CS 475 Machine Learning: Homework 1
Supervised Classifiers 1

Due: Thursday September 26, 2019, 11:59pm

100 Points Total Version 1.0

Josh Popp (jpopp4)

Instructions

We have provided this LaTeX document for turning 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.

For written answers, replace the **\TextRequired** (**Place Answer Here**) command with your answer. For the following example *Question 0.1* , you would place your answer where **\TextRequired** (**Place Answer Here**) is located,

Place Answer Here

Do not change the height or title of the box. If your text goes beyond the box boundary, it will be cut off. We have given sufficient space for each answer, so please condense your answer if it overflows. The height of the box is an upper bound on the amount of text required to answer the question - many answers can be answered in a fraction of the space. Do not add text outside of the boxes. We will not read it.

For True/False or Multiple Choice questions, place your answers within the defined table. To mark the box(es) corresponding to your answers, replace **\Unchecked** (□) commands with the **\Checked** (☒) command. Do not make any other changes to the table. For example, in *Question 0.2* ,

☒ Logistic Regression

□ Perceptron

For answers that require a single equation, we will provide a specific type of box, such as in the following example *Question 0.3* . Please type the equation where **\EquationRequired** (**Type Equation Here**) without adding any $ signs or **\equation** commands. Do not put any additional text in this field.

$\mathbf{w}$ = **Type Equation Here**

For answers that require multiple equations, such as a derivation, place all equations within the specified box. You may include text short explanations if you wish (as shown in *Question 0.4*). You can put the equations in any format you like (e.g. within \$ or \$\$, the **\equation** environment, the **\align** environment) as long as they stay within the box.

$$
\begin{aligned}
&x + 2 && \text{x is a real number} \\
& && \text{the following equation uses the variable } y \\
&y + 3
\end{aligned}
$$

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.

1) Regularization (26 points)

In class, we discussed adding a regularization penalty term to our objective function. This gives us an optimization problem of the following general form

$$\operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^D} \ell(\mathbf{w}) + \lambda\cdot\Omega(\mathbf{w}) \tag{1}$$

where ℓ is our usual loss function, $\lambda \geq 0$, and

$$\Omega_q(\mathbf{w}) \overset{\text{def}}{=} \sum_{j=1}^{M} |w_j|^q \tag{2}$$

where $q \geq 0$ is a hyper-parameter that can be experimented with. In class, we discussed $q = 2$ and $q = 1$ as they are the most common in practice. This is a nice family of regularization functions because $\Omega_q(\mathbf{w})$ is convex in $\mathbf{w}$ for $q \geq 1$.

(1) (5 points) Show that $\Omega_q(\mathbf{w})$ is convex in $\mathbf{w}$ for $q = 1$ and $q = 2$.[1]

If the Hessian of $\Omega_q(\mathbf{w})$ is positive semi-definite, then $\Omega_q(\mathbf{w})$ is convex

$$\begin{array}{r@{\;}l|r@{\;}l}
\Omega_1(\mathbf{w}) & = \sum_{j=1}^{M} |w_j|^1 & \Omega_2(\mathbf{w}) & = \sum_{j=1}^{M} |w_j|^2 \\
\frac{\partial \Omega_1(\mathbf{w})}{\partial w_i} & = \frac{\partial}{\partial w_i} \sum_{j=1}^{M} |w_j| & \frac{\partial \Omega_2(\mathbf{w})}{\partial w_i} & = \frac{\partial}{\partial w_i} \sum_{j=1}^{M} |w_j|^2 \\
\frac{\partial \Omega_1(\mathbf{w})}{\partial w_i} & = \frac{\partial}{\partial w_i} |w_i| & \frac{\partial \Omega_2(\mathbf{w})}{\partial w_i} & = \frac{\partial}{\partial w_i} |w_i|^2 \\
\frac{\partial \Omega_1(\mathbf{w})}{\partial w_i} & = \operatorname{sign}(w_i) & \frac{\partial \Omega_2(\mathbf{w})}{\partial w_i} & = 2|w_i| \operatorname{sign}(w_i) \\
\frac{\partial (\Omega_1(\mathbf{w}))^2}{\partial^2 w_i} & = 2\delta(w_i) & \frac{\partial (\Omega_2(\mathbf{w}))^2}{\partial^2 w_i} & = 2(\operatorname{sign}(w_i))^2 + 4|w_i|\delta(w_i) \\
\frac{\partial (\Omega_1(\mathbf{w}))^2}{\partial w_i \partial w_j} & = 0 & \frac{\partial (\Omega_2(\mathbf{w}))^2}{\partial w_i w_j} & = 0
\end{array}$$

Both of these Hessian matrices are diagonal, with non-negative entries along the diagonal, therefore they are both positive semi-definite and both Ω_1 and Ω_2 are convex. To avoid worrying about infinite values of the δ function we could get away with assigning $\delta^*(0) \to 0$.

