Chloe Quinto
CPE 695 WS
Homework 2
February 25th 2020
I pledge my honor that I have abided by the Stevens Honor System - Chloe Quinto

1. **Prove Bayes' Theorem. Briefly explain why it is useful for machine learning problems i.e. by converting posterior probability to likelihood and prior probability.**

Bayes' Theorem describes the probability of an event based on prior knowledge conditions that may be related to the event. The following is the equation for Bayes' Theorem

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \quad \text{where,}$$

$$A, B = events$$
$$P(A|B) = probability\ of\ A\ given\ B\ is\ true$$
$$P(B|A) = probability\ of\ B\ given\ A\ is\ true$$
$$P(A),\ P(B) = the\ independent\ probabilities\ of\ A\ and\ B$$

Proof:

The probability of two events A and B happening $P(A \cap B)$ is the probability of A P(A) times the probability of B given A occured $P(B|A)$ is given by this equation:

$$P(A \cap B) = P(A) * P(B|A)$$

In a similar idea,

$$P(A \cap B) = P(B) * P(A|B)$$

Setting these two equations equal to each other

$$P(B) * P(A|B) = P(A) * P(B|A)$$

Therefore,

$$P(A|B) = \frac{P(A) * P(B|A)}{P(B)}$$

Applications in Machine Learning,

Bayes’ Theorem applies heavily in machine learning applications. Bayes’ Theorem provides a probabilistic model to describe the relationship between data (D) and hypothesis (h). For example:

$$P(h|D) = \frac{P(D|h) * P(h)}{P(D)} \quad \text{where,}$$

$$P(h|D) = Posterior\ probability\ of\ the\ hypothesis$$
$$P(h) = prior\ probability\ of\ the\ hypothesis$$

This is a useful framework for thinking about machine learning models. The core idea is that testing different models on a dataset can be thought of estimating probability of each hypothesis being true given the observed data. Additionally, seeking the hypothesis with the maximum posterior probability in modeling is called maximum a posteriori or MAP. There are other applications of Bayes’ Theorem such as Bayes’ Theorem for Classification and Naive Bayes’ Theorem.

Classification is a predictive model that involves assigning a label to an input data. Bayes’ Theorem for classification uses the same equation:

$$P(class|data) = \frac{P(data|class) * P(class)}{P(data)}$$

which is finding the class label given the data.

2. Prove the closed form solution for Ridge Regression is $w = (\lambda I + X^T * X)^{-1} * X^T * y$

Ridge Regression

Closed Form: $w = (\lambda I + X^T * X)^{-1} * X^T * y$

Cost Function: $E(w) = MSE(w) + \frac{\lambda}{2}\sum_{i=1}^{m} w_i^2$

Normal Equation

Closed Form: $w = (X^T * X)^{-1} * X^T * y$

Cost Function: $MSE(w) = \sum_{i=1}^{m} (w^T * x^i - y^i)^2$

Proof

Cost function of the Ridge Regression

$$E(w) = MSE(w) + \frac{\lambda}{2}\sum_{i=1}^{m} w_i^2 ,$$

adding in the normal equation cost function

$$E(w) = \sum_{i=1}^{m} (w^T * x^i - y^i)^2 + \frac{\lambda}{2}\sum_{i=1}^{m} w_i^2 ,$$

The left hand side can be simplified from the proof of the normal form of the linear regression

$\sum_{i=1}^{m} (w^T * x^i - y^i)^2$ now becomes $J(w) = (xw - y)^T (xw - y)$

Similarly on the right hand side,

$$(xw - y)^T (xw - y) + \lambda w^T w$$

We want to minimize.

