Title	GPU Accelerated Adaptive Random Forest
Author(s)	渡邉，純司
Citation	
Issue Date	2023-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/18364
Rights	
Description	Supervisor: 井口 寧，先端科学技術研究科，修士(情報科学)

GPU Accelerated Adaptive Random Forest

Junji Watanabe (Inoguchi Lab.)

[Background and purpose] In the era of Big Data, due to the rapid growth of mobile devices, IoT devices etc., the amount of data being generated every moment is drastically increasing. However, learning from a high-volume and continuous data stream is extremely difficult. Hoeffding Tree [1] is a decision tree algorithm that can learn from data stream. Adaptive Random Forest (ARF) [2] utilizes Random Forest technique based on Hoeffding Trees. Multiple CPU cores can speed up the ARF algorithm by training trees in parallel. However, further speed up is desired to apply ARF for high volume data stream learning problems. GPU RF [3] also parallelized Hoeffding Tree based Random Forest. However, it had the limitation on the max tree depth, which is a problem when applying GPU to complex problems. This thesis aims to propose GPU ARF, a GPU implementation of ARF with the following objectives: 1. Speed up ARF with GPU. 2. Resolve the max tree depth limitation that GPU RF had. 3. Keep the accuracy as much as possible.

[Experimental results] With the proposed dynamic tree node memory allocation, the max tree depth limitation was completely resolved. GPU ARF shows better scalability to the number of trees than CPU ARF, and GPU ARF is faster when the number of trees is larger than the number of CPU cores. Figure 1 shows the accuracy and the execution time of GPU ARF and CPU ARF. For the both of LED dataset and Covertype dataset, GPU ARF is 7.3 times faster and 3.8 times faster respectively, while keeping the accuracy at the same levels with CPU ARF.

[Conclusions] GPU ARF resolved the max tree depth limitation by allocating memory dynamically from memory pools. This allows GPU ARF to be applied to complex problems. GPU ARF shows 7.3 times and 3.8 times speed up for data stream classification problem with LED dataset and Covertype dataset in comparison to CPU ARF, while keeping the accuracy. There are some cases where GPU ARF doesn't get benefit from the parallelism. If the problem is simple enough and the required number of trees is less than the number of CPU cores, CPU ARF would work better in terms of both of speed and accuracy. Also, if the data stream has concept drifts at fast pace, batching data instances would cause a regression on the accuracy. Applying GPU ARF for numeric attributes and regression problems are the future works to further examine the performance gain of GPU ARF in various settings.

[Reference] [1] P. Domingos and G. Hulten. Mining high-speed data streams. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ' 00, page 71–80, New York, NY, USA, 2000. Association for Computing Machinery. [2] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, and T. Abdessalem. Adaptive random forests for evolving data stream classification. *Machine Learning*, 106(9-10):1469–1495, jun 2017. [3] D. Marron, A. Bifet, and G. D. F. Morales. Random forests of very fast decision trees on gpu for mining evolving big data streams. In *Proceedings of the Twenty-First European Conference on Artificial Intelligence*, ECAI ' 14, page 615–620, NLD, 2014. IOS Press.

Dataset	Implementation	Accuracy (%)	Execution time (secs)
LED	CPU ARF (C++)	73.97	127.70
LED	GPU ARF	72.31	17.59
Covertype	CPU ARF (C++)	72.85	61.70
Covertype	GPU ARF	72.84	16.36

Figure 1: Performance comparisons between GPU ARF and CPU ARF

[Keywords] Data stream learning, GPU acceleration