Performance Benchmark Report for estOLS

Summary

In this GitHub repository, python codes for calculating ordinary least square via matrix operation are provided. In the implementation, numpy and cupy python packages are used for CPU and GPU implementation, respectively. This report will be focused on the computational performance of the current implementations.

The computational performance (cost) for different implementations is shown in Figure 1.

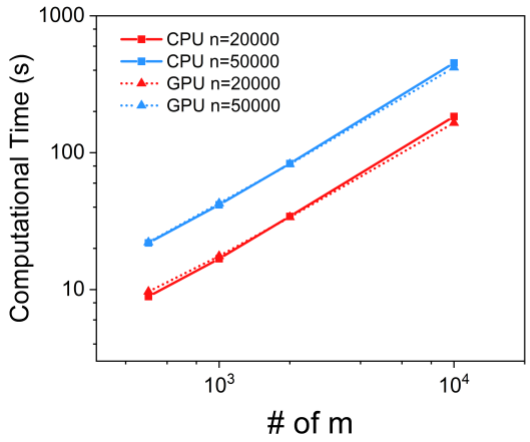

Figure 1. Computational time verse the number of columns for different implementations at different number of rows.

Figure 1 shows the computational cost versus the number of columns for both CPU and GPU implementations. For both the GPU and CPU implementations, the computational cost increases exponentially with the increase of number of columns. The difference of computational cost between different implementations is very small because the computational bottleneck is the file I/O. But it is still clear that from Figure 1, when the size of matrix is relatively small, the CPU implementation is faster than GPU implementation because of the memory copy overhead in GPU implementation. When the size of matrix is relatively large, the computational performance

of GPU implementation is better because the computational speedup for the matrix operation via GPU-acceleration is more significant than memory copy overhead.

Future Improvement

As discussed above, the current computational bottleneck is at the file I/O. In the future version, parallel file I/O shall be considered in order to further improve the computational performance. In addition, it would also be beneficial if this computational routine could be integrated with the other part of computational which would significantly reduce the computational cost involved in file I/O. If the current implementation is going to be integrated into other python routines, it will also be convenient for users with different environments to be able to install the package via conda.