(2) (5 points) If the loss function $\ell(\mathbf{w})$ is convex in $\mathbf{w}$, is $\ell(\mathbf{w}) + \lambda\cdot\Omega(\mathbf{w})$ necessarily convex? Why or why not?

Yes. If both $\ell(\mathbf{w})$ and $\Omega(\mathbf{w})$ (proven in the last question) are convex, we know their Hessian matrices are positive semi-definite, which implies that the Hessian matrix of $\ell(\mathbf{w}) + \lambda\Omega(\mathbf{w})$ ($\lambda \geq 0$) will also be positive semidefinite, since

$$\frac{\partial^2}{\partial w_i w_j}(\ell(\mathbf{w}) + \lambda\Omega(\mathbf{w})) = \frac{\partial^2}{\partial w_i w_j}\ell(\mathbf{w}) + \lambda\frac{\partial^2}{\partial w_i w_j}\Omega(\mathbf{w})$$

$$\frac{\partial^2}{\partial w_i^2}(\ell(\mathbf{w}) + \lambda\Omega(\mathbf{w})) = \frac{\partial^2}{\partial w_i^2}\ell(\mathbf{w}) + \lambda\frac{\partial^2}{\partial w_i w_j}\Omega(\mathbf{w})$$

[1] For a challenge, but no extra points, show that $\Omega_q(\mathbf{w})$ is convex for all $q \geq 1$.

(3) (5 points) In words, what does Ω_0 compute? Why can't we use it in gradient-based optimization?

Ω_0 computes the number of non-zero weights in $\mathbf{w}$. We cannot use this in gradient-based optimization because it is not differentiable, and therefore has no gradient to be utilized for optimization.

(4) (6 points) Consider the following modification to the optimization problem in equation (1): set $q = 2$ but add $\lambda \geq 0$ to the set of variables being optimized.

$$\underset{\mathbf{w} \in \mathbb{R}^D, \lambda \geq 0}{\operatorname{argmin}} \quad \ell(\mathbf{w}) + \lambda \cdot \Omega_2(\mathbf{w}) \tag{3}$$

Why will the optimal value of λ be *zero*?

$\Omega(\mathbf{w})$ is strictly positive (according to its definition), so for any $\lambda > 0$ the term $\lambda \cdot \Omega_2(\mathbf{w})$ is going to increase the value of the sum $\ell(\mathbf{w}) + \lambda \cdot \Omega_2(\mathbf{w})$. Obviously, for $\lambda = 0$, this term will not increase the sum, so the optimal value for λ under the constraint $\lambda \geq 0$ will always be 0.

(5) (5 points) Since we can't seem to optimize λ with training data, how might we modify our experimental setup to enable choosing the parameters λ and q in a principled way?

We can perform a grid search, testing multiple combinations of λ and q, and seeing which gives us the minimum loss. We could also limit our search by seeing what methods have been used on data similar to our own in separate experiments - if others have determined that a specific combination of λ and q work well, we can start there and perform a less thorough search ourselves (though some verification is still necessary).

2) Linear Regression (12 points)

Suppose you observe n data points, (x_1, y_1), $\ldots$, (x_n, y_n), where all x_i and all y_i are *scalars* (i.e., one-dimensional). Suppose further that each data point is paired with an *example weight*, $\alpha_i \geq 0$. These weights can be useful, for example, if some data points should have more (large α_i) or less (small α_i) influence on the loss. Suppose you choose the model $\hat{y} = w \cdot x$ and aim to minimize the α-weighted sum of squares error

$$\frac{1}{2}\sum_{i=1}^{n} \alpha_i (w \cdot x_i - y_i)^2 \tag{4}$$

Derive the closed-form solution for w showing each step. Is the solution necessarily a global minimum? Explain why or why not.

