

Performance Benchmark Report for estOLS

Summary

In this GitHub repository, source codes of calculating ordinary least squares via python packages are provided. While the CPU version mainly uses python module – numpy, the GPU-accelerated version uses the python module – cupy. And the matrix operation approach is used to calculate the ordinary least squares. The computational performance (cost) for different implementations is shown in Figure 1.

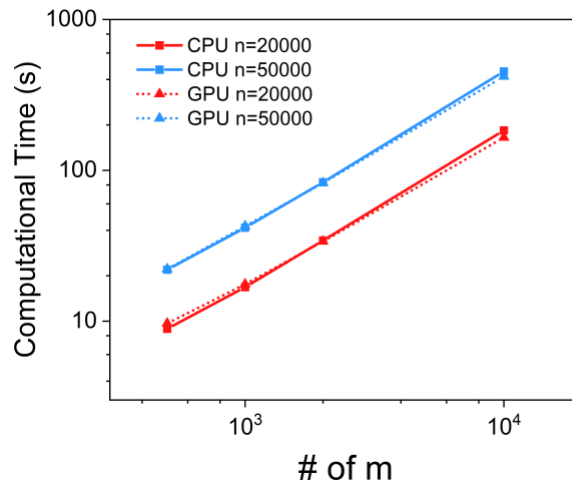


Figure 1. Computational time verse the number of rows for different implementations at different number of columns.

As shown in Figure 1, the difference of computational cost between different implementations is very small because the computational bottleneck is at the stage of 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 overhead for memory copy 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.