Note that

$$\frac{\delta (xw-y)^T (xw-y)}{\delta w} = -2x\ (y - w^T x) \text{ and } \frac{\delta(\lambda w^T w)}{\delta w} = 2\lambda w$$

Together,

$$x^T y = x^T x w + \lambda w$$

Then isolating w yields

$$w = (\lambda I + x^T x) * x^T * y$$

3. **Recall the multiclass SoftMax Regression model on page 16 of Lecture 3-3. Assume we have K different classes. The posterior probability is**

$$\widehat{p}_k = \delta(s_k(x))_k = \frac{exp(s_k(x))}{\sum_{j=1}^{k} exp(s_j(x))}$$

for k = 1,2,...K where

$s_k(x) = \theta^T * x$ and input x is an n-dimensional vector

   a. **To learn this SoftMax Regression model, how many parameters we need to estimate? What are these parameters?**

   If we replace $s_k(x)$ with $\theta^T * x$, we see the equation:

$$\widehat{p}_k = \delta(s_k(x))_k = \frac{exp(\theta_K{}^T * x)}{\sum_{j=1}^{k} exp(\theta_j{}^T * x)}$$

Here, our parameters are $\theta_1,\ \theta_2, ..., \theta_K\ \varepsilon\ \Re^n$ are the parameters of the model where K is the number of classes. Therefore, we need to estimate *K* number of parameters.

Usually, θ is denoted by an n-by-K matrix

$$\theta = [\theta_1\ \theta_2\ _{....}\ \theta_k]$$

The parameters θ is (on a high level) the sum of the score of each occurring element in the vector.

   b. **Consider the cross-entropy function $J(\Theta)$ (see page 16 of lecture 3-3) of m training samples $\{(x_i, y_i)\}_{i = 1,2,...m}$. Derive the gradient of $J(\Theta)$ regarding to θ_k as shown in page 17 of lecture 3-3**

$$J(\Theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{k} y_k{}^{(i)}\ log(\widehat{p}_k{}^{(i)})$$

then we want

$$\nabla J(\Theta) = \frac{1}{m} \sum_{i=1}^{m} (\widehat{p_k}^{(i)} - y_k{}^{(i)})x^{(i)}$$

We know that $\widehat{p}_k = \delta(s_k(x))_k = \frac{exp(\theta_K{}^T * x)}{\sum_{j=1}^{k} exp(\theta_j{}^T * x)}$, so let's replace the cross entropy cost function

$$J(\Theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{k} y_k{}^{(i)}\, log(\frac{exp(\theta_K{}^T * x^{(i)})}{\sum_{j=1}^{k} exp(\theta_j{}^T * x^{(i)})})$$, we can simplify this

$$J(\Theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{k} y_k{}^{(i)}[\, log(exp(\theta_j{}^T * x^{(i)})) - log(\sum_{j=1}^{k} exp(\theta_j{}^T * x^{(i)}))]$$

$$J(\Theta) = -\frac{1}{m}(\sum_{i=1}^{m}\sum_{k=1}^{k} y_k{}^{(i)}\, log(exp(\theta_j{}^T * x^{(i)})) + \sum_{i=1}^{m}\sum_{k=1}^{k} y_k{}^{(i)}\; log(\sum_{j=1}^{k} exp(\theta_j{}^T * x^{(i)})))$$

We know that the $log(e^x) = x$

$$J(\Theta) = -\frac{1}{m}(\sum_{i=1}^{m}\sum_{k=1}^{k} y_k{}^{(i)} * \theta_j{}^T * x^{(i)} + \sum_{i=1}^{m}\sum_{k=1}^{k} y_k{}^{(i)}\, log(\sum_{j=1}^{k} exp(\theta_j{}^T * x^{(i)})))$$

Now taking the gradient

$$\nabla_{\theta_k} J(\Theta) = \frac{1}{m}\sum_{i=1}^{m} y_k{}^{(i)} * x + \sum_{i=1}^{m}\frac{1}{\sum_{j=1}^{k} exp(\theta_j{}^T * x)} exp(\theta_k{}^{Tx^{(i)}}) * x^{(i)}$$

We simplify

$$\nabla_{\theta_k} J(\Theta) = \frac{1}{m}\sum_{i=1}^{m}(\frac{exp(\theta_K{}^T * x)}{\sum_{j=1}^{k} exp(\theta_j{}^T * x)} - y_k{}^{(i)})\, x^{(i)}$$

Since we know that $\widehat{p}_k = \delta(s_k(x))_k = \frac{exp(\theta_K{}^T * x)}{\sum_{j=1}^{k} exp(\theta_j{}^T * x)}$, we can replace it in the equation

$$\nabla_{\theta_k} J(\Theta) = \frac{1}{m}\sum_{i=1}^{m}(\widehat{p}_k - y_k{}^{(i)})\, x^{(i)}$$