$$\begin{aligned}
\ell &= \frac{1}{2}\sum_{i=1}^{n} \alpha_i (w \cdot x_i - y_i)^2 \\
\frac{\partial \ell}{\partial w} &= \frac{1}{2}\sum_{i=1}^{n} \alpha_i [2(w \cdot x_i - y_i) x_i] \\
0 &= \sum_{i=1}^{n} \alpha_i (w \cdot x_i - y_i) x_i \\
&= w \sum_{i=1}^{n} \alpha_i (x_i)^2 - \sum_{i=1}^{n} \alpha_i y_i x_i \\
\sum_{i=1}^{n} \alpha_i y_i x_i &= w \sum_{i=1}^{n} \alpha_i (x_i)^2 \\
w &= \frac{\sum_{i=1}^{n} \alpha_i y_i x_i}{\sum_{i=1}^{n} \alpha_i (x_i)^2}
\end{aligned}$$

Will this be a global minimum? To answer this we can look at the second derivative

$$\begin{aligned}
\frac{\partial^2 \ell}{\partial w^2} &= \frac{1}{2}\sum_{i=1}^{n} 2\alpha_i (x_i)^2 \\
&= \sum_{i=1}^{n} \alpha_i (x_i)^2
\end{aligned}$$

Since $\alpha_i \geq 0$ and $(x_i)^2 \geq 0$, the second derivative of the loss with respect to w is a nonnegative constant. Therefore, there will not any solution to w that leads to a *lower* loss ℓ. If it's positive, the function is convex and w will be a global minimum. The only way for the second derivative to be equal to zero is for the sum $\sum_{i=1}^{n} \alpha_i (x_i)^2$ to be equal to 0, in which case we already would have run into an error trying to divide by 0 to determine w. This shouldn't be a problem, though: if all the α values are equal to zero, then you're saying not to even look at x in your assignment of labels. If all your x values are equal to zero, you have no hope of identifying a non-trivial pattern. If the sum is zero due to a combination of the two, you need to redefine the α values you've assigned.

3) Support vector machines (12 points)

In this question, we will ask you to extend the slack formulation of the support vector machine to allow for asymmetric costs for misclassification. Consider the following scenario, a doctor using a classifier to predict whether or not they should order more tests ($y = +1$) or triage the patient ($y = -1$) based on a preliminary set of tests they have already done (i.e., features). Clearly, we prefer to have more information that can be provided by additional tests, however, tests carry some risk and may be unnecessary. Mathematically, what we have is an asymmetry between *false-positives* and *false-negatives*.

Extend the slack formulation of the SVM from class to penalize the slack variables for *false-positives* and *false-negatives* differently. Rather than a single $C \geq 0$ coefficient, the new formulation should leverage two coefficients $C^{(+)}, C^{(-)} \geq 0$.

For reference, here is the slack formulation. Feel free to copy-paste and modify it.

$$\underset{w \in \mathbb{R}^D}{\text{minimize}} \ \frac{1}{2}||w||^2 + C \sum_{i=1}^{n} \xi_i \tag{5}$$

$$\text{subject to} \quad (w^\top x_i) y_i + \xi_i \geq 1, \quad i = 1 \ldots n \tag{6}$$

$$\xi_i \geq 0, \quad i = 1 \ldots n \tag{7}$$

$$\underset{w \in \mathbb{R}^D}{\text{minimize}} \ \frac{1}{2}||w||^2 + C^+ \sum_{y_i=-1} \xi_i + C^- \sum_{y_i=1} \xi_i$$

$$\text{subject to} \quad (w^\top x_i) y_i + \xi_i \geq 1, \quad i = 1 \ldots n$$

$$\xi_i \geq 0, \quad i = 1 \ldots n$$

Why does this work?

If we are correct (ie sign($w^\top x_i$)= y_i):
Since we are trying to find a minimum, we will set $\xi_i = 0$ and no penalty will be applied (the penalty terms are strictly positive so they will be avoided whenever possible)

If we have a false positive:
A false positive means that $y_i = -1$, $\hat{y_i} = 1$. Since we got it wrong, we'll be leaning on a ξ_i to satisfy our first constraint. In our optimization we know that $y_i = -1$ so we'll be applying the C^+ (cost for a false positive) to this ξ_i, as desired

If we have a false negative:
A false negative means that $y_i = 1$, $\hat{y_i} = -1$. Since we got it wrong, we'll be leaning on a ξ_i to satisfy our first constraint. In our optimization we know that $y_i = 1$ so we'll be applying the C^- (cost for a false negative) to this ξ_i, as desired