CS480/680: Introduction to Machine Learning

Homework 2

Due: 11:59 pm, June 8, 2022, 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: 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: The likelihood function of poisson distribution is:

$$\prod_{i=1}^{n} \frac{\mu_i^{y_i}}{y_i!} \exp(-\mu_i)$$

By taking the logrithm of that, we will get the log-likelihood function:

$$\begin{aligned}
\log\left(\prod_{i=1}^{n} \frac{\mu_i^{y_i}}{y_i!} \exp(-\mu_i)\right) &= \sum_{i=1}^{n} \log\left(\frac{\mu_i^{y_i}}{y_i!} \exp(-\mu_i)\right) \\
&= \sum_{i=1}^{n} y_i \log(\mu_i) - \mu_i - \log(y_i!) \\
&= \sum_{i=1}^{n} \left(y_i \log(\mu_i) - \mu_i - \sum_{j=1}^{y_i} \log(j)\right)
\end{aligned}$$

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

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

Ans: $\mu_i \geq 0$, so $\log \mu_i$ ranges from negative infinity to positive infinity, which is the same range as $\mathbf{w}^\top \mathbf{x}_i + b$. In other words, they both range over $\mathbb{R}$.

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: Since we are doing the maximum likelihood estimation, the objective function is to find the parameter w, b which maximizes the likelihood.

$$\begin{aligned}
\hat{w} &= \underset{w,b}{arg\max} \prod_{i=1}^{n} \frac{\mu_i^{y_i}}{y_i!} \exp(-\mu_i) \\
&= \underset{w,b}{arg\max} \sum_{i=1}^{n} y_i \log(\mu_i) - \mu_i - \log(y_i!) \\
&= \underset{w,b}{arg\max} \sum_{i=1}^{n} y_i(\mathbf{w}^\top \mathbf{x}_i + b) - e^{\mathbf{w}^\top \mathbf{x}_i + b} - \log(y_i!) \\
&= \underset{w,b}{arg\max} \sum_{i=1}^{n} y_i(\mathbf{w}^\top \mathbf{x}_i + b) - e^{\mathbf{w}^\top \mathbf{x}_i + b} \\
&= \underset{w,b}{arg\min} \sum_{i=1}^{n} -y_i(\mathbf{w}^\top \mathbf{x}_i + b) + e^{\mathbf{w}^\top \mathbf{x}_i + b}
\end{aligned}$$

   If we use the padding trick, let $w = [w, b]^\top$ and $x = [x, 1]^\top$:

$$\hat{w} = \underset{w}{arg\min} \sum_{i=1}^{n} -y_i(\mathbf{w}^\top \mathbf{x}_i) + e^{\mathbf{w}^\top \mathbf{x}_i}$$

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: Let

$$l_{w,b}(x_i, y_i) = -y_i(\mathbf{w}^\top \mathbf{x}_i + b) + e^{\mathbf{w}^\top \mathbf{x}_i + b}$$

   So we need to compute:

$$\underset{w,b}{arg\min} \sum_{i=1}^{n} l_{w,b}(x_i, y_i)$$

   The gradient is:

$$\begin{aligned}
\frac{\partial l_{w,b}}{\partial w} &= \frac{\partial}{\partial w} - y_i(\mathbf{w}^\top \mathbf{x}_i + b) + e^{\mathbf{w}^\top \mathbf{x}_i + b} \\
&= -y_i \mathbf{x}_i + e^{\mathbf{w}^\top \mathbf{x}_i + b} \mathbf{x}_i \\
\frac{\partial l_{w,b}}{\partial b} &= \frac{\partial}{\partial b} - y_i(\mathbf{w}^\top \mathbf{x}_i + b) + e^{\mathbf{w}^\top \mathbf{x}_i + b} \\
&= -y_i + e^{\mathbf{w}^\top \mathbf{x}_i + b}
\end{aligned}$$

