JSIAM Letters Vol.10 (2018) pp.61–64 ©2018 Japan Society for Industrial and Applied Mathematics JSIAM Letters

Cost-efficient cutoff method for tensor renormalization group with randomized singular value decomposition

Haruka Yamada[1], Akira Imakura[1] and Tetsuya Sakurai[1,2]

[1] University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan
[2] JST/CREST

E-mail *yamada@mma.cs.tsukuba.ac.jp*

Received May 01, 2018, Accepted August 20, 2018

Abstract

Tensor renormalization group (TRG) is a coarse-graining algorithm for approximating the partition function using a tensor network in the field of elementary particle physics. Although the computational cost of TRG can be reduced using a randomized singular value decomposition, its computation time is still large. In this paper, we propose a cost-efficient cutoff method for calculating TRG by truncating small tensor elements. Numerical experiments showed that the proposed method is faster than the conventional one without degrading accuracy.

Keywords tensor renormalization group (TRG), randomized singular value decomposition, cutoff

Research Activity Group Algorithms for Matrix / Eigenvalue Problems and their Applications

1. Introduction

In physics, the partition function is an important physical quantity to obtain the critical temperature of a system. Because the computational cost of calculating the partition function exactly increases exponentially with system size, it is often calculated approximately. Monte Carlo method has been widely used as an approximation method, but it is difficult to apply to a large system with complex numbers. Tensor network approaches have attracted attention as a new approximation method that overcomes this difficulty. In this approach, the partition function is represented by a network of tensors that is then approximated by coarse-graining the tensor network. There are many coarse-graining methods for different target physical models such as DMRG [1, 2], tensor renormalization group (TRG) [3,4], TNR [5] and HOTRG [6]. For the details of the tensor network methods, we refer [7] and the references therein.

In this paper, we focus on TRG for a two-dimensional (2D) square lattice. TRG approximates the partition function by a sequential coarse-graining procedure which comprises an approximation step based on singular value decomposition (SVD) and a contraction step to contract some tensors. The computational cost of both steps is $\mathcal{O}(D^6)$, where a parameter D is related to the accuracy and is called the bond dimension. Morita et al. reduced the computational cost of these two steps to $\mathcal{O}(D^5)$ [4] using a randomized SVD (RSVD) [8]. In general, the partition function should be calculated for various temperatures to obtain the critical temperature. Therefore, although the computational cost has been reduced, it is still a serious problem of TRG.

For efficient implementations, the tensor contractions are implemented with GEMM and reorderings, which sort elements of the matrices. Because so many reorderings are required and the size of the matrices are large, the computation time for reorderings is also a problem of TRG. In this paper, we propose a cost-efficient method that uses a cutoff for small elements of tensors to reduce the computation time of matrix products and reorderings. We also evaluate its effect on accuracy with a 2D classical square lattice Ising model by comparing the exact solution.

The remainder of this paper is organized as follows. Section 2 introduces the basic concept of TRG and TRG with RSVD. We also introduce its GEMM-based implementation. In Section 3, we propose the cost-efficient method using the cutoff. The performed numerical experiments are reported in Section 4, and Section 5 concludes the paper.

Throughout this paper, we use the MATLAB colon notations.

2. TRG

Here, we consider computing the free energy f per site of the 2D classical square lattice Ising model with zero magnetic field and a periodic boundary condition. The partition function Z_V with volume V is defined as

$$Z_V = 2^V (\cosh \beta)^{2V} \sum_{\{n\}} \prod_{i=1}^{N} \mathcal{T}^i_{n_a n_b n_c n_d}, \tag{1}$$

where $\mathcal{T}^i_{pqrs} \in \mathbb{R}^{2\times 2\times 2\times 2}$ represent two-dimensional 4th order tensors and N is the number of tensors. Then, the free energy per site is given by

$$\frac{f}{V} = -\frac{1}{\beta} \log \frac{Z_V}{V}.$$

Because each tensor represents one site of this system, $V = N$. In this system, all the tensors $\mathcal{T}^i_{pqrs}$ are the same

and their elements are given by

$$\mathcal{T}_{pqrs} := \begin{cases} 1 & (w=0), \\ \tanh\beta & (w=2), \\ \tanh^2\beta & (w=4), \\ 0 & (other), \end{cases}$$

where β is the inverse temperature, and $w = p+q+r+s$. In the following subsections, we overview TRG and TRG with RSVD; and present the GEMM-based implementation for the tensor contractions.

