SDA Report Group - 24

Sayam Kumar
Raahul Singh
Hrishabh Pandey
Ram Nad

Abstract

The emergence of programmable graphics hardware has led to increasing interest in offloading numerically intensive computations to GPUs.This has made GPUs very prevalent in this era of high performance computing, but GPU programming remains challenging and it would therefore be handy to have a model which captures the main performance factors of GPU kernels that helps to estimate the run time, which could also potentially be used to identify its bottlenecks.

In this analysis, we analyse the SGEMM GPU kernel performance. This data set measures the running time of a matrix-matrix product A*B = C, where all matrices have size 2048 x 2048, using a parameterizable SGEMM GPU kernel with 261400 possible parameter combinations.

Using a randomly sampled set from the population, we test if any relationship exists between the different parameters. We further cluster the samples using popular clustering algorithms like KMeans. For Dimensionality Reduction, we use PCA.

Introduction

In this analysis, the dataset used has the following description.

1. Relevant Information about the dataset: This data set measures the running time of a matrix-matrix product A*B = C, where all matrices have size 2048 x 2048, using a parameterizable SGEMM GPU kernel with 261400 possible parameter combinations. For each tested combination, 4 runs were performed and their results are reported as the 4 last columns. All times are measured in milliseconds.

There are 14 parameters, the first 10 are ordinal and can only take up to 4 different powers of two values, and the 4 last variables are binary. Out of 1327104 total parameter combinations, only 261400 are feasible (due to various kernel constraints). This data set contains the results for all these feasible combinations. The experiment was run on a desktop workstation running Ubuntu 16.04 Linux with an Intel Core i5 (3.5GHz), 16GB RAM, and a NVidia Geforce GTX 680 4GB GF580 GTX-1.5GB GPU. We use the "gemm_fast" kernel from the automatic OpenCL kernel tuning library "CLTune" (https://github.com/CNugteren/CLTune).

2. Number of Instances: 241600

3. Number of Attributes: 18 (14 predictive attributes, 4 goal fields)

4. Attribute Information:

    Independent variables:

    1-2. MWG, NWG: per-matrix 2D tiling at workgroup level: {16, 32, 64, 128}

    3. KWG: inner dimension of 2D tiling at workgroup level: {16, 32}

    4-5. MDIMC, NDIMC: local workgroup size: {8, 16, 32}

6-7. MDIMA, NDIMB: local memory shape: {8, 16, 32}
8. KWI: kernel loop unrolling factor: {2, 8} (integer)
9-10. VWM, VWN: per-matrix vector widths for loading and storing: {1, 2, 4, 8}
11-12. STRM, STRN: enable stride for accessing off-chip memory within a single thread: {0, 1}
13-14. SA, SB: per-matrix manual caching of the 2D workgroup tile: {0, 1}

