Sen Yang
Zhuo Zhao

MF850

Problem Set 4

Due date: See Blackboard.

Instructions: You submit on Blackboard. You may solve this assigment in groups of two. A submission is constituted by answers to the problems along with the code used. A file called `hw4.py` should contain your code, or your entry point if you separate your code into multiple files. This file should run without errors from a fresh instance/REPL. In other words, submissions in notebook format are not accepted (but you may of course develop in them before creating the submission).

Please contact the instructor or a TA if you have questions regarding these instructions or if you find the problem formulation unclear.

Problem 4.1 Consider the problem of finding the maximum (log) likelihood of a multinomial distribution of N categories of sample count y_i and $K = \sum_{i=1}^N y_i$ observations, i.e., to optimize

逐步分配给每个类别一个probability，
而不是一次性分配完所有的

类别i的样本数

$$\max_{q_i \text{ s.t. } \sum_{i=1}^N q_i = 1} \log(K!) - \sum_{i=1}^N \log(y_i!) + \sum_{i=1}^N y_i \log(q_i).$$

从sj=1开始，此时有全部的概率预算；
每一步分配qj后sj不断减少

类别i的概率
sum qi = 1

This problem can be formulated as a control problem:

在第j步，剩余概率预算为sj时能得到的maximum likelihood

$$V_j(s_j) = \max_{q_j + \cdots + q_N = s_j} \sum_{i=j}^N y_i \log(q_i),$$ 因为只有qi是optimizer所以只看最后一项

with dynamics $s_{j+1} = s_j - q_j$ and where we think if j as the time point. Thus, it must hold that $q_j \in [0, s_j]$. The problem can be solved using the following dynamic programming formula:

$$V_j(s_j) = \max_{q_j \in [0,s]} (y_j \log(q_j) + V_{j+1}(s_j - q_j)).$$

在第j步找到能优化log likelihood，以及优化后续步骤的log likelihood的qN,

Remark: This is a sequential allocation problem. Here we think of the state s as a 'probability budget' and we take from this budget, in the form of q_j to allocate probability to outcome j. This reduces the available probability for the remaining outcomes $j+1, \dots, N$ to $s - q_j$, which leads to the probability budget interpretation.

(a) Let $j = N$. What q_N optimizes V_N and what is V_N?

(b) Use dynamic programming to find V_{N-1}.

(c) Use dynamic programming to find the parameters q_i.

If you cannot solve the generalcase, you may use $N = 4$ for partial points.

Hint: The structure of the value function is similar in all time steps. Use the structure of V_{N-1} as an ansatz.

Hint: With starting state $s = 1$, $q_i = y_i/K = y_i/\sum y_i$.

Problem 4.2 Consider logistic regression: We have data (X, y) where y is a vector with values in $\{0, 1\}$. If $p(x)$ is the probability that $y = 1$ for some data x, the goal of logistic regression is to estimate p by fitting the log-odds

$$\log_b \frac{p}{1-p}$$

with a linear model $l(x;\theta) = x \cdot \theta$. Let q be an estimate of p. Then, solving for q,

$$q(x) = \frac{1}{1 + e^{-l(x;\theta)}} = \sigma(l(x;\theta)),$$

where

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

is the so-called logistic function. The parameters θ are found by maximizing the log-likelihood of q.

Remark: For this problem, you are encouraged to experiement with other datasets of your choice.

(a) We have learned that the MLE for logistic regression is equivalent to minimizing the cross-entropy. In light of this, solve the logistic regression problem by minimizing the cross entropy using stochastic gradient descent in Pytorch. Use a 2-parameter neural network (similar to the linear regression example in class) and train on the dataset given by the data generating function `hw.Make_classification(n_samples).data_split()`.

Hint: Pytorch includes the function `BCELoss` that is useful for computing the cross entropy with only two outcomes (here 0 and 1): the *binary* cross entropy.

Test your solution on an independent dataset of the same distribution. Compare your results to the logistic regression solver from Scikit-learn.

(b) How does the performance change on the data sets given by `hw.Make_moons(n_samples).data_split()` and `hw.Make_circles(n_samples).data_split()`? Visualize what happens by plotting.

(c) Repeat the above but replace l with a deeper and wider (nonlinear) neural network. Visualize the differences by plotting.