Machine Learning, Spring 2018
Homework 3
Xiaohe He(486553895)

Due on 23:59 Apr 19, 2018

1 Perceptron

The weight update rule in Pereptron Learning Algorithm (PLA) has the nice interpretation that it moves in the direction of classifying $x(t)$ correctly at iteration t.

(1) Show that $y(t)w^T(t+1)x(t) > y(t)w^T(t)x(t)$. (5 points)

(2) As far as classifying $x(t)$ is concerned, argue that the move from $w(t)$ to $w(t+1)$ is a move "in the right direction". (10 points)

Solution:

(1)

$$\begin{aligned} y(t)w^T(t+1)x(t) &= y(t)[w^T(t) + x(t)y(t)]^T x(t) \\ &= y(t)w^T(t)x(t) + y(t)^2 x^T(t)x(t) \\ &> y(t)w^T(t)x(t) \end{aligned}$$

(2) If the data set is linear separable, there must exits a perfect w_f , for all $(x_i, y_i), y_i = sign(w_f^T x_i)$. Set initial $w(0) = 0$ After T times update, we have:

$$\begin{aligned} w_f^T w(T) &= w_f^T (w(T-1) + y_i x) \\ &\geq w_f^T w(T-1) + min_n(y_n w_f^T x_n) \\ &\geq \cdots \geq w_f^T w(0) + T * min_n(y_n w_f^T x_n) \\ &= T * min_n(y_n w_f^T x_n) \end{aligned}$$

$$\begin{aligned}||w(T+1)||^2 &= ||w(T) + y_{n(T)}x_{n(T)}||^2 \\ &= ||w(T)||^2 + 2y_{n(T)}w_t^T x_{n(T)} + ||y_{n(T)}x_{n(T)}||^2 \\ &\leq ||w(T)||^2 + 0 + ||y_{n(T)}x_{n(T)}||^2 \\ &\leq ||w(T)||^2 + max_n||y_{n(T)}x_{n(T)}||^2\end{aligned}$$

$$\begin{aligned}||w(T)||^2 &\leq ||w(T-1)||^2 + max_n||x_n||^2 \\ &\leq ||w_0||^2 + T * max_n||x_n||^2 \\ &\leq T * max_n||x_n||^2\end{aligned}$$

Then we have:

$$\begin{aligned}\cos(\theta) &= \frac{w_f^T w(T)}{||w_f||||w(T)||} \\ &\geq \frac{T * min_n(y_n w_f^T x_n)}{||w_f||||w(T)||} \\ &\geq \frac{T * min_n(y_n w_f^T x_n)}{\sqrt{T} * max_n||x_n||} \\ &= \sqrt{T} * \frac{min_n(y_n w_f^T x_n)}{||w_f|| * max_n||x_n||}\end{aligned}$$

Where $\frac{min_n(y_n w_f^T x_n)}{||w_f||*max_n||x_n||}$ is a constant, hence the angle between vector w_f and $w(t)$ is decreasing with increasing T. That is, $w(T)$ is approaching the perfect w_f , we are on the right direction.

2 Understanding logistic regression

Given training data set $\{x(i),\ y(i)\}_{i=1}^m$, $y(i) \in \{0, 1\}$. Logistic regression can be seen as a special case of generalized linear model and thus analogous to linear regression. The model of logistic regression, however, is based on quite different assumptions (about the relationship between dependent and independent variables) from those of linear regression.

Question:

(1) Try to explain why we can set $\mathbb{P}(\boldsymbol{y} = 1|\boldsymbol{x}) = \sigma(\boldsymbol{w}^T\boldsymbol{x})$. (Hint: The conception *odd ratio* may be helpful.) (5 points)

(2) The MSE for logistic regression, i.e.,

$$J(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m}(\sigma(\boldsymbol{w}^T\boldsymbol{x}^{(i)}) - y^{(i)})^2.$$

is not a good loss function. Why? (5 points)

(3) Now suppose we have a 3-class task, i.e., $y(i) \in \{1, 2, 3\}$, find the Negative Log-Likelyhood of the given data (the objective of the softmax regression problem). (10 points)

Solution:

(1) By setting $\mathbb{P}(\boldsymbol{y} = 1|\boldsymbol{x})$ we have

$$logit(p) = log\frac{p}{1-p} = \boldsymbol{w}^T\boldsymbol{x}$$

$\Rightarrow$

$$\mathbb{P}(\boldsymbol{y} = 1|\boldsymbol{x}) = p = \frac{1}{1+e^{-\boldsymbol{w}^T\boldsymbol{x}}} = \sigma(\boldsymbol{w}^T\boldsymbol{x})$$

(2) If $\boldsymbol{x}$ is not 0, the squared error loss is not convex, which is not easy to optimize.