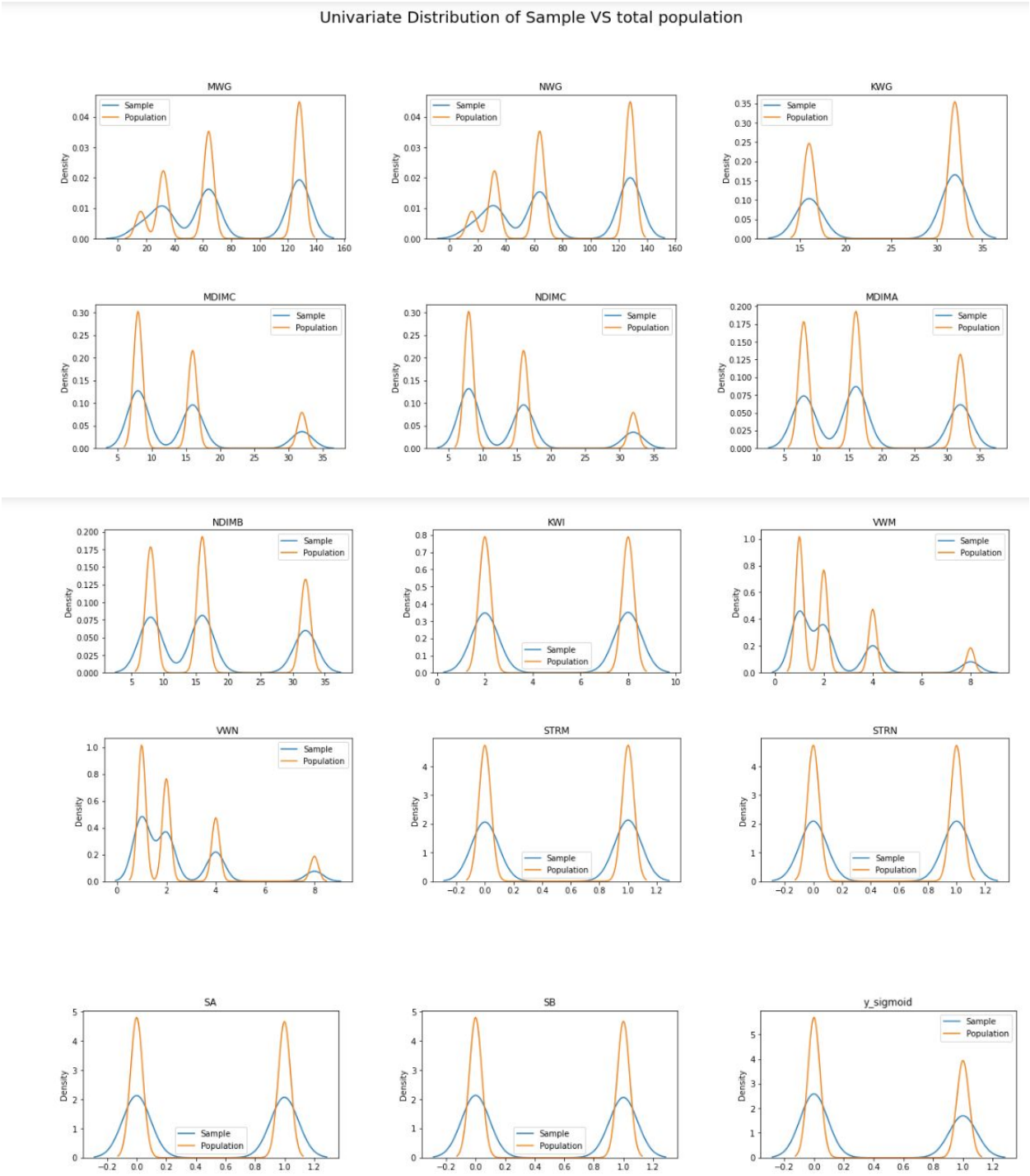
Output - 15-18. Run1, Run2, Run3, Run4: performance times in milliseconds for 4 independent runs using the same parameters. They range between 13.25 and 3397.08.

Methodology:

1. We take the mean on the four runtimes to get an average run.
   Based on the information from the official data description, we further convert this to the logarithmic scale.

2. To Remove Outliers: Z Test
   A z-score measures exactly how many standard deviations above or below the mean a data point is. ... A negative z-score says the data point is below average. A z-score close to 0 says the data point is close to average. A data point can be considered unusual(an outlier) if its z-score is above 3 or below –3.

3. Randomly Sampling data from the total population.



4. Chi-Square Test to test if any relationship exists between the parameters:

   The Chi-Square test of independence is used to determine if there is a significant relationship between two nominal (categorical) variables. The frequency of each category for one nominal variable is compared across the categories of the second nominal variable. The data can be displayed in a contingency table where each row represents a category for one variable and each column represents a category for the other variable. For example, say a researcher wants to examine the relationship between gender (male vs. female) and empathy (high vs. low). The chi-square test of independence can be used to examine this relationship. The null hypothesis for this test is that there is no relationship between gender and empathy. The alternative hypothesis is that there is a relationship between gender and empathy (e.g. there are more high-empathy females than high-empathy males).

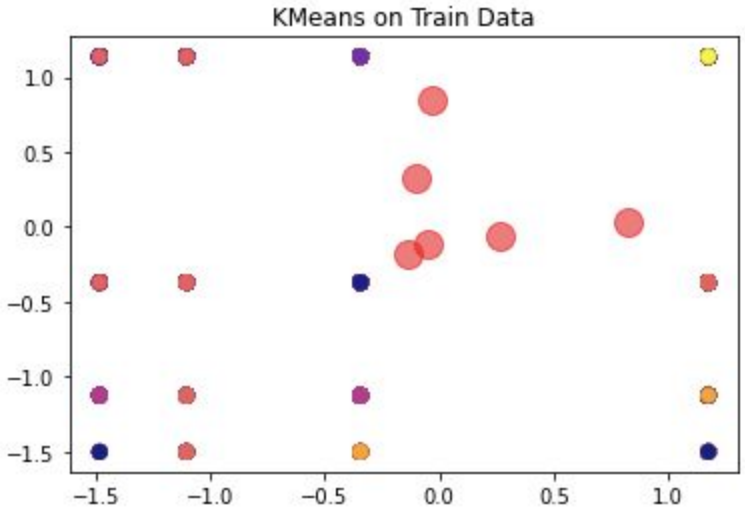
Inferential Analysis -

KMeans Clustering:

The algorithm will categorize the items into k groups of similarity. To calculate that similarity, we will use the euclidean distance as measurement. The algorithm works as follows:

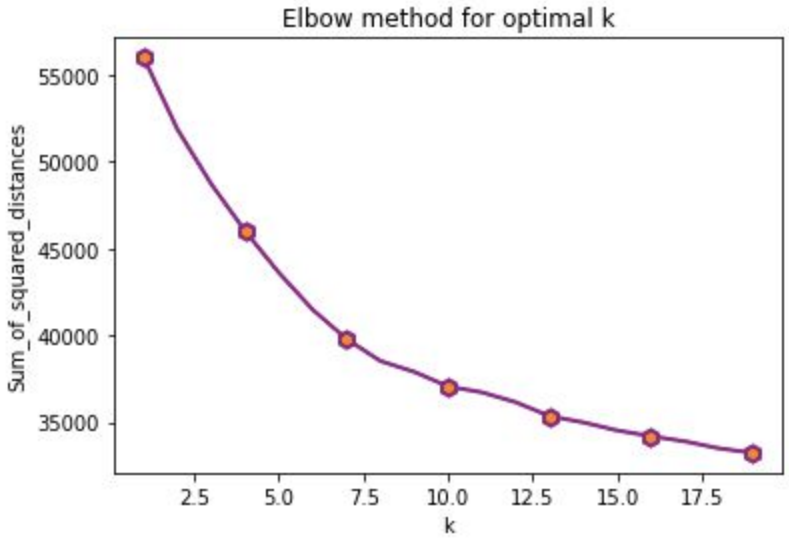
1. First we initialize k points, called means, randomly.
2. We categorize each item to its closest mean and we update the mean's coordinates, which are the averages of the items categorized in that mean so far.
3. We repeat the process for a given number of iterations and at the end, we have our clusters.

The “points” mentioned above are called means, because they hold the mean values of the items categorized in it. To initialize these means, we have a lot of options. An intuitive method is to initialize the means at random items in the data set. Another method is to initialize the means at random values between the boundaries of the data set (if for a feature *x* the items have values in [0,3], we will initialize the means with values for *x* at [0,3]).



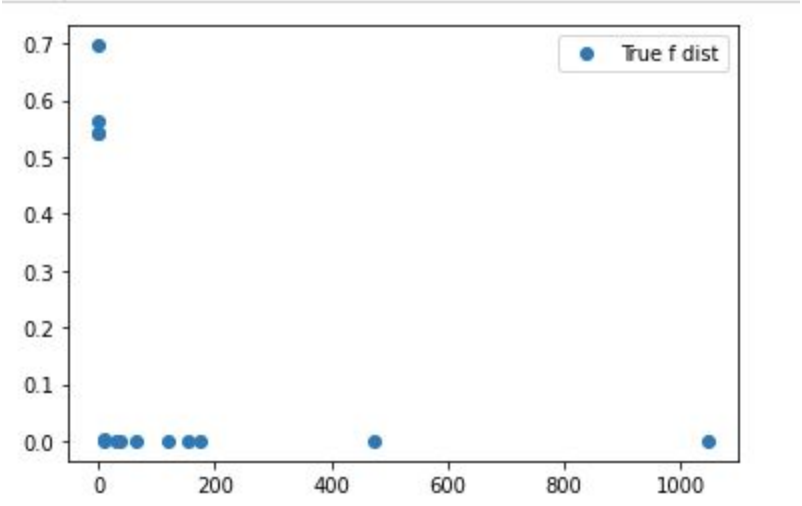
Elbow Method for optimal value of k in KMeans:

The “elbow” method is used to select the optimal number of clusters by fitting the model with a range of values for K. If the line chart resembles an arm, then the “elbow” (the point of inflection on the curve) is a good indication that the underlying model fits best at that point.



