Problem Set #4

ECE 4424 / CS 4824 - Machine learning
Virginia Tech

November 2, 2020

- Feel free to collaborate with other classmates in doing the homework. Please indicate your collaborators with their student ID. You should, however, write down your solution yourself. Please try to keep the answers brief and clear.
- Whenever you need clarification, please post the related questions on Piazza under the corresponding homework folder.
- Total: 100 points
- **Due date: 11/13/2020, 11:59PM ET**
- Late submission: each student will have a total of **four** free late (calendar) days to use for homeworks. **Note: this is the total number of late days accumulated through all the homeworks so far. It's NOT reset after each homework!** Once these late days are exhausted, any assignments turned in late will be penalized 20% per late day. However, no assignment will be accepted more than three days after its due date. Each 24 hours or part thereof that a homework is late uses up one full late day.

1 Perceptron algorithm: proof of convergence (40 pts)

Recall that the perceptron algorithm iteratively finds a linear decision boundary for binary classification. Given a dataset $\{(x_1, y_1), ..., (x_n, y_n)\}$ where $x_i \in \mathbb{R}^d$ are feature vectors and $y_i \in \{-1, +1\}$ are labels. The linear classifier is parametrized by $\theta \in \mathbb{R}^d$ (for simplicity, we assimilate the intercept into the parameters θ), and predicts +1 at a point x if $\theta \cdot x > 0$ and -1 otherwise. The perceptron algorithm is given in Algorithm 1.

In this problem, we are going to go through the proof for the convergence of Perceptron algorithm. Your job is to read the proof and indicate the reasoning for some key steps. The theorem of convergence is shown below:

Algorithm 1 Perceptron algorithm

```
Require: Dataset $\{(x_1, y_1), ..., (x_n, y_n)\}$
Initialize $\theta^1 = 0$, iteration number $t = 1$, and number of updates $m_{\text{update}} = 0$.
while We encounter a point $(x, y)$ that is misclassified do
    Update $\theta^{t+1} \leftarrow \theta^t + yx$ (Here, we denote $\theta^t$ as the parameter $\theta$ before the $t$-th update.)
    Update iteration number $t \leftarrow t + 1$
    Increment the counter for updates: $m_{\text{update}} \leftarrow m_{\text{update}} + 1$
end while
```

Theorem (Convergence of Perceptron algorithm). Assume that there exists some parameter vector θ^* such that

$$\|\theta^*\| = 1 \tag{1}$$

(note that for any vector x, $\|x\|$ refers to the Euclidean norm of x, i.e., $\|x\| = \sqrt{\sum_j (x^{(j)})^2}$), and some scalar $\gamma > 0$ such that for all $i = 1, ..., n$,

$$y_i(x_i \cdot \theta^*) \geq \gamma \tag{2}$$

(note the above condition simply means that we can find a linear classifier that perfectly classify every data point with a margin γ.) Also assume that for all $i = 1, ..., n$, the feature x_i is bounded by a constant $R > 0$:

$$\|x_i\| \leq R \tag{3}$$

During the learning process, whenever we found a data that is misclassified, we need to update the parameters. The number of times we update the parameters is given by m_{update} in Algorithm 1. It can be proved that the perceptron algorithm makes at most $\frac{R^2}{\gamma^2}$ updates:

$$m_{\text{update}} \leq \frac{R^2}{\gamma^2} \tag{4}$$

Proof. First, we define θ^t to be the parameter vector when the algorithm makes its t-th error (thus needs to update the parameters). You can also think of it as the parameter θ before the t-th update (as indicated in Algorithm 1). Note that we have

$$\theta^1 = 0 \tag{5}$$

Next, assume the t-th error is made on example k, we have

$$\begin{aligned}
\theta^{t+1} \cdot \theta^* &= (\theta^t + y_k x_k) \cdot \theta^* && (6)\\
&= \theta^t \cdot \theta^* + y_k x_k \cdot \theta^* && (7)\\
&\geq \theta^t \cdot \theta^* + \gamma && (8)
\end{aligned}$$

