MATH3027 Optimization 2021: Coursework 1

This is the first piece of assessed coursework for MATH3027 Optimization 2021. It is worth **20% of your final mark** for the module.

The deadline for submission is Thursday 11 November at 10am. Your solution should be submitted electronically as a pdf file via the MATH3027 Moodle page. Late submissions will be subject to the usual penalties.

Since this work is assessed, **your submission must be entirely your own work** (see the University's policy on Academic Misconduct). You can use, without acknowledgement, any of the material or code from the lecture notes or computer labs.

There are three questions below - you must answer all the questions. A handwritten and scanned solution is acceptable for question 1. For question 2 and 3 you must submit the code you use to find your solution. The easiest way to do this is to use Rmarkdown to create a pdf file.

This assignment will be marked out of 20.

Question 1 (4 marks)

Consider the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined as

$$f(x, y) = 2y^4 - x^2 + 1 + (x^2 + 2y^2 - 1)^2.$$

Find all of the stationary points of f, and classify them as local/global, strict/non-strict, and maximum/minimum or saddle points.

Question 2 (10 marks)

Logistic regression is a form of statistical model used to model the probability of a binary event occuring as a function of some covariate information. In this question we will use logistic regression to predict the probability an individual has diabetes, using the covariate information

- `pregnant`: the number of times the individual has been pregnant
- `glucose`: the plasma glucose concentration in an oral glucose tolerance test
- `pressure`: Diastolic blood pressure (mm Hg)

We will use a subset of the Pima Indians dataset which were collected by the National Institute of Diabetes and Digestive and Kidney Diseases. You can download from Moodle and load it into R as follows:

```r
load(file='CW1_PimaData.rda')
head(X)
```

```
##   pregnant glucose pressure
## 1        6     148       72
## 2        1      85       66
## 3        8     183       64
## 4        1      89       66
## 5        0     137       40
## 6        5     116       74
```

```r
head(y)
```

```
##   diabetes
## 1        1
## 2        0
## 3        1
## 4        0
## 5        1
## 6        0
```

This contains information on $n = 100$ females of Pima Indian heritage, who are at least 21 years old. The vector $\mathbf{y} \in \mathbb{R}^{100}$ is our response vector, with y_i the binary response for woman i,

$$y_i = \begin{cases} 1 & \text{if woman i has diabetes} \\ 0 & \text{otherwise} \end{cases}$$

The matrix $\mathbf{X}$ is a 100×3 matrix of covariate information:

$$\mathbf{X} = \begin{pmatrix} - & \mathbf{x}_1^\top & - \\ & \vdots & \\ - & \mathbf{x}_{100}^\top & - \end{pmatrix} \in \mathbb{R}^{100\times 3}, \qquad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{100} \end{pmatrix} \in \mathbb{R}^{100}.$$

Our aim is to build a logistic regression model to predict whether a patient has diabetes or not using the covariate information. The logistic regression model takes the form

$$\begin{aligned} \mathbb{P}(y_i|\mathbf{x}_i, \boldsymbol{\theta}) &= \begin{cases} h(\mathbf{x}_i^\top \boldsymbol{\theta}) & \text{if } y_i = 1 \\ 1 - h(\mathbf{x}_i^\top \boldsymbol{\theta}) & \text{if } y_i = 0 \end{cases} \\ &= h(\mathbf{x}_i^\top \boldsymbol{\theta})^{y_i} \left(1 - h(\mathbf{x}_i^\top \boldsymbol{\theta})\right)^{1-y_i}. \end{aligned}$$

Here $\boldsymbol{\theta} \in \mathbb{R}^3$ is an unknown parameter vector that determines the importance of each piece of covariate information, and h is the inverse *logit* function:

$$h : \mathbb{R} \to [0, 1] \text{ with } h(t) = \frac{1}{1 + \exp(-t)}.$$

Our aim in this question is to learn the unknown parameters $\boldsymbol{\theta}$ by maximum-likelihood estimation. The log-likelihood of the parameter $\boldsymbol{\theta}$ is

$$\begin{aligned}\ell(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}) &= \log \prod_{i=1}^{n} P(y_i \mid x_i, \boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} y_i \log h(\mathbf{x}_i^\top \boldsymbol{\theta}) + (1 - y_i) \log(1 - h(\mathbf{x}_i^\top \boldsymbol{\theta}))\end{aligned}$$

and we want to find the value of $\boldsymbol{\theta}$ that maximizes $\ell(\boldsymbol{\theta})$ (or minimizes the negative log-likelihood). There is an in-built command for doing this in R:

```
fit <- glm(diabetes~.-1, family=binomial, data=Pima.dat)
coef(fit)
```

```
##     pregnant      glucose     pressure
##  0.056509905  0.008079294 -0.022841519
```

but we will write our own routine. You can compute the log-likelihood using the following R function.

```
h <- function(tt){
  1/(1+exp(-tt))
}

X_pos = X[y==1,]
X_neg = X[y==0,]

loglike <- function(theta){
  sum(log(h(X_pos%*%theta)))+sum(log(1-h(X_neg%*%theta)))
}

#
# NOTE: You might think about coding this function as
#  sum(y*log(h(X%*%theta))+(1-y)*log(1-h(X%*%theta)))
# but this is more likely to return numerical errors (particularly NaNs)
# as log(0)=NaN and although this should get cancelled by the (1-y)
# the computer isn't clever enough to figure this out.
```

i. Show that the gradient of $\ell(\boldsymbol{\theta})$ is

$$\nabla \ell(\boldsymbol{\theta}) = \mathbf{X}^\top (\mathbf{y} - \hat{\mathbf{y}}(\boldsymbol{\theta}))$$

where

$$\mathbf{X} = \begin{pmatrix} - & \mathbf{x}_1^\top & - \\ & \vdots & \\ - & \mathbf{x}_n^\top & - \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \qquad \hat{\mathbf{y}}(\boldsymbol{\theta}) = \begin{pmatrix} h(\mathbf{x}_1^\top \boldsymbol{\theta}) \\ \vdots \\ h(\mathbf{x}_n^\top \boldsymbol{\theta}) \end{pmatrix}.$$

