STAT 535 Homework 3

Out October 29, 2020

Due November 5, 2020

©Marina Meilă

mmp@stat.washington.edu

Problem 1 – Logit loss Hessian

Assume that you have a data set $\mathcal{D} = \{(x^i, y^i), i = 1 : N\}, x^i, w \in \mathbb{R}^n$.

a. Calculate the expression of $\nabla^2 L_{logit}$ for a single data point x. Simplify your result using $\phi(yw^T x)$ conveniently.

[**b.– Not graded.** Show that the gradient of $L_{logit}(w; \mathcal{D})$ is a linear combination of the x^i vectors. *This was shown in the lecture.*

c. Show that if $N < n$ the Hessian of $L_{logit}(w; \mathcal{D})$ has at least one 0 eigenvalue, or equivalently that it is not full rank, and conclude that $L_{logit}(w; \mathcal{D})$ is not strongly convex in this case.

d. – Optional, extra credit If $||x^i|| \leq R$, find a constant M sufficiently large so that $\nabla^2 L_{logit}(w; \mathcal{D}) \prec MI_n$.

A reminder that you are allowed and even encouraged to use results from previous homeworks, course notes, lectures withouth proof.

Problem 2 – Regularization is monotonic w.r.t. λ

Let $J_\lambda(w) = \hat{L}_h(w) + \frac{\lambda}{2}||w||^2$ be a regularized objective functions, where w are the parameters. For example, the linear support vector formulation from Lecture IV, $f(x) = w^T x$. Let $\lambda_1 > \lambda_2 > 0$ and denote $w_{1,2} = \text{argmin}_w J_{\lambda_{1,2}}$ the optimal solutions for λ_1, respectively λ_2, and assume further that $J_{\lambda_{1,2}}$ have *unique global minima.*

a. Prove that $||w_1|| < ||w_2||$ whenever $w_{1,2} \neq 0$.

b. Prove also that $\hat{L}_h(w_1) > \hat{L}_h(w_2)$.

In other words, imposing more regularization reduces the regularized quantity $||w||$, and increases the un-regularized one (i.e., the loss).

Problem 3 – Descent algorithms for training a neural network

This problem asks you to train a neural network to classify the data sets given on the Assignments web page. The inputs are 2-dimensional, outputs are ±1, one data point/line. *Submit the code for this problem.*

Objective to minimize is $\hat{L_{logit}}(\beta, W) = -\frac{1}{N}\text{log-likelihood}(\mathcal{D}|\beta, W)$ and $\beta \in \mathbb{R}^{m+1}, W \in \mathbb{R}^{n\times m}$ are the neural net parameters.

Choose a number $m = 3$ to 5 hidden units (suggested) or go as high as you want (recommended to try both).

Algorithms: steepest descent with fixed step size. You need to implement the algorithm yourself.

[Optional, for extra credit: implement Newton, or run Newton, LBFGS quasi Newton from library code.]

Dataset $\mathcal{D}$ given `hw4-nn-train-100.dat`

a. Explain how you chose the initial points. It's ok to plot the data and look at it or even to make a sketch of the solution you want to find. (If you implement more than one algorithm, start them all from the same initial point.)

The training algorithm will converge to a local optimum. It's OK to look at this local optimum and try other initial points if the found optimum is bad. (Don't forget to use the same initial point for all algos in the results you present in the homework.) It's also recommended to challenge the algorithm by giving it random/uninformative initial points. *Do not start all the parameters at 0 [Why?].*

Chose the stopping criterion $1 - \frac{\hat{l}^{k+1}}{\hat{l}^{k}} \leq tol$ with $tol = 10^{-4}$. If this tolerance cannot be reached in a reasonable number of steps, set a higher tol and report that value.

b. The choices above should be kept the same for all estimation algorithms (except maybe SG). Describe briefly the implementation details of your algorithms. Size of the fixed step, if you bracketed the min or not in line search, what line search method you used (*you can use code from other sources to bracket the mininmum, and you can implement another line search method than Armijo.)*, how you chose C in the stochastic gradient algorithm (trial and error OK) and what value you used, etc. For each algorithm, give the number of iterations (and if it converged or not) and final value of loss function. Record also the time each algorithm takes and report it.

Estimate the value of L_{logit}, L_{01} by averaging them on the test set `hw4-nn-test.dat` for the final classifier obtained. Optionally, compute these values at each iteration and plot them in the graphs for **c**.

c. Plot the values of $\hat{L_{logit}}$, $\hat{L}_{01}$ and the respective costs on the test set vs the iteration number k. Make two separate plots for the two costs. If you have computed the test set costs at each iteration, plot these too on the respective graphs.

d. Plot the final decision region superimposed on the data.

[e. Optional but encouraged] Plot (some of) the β parameters vs k; on a separate plot, show trajectories of β parameters coming from different initializations.

Please make clear, well-scaled, well labeled graphs.

f. Repeat steps **b,c,d,[e**] for the larger data set `hw4-nn-train-10000.dat` on the same models as before. Use the same parameter initialization as in the previous case to get meaningful comparisons.

Do not plot the data set for this part of the problem.

[Optional, extra credit: repeat the training initializing from the *final values* obtained in the small sample run. Plot what you think is meaningful to compare performances.]

g. Discuss the differences that you observe between the algorithms' behavior on the large and small samples.

Problem 4 – Ridge regression

In this problem you will perform ridge regression on the function $f^*(x) = x^2 + 1$ on $[0, 1]$. In the file `hw3_rr.dat` you will find a set of N (x^i, y^i) values with $y^i = f^*(x^i)$.

a. Let $f(x) = \beta_0 + \beta_1 x$ be the predictor of y; β_0, β_1 will be estimated by Ridge Regression with regularization parameter λ. Denote $\beta_{0,1}(\lambda)$ the result of this estimation. Let the data matrix be the row vector $X = [x^1 \ \ldots x^N]$, and define the column vector $y = [y^1 \ \ldots \ y^N]^T$

Write the expressions of $\beta_0(\lambda)$, $\beta_1(\lambda)$ as functions of X, y, λ.

b. Now choose a set of λ values including 0 and N. Calculate $\beta_{0,1}(\lambda)$, $\hat{l}_{LS}(\lambda)$ and $J(\lambda)$. Plot on the same graph $\beta_{0,1}(\lambda)$ vs λ.

c. Plot on the same graph $\hat{l}_{LS}(\lambda)$ and $J(\lambda)$ vs λ. Comment on what you observe in the graphs of b, c.