Intro to Machine Learning Homework Assignment 1

Goktug Saatcioglu

NetId: gs2417

1. It is okay to weigh each per-example cost equally since we are using a Monte Carlo method to approximate the expected cost. Already, by the law of large numbers, if the random samples x are i.i.d (idependently and identically distributed) then the empirical cost should converge to the expected cost. Because of this, even if every data x is not equally likely to occur, taking the per-example costs and dividing by N is an reasonable approach. Furthermore, most of the time (for most of the cases) we do not have access to the input distribution p_x. The Monte Carlo method allows us to estimate the distribution p_x without knowing p_x itself. If we were to weigh each per-example cost differently according to what we think (or estimate) how likely each x is then we may introduce bias into our dataset or, if we feel the need to introduce per-example costs, possibly due to an imbalanced dataset, then it may be so that the dataset is not representative of the input distribution rather than there being a specific likelihood of occurrence for each x (note: in some cases it may not even be necessary to process an imbalanced dataset).

2. The bias b is necessary because the decision boundary (hyperplane) determined by our perceptron may not always occur through the origin. Thus, the bias shifts the decision boundary by some b away from the origin of our space. For example, if our space were to be $\mathbb{R}^2$, then b would be an horiziontal translation of the hyperplane w^T by the amount of b. The sign function is used to determine whether our perceptron classifies an input x, where $x \in \mathbb{R}^b$, as a positive or negative classification. The perceptron is a binary classifier, meaning all x must be either classified negatively or positively. For example, in the space $\mathbb{R}^2$, consider an $n \in \mathbb{N}$ amount of points where each point $x_i = (x_{i1}, x_{i2})$ is such that $x_{i1} > 0$ and $x_{i2} > 0$ for all i. Assume that some points are labelled negatively and some points are labelled positively such that they are linearly seperable. If we do not include the bias term, i.e. $b = 0$, then $sign(w^T x)$ will make a large amount of wrong classifications for each iteration of the perceptron algorithm since the seperation does not occur through the origin. By our construction, we know that the seperation does not occur through the origin as $\forall x \quad x > 0$. This will lead to having an high error rate for our perceptron. Therefore, the bias is necessary in situations where the decision boundary does not go through the origin.

3. The trivial solution occurs when $w^T x + b = 0$ (or $w^T x = -b$) meaning $M(x) = 0$ but the reference machine $M^*(x) = 1$. This means that the distance function should not be 0 as M^* and M disagree but this won't be the case since $w^T x + b = 0 \implies D(M^*(x), M, x) = 0$. A solution to the trivial solution should involve forcing a value on the distance function when the trivial solution occurs so that the cost function can take this misclassification into account. Consider the following if-else check.

```
**if** $w^T x + b = 0 \wedge M^*(x) \neq M(x)$ **then** $D(M^*(x), M, x) = -(M^*(x) - M(x))(-M^*(x)) = M^*(x)$
**else** $D(M^*(x), M, x) = -(M^*(x) - M(x))(w^T x + b)$
**end if**
```

Since the distance function should be positive for all disagreements so that we can optimize the cost function, we multiply $-(M^*(x) - M(x))$ by $(-M^*(x))$ which will always evaluate to $M^*(x)$ when the trivial solution occurs. Thus, checking for the trivial solution and then setting $D(M^*(x), M, x) = M^*(x)$ if the trivial solution conditions occur will work. This way we can make sure to take into account that an update to the weight vector should be made when the trivial solution occurs and the possibility of a trivial solution is taken into account for the perceptron learning rule.

4. Consider the following property of the sigmoid function σ.

$$\begin{aligned}
\frac{\partial}{\partial a}\sigma(a) &= \frac{\partial}{\partial a}(\frac{1}{1+\exp(-a)}) \\
&= \frac{\exp(-a)}{(1+\exp(-a))^2} \\
&= (\frac{1}{1+\exp(-a)})(\frac{\exp(-a)}{1+\exp(-a)}) \\
&= \sigma(a)(\frac{1+\exp(-a)}{1+\exp(-a)} - \frac{1}{1+\exp(-a)}) \\
&= \sigma(a)(1-\sigma(a)) \\
\therefore \frac{\partial}{\partial a}\sigma(a) &= \sigma(a)(1-\sigma(a)) \qquad (4.1)
\end{aligned}$$

Using 4.1 we now derive the gradient of the distance function of logistic regression with respect to the weight vector w.

$$\begin{aligned}
D(y^*, w, x) &= -(y^* \log(M(x)) + (1-y^*)\log(1-M(x))) \\
&= -(y^* \log(\sigma(wx)) + (1-y^*)\log(1-\sigma(wx))) \\
\frac{\partial}{\partial w}D(y^*, w, x) &= \frac{\partial}{\partial w} - (y^* \log(\sigma(wx)) + (1-y^*)\log(1-\sigma(wx))) \\
&= -(y^* \frac{\partial}{\partial w}(\log(\sigma(wx))) + (1-y^*)\frac{\partial}{\partial w}(\log(1-\sigma(wx)))) && \text{by linearity} \\
&= -(y^* \frac{\frac{\partial}{\partial w}\sigma(wx)}{\sigma(wx)} + (1-y^*)\frac{\frac{\partial}{\partial w}1-\sigma(wx)}{1-\sigma(wx)}) && \text{by chain rule} \\
&= -(y^* \frac{\sigma(wx)(1-\sigma(wx))\frac{\partial}{\partial w}(wx)}{\sigma(wx)} + (1-y^*)\frac{-(\sigma(wx)(1-\sigma(wx))\frac{\partial}{\partial w}(wx))}{1-\sigma(wx)}) && \text{by property 4.1} \\
& && \text{and chain rule} \\
&= -(y^*(1-\sigma(wx))(\frac{\partial}{\partial w}(wx)) + (1-y^*)(-\sigma(wx))(\frac{\partial}{\partial w}(wx))) && \text{simplify fractions} \\
&= -(y^*(\frac{\partial}{\partial w}(wx)) - y^*\sigma(wx)(\frac{\partial}{\partial w}(wx)) - \sigma(wx)(\frac{\partial}{\partial w}(wx)) + y^*\sigma(wx)(\frac{\partial}{\partial w}(wx))) && \text{expand terms} \\
&= -(y^* - \sigma(wx))(\frac{\partial}{\partial w}(wx)) && \text{simplify} \\
&= -(y^* - \sigma(wx))(x) && \text{evaluate } \frac{\partial}{\partial w} \\
&= -(y^* - M(x))(x) && \sigma(w,x) = M(x) \\
\therefore \nabla_w D(y^*, w, x) &= -(y^* - M(x))x \qquad (4.2)
\end{aligned}$$

Thus, the gradient of the distance function of logistic regression with respect to the weight vector is given by Eq. 4.2. We can also verify Eq. 4.2 with Eq. (1.14) as given in the lecture notes.

$$\begin{aligned}
\nabla_w D(y^*, w, x) &= -(y^* - M(x))x && y^* = M^*(x) \\
&= -(M^*(x) - M(x))x && \text{which equals Eq. (1.14)}
\end{aligned}$$

5. See Perceptron1.ipnyb and Logtistic%20Regression%20.ipnyb.