CSCI 567–Homework 4

Chengliang Dong (Meng Liu, Kan Wang, Ye Feng and Peehoo Dewan)Wednesday 19th November, 2014

1 Boosting

1.1 Gradient descent

$$\begin{aligned} g_i &= \frac{\partial L(y_i, \hat{y}_i)}{\partial \hat{y}_i} \\ &= 2(\hat{y}_i - y_i) \end{aligned}$$

1.2 Week learner selection

Define,

$$\gamma^* = min_{\gamma \in R} \sum_{i=1}^{n} (-g_i - \gamma h(x_i))^2$$

Solve for γ^*, we have,

$$\begin{aligned} \frac{\partial \sum_{i=1}^{n} (-g_i - \gamma h(x_i))^2}{\partial \gamma} &= \sum_{i=1}^{n} 2h(x_i)(g_i + \gamma h(x_i)) \\ &= 0 \\ \gamma^* &= \frac{-\sum_{i=1}^{n} h(x_i) g_i}{\sum_{i=1}^{n} h(x_i)^2} \end{aligned}$$

Note that the second derive $\frac{\partial^2 \sum_{i=1}^{n} (-g_i - \gamma h(x_i))^2}{\partial \gamma^2} > 0$, indicating that minimum value of γ is achived. Therefore, γ^* is a function of $h(x_i)$ and can be computed in a closed form in this step. Next solve for h^*, we have,

$$\begin{aligned} \frac{\partial \frac{-\sum_{i=1}^{n} h(x_i) g_i}{\sum_{i=1}^{n} h(x_i)^2}}{\partial h(x_i)} &= \frac{-\sum h'(x_i) g_i}{\sum h(x_i)^2} + \frac{\sum h(x_i) g_i \sum 2h(x_i) h'(x_i)}{(\sum h(x_i)^2)^2} \\ &= 0 \end{aligned} \tag{1}$$

Since there is no γ in equation (1), we have the optimal solution of h^* is indenpendent of γ.

1.3 Step Size Selection

Solve for α, we have,

$$\begin{aligned} \frac{\partial \sum_{i=1}^{n} (\hat{y}_i + \alpha h^*(x_i) - y_i)^2}{\partial \alpha} &= \sum_{i=1}^{n} 2h^*(x_i)(\hat{y}_i + \alpha h^*(x_i) - y_i) \\ &= 0 \\ \alpha^* &= \frac{\sum_{i=1}^{n} h^*(x_i)(y_i - \hat{y}_i)}{\sum_{i=1}^{n} h^*(x_i)^2} \end{aligned}$$

Notice that the second derivative $\frac{\partial^2 \sum_{i=1}^{n}(\hat{y}_i+\alpha h^*(x_i)-y_i)^2}{\partial \alpha^2} > 0$, indicating that minimum value of α is achieved. Therefore, we have the updating function as follows.

$$\hat{y}_i := \hat{y}_i + \frac{\sum_{i=1}^{n} h^*(x_i)(y_i - \hat{y}_i)}{\sum_{i=1}^{n} h^*(x_i)^2} h^*(x_i)$$

Where, $h^*(y_i)$ is the solution from (1).

2 Neural Network

2.1 Logistic Regression

This is very obvious. No matter how many hidden layers you have in a Neural Network with linear activation function, in the last hidden layer, linear combination of nodes in the last hidden layer can also be rewritten as the linear combination of nodes in the layer before it. By recursively plugging in the linear combination from last layer to the next one, finally linear combination of nodes in the last hidden layer can be rewritten as the linear combination of nodes in the input layer. So, after sigmoid transformation, the final output is exactly the same as output of logistic regression.

2.2 Back propogation

First, forward propagate to compute $\hat{y}_j$.

$$\begin{aligned}\hat{y}_j &= \sum_{k=1}^{4} v_{jk} z_k \\ &= \sum_{k=1}^{4} v_{jk} tanh(\sum_{i=3}^{3} w_{ki} x_i)\end{aligned}$$

Next, our goal is to get,

$$\frac{\partial L(y,\hat{y})}{\partial v_{jk}} = \frac{\partial L(y,\hat{y})}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial v_{jk}} \tag{2}$$

$$\frac{\partial L(y,\hat{y})}{\partial w_{ki}} = \frac{\partial L(y,\hat{y})}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial w_{ki}} \tag{3}$$

So we need,

$$\begin{aligned}\frac{\partial L(y,\hat{y})}{\partial \hat{y}_j} &= \frac{\partial \frac{1}{2} \sum_{j=1}(y_j - \hat{y}_j)^2}{\partial \hat{y}_j} \\ &= \hat{y}_j - y_j \\ \frac{\partial \hat{y}_j}{\partial v_{jk}} &= \frac{\partial \sum_{k=1}^{4} v_{jk} tanh(\sum_{i=3}^{3} w_{ki} x_i)}{\partial v_{jk}} \\ &= tanh(\sum_{i=3}^{3} w_{ki} x_i) \\ \frac{\partial \hat{y}_j}{\partial w_{ki}} &= \frac{\partial \sum_{k=1}^{4} v_{jk} tanh(\sum_{i=3}^{3} w_{ki} x_i)}{\partial w_{ki}} \\ &= v_{jk} x_i (1 - tanh^2(\sum_{i=3}^{3} w_{ki} x_i))\end{aligned} \tag{4}$$