If we calculate the second degree gradients:

$$\begin{aligned}\frac{\partial^2 l_{w,b}}{\partial w^2} &= \frac{\partial^2}{\partial w^2} - y_i \mathbf{x}_i + e^{\mathbf{w}^\top \mathbf{x}_i + b}\mathbf{x}_i \\ &= e^{\mathbf{w}^\top \mathbf{x}_i + b}||\mathbf{x}_i||_2^2 \geq 0 \\ \frac{\partial^2 l_{w,b}}{\partial b^2} &= \frac{\partial^2}{\partial b^2} - y_i + e^{\mathbf{w}^\top \mathbf{x}_i + b} \\ &= e^{\mathbf{w}^\top \mathbf{x}_i + b} \geq 0\end{aligned}$$

So the funciton is convex, we can use gradient descent algorithm to compute w, b. We start by initialize the weight vector w_0, b_0. Then, for every $t = 1, 2, 3, \ldots$, in each iteration, we choose a step size η_t and calculate the next w_t, b_t by:

$$w_t = w_{t-1} - \eta_t \frac{1}{n}\sum_{i=1}^{n} \frac{\partial l_{w,b}}{\partial w}(w_{t-1}) = w_{t-1} - \eta_t \frac{1}{n}\sum_{i=1}^{n}(-y_i \mathbf{x}_i + e^{\mathbf{w}_{t-1}^\top \mathbf{x}_i + b}\mathbf{x}_i)$$

$$b_t = b_{t-1} - \eta_t \frac{1}{n}\sum_{i=1}^{n} \frac{\partial l_{w,b}}{\partial b}(b_{t-1}) = b_{t-1} - \eta_t \frac{1}{n}\sum_{i=1}^{n}(-y_i + e^{\mathbf{w}_{t-1}^\top \mathbf{x}_i + b})$$

We repeat until some stop criteria is hit. For exmaple, after 1000 iterations or $|w_t - w_{t-1}| < 0.001$

Exercise 2: 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: It turns out that the logistic regression method does not work with those data, and the error message is "Perfect separation detected, results not available". As the name suggestes, all the data points in this dataset are perfectly seperated, i.e, there exists a certain parameter vector β such that:

   $$\beta^\top \mathbf{x}_i < 0 => y = 0 \text{ and } \beta^\top \mathbf{x}_i > 0 => y = 1 \text{ and } \beta^\top \mathbf{x}_i \neq 0 \quad \forall i$$

   Then, for any $t \in \mathbb{R}$, for those data points with $y_i < 0$, we have $\beta^\top \mathbf{x}_i < 0$, we have:

   $$\lim_{t->\infty} t\beta^\top \mathbf{x}_i = -\infty \quad \forall i \text{ s.t } y_i < 0$$

   And Similarly, for those data points with $y_i > 0$, we have $\beta^\top \mathbf{x}_i > 0$:

   $$\lim_{t->\infty} t\beta^\top \mathbf{x}_i = \infty \quad \forall i \text{ s.t } y_i > 0$$

   We can take $\mathbf{w} = t\beta$, so the above expression becomes:

   $$\lim_{\mathbf{w}->\infty} \mathbf{w}^\top \mathbf{x}_i = -\infty \quad \forall i \quad y_i < 0 \text{ and } \lim_{\mathbf{w}->\infty} \mathbf{w}^\top \mathbf{x}_i = \infty \quad \forall i \quad y_i > 0$$

   Now, let us take a look at the probablity function. $Pr[y = 1|x, w] = p(\mathbf{x}, \mathbf{w}) = \frac{1}{1+e^{-\mathbf{w}^\top \mathbf{x}_i}}$. To maximize the likelihood, we want $p(\mathbf{x}, \mathbf{w}) \to 1$ for those data points with $y_i = 1$, and $p(\mathbf{x}, \mathbf{w}) \to 0$ for those data

points with $y_i = 0$. However, with the above limits, we can have the expression below:

$$\lim_{\mathbf{w}->\infty} Pr[y = 1|x, w] = \lim_{\mathbf{w}->\infty} \frac{1}{1 + e^{-\mathbf{w}^\top \mathbf{x}_i}} = 1 \quad \forall i \quad y_i > 0$$
$$\lim_{\mathbf{w}->\infty} Pr[y = 0|x, w] = \lim_{\mathbf{w}->\infty} 1 - \frac{1}{1 + e^{-\mathbf{w}^\top \mathbf{x}_i}} = 1 \quad \forall i \quad y_i < 0$$

