Homework 3

36-462/662, Spring 2022

Due at 6 pm on Thursday, 10 February 2022

Agenda: More practice with decision theory; relating the log loss to maximum likelihood; practice reasoning about optimization problems and optimization algorithms.

1. **Online questions** (10) are on Canvas and due at 6 pm on Monday.
2. **Log loss and maximum likelihood** In this problem, p is the true, but unknown, probability density function (pdf) of a random variable X, and q is a pdf we're considering as a possible probabilistic prediction about X. Suppose that if we guess the pdf is q and the actual value we see is x, we suffer the loss $-\log q(x)$. This is (as we've seen before) the "negative log probability loss function", or just the "log loss".
   a. (5) What sorts of predictions will get large losses and which ones will get small losses? Does this seem reasonable?
   b. (4) Explain why the risk of q, under the log loss, is $-\int p(x) \log q(x) dx$.
   c. (4) It can be shown that $\int w(t) \log t dt \leq \log\left(\int t w(t) dt\right)$ for *any* probability distribution w. (This is a special case of a theorem about convex functions called "Jensen's inequality"; you don't have to show it.) In fact, $\int w(t) \log t dt = \log \int t w(t) dt$ if, but only if, w puts probability 1 on one particular value of t, and probability 0 on all others. Use this to show that $\int p(x) \log \frac{q(x)}{p(x)} dx \leq 0$. Explain how $q(x)$ must relate to $p(x)$ for the integral to be 0. *Hint*: Probability densities integrate to 1.
   d. (5) Using Q2c, show that $-\int p(x) \log p(x) dx < -\int p(x) \log q(x) dx$ unless $p(x) = q(x)$ everywhere.
   e. (5) Suppose we see independent data points $x_1, \ldots x_n$, all drawn from the same distribution. Explain why the likelihood of the pdf q is $\prod_{i=1}^{n} q(x_i)$. How is this related to the log loss of q? How is minimizing the log loss related to maximizing the likelihood? What doesQ2d tell you about maximum likelihood estimation?
3. **Fitting logistic regression, part 1** In this problem, we'll work through some of the math for fitting a logistic regression. For each data point, we observe a p-dimensional vector of predictors, which are real numbers, and a binary response, which is either 0 or 1, so our data consists of $(x_{11}, x_{12}, \ldots x_{1p}, y_1), (x_{21}, x_{22}, \ldots x_{2p}, y_2), \ldots (x_{n1}, x_{n2}, \ldots x_{np}, y_n)$. Abbreviate $(x_{i1}, \ldots x_{ip})$ as $\vec{x}$ (in LaTeX: `vec{x}`). Our logistic regression model, with intercept b_0 and coefficients $\vec{b}$, is that

$$p(\vec{x}; b_0, \vec{b}) \equiv \mathbb{P}\left(Y = 1 | \vec{X} = \vec{x}\right) = \frac{e^{b_0 + \vec{b} \cdot \vec{x}}}{1 + e^{b_0 + \vec{b} \cdot \vec{x}}}$$

   where (as usual) $\vec{b} \cdot \vec{x} = \sum_{j=1}^{p} b_j x_j$.
   a. (5) Write out an expression for the negative log likelihood per observation, $L(b_0, \vec{b})$. Show that this is linear in the y_i. (*Hint*: Lecture 3.) This is an empirical risk under some loss function; what's the loss function?
   b. (5) Show that

$$\frac{\partial L}{\partial b_j} = -\frac{1}{n} \sum_{i=1}^{n} (y_i - p(\vec{x}_i; b_0, \vec{b})) x_{ij}$$

   c. (5) Suppose the logistic regression model is correct, when the parameters are $(\beta_0, \beta_1, \beta_2, \ldots \beta_p)$. Explain why $\mathbb{E}\left[\frac{\partial L}{\partial b_j}(\vec{x}; \beta_0, \vec{\beta}) \mid |\vec{X} = \vec{x}\right] = 0$, for all $\vec{x}$.

d. (4) Find $\frac{\partial L}{\partial b_0}$.

e. (5) Define $(\hat{\beta}_0, \widehat{\vec{\beta}}) = \text{argmin}_{b_0 \in \mathbb{R}, \vec{b} \in \mathbb{R}^p} L(b_0, \vec{b})$. Write out the first-order conditions for $\hat{\beta}_0, \widehat{\vec{\beta}}$. Simplify as much as you can (but see next question).

