Probabilistic Time Context Framework for Big Data Collaborative Recommendation

Emelia Opoku Aboagye
University of Electronic Science and Technology of China
0086-18628333154
eoaboagye@yahoo.co.uk

Gee C. James
University of Electronic Science and Technology of China
008618982067786
gee@uestc.edu.cn

Gao Jianbin
University of Electronic Science and Technology of China
008618982067786
gaojb@uestc.edu.cn

Rajesh Kumar
University of Electronic Science and Technology of China
0086-15520777096
rajakumarlohano@gmail.com

Riaz Ullah Khan
University of Electronic Science and Technology of China
0086-15520763595
rerukhan@gmail.com

ABSTRACT

A parallel scheme based on Probabilistic Tensor Factorization which addresses the scalability problem of Collaborative Filtering (CF) is proposed for big data processing. Parallel algorithms for large scale recommendation problems have witnessed advancements in the big data era in recent times. Matrix Factorization models have been enormously used to tackle such constraints, which we see as not scalable and does not converge easily unless numerous iterations making it computationally expensive. This study proposes a novel coordinate descent based probabilistic Tensor factorization method; Scalable Probabilistic Time Context Tensor Factorization (SPTTF) for collaborative recommendation. Our experiments with natural datasets show its efficiency.

CCS Concepts

• **Mathematics of computing → Probabilistic algorithms**

Keywords

Time contest; tensor; algorithm integration; SPTTF.

1. INTRODUCTION

Recommendation systems (RS) typically produces a list of recommendations through collaborative and content-based filtering techniques. It is a subclass of information filtering that seeks to predict ratings or preference a user or customer will give to an item Different approaches have been employed to tackle the issue of scalability for large-scale tensor data. Tensor Factorization Algorithms have been very influential in such paradigms and have recently achieved significant developments and impacts in signal processing [1], computer vision [2] numerical analysis [3], social network analysis [4] recommendation systems [5], just to mention a few. A comprehensive overview can be found from the survey paper by [6]. In this paper, we present Scalable Probabilistic Time Context Tensor Factorization (SPTTF) dedicated for parallelizing large-scale tensor factorization problems in collaborative filtering. The main contribution of this paper are;

We propose new parallelization scheme; SPTTF which partitions big data or large-scale problem into several sub-modules for concurrent executions that makes it scalable. The convergence of SPTTF is theoretically guaranteed through our experiments on real world datasets. Temporal modeling has been greatly ignored in the community of collaborative filtering until the time SVD++ algorithm.

1.1. Tensor Factorization Model

Tensor decompositions have been defined in several ways [9]. Our definition is based on CANDECOMP/PARAFAC (CP) decomposition, which is one of the most popular decomposition methods. Details about CP decomposition can be found in [10]. In this paper, we only derive our model in the third-order case. Let Problem Formulation $\chi \in \Re^{I \times J \times K}$ be a third order tensor with observable entries $\{\chi_{ijk}(_{i,j,k}) \in \varphi\}$. The factor matrices

$$U \in \Re^{I \times R}, V \in \Re^{J \times R}, T \in \Re^{K \times R}$$

is minimized as ;

$$\min_{U,V,T} \sum_{(i,j,k)\in\varphi} (x_{i,j,k} - < U_i, V_j, T_k >)^2 +$$

$$\frac{\beta}{2}(\| U \|_F^2 + \|\|\| V \|_F^2 + \| T \|_F^2) \quad (1)$$

$$<\mathrm{U}_i, \mathrm{V}_j, T_k > \equiv \sum_{r=1}^{R} u_{ir} v_{jr} t_{kr}$$

$T_k > \equiv \sum_{r=1}^{R} u_{ir} v_{jr} t_{kr}$ denotes the inner product of three R-dimensional vectors, (u_i) denotes the i-th row of (U). So does (V_j) and (T_k). Probabilistic Temporal Tensor Factorization (PTTF) model can be considered as the extension of PMF model [11]-[13] by adding a purposeful constrained 'Time' dimension. We therefore model our integrated model in equation (1) as a tensor model. For the third order tensor X if the third dimension denotes the time corresponding to the factor matrix T, we assume the

ICCAI 2018, March 12–14, 2018, Chengdu, China

ACM ISBN 978-1-4503-6419-5/18/03…$15.00
https://doi.org/10.1145/3194452.3194458

following conditional prior for

$$T_k \sim N(T_{k-1}, \eta^2{}_t \mathrm{I}_R) K = 1, \ldots, K$$

$$+\frac{\beta\ T}{2}\sum_{|0|}^{K}||T_K^N||T_K^N - T_{k=1}^N||_2^2 + \frac{\beta\ _0}{2}||T_0^N - \alpha_T||_2^2$$

$$\frac{1}{2}\sum_{(i,j,k)\in R}(x_{ijk} - <U_i V_j T_k>)^2 + \frac{\beta_0}{2}\|T_0 - \alpha_t\|_2^2 +$$

$$\frac{\beta_u}{2}\|U\|_F^2 + \frac{\beta V}{2}\|V\|_F^2 + \frac{\beta T}{2}|\sum_{k=1}^{K}\|T_k - T_{k-1}\|_2^2$$

Here $\beta_U, \beta_V, \beta_T, \beta_0$ are regularizing parameters. For issues of tractability and most importantly, to design parallel learning algorithms for scalability for big data problems where data abounds in volume and in variety, we propose Coordinate Based Recommendation System which is shown to outperform the most popular algorithmic techniques in the field of recommendation [14]. Coordinate descent algorithms solve optimization problems by successively performing approximate minimization along coordinate directions or coordinate hyperplane. They have been used in applications for many years, and their popularity continues to grow because of their usefulness in data analysis, machine learning, and other areas of current interests.

2. PROBABILISTIC TENSOR DATA PARTITIONING

We divide the tensor (X) into N sub-blocks along the mode. Each sub-block contains N1 horizontal slices. This partition strategy enables the CP decomposition problem to fit in the parallel Coordinate Descent framework. Algorithm (1) demonstrates the details of proposed partitioning for parallel processing. In the partition setting, the minimization problem is viewed as multi-core system processing for parallelization. Thus the minimization problem in Equation (2) can be