CS480/680: Introduction to Machine Learning

Homework 2

Due: 11:59 pm, October 25, 2023, submit on LEARN and CrowdMark.

Include your name and student number!

Submit your writeup in pdf and all source code in a zip file (with proper documentation). Fill in the provided stub files, keeping the directory structure. You do not have to submit the provided datasets. Make sure your code runs! [Text in square brackets are hints that can be ignored.]

Exercise 1: Fun with Classification (5 pts)

For this problem, you are allowed to use `statsmodels` and `sklearn` as directed.

1. (2 pts) Run logistic regression, SVM with ℓ_2 regularization with parameter 1 (soft-margin SVM), and SVM with regularization parameter `float('inf')` (hard-margin SVM) on Mystery Dataset A (note that there is only a training dataset and no test dataset). Use `Logit` from `statsmodels` and `SVC` (with linear kernel) from `sklearn`. One of these methods will not work – give a mathematically rigorous explanation as to why this happens. [Think carefully about the loss function used for this method. Look at the error message, and think about what happens in the case indicated.] How could the associated problems be remedied? Discuss similarities and differences between the solution obtained via these two working methods.

   Ans:

2. (3 pts) Take your solution for the soft-margin SVM from the previous part. For each point in the dataset, take its inner product with the produced coefficient vector, and scale the result by the sign of each point's label (replace 0's with -1's). How many of these values are ≤ 1? [Be sure you're getting all of them – there may be numerical precision issues, so if in doubt, err on the side of counting a point.] Based on your answer to these questions, sketch a 2D caricature of what the points and the hyperplane defined by the SVM solution look like. (A "caricature" in this context means a rough sketch of what you think Mystery Dataset A looks like, using the information you have learned in this question and the previous part. There will be a few key features of the dataset you can emphasize in your caricature. This might use more information than the literal exact answers to the questions that have been asked (though not much more).) Write the parameter vector solution to the SVM problem as a linear combination of some points in your dataset. How many points did you require? [You may find built-in functions useful for this purpose.]

   Compute the solution for the same three methods on Mystery Dataset B. You will again run into issues with one of them – explain why. [The answer is likely to be simpler than last time.] Find a way to write the parameter vector solution to the SVM problem as a linear combination of some points in your dataset – do not report the solution itself, but report how many points you used, and how you arrived at this answer. Compare the empirical prediction accuracy (i.e., using 0-1 loss) of the successfully-trained classifiers on the test set.

   Ans:

3. (Optional – 0 pts) Construct a dataset in which every point is a support vector. Do this for the soft margin ($C = 1$) and hard margin ($C = \infty$) cases. Make sure this dataset has $n \geq 100$ points. Submit these datasets in the `data` folder, as well as code required to generate them and demonstrate the number of support vectors.

   Ans:

Exercise 2: Support Vector Regression (8 pts)

Let us consider support vector regression:

$$\min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n} \max\{|y_i - (\mathbf{w}^\top \mathbf{x}_i + b)| - \varepsilon, 0\}, \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, and $\|\mathbf{w}\|_2 := \sqrt{\sum_{j=1}^d w_j^2}$ is the Euclidean norm. The above expression is the loss function, the error is simply the latter term $C\sum_{i=1}^n \max\{|y_i - (\mathbf{w}^\top \mathbf{x}_i + b)| - \varepsilon, 0\}$.

1. (2 pts) Derive the Lagrangian dual of the support vector regression loss function (1). Please include intermediate steps so that you can get partial credit.

   Ans:

   In the following you will complete and implement the following gradient algorithm for solving support vector regression in Equation (1):

   Algorithm 1: GD for SVR.

   ```
      Input: $X \in \mathbb{R}^{n\times d}$, $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{w} = \mathbf{0}_d$, $b = 0$, max_pass $\in \mathbb{N}$, step size $\eta$
      Output: $\mathbf{w}, b$
   1  for $t = 1, 2, \ldots,$ max_pass do
   2      for $i = 1, 2, \ldots, n$ do
   3          choose step size $\eta$
   4          if $|y_i - (\langle \mathbf{x}_i, \mathbf{w}\rangle + b)| \geq \varepsilon$ then
   5              $\mathbf{w} \leftarrow$                    // $\mathbf{x}_i$ is the $i$-th row of $X$
   6              $b \leftarrow$
   7          $\mathbf{w} \leftarrow$                        // proximal step
   ```

   Note that this differs a bit from what you've seen so far, in terms of gradient descent. Rather than taking steps based on the entire loss function, we instead take a step based on the unregularized loss, and then perform a projection step based on the regularizer (sometimes called a "proximal step").

