Implement Parallelization on Linear Regression

PP-f19 Project Proposal

傅信瑀
0516319
資工系資工組 A
karta2155802.cs05@g2.nctu.edu.tw

尤健羽
0516306
資工系資電組
oilandy870103@gmail.com

許程翔
0516314
資工系資工組 B
tesseract324.cs05@nctu.edu.tw

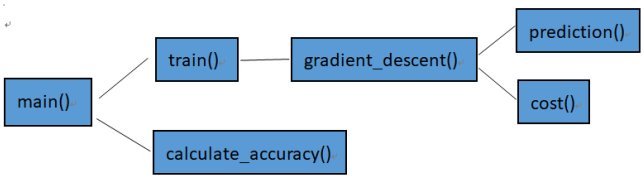
INTRODUCTION/MOTIVATION

Machine learning has become more and more popular in computer science nowadays. While computing power is increasing in nearly exponential rate, the requiring resources of those techniques still grow much faster than hardware does. The processing speed is very important if the data size is getting larger. In this case, optimizing the program to fully utilize the hardware become an important question, especially for ML, which requires large amount of computing resources to make it work.

STATEMENT OF THE PROBLEM

Last year, we used linear regression to handle data in Machine Learning class. And now, we want to accelerate our processing speed by applying parallelization. We will try to implement different parallelization methods including Pthread, OpenMP, MPI and other parallelization techniques that will be introduced in this course. The final results are expected to show significant difference between serial programs and parallel programs, and We will further compare different techniques of parallelization to choose the best one.

PROPOSED APPROACHES



LANGUAGE SELECTION

We will use C++ as our project language, with parallel language such as Pthread and OpenMP, since we are more familiar with these parallel model.

RELATED WORK

We did some research on the topic and found out that programmers tend to parallelize linear regression with R language. Since we have decided to use C++ to implement, there will be little information on the Internet and thus we can try to implement with what we learned in class.

STATEMENT OF EXPECTED RESULTS

The linear regression model using parallelization can deal with much more input data and generate more complicated regression model. However, with the implement of parallelization, the accuracy of our linear regression model may drop. We expect to construct the model that is scalable, minimizing the decrease of accuracy.

TIMETABLE

We plan to finish this project before January, starting from November .

1. One month for developing the algorithm in serial.
2. Three weeks for program parallelizing, using techniques such as Pthreads, OpenMP, MPI.
3. Two weeks for result observing and data analysing.

REFERENCES

[1] Linux Tutorial: POSIX Threads
http://man7.org/linux/man-pages/man7/pthreads.7.html

[2] Machine Learning: C++ Linear Regression Example
https://medium.com/@dr.sunhongyu/c-linear-regression-example-14243d6b9dc2