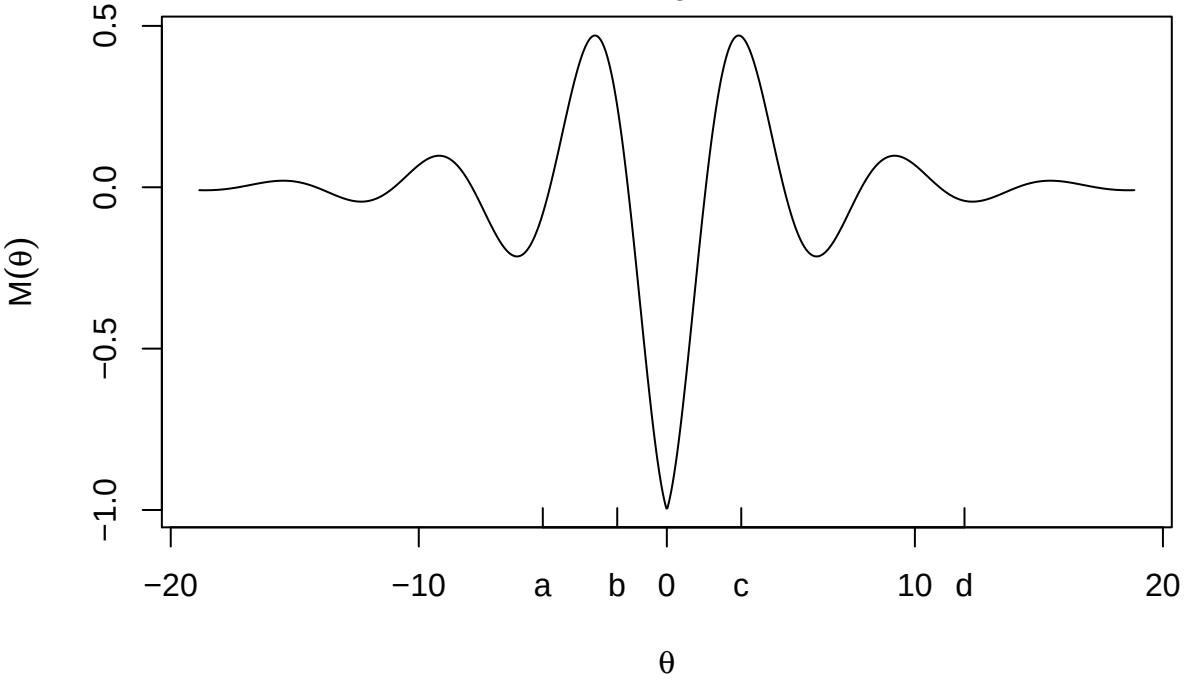
f. (5) Why will you not be able to solve the first order-conditions for $\hat{\beta}_0, \widehat{\vec{\beta}}$?

4. **Fitting a logistic regression, part 2** (10) Suppose now that the features we used in our logistic regression in Q3 are not the actual data, but nonlinear transformations of the original data z, so

$$x_{ij} = f_j(z_i)$$

for some *fixed* set of functions $f_1, f_2, \ldots f_p$. What parts of your answers to Q3 do you need to change?

5. **Gradient descent** Consider the function shown in the figure:



a. (4) Suppose we try to minimize $M(\theta)$ using gradient descent. Where (roughly) will the algorithm end up when the initial guesses are the four points marked a, b, c, d? (You can assume the "learning rate" in gradient descent is small here.)

b. (5) How does Q5a illustrate the utility of multiple random starting locations for optimization?

6. **Newton's method** Around any point θ_0, a second-order Taylor expanson of the function M is

$$M(\theta) \approx M(\theta_0) + (\theta - \theta_0) \cdot \nabla M(\theta_0) + \frac{1}{2}(\theta - \theta_0) \cdot \nabla\nabla M(\theta_0)(\theta - \theta_0)$$

a. (4) Show that the first-order condition for minimizing this approximation is met when

$$\theta_1 = \theta_0 - (\nabla\nabla M(\theta_0))^{-1} \nabla M(\theta_0)$$

(If you can only do this in one dimension, you will still get partial credit, but only partial credit.)

b. (4) What does this have to do with Newton's method?

7. **Timing** (1) How long, roughly, did you spend on this problem set?

Presentation rubric (10): The text is laid out cleanly, with clear divisions between problems and sub-problems. The writing itself is well-organized, free of grammatical and other mechanical errors, and easy to follow. Plots are carefully labeled, with informative and legible titles, axis labels, and (if called for) sub-titles and legends; they are placed near the text of the corresponding problem. All plots and tables are generated by code included in the R Markdown file. All quantitative and mathematical claims are supported

by appropriate derivations, included in the text, or calculations in code. Numerical results are reported to appropriate precision. All parts of all problems are answered with actual coherent sentences, and raw computer code or output are only shown when explicitly asked for.

Extra credit: Fitting logistic regression, part 3 Go back to the `spam` data set from homework 2. Use the training set you used in HW 2 (or, if you prefer, the training set from the solutions).

a. (2) Write a function which takes in a vector of 58 parameters, and returns the average log loss of a logistic regression of the `spam` variable on the other 57 features, with those parameters. (*Hint*: Q3.) Check that it works by seeing that it matches what you found on the training set in HW 2 (or that the solutions found).

b. (2) Using your results from Q3, write a function that calculates the gradient of the average log loss. It should take as its input a 58 dimensional vector, and return a 58 dimensional vector. It should give the zero vector, or at least a very small vector, at the parameter estimate you found in HW 2. Why is this so? Check that that's right.

c. (2) Run a *linear* regression of `spam` on the other 57 features; save (but do not display) the resulting 58 coefficients. What is the average log loss with these parameters?

d. (4) Use `optim()`, with `method="BFGS"`, to minimize your function from ECa, with gradient from ECb, and starting values from ECc. Does the final value of the objective function improve on the starting value? How close is the final location you get to what `glm()` gave you in Homework 2?