COMP 632: Assignment 1

Due on Tuesday, January 27 2015

Presented to Dr. Doina Precup

Geoffrey Stanley
Student ID: 260645907

Question 1

A)

See source code.

B)

Let a be a vector equal to $[1/max_i|x_j|, ..., 1/max_i|x_n|]$. And let A be a matrix such that $A = aI$. Substituting X by AX in the closed form regression algorithm we obtain:

$$\begin{aligned} 2(AX)^T AXw &= 2(AX)^T Y \\ 2A^T X^T AXw &= 2A^T X^T Y \\ X^T AXw &= X^T Y \\ Aw &= (X^T X)^{-1} X^T Y \end{aligned} \tag{1}$$

As such, we can conclude that normalization is equivalent to multiplying the w vector by some scalar.

C)

See source code.

D)

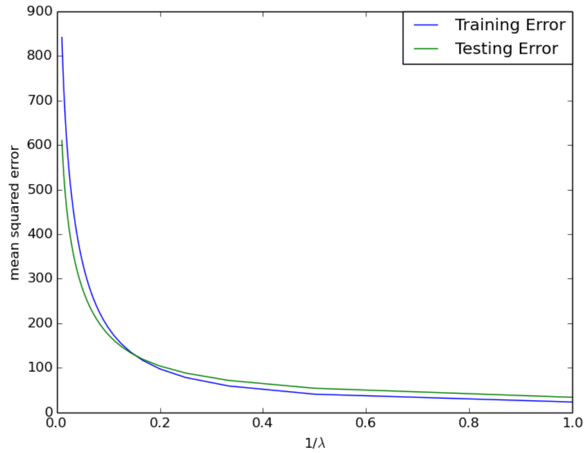
See source code.

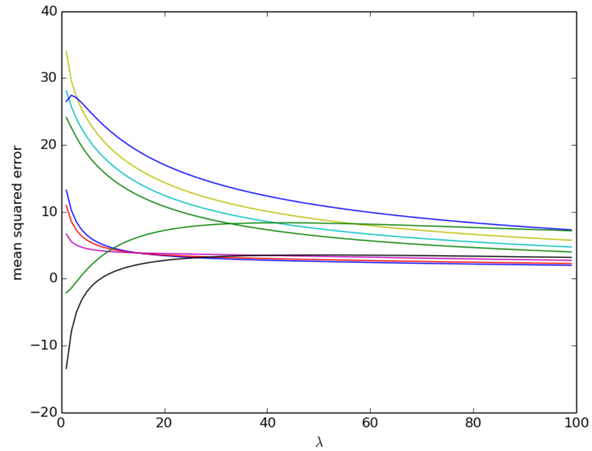
E)

Fold	1	2	3	4	5	Avg	Std
Training Error	1.0728	0.7786	0.9651	0.7456	0.9245		
Testing Error	0.5850	2.0271	0.9211	1.9323	0.9507		
Best Hypothesis Class	3	3	3	3	3		

Fold	Weights
1	54.8, 0.8, -3.8, 120.5, -31.5, -2.1, -2.7
2	52.7, 5.9, -1.5, 114.6, -32.1, -1.2, -2.4
3	51.1, -1.0 , -1.4, 119.5, -31.0, -1.1, -2.7
4	52.9, -5.0, -3.8, 123.2, -30.3, -1.2, -2.9
5	54.2, -5.8, -3.0, 125.3, -31.3, -1.6, -3.2

F)





The graph on the left shows the impact of λ on the mean squared error while the graph on the right shows the impact of λ on the weights associated to the features being used.

These graphs show that as λ increases it puts a greater pressure on the features, decreasing their significance while increasing error. It also shows that there are 5 features that are more predictive then the rest. We can conclude this because the 4 features that quickly approach zero as λ is increased.

Lastly we can see from the graph on the left that the optimum λ will be approximately between 5 and 6.

Question 2

When simplifying the maximum likelihood equation for a regression whose variables maintain a constant normal distribution we are able to obtain the sum-squared-error function. However, when the standard deviation varies from one variable to another this simplification is no longer achievable.

$$L(w) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{1}{2}\left(\frac{y_i - h_w(x_i)}{\sigma_i}\right)^2} \tag{2}$$

$$\log L(w) = \sum_{i=1}^{m} \log\left(\frac{1}{\sqrt{2\pi\sigma_i^2}}\right) - \sum_{i=1}^{m} \frac{1}{2}\left(\frac{y_i - h_w(x_i)}{\sigma_i}\right)^2 \tag{3}$$

$$\log L(w) = \sum_{i=1}^{m} \log\left((2\pi)^{-1/2}\sigma_i^{-1}\right) - \sum_{i=1}^{m} \frac{1}{2}\left(\frac{y_i - h_w(x_i)}{\sigma_i}\right)^2 \tag{4}$$

$$\log L(w) = -\frac{1}{2}\log(2\pi) - \sum_{i=1}^{m} \log\sigma_i - \sum_{i=1}^{m} \frac{1}{2}\left(\frac{y_i - h_w(x_i)}{\sigma_i}\right)^2 \tag{5}$$

$$\frac{\partial}{\partial w_j} \log L(w) = \frac{\partial}{\partial w_j}\left(-\frac{1}{2}\log(2\pi) - \sum_{i=1}^{m} \log\sigma_i - \sum_{i=1}^{m} \frac{1}{2}\left(\frac{y_i - h_w(x_i)}{\sigma_i}\right)^2\right) \tag{6}$$

$$\frac{\partial}{\partial w_j} \log L(w) = -\sum_{i=1}^{m} \frac{x_i}{\sigma_i}\left(\frac{y_i - h_w(x_i)}{\sigma_i}\right) \tag{7}$$

