Machine Learning, Spring 2018
Homework 3

Due on 23:59 Apr 17, 2018
Send to $cs282\_01@163.com$
with subject "Chinese name+student number+HW3"

1 Perceptron

(1) *Proof.* $\because \boldsymbol{w}(t+1) \leftarrow \boldsymbol{w}(t) + y(t)\boldsymbol{x}(t)$, we have

$$\begin{aligned} y(t)\boldsymbol{w}^{\boldsymbol{T}}(t+1)\boldsymbol{x}(t) &= y(t)[\boldsymbol{w}^{\boldsymbol{T}}(t) + y(t)\boldsymbol{x}^{\boldsymbol{T}}(t)]\boldsymbol{x}(t) \\ &= y(t)\boldsymbol{w}^{\boldsymbol{T}}(t)\boldsymbol{x}(t) + |y(t)|\,\|\boldsymbol{x}(t)\|_2 \\ &\geq y(t)\boldsymbol{w}^{\boldsymbol{T}}(t)\boldsymbol{x}(t) \end{aligned} \tag{1}$$

Since, $\boldsymbol{x}(t)$ is misclassified by $\boldsymbol{w}(t)$
$\therefore y(t)\boldsymbol{w}^{\boldsymbol{T}}(t)\boldsymbol{x}(t) < 0$
So, we have $\|\boldsymbol{x}(t)\|_2 \neq 0$
Therefore, $y(t)\boldsymbol{w}^{\boldsymbol{T}}(t+1)\boldsymbol{x}(t) > y(t)\boldsymbol{w}^{\boldsymbol{T}}(t)\boldsymbol{x}(t)$. □

(2) Suppose $\boldsymbol{x}(t)$ was misclassified.
$\because \boldsymbol{w}(t+1) \leftarrow \boldsymbol{w}(t) + y(t)\boldsymbol{x}(t)$, we have
$\boldsymbol{w}^{\boldsymbol{T}}(t+1)\boldsymbol{x}(t) = \boldsymbol{w}^{\boldsymbol{T}}(t)\boldsymbol{x}(t) + y(t)(\boldsymbol{x}(t) \cdot \boldsymbol{x}(t))$

If $\boldsymbol{x}(t)$ was incorrectly classified as negative, then $y(t) = +1$. It follows that the new dot product increased by $\boldsymbol{x}(t) \cdot \boldsymbol{x}(t)$ (which is positive). The boundary moved in the right direction as far as $\boldsymbol{x}(t)$ is concerned, therefore. Conversely, if $\boldsymbol{x}(t)$ was incorrectly classified as positive, then $y(t) = -1$. It follows that the new dot product decreased by $\boldsymbol{x}(t) \cdot \boldsymbol{x}(t)$ (which is positive). The boundary moved in the right direction as far as $\boldsymbol{x}(t)$ is concerned, therefore.

2 Understanding logistic regression

Answer:

(1) Suppose $p = \mathbb{P}(\boldsymbol{y} = 1|\boldsymbol{x})$

$$\begin{aligned} \text{logit}(p) &= \log \frac{p}{1-p} = \boldsymbol{w^T x^{(i)}} \\ p &= \frac{1}{1+e^{-\boldsymbol{w^T x^{(i)}}}} \\ &= \sigma(\boldsymbol{w}^T \boldsymbol{x^{(i)}}) \end{aligned} \tag{2}$$

(2) As long as $\boldsymbol{x}$ is not zero, the squared error loss with respect to $\boldsymbol{w}$ will be non-convex. It's hard to optimize. Whereas the log loss is convex.

(3)

$$\mathbb{P}(y^{(i)} = k|\boldsymbol{x} = \boldsymbol{x}^{(i)}) = \sum_{k=1}^{K} \mathbb{1}\{y^{(i)} = k\} \frac{e^{f_{y^{(i)}}}}{\sum_{j=1}^{3} e^{f_j}}$$

in there $f(\boldsymbol{x}^{(i)}, \boldsymbol{W}) = \boldsymbol{W}^T \boldsymbol{x}^{(i)}$,$k = 1, 2, 3$

$$L(y^{(1)}, \ldots, y^{(N)}|\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}; \boldsymbol{W}) = \prod_{i=1}^{N}\sum_{k=1}^{K} \mathbb{1}\{y^{(i)} = k\} \frac{e^{f_{y^{(i)}}}}{\sum_{j=1}^{3} e^{f_j}}$$

The Negative Log-Likelyhood of the N samples as follows:

$$-\ln L = -\sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{1}\{y^{(i)} = k\}(f_{y^{(i)}} - \ln \sum_{j=1}^{3} e^{f_j})$$