This essentially means that we can maximize the likelihood as $\mathbf{w} \to \infty$:

$$\lim_{\mathbf{w}->\infty} \prod_{i=1}^{n} Pr[(\mathbf{x}_i, y_i)|\mathbf{w}] = 1$$

The objective funciton of logistic regression is as follows:

$$\hat{\mathbf{w}} = \operatorname*{argmax}_{\mathbf{w}} \prod_{i=1}^{n} Pr[(\mathbf{x}_i, y_i)|\mathbf{w}]$$

And we can tell from the function of p that, $Pr[y = 1|x, w]$ increases monotonically ($\frac{d}{dx}\frac{1}{1+e^{-x}} = (1 + e^{-x})e^{x} > 0$). However, from the limit we derived above, we can see that we can maximize the likelihood to 1 (which is the global maximum) as $\mathbf{w}$ goes to infinity. This essentially means, such $\mathbf{w}$ does not exist. So we cannot use logistic regression in this case.

To solve this problem, we can regularize the logistic regression. In other words, we can add some restrictions to the vector $\mathbf{w}$ so that it won't go to infinity. For example, we can change the objective function to:

$$\hat{\mathbf{w}} = \operatorname*{argmax}_{\mathbf{w}} \prod_{i=1}^{n} Pr[(\mathbf{x}_i, y_i)|\mathbf{w}] + \frac{1}{2}\|\mathbf{w}\|_2^2$$

As for the two working solutions, we got exactly same results. i.e., the $\mathbf{w}, b$ from both methods are exactly the same.

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: All of the 2000 values are less than or equal to 1. Moreover, those values are very close to either 1 or -1. This essentially means that all the datapoints are sitting right on the margin. The caricature is draw below.

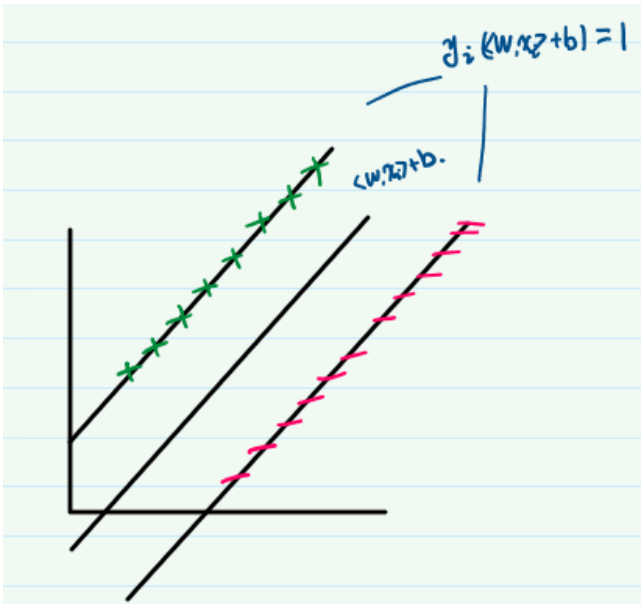

Figure 1: Caricature of Data points

The parameter vector solution to the SVM problems can be written as:

$$\mathbf{w} = -\frac{1}{2}\mathbf{x}_{1999} + \frac{1}{2}\mathbf{x}_{1998}$$

So only two points are required (i.e., only two points are the support vectors)

The hard-margin SVM will have some issues with the dataset B. It runs forever. This is because that the data points in dataset B are not linearly seperable, resulting the hard-margin SVM never halt. We can also tell that these data points are not linearly seperable by the fact that the logistic regression succeeded. Since if they are linearly seperable, that means perfect seperation and logistic regression will complain.

192 points are used to express the parameter vector as a linear combination of them. These points are the support vectors. I got this result by calling soft_svc.support_ to get the list of indices of all support vectors, then count the length of this list. As for the accuracy, there are 57 wrong predictions for the soft-margin SVC method and 571 wrong predictions for the logistic regression method, so the test loss for soft-margin SVC is around 0.0285 and for logistic regression is around 0.2855.

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 3: Support Vector Regression (8 pts)

Let us consider support vector regression:

$$\min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n} \max\{|y_i - (\mathbf{w}^\top \mathbf{x}_i + b)| - \varepsilon, 0\}, \tag{4}$$

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 (4). Please include intermediate steps so that you can get partial credit.

   Ans: We can rewrite the loss function as:

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

   to get rid of the absolute value. Then, we can introduce a slack variable γ_i:

$$\min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}, \gamma in\mathbb{R}^n} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n} \gamma_i \text{ s.t. } \gamma_i \geq y_i - (\mathbf{w}^\top \mathbf{x}_i + b) - \varepsilon \text{ and } \gamma_i \geq \mathbf{w}^\top \mathbf{x}_i + b - y_i - \varepsilon \text{ and } \gamma_i \geq 0 \ \forall i$$

   Then we introduce dual variables and take Lagrangian:

$$\max_{\alpha_i,\beta_i,\tau_i\in\mathbb{R}^n, \alpha_i,\beta_i,\tau_i\geq 0} \min_{\mathbf{w},b,\gamma} \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{n}(C\gamma_i + \alpha_i(y_i - (\mathbf{w}^\top \mathbf{x}_i + b) - \varepsilon - \gamma_i) + \beta_i(\mathbf{w}^\top \mathbf{x}_i + b - y_i - \varepsilon - \gamma_i) - \tau_i\gamma_i)$$

   Then, let us take the derivative of this expression with respect to each variable and set them to zero:

$$F = \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{n}(C\gamma_i + \alpha_i(y_i - (\mathbf{w}^\top \mathbf{x}_i + b) - \varepsilon - \gamma_i) + \beta_i(\mathbf{w}^\top \mathbf{x}_i + b - y_i - \varepsilon - \gamma_i) - \tau_i\gamma_i)$$

$$\frac{\partial F}{\partial \mathbf{w}} = \mathbf{w} + \sum_{i=1}^{n}\left(\frac{\partial}{\partial \mathbf{w}}\left(-\alpha_i \mathbf{w}^\top \mathbf{x}_i + \beta_i \mathbf{w}^\top \mathbf{x}_i\right)\right) = \mathbf{w} - \sum_{i=1}^{n}(\alpha_i - \beta_i)\mathbf{x}_i = 0$$

$$\frac{\partial F}{\partial b} = \sum_{i=1}^{n}\left(\frac{\partial}{\partial b}\left(-\alpha_i b + \beta_i b\right)\right) = \sum_{i=1}^{n}(-\alpha_i + \beta_i) = 0$$

$$\frac{\partial F}{\partial \gamma_i} = \frac{\partial}{\partial \gamma_i}\sum_{j=1}^{n}(C\gamma_j - \alpha_j\gamma_j - \beta_j\gamma_j - \tau_j\gamma_j) = C - \alpha_i - \beta_i - \tau_i = 0$$

   Then we can simplify the expression further with the result above:

$$\begin{aligned}
F &= \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{n}(C - \alpha_i - \beta_i - \tau_i)\gamma_i + \sum_{i=1}^{n}(\alpha_i - \beta_i)y_i + \sum_{i=1}^{n}(-\alpha_i - \beta_i)\varepsilon + \sum_{i=1}^{n}(-\alpha_i + \beta_i)b + \sum_{i=1}^{n}(-\alpha_i + \beta_i)\mathbf{w}^\top \mathbf{x}_i \\
&= \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} 0\gamma_i + \sum_{i=1}^{n}(\alpha_i - \beta_i)y_i - \varepsilon\sum_{i=1}^{n}(\alpha_i + \beta_i) + b\sum_{i=1}^{n}(-\alpha_i + \beta_i) - \sum_{i=1}^{n}(\alpha_i - \beta_i)\mathbf{w}^\top \mathbf{x}_i \\
&= \frac{1}{2}\left\|\sum_{i=1}^{n}(\alpha_i - \beta_i)\mathbf{x}_i\right\|_2^2 + \sum_{i=1}^{n}(\alpha_i - \beta_i)y_i - 2\varepsilon\sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}(\alpha_i - \beta_i)\left(\sum_{j=1}^{n}(\alpha_j - \beta_j)\mathbf{x}_j\right)^\top \mathbf{x}_i \\
&= \frac{1}{2}\left\|\sum_{i=1}^{n}(\alpha_i - \beta_i)\mathbf{x}_i\right\|_2^2 + \sum_{i=1}^{n}(\alpha_i - \beta_i)y_i - 2\varepsilon\sum_{i=1}^{n}\alpha_i - \left(\sum_{j=1}^{n}(\alpha_j - \beta_j)\mathbf{x}_j\right)^\top \sum_{i=1}^{n}(\alpha_i - \beta_i)\mathbf{x}_i \\
&= \frac{1}{2}\left\|\sum_{i=1}^{n}(\alpha_i - \beta_i)\mathbf{x}_i\right\|_2^2 + \sum_{i=1}^{n}(\alpha_i - \beta_i)y_i - 2\varepsilon\sum_{i=1}^{n}\alpha_i - \left\|\sum_{i=1}^{n}(\alpha_i - \beta_i)\mathbf{x}_i\right\|_2^2 \\
&= -\frac{1}{2}\left\|\sum_{i=1}^{n}(\alpha_i - \beta_i)\mathbf{x}_i\right\|_2^2 + \sum_{i=1}^{n}(\alpha_i - \beta_i)y_i - 2\varepsilon\sum_{i=1}^{n}\alpha_i
\end{aligned}$$

So the problem is simplified to:

$$\max_{\alpha_i,\beta_i,\tau_i\in\mathbb{R}^n,\alpha_i,\beta_i,\tau_i\geq 0} -\frac{1}{2}\left\|\sum_{i=1}^{n}(\alpha_i-\beta_i)\mathbf{x}_i\right\|_2^2+\sum_{i=1}^{n}(\alpha_i-\beta_i)y_i-2\varepsilon\sum_{i=1}^{n}\alpha_i$$
$$\text{s.t } \sum_{i=1}^{n}(-\alpha_i+\beta_i)=0 \text{ and } C-\alpha_i-\beta_i-\tau_i=0$$

We can further simplified this by eliminating τ_i because $\tau_i = C-\alpha_i-\beta_i \geq 0$ to get:

$$\max_{\alpha_i,\beta_i\in\mathbb{R}^n,\alpha_i,\beta_i\geq 0,C\geq\alpha_i+\beta_i} -\frac{1}{2}\left\|\sum_{i=1}^{n}(\alpha_i-\beta_i)\mathbf{x}_i\right\|_2^2+\sum_{i=1}^{n}(\alpha_i-\beta_i)y_i-2\varepsilon\sum_{i=1}^{n}\alpha_i$$
$$\text{s.t } \sum_{i=1}^{n}(-\alpha_i+\beta_i)=0$$

or

$$\min_{\alpha_i,\beta_i\in\mathbb{R}^n,\alpha_i,\beta_i\geq 0,C\geq\alpha_i+\beta_i} \frac{1}{2}\left\|\sum_{i=1}^{n}(\alpha_i-\beta_i)\mathbf{x}_i\right\|_2^2-\sum_{i=1}^{n}(\alpha_i-\beta_i)y_i+2\varepsilon\sum_{i=1}^{n}\alpha_i$$
$$\text{s.t } \sum_{i=1}^{n}(-\alpha_i+\beta_i)=0$$

In the following you will complete and implement the following gradient algorithm for solving support vector regression in Equation (4):

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 η
Output: $\mathbf{w}, b$

```
for t = 1, 2, ..., max_pass do
    for i = 1, 2, ..., n do
        choose step size η
        if |y_i − (⟨x_i, w⟩ + b)| ≥ ε then
            w ←                                   // x_i is the i-th row of X
            b ←
        w ←                                       // 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 (4). Note that in places where the function is non-differentiable, you might have to compute a sub-gradient.

$$C\sum_{i=1}^{n}\max\{|y_i-(\mathbf{w}^\top\mathbf{x}_i+b)|-\varepsilon,0\} \tag{5}$$

Ans: Take the loss function in an equivalent form as:

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

We can calculate the gradient of each expression inside the max:

$$\frac{\partial}{\partial \mathbf{w}} y_i - \mathbf{w}^\top \mathbf{x}_i - b - \varepsilon = -\mathbf{x}_i$$
$$\frac{\partial}{\partial \mathbf{w}} \mathbf{w}^\top \mathbf{x}_i + b - y_i - \varepsilon = \mathbf{x}_i$$
$$\frac{\partial}{\partial \mathbf{w}} 0 = 0$$

The gradient can be expressed as:

$$\frac{\partial L}{\partial \mathbf{w}} = C \sum_{i=1}^{n} \varsigma_i \quad \frac{\partial L}{\partial d} = C \sum_{i=1}^{n} \sigma_i$$

where ς_i is:

$$\varsigma_i = \begin{cases} -\mathbf{x}_i & y_i \geq \mathbf{w}^\top \mathbf{x}_i + b \text{ and } y_i - \mathbf{w}^\top \mathbf{x}_i - b - \varepsilon \geq 0 \\ \mathbf{x}_i & y_i \leq \mathbf{w}^\top \mathbf{x}_i + b \text{ and } \mathbf{w}^\top \mathbf{x}_i + b - y_i - \varepsilon \geq 0 \\ 0 & y_i - \mathbf{w}^\top \mathbf{x}_i - b - \varepsilon \leq 0 \text{ and } \mathbf{w}^\top \mathbf{x}_i + b - y_i - \varepsilon \leq 0 \end{cases}$$
$$\sigma_i = \begin{cases} -1 & y_i \geq \mathbf{w}^\top \mathbf{x}_i + b \text{ and } y_i - \mathbf{w}^\top \mathbf{x}_i - b - \varepsilon \geq 0 \\ 1 & y_i \leq \mathbf{w}^\top \mathbf{x}_i + b \text{ and } \mathbf{w}^\top \mathbf{x}_i + b - y_i - \varepsilon \geq 0 \\ 0 & y_i - \mathbf{w}^\top \mathbf{x}_i - b - \varepsilon \leq 0 \text{ and } \mathbf{w}^\top \mathbf{x}_i + b - y_i - \varepsilon \leq 0 \end{cases}$$

we can simplify this a little bit if we consider that $\varepsilon \geq 0$:

$$\varsigma_i = \begin{cases} -\mathbf{x}_i & y_i - (\mathbf{w}^\top \mathbf{x}_i + b) \geq \varepsilon \\ \mathbf{x}_i & y_i - (\mathbf{w}^\top \mathbf{x}_i + b) \leq -\varepsilon \\ 0 & -\varepsilon \leq y_i - (\mathbf{w}^\top \mathbf{x}_i + b) \leq \epsilon \end{cases}$$
$$\sigma_i = \begin{cases} -1 & y_i - (\mathbf{w}^\top \mathbf{x}_i + b) \geq \varepsilon \\ 1 & y_i - (\mathbf{w}^\top \mathbf{x}_i + b) \leq -\varepsilon \\ 0 & -\varepsilon \leq y_i - (\mathbf{w}^\top \mathbf{x}_i + b) \leq \epsilon \end{cases}$$

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

$$\mathsf{P}^\eta(\mathbf{w}) = \underset{\mathbf{z}}{\operatorname{argmin}} \ \frac{1}{2\eta} \|\mathbf{z} - \mathbf{w}\|_2^2 + \frac{1}{2} \|\mathbf{z}\|_2^2 \tag{6}$$

Ans: To find the mininum value, we take the derivatives of this expression:

$$\frac{\partial}{\partial \mathbf{z}} \left(\frac{1}{2\eta} \|\mathbf{z} - \mathbf{w}\|_2^2 + \frac{1}{2} \|\mathbf{z}\|_2^2 \right) = \mathbf{z} + \frac{1}{\eta}(\mathbf{z} - \mathbf{w})$$
$$\frac{\partial^2}{\partial \mathbf{w}^2} \left(\frac{1}{2\eta} \|\mathbf{z} - \mathbf{w}\|_2^2 + \frac{1}{2} \|\mathbf{z}\|_2^2 \right) = \frac{\partial}{\partial \mathbf{w}} \left(\mathbf{z} + \frac{1}{\eta}(\mathbf{z} - \mathbf{w}) \right)$$
$$= 1 + \frac{1}{\eta} > 0$$

The second derivative about $\mathbf{z}$ is positive, so the expression is convex. Therefore, the expression will take

its minimum at the point when we set the first derivative to zero.

$$\mathbf{z} + \frac{1}{\eta}(\mathbf{z} - \mathbf{w}) = 0$$
$$\eta\mathbf{z} + \mathbf{z} - \mathbf{w} = 0$$
$$\mathbf{z} = \frac{\mathbf{w}}{1+\eta}$$

Therefore:

$$\mathsf{P}^{\eta}(\mathbf{w}) = \frac{\mathbf{w}}{1+\eta} \tag{7}$$

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: Please see the implementation in the script file. The report is:
   training error: 610.2011152463025
   training loss: 610.7272490389753
   test error: 775.7902535073246

Exercise 4: 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 $x^T Mx \geq 0$ 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: The corresponding feature map is:

$$\begin{aligned}\phi(x) &= e^{-\alpha x^2}[1, \sqrt{\frac{1}{1!}}(-\sqrt{2\alpha}x), \sqrt{\frac{1}{2!}}(-\sqrt{2\alpha}x)^2, \cdots]^T \\ &= e^{-\alpha x^2} v^T\end{aligned}$$

   where $v^\top$ is a $\mathbb{R}^\infty$ vector and $v_1 = 1$, $v_i = \sqrt{\frac{1}{(i-1)!}}(-\sqrt{2\alpha}x)^{i-1}$ for all other i.
   I prefer solve the dual because we only need to consider about the dot product, or the kernel in the dual representation. Since $\phi(x)$ will be a vector with infinite size, calculating the vector itself will be really expensive in the primal representation.

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: This function is a valid kernel. The feature map is

$$\begin{aligned}\phi(x) &= [1, x, x^2, x^3, \cdots]^\top \\ &= v^\top\end{aligned}$$

   where $v^\top$ is a $\mathbb{R}^\infty$ vector and $v_i = x^{i-1}$ for all integer $i = 1, 2, 3, \cdots$.

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: This is not a valid kernel. We can consider the dataset:

$$x_1 = 1, y_1 = 2, x_2 = 2, y_2 = 1$$

and

$$v = [-1, 1]^\top$$

So the kernel will be:

$$K = \begin{bmatrix} \log(3) & \log(2) \\ \log(5) & \log(3) \end{bmatrix}$$

Therefore:

$$\begin{aligned} v^\top K v &= [-1, 1] \begin{bmatrix} \log(3) & \log(2) \\ \log(5) & \log(3) \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \\ &= [-\log 3 + \log 5, -\log 2 + \log 3] \begin{bmatrix} -1 \\ 1 \end{bmatrix} \\ &= \log 3 - \log 5 - \log 2 + \log 3 \\ &= \log(\frac{9}{10}) < 0 \end{aligned}$$

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: This is not a valid kernel. We can consider the dataset:

$$x_1 = 0, y_1 = \frac{3\pi}{2}, x_2 = \frac{\pi}{2}, y_2 = 0$$

and

$$v = [-1, 1]^\top$$

So the kernel will be:

$$K = \begin{bmatrix} \cos(\frac{3\pi}{2}) & \cos(0) \\ \cos(2\pi) & \frac{\pi}{2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Therefore:

$$\begin{aligned} v^\top K v &= [-1, 1] \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \\ &= [1, -1] \begin{bmatrix} -1 \\ 1 \end{bmatrix} \\ &= -2 < 0 \end{aligned}$$