Efficient Multi-Class Probabilistic SVMs on GPUs
(Extended Abstract)

Zeyi Wen[†1], Jiashuai Shi[†‡2], Bingsheng He[†3], Jian Chen[‡4], Yawen Chen[‡5]
[†]*National University of Singapore,* [‡]*South China University of Technology*
{[1]wenzy, [3]hebs}@comp.nus.edu.sg, [4]ellachen@scut.edu.cn
[2,5]{shijiashuai,ywchenscut}@gmail.com

***Abstract*—Multi-class SVMs with the probabilistic output (MP-SVMs) are important techniques in pattern recognition. Two key challenges for efficient GPU accelerations for MP-SVM are: (i) many kernel values are repeatedly computed as a binary SVM classifier is trained iteratively, resulting in repeated accesses to the high latency GPU memory; (ii) performing training or estimating probability in parallel requires a much larger memory footprint than the GPU memory. To overcome the challenges, we propose *GMP-SVM* to reduce high latency memory accesses and memory consumption through batch processing, computation/data reusing and sharing. Experimental results show that our solution (available in https://github.com/Xtra-Computing/thundersvm) outperforms LibSVM by 100 times while retaining the same accuracy.**

1. Introduction

The development of more effective and more efficient algorithms are essential to the success of machine learning. Recently, many researchers are improving traditional machine learning algorithms using the high-performance hardware [1]. In this work, we propose a novel and efficient solution to multi-class SVMs with probabilistic output (MP-SVMs) using GPUs. A key barrier that hinders the wide adoption of MP-SVMs is its high training and probability estimation cost, since an MP-SVM classifier consists of many binary SVMs. Those costs can be prohibitively high for ever increasingly large datasets. GPUs are potentially excellent hardware to accelerate MP-SVMs. A naive approach of using GPUs is to train the binary SVMs on the GPU one by one, and to estimate probability for multiple instances using one binary SVM at a time. This naive approach has a relatively small memory footprint which fits into the GPU. However, the naive approach severely underutilizes the GPU. Two key challenges are: (i) many kernel values are repeatedly computed as a binary SVM classifier is trained iteratively, resulting in repeated accesses to the high latency GPU memory; (ii) performing training or estimating probability in parallel requires a much larger memory footprint than the GPU memory.

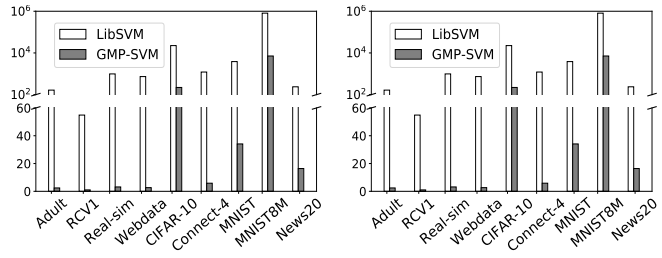
To address the challenges, we propose an efficient parallel solution called "*GMP-SVM*" which exploits two-level optimization for training MP-SVMs and high parallelism for estimating probability. GMP-SVM reduces high latency memory accesses and memory consumption through batch processing, kernel value reusing and sharing, and support vector sharing. In the binary SVM level, we compute kernel values in batches with the consideration of reusing kernel values via a GPU buffer. In the MP-SVM level, we develop techniques to concurrently train multiple binary SVMs with kernel value sharing among the binary SVMs. When estimating probability, GMP-SVM concurrently computes the probabilities for multiple instances using multiple binary SVMs with support vector and kernel value sharing among the SVMs. GMP-SVM is integrated into ThunderSVM [2] at https://github.com/Xtra-Computing/thundersvm. Experiments show that GMP-SVM outperforms LibSVM by 100 times while retaining the same accuracy.

2. Background

The goal of the SVM training is to find a hyperplane that separates the positive and negative instances with the maximum margin and meanwhile, with the minimum misclassification error. We use Sequential Minimal Optimization (SMO) for training the SVM. SMO is simply a straightforward subspace-ascent algorithm restricted to two-dimensional subspaces [3], where subproblems are solved optimally. To convert the decision value from a binary SVM to a probability, we use a common approach based on the sigmoid function. The sigmoid is fitted to the SVM output, which ideally is a monotonic function of the probability. The intuition is that given $v_i > v_{i'} \geq 0$, the probability of $\boldsymbol{x}_i$ being a positive instance should be greater than that of $\boldsymbol{x}_{i'}$. During training MP-SVMs of k classes, $k(k-1)/2$ binary SVMs are trained using SMO and then their predicted values on the training instances are used to train the sigmoid function discussed above. When estimating probability, MP-SVMs first estimate local probability using binary SVMs with probability output and then estimate multi-class probability using all the local probabilities by solving a convex quadratic problem with a linear equality constraint [4].

3. Our solution

To address the challenges of developing efficient multi-class SVMs with probabilistic output (*MP-SVMs*) algo-



(a) training time (sec) (b) prediction time (sec)

Figure 1: Training and prediction efficiency

rithms, we develop a novel solution called ***GMP-SVM*** ("G" stands for "GPU") with two-level optimization for training MP-SVMs and high parallelism for estimating probability. GMP-SVM reduces high latency memory accesses and memory consumption through batch processing, kernel value reusing and sharing, and support vector sharing.

In the binary SVM level, we use a larger working set of violating instances than the SMO solver does (i.e., SMO selects two instances), such that we can amortize the data access overhead on GPU memory and solve SMO subproblems in a batch. We (i) precompute all the kernel values for the violating instances in a batch to achieve cheaper cost per kernel value, (ii) store the kernel values to a GPU buffer which stores all the kernel values in the working set, and (iii) optimize the SVM on the working set with an approach to reduce the negative effect of local optimization on the working set. In the MP-SVM level, we concurrently train multiple binary SVMs with kernel value sharing among the binary SVMs. When estimating probabilities, GMP-SVM concurrently computes the probabilities for multiple instances using multiple binary SVMs with support vector and kernel value sharing. With the two level optimization for training and support vector and kernel value sharing for estimating probability, we address the memory access and GPU utilization problems of the GPU baseline.

Concurrently estimating probabilities for multiple instances using multiple binary SVMs requires more memory than the GPU memory footprint. To enable decision value prediction using multiple binary SVMs, we propose to share support vectors between SVMs and to share kernel values while computing the decision values. The reason behind support vector sharing is that the training datasets of two binary SVMs may have more than a half of the training instances in common, and some common training instances may become support vectors which can be shared between the SVMs. Without support vector sharing, the same training instance may be stored in $(k-1)$ binary SVMs as a support vector. Our support vector sharing technique reduces the GPU memory consumption by up to a factor of $(k-1)$.

4. Experiments

Here, we evaluate the performance of GMP-SVM in ThunderSVM. We conducted all of our experiments on a workstation running Linux with two Xeon E5-2640v4 CPUs, 256GB main memory and a Tesla P100 GPU of 12GB memory. LibSVM was downloaded from its official website. Nine publicly available data sets were used in our experiments. More details on the experimental setup can be found in the journal publication [4]. As we can see from Figure 1a, GMP-SVM often outperforms LibSVM by 100 times, and even over 1000 times on the *RCV1* data set. Figure 1b presents the results on prediction. GMP-SVM is 100 to 1000 times faster than LibSVM in prediction.

GMP-SVM can be viewed as a highly parallelized version of LibSVM, so they produce identical classifiers. We have measured the training and test errors to confirm if GMP-SVM produces identical results as LibSVM [4]. We found that the training and test errors are identical, which implies that GMP-SVM and LibSVM produce the same SVMs. To further confirm the SVMs trained by GMP-SVM and LibSVM are the same, we also compared the bias of the trained MP-SVMs. We find that the bias terms are also identical. More details of the comparison are available in the journal publication [4].

5. Conclusion

In this work, we have proposed a GPU based solution, called GMP-SVM, for multi-class SVMs with probabilistic output (MP-SVMs). GMP-SVM reduces high latency memory accesses and memory consumption through batch processing, kernel value reusing and sharing, and support vector sharing. Experimental results have shown that GMP-SVM outperforms LibSVM by 100 times while producing the same classifier.

Acknowledgements

This work is supported by a MoE AcRF Tier 1 grant (T1 251RES1610) and Tier 2 grant (MOE2017-T2-1-122) in Singapore. Prof. Chen is supported by the Guangdong special branch plans young talent with scientific and technological innovation (No. 2016TQ03X445), Guangzhou science and technology planning project (No. 2019-03-01-06-3002-0003) and Guangzhou Tianhe District science and technology planning project (No. 201702YH112). Bingsheng He and Jian Chen are corresponding authors. We thank NVIDIA for the hardware donations.

References

[1] Z. Wen, B. He, R. Kotagiri, S. Lu, and J. Shi, "Efficient gradient boosted decision tree training on GPUs," in *International Parallel and Distributed Processing Symposium (IPDPS)*, 2018, pp. 234–243.

[2] Z. Wen, J. Shi, Q. Li, B. He, and J. Chen, "ThunderSVM: a fast SVM library on GPUs and CPUs," *Journal of Machine Learning Research (JMLR)*, vol. 19, no. 1, pp. 797–801, 2018.

[3] S. Shalev-Shwartz and T. Zhang, "Accelerated mini-batch stochastic dual coordinate ascent," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2013, pp. 378–385.

[4] Z. Wen, J. Shi, B. He, J. Chen, and Y. Chen, "Efficient multi-class probabilistic SVMs on GPUs," *Transactions on Knowledge and Data Engineering (TKDE)*, 2018.