CSE 555: Pattern Recognition
Problem Set 2: Linear Discriminant Functions and Support Vector Machines

Upasana Ghosh
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
upasanag@buffalo.edu

Problem 1:

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 report 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:

We have used the MNIST training dataset for the development and the simulation of the Support Vector Classifier presented in this report and the MNIST testing dataset has been used to report the performance of the developed Classifier. The steps involved for the development and the testing of the model Classier are as follows:

Step 1: Extracting the train and test data from the MNIST dataset

The MNIST dataset has been used for the training and testing of the model Classifier.

Results:

```
********* EXTRACTING THE MNIST DATASET *********

 Reading Training Data Features ...
 Reading Training Data Labels ...

 Reading Test Data Features ...
 Reading Test Data Labels ...

 ********* DATASET EXTRACTION COMPLETE *********
```

Step 2: Building the Model Classifier

The Linear SVC model of the SVM package in Sklearns has been used to build the Model Classifier.

Results:

```
********* BUILDING THE MODEL CLASSIFIER *********

Time:    393.7899131774902
Accuracy:  88.99

********* BUILDING THE CLASSIFIER COMPLETE*********
```

Step 3: Performing Cross-Validation and Hyperparameter Tuning

We have taken the Cross-Validation Score as 5 for validating the training dataset using Cross Validation. Tuning the hyperparameters by varying the values of the C and the gamma parameters has been time consuming, as the code took a long time to run. But, the best param value predicted, which is C = 0.001 and gamma = 0.1 has improved the score to 0.9783 and the accuracy to 97.83%.

Results:

```
********* PERFORMING CROSS-VALIDATION AND HYPERPARAMETER TUNING *********

Time Elapse (for Hyperparameter Tuning): 28999.798654322111
Best Parameter (found using GridSearchCV): {'SVM_C': 0.001,'SVM_gamma': 0.1}
Score: 0.9783

Accuracy (calculated using self-defined method): 97.83

********* CROSS-VALIDATION AND HYPERPARAMETER TUNING COMPLETE *********
```

Problem 2:

Identify the Lagrange dual problem of the following primal problem:

Given features $(x_1,y_1),\cdots,(x_N,y_N)$, where $y_1,\cdots,y_N \in \{-1,1\}$,

Minimize $w^\top \cdot w + C\sum_{i=1}^{N} \xi_i$, the weighted sum between the squared length of the separating vector and the errors, where w is the separating vector, $w^\top \cdot w$ is the dot product, and ξ_i is the error made by separating vector w on feature (x_i,y_i).

Subject to $y_i \cdot (w^\top \cdot x_i) \geq 1-\xi_i$ and $\xi_i \geq 0$ for $i=1,\cdots,N$. In other words, if the "normalized feature" $y_i x_i$ has a margin less than 1, $w^\top \cdot (y_i x_i) \leq 1$, we add a slackness term to make it 1.

Point out what is the "margin" in both the primal formulation and the dual formulation, what are the benefits of maximizing the margin. Characterize the support vectors. Point out the benefit of solving the dual problem instead of the primal problem.

Solution:

Identifying the Lagrange Dual Problem form the Primal Problem:

Since there are inequalities in the constraints, so instead of using the Lagrangian formulation, we will use the KKT equations, also known as the “Lagrangian Inequalities” to find the Lagrange Dual Problem from the given primal problem.

Let us set up a Lagrangian function which involves both primal variables (w, b and Ɛ) and the new dual variables i.e., the Lagrange multipliers α and μ.

To transform this optimization problem into its corresponding dual problem, we first find the primal Lagrangian problem, which can be written as follows:

$$\mathcal{L}(\omega, b, \varepsilon, \alpha, \mu) = \frac{1}{2}\omega^\top.\omega + C\sum_{i=1}^{N}\varepsilon_i - \sum_{i=1}^{N}\alpha_i\left[y_i(\omega^\top x_i + b) - 1 + \varepsilon_i\right] - \sum_{i=1}^{N}\mu_i\varepsilon_i,$$

where, $\frac{1}{2}\omega^\top.\omega \rightarrow$ Regularization term

$C \rightarrow$ penalization of the slack variable.

The optimum solution is given by a saddle point, which is minimum with respect to the primal variable and maximum with respect to the dual variables.