From the above relationship, we can derive that

$$\theta^{t+1} \cdot \theta^* \geq t\gamma \tag{9}$$

From the above, we can show that

$$\|\theta^{t+1}\| \geq t\gamma \tag{10}$$

In the second part of the proof, we will derive an upper bound on $\|\theta^{t+1}\|$. We have

$$\begin{aligned}
\|\theta^{t+1}\|^2 &= \|\theta^t + y_k x_k\|^2 & (11)\\
&= \|\theta^t\|^2 + y_k^2\|x_k\|^2 + 2y_k x_k \cdot \theta^t & (12)\\
&\leq \|\theta^t\|^2 + R^2 & (13)
\end{aligned}$$

From the above, we can show that

$$\|\theta^{t+1}\|^2 \leq tR^2 \tag{14}$$

STUDENT: Finish the rest of the proof. □

Read and understand the above proof and answer the following questions. Hint: for most questions, you need to closely examine the algorithm (listed in Algirthm 1) and the assumptions Equ. (1), (2) and (3) made in the Theorem.

a) (3 pts) Explain why it is true for Equ. (5).

b) (3 pts) Explain why it is true for Equ. (6).

c) (4 pts) Explain why it is true for Equ. (8).

d) (8 pts) Explain why it is true for Equ. (9).

e) (6 pts) Explain why it is true for Equ. (10). Hint: you can use the Cauchy–Schwarz inequality $x \cdot z \leq \|x\|\|z\|$ for $x, z \in \mathbb{R}^d$.

f) (6 pts) Explain why it is true for Equ. (13).

g) (10 pts) Finish the rest of the proof.

2 Programming assignment (60 pts)

For the following programming assignment, please download the datasets and iPython notebooks from Canvas and submit the following:

- Completed and ready-to-run iPython notebooks. Note: we will inspect the code and run your notebook if needed. If we cannot run any section of your notebook, you will not receive any points for the task related to that section.

- Responses (texts, codes, and/or figures) to the following problems/tasks

In this programming exercise, you will build a soft-margin support vector machine model for sentiment analysis.

Recall that support vector machine (SVM) finds a linear decision boundary with the largest margin for a binary classification problem. Suppose we have a training dataset $\{(x_1, y_1), ..., (x_n, y_n)\}$ where $x_i \in \mathbb{R}^d$ are feature vectors and $y_i \in \{-1, +1\}$ are labels. The linear classifier is parametrized by $\theta \in \mathbb{R}^d$ and $\theta_0 \in \mathbb{R}$, and predicts +1 at a point x if $\theta \cdot x + \theta_0 > 0$ and -1 otherwise.

To train a soft-margin SVM, we need to solve the following constrained optimization problem:

$$\min_{\theta\in\mathbb{R}^d,\theta\in\mathbb{R},\xi_1,...,\xi_n} \quad \|\theta\|^2 + C\sum_{i=1}^{n} \xi_i \tag{15a}$$

$$\text{s.t.} \quad y_i(\theta \cdot x_i + \theta_0) \geq 1 - \xi_i, \quad \text{for } i = 1, ..., n \tag{15b}$$

$$\xi_i \geq 0, \quad \text{for } i = 1, ..., n \tag{15c}$$

where $\theta \in \mathbb{R}^d, \theta \in \mathbb{R}$ are the parameters of a linear decision boundary, and ξ_i is the slack variable assigned to each data point i.

It turns out that the above optimization is equivalent to the following unconstrained optimization:

$$\min_{\theta\in\mathbb{R}^d,\theta\in\mathbb{R}} \quad \|\theta\|^2 + C\sum_{i=1}^{n} \ell_{\text{hinge}}(y_i(\theta \cdot x_i + \theta_0)) \tag{16}$$

where $\ell_{\text{hinge}}(t) = \max(0, 1 - t)$ is called the "hinge loss," which takes value $1 - t$ if $t < 1$ and 0 otherwise. For example, $\ell_{\text{hinge}}(-1) = 2$, and $\ell_{\text{hinge}}(2) = 0$.

Task P1 (10 pts): Reason why optimization (15) and (16) are equivalent.

