vector classifier with dot-product kernel and 1-norm soft margin using the MNIST training data set. Then reporting the performance using MNIST test data set. There is a hyper-parameter that sets the trade-off between the margin and the training error --- tune this hyper-parameter through cross-validation.

Solution 1:

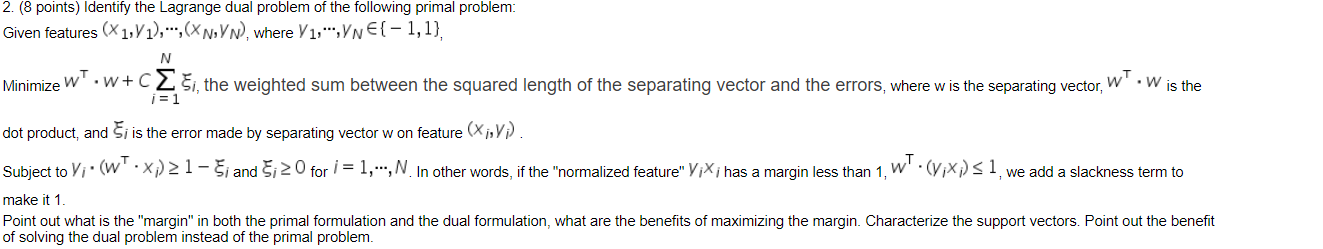
We coded SVM in python with dot-product kernel and 1-norm soft margin using the MNIST training data set and got the following results.

**Here C is the hyper-parameter that sets the trade-off between the margin and the training error.We tuned it to get different accuracy scores.**

The following table shows performance/accuracy results using MNIST test dataset w.r.t dot-product kernel and hyper parameter C :

|  |  |  |
| --- | --- | --- |
| **Gamma** | **C** | **Accuracy Score** |
|  |  |  |
| 0.001 | 2 | 92.26% |
| 0.01 | 2 | 94.72% |
| 0.001 | 3 | 93.27% |
| 0.01 | 3 | 95.31% |

The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points. For very tiny values of C, you should get misclassified examples, often even if your training data is linearly separable.



Solution 2:

**Objective:**

**(**)

s.t ,

The Lagrangian can be written as :

s.t

Taking the partial derivative w.r.t. primal variables ,

,

Thus,we can substitute the dual variables , However ,the constraint yields a new constraint and . Substituting back :

**Finally we get :**

This is the Lagrange dual problem of the primal problem :

It follows the K.K.T conditions :

**The support vectors here are :**

**1. Points on the margin (ξn = 0)**

**2. Inside the margin but on the correct side (0 < ξn < 1)**

**3. On the wrong side of the hyperplane (ξn ≥ 1)**

Here, we try to minimize the error( the number of misclassified examples) as much as possible so that most of the points are either on or away from the margin. Here c is the hyper-parameter that sets the trade-off between the margin and the training error.

If instead we use a squared penalty tern ( L2 sift margin SVM), we can safely remove the set of inequality constraints which ensure that is positive because , then setting

makes the margin constraint easier to satisfy and makes the objective better, Thus ,by using this,we can find an optimal solution without the .