The extremum can be found by differentiating this Lagrangian equation with respect to its primal variables. So, we differentiate the equation with respect to w (normal vector), b (bias) and Ɛ (slack variables) and impose stationarity (i.e., set the derivative to 0):

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \Rightarrow \boxed{w = \sum_{i=1}^{N} \alpha_i y_i x_i} \quad -(1)$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \Rightarrow \boxed{\sum_{i=1}^{N} \alpha_i y_i = 0} \quad -(2)$$

$$\frac{\partial \mathcal{L}}{\partial \xi} = C - \alpha_i - \mu_i = 0 \Rightarrow \boxed{\alpha_i - \mu_i = C} \quad -(3)$$

Resubstituting the relations found above in the primal, we get:

$$\boxed{\mathcal{L}(w, b, \xi, \alpha, \mu) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j \langle x_i \cdot x_j \rangle}$$

such that, $\sum_{i=1}^{N} \alpha_i y_i = 0$ and $\alpha_i \geq 0$, $\mu_i \geq 0$

Identifying the Margin for the Primal and the Dual Problem:

Margin For Primal

For a binary class classifier, for any two features (x_1, y_1) and (x_2, y_2) in two opposite classes $(+1$ and $-1)$, we have the following equations:

$$wx_1 + b = 1$$

$$wx_2 + b = -1.$$

$$\therefore w(x_1 - x_2) = 0$$

The separating vector w is perpendicular to the boundaries.

The margin $\boxed{M = \|x_1 - x_2\|}$

Now, $wx_1 + b = 1$. — (1)

Taking any point x_2 in Region -1, and let x_1 be any point as close as possible to x_2 in Region $+1$, so hence,

$$x_1 = x_2 + rw \text{ — (2) (taking 'r' as any constant)}$$

Replacing the value of x_1 in eqn (1) with eqn (2),

$$w(x_2 + rw) + b = 1$$

$$\text{or, } r\|w\|^2 + wx_2 + b = 1.$$

$$\text{or, } r\|w\|^2 - 1 - 1.$$

$$\text{or, } r = \frac{2}{\|w\|^2}$$

$$\therefore M = \|x_1 - x_2\| = \|rw\| = \frac{2}{\|w\|^2} \cdot \|w\| = \frac{2}{\|w\|}$$

$$\therefore \boxed{M = \frac{2}{\|w\|}}$$

Margin For Dual

The dual problem can be written as:

$$\max L_D(\alpha_i) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

such that, $\sum_{i=1}^{N} \alpha_i y_i = 0$ and $\alpha_i \geq 0$.

Taking the derivative w.r.t α and setting it equal to zero, we get the following solution, so that we can solve for α_i:

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \qquad 0 \leq \alpha_i \leq C$$

Now, knowing the α_i, we can find the weights w for the maximal margin separating the hyperplane:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

And now, after training and finding the w by this method, given an unknown point 'u' measured on features x_i, we can classify it by looking at the sign of:

$$f(x) = w.u + b = \left(\sum_{i=1}^{N} \alpha_i y_i x_i \cdot u\right) + b.$$

Characterizing the Support Vectors:

Characterize the Support Vectors:

From the dual formulation, it is known that the weight(w) is a linear combination of the training inputs and the training outputs, i.e., x_i and y_i, and the values of α.

Now, differentiating the dual problem w.r.t α and setting it to 0, we solve for α. Only for the inputs x_i, for which the functional margin is one and that therefore lies closest to the hyperplane, the corresponding α_i non-zero. For all the other inputs, α_i will turn out to be zero. These non-zero α_s will correspond to the "Support Vectors".

We write the dual problem as,

$$\max L_D(\alpha_i) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \langle x_i \cdot x_j \rangle \quad \text{such that} \quad \alpha_i \geq 0.$$

Taking the derivative w.r.t α and setting it to 0, we get the following solution,

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C$$

Now solving for α_i, we can find w for the maximal margin separating the hyperplanes:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

After training and finding w by this method, given an unknown point 'u' measured using features x_i, we can classify it as:

$$f(x) = w.u = \sum_{i=1}^{N} (\alpha_i y_i x_i \cdot u)$$

Benefits of Maximizing the Margin

We maximize the margin in SVM for the following reasons:

1) SVM maximizes the margin, so that the model becomes slightly more robust (compared to linear regression) but more importantly - SVM supports kernels, so that non-linear relationships can also be modelled.