Hint: The following idea is often used in optimization (also known as the "epigraph" idea). Suppose you want to find the optimal solution θ of the optimization:

$$\mathit{min}_{\theta\in\mathbb{R}^d,t} \quad t \tag{17a}$$

$$\mathit{s.t.} \quad t \geq h(\theta) \tag{17b}$$

where $h(\theta)$ is any function of θ, it is equivalent to solve the following unconstrained optimization:

$$\mathit{min}_{\theta\in\mathbb{R}^d} \quad h(\theta) \tag{18}$$

The reason is intuitive. For (17)*, by the constraint $t \geq h(\theta)$, we know that for a given θ, $h(\theta)$ will be the lower bound for t. So if we find a parameter θ, then to minimize the objective, we should just set $t = h(\theta)$. Since we want to find the minimum t possible, all we need is to find the θ such that $h(\theta)$ is minimum, and set t to that number. In other words, the inequality constraint* (17b) *at optimal will become equality, so* (17) *is equivalent to:*

$$\begin{aligned} &\underset{\theta\in\mathbb{R}^d, t}{min} \quad t && \text{(19a)} \\ &s.t. \quad\; t = h(\theta) && \text{(19b)} \end{aligned}$$

which is equivalent to (18) *at optimal by simply replace t with $h(\theta)$ in the objective. Note that here, we only care about θ. Once we get an optimal θ, then we can simply find the optimal t by setting it equal to $h(\theta)$. This gist of this idea can be used to reason about the equivalence between* (15) *and* (16)*.*

It turns out that for gradient-based optimization, hinge loss may be difficult to deal with because it is not differentiable at point $t = 1$. One solution is to use the "smoothed version" of hinge loss:

$$\ell_{\text{smooth-hinge}}(t) = \begin{cases} \frac{1}{2} - t & \text{if } \; t \leq 0, \\ \frac{1}{2}(1-t)^2 & \text{if } \; 0 < t < 1, \\ 0 & \text{if } \; 1 \leq t \end{cases}$$

Thus, in the rest of the problem, we will consider the following optimization:

$$\min_{\theta\in\mathbb{R}^d, \theta\in\mathbb{R}} \qquad \|\theta\|^2 + C\sum_{i=1}^{n} \ell_{\text{smooth-hinge}}(y_i(\theta \cdot x_i + \theta_0)) \tag{20}$$

Task P2 (8 pts): Implement the hinge loss function and the smooth hinge loss function. Plot the function $\ell_{\text{hinge}}(t)$ and $\ell_{\text{smooth-hinge}}(t)$for $t \in [-5, 5]$.

Task P3 (8 pts): Find the derivative of the smoothed hinge loss function $\ell_{\text{smooth-hinge}}(t)$. Hint: you need to consider the gradients separately where $t \leq 0$, $0 < t < 1$, and $t \geq 1$

Task P4 (10 pts): Find the derivative of the hinge loss function $\ell_{\text{smooth-hinge}}(y_i(\theta \cdot x_i + \theta_0))$ with respect to θ and θ_0. Hint: you need to consider the gradients separately where $y_i(\theta \cdot x_i + \theta_0) \leq 0$, $y_i(\theta \cdot x_i + \theta_0) \in (0, 1)$, and $y_i(\theta \cdot x_i + \theta_0) \geq 1$.

Task P5 (10 pts): Let $f(\theta, \theta_0) = \|\theta\|^2 + C\sum_{i=1}^{n} \ell_{\text{smooth-hinge}}(y_i(\theta \cdot x_i + \theta_0))$ be the objective function of (20). Implement the function that obtains the partial derivative $\frac{\partial}{\partial\theta} f(\theta, \theta_0)$ and $\frac{\partial}{\partial\theta_0} f(\theta, \theta_0)$. Also, print out the output of the code that calculates the derivatives at $\theta = 1$ and $\theta_0 = 1$ with $C = 1$. Hint: you need to calculate the partial derivative of the smoothed hinge loss for each data point separately, and add them together to obtain the result.

Task P6 (10 pts): For sentiment analysis data, choose a value for the trade-off parameter C in (20). Report the training error at convergence and the testing error.

Task P7 (4 pts): List 4 example sentences that are correctly classified by SVM, and 4 example sentences that are incorrectly classified by SVM. Explain what you have found.