STAT 341: Assignment 2

DUE: Monday June 17, 2024 by 5:00pm ET

NOTES

Your assignment must be submitted by the due date listed at the top of this document, and it must be submitted electronically in .pdf format via Crowdmark. This means that your responses for different questions should begin on separate pages of your .pdf file. Note that your .pdf solution file must have been generated by R Markdown. Additionally:

- For mathematical questions: your solutions must be produced by LaTeX (from within R Markdown). Neither screenshots nor scanned/photographed handwritten solutions will be accepted – these will receive zero points.
- For computational questions: R code should always be included in your solution (via code chunks in R Markdown). If code is required and you provide none, you will receive zero points.
  - **Exception** any functions defined in the lecture notes can be loaded using `echo=FALSE` but any other code chunks should have `echo=TRUE`. e.g., the code chunk loading `gradientDescent` can use `echo=FALSE` but chunks that call `gradientDescent` should have `echo=TRUE`.
- For interpretation questions: plain text (within R Markdown) is required. Text responses embedded as comments within code chunks will not be accepted.

Organization and comprehensibility is part of a full solution. Consequently, points will be deducted for solutions that are not organized and incomprehensible. Furthermore, if you submit your assignment to Crowdmark, but you do so incorrectly in any way (e.g., you upload your Question 2 solution in the Question 1 box), you will receive a 5% deduction (i.e., 5% of the assignment's point total will be deducted from your point total).

QUESTION 1: Classification with Logistic Regression [15 points]

In Assignment 1 you gained some exposure to the problem of 1-dimensional classification (i.e., predicting whether $y_u = 0$ or $y_u = 1$ on the basis of the value of x_u). Here you will reconsider this same task, but instead of using the k-nearest neighbour algorithm, you're going to use *logistic regression.* Logistic regression is the counterpart to ordinary linear regression when the response variate y is binary, and it's one of the most commonly used machine learning methods for binary classification.

When $y_u \in \{0, 1\}$, the equation $y_u = \alpha + \beta x_u$ is no longer relevant since x_u can (in general) be any real number. As such, there is no reason to believe that $\alpha + \beta x_u$ will be identically 0 or 1. Nevertheless, we would still like to understand how the value of y_u depends on x_u. To do so, we instead investigate how $p_u \equiv \Pr(y_u = 1)$ is related to x_u. That said, we also cannot directly consider $p_u = \alpha + \beta x_u$ because $p_u \in [0, 1]$ and there's no reason to believe $\alpha + \beta x_u$ must be constrained in this way. As such, it's typical to model the relationship between a binary response variate y_u and a (potentially continuous) explanatory variate x_u as $f(p_u) = \alpha + \beta x_u$ where $f : [0, 1] \to \mathbb{R}$ is some function that maps the $[0, 1]$ interval to the real line. Logistic regression is specifically the model where $f(\cdot)$ is taken to be the *logit* function:

$$\log\left(\frac{p_u}{1 - p_u}\right) = \alpha + \beta x_u.$$

In this question you will learn to fit such a simple logistic regression model (i.e., estimate $\boldsymbol{\theta} = (\alpha, \beta)^T$), and use it for purposes of binary classification in the context of the same Kendrick vs. Drake data from Assignment 1.

(a) [2 points] Maximum Likelihood (ML) estimation is a commonly used method for estimating unknown model parameters, given an observed dataset. With this method, a likelihood function is maximized in order to determine which parameter values are most consistent with the observed data, assuming the underlying model is correct. The likelihood function for the logistic regression model above is:

$$L(\alpha, \beta; \mathcal{P}) = \prod_{u \in \mathcal{P}} p_u^{y_u}\,(1 - p_u)^{(1-y_u)} = \prod_{u \in \mathcal{P}} \left(\frac{p_u}{1 - p_u}\right)^{y_u} (1 - p_u) = \prod_{u \in \mathcal{P}} e^{y_u(\alpha + \beta x_u)}(1 + e^{\alpha + \beta x_u})^{-1}.$$

Derive the log-likelihood function $l(\alpha, \beta; \mathcal{P})$.

