Uppsala University

Introduction to Machine Learning, Big Data, and AI

Assignment 1

General information

- The recommended tool in this course is R (with the IDE R-Studio). You can download R **here** and R-Studio **here**.
- Report all results in a single, *.pdf -file. *Other formats, such as Word, Rmd or similar will automatically be failed.*
- The report should be submitted to the Student Portal.
- When working with R, we recommend writing the reports using R markdown and the provided **R markdown template**. The remplate includes the formatting instructions and how to include code and figures.
- If you have problem with creating a PDF file directly from R markdown, start by creating an HTML file and the just print the HTML to a PDF.
- Instead of R markdown, you can use other software to make the pdf report, but the the same instructions for formatting should be used. These instructions are available also in **the PDF produced from the R markdown template**.
- We collect common questions regarding installation and technical problems in a course Frequently Asked Questions (FAQ). This can be found **here**.
- Deadline for all assignments are **Sunday at 23.59**. See the course page for dates.
- If you have any suggestions or improvements to the course material, please post in the course chat feedback channel, create an issue, or submit a pull request to the public repository!

1 Gradient Descent, Stochastic Gradient Descent and Adam

In this assignment we will study different ways to optimize the objective functions that is common in many areas of Machine Learning, namely Stochastic gradient descent. Here we will test to implement these optimizers for a well known approach, logistic regression.

We are going to work with this data as at test case:

```
bin_data <- read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")

mydata$gre_sd <- (mydata$gre - mean(mydata$gre))/sd(mydata$gre)
mydata$gpa_sd <- (mydata$gpa - mean(mydata$gpa))/sd(mydata$gpa)
X <- model.matrix(admit ~ gre_sd + gpa_sd, mydata)
y <- mydata$admit
```

1.1 Implement the gradient for logistic regression

The likelihood function for logistic regression is

$$L(\theta, \mathbf{y}, \mathbf{X}) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}\,,$$

where

$$\log\frac{p_i}{1-p_i} = \mathbf{x}_i\theta\,,$$

and $\mathbf{x}_i$ is the ith row from the design matrix $\mathbf{X}$ and $\theta \in \mathbb{R}^P$ is a $1 \times P$ matrix with the parameters of interest.

Commonly, to find maximum likelihood estimates of θ we usually use the log likelihood as the objective function we want to optimize, i.e.:

$$\begin{aligned}
l(\theta, \mathbf{y}, \mathbf{X}) &= \sum_{i=1}^{n} y_i \log(p_i) + (1-y_i)\log(1-p_i) && (1)\\
&= \sum_{i=1}^{n} y_i\mathbf{x}_i\theta + \log(1-p_i) && (2)\\
&= \sum_{i=1}^{n} y_i\mathbf{x}_i\theta - \log(1+\exp(\mathbf{x}_i\theta))\,. && (3)
\end{aligned}$$

Although, in our case we instead want to minimize the negative log likelihood $\mathrm{NLL}(\theta, \mathbf{y}, \mathbf{X}) = -l(\theta, \mathbf{y}, \mathbf{X})$.

1. Derive the the gradient for $\mathrm{NLL}(\theta, \mathbf{y}, \mathbf{X})$ with respect to θ.
2. Implement the gradient as a function in R. Below are two examples of how it should work.

```
l_grad(y, X, theta = c(0,0,0))

## (Intercept)      gre_sd      gpa_sd
##     -0.1825      0.0857      0.0829
```

```
l_grad(y, X, theta = c(-1,0.5,0.5))

## (Intercept)      gre_sd      gpa_sd
##      0.0217     -0.0395     -0.0426
```

1.2 Implement Gradient Descent

We now have the main tool for implementing gradient descent and stochastic gradient descent.

1. Implement the log likelihood l or the negative log likelihood NLL in R.

   ```
   l(y, X, theta = c(0,0,0))

   ## [1] -277.2589
   ```

   ```
   l(y, X, theta = c(-1,0.5,0.5))

   ## [1] -244.5342
   ```

2. Run logistic regression in R to get a MLE estimate.
3. Implement the following gradient descent algorithms:
   (a) ordinary (full/batch) gradient descent
   (b) stochastic gradient descent
   (c) mini-batch (stochastic) gradient descent using 10 samples to estimate the gradient
4. Try different learning parameters η when do the algorithm converge or diverge? Visualize the iterations (x-axis) and the log likelihood (y-axis). Show at least one plot per algorithm that converge. Describe your conclusions. Show the code (loop) you use.