Problem Set 2

CS 6375

Due: 3/6/2022 by 11:59pm

Note: all answers should be accompanied by explanations and relevant code for full credit. All code (Python or MATLAB only) should be turned in with your answers to the following questions. Late homeworks will not be accepted.

Problem 1: Parkinson's Disease (40 pts)

For this problem, you will use the cancer data set provided with this problem set. The data has been divided into three pieces park_train.data, park_validation.data, and park_test.data. These data sets were generated using the UCI Parkinsons Data Set data set (follow the link for information about the format of the data). Note that class label, health status of the subject, is the first column in the data set. All code (Python or MATLAB only) should be turned in with your answers to the following questions.

$Gx \leq \underset{\overset{\uparrow}{0}}{h}$ $c \geq \lambda_i \geq 0$

$-\lambda_i \leq 0$

1. Primal SVMs

   (a) Using gradient descent or quadratic programming, apply the SVM with slack formulation to train a classifier for each choice of
   $c \in \{10^{-4}, 10^{-3}, \cdots, 10^3, 10^4\}$ without using any feature maps.

   (b) What is the accuracy of the learned classifier on the training set for each value of c?

   (c) Use the validation set to select the best value of c. What is the accuracy on the validation set for each value of c?

   (d) Report the accuracy on the test set for the selected classifier.

2. Dual SVMs with Gaussian Kernels

   (a) Using quadratic programming, apply the dual of the SVM with slack formulation to train a classifier for each choice of
   $c \in \{10^{-4}, 10^{-3}, \cdots, 10^3, 10^4\}$ using a Gaussian kernel with
   $\sigma^2 \in \{10^{-3}, \cdots, 10^3\}$.

   (b) What is the accuracy of the learned classifier on the training set for each pair of c and σ^2?

   (c) Use the validation set to select the best value of c and σ^2. What is the accuracy on the validation set for each pair of c and σ^2?

   (d) Report the accuracy on the test set for the selected classifier.

3. Which of these approaches (if any) should be preferred for this classification task? Explain.

Problem 2: Method of Lagrange Multipliers (15 pts)

Suppose that we modified the objective function in the SVM with slack formulation to be a quadratic penalty instead of a linear penalty, that is minimize $\frac{1}{2}||w||^2 + c\sum_i \xi_i^2$ subject to the same constraints as the standard SVM with slack. What is the dual of this new quadratic penalized SVM with slack problem for a fixed c? Can the kernel trick still be applied?

Problem 3: Poisonous Mushrooms? (25 pts)

For this problem, you will use the mushroom data set provided with this problem set. The data has been divided into two pieces mush_train.data and mush_test.data. These data sets were generated using the UCI Mushroom data set (follow the link for information about the format of the data). Note that the class label is the first column in the data set.

1. Assuming you break ties using the attribute that occurs **last** (left to right) in the data, draw the resulting decision tree and report the maximum information gain for each node that you added to the tree.

2. What is the accuracy of this decision tree on the test data?

3. Now consider arbitrary input data. Suppose that you decide to limit yourself to decision trees of height one, i.e., only one split. Is the tree produced by the information gain heuristic optimal on the training data (that is, no other decision tree has higher accuracy)?

Problem 4: Cross-Validation (20 pts)

Using a single tuning set for the hyperparameters can yield an unreliable predictor of the class label, i.e., maybe it was not a representative sample of the data, plus some data is "wasted" using this approach. An alternative approach that is particularly applicable for small data sets is k-fold cross-validation.

1. Partition the non-test data into k equally sized buckets.

2. For each possible set of hyperparameters you will train the model using exactly $k-1$ of the partitions while the held out partition is used as a validation data set.

3. As there are k different ways to hold out one partition, all k possibilities are tried and the average validation set accuracy (as measured by the appropriate held-out data) of the k different models learned for each of the hyperparameter settings is used to select the winning hyperparameters.

4. Finally, the model is retrained using all of the non-test data with the winning hyperparameters and then evaluated using the test data.

Apply 10-fold cross validation to fit an SVM with slack classifier (no feature maps) to the data set wdbc_train.data (each row corresponds to a single data observation and the class label +1/-1 is the first entry in each row). Use the same hyperparameter ranges as Problem 1.1 and the partitions for cross validation should be selected as equally sized contiguous blocks of data starting from the first data element. Report the best setting of the hyperparameters and the accuracy on the test set wdbc_test.data.

Problem 2: Method of Lagrange Multipliers (15 pts)

Suppose that we modified the objective function in the SVM with slack formulation to be a quadratic penalty instead of a linear penalty, that is minimize $\frac{1}{2}||w||^2 + c\sum_i \xi_i^2$ subject to the same constraints as the standard SVM with slack. What is the dual of this new quadratic penalized SVM with slack problem for a fixed c? Can the kernel trick still be applied?

$\min_{w b \xi} \frac{1}{2}||w||^2 + C\sum \xi_i^2$ such that $y_i(w^Tx^{(i)} + b) \geq 1 - \xi_i$ for all i , $\xi_i \geq 0$ for all i

$$L(w,b,\xi,\lambda,\mu) = \frac{1}{2}w^Tw + C\sum_i \xi_i^2 + \sum_i \lambda_i\left(1 - \xi_i - y_i(w^Tx^{(i)}+b)\right) + \sum_i -\mu_i\xi_i$$

Convex in w, b, ξ.

$1 - \xi_i - y_i(w^Tx^{(i)}+b) \leq 0$

$-\xi_i \leq 0$

$$\left.\begin{array}{l} \dfrac{\partial L}{\partial w_k} = w_k + \sum_i -\lambda_i y_i x_k^{(i)} \\ \dfrac{\partial L}{\partial b} = \sum \lambda_i y_i \\ \dfrac{\partial L}{\partial \xi_k} = 2C\sum_i \xi_k^{(i)} - \lambda_k - \mu_k \end{array}\right\} = 0$$

$$w_k = -\left(\sum_i -\lambda_i y_i x_k^{(i)}\right)$$

$$2C\sum_i \xi_k^{(i)} = \lambda_k + \mu_k$$

$$\sum_i \xi_k^{(i)} = \frac{\lambda_k + \mu_k}{2C}$$

t : Total # of values in array i)

$$\bar{\xi}_k = \frac{\lambda_k + \mu_k}{2Ct}$$

$$\sum \xi^2 = \sum(\xi_i - \bar{\xi})^2$$

$$\sum_i \left(x^{(i)} - \frac{\lambda_i + \mu_i}{2C \cdot K}\right)^2$$

$\sum \xi^2$

$\Uparrow$

plug back to L

$$\max_{\lambda \geq 0} -\frac{1}{2}\sum_i\sum_j \lambda_i\lambda_j y_i y_j x^{(i)T}x^{(j)} + \sum_i \lambda_i + \sum_i (\xi_i - \bar{\xi}_i)^2 + \sum_i \mu_i \xi_i$$

$$\max_{\lambda \geq 0} -\frac{1}{2}\sum_i\sum_j \lambda_i\lambda_j y_i y_j x^{(i)T}x^{(j)} + \sum \lambda_i + \sum_i \left(\xi_i - \frac{\lambda_k + \mu_k}{2Ct}\right)^2 + \sum_i \mu_i \xi_i$$