Parallelizing Logistic Regression

Wansang Lim and Valerie Angulo

Abstract— **Logistic regression is a statistical analysis tool used as a predictive analytic in a variety of disciplines. In this paper, we focus on parallelizing binary logistic regression analysis for predictive analytics in data science for environmental science datasets. Parallelizing logistic regression would improve computation time and decrease memory latency when analyzing large data sets, allowing for more data to be processed faster. In this paper, we compare a sequential implementation of binary logistic regression with a CUDA implementation, a parallelized R implementation and a multiprocessor OpenCV version, looking at the relationship between dataset sizes and time taken to process the data. Our findings point to improvements in computation time and less memory latency for CUDA versions of logistic regression.**

I. INTRODUCTION

Logistic Regression is used as a predictive analytic in many disciplines ranging from biology and conservation to business. It models a binary dependent and one or more binary or nonbinary independent variables. This is useful in cases of observing phenomena that may occur due to a specific event. The purpose of logistic regression is to predict the occurrence of phenomena based on acquired current data. Mathematically, the binary dependent variable is either 0 or 1 and indicates the presence or absence of a certain condition, such as alive/dead or win/lose, that may be related to the independent conditions. In this paper, we are primarily concerned with the applications of binary logistic regression analysis for predictive analytics in data science, particularly for environmental science datasets. For our purpose, logistic regression is used to classify dependent variables into different groups.

Currently, there is an abundance of large datasets that are open sourced and easily accessible to the public. This is especially beneficial for scientific research. However, processing large data sets is time consuming and resource intensive for CPU in terms of memory and computation time. Sequential implementations of logistic regression require a lot of time to process smaller amounts of data and have a high memory latency. Implementation of a parallelized binary logistic regression would allow for an increased amount of data to be processed in less time and with less resource intensive computations. Logistic regression is an excellent analytic to parallelize because it primarily utilizes matrix multiplication, which is easy to convert to parallel code. It also utilizes an inverse function which is a variation of matrix multiplication and determines the natural log of a matrix, a function that is easily supported by parallelism. In this paper, we compare a sequential implementation of binary logistic regression with a CUDA implementation, a parallelized R implementation and a multiprocessor OpenML(???) version, looking at the relationship between dataset sizes and time taken to process the data. Our findings point to improvements in computation time and less memory latency for CUDA versions of logistic regression, as well as parallelized R implementations.

II. BACKGROUND

Logistic regression is a predictive analytic that is used to categorize data into different groups. It can be binomial, ordinal or multinomial and is based on linear regression; a logistic regression estimates a multiple linear regression function. It is the correct regression to use if there is a binary dependent variable, such as time spent studying versus whether a student passes or fails a course, or other situations depending on independent variables such as win/lose, dead/alive, yes/no, present/absent etc. Binary logistic regression is used to describe the relationship between a binary dependent variable and one or many independent variables. This predictive analytic tool is useful in data science and many other fields for predicting how likely an event will occur when certain independent factors are present. Following are the steps needed to implement a logistic regression.

A. Linear Regression

In order to do a logistic regression, we must first start with a linear regression.

$$Y = b_0 + b_1 X_1 + b_2 X_2 + b_n X_n$$

In this linear model, the xs are the predictors/independent variables and the Bns are the parameters of the model or the coefficients of the independent variables, with b_0 being a constant term. The Y value is the outcome /dependent variable and can vary from negative to positive infinity for a linear regression.

B. Logistic Function

The next step is to turn the linear regression into a sigmoid function, also known as a logistic function.

$$p(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x} = \frac{e^{(b_0 + b_1 X_1 + ... + b_n X_n)}}{1 + e^{(b_0 + b_1 X_1 + ... + b_n X_n)}}$$

P(x) is the probability of the dependent variable equaling a success. E(Y) is the expected value of the binary dependent variable Y. X is the linear regression formula. The function is as such because the range of p(x) should be between 0 to 1, rather than -infinity to +infinity. This function provides a limited range of values for probability, so the odds of the probability must be taken next to fully convert this logistic function into a logistic regression.

C. Logistic Regression

The logistic regression is the natural log of the inverse of the logistic function. The core of logistic regression is estimating the log odds of an event, also known as the logit of the probability, where the log odds is a prediction of the odds of a dependent variable based on one or more binary or real valued independent predictors/x-values.