2.1 Outline of TRG

Because the computational costs increase exponentially with system size N, to compute as defined the partition function by (1) is difficult. TRG contracts the tensors using a sequential procedure and reduces the computational costs by restricting the dimensions of the tensors to the bond dimension D which is the parameter relating accuracy.

Let tensor $\mathcal{T}^{(K)}$ be obtained by coarse-graining 2^K initial tensors ($V = N = 2^K$). Then, the partition function Z_V is approximated by

$$Z_V \approx Z_K^{'} = 2^{2^K}(\cosh\beta)^{2^{2(K+1)}}\mathrm{Tr}(\mathcal{T}^{(K)}).$$

Because the boundary condition is periodic, the tensor trace is calculated as $\mathrm{Tr}(\mathcal{T}^{(K)}) = \sum_i \sum_j \mathcal{T}^{(K)}_{ijij}$.

TRG performs the coarse-graining procedure sequentially. A tensor $\mathcal{T}^{(k+1)}$ is computed from $\mathcal{T}^{(k)}$ as follows. Here, the size of the tensor $\mathcal{T}^{(k)}$ is $D_k \times D_k \times D_k \times D_k$, where $D_k = \min(D_{k-1}^2, D)$. TRG computes the low-rank approximation of the tensor $\mathcal{T}^{(k)}$ with two 3rd order tensor $\mathcal{S} \in \mathbb{R}^{D_k \times D_k \times D_{k+1}}$ as following:

$$\min_{\mathcal{S}^1,\mathcal{S}^3} \|\mathcal{T}^{(k)}_{pqrs} - \sum_\alpha \mathcal{S}^1_{sp\alpha}\mathcal{S}^3_{qr\alpha}\|, \tag{2}$$

$$\min_{\mathcal{S}^2,\mathcal{S}^4} \|\mathcal{T}^{(k)}_{pqrs} - \sum_\alpha \mathcal{S}^2_{pq\alpha}\mathcal{S}^4_{rs\alpha}\|. \tag{3}$$

Then, $\mathcal{T}^{(k+1)}$ is obtained as

$$\mathcal{T}^{(k+1)}_{pqrs} = \sum_{i,j,k,l} \mathcal{S}^1_{ijp}\mathcal{S}^2_{liq}\mathcal{S}^3_{klr}\mathcal{S}^4_{jks}.$$

From the above equations, each iteration of TRG comprises the approximation step including (2) and (3) and the contraction step (2.1). To simplify the following equations, we set all the dimensions of the tensor $\mathcal{T}^{(k)}$ as D.

First, we describe the approximation step. The tensors $\mathcal{S}$ are calculated by the low-rank approximation of matrices M^{13}, $M^{24} \in \mathbb{R}^{D^2 \times D^2}$, which are obtained by unfolding the tensor $\mathcal{T}^{(k)}_{pqrs}$ in the following ways:

$$M^{13}_{(sp)(qr)} = \mathcal{T}^{(k)}_{pqrs}, \quad M^{24}_{(pq)(rs)} = \mathcal{T}^{(k)}_{pqrs}. \tag{4}$$

