Homework 5

November 12, 2016

* For each question, give your answer. Justify your answer using a detailed explanation.

1 Understanding logistic regression

Given training data set $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{m}$, $y^{(i)} \in \{1, 0\}$, 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.

(i) The predicted values are probabilities and are therefore restricted to (0,1) through the logistic distribution function because logistic regression predicts the probability of particular outcomes. Suppose $\mathbb{P}(y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x})$, where $\sigma(\cdot)$ is the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$.The odd ratio of the Bernoulli distribution can be defined as

$$\frac{\mathbb{P}(y = 1 \mid \mathbf{x})}{\mathbb{P}(y = 0 \mid \mathbf{x})} = \frac{\sigma(\mathbf{w}^T\mathbf{x})}{1 - \sigma(\mathbf{w}^T\mathbf{x})},$$

and the *logit* is defined as the log of the odd ratio, i.e., $\mathit{logit}(\sigma) = \log \frac{\sigma}{1-\sigma}$. Show that the linear relationship

$$\mathbf{w}^T\mathbf{x} = \mathit{logit}(\sigma)$$

holds true.

(ii) The conditional distribution $(y \mid x)$ is a Bernoulli distribution rather than a Gaussian distribution, because the dependent variable is binary. Find the *negative log likelihood* (NLL) of the given data set.

(iii) Now suppose we have a 3-class task, i.e., $y^{(i)} \in \{1, 2, 3\}$, find the NLL of the given data (the objective of the *softmax regression* problem).

2 Solving a logistic regression

The 2-class classification training data is given below

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

where the first two columns are the attributes (the $\mathbf{X}$ matrix) and the last column is the KPI (key performance indicator, the $\mathbf{y}$ vector).

Using the results you found in the last part, write down the explicit form of the NLL in this case. Write three small pieces of code to implement the iterative method

$$\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + \mathbf{p}_k$$

where $\mathbf{p}_k$ takes three cases:

- Negative gradient ($\mathbf{p}_k = -\mathbf{g}_k$)
- Newton's direction ($\mathbf{p}_k = -\mathbf{H}_k^{-1}\mathbf{g}_k$)
- BFGS direction[1] ($\mathbf{p}_k = -\mathbf{B}_k\mathbf{g}_k$)

Here $\mathbf{g}$ is the gradient of your NLL, $\mathbf{H}$ is the Hessian, and $\mathbf{B}$ is the BFGS approximation of the inverse of the Hessian. Initial point is chosen as $\mathbf{w}_0 = [0, 0, 0]^T$. Terminate each of your algorithms when $\|\mathbf{g}\| \leq 10^{-5}$.

(i) Write down the final solution $\mathbf{w}$ you find for each algorithm.

(ii) Use your solution to predict on the training data (so you get $\bar{y}^{(i)}$). How many data points are wrongly predicted for each algorithm (count the number of data points where $\bar{y}^{(i)} \neq y^{(i)}$)?

(iii) For each algorithm, plot $\log \|\mathbf{g}\|$. Which one is the fastest? (if your algorithm need more than 100 iterations, just plot $\log \|\mathbf{g}\|$ over the first 100 iterations)

[1] Note that the first iteration of a Quasi-Newton is generally a Gradient Descent step