The odds are determined by the inverse of the logistic function, where the probability of an outcomes success (Y) is divided by the probability that it will not occur (1-Y). To determine the log-odds of an event occurring, we must take the natural log of this inverse. The purpose of this is to fit the logit of the probability of success with the predictors/x-values. We are then able to obtain a continuous predictor for the odds of an event happening.

$$logitp(x) = ln[\frac{p(x)}{1 - p(x)}]$$

The dependent variable turns into a logit variable (the natural log of the odds of the dependent variable occurring or not). We then estimate the probability of the occurrence of a certain event based on the independent variables. The logit serves as a link between the linear regression and logistic function. The probability of success of obtaining a particular value from a binary dependent is equivalent to the odds of the dependent variable Y equaling a particular case.

$$ln[\frac{Y}{1 - Y}] = b_0 + bX$$

Logistic regression seeks to find the equation that best predicts the value of Y for each value of X. The Y variable in logistic regression is the probability of obtaining a particular value of a binary variable, whereas the Y variable in linear regression is measured directly.

III. LITERATURE SURVEY

We've looked at a variety of papers regarding parallelizing logistic regression, as well as the applications of parallelized and sequential logistic regressions. The following related works are presented because they either implement a bio-statistic(?) model of logistic regression or they compare sequential and parallel versions of logistic regression.

A. Biostatistic Logistic Regression

The algorithm that we have coded sequential and parallel versions of is applicable to biology and environmental studies. The work done by (Deforestation modelling using logistic regression and GIS) uses logistic regression in the manner that we implement and parallelize. In their work, they(NAMES!) look at binary logistic regression, determining whether deforestation is present or not in an area based on slope as an independent variable. The setup of the experiment is similar to ours, and their work is what drove us to parallelize logistic regression for scientific fields. While they analyze a sequential version of the logistic regression, there is no attempt to parallelize it. The data input is also far different than ours, where the focus is to use GIS with logistic regression, making comparison with their work difficult. They(NAMES!) analyze digital thematic-topographic maps with 63 different data layers such as forests, ranges and gardens. Our tests read data from a file containing random input values that fall within a certain range to represent our x and b values, rather than topographic maps or open source data.

B. Parallelizing Logistic Regression

Others, such as (NAMES liang, choi)(Evaluating Parallel Logistic Regression Models citation!), have parallelized the logistic regression for machine learning. Liang et. all(?) have looked at parallelized regressions in distributed platforms, parallel algorithms and sub-linear approximations. They worked with Hadoop and Spark, two distributed systems used in data science and analytics, to run sequential and parallel versions of code and compare platform/algorithm combinations with sub-linear algorithms specific to machine learning. This is clearly useful for machine learning but is not applicable to other fields that utilize logistic regression. Liang et all focus on parallelizing machine learning algorithms, while we are looking to parallelize logistic regression itself. They used five open datasets related to machine learning to make sure that they could accurately compare their methods whereas our data is randomly generated and doesnt come from open source. Using open datasets are beneficial because the results allow for comparability with the performance of sequential machine learning algorithms that already exist.

The work done by (NAMES) in (Breast Cancer Prediction by Logistic Regression with CUDA Parallel Programming Support)(CITE) focuses on parallelizing logistic regression for the purposes of machine learning, bioinformatics and data analysis. While it is primarily a machine learning approach and our work aims to be more generalized, the parallelization is reached by using CUDA parallel programming support, which we use in our experimental design as well. However, not much detail is gone into about how CUDA is implemented, only that different CUDA software versions are compared in determining the loss function for the machine learning logistic regression. Nevertheless, this work is useful in that it this approach is applicable to bioinformatics and data analysis. The dataset used is a real dataset containing anonymized patient information obtained from the Wisconsin Diagnostic Breast Cancer database. It contains 569 instances, with each instance corresponding to a patient (CITE!!!). This is beneficial in that the results obtained can be tested in real life, and they are. From the estimated Y values obtained per patient, there was a better chance at predicting whether a patient had breast cancer or not.

C. Implementations Used or Referenced

In Evaluating Parallel Logistic Regression Models (CITE), Liang and Choi implement a binary logistic regression, and we reference the parallel algorithm approach, but without the distributed platform or sub-linear approximation. They analyze their experimental results in accuracy, efficiency,

scalability, robustness and running time. We analyze our results in terms of accuracy and running time and use this work as a reference on how to proceed with such an analysis.

From (NAME et. alls) work in Deforestation modelling using logistic regression and GIS, we use a similar logistic regression and proceed with the sequential code and then the parallelized code in a step-by-step approach as well. We break down our implementation into linear regression, then logistic function and finally into logistic regression in order to code logistic regression. (NAMES) use R^2 chi square test(?) to determine the fitness of their model, which is something we consider in our work but did not have enough time to do.

IV. PROPOSED SOLUTION

We aim to parallelize the logistic regression as applied to biology and data science by using CUDA. There has been a lot of research on parallelizing logistic regression for machine learning techniques and algorithms, but nothing that weve seen that is general enough to cover parallelized logistic regression outside of machine learning purposes. The biostatistic research that we have come across uses sequential logistic regression with other new technologies but does not attempt to parallelize the logistic regression itself. From our research, we havent found others who attempt to parallelize a general logistic regression using CUDA. Therefore, we propose to parallelize the logistic regression in order to improve predictive behavior in a number of fields. The purpose of this is to show that logistic regression can be parallelized and used to analyze more data in less time.