For example, the equations to obtain $\mathcal{S}^1$ and $\mathcal{S}^3$ from M^{13} are

$$[U,\Sigma,V] = \mathrm{svd}(M^{13}), \tag{5}$$

$$S^1 = U(:,1:D)\sqrt{\Sigma(1:D,1:D)},$$

$$S^3 = V(:,1:D)\sqrt{\Sigma(1:D,1:D)},$$

$$\mathcal{S}^1_{sp\alpha} = S^1_{(sp)\alpha},$$

$$\mathcal{S}^3_{qr\alpha} = S^3_{(qr)\alpha},$$

where $[U,\Sigma,V] = \mathrm{svd}(M^{13})$ denotes SVD of M^{13} as $M^{13} = U\Sigma V^{\mathrm{T}}$. The tensors $\mathcal{S}^2$ and $\mathcal{S}^4$ are calculated from M^{24} in the same way as $\mathcal{S}^1$ and $\mathcal{S}^3$ are calculated from M^{13}.

Next, we consider the contraction step (2.1). Although the computational cost of (2.1) is $\mathcal{O}(D^8)$ if we contract all tensors at the same time, it can be reduced to $\mathcal{O}(D^6)$ by e.g.,

$$\mathcal{X}^{12}_{jlpq} = \sum_i \mathcal{S}^1_{ijp}\mathcal{S}^2_{liq}, \quad \mathcal{X}^{34}_{ljrs} = \sum_k \mathcal{S}^3_{klr}\mathcal{S}^4_{jks},$$

and

$$\mathcal{T}^{(k+1)}_{pqrs} = \sum_{j,l} \mathcal{X}^{12}_{jlpq}\mathcal{X}^{34}_{ljrs}. \tag{6}$$

Therefore, the bottleneck of TRG is SVD (5) in the approximation step and the tensor contraction (6) in the contraction step.

2.2 Outline of TRG with RSVD

Using RSVD [8], Morita et al. reduced computational cost for both of the approximation step and the contraction step [4]. In this subsection, we present an outline of the algorithm for TRG with RSVD. The RSVD algorithm for computing the rank-k approximation of a matrix $A \in \mathbb{R}^{m\times n}$ is shown in Algorithm 1 and its computational cost is $\mathcal{O}(mnk)$. Here, $\mathrm{orth}(Y)$ denotes the orthogonalization of the matrix Y. In TRG with RSVD, because the size of matrices M^{13} and M^{24} is $D^2 \times D^2$ and the rank-D approximation is required, the computational cost of the approximation step is $\mathcal{O}(D^5)$.

We consider reducing the computational costs for the contraction step by avoiding computing the tensor $\mathcal{T}^{(k+1)}_{pqrs}$ (6) and the matrices M^{13} and M^{24} (4). Instead, we compute the matrices

$$Y^{13} = M^{13}(M^{13\mathrm{T}}M^{13})^q\Omega,$$

$$Y^{24} = M^{24}(M^{24\mathrm{T}}M^{24})^q\Omega,$$

in Step 1 of RSVD of the $(k+1)$th iteration directly from the tensors $\mathcal{S}^1$, $\mathcal{S}^2$, $\mathcal{S}^3$ and $\mathcal{S}^4$ of the kth iteration. From the definition of the tensor $\mathcal{T}^{(k+1)}_{pqrs}$ (2.1) and the matrices M^{13}, M^{24} (4), the matrices Y^{13}, Y^{24} can be computed by the following procedures:

$$X^{14}_{(sp)(ik)} = \sum_j \mathcal{S}^1_{ijp}\mathcal{S}^4_{jks}, \tag{7}$$

$$X^{23}_{(ik)(qr)} = \sum_l \mathcal{S}^2_{liq}\mathcal{S}^3_{klr}, \tag{8}$$

$$Y^{13} = X^{14}X^{23}\left[(X^{23})^{\mathrm{T}}(X^{14})^{\mathrm{T}}X^{14}X^{23}\right]^q\Omega, \tag{9}$$

$$X^{12}_{(pq)(jl)} = \sum_i \mathcal{S}^1_{ijp}\mathcal{S}^2_{liq}, \tag{10}$$

$$X^{34}_{(jl)(rs)} = \sum_k \mathcal{S}^3_{klr}\mathcal{S}^4_{jks}, \tag{11}$$

$$Y^{24} = X^{12}X^{34}\left[(X^{34})^{\mathrm{T}}(X^{12})^{\mathrm{T}}X^{12}X^{34}\right]^q\Omega. \tag{12}$$

