CS480/680: Introduction to Machine Learning

Homework 2

Due: 11:59 pm, February 12, 2021, submit on LEARN and CrowdMark (see Piazza).

Include your name and student number!

Submit your writeup in pdf and all source code in a zip file (with proper documentation). Write a script for each programming exercise so that the TAs can easily run and verify your results. Make sure your code runs!

[Text in square brackets are hints that can be ignored.]

Exercise 1: k-Nearest Neighbours (5 pts)

The KNN-dataset provides a 4x4 representation of the handwritten digits 5, 6, and 8. Using the notebook provided, train a KNN classifier using 5-fold cross validation to determine the optimal k-value for this task.

1. (3 pt) Create a graph that shows the average accuracy based on 5-fold cross validation when varying the number of neighbours from 1 to 30.

   Ans:

2. (1 pt) Report the best hyperparameter found by 5-fold cross-validation and the cross-validation accuracy.

   Ans:

3. (1 pt) Report the test accuracy based on the best hyperparameter.

   Ans:

Exercise 2: Poisson Regression (4 pts)

Recall that in logistic regression we assumed the binary label $Y_i \in \{0, 1\}$ follows the Bernoulli distribution: $\Pr(Y_i = 1|X_i) = p_i$, where p_i also happens to be the mean. Under the independence assumption we derived the log-likelihood function:

$$\sum_{i=1}^{n} (1 - y_i) \log(1 - p_i) + y_i \log(p_i). \tag{1}$$

Then, we parameterized the mean parameter p_i through the logit transform:

$$\log \frac{p_i}{1 - p_i} = \mathbf{w}^\top \mathbf{x}_i + b, \quad \text{or equivalently} \quad p_i = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x}_i - b)}. \tag{2}$$

Lastly, we found the weight vector $\mathbf{w}$ and b by maximizing the log-likelihood function.

In the following we generalize the above idea to the case where $Y_i \in \mathbb{N}$, i.e., Y_i can take any natural number (for instance, when we are interested in predicting the number of customers or network packages).

1. (1 pt) Naturally, we assume $Y_i \in \mathbb{N}$ follows the Poisson distribution (with mean $\mu_i \geq 0$):

$$\Pr(Y_i = k|X_i) = \frac{\mu_i^k}{k!} \exp(-\mu_i), \quad k = 0, 1, 2, \ldots. \tag{3}$$

   Given a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$, what is the log-likelihood function (of μ_i's) given $\mathcal{D}$?

   Ans:

2. (1 pt) Can you give some justification of the parameterization below?

$$\log \mu_i = \mathbf{w}^\top \mathbf{x}_i + b. \tag{4}$$

   Ans:

3. (1 pt) Based on the above, write down the objective function for Poisson regression. Please specify the optimization variables and whether you are maximizing or minimizing. [Constants can be dropped.]

   Ans:

4. (1 pt) Compute the gradient of your objective function above and formulate a gradient algorithm for finding the weight vector $\mathbf{w}$ and b.

   Ans:

Exercise 3: 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 – mathematically explain why (that is, give an explanation which is more informative than any error message you encounter). How could the associated problems be remedied? Discuss similarities and differences between the solution obtained via these three 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. 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:

Exercise 4: Support Vector Regression (8 pts)

Let us consider support vector regression:

$$\min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n} \max\{|y_i - (\mathbf{w}^\top \mathbf{x}_i + b)| - \varepsilon, 0\}, \tag{5}$$

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.

1. (2 pts) Derive the Lagrangian dual of the support vector regression loss function (5). Please include intermediate steps so that you can get partial credits.

   Ans:

   In the following you will complete and implement the following gradient algorithm for solving support vector regression in Equation (5):

Algorithm 1: GD for SVR.

Input: $X \in \mathbb{R}^{n \times d}$, $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{w} = \mathbf{0}_d$, $b = 0$, $\mathsf{max\_pass} \in \mathbb{N}$, step size η
Output: $\mathbf{w}, b$

1 **for** $t = 1, 2, \ldots, \mathsf{max\_pass}$ **do**
2   **for** $i = 1, 2, \ldots, n$ **do**
3   choose step size η
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.

2. (2 pts) Compute the gradient w.r.t. $\mathbf{w}$ and b for each second term in Equation (5). Note that in places where the function is non-differentiable, you might have to compute a sub-gradient.

$$C \sum_{i=1}^{n} \max\{|y_i - (\mathbf{w}^\top \mathbf{x}_i + b)| - \varepsilon, 0\} \tag{6}$$

Ans:

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

$$\mathsf{P}^\eta(\mathbf{w}) = \operatorname*{argmin}_{\mathbf{z}} \ \frac{1}{2\eta}\|\mathbf{z} - \mathbf{w}\|_2^2 + \frac{1}{2}\|\mathbf{z}\|_2^2 \tag{7}$$

Ans:

4. (3 pts) Implement Algorithm 1. You should use Ex 1.3 to complete lines 5-6, and Ex 1.4 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 $\varepsilon = 0.1$.

Ans: