CS 475 Machine Learning: Homework 3
Supervised Learning 2
Analytical Questions

Due: Monday, October 12, 2020, 11:59 pm US/Eastern

20 Points Total Version 1.1

PARTNER1_NAME (PARTNER1_JHED), PARTER2_NAME (PARTNER2_JHED)

Instructions

We have provided this LaTeX document for turning in this homework. We give you one or more boxes to answer each question. The question to answer for each box will be noted in the title of the box.

Other than your name, do not type anything outside the boxes. Leave the rest of the document unchanged.

Do not change any formatting in this document, or we may be unable to grade your work. This includes, but is not limited to, the height of textboxes, font sizes, and the spacing of text and tables. Additionally, do not add text outside of the answer boxes. Entering your answers are the only changes allowed.

We strongly recommend you review your answers in the generated PDF to ensure they appear correct. We will grade what appears in the answer boxes in the submitted PDF, NOT the original latex file.

Notation

$\mathbf{x_i}$	One input data vector. $\mathbf{x_i}$ is M dimensional. $\mathbf{x_i} \in \mathbb{R}^{1\times M}$. We assume $\mathbf{x_i}$ is augmented with a 1 to include a bias term.
$\mathbf{X}$	A matrix of concatenated $\mathbf{x_i}$'s. There are N input vectors, so $\mathbf{X} \in \mathbb{R}^{N\times M}$
y_i	The true label for input vector $\mathbf{x_i}$. In regression problems, y_i is continuous. In general ,y_i can be a vector, but for now we assume it's a scalar: $y_i \in \mathbb{R}^1$.
$\mathbf{y}$	A vector of concatenated y_i's. There are N input vectors, so $\mathbf{y} \in \mathbb{R}^{N\times 1}$
$\mathbf{w}$	A weight vector. We are trying to learn the elements of $\mathbf{w}$. $\mathbf{w}$ is the same number of elements as $\mathbf{x_i}$ because we will end up computing the dot product $\mathbf{x_i} \cdot \mathbf{w}$. $\mathbf{w} \in \mathbb{R}^{M\times 1}$. We assume the bias term is included in $\mathbf{w}$.
$h((x))$	The true regression function that describes the data.
i.i.d.	Independently and identically distributed.
Bias-variance decomposition	We can write $E_D[(f(x, D) - h(x))^2] =$ $(E_D[f(x, D) - h(x))^2 + E_D[(f(x, D) - E_D[f(x, D)])^2]$ where the first term is the bias squared, and the second term is the variance.
Notes:	In general, a lowercase letter (not boldface), a, indicates a scalar. A boldface lowercase letter, $\mathbf{a}$, indicates a vector. A boldface uppercase letter, $\mathbf{A}$, indicates a matrix.

1) Multi-class Logistic Regression with $L2$ Regularization (10 points)

In class, we have dealt with binary classification,. i.e. the label y is binary $y \in \{0, 1\}$. Now suppose we have more than two classes, and so y is no longer a binary random variable, but instead $y \in \{1, \ldots, C\}$ with C as the number of classes. Multi-class classification tasks are common, for example, labeling an image as to whether it contains a cat, dog, lizard, and so on.

Generalizing from binary (two classes) to multiple classes takes us from the logistic regression we covered in lecture to multi-class logistic regression. In this model, the probability of a label given the data and the weights can be written as:

$$p(y = c \mid \mathbf{x}, \mathbf{W}, \mathbf{b}) = \frac{\exp(b_c + \mathbf{w}_c^T \mathbf{x})}{\sum_{k=1}^{C} \exp(b_k + \mathbf{w}_k^T \mathbf{x})} \tag{1}$$

Here $\mathbf{W}$ is a $D \times C$ weight matrix, where D is the number of dimensions and C is the number of classes. $\mathbf{b}$ is a $1 \times C$ bias vector. For each class c, the terms $\mathbf{w_c}$ and b_c are the weights and bias for predicting that class.[1]

Suppose we write our loss function, including L2 regularization on the weights, as:

$$\sum_{i=1}^{N} \log p(y_i \mid \mathbf{x}_i, \mathbf{W}) - \lambda \sum_{c=1}^{C} ||\mathbf{w}_c||_2^2 \tag{2}$$

(a) Use the probability of a label given the data (Equation (1)) to rewrite the loss function (Equation (2)) in terms of $\mathbf{x_i}, y_i, \mathbf{w}$, and $\mathbf{b}$.

[1] While it shouldn't matter for the purposes of this problem, you can assume that bias term b_0 is fixed to be 0. This ensures a unique solution to the optimization problem.

(b) Using the expression calculated above, compute:

$$\frac{\partial}{\partial w_{c,j}} \left(\sum_{i=1}^{N} \log p(y_i \mid \mathbf{x}_i, \mathbf{W}, \mathbf{b}) - \lambda \sum_{c=1}^{C} ||\mathbf{w}_c||_2^2 \right)$$

Where $w_{c,j}$ is the jth element of the vector of weights for class c.

(c) The MLE estimate for $\mathbf{w}$, denoted $\hat{\mathbf{w}}$ is obtained by setting your derivative from part (b) equal to 0. Show that, after we solve for $\hat{\mathbf{w}}$, we have $\sum_{c=1}^{C} \hat{\mathbf{w}}_{cj} = 0$ for $j = 1 \ldots D$.

2) Fitting A Support Vector Machine by Hand (10 points)

Consider the following dataset with 2 examples, where $x \in \mathbb{R}$ and $y \in \{-1, 1\}$. The two examples in this dataset are $(x_1 = 2, y_1 = -1)$ and $(x_2 = -3, y_2 = 1)$. We will map our one-dimensional x_i to three-dimensional space using the feature vector $\phi(x) = [1, 2x, x^2]^T$.

Recall that we defined the optimization problem for the maximum margin classifier as:

$$\min ||w||^2 \text{ such that}$$
$$y_1(\mathbf{w}^T\phi(x_1) + b) \geq 1 \tag{3}$$
$$y_2(\mathbf{w}^T\phi(x_2) + b) \geq 1 \tag{4}$$

(a) Find a vector (v_1, v_2, v_3) that is perpendicular to the maximum-margin decision boundary in our 3D feature space. Hint: this will be parallel to the optimal $\mathbf{w}$ weight vector for our SVM, but $\mathbf{v}$ can have arbitrary magnitude.

(b) What is the geometric margin of the optimal decision boundary (in our 3D space)?

(c) Remember that we can arbitrarily set our functional margin equal to 1, and then our geometric margin is equal to $\frac{1}{||\mathbf{w}||}$. Use this to solve for $\mathbf{w}$.

(d) Using Equations (3, 4) above to solve for b.

(e) Using your solutions for b and $\mathbf{w}$, write out the classifier's decision function $f(x) = \text{sign}(\ldots)$.