Algorithm 1 RSVD for rank-k approximation

Input: $A \in \mathbb{R}^{m\times n}$, Gaussian test matrix $\Omega \in \mathbb{R}^{n\times 2k}$
Output: $A \approx U\Sigma V^{\mathrm{T}}, U \in \mathbb{R}^{m\times k}, \Sigma \in \mathbb{R}^{k\times k}, V \in \mathbb{R}^{n\times k}$
1: Compute $Y = A(A^{\mathrm{T}}A)^q\Omega$
2: $Q = \mathrm{orth}(Y)$
3: Compute $B = Q^{\mathrm{T}}A$
4: Compute SVD $[\widehat{U}, \widehat{\Sigma}, \widehat{V}] = \mathrm{svd}(B)$
5: Set $U = Q\widehat{U}(:, 1:k), \Sigma = \widehat{\Sigma}(1:k, 1:k)$ and $V = \widehat{V}(:, 1:k)$

By computing (9) and (12) from the right side, the computational cost of the contraction step is reduced to $O(D^5)$.

2.3 GEMM-based implementation

To compute the contraction of two tensors using a matrix product, reordering the indexes of the tensors is necessary. Here, we explain the GEMM-based implementation using tensor reorderings via an example of the contraction of two tensors as follows: $\mathcal{C}_{abcd} = \sum_{i,j} \mathcal{A}_{aicj}\mathcal{B}_{bjdi}$. To compute this contraction using a matrix product, the index reorderings should satisfy the following conditions:

- The contracting indexes (i, j) and the others (a, b, c, d) are separated into columns and rows in each matrix.
- The contracting indexes of two matrices are lined up in the same order.

Many procedures satisfy these conditions. One possible way is as follows:

Step 1 **[RO]** $A_{(ac)(ij)} \leftarrow \mathcal{A}_{aicj}$
Step 2 **[RO]** $B_{(ij)(bd)} \leftarrow \mathcal{B}_{bjdi}$
Step 3 **[MM]** $C = AB$
Step 4 **[RO]** $\mathcal{C}_{abcd} \leftarrow C_{(ac)(bd)}$

Here, **[RO]** and **[MM]** denote tensor reordering and matrix product, respectively.

In TRG with RSVD, the tensor contractions (7) – (12) are implemented with GEMM and reorderings.

3. Proposed cost-efficient method

In this section, we breakdown the conventional method for a 2D classical square lattice Ising model. Then, we propose a cost-efficient method using matrix element cutoff.

3.1 Conventional method

Fig. 1 shows the breakdown of the elapsed time for the conventional method around $\beta = 2.269$ which is the critical temperature of this system. Here, we note that the target model and the parameters are the same as in Section 4.2. The total elapsed time to compute 41 temperatures is 2.4×10^4 seconds. Fig. 1 shows that the most time-consuming part consists of the reorderings and GEMM. This occurs because the tensors are large ($D = 100$) and dense. Sparsifying the tensors can reduce the cost of the reorderings and matrix products.

On the other hand, Fig. 2 shows that the largest truncated singular values $\Sigma(D+1, D+1)$ in (5) after the 6th

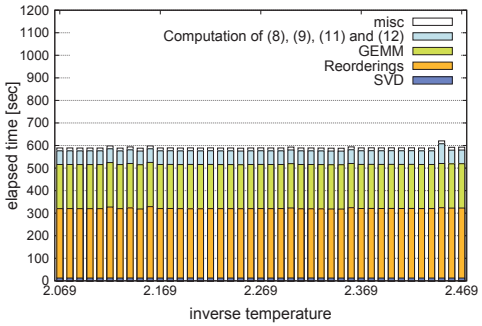

Fig. 1. Elapsed time for the conventional method for 30 iterations of TRG with $D = 100$.