(3) Since $ln\frac{P(y^{(i)}=1|x^{(i)})}{P(y^{(i)}=3|x^{(i)})} = \boldsymbol{w_1^T x^{(i)}}$, $ln\frac{P(y^{(i)}=2|x^{(i)})}{P(y^{(i)}=3|x^{(i)})} = \boldsymbol{w_2^T x^{(i)}}$, we have

$$P(y^{(i)} = 1|x^{(i)}) = e^{\boldsymbol{w_1^T x^{(i)}}} P(y^{(i)} = 3|x^{(i)})$$

$$P(y^{(i)} = 2|x^{(i)}) = e^{\boldsymbol{w_2^T x^{(i)}}} P(y^{(i)} = 3|x^{(i)})$$

For $P(y^{(i)} = 1|x^{(i)}) + P(y^{(i)} = 2|x^{(i)}) + P(y^{(i)} = 3|x^{(i)}) = 1$, we get that

$$p_1 = P(y^{(i)} = 1|x^{(i)}) = \frac{e^{\boldsymbol{w_1^T x^{(i)}}}}{1+\sum_{k=1}^{2} e^{\boldsymbol{w_k^T x^{(i)}}}}$$

$$p_2 = P(y^{(i)} = 2|x^{(i)}) = \frac{e^{\boldsymbol{w_2^T x^{(i)}}}}{1+\sum_{k=1}^{2} e^{\boldsymbol{w_k^T x^{(i)}}}}$$

$$p_3 = P(y^{(i)} = 3|x^{(i)}) = \frac{1}{1+\sum_{k=1}^{2} e^{\boldsymbol{w_k^T x^{(i)}}}}$$

$\Rightarrow$

$$NLL = -\sum_{i=1}^{m}\left[T_{i,3}\cdot ln\frac{1}{1+\sum_{k=1}^{2} e^{\boldsymbol{w_k^T x^{(i)}}}} + \sum_{j=1}^{2} T_{i,j}\cdot ln\frac{e^{\boldsymbol{w_j^T x^{(i)}}}}{1+\sum_{k=1}^{2} e^{\boldsymbol{w_k^T x^{(i)}}}}\right]$$

Where

$$T_{i,j} = \begin{cases} 1 & \text{if } y^{(i)} = j \\ 0 & \text{otherwise} \end{cases}$$

3 Regularization

The goal in the prediction problem is to be able to make prediction for the target variable t given some new value of the input variable x on the basis of a set of training data comprising N input values $\mathbf{x} = (x_1, \ldots, x_N)^T$ and their corresponding target variable $\mathbf{t} = (t_1, \ldots, t_N)^T$. we could assume that, given the value of x , the corresponding value of t has a Guassian distribution with a mean equal to the value $y(x, w)$ and the variance σ, where $y(x, w)$ is the prediction function. For example, for the linear regression, the $y(x, \mathbf{w}) = w_0 + w_1 x$.

Thus, we have

$$p(t|x, \mathbf{w}, \sigma) = \mathcal{N}(t|y(x, \mathbf{w}), \sigma)$$

Here we only consider the case of a single real-valued variable x. Now you need to use the training data $\{\mathbf{x}, \mathbf{t}\}$ to determine the parameter $\mathbf{w}$ and σ by maximum likelihood.

(1) Show that maximizing the log likehood is equal to minimizing the sum-of-squares error function. (10 points)

(2) More, if we assume that the ploynomial codfficients $\mathbf{w}$ is distributed as the Guassian distribution of the form

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha\mathbf{I})$$

where α is the paramater of the distribution. Then what is the formulation of the prediction problem? And give us the regularization parameter. Please show us the induction of the procedure.
(Hint. Using Bayes' theorem) (15 points)

Solution:

(1)

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

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

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

Hence maximizing the log likehood is equal to minimizing the sum-of-squares error function.