3 Regularization

(1) *Proof.* For single sample:

$$p(t|x, \mathbf{w}, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{[t-y(x,\mathbf{w})]^2}{2\sigma^2}}$$

For all samples, likehood is :

$$L = \prod_{i=1}^{N} p(t^{(i)}|x^{(i)}, \boldsymbol{w}, \sigma) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t^{(i)}-y^{(i)})^2}{2\sigma^2}}$$

log-likehood:

$$\ln L = -N \ln \sqrt{2\pi}\sigma - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (t^{(i)} - y^{(i)})^2$$

$\therefore$ Maximizing the log likehood is equal to minimizing the sum-of- squares error function. $\square$

(2) According to Bayes' theorem, for single sample:

$$\begin{aligned} p(t, \boldsymbol{w}|x, \alpha, \sigma) &= p(t|\boldsymbol{w}, x, \alpha, \sigma) \cdot p(\boldsymbol{w}|x, \alpha, \sigma) \\ &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{[t-y(x,\mathbf{w})]^2}{2\sigma^2}} \cdot \frac{1}{\sqrt{(2\pi)^D |\alpha I|}} e^{-\frac{1}{2}\boldsymbol{w}^T(\alpha I)^{-1}\boldsymbol{w}} \\ &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{[t-y(x,\mathbf{w})]^2}{2\sigma^2}} \cdot \frac{1}{\sqrt{(2\pi\alpha)^D}} e^{-\frac{1}{2\alpha}\|\boldsymbol{w}\|_2^2} \end{aligned} \tag{3}$$

For all samples, likehood is :

$$\begin{aligned} L &= \prod_{i=1}^{N} p(t^{(i)}, \boldsymbol{w}|x^{(i)}, \alpha, \sigma) \\ &= (\frac{1}{\sqrt{2\pi}\sigma})^N (\frac{1}{\sqrt{(2\pi\alpha)^D}})^N exp(\sum_{i=1}^{N}(-\frac{1}{2\sigma^2}(t^{(i)} - y^{(i)})^2 - \frac{1}{2\alpha}\|\boldsymbol{w}\|_2^2) \end{aligned} \tag{4}$$

$\frac{1}{N}$log-likehood:

$$\frac{1}{N}\ln L = C - \frac{1}{N}\sum_{i=1}^{N}(\frac{1}{2\sigma^2}(t^{(i)} - y^{(i)})^2 + \frac{1}{2\alpha}\|\boldsymbol{w}\|_2^2), C \text{ is constant}$$

The formulation of the prediction problem is:

$$\text{minimize}\ \frac{1}{N}\sum_{i=1}^{N}(\frac{1}{2\sigma^2}(t^{(i)} - \boldsymbol{w}^T x^{(i)}))^2 + \frac{1}{2\alpha}\|\boldsymbol{w}\|_2^2)$$

Regularization parameter is: $\frac{1}{2\alpha}$

4 Program Logistic regression in matlab

(1) Fig.1. shows the norm of gradient corresponding to three algorithms. All of three algorithms were terminated when the norm of gradient small than 0.00001(In last question, I change it to 0.001 for BFGS and add normalization, so plot maybe different as shown.) and keep in one Matlab function file named *lr_lindq.m*. Each algorithm was called by a extra arguments *type*, 0, 1 and 2 corresponding to Negative gradient, Newton's direction and BFGS, respectively.

(2) Terminate the BFGS when the norm of gradient small than 0.001. Accuracy of prediction in training set is equal to 0.9699.

(3) Consider very small amount of champion data, I amplified some entries in Hessian matrix relate to champion data. Since $H = X^T DX$, I zoomed in the entries in D with 20000x, where is champion data, say $y_i = 1$. Terminate the BFGS when the norm of gradient small than 0.001. Accuracy of prediction in training set is equal to 0.9725. See code in *lr_lindq.m* when *type* = 3.

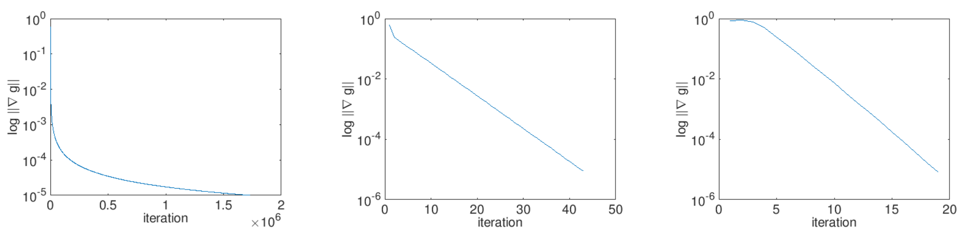

Figure 1: Negative gradient(left), Newton's direction(middle) and BFGS(right)