Write a function to compute the gradient. Your answer should be a vector of length 3. Check your answer numerically using a finite difference approximation. What is the gradient of $\ell(\boldsymbol{\theta})$ at $\boldsymbol{\theta} = (0,\ 0,\ 0)^\top$?

ii. Implement the gradient descent method to find the maximum likelihood estimates of $\boldsymbol{\theta}$, using $\boldsymbol{\theta}_0 = (0, 0, 0)^\top$ as an initial starting point. Use a fixed stepsize of $\bar{t} = 10^{-6}$, using

$$||\nabla \ell(\boldsymbol{\theta})|| < 10^{-3}$$

as the stopping criterion. How many iterations are required for the algorithm to converge? Plot the trajectory of the three parameters.

Note: we want to find the maximum here. The methods described in the notes were designed to find minima.

iii. What happens if you use a much larger step-size, e.g., $\bar{t} = 10^{-5}$?

iv. Instead of using a constant stepsize, use the gradient method with backtracking with parameters $s = 1, \alpha = 0.25, \beta = 0.5$. Does this work well? Try other values of s, α and β and discuss what you find.

v. Compute the Hessian matrix. Code a function to compute the Hessian and check your calculations numerically.

vi. Implement a Pure Newton's method to find the maximum likelihood estimate of $\boldsymbol{\theta}$. Starting from $\boldsymbol{\theta}_0 = (0,\ 0,\ 0)^\top$, and using

$$||\nabla \ell(\boldsymbol{\theta})|| < 10^{-3}$$

as the stopping criterion, how many iterations does it take for the algorithm to converge?

vii. In statistics and machine learning, models are often **regularized** in order to improve the prediction accuracy by stopping overfitting. In Tikhonov or ridge regularization, instead of finding parameters to maximize the log-likelihood $\ell(\boldsymbol{\theta})$, we instead try to find parameter to minimize

$$-\ell(\boldsymbol{\theta}) + \lambda ||\boldsymbol{\theta}||_2^2.$$

Here λ is a parameter that controls the importance of the regularization term. If $\lambda = 0$ then this is equivalent to the maximum likelihood case. As $\lambda \to \infty$, the parameter estimates tends to 0.

Implement a Newton method to solve the regularized logistic regression problem, and find the optimal $\boldsymbol{\theta}$ when $\lambda = 10^3$.

Plot the estimated value of the parameters as λ varies from 1 to 10^6, using a log-scale for λ.

Question 3 (6 marks)

You wake up and find yourself on a strange planet. You find a spaceship and wish to fly home. But before you can safely do so, you need to estimate the acceleration due to gravity, g, and the drag coefficient of air-resistance, k. Thankfully, you remember your physics training and recall that for an object in freefall near the surface of the planet, the position of the object at time t, $x(t)$ say, obeys the differential equation

$$\frac{d^2x}{dt^2} + k\frac{dx}{dt} + g = 0.$$

For parameters g and k, if we drop an object from a height of 20m at time $t = 0$ (so that $x(0) = 20$ and $\frac{dx}{dt}(0) = 0$) then at time t, it will be at height

$$f(t; g, k) = 20 - \frac{g}{k}\left(t + \frac{1}{k}\exp(-kt) - \frac{1}{k}\right).$$

In order to estimate g and k, you set up an experiment, dropping a weight from a height of 20m, and then you observe (noisily) its location at times

$$t_1 = 0.25,\ t_2 = 0.5,\ t_3 = 0.75, \dots,\ t_8 = 2.$$

Your measurements of the height of the weight at these times are

$$y_1 = 19.956m,\ y_2 = 17.528m,\ y_3 = 15.987m,\ y_4 = 14.445m,\ y_5 = 9.631m,\ y_6 = 6.663m,\ y_7 = 2.134m,\ y_8 = 0.121m$$

We want to use these measurements to estimate g and k. We will do this using non-linear least squares and solve the optimization problem

$$\min_{g,k} \sum_{i=1}^{b} (f(t_i; g, k) - y_i)^2$$

- Code a function to compute the objective function above as a function of $\theta = (g, k)^\top$. Compute its gradient and implement a function to calculate this. Check your answer numerically.
- Create a contour plot of
  $$s(g, k) = \sum_{i=1}^{b} (f(t_i; g, k) - y_i)^2$$
  for $g \in [1, 20]$ and $k \in [0, 5]$.
- Implement a pure Gauss-Newton algorithm to find the optimal values of g and k, starting from $g = 10, k = 2$, and using $||\nabla g(\theta)|| < 10^{-3}$ as the stopping criterion. Report the number of iterations taken until convergence, the optimal values of g, k, and of the value of the objective at the optima.