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.

$$Y = b_0 + b_1X_1 + b_2X_2 + b_nX_n$$

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.

$$p(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x} = \frac{e^{(b_0+b_1X_1+\ldots+b_nX_n)}}{1+e^{(b_0+b_1X_1+\ldots+b_nX_n)}}$$

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.

A. What have the others do to solve this problem and What are the pros and cons of this previous work

Evaluating Parallel Logistic Regression Models - liang, choi—- Others have parallelized the logistic regression in terms of machine learning problems and have looked at parallelized regressions in distributed platform, parallel algorithm and sublunar approximation. They compared Hadoop and Spark, two distributed systems used in data science and analytics. This research compares platform/algorithm combinations with sub-linear algorithms specific to machine learning, which is useful for machine learning but isnt applicable to all the other fields that utilize the regression. They also focused on parallelizing machine learning algorithms, while we are looking to parallelize logistic regression itself. We aim to parallelize the logistic regression in terms of how a scientist would use it, as well as how a data scientist would use it. Since data science is applicable to every field, this makes our work more accessible. They did use five open datasets though related to machine learning to make sure that they could accurately compare their methods. Our data is randomly generated and doesnt come from open source.

Deforestation modelling using logistic regression and GIS—- This paper uses logistic regression in the manner that we intend to use and parallelize it. The pros are that they analyze a sequential version of the logistic regression, however, they dont attempt to parallelize the logistic regression and their input data is in a much different format than ours. The R2 calculated in their paper also doesnt provide much information about the scatter of the data around the fitted line. They also 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 their research is very similar to ours, and is the paper that we decided to base parallelizing logistic regression on. However, their data input is far different than ours in that they wished to combine logistic regression with GIS. They analyze digital thematic-topographic maps with 63 different data layers such as forests, ranges and gardens. Our paper looks at a much simpler form of data, random file input values that fall within a certain range to represent our x and b values.

Breast Cancer Prediction by Logistic Regression with CUDA Parallel Programming Support—- In this paper, there is a focus on parallelizing logistic regression for the purposes of machine learning, bioinformatics and data analysis. While it is a machine learning approach and our research 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 versions of their software are compared in determining the loss function for the machine learning logistic regression.

B. Implementations from others that weve used or referenced

Evaluating Parallel Logistic Regression Models - liang, choi—- They implement a binary logistic regression as well, and we use the parallel algorithm approach but no distributed platform or sub-linear approximation. They also analyze their experimental results in accuracy, efficiency, scalability and robustness, as well as running time, which is what we look at in our analysis as well. Deforestation modelling using logistic regression and GIS—- From this work, we use a similar logistic regression and we also broke down our implementation first into linear regression, then logistic function and finally into logistic regression. They use R^2 to determine the fitness of the model, which is something we do too in order to make sure our randomly generated data makes sense.

IV. PROPOSED SOLUTION

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. We propose to parallelize the logistic regression in order to improve predictive behavior in a number of fields. This will allow researchers to analyze more data in less time. We are attempting a general parallelization of logistic regression, where our work can benefit all fields that utilize logistic regression. From our research, we havent found others who attempt to parallelize a general logistic regression. From our work we wish to show that logistic regression is parallelizable enough to offer increased data sets to analyze in lesser time than the sequential counterpart.

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. 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 and found a dependent Y 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 for calculating the inverse of a matrix.

C. Machine Setup

We used NYU linux CIMS accounts to run our code. We ssh'ed into cuda5 and cuda2 machines, which run cuda version 9 (?) and we loaded the cuda version 9 module. We used ./time to record the time taken to compute the logistic regression of our sequential code and CUDA code, and we used nvprof to further 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

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

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 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] G. O. Young, Synthetic structure of industrial plastics (Book style with paper title and editor), in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 1564.

[2] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123135.

[3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.

[4] B. Smith, An approach to graphs of linear forms (Unpublished work style), unpublished.

[5] E. H. Miller, A note on reflector arrays (Periodical styleAccepted for publication), IEEE Trans. Antennas Propagat., to be publised.

[6] J. Wang, Fundamentals of erbium-doped fiber amplifiers arrays (Periodical styleSubmitted for publication), IEEE J. Quantum Electron., submitted for publication.

[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style), IEEE Transl. J. Magn.Jpn., vol. 2, Aug. 1987, pp. 740741 [Dig. 9th Annu. Conf. Magnetics Japan, 1982, p. 301].

[9] M. Young, The Techincal Writers Handbook. Mill Valley, CA: University Science, 1989.

[10] J. U. Duncombe, Infrared navigationPart I: An assessment of feasibility (Periodical style), IEEE Trans. Electron Devices, vol. ED-11, pp. 3439, Jan. 1959.

[11] S. Chen, B. Mulgrew, and P. M. Grant, A clustering technique for digital communications channel equalization using radial basis function networks, IEEE Trans. Neural Networks, vol. 4, pp. 570578, July 1993.

[12] R. W. Lucky, Automatic equalization for digital communication, Bell Syst. Tech. J., vol. 44, no. 4, pp. 547588, Apr. 1965.

[13] S. P. Bingulac, On the compatibility of adaptive controllers (Published Conference Proceedings style), in Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory, New York, 1994, pp. 816.

[14] G. R. Faulhaber, Design of service systems with priority reservation, in Conf. Rec. 1995 IEEE Int. Conf. Communications, pp. 38.

[15] W. D. Doyle, Magnetization reversal in films with biaxial anisotropy, in 1987 Proc. INTERMAG Conf., pp. 2.2-12.2-6.

[16] G. W. Juette and L. E. Zeffanella, Radio noise currents n short sections on bundle conductors (Presented Conference Paper style), presented at the IEEE Summer power Meeting, Dallas, TX, June 2227, 1990, Paper 90 SM 690-0 PWRS.

[17] J. G. Kreifeldt, An analysis of surface-detected EMG as an amplitude-modulated noise, presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.

[18] J. Williams, Narrow-band analyzer (Thesis or Dissertation style), Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[19] N. Kawasaki, Parametric study of thermal and chemical nonequilibrium nozzle flow, M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

[20] J. P. Wilkinson, Nonlinear resonant circuit devices (Patent style), U.S. Patent 3 624 12, July 16, 1990.