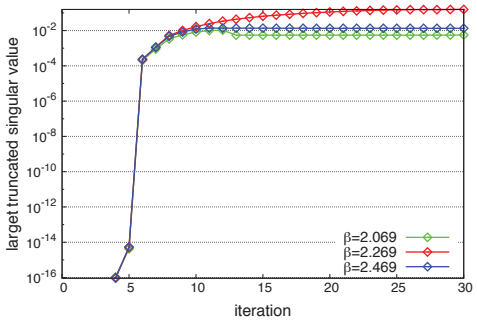

Fig. 2. Largest truncated singular values on each iteration at three temperatures.

iteration are larger than 10^{-5}, where the largest singular value is normalized to 1.0. This result indicates that perturbations smaller than 10^{-5} for the matrices M^{13} and M^{24} do not substantially affect the accuracy of the partition function.

3.2 Proposed method

Because the matrices in TRG are large and dense and perturbations of M^{13} and M^{24} would not substantially affect on the accuracy, it is considered that sparsifying the tensors by cutting off small tensor elements would hardly affect the accuracy while reducing the time required for reorderings and GEMM.

Here, we propose a cutoff method to reduce the computation time of TRG with RSVD. In this method, matrix elements are either changed to zero or not unchanged depending on a threshold. Let δ be the cutoff threshold, then a matrix X is sparsified as follows:

$$X = [x_{ij}] \approx \widetilde{X} = [\widetilde{x}_{ij}],$$

$$\widetilde{x}_{ij} = \begin{cases} x_{ij} & (|x_{ij}| \geq \delta), \\ 0 & (|x_{ij}| < \delta). \end{cases}$$

In the proposed method, we apply this cutoff to matrices X^{14}, X^{23}, X^{12} and X^{34} in (7), (8), (10) and (11) before computing (9) and (12).

After the cutting off has been applied to the matrices, (9) and (12) are calculated with reorderings and Sparse GEMM. We store the sparse matrices in coordinate (COO) format, because reordering in COO format is simpler than in other formats such as CRS. The reorderings in other formats require extra sorting of arrays, but in the tensor reordering in COO format, the indexes of the tensor can be calculated from row or column in-

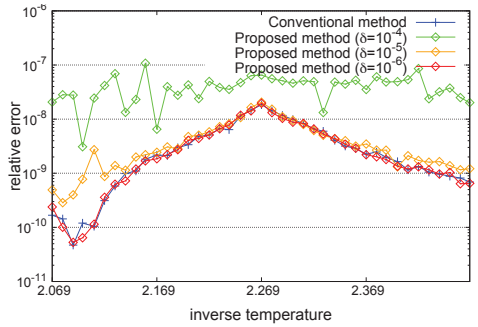

Fig. 3. Relative error of the proposed method with the cutoff threshold $\delta = 10^{-4}, 10^{-5}$ and 10^{-6}.

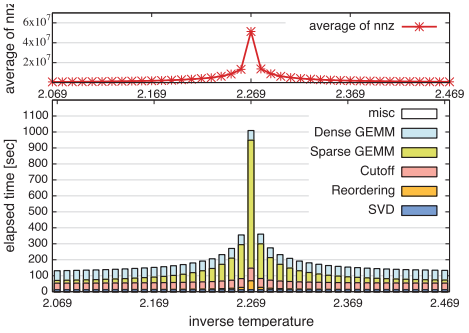

Fig. 4. Elapsed time for the proposed method with $\delta = 10^{-6}$ for 30 iterations of TRG with $D = 100$.

dexes and stored directly into COO format.

