INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Introduction to Parallel Scientific Computing (CSE-504)

Dr. Pawan Kumar

Project Report

Parallel PCA using Covariance Matrix and SVD in CUDA

Aditya Tewari (2018201082)
Akshansh Sharma (2018201081)

1 Overview

1.1 PCA

Dimensionality reduction is one of the pre-processing steps in many machine learning applications and it is used to transform the features into a lower dimension space.Principal Component Analysis(PCA) technique is one of the most famous unsupervised dimensionality reduction techniques. The goal of the PCA is to find the space, which represents the direction of the maximum variance of the given data. It is not specific to any particular ML/AI algorithm.

PCA technique has many goals including finding relationships between observations,extracting the most important information from the data, outlier detection and removal,and reducing the dimension of the data by keeping only the important information. All these goals are achieved by finding the PCA space, which represents the direction of the maximum variance of the given data. The PCA space consists of orthogonal principal components, i.e. axes or vectors. The principal components (PCs) are calculated by solving the co-variance matrix or using Singular Value Decomposition(SVD).

1.2 Implementation Insights

In our project we have written a parallel code for computing PCA and test it by reducing the dimension of Intrusion-Detection Dataset. We test it by using Clustering algorithms (k-Means, Hierarchical ,GMM) and test the performance of the algorithm for both reduced and original dataset.

2 Algorithm :

1. Compute the mean feature vector

   $\mu = \frac{1}{p}\sum_{k=1}^{p} x_k$, where, x_k is a pattern ($k = 1$ to p), p = number of patterns, x is the feature matrix
2. Find the covariance matrix

   $C = \frac{1}{p}\sum_{k=1}^{p}\{x_k - \mu\}\{x_k - \mu\}^T$ where, T represents matrix transposition
3. Compute Eigen values λ_i and Eigen vectors v_i of covariance matrix

   $Cv_i = \lambda_i v_i$ $(i = 1, 2, 3,....q)$, q = number of features
4. Estimating high-valued Eigen vectors
   (i) Arrange all the Eigen values (λ_i) in descending order
   (ii) Choose a threshold value, θ
   (iii) Number of high-valued λ_i can be chosen so as to satisfy the relationship

   $\left(\sum_{i=1}^{s}\lambda_i\right)\left(\sum_{i=1}^{q}\lambda_i\right)^{-1} \geq \theta$, where, s = number of high valued λ_i chosen

   (iv) Select Eigen vectors corresponding to selected high valued λ_i
5. Extract low dimensional feature vectors (principal components) from raw feature matrix.

   $P = V^T x$, where, V is the matrix of principal components and x is the feature matrix

Figure 1: PCA Algorithm

The algorithm is adopted form this paper (1). Threshold value (θ) of 0.90 was used in our algorithm. We wrote kernels to compute the following part of our code parallely :

- Calculating the mean of a data column.
- Calculating std of the data column.
- Normalizing the dataset.
- Getiing the transpose of matrix (used for the next step)
- Calculating the Covariance matrix.
- cuSolver(2) library was used to calculate the SVD of the covariance matrix.

After these, with the reduced dimensions dataset was written onto a new file "reduced_data.csv" for further testing.

3 Results

The parallel code was tested on Google Cloud (GPU).

- Serial code Running time : 1.2s
- Parallel code Running time : 2.2s

As discussed during the evaluation also, the parallel code can be further optimized by using more number of blocks instead of using more threads for our application. Also redundant Gpu-to-Host and Host-to-Gpu copies can help as they are expensive operations. Further optimizing by these techniques can help us attaining a very high speedup.

The testing code is included in the repository in a .ipynb format which can be seen to analyze that the reduced dimension help us to attain nearly same accuracy in shorter time. The details of the dataset, the algorithms are also mentioned in the 'ipynb notebook'.

References

[1] Gs, Vijay Pai, P Sriram, N. S. BKN Rao, Raj. (2013). Radial basis function neural network based comparison of dimensionality reduction techniques for effective bearing diagnostics. Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology. 227. 640-653. 10.1177/1350650112464927. Paper

[2] cuSolver CUDA

[3] PCA and applications (as discussed in class) link

[4] PCA Introduction and in-depth Explanation