Simplying further while converting standard deviation to variance. Where $\sigma^2 = \Omega$.

$$\frac{\partial}{\partial w_j} \log L(w) = \sum_{i=1}^{m} \frac{x_i y_i}{\Omega_i} - \sum_{i=1}^{m} \frac{h_w(x_i) x_i}{\Omega_i} \tag{8}$$

$$\frac{\partial}{\partial w_j} \log L(w) = X^T \Omega^{-1} Y - w X^T \Omega^{-1} X \tag{9}$$

Now setting the gradient to zero.

$$0 = X^T \Omega^{-1} Y - w X^T \Omega^{-1} X \tag{10}$$

$$w^* = (X^T \Omega^{-1} X)^{-1} X^T \Omega^{-1} Y \tag{11}$$

Question 3

Given the following huberized loss function:

$$L_H(w,\delta) = \begin{cases} (y_i - w^T x_i)^2/2 & \text{if } |y_i - w^T x_i| \leq \delta \\ \delta |y_i - w^T x_i| - \delta^2/2 & \text{otherwise} \end{cases} \tag{12}$$

A)

Find the derivative:

$$\frac{\partial}{\partial w} L_H(w,\delta) = \begin{cases} \frac{\partial}{\partial w}\left((y_i - w^T x_i)^2/2\right) & \text{if } |y_i - w^T x_i| \leq \delta \\ \frac{\partial}{\partial w}\left(\delta |y_i - w^T x_i| - \delta^2/2\right) & \text{otherwise} \end{cases} \tag{13}$$

The equation $|y_i - w^T x_i|$ is equivalent to $\sqrt{(y_i - w^T x_i)^2}$ as such the derivative becomes:

$$\frac{\partial}{\partial w} L_H(w,\delta) = \begin{cases} -x_i (y_i - w^T x_i) & \text{if } |y_i - w^T x_i| \leq \delta \\ -x_i \delta \frac{y_i - w^T x_i}{|y_i - w^T x_i|} & \text{otherwise} \end{cases} \tag{14}$$

The expression $\frac{y_i - w^T x_i}{|y_i - w^T x_i|}$ is equivalent to the sign of $y_i - w^T x_i$ as such the derivative can be written as:

$$\frac{\partial}{\partial w} L_H(w,\delta) = \begin{cases} -x_i (y_i - w^T x_i) & \text{if } |y_i - w^T x_i| \leq \delta \\ -x_i \delta sign(y_i - w^T x_i) & \text{otherwise} \end{cases} \tag{15}$$

Where $sign(y_i - w^T x_i)$ is replaced by 1 or -1 depending on whether the result of the equation is positive or negative.

B)

The algorithm was implemented in python source code and is defined as function GradientDescent.

Data: a dataset containing features and targets, learning rate, lambda, delta
Result: a weight vector
initialize weights to 1 or random numbers;
while *max iterations not achieved* **do**
  initialize gradient = 0;
  while *not at the last row of the data array* **do**
    hypothesis = dot product (weights , row of features);
    regularizor = lambda/2 * dot product(transpose of weights, weights);
    if $|hypothesis| \leq delta$ **then**
      loss = hypothesis - row of targets;
      J = dot product (transpose of the row features, loss);
      gradient += J + lambda/2 * dot product(transpose of weights, weights);
    else
      loss = hypothesis - row of targets;
      if *loss is negative* **then**
        gradient += delta * row of features + regularizor;
      else
        gradient -= delta * row of features + regularizor;
      end
    end
    go to the next row;
  end
  if *gradient * learning rate* $\leq$ *desired precision* **then**
    return weights;
  end
  weights = weights + (learning rate) * gradient
end
return weights;

Algorithm 1: Gradient descent using huber loss function

C)

Question 4

$$h_{w1,\dots wk}(x) = \prod_{k=1}^{K} \phi_k(x) \tag{16}$$

Where:

$$\phi_k(x) = e^{w_k^T x} = e^{\sum_{i=1}^{m} w_{ik} x_i} \tag{17}$$

Replacing $\phi_k(x)$

$$h_{w1,\dots wk}(x) = \prod_{k=1}^{K} e^{\sum_{i=1}^{m} w_{ik} x_i} \tag{18}$$

$$\log h_{w1,\dots wk}(x) = \sum_{k=1}^{K} \sum_{i=1}^{m} w_{ik} x_i \tag{19}$$

$$\frac{\partial}{\partial w} \log h_{w1,\dots wk}(x) = \frac{\partial}{\partial w} \left(\sum_{k=1}^{K} \sum_{i=1}^{m} w_{ik} x_i \right) \tag{20}$$

$$\frac{\partial}{\partial w} \log h_{w1,\ldots wk}(x) = \sum_{i=1}^{m} x_i \tag{21}$$