Alternating Mixing Stochastic Gradient Descent for Large-scale Matrix Factorization

Zhenhong Chen, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng
Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
chenzhenhong@software.ict.ac.cn, {lanyanyan, guojiafeng, junxu, cxq}@ict.ac.cn

***Abstract*—This paper is concerned with distributed stochastic gradient descent (SGD) for large scale matrix factorization (MF), which seeks to approximate a data matrix V by the product of two low rank matrices W and H. Among many distributed methods, iterative parameter mixing (IPM) has been proven to be one of the most resource-efficient and effective techniques. However, some recent empirical studies showed that IPM fails in MF. The main reason lies in the coupling of W and H, which makes the direct mixing strategy no longer correct in MF. To address the problem, we propose an alternating mixing stochastic gradient descent algorithm for the MF problem, namely AM-SGD. In the new algorithm, matrices W and H are updated alternatively with parameter mixing strategy being applied in both procedures. In this way, the correctness of mixing is guaranteed and the effectiveness of IPM is preserved. Theoretical analysis and experiment results demonstrated that AM-SGD is more efficient and effective compared to state-of-the-art distributed SGD algorithms for MF.**

I. Introduction

Recently, large-scale matrix factorization have received much attention. In this paper, we investigate it in the literature of distributed stochastic gradient descent.

Among many distributed (parallel) SGD algorithms, iterative parameter mixing seems to be elegant and is easy to implement. IPM trains models on separate data independently and iteratively mixing the model parameters. It has been shown that IPM outperforms many other distributed SGD algorithms in many optimization tasks, i.e., training conditional maximum entropy models and structured perceptron [1].

However, when apply IPM to matrix factorization, known as partition stochastic gradient descent (PSGD, unless stated otherwise, PSGD and IPM are the same algorithm in this paper), empirical results show that "averages of good local factors as computed by PSGD is converged to suboptimal solutions" [2]. Why this happens and how to solve this problem motivate the paper. Our theoretical analysis showed that the product of the mixed W and H is no longer a good approximation of V since the cross terms are introduced.

To solve the problem, we propose an alternating mixing stochastic gradient descent algorithm for MF. The new algorithm, called AM-SGD, updates H and W alternatively and the parameter mixing strategy is applied in both of the updating procedures. In this way, we show that AM-SGD can get rid of the problem of coupled mixing in PSGD.

Contributions of the paper include: (1) Theoretical analysis of the reason why directly applying the mixing strategy to MF does not work well; (2) Proposal of the alternating mixing SGD algorithm for MF, which avoid the coupled mixing problem of PSGD, thus preserved the bagging-like effect of it. (3) Theoretical analysis of the complexity and convergence rate of the proposed AM-SGD algorithm. (4) Experimental results demonstrate that AM-SGD is more effective and efficient than the state-of-the-art distributed SGD algorithms for MF.

The rest of this paper is organized as follows. After a summary of related work on distributed SGD in Section II, we analyze the underlying reason on the failure of directly applying parameter mixing strategy in SGD when applied to MF in Section III. In section IV the new algorithm AM-SGD is proposed. Experimental results with discussions are given in Section V. Section VI concludes this paper.

II. Related work

In general, distributed SGD methods can be categorized as distributed gradient descent, asynchronous updating and iterative parameter mixing. Among them, iterative parameter mixing method is the most efficient and effective one [1]. However, it fails in matrix factorization. Gemulla et al. [2] experimental showed that PSGD suffered with slow convergence. To address this issue, the authors propose another distributed SGD algorithm for matrix factorization, called DSGD. DSGD utilizes the property that some blocks of the matrix V are mutually independent in the computation. Therefore, corresponding variables of independent blocks can be updated in parallel. DSGD has become a state-of-the-art distributed matrix factorization algorithm in the literature.