Plug (4) to (2) and (3), we get,

$$\begin{aligned}
\frac{\partial L(y,\hat{y})}{\partial v_{jk}} &= \frac{\partial L(y,\hat{y})}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial v_{jk}} \\
&= (\hat{y}_j - y_j)tanh(\sum_{i=3}^{3} w_{ki}x_i) \\
&= (\sum_{k=1}^{4} v_{jk}tanh(\sum_{i=3}^{3} w_{ki}x_i) - y_j)tanh(\sum_{i=3}^{3} w_{ki}x_i) \\
\frac{\partial L(y,\hat{y})}{\partial w_{ki}} &= \frac{\partial L(y,\hat{y})}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial w_{ki}} \\
&= (\sum_{k=1}^{4} v_{jk}tanh(\sum_{i=3}^{3} w_{ki}x_i) - y_j)(v_{jk}x_i(1 - tanh^2(\sum_{i=3}^{3} w_{ki}x_i)))
\end{aligned}$$

Thus, we apply gradient descent and get update function at t+1 the iteration (with step size α) as follows,

$$\begin{aligned}
v_{jk}^{t+1} &= v_{jk}^{t} - \alpha(\sum_{k=1}^{4} v_{jk}^{t}tanh(\sum_{i=3}^{3} w_{ki}^{t}x_i) - y_j)tanh(\sum_{i=3}^{3} w_{ki}^{t}x_i) \\
w_{ki}^{t+1} &= w_{ki}^{t} - \alpha(\sum_{k=1}^{4} v_{jk}^{t}tanh(\sum_{i=3}^{3} w_{ki}^{t}x_i) - y_j)(v_{jk}^{t}x_i(1 - tanh^2(\sum_{i=3}^{3} w_{ki}^{t}x_i)))
\end{aligned}$$

3 Clustering

3.1 Mean

For cluster i, solve for μ_i, we have,

$$\begin{aligned}
\frac{\partial D}{\partial \mu_i} &= \sum_{n=1}^{N} 2r_{ni}(\mu_i - x_n) \\
&= 0 \\
\mu_i &= \frac{\sum_{n=1}^{N} r_{ni}x_n}{\sum_{i=1}^{N} r_{ni}}
\end{aligned}$$

3.2 Median

For cluster i, solve for μ_i, we have,

$$\begin{aligned}
\frac{\partial D}{\partial \mu_i} &= \sum_{n=1}^{N} r_{ni}(I_{\mu_i >= x_n} - I_{\mu_k < x_n}) \\
&= 0 \\
\sum_{n=1}^{N} r_{ni}I_{\mu_i >= x_n} &= \sum_{n=1}^{N} r_{ni}I_{\mu_k < x_n}
\end{aligned} \tag{5}$$

(5) means that the number of points in the ith cluster, which are smaller than μ_i, is the same as the number of points in this cluster, which are larger than μ_i. According to the definition of median, μ_i is the median of ith cluster.

4 Mixture models

4.1 Log likelihood

$$logL = nlog\lambda - \sum_{i=1}^{n} \lambda X_i$$

4.2 E step

At t+1 th iteration, we have,

$$\begin{aligned}
Q(\lambda|\lambda^t) &= E_{X|Y,\lambda^t}[logL] \\
&= E_{X|Y,\lambda^t}[nlog\lambda - \sum_{i=1}^{n} \lambda X_i] \\
&= nlog\lambda - \sum_{i=1}^{n} \lambda E[X_i|Y_i] \\
&= nlog\lambda - \sum_{i=1}^{r} \lambda E[X_i|Y_i = X_i, \lambda^t] - \sum_{i=r+1}^{n} \lambda E[X_i|Y_i = Y_i, \lambda^t] \\
&= nlog\lambda - \sum_{i=1}^{r} \lambda Y_i - \sum_{i=r+1}^{n} \lambda E[X_i|X_i >= Y_i, \lambda^t] \\
&= nlog\lambda - \sum_{i=1}^{r} \lambda Y_i - \sum_{i=r+1}^{n} \lambda \int_{Y_i}^{\infty} x \frac{f(x)}{1 - F(Y_i)} dx \\
&= nlog\lambda - \sum_{i=1}^{r} \lambda Y_i - \sum_{i=r+1}^{n} \lambda (\frac{1}{\lambda^t} + Y_i)
\end{aligned}$$

4.3 M step

At t+1 th iteration, we have,

$$\begin{aligned}
\lambda^{t+1} &= argmax_\lambda Q(\lambda|\lambda^t) \\
\frac{\partial Q(\lambda|\lambda^t)}{\partial \lambda} &= \frac{n}{\lambda} - \sum_{i=1}^{n} Y_i - \frac{n-r}{\lambda^t} \\
&= 0 \\
\lambda^{t+1} &= \frac{n}{\sum_{i=1}^{n} Y_i + \frac{n-r}{\lambda^t}}
\end{aligned}$$