2) Also, 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.

3) Thirdly, "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 over the Primal Problem

The benefits of solving the dual formulation over the primal are listed below:

1) The most important benefit of solving the dual over the primal is that we can easily use the Kernel trick in the dual formulation, which is not available to us if we use the primal formulation. The optimization problem can be written as:

$$\begin{aligned} &\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ &\text{s.t.} \quad \forall i : \alpha_i \geq 0 \wedge \sum_{i=1}^{n} y_i \alpha_i = 0 \end{aligned}$$

This is almost same as the original dual formulation, except that we compute the kernel function instead of the ordinary dot-product. This is not possible in the primal formulation, where it would be necessary to explicitly compute the mapping for each data point:

$$\begin{aligned} &\min_{w, w_0} \quad \frac{1}{2} w^T w \\ &\text{s.t.} \quad \forall i : y_i (w^T \phi(x_i) + w_0) \geq 1 \end{aligned}$$

Also, classifying a new data point becomes much easier. If we solved the primal, we compute:

$$f(x) = w^T \phi(x) + w_0$$

for a new data point x, and classify it depending on the sign of the above expression. But if we solve the dual, we would get:

$$f(x) = (\sum_i \alpha_i y_i K(x_i, x)) + w_0$$

So, we have kernel function evaluations instead of explicit mapping computations, and the fact that most of the α_i would be zero. (They are non-zero only for the "support vectors", which would be few).

2) Secondly, understanding the dual problem leads to forming specialized algorithms for some important classes of linear programming problems. Examples include the transportation simplex method, the Hungarian algorithm for the assignment problem, and the network simplex method. Even column generation relies partly on duality.

3) **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)

4) 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 $Ax \geq B$, $x \geq 0$, for $b \geq 0$. The dual constraints would then likely be of the form $A^T y \leq c$, $y \geq 0$, for $c \geq 0$. The origin is feasible for the latter problem but not for the former.

5) The dual variables give the shadow prices for the primal constraints. Suppose you have the profit maximization problem with a resource constraint i. Then the value y_i of the corresponding dual variable in the optimal solution tells you that you get an increase of y_i in the maximum profit for each unit increase in the amount of resource i (absent degeneracy and for small increases in resources i)

Problem 3:

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

Solution:

For a multiclass classifier problem, we would be using a new classifier known as the 'Non-parallel Classifier for Multiclass Classification (or, NHCMC)'. Let us consider a multiple classification problem with the following training set:

$$T = \{(x_1, y_1) \dots (x_i, y_i)\} \quad \text{where } x_i \in R^n,\ i = 1 \dots k \quad \text{and } y_i \in \{1 \dots k\} \text{ in the corresponding pattern of } x_i \qquad (1)$$

For performing multiple classification, we require k non-parallel hyperplanes:

$$(w_k \cdot x) + b_k = 0, k = 1 \dots k \qquad (2)$$

For our convenience, we will denote the number of each class in the training set (T) as l_K and the points belonging to the k-th class as $A_k \in R^{lm*n}$ where l = 1 … k. Also, we define the matrix Bk as,

$Bk = [A_1^T, \dots, A_{k-1}^T, A_{k+1}^T, \dots, A_k^T]^T$

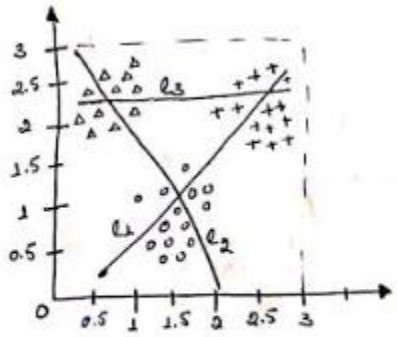

Fig 1: Example of a Linear NHCMC Learning

Linear Case

The Primal Problem

We seek to construct k non-parallel hyperplanes by solving the following convex quadratic programming problems (QPPS):

$$\min_{w_k, b_k, \eta_k, \epsilon_k} \frac{1}{2} c_1 \|w_k\|^2 + \frac{1}{2} \eta_k^T \eta_k + c_2 c_k^T \epsilon_k \quad \text{——— (3)}$$

