GPU parallelization of massive sample-size survival analysis

Jianxiao Yang, Marc A. Suchard

Background

The accessibility of observational health data provides rich opportunities to study comparative effectiveness and safety of medical products, but also poses unprecedented challenges. Typical administrative claims and electronic health record (EHR) databases now contain millions of individuals[1] with thousands of patient characteristics and up to 10 years of data per life[2]. The massive scales of these databases offer more power for statistical analyses but also bring taxing computational burden.

This big-data problem is further exacerbated by the increasing complexity of common statistical models. For instance, the Cox proportional-hazards model and Fine-Gray model are widely applied in comparative effectiveness and safety studies. The complexity of the likelihood evaluations of Cox model and Fine-Gray model grows quadratically with sample size. In addition, some form of regularization[3] is often needed to achieve parsimonious model selection, which typically requires computationally intensive cross-validation. This further strains our limited computational resources.

To address the issue of computational burden, we leverage GPU parallelization to the Cox model and Fine-Gray model through cyclic coordinate descent for accelerating survival analysis utilizing big health data.

Methods

Maximizing the partial likelihood under the Cox model and Fine-Gray model requires the calculation of the score function and the Hessian diagonals. This step consumes over 95% of the run-time in the CPU implementation in our Cyclops package. Careful benchmarks reveal that the most costly calculation is updating the series of prefix sums introduced through the growing risk sets of survival data and the followed reduction, which are amenable to parallelization.

We apply the implementation of prefix sums and reduction from the CUB library for the benefit of the most cutting-edge decoupled look-back strategy[4]. This work-efficient, communication-avoiding, single-pass method for scan requires the optimal ~2n data movements: n reads and n writes to the global memory.

Since data movement is often the main bottleneck in modern computer systems[5], we fuse prefix sums and reductions in our likelihood calculations into a single kernel to avoid unnecessary memory transactions. To exploit the sparsity of the design matrix $\boldsymbol{X}$, we implement a sparse CUDA kernel, which only reads in and processes the non-zero entries while keeping other entries as zeros all the time, for saving the data movement as well as reducing memory bandwidth requirements on GPU significantly.

As we use cross-validation to search for the optimum tuning parameters in regularization, we further improve the efficiency of cross-validation by adding another layer of parallelization with multi-stream GPU.

Results

We examine the performance of CPU vs GPU in simulated data of varying sizes. We also reproduce a real-world study using our GPU implementation. To produce CPU results, we use an Intel(R) Xeon(R) W-2155 CPU that runs at 3.3GHz and has 10 cores, achieving 574.5 Gflops double-precision point performance. For the GPU results, we use an NVIDIA Quadro GV100 with 5120 CUDA cores and 32GB HBM2 Memory, achieving 7.4 Tflops double-precision point performance.

In our simulation experiments, GPU parallelization generates up to a 38-fold speedup for Cox model and a 42-fold speedup for Fine-Gray model with a fixed L1 penalty. We then test our multi-stream L1 regularized Cox regression on simulated data with a 10-fold cross-validation with 10 repetitions. Then the 100 cross-validation replicates are distributed to s CUDA streams allocated by s CPU threads (s = 1, 2, 4, 8). Although our multi-stream implementation does not achieve the optimal s× speedups with s streams, it still reduces the runtimes remarkably, especially on the larger simulated data.

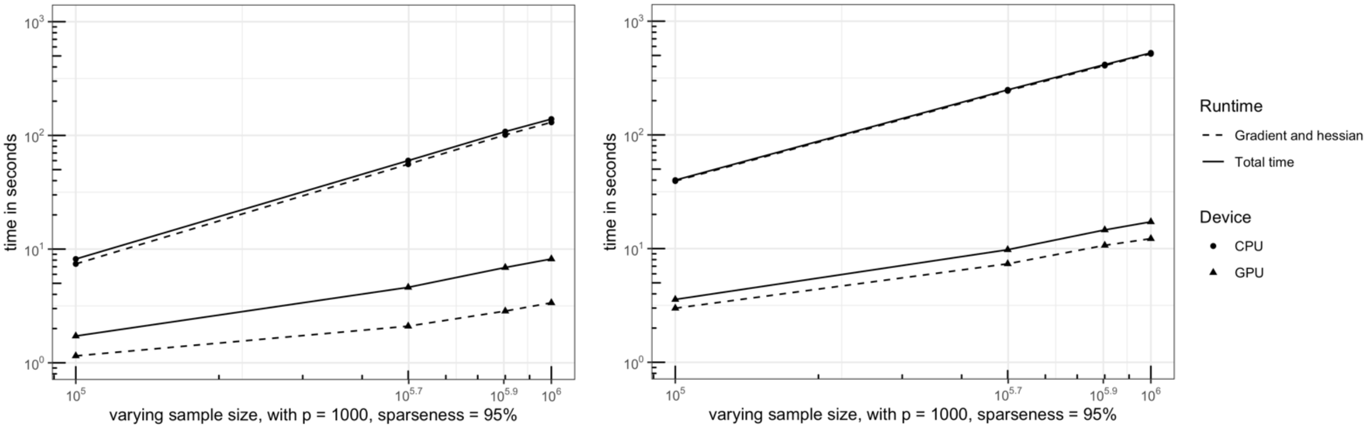

Figure 1. GPU vs CPU runtimes for Cox model (left) and Fine-Gray model (right).

