Parallel Principle Component Analysis using Covariance Matrix and SVD in CUDA

Outlines:

1. Aim

To reduce the dimensionality of the input data such that we don't lose valuable information and also reduce the computation overhead. This would be achieved using standard PCA algorithm on top on which we apply parallelism (as explained in the sections below).

2. SVD

The main objective in the PCA is to calculate the eigen vector and eigen values of the covariance matrix (of the input data). This vector matrix is then used to project the original data with higher dimension to a lower dimension data.
We will decompose our covariance matrix using the svd as implemented in cuSolver library.
https://docs.nvidia.com/cuda/cusolver/index.html (https://docs.nvidia.com/cuda/cusolver/index.html)

3. Analyse the optimal dimension

After we have calculated eigen values and eigen vectors, we aim to find the best reduced dimension which can represent our data without losing useful information. This is done by using a threshold value and using the inequality as below.

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} > \theta$$

Where k is the reduced number of dimensions and n is the total number of dimensions and theta is a threshold value. Note that the numerator picks up the eigen values one by one in descending order.

4. Projected data

The projected data is constructed by using the k eigen vectors (eigen vectors corresponding to the eigen values used in the inequality above).

$$ProjectedData = XV[:, : k]$$

Here X is the original data and V is the eigen vector matrix.

5. Comparing performance

Applying some Machine Learning algorithms (Classification, Clustering, etc.) and comparing the results between Projected Data and Original Data. This will give us a measure of how good we can perform even with reduced dimensions and in turn saving much precious computation time.

6. Parallelized Components

1. Calculating Covariance Matrix.(Matrix-Matrix multiplication)
2. SVD (cuSolver)
3. Projected Data construction (Matrix-Matrix multiplication)