(b) [3 points] Write an R function called `logistic_nll()` that implements the *negative* of the function you derived in part (a). In particular, it should output the logistic regression's *negative* log-likelihood value when supplied the following three inputs:

- `theta`: a 2-element vector whose first element represents α and whose second element represents β.
- `x`: an N-element vector containing the explanatory variate measurements for every unit in the population.
- `y`: an N-element vector containing the response variate measurements for every unit in the population.

In addition to producing this function, explain why it must return the *negative* log-likelihood, keeping in mind that this function will be used to *maximize* the likelihood.

(c) [2 points] Use the `optim()` function discussed in class together with the `logistic_nll()` function from part (b) to find the maximum likelihood estimates of $\boldsymbol{\theta} = (\alpha, \beta)^T$, in a logistic regression model relating $y = \{0, 1\}$ ({Drake, Kendrick}) to $x =$ `Plays/1000000` (using the data found in `kendrick_v_drake.csv`). Initialize `optim()` with $\boldsymbol{\theta}_0 = (0, 0)^T$. *Hint:* You can check your answer by looking at the coefficient estimates produced by `glm(y ~ x, family = binomial(link = logit))`. Yours should be close to this.

(d) [4 points] Binary classification with logistic regression is carried out as follows. For a given x_u, we estimate $p_u = \Pr(y_u = 1)$ and hence calculate

$$\hat{p}_u = \frac{e^{\hat{\alpha} + \hat{\beta} x_u}}{1 + e^{\hat{\alpha} + \hat{\beta} x_u}}.$$

This estimated probability is then compared to a decision threshold c. If $\hat{p}_u \geq c$ then y_u is classified as 1 (i.e., $\hat{y}_u = 1$) and if $\hat{p}_u < c$ then y_u is classified as 0 (i.e., $\hat{y}_u = 0$). The most common decision threshold used is $c = 0.5$.

Using your estimated coefficients from part (c) and a decision threshold of $c = 0.5$, predict the artist of each song based on how many millions of times it's been played. Provide the confusion matrix associated with this classification, and calculate its accuracy.

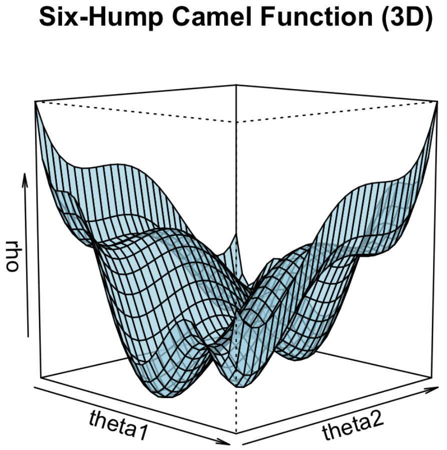
(e) [4 points] The quality of logistic regression-based classification depends on the decision threshold c. In this question you will consider many different values of c, and identify the one that appears to maximize the accuracy of the classification. In particular, perform logistic regression-based classification as in part (d), but for all `c = seq(from=0, to=1, length.out=500)`, each time calculating (and saving) the corresponding accuracy value. Then construct a plot of *accuracy* vs. c.

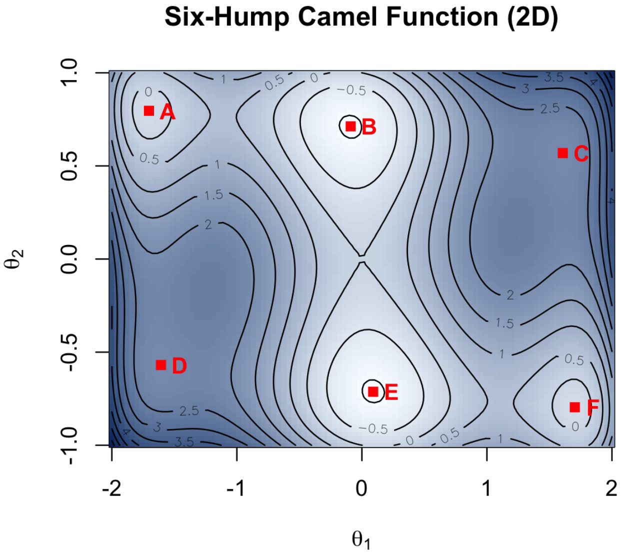
QUESTION 2: The Six-Hump Camel Function [18 points]

