Solving Logistic Regression with Newton's Method

Fumin
fumin@shanghaitech.edu.cn

ShanghaiTech University

2018 年 12 月 5 日

Overview

Introduction

Classify if a tumour is malignant or benign

- Mean radius, mean texture, ... input $x \in \mathbb{R}^{30}$
- A malignant tumour or not output $y \in \{1, 0\}$
- True relationship between x and y target function f
- Data on patients data set $\mathcal{D} = \{(x_1, y_1), ..., (x_N, y_N)\}$

Data is from the Kaggle Breast Cancer Wisconsin Data Set which has thirty features. And Using 10%of dataset for validation,10%of dataset for testing.

Problem Formulation

Logistical regression

- Hypothesis Function is:

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma(\boldsymbol{\theta}^T \mathbf{x})$$

where $\sigma(t) = \frac{1}{1+e^{-t}}$, called sigmoid function. And if $h_{\boldsymbol{\theta}}(\mathbf{x})$ were greater than 0.5, we shoud diagnose the tumor as malignant.

- According to maximizing the likelihood funxtion, Cost Function is formed as:

$$\begin{aligned} Cost(h_{\boldsymbol{\theta}}(\mathbf{x}), y) &= \begin{cases} -\ln(h_{\boldsymbol{\theta}}(\mathbf{x})) & \text{y=1} \\ -\ln(1 - h_{\boldsymbol{\theta}}(\mathbf{x})) & \text{y=0} \end{cases} \\ &= -y\ln(h_{\boldsymbol{\theta}}(\mathbf{x})) - (1-y)(\ln(1 - h_{\boldsymbol{\theta}}(\mathbf{x}))) \end{aligned}$$

- we have m data points and add regularizer. Then the optimization problem become:

$$\min \ f(\boldsymbol{\theta}) = \sum_{i=1}^{m} Cost(h_{\boldsymbol{\theta}}(x_i), y_i) + \lambda \boldsymbol{\theta}^T \boldsymbol{\theta}$$

Solution method

Newton's method in python

Algorithm 1 Newton's Algorithm

Input: $\{\mathbf{x}_i \in \mathbb{R}^{1\times 31}\}_{i=1}^{m}$, $\{\mathbf{y}_i \in \mathbb{R}\}_{i=1}^{m}$,a small value η

Output: $\boldsymbol{\theta}$

Initialisation : $\boldsymbol{\theta}^0 = \mathbf{0} \in \mathbb{R}^{1\times 31}$

LOOP Process

for $t = 1, 2, \ldots$ **do**

$\nabla f(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{t-1}} = \mathbf{X}^T(\sigma - \mathbf{y}) + \lambda\boldsymbol{\theta}_{t-1}$

$\nabla^2 f(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{t-1}} = \mathbf{X}^T\mathbf{D}\mathbf{X} + \lambda\mathbf{I}$

$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - [\nabla^2 f(\boldsymbol{\theta}_{t-1})]^{-1}\nabla f(\boldsymbol{\theta}_{t-1})$

if $||\nabla f(\boldsymbol{\theta}_{t-1}|| < \eta$ **then**

break

end if

end for

where $\mathbf{D} = \mathrm{diag}(\sigma^i(1-\sigma^i))$, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]^T$, $\mathbf{y} = [\mathbf{y}_1, \ldots, \mathbf{y}_m]^T$

Numerical Results

Choose the λ

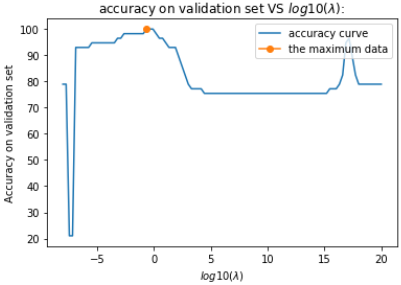
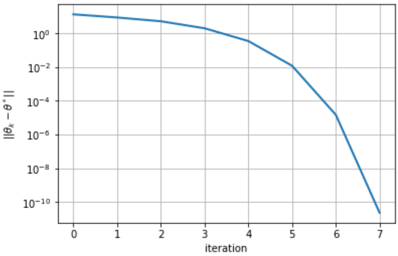


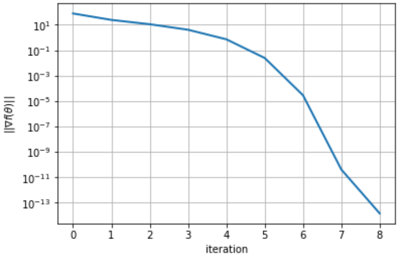
图: find the optimal λ

In the end, using the optimal $\lambda = 0.225$ trains the model.

Local Coveragence



(a) iterations VS $||\theta_k - \theta^*||$



(b) iterations VS $||\nabla f(\theta)||$

图: Local Covergence

Conclusion

Conclusion

- The accuracy on test data is 96.5 %.
- In generally, the trainning data contains the noise and outliers. I added a penalized term to avoid overfitting. By changing λ, I attained the right λ which maxmizes the accuracy on validation data.
- If the Newton's method didn't get a good initial point, the algorithm might faill to the local convergence. Therefore, to make the method works, I choosed the initial point which is close to optimal point.