

CS546 “Parallel and Distributed Processing” Homework 4

Submission:

- *This is a programming assignment.*
 - *Due by 11:59pm of Nov. 22, 2015.*
 - *Late penalty: 20% penalty for each day late*
 - *Please upload your assignment on the Blackboard with the following name:*
CS546_SectionNumber_LastName_FirstName_HW4
 - *Please also upload your code on Jarvis under your home directory/hw4/.*
 - *Please do NOT email your assignment to the instructor and TA!*
-

In this assignment, you will design and code a CUDA-C/C++ version of a Matrix normalization algorithm. Reading the Wikipedia page on Standard Score is encouraged http://en.wikipedia.org/wiki/Standard_score. In Data Analytics and Machine Learning, matrices usually contain data points (rows) in a particular space defined by their attributes (columns). Sometimes, and because attributes may represent things that are very different in nature, normalization by column is required.

Your goal is to take advantage of the power of GPUs to perform this task as fast as possible. Realize that column normalization is composed of three steps per column:

1. Calculating the mean of the column
2. Calculating the standard deviation (which requires the mean)
3. Finally, calculating the normalized value by performing the following calculation (where B is the normalized matrix of A)

$$B[\text{row}][\text{col}] = (A[\text{row}][\text{col}] - \text{mean}) / \text{standard_deviation}$$

It is possible to see that the first two steps can be achieved with a REDUCTION algorithm (recall that we saw the SUM-reduction algorithm in class). In this part, you need to be very careful with your design decisions. Where you put the data, and how you perform the reductions matter. As a hint, you may consider splitting the reductions in two: first, inside the values in each block and, second, reducing the totals for every block. Once the mean and standard deviation are calculated, the third step is straightforward.

The sequential code is provided with this assignment, and can be used as a reference for debugging and performance comparison. **Your code will be graded partially on the efficiency of your algorithm.** You should write a memo explaining your design decisions very clearly and your performance evaluation. Even if your code does not work, or is not efficient enough, you should write the reasons you think that is so.