Machine Learning, Spring 2019
Homework 4

Due on 23:59 Apr 26, 2019

---

1. Submit your solutions to Gradescope (www.gradescope.com). Homework of this week contains two part, **theoretical part** and **programming part**. The results of programming part should be put in theoretical part.
2. **Make sure that you have selected the correct pages for the problem in the outline.**
3. Any programming language is allowed for your code, but make sure it is clear and readable with necessary comments.

---

1 The relationship between the maximum likelihood and distance metric

Suppose $y = f(\theta; \mathbf{x})$ is our model (such as classifier or regression model), $\mathbf{x} \in \Re^n$ is feature, $y \in \Re$ is response, and $\theta \in \Re^K$ is the parameters of model f. As you know, the standard processing in machine learning is: We first collocate train samples $\{(\mathbf{x}_i, y_i)\}, i = 1, 2, \cdots, n$, and some loss function will be defined, then we can find the optimal (or suboptimal) θ^* by minimizing the loss function. Mean square error (MSE) is a common loss function.

$$E = \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2.$$

Suppose $y = f(\theta; \mathbf{x}) + \varepsilon$, and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

1. Write down the log likelihood function of θ;(5 points)
2. Use the maximum likelihood principle to explain why MSE is a good loss function.(5 points)

2 Understanding VC dimension

Generally, VC dimension is not related to the number of parameters used by a hypothesis, though we have seen that in linear and polynomial regression, more parameters imply higher VC dimension. Now, consider the hypothesis set

$$\mathcal{H} = \{f(x;\alpha) = sign(\sin(\alpha x))|\alpha \in \mathbb{R}\}$$

for one dimensional classification.

1. Show that $\mathcal{H}$ cannot shatter $m = 4$ points $x^{(1)} = 1, x^{(2)} = 2, x^{(3)} = 3, x^{(4)} = 4$.(5 points)
2. Prove the VC dimension of $\mathcal{H}$ is ∞.(5 points)

(Hints: for part (i), you need to find a set of labels $y^{(1)}, y^{(2)}, y^{(3)}, y^{(4)}$ such that for any α, $f(x;\alpha)$ cannot generate this set of labels, for example, you may want to use $+1, +1, -1, +1$; for part (ii), we have already shown it in class, you may want to use points $x^{(i)} = 10^{-i}, i = 1, \ldots, m$.)

3 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.

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)

4 Solving a logistic regression

The 2-class classification training data is given below

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

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

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

- Negative gradient ($\boldsymbol{p}_k = -\boldsymbol{g}_k$)
- Newtons direction ($\boldsymbol{p}_k = -\boldsymbol{H}_k^{-1}\boldsymbol{g}_k$)
- BFGS direction[1] ($\boldsymbol{p}_k = -\boldsymbol{B}_k\boldsymbol{g}_k$)

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

1. Write down the final solution $\boldsymbol{w}$ you find for each algorithm.(10 points)
2. 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 $\bar{y}^{(i)} \neq y^{(i)}$)?(10 points)
3. For each algorithm, plot log $\|\boldsymbol{g}\|$. Which one is the fastest? (if your algorithm need more than 100 iterations, just plot log $\|\boldsymbol{g}\|$ over the first 100 iterations)(10 points)

5 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 X is the data matrix and y is the label. You may like to read the matlab script **hw4_demo.m** first and following the description in it.

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 X) and the last column is the label y.

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

(2). The dataset **nbadata.mat** is class-imbalance. Modifying your algorithm (again, based on BFGS) that enable it to be applied to the situation of

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

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)