F Test:

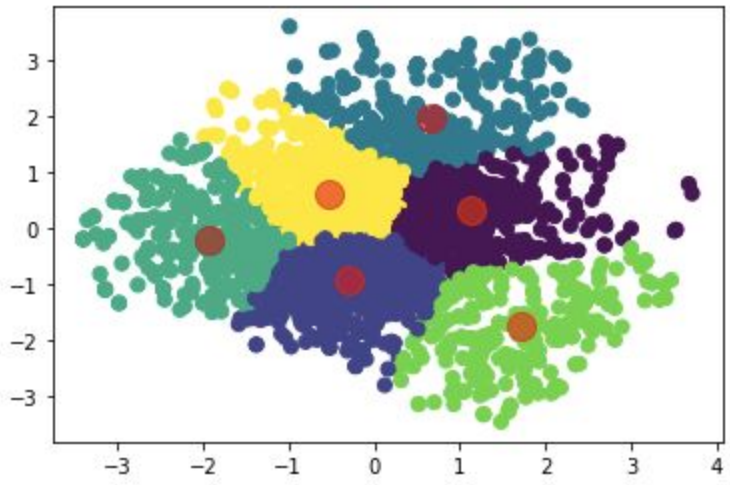
An F-test is any statistical test in which the test statistic has an F-distribution under the null hypothesis. It is most often used when comparing statistical models that have been fitted to a data set, in order to identify the model that best fits the population from which the data were sampled.



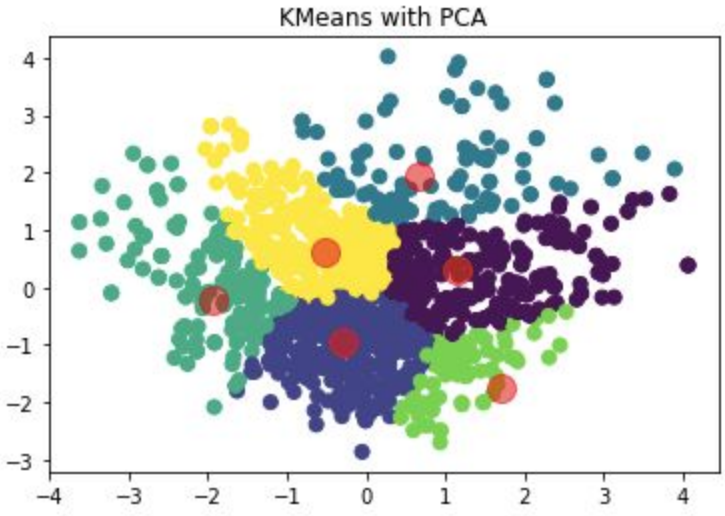
PCA:

Principal component analysis (PCA) is the process of computing the principal components and using them to perform a change of basis on the data

K Means on training set after PCA



K Means on test set after PCA



Conclusion:

From this analysis, we see that the distribution of a randomly sampled set from the total population, closely approximates the distribution of the population.

Using Several statistical tests, we conclude the parameters mostly do not have any inter relationship.

Further, Using dimensionality reduction improves the accuracy of clustering the dataset.

Finally, Logistic Regression outperforms the Clustering Methods in terms of accuracy.

```
In [122]: 1 fit = LogisticRegression().fit(X_train, y_train.values)
          2 print(f"Accuracy of Logistic regression on train set {fit.score(X_train, y_train.values)}")

          Accuracy of Logistic regression on train set 0.89

In [123]: 1 print(f"Accuracy of Logistic regression on train set {fit.score(X_test, y_test.values)}")

          Accuracy of Logistic regression on train set 0.9183333333333333
```

References:

1. UCL Machine Learning Repository

2. On Chi Squared Test
3. On K Means Clustering
4. Elbow Method for Kmeans
5. On PCA
6. On F test
7. Enrique G. Paredes (eg paredes '@' ifi.uzh.ch). Visualization and MultiMedia Lab, Department of Informatics, University of Zurich. Zurich, 8050. Switzerland
8. Rafael Ballester-Ripoll, Enrique G. Paredes, Renato Pajarola.
   Sobol Tensor Trains for Global Sensitivity Analysis.
   In arXiv Computer Science / Numerical Analysis e-prints, 2017 ([Web Link]).
9. Cedric Nugteren and Valeriu Codreanu.
   CLTune: A Generic Auto-Tuner for OpenCL Kernels.
   In: MCSoC: 9th International Symposium on Embedded Multicore/Many-core Systems-on-Chip. IEEE, 2015 ([Web Link])

Thank You!!