4. Numerical Experiments

In this section, we evaluate the effect on accuracy and the performance of the cutoff method proposed in Section 3 for the same model in Section 3. We set the bond dimension D to 100 and set $q = 0$ in RSVD as in [4]. The singular values are normalized so that the largest singular value equals to 1.0. In all the numerical experiments, the algorithms are implemented in C++ and the experiments were conducted using double precision arithmetic on OS: CentOS 6.6, CPU: Intel Xeon E5-2667v3, and memory: 512GB (DDR4). We used icpc 15.0.2 as the compiler and "-O3" as the optimization option.

4.1 Experiment I

In this experiment, we check the accuracy of approximated partition functions obtained by the proposed method while varying the values of cutoff thresholds $\delta = 10^{-4}$, 10^{-5} and 10^{-6}. The partition function is calculated for 41 points of inverse temperatures $\beta = 2.069$, $2.079, \ldots, 2.469$.

The relative error of the proposed method and the conventional method compared with the exact solutions are shown in Fig. 4.1. We see that the proposed method with the value of $\delta = 10^{-6}$ does not affect the accuracy. In contrast, the relative error when $\delta = 10^{-4}$ and 10^{-5} is used increases for all or some temperatures.

4.2 Experiment II

In this experiment, we evaluate the performance of the proposed method when $\delta = 10^{-6}$.

Fig. 4.2 presents the total elapsed time and average of the number of nonzero elements of sparsified matrices for 30 iterations of coarse-graining at each temperature. The total time of the proposed method is 1.0×10^4 seconds. A comparison with the total time of Fig. 1 (2.4×10^4 seconds) shows that the proposed method is 2.4 times faster than the conventional method. Moreover, we observe that the elapsed time was reduced except $\beta = 2.269$. Computation time at $\beta = 2.269$ increased because the number of nonzero elements is large. Although the total elapsed time increased at $\beta = 2.269$, sum of the elapsed time for reordering and a cutoff is less than the reordering time in the conventional method. To avoid increasing time at $\beta = 2.269$, we should use dense matrix products.

5. Conclusion

In this paper, we applied the cutoff method to the matrices in TRG with RSVD. The numerical results showed that the proposed method reduces the computation time of reordering at all temperatures, and is 2.4 times faster than the conventional one.

In the future, we aim to work on the following tasks:

- the calculation of other physical quantities or physical models
- precise analysis of the relation between the cutoff threshold δ and truncated singular values
- the dependency of the number of nonzero elements after cutoff on temperature

Acknowledgments

This research was supported partly by MEXT as "Priority Issue on Post-K computer" (Elucidation of the Fundamental Laws and Evolution of the Universe), JST/ACT-I (No. JPMJPR16U6), JST/CREST, JSPS KAKENHI (No. 17K12690).

References

[1] S. R. White, Density matrix formulation for quantum renormalization groups, Phys. Rev. Lett., **69** (1992), 2863.
[2] U. Schollwoeck, The density-matrix renormalization group in the age of matrix product states, Ann. Phys., **326** (2011), 96–192.
[3] M. Levin and C. P. Nave, Tensor renormalization group approach to two-dimensional classical lattice models, Phys. Rev. Lett., **99** (2007), 120601.
[4] S. Morita, R. Igarashi, H. Zhao and N. Kawashima, Tensor renormalization group with randomized singular value decomposition, arXiv:1712.01458v1, 2017.
[5] G. Evenbly and G. Vidal, Tensor network renormalization, Phys. Rev. Lett., **115** (2015), 180405.
[6] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang and T. Xiang, Coarse graining tensor renormalization by the higher-order singular value decomposition, Phys. Rev. B, **86** (2012), 045139.
[7] R. Orus, A practical introduction to tensor networks: matrix product states and projected entangled pair states, Ann. Phys., **349** (2014), 117–158.
[8] N. Halko, P. G. Martinsson and J. A. Tropp, Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev., **53** (2011), 217–288.