(2) By Bayes' theorem,

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

$$
\begin{aligned}
L &= \prod_{i=1}^{N} p(t^{(i)}, \boldsymbol{w} | x^{(i)}, \alpha, \sigma) \\
&= (\frac{1}{\sqrt{2\pi}\sigma})^N (\frac{1}{\sqrt{(2\pi\alpha)^D}})^N exp(\sum_{i=1}^{N}(-\frac{1}{2\sigma^2}(t^{(i)} - y^{(i)})^2 - \frac{1}{2\alpha}\left\|\boldsymbol{w}\right\|_2^2) \\
\frac{1}{N}\ln L &= C - \frac{1}{N}\sum_{i=1}^{N}(\frac{1}{2\sigma^2}(t^{(i)} - y^{(i)})^2 + \frac{1}{2\alpha}\left\|\boldsymbol{w}\right\|_2^2), C \text{ is constant}
\end{aligned}
$$

The formulation of the prediction problem is:

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

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

4 Program Logistic regression in matlab

Program a matlab function based on the algorithms (Negative gradient, Newton's direction, and BFGS) you have learned

`[weight, gradnormList] = logisticRegression(X, y),`

where $\boldsymbol{X}$ is the data matrix and $\boldsymbol{y}$ is the label. You may like to read the matlab script `hw3_demo.m` first and following the description in it.

(1) The first case is a very simple dataset given as follows

$$
\begin{bmatrix}
0 & 0 & 0 \\
2 & 2 & 0 \\
2 & 0 & +1 \\
3 & 0 & +1
\end{bmatrix}
$$

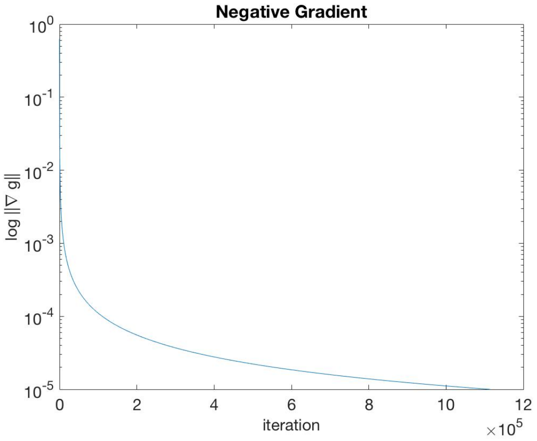
where the first two columns are the attributes (data matrix $\boldsymbol{X}$) and the last column is the label $\boldsymbol{y}$. Plot the norm of gradient corresponding to three algorithms to illustrate the convergence. (10 points)

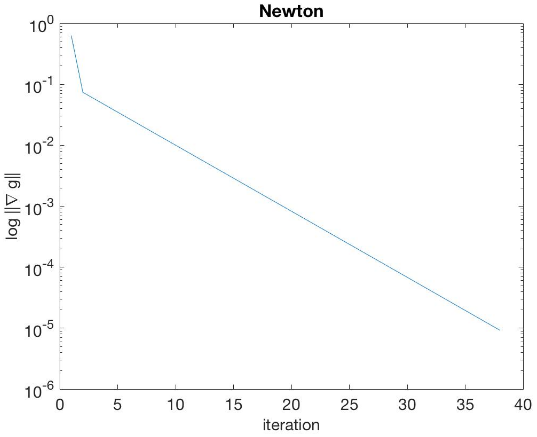
(2) Run your function on `nbadata.mat` dataset based on BFGS and compute the accuracy of prediction in training set. (15 points)

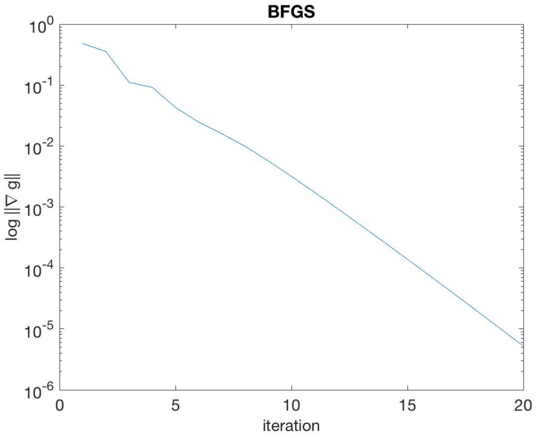
(3) The dataset `nbadata.mat` is class-imbalance. Modifying your algorithm (again, based on BFGS) that enable it to be applied to the situation of class-imbalance and compute the accuracy of prediction in training set. (Please give a detailed description of your methods and explain why in report.) (15 points)

Solution:

(1) Plots of the norm of gradient are as follows:







(2) Set epsilon as 1e-3, we could get the accuracy of prediction in nbadata is 0.9699. Codes $BFGSOnNBA$ shows it.

(3) Set epsilon as 1e-3, and muiltiplay the elements in D 20000. Then we could get the modified prediction accuracy is 0.9725. Codes $ModifiedBFGSOnNBA$ shows it.