$$\text{such that, } B_k \cdot w_k + c_{k1} \cdot B_k \cdot \eta_k$$

$$(A_k \cdot w_k + c_{k2} \cdot b_k) + \epsilon_k \geq c_{k2}$$

$$\epsilon_k \geq 0$$

where $\eta_k \in R^{(l-l_k)}$ is a variable and ϵ_k is a slack variable, $c_{k1} \in R^{(l-l_k)}$ and $c_{k2} \in R^{l_k}$ are the vectors. $c_1 \geq 0$ and $c_2 \geq 0$ are the penalty parameters.

To illustrate the primal problem of NHCMC, we generated an artificial two dimensional 3-class dataset. The geometric interpretation of the above problem is that the $i^{th}\alpha$ belongs to R^2 as shown in the fig1 above, where α minimizes the sum of the square distance from the hyperplane of k-1 classes i.e., all classes except for those of the k^{th} class, and the points of the k^{th} class are far from the i^{th} hyperplane. For example, taking the '+' class in the figure, we hope that the hyperplane of the 'o' class l_1 is far from the '+' points and closest to the 'o' and 'x' points. To minimize the classification error, the points of the kth class are at a distance from the hyperplane and we minimize the sum of error variables with soft margin SVM.

The differences between the Multiple Birth Support Vector Machines (MBSVM) and NHCMC are that we introduce a regularization form to implement Structural Risk Minimization (SRM) principle and a variable is introduced to make a set of objective functions that serves as the constraints. These changes have many positive effects on the original NHCMC.

The Dual Problem:

To get the solution of the problem (3), we need to derive its dual problem. The Lagrangian of the problem (3) is given by:

$$L(w_k, b_k, \eta_k, \epsilon_k, \alpha, \beta, \lambda) = \frac{1}{2} c_1 \|w_k\|^2 + \frac{1}{2} \eta_k^T \eta_k + c_2 e_{k2}^T \epsilon_k \quad \text{——— (4)}$$
$$+ \lambda^T (B_k w_k + e_{k2} \cdot b_k - \eta_k) - \alpha^T (A_k w_k + e_{k2} \cdot b_k + \epsilon_k - e_{k2})$$
$$- \beta^T \epsilon_k, \text{ where } \alpha = (\alpha_1, \ldots, \alpha_{e_k})^T,$$
$$\beta = (\beta_1, \ldots, \beta_{e_k})^T, \quad \lambda = (\lambda_1, \ldots, \lambda_{L-e_k})^T$$

are the Lagrange multiplier vectors.

The KKT conditions for $w_k, b_k, \eta_k, \epsilon_k$ and α, β, λ are given by

$$\nabla_{w_k} L = c_1 w_k + B_k^T \cdot \lambda - A b^T \alpha = 0 \quad \text{——— (5)}$$

$$\nabla_{b_k} L = e_{k_1}^T \lambda - e_{k_2}^T \alpha = 0 \quad \text{——— (6)}$$

$$\nabla_{\eta_k} L = \eta_k - \lambda = 0 \quad \text{——— (7)}$$

$$\nabla_{\epsilon_k} L = c_2 e_{k_2} - \alpha - \beta = 0 \quad \text{——— (8)}$$

$$B_k w_k + e_{k_1} b_k = \eta_{k1} \quad \text{——— (9)}$$

$$(\Delta_k w_k + e_{k_2} b_k) + \epsilon_k \geq e_{k_2}, \epsilon_{1k} \geq 0 \quad \text{——— (10)}$$

$$\alpha^T ((\Delta_k w_k + e_{k_2} b_k) + \epsilon_k) = 0, \; \beta^T \epsilon_k = 0, \; \alpha \geq 0, \; \beta \geq 0 \quad \text{——— (11)}$$

Since $\beta \geq 0$, from eqn(8) we have $0 \leq \alpha < c_2 e_{k2}$ ------- (12)

And from eqn(5) we have $w_k = (-1/c_1)[B_k^T \lambda - A_k^T \alpha]$ ------- (13)

Now putting in eqn(13) and eqn(7) into the Lagrangian and using eqn(11), we obtain the dual problem of eqn(3).