**The margin for both dual and primal will be**

**Benefits of maximizing the margin :**

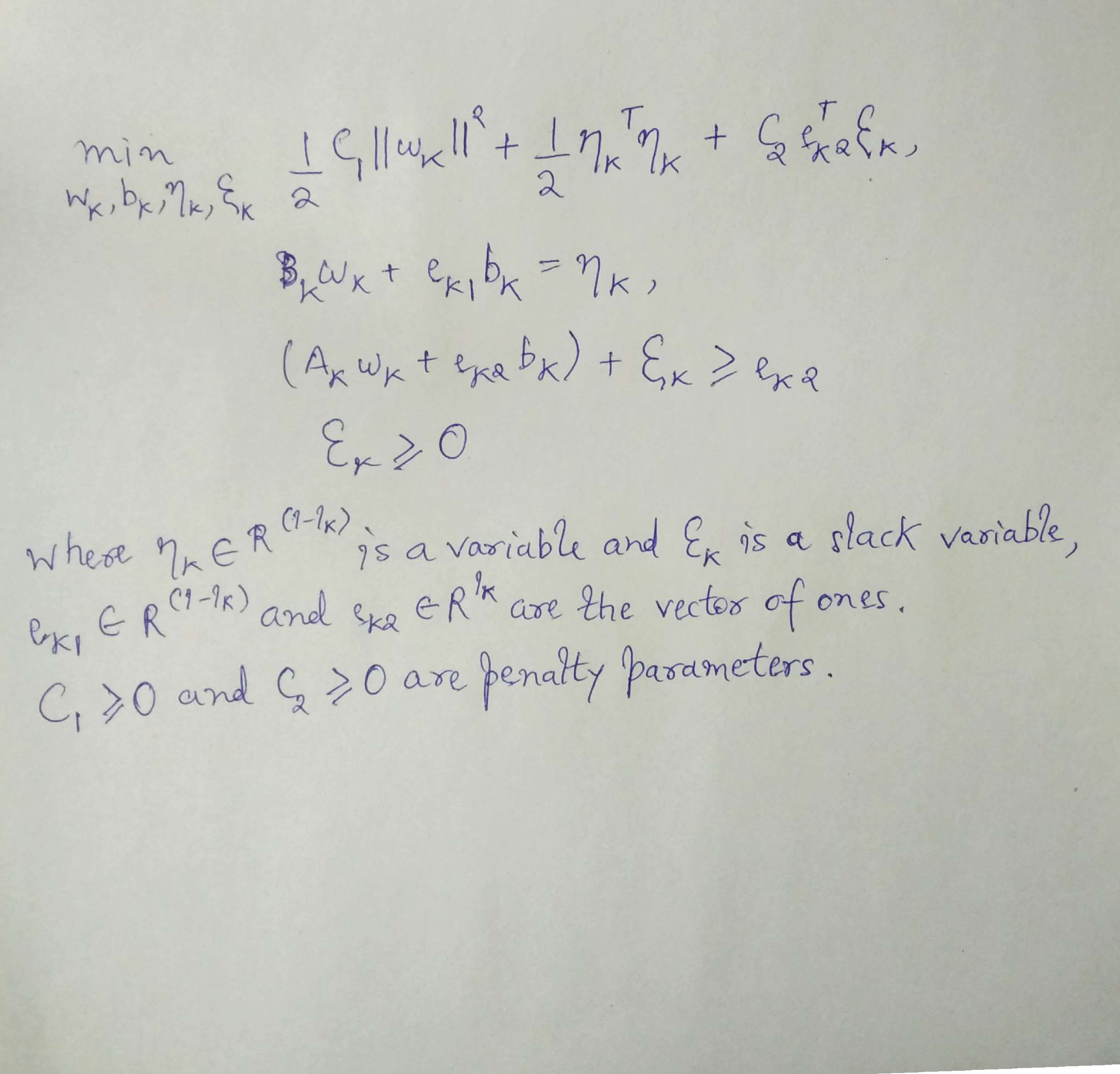
* A large margin effectively corresponds to a regularization of SVM weights which prevents overfitting. Hence, we prefer a large margin (or the right margin chosen by cross-validation) because it helps us generalize our predictions and perform better on the test data by not overfitting the model to the training data.
* Maximizing the margin seems good because points near the decision surface represent very uncertain classification decisions: there is almost a 50% chance of the classifier deciding either way. A classifier with a large margin makes no low certainty classification decisions. This gives you a classification safety margin: a slight error in measurement or a slight document variation will not cause a mis-classification.

**Benefits of solving the dual problem** **instead of the primal problem** :

* **The dual can be helpful for sensitivity analysis.** Changing the primal's right-hand side constraint vector or adding a new constraint to it can make the original primal optimal solution infeasible. However, this only changes the objective function or adds a new variable to the dual, respectively, so the original dual optimal solution is still feasible (and is usually not far from the new dual optimal solution).
* **Sometimes finding an initial feasible solution to the dual is much easier than finding one for the primal.** For example, if the primal is a minimization problem, the constraints are often of the form Ax≥bAx≥b, x≥0x≥0, for b≥0b≥0. The dual constraints would then likely be of the form ATy≤cATy≤c, y≥0y≥0, for c≥0c≥0. The origin is feasible for the latter problem but not for the former.
* **The dual variables give the shadow prices for the primal constraints.** Suppose you have a profit maximization problem with a resource constraint ii. Then the value yiyi of the corresponding dual variable in the optimal solution tells you that you get an increase of yiyi in the maximum profit for each unit increase in the amount of resource ii (absent degeneracy and for small increases in resource ii).
* **Sometimes the dual is just easier to solve.**  A problem with many constraints and few variables can be converted into one with few constraints and many variables. It is easier to solve or minimize first and then maximize rather than maximizing first and miminimzing later

3. (Optional) Formulate the primal problem and derive the dual problem if there are multiple classes.

We seek to construct k nonparallel hyperplanes (15) by solving the following convex quadratic programming problems . **The primal problem is given as :**



Now we derive the dual problem for the primal problem. This is shown as follows :