The Six-Hump Camel Function is one of many non-convex test functions used for evaluating the performance of optimization methods. For $\boldsymbol{\theta} = (\theta_1, \theta_2)^T \in \mathbb{R}^2$ the function is defined as follows

$$\rho(\boldsymbol{\theta}) = 4\theta_1^2 - 2.1\theta_1^4 + \frac{\theta_1^6}{3} + \theta_1\theta_2 - 4\theta_2^2 + 4\theta_2^4.$$

The figures below depict the function (as a 3-dimensional surface and with 2-dimensional contours) for $\theta_1 \in [-2, 2]$ and $\theta_2 \in [-1, 1]$.





As can be seen in these plots, the function has six minima, labeled (on the contour plot) $A = (-1.703, 0.796)$, $B = (-0.09, 0.713)$, $C = (1.607, 0.569)$, $D = (-1.607, -0.569)$, $E = (0.09, -0.713)$, $F = (1.703, -0.796)$. However, only $B = (-0.09, 0.713)$ and $E = (0.09, -0.713)$ are global minima.

(a) [2 points] By taking appropriate derivatives, determine the 2×1 gradient vector $\boldsymbol{g} = \nabla\rho(\boldsymbol{\theta})$. Show your work.

(b) [4 points] Write `rho` and `gradient` functions for the Six-Hump Camel Function which take a single vector-valued input `theta`. Use the partial derivatives calculated in part (a) in your definition of `gradient`.

(c) [4 points] In this question you will explore the surface of the Six-Hump Camel Function using gradient descent. In particular, you will consider 4 different starting values and explore the impact of changing one's starting location. Using the `gradientDescent()` function (from class) together with the `gridLineSearch()` and `testConvergence()` functions (from class) as well as the `rho` and `gradient` functions from part (b), find the solution to

$$\underset{\boldsymbol{\theta}\in\mathbb{R}^2}{\operatorname{argmin}}\, \rho(\boldsymbol{\theta})$$

for each of the following four starting values. In each case, state which minima you've converged to (A, B, C, D, E, F) and the value of the objective function at these locations. Be sure to include the output from the `gradientDescent()` function.

i. $\widehat{\boldsymbol{\theta}}_0 = (-1.5, 0)^T$

ii. $\widehat{\boldsymbol{\theta}}_0 = (-1, 0)^T$

iii. $\widehat{\boldsymbol{\theta}}_0 = (1, 0)^T$

iv. $\widehat{\boldsymbol{\theta}}_0 = (1.5, 0)^T$

(d) [5 points] Recreate the contour plot shown above. You may find the functions `outer()`, `image()`, and `contour()` useful for this task. Include on this plot `gold` circles at each of the starting points specified in (c) as well as `gold` `arrows()` tracing the iterative paths connecting these starting points with their respective points of convergence. Note you will need to modify the `gradientDescent()` function from class to save (and return) the iterates.

(e) [2 points] Based on what you found in part (c), and visualized in part (d), explain the importance of the starting value when performing non-convex optimization (when locating a global optimum is desired).

(f) [1 point] Repeat part (c) but start at $\widehat{\boldsymbol{\theta}}_0 = (0, 0)^T$. This should result in an error. Explain why.