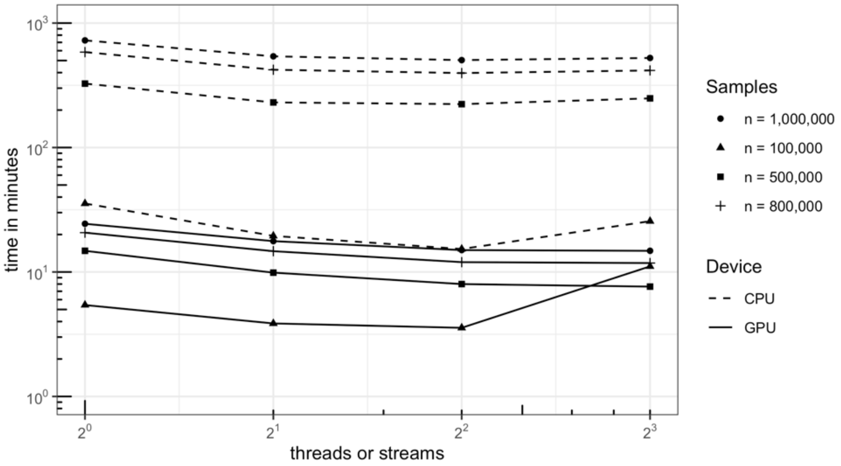

Figure 2. Runtimes of multi-stream (threads) cross-validated Cox model on GPU (CPU).

We examine patients initiating angiotensin-converting enzyme inhibitors (ACEi) and thiazide or thiazide-like diuretics (THZ), where the outcome is any major cardiovascular event (acute myocardial infarction, hospitalization for heart failure, or stroke) from IBM MarketScan Commercial Claims and Encounters (CCAE) database follow the design of LEGEND-HTN study[6]. A total of 1,065,745 patients were included in our study, 71.7% of whom initiated an ACEi and 28.3% of whom initiated a THZ. We consider the main treatment covariate, in addition to 7891 patient characteristic covariates. Without performing propensity score matching or stratification, we simply run a Cox regression with L1 regularization on all baseline covariates. Our GPU parallelization reduce the time of parameter estimation from 16 hours on multi-core CPU to less than one hour.

Conclusion

We implement the massive parallelization of the Cox proportional hazards model and Fine-Gray model by using NVIDIA’s CUB library for parallel computing of prefix sums and exploiting the sparsity of data. By saving data movement and clever manipulation of likelihood structure, our parallelization significantly reduces the runtime of large-scale survival analysis.

References/Citations

1. G. Hripcsak, P. B. Ryan, J. D. Duke, N. H. Shah, R. W. Park, V. Huser, M. A. Suchard, M. J. Schuemie, F. J. DeFalco, A. Perotte, et al., "Characterizing treatment pathways at scale using the ohdsi network," Proceedings of the National Academy of Sciences, vol. 113, no. 27, pp. 7329-7336, 2016.
2. M. A. Suchard, S. E. Simpson, I. Zorych, P. Ryan, and D. Madigan, "Massive parallelization of serial inference algorithms for a complex generalized linear model," ACM Transactions on Modeling and Computer Simulation (TOMACS), vol. 23, no. 1, p. 10, 2013.
3. D. Madigan, P. Ryan, S. Simpson, and I. Zorych, "Bayesian methods in pharmacovigilance," Bayesian Statistics, vol. 9, pp. 421-438, 2010.
4. D. Merrill and M. Garland, "Single-pass parallel prefix scan with decoupled look-back," NVIDIA, Tech. Rep. NVR-2016-002, 2016.
5. A. J. Holbrook, P. Lemey, G. Baele, S. Dellicour, D. Brockmann, A. Rambaut, and M. A. Suchard, "Massive parallelization boosts big bayesian multidimensional scaling," Journal of Computational and Graphical Statistics, no. just-accepted, pp. 1-34, 2020.
6. M. A. Suchard, M. J. Schuemie, H. M. Krumholz, S. C. You, R. Chen, N. Pratt, C. G. Reich, J. Duke, D. Madigan, G. Hripcsak, et al., "Comprehensive comparative effectiveness and safety of first-line antihypertensive drug classes: a systematic, multinational, large-scale analysis," The Lancet, vol. 394, no. 10211, pp. 1816-1826, 2019.