V. EXPERIMENTAL SETUP

A. Assumptions

For our experiment, we are using a binary logistic regression, where we are dealing with a dependent variable that can only be either 0 or 1, whereas our independent variables are real numbers. To make sure our generated data set was good, we followed the assumptions of data used for logistic regression applications such as that the dependent variable is binary, there are no outliers in the data or strong correlations between independent data sets and that there were no values present in the data that were below -3.29 or above 3.29. (https://www.statisticssolutions.com/what-is-logistic-regression/). Another consideration we had to take into account was the model fit, having more independent variables increases the amount of variance (R^2) but adding too many variables will decrease the generalizability of the predictive analytic, rendering it less accurate. To analyze the accuracy of our logistic regression, we computed its goodness-of-fit based on the Chi square test, as well as the amount of variance R^2.

B. Methods

How we parallelized the code step by step

C. Data Collection and Code

The logistic regression model is useful in determining the relationship between a random value and its covariants. Therefore, our first step was to create a file of random generated values that fit the assumptions of input data of logistic regression. We used randomly generated values for our input with the purpose of finding Y values according to these values. This ascertained that the independent x-variables are independent within themselves, one of the assumptions needed for data used in a logistic regression. The B values are produced by the regression data program. From this data, we populate an array of new X values which, along with the B values, will give us a predicted Y value.

From our output file of x by y values, we ran the data to be processed to find the predicted Y value for the sequential version of the logistic regression and the parallelized version of the logistic regression. We coded a sequential version of a linear regression and then modified it to be a logistic regression based on the steps in the background section. We then parallelized the code where we could, such as for matrix multiplication and calculating the inverse of a matrix.

D. Machine Setup

We used NYU linux CIMS accounts to run our sequential and parallelized logistic regression code. Before running the sequential and parallelized code, we made sure to generate the random data files that we needed to run our code with. To run the parallelized code, we used the cuda5 machine on the NYU prince cluster and loaded the cuda version 9.1 module. For the sequential code, we used the snappy machine on NYUs prince cluster to obtain faster results for sequential code. We used the time flag to record time taken to compute the logistic regression of our sequential code and CUDA code, and we used nvprof to profile our CUDA code.

VI. RESULTS AND DISCUSSION

Add stuff...

A. Figures and Tables

Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation Fig. 1, even at the beginning of a sentence.

TABLE I
AN EXAMPLE OF A TABLE

One	Two
Three	Four

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an document, this method is somewhat more stable than directly inserting a picture.

Fig. 1. Inductance of oscillation winding on amorphous magnetic core versus DC bias magnetic field

an example, write the quantity Magnetization, or Magnetization, M, not just M. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write Magnetization (A/m) or Magnetization A[m(1)], not just A/m. Do not label axes with a ratio of quantities and units. For example, write Temperature (K), not Temperature/K.

VII. CONCLUSIONS

Summary of what we found, why its useful, future implementations or things we'd do or change.

- final bullets
- final bullets
- final bullets
- final bullets

ACKNOWLEDGMENT

Thank you Professor Zahran for all your help.

REFERENCES

[1] Bavaghar, M. Pir. "Deforestation Modelling Using Logistic Regression and GIS." Journal of Forest Science 61, no. No. 5 (2016): 193-99.
[2] Peng, Haoruo, Ding Liang, and Cyrus Choi. "Evaluating Parallel Logistic Regression Models." 2013 IEEE International Conference on Big Data, 2013.
[3] Peretti, Alessandro, and Francesco Amenta. "Breast Cancer Prediction by Logistic Regression with CUDA Parallel Programming Support." Breast Cancer: Current Research 01, no. 03 (2016).
[4] Salame, Camil Wadih, Joaquim Carlos Barbosa Queiroz, Gilberto De Miranda Rocha, Mario Miguel Amin, and Edson Paulino Da Rocha. "Use of Spatial Regression Models in the Analysis of Burnings and Deforestation Occurrences in Forest Region, Amazon, Brazil." Environmental Earth Sciences 75, no. 3 (2016).
[5] Singh, Sameer, Jeremy Kubica, Scott Larsen, and Daria Sorokina. "Parallel Large Scale Feature Selection for Logistic Regression." Proceedings of the 2009 SIAM International Conference on Data Mining, 2009, 1172-183.
[6] Sperandei, Sandro. "Understanding Logistic Regression Analysis." Biochemia Medica, 2014, 12-18.
[7] Van, Phu Nguyen, and Thophile Azomahou. "Nonlinearities and Heterogeneity in Environmental Quality: An Empirical Analysis of Deforestation." Journal of Development Economics 84, no. 1 (2007): 291-309.