QUESTION 3: GameStop Stock Volatility [23 points]

GameStop is an American retailer specializing in video game hardware, video game software, video game accessories, mobile and consumer electronics, and pop culture merchandise. Since February 2002, GameStop shares have been publicly traded on the New York Stock Exchange under the ticker symbol "GME" (NYSE: GME). In 2020, a redditor named Keith Gill noticed that the GameStop stock was heavily shorted[1] (primarily by institutional investors) even though the underlying fundamentals of the company didn't support the theory that GameStop was about to go bankrupt. In January of 2021, the stock price jumped from $1/share to $5/share. This gained traction on Reddit, particularly among the redditors on the r/wallstreetbets subreddit. As a result, retail investors[2] started buying into the stock at unprecedented rates anticipating a "short squeeze"[3], and this buying frenzy drove the stock price up to a peak value of $347.51/share on January 27, 2021. Within a few weeks the stock price had fallen dramatically (due in part to Robinhood halting purchasing of the stock), but many retail investors who took advantage of the short squeeze profitted handsomely. For instance, Keith Gill posted a final account update in April of 2021 showing that the value of his holdings had increased from $50,000 to $34,473,248.01. Sony Pictures made a movie about this saga.

In July of 2022, GameStop had a 4-for-1 stock split, multiplying it's share count and decreasing the price. The stock closed at $153.47/share on the day of the split and reopened at $38.37/share. By April of 2024 the stock was trading under $11 per share. However, earlier this month, the GameStop share price spiked again, peaking on May 14 at $64.83 (an increase of roughly 500% from two weeks prior, and equivalent to $259.32 in pre-split prices). Speculation is rampant as to what caused the recent price increase, ranging from short sellers closing positions, regulatory changes, LEAP expiry, and the re-emergence of Keith Gill, who has recently begun posting on social media again after a several-year hiatus. However, no one actually knows why these significant price changes have occurred.

In this question you will analyze the daily adjusted closing price of GME over the past year. The data available to you is stored in the `GME.csv` file and summarized below. For each of the $N = 253$ trading days in the last year (i.e., between May 24, 2023 – May 25, 2024 inclusive) we have observations for each of the following variates:

Variate	Description
`Date`	A string recorded as `YYYY-MM-DD` indicating the date.
`Day_Num`	An integer counter starting at 1 and ending at 253.
`Open`	The opening price of GME in US dollars on a given day.
`High`	The highest value of GME in US dollars on a given day.
`Low`	The lowest value of GME in US dollars on a given day.
`Close`	The closing price of GME in US dollars on a given day.
`Adj_Close`	The adjusted closing price of GME in US dollars on a given day.
`Volume`	The number of GME shares traded on a given day.

In this question, you will consider modeling the relationship between $y =$ `Adj_Close` and $x =$ `Day_Num` with the following simple linear regression:

$$y_u = \alpha + \beta(x_u - \bar{x}) + r_u, \quad u \in \mathcal{P}, \quad \alpha, \beta \in \mathbb{R}$$

[1] The stock being heavily shorted means that many traders had bet the stock price would plummet to $0 per share.

[2] Retail investors are average people like you and I who do not trade professionally.

[3] A short squeeze is when the price of an asset rises sharply, causing traders who sold short to close their positions at a loss.

(a) [3 points] Construct a scatter plot of $y =$ `Adj_Close` versus $x =$ `Day_Num`, and add the least squares regression line $\hat{y}_u = \hat{\alpha} + \hat{\beta}(x_u - \bar{x})$ to this plot. Note that you may use the `lm()` function to determine the equation of this line. Be sure to add a title and informative axis labels.

(b) [6 points] For each unit (i.e., day) in the population, calculate the influence that it has on the fitted regression line from part (a), using each of the following three definitions of influence:

- $\Delta(\alpha, u) = ||\widehat{\alpha} - \widehat{\alpha}_{[-u]}||_1$
- $\Delta(\beta, u) = ||\widehat{\beta} - \widehat{\beta}_{[-u]}||_1$
- $\Delta(\boldsymbol{\theta}, u) = ||\widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}_{[-u]}||_1$

where $\widehat{\boldsymbol{\theta}} = \left(\widehat{\alpha}, \widehat{\beta}\right)^T$ are the regression coefficients estimated from all of the data, $\widehat{\boldsymbol{\theta}}_{[-u]} = \left(\widehat{\alpha}_{[-u]}, \widehat{\beta}_{[-u]}\right)^T$ are the regression coefficients estimated from all of the data *excluding* unit u, and $||\cdot||_1$ is the L_1-norm. Construct three scatterplots, one for each of these influence values, and determine the four days that had the largest influence on this regression.