2. (2 pts) Compute the gradient with respect to $\mathbf{w}$ and b for each second term in Equation (1). Note that in places where the function is non-differentiable, you might have to compute a sub-gradient.

$$C\sum_{i=1}^n \max\{|y_i - (\mathbf{w}^\top \mathbf{x}_i + b)| - \varepsilon, 0\} \tag{2}$$

   Ans:

3. (1 pt) Find the closed-form solution of the following proximal step:

$$\mathsf{P}^\eta(\mathbf{w}) = \underset{\mathbf{z}}{\operatorname{argmin}}\ \frac{1}{2\eta}\|\mathbf{z} - \mathbf{w}\|_2^2 + \frac{1}{2}\|\mathbf{z}\|_2^2 \tag{3}$$

   Ans:

4. (3 pts) Implement Algorithm 1. You should use part 2 to complete lines 5-6, and part 3 for line 7. Run it on Mystery Dataset C (this is Mystery Dataset A from Assignment 1, but reused), and report your training error, training loss, and test error. Use $C = 1$ and $\varepsilon = 0.5$.

   Ans:

Exercise 3: Kernels (5 pts)

For the following questions you might find it useful to recall the definition of a matrix being positive semidefinite (PSD). A matrix $M \in \mathbb{R}^{d\times d}$ is PSD if and only if $x^T M x \geq 0$ for all vectors $x \in \mathbb{R}^d$. It may also be helpful to refresh yourself on Taylor series.

1. (2 pt) For $x, y \in \mathbb{R}$, consider the kernel function $k(x, y) = \exp\left(-\alpha\,(x-y)^2\right)$. What is the corresponding feature map $\phi(\cdot)$ such that $\phi(x)^T\phi(y) = k(x, y)$? If you were using this kernel for an SVM model, would

you prefer to solve the primal or dual representation? Why?

Ans:

2. (1 pt) Consider the function $\frac{1}{1-xy}$, where $x, y \in (-1, 1)$. Is this function a valid kernel? If so, write out the corresponding feature map $\phi(\cdot)$, if not, explain why.

   Ans:

3. (1 pt) Consider the function $\log(1 + xy)$, where $0 < x, y \in \mathbb{R}$. Is this function a valid kernel? If so, write out the corresponding feature map $\phi(\cdot)$, if not, explain why.

   Ans:

4. (1 pt) Consider the function $\cos(x + y)$, where $x, y \in \mathbb{R}$. Is this function a valid kernel? If so, write out the corresponding feature map $\phi(\cdot)$, if not, explain why.

   Ans:

Exercise 4: Decision Trees (8 pts)

In this exercise we will implement decision trees for binary classification. Use the provided stub files.

Recall: decision trees are constructed by repeatedly splitting of nodes. We split a node by measuring the loss of splitting with respect to each possible feature and threshold, and split based on the feature and threshold that minimizes this loss. Mathematically:

$$X_L = \{x \ : \ x \in X \wedge x[i] \leq j\},$$

$$X_R = \{x \ : \ x \in X \wedge x[i] > j\},$$

where $x[i]$ is the ith coordinate of point x. The vector of labels y is split into vectors y_L and y_R using the same indices.

The loss of splitting a training dataset into a left and right half is computed as

$$\ell(X, y, i, j) = \frac{|y_L|}{|y|}\ell(y_L) + \frac{|y_R|}{|y|}\ell(y_R)$$

We will consider the following loss functions ℓ, specialized for binary classification.[a] Define $\hat{p}$ for a vector of labels y to be $\frac{|\{y_j = 1 : y_j \in y\}|}{|y|}$, that is, the fraction of labels which are 1. We have the following three loss functions.

Misclassification error:

$$\min\{\hat{p}, 1 - \hat{p}\}$$

Gini coefficient:

$$\hat{p}(1 - \hat{p})$$

Entropy:

$$-\hat{p}\log_2(\hat{p}) - (1 - \hat{p})\log_2(1 - \hat{p})$$

We do not split a node if it is pure (i.e., consists entirely of either 0's or 1's), or if a split would exceed a maximum depth hyperparameter provided to the decision tree (recall that the depth of a single-node tree is 0).

a) (6 pts) Implement and train decision trees on the provided dataset D. Create a different plot for each of the three loss functions (misclassification error, Gini index, and entropy). The x-axis of each plot should show the maximum depth of the tree (starting from 0), and the y-axis should indicate the accuracy. Include two trend lines, one for the training accuracy and test accuracy. Observe and comment on how the different loss functions perform, and how train and test accuracy change as a function of the maximum depth.

b) (2 pts) Implement and use Bagging on decision trees. Use the entropy loss, and create an ensemble of 101 decision trees, capping the maximum depth at 3. Repeat 11 times, report the median, minimum, and maximum test classification accuracy. Repeat, but using the random forests method. In particular, at each split, consider only a random $\sqrt{d}$ features when deciding which dimension to split on. Since $\sqrt{d} \approx 3.6$ for the given dataset, choose a random 4 features for each time. Report the classification accuracies as before. Comment on performance between the two, as well as how they perform in comparison to the non-ensemble methods from the previous part.

[a] Note that these are slightly different from what we discussed in class, since we are only focusing on binary classification. Additionally, there was some implicit rescaling which we omit here.