III. Problem settings

In this section, we briefly analyze the underlying reason of the failure of PSGD on MF. Considering the case that the data matrix V is partitioned into d subsets, $V^{(1)}, V^{(2)}, \cdots, V^{(d)}$, where each $V^{(i)}$ is stored in node c_i, $\forall i = 1, \cdots, d$. At iteration $t+1$, each node starts with the same parameters (W_t, H_t). SGD is then conducted on each node c_i in parallel to generate local $(W_{t'}^{(i)}, H_{t'}^{(i)})$. Based on these matrices, the parameter mixing strategy is performed, and parameters (W, H) are updated as $(W_{t+1}, H_{t+1}) = (\frac{1}{d}\sum_{i=1}^{d} W_{t'}^{(i)}, \frac{1}{d}\sum_{i=1}^{d} H_{t'}^{(i)})$. Note that matrix factorization is to approximate V by WH. However, in the parameter mixing process shown above, the multiply of W_{t+1} and H_{t+1} takes the form

$$\begin{aligned} W_{t+1}H_{t+1} &= \left(\frac{1}{d}\sum_{i=1}^{d} W_{t'}^{(i)}\right)\left(\frac{1}{d}\sum_{i=1}^{d} H_{t'}^{(i)}\right) \\ &= \frac{1}{d^2}\sum_{i=1}^{d} W_{t'}^{(i)} H_{t'}^{(i)} + \frac{1}{d^2}\sum_{i\neq j} W_{t'}^{(i)} H_{t'}^{(j)}. \end{aligned}$$

We can see that $W_{t'}^{(i)} H_{t'}^{(i)}, \forall i = 1, \cdots, d$ are good approximation of V, respectively. However, the cross terms $W_{t'}^{(i)} H_{t'}^{(j)}$ $(i \neq j)$ are meaningless as $W_{t'}^{(i)}$ and $H_{t'}^{(j)}$ are trained on different subset of training data when $i \neq j$. Therefore, the multiply of W_{t+1} and H_{t+1} deviate from the approximation of V. The correctness of parameter updates are no longer guaranteed, which leads to the slow convergence rate and poor performance of PSGD.

IV. Our Approach

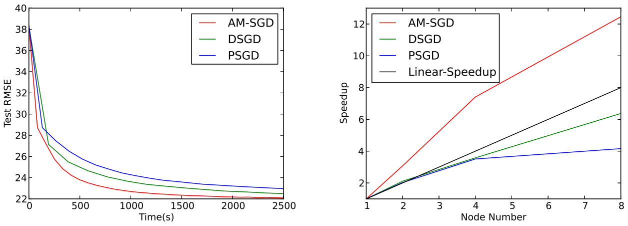
To solve the above problem, we propose an alternating mixing stochastic gradient descent algorithm. AM-SGD updates W and H alternatively and the parameter mixing strategy is applied in both of the updating procedures. Specifically, the input data matrix V as well as one factor matrix W are partitioned into $d \times 1$ blocks and distributed to d nodes. Another factor matrix H is broadcasted and each node stores a local copy of H. The alternating update procedures at the $(t+1)$-th iteration are conducted as follows: (1) Holding H_t fixed and updating W_t: at each node c_i, the assigned local $V^{(i)}$ block is further randomly partitioned into p parts. To benefit from the parameter mixing strategy, p is usually larger than one. SGD is then conducted on different local parts independently and the updated $W_{t'(j)}^{(i)}, \forall j = 1, \cdots, p$, are mixed to get $W_{t+1}^{(i)}$; (2) Holding W_{t+1} fixed and updating H_t: each node updates its local copy of H_t in parallel. The results are then aggregated to form H_{t+1} and broadcasted to each node.

Next, we will briefly show that the AM-SGD algorithm works correctly for MF. (1) When updating W_t with H_t fixed in the $(t+1)$-th iteration, the parameter mixing strategy is actually conducted independently on each node. Suppose that the $V^{(i)}$ in node c_i is randomly split into p partitions. The initial parameters of each part are $(W_t^{(i)}, H_t)$ and the updated parameters are $(W_{t'(1)}^{(i)}, H_t), \cdots, (W_{t'(p)}^{(i)}, H_t)$, respectively. Therefore, the mixed result is $\left(\frac{1}{p} \sum_{j=1}^{p} W_{t'(j)}^{(i)}, H_t\right)$. As a result, the multiply of updated W and H becomes $\frac{1}{p} \sum_{j=1}^{p} W_{t'(j)}^{(i)} H_t$, where each $W_{t'(j)}^{(i)} H_t$ $(\forall j = 1, \cdots, p)$ are good approximations of $V^{(i)}$. (2) When updating H_t with W_{t+1} fixed in the $(t+1)$-th iteration, the parameter mixing strategy is conducted over all the nodes. Given d nodes, the mixing result is $(W_{t+1}, \frac{1}{d} \sum_{i=1}^{d} H_{t'}^{(i)})$. As a result, the multiply of W and updated H becomes $\frac{1}{d} \sum_{i=1}^{d} W_{t+1} H_{t'}^{(i)}$, where each $W_{t+1} H_{t'}^{(i)}$ $(\forall i = 1, \cdots, d)$ are good approximations of V (since each $W_{t+1}^{(i)} H_{t'}^{(i)}$ is a good approximation of $V^{(i)}$). In this way, the problem of additional cross terms caused by multiply in PSGD can be well addressed.

Besides, AM-SGD achieves better scalability than PSGD with the distributing of V and W. More detail on complexity and convergence rate analysis are omitted for space saving.

V. Experiments

Settings. We used an MPI cluster as our distributed platform, which consists of 16 servers and each equipped with a four-core 2.30GHz AMD Opteron processor and 8GB RAM. We compare the proposed method with two state-of-the-art distributed stochastic gradient descent algorithms, PSGD and DSGD [2]. Results are reported on Netflix, Yahoo-music and a much larger synthetic dataset with RMSE on test set. We adopted two kinds of learning rate, const value and α/T, where T is the number of iteration. Two initialize method of W_0 and H_0 were used, sampling uniformly and at random from [-0.5,0.5] and [Mean-0.5,Mean+0.5], respectively, where Mean denote the mean value of training entries in V. Rank K of W and H were carefully chosen to achieve best performance. All parameters were selected on 10% held out training data.



(a) Yahoo-music

(b) Speedup

Figure 1. Experimental results.

Results. Due to space limited, we only report the results on the Yahoo-music dataset, with rank $K = 100$. Other parameter details and results on other methods will be included in the extension of the paper. From Figure 1(a), we could see that AM-SGD converged faster than the baselines. Among the three algorithms, PSGD converged to inferior test RMSE. This empirically conformed to our theoretically analysis that PSGD suffers slow convergence rate and poor performance, while the proposed AM-SGD algorithm gets rid of the failure of PSGD and benefits from the bagging-like effect of it.

Next, we compare the scalability of the three algorithms. We compute the speedup factor relative to the running time with one node. It is obvious from Figure 1(b) that AM-SGD achieves the best speedup. Besides, we found that with the number of nodes increased, AM-SGD still enjoys good speed up, while DSGD spends much more time on communication which results inferior speedup.

VI. Conclusion

This paper proposes a new distributed stochastic gradient descent algorithm for matrix factorization, namely AM-SGD. The motivation comes from the previous observation in [2] that PSGD fails on matrix factorization, however, it has been proven to be efficient and effective in many other optimization tasks [1]. We theoretically analyze the failure of PSGD on MF and proposed the AM-SGD algorithm to get rid of it. Theoretical and experimental results demonstrate that AM-SGD is more efficient and effective than state-of-the-art algorithms including PSGD and DSGD.

The future work includes full theoretical analysis and experimental results on the proposed AM-SGD algorithm.

References

[1] K. B. Hall, S. Gilpin, and G. Mann, "Mapreduce/bigtable for distributed optimization," in *NIPS LCCC Workshop*, 2010.

[2] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 69–77.