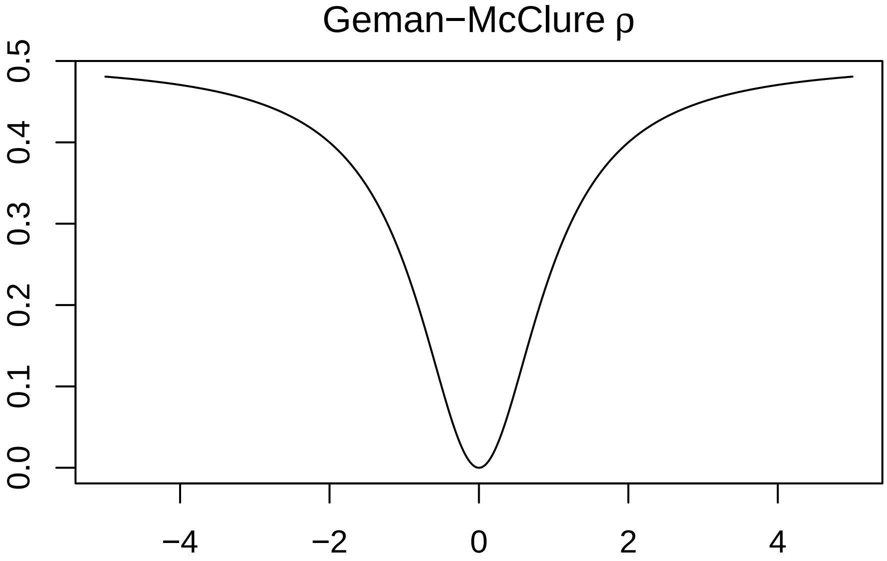
(c) [14 points] One way of mitigating the influence of highly influential observations is to perform a robust linear regression. In class we saw that *robust regression* is an outlier-resistant means to estimate $\boldsymbol{\theta} = (\alpha, \beta)^T$ in the context of the simple linear regression model defined above. One particular objective function that facilitates robust regression is the so called **Geman-McClure objective function** :

$$\rho(\boldsymbol{\theta}; \mathcal{P}) = \sum_{u \in \mathcal{P}} \rho(r_u)$$

where $\boldsymbol{\theta} = (\alpha, \beta)^T$, $r_u = y_u - \alpha - \beta(x_u - \bar{x})$ and

$$\rho(r) = \frac{r^2/2}{1 + r^2}.$$

This function is shown below. In the questions that follow you will fit a robust linear regression using the Geman-McClure objective function.



i. [4 points] By taking appropriate derivatives, determine the 2×1 gradient vector $\boldsymbol{g} = \nabla\rho(\boldsymbol{\theta}; \mathcal{P})$. Show your work.

ii. [4 points] Write *factory functions* `createRobustGMRho(x, y)` and `createRobustGMGradient(x, y)` which take in as inputs the data and which respectively return as output the Geman-McClure objective function and the corresponding gradient function. **Hint:** Use the `createRobustHuberRho()` and `createRobustHuberGradient()` functions from the lecture notes as a guide.

iii. [2 point] Using the `gradientDescent()` function from class with the `rho` and `gradient` functions created by your factory functions from part ii., find $\widehat{\boldsymbol{\theta}} = \left(\widehat{\alpha}, \widehat{\beta}\right)^T$, the solution to

$$\underset{\boldsymbol{\theta} \in \mathbb{R}^2}{\operatorname{argmin}}\, \rho(\boldsymbol{\theta}; \mathcal{P}).$$

Start the optimization at the least squares slope and intercept estimates from part (a) (i.e., $\widehat{\boldsymbol{\theta}}_0 = (\widehat{\alpha}_{LS}, \widehat{\beta}_{LS})^T$). For full points be sure to include the output from the `gradientDescent()` function.

iv. [4 points] Re-construct the scatter plot from part (a). In a second colour, add the Geman-McClure regression line calculated in part iii. Be sure to include a legend that distinguishes the two lines. Comment on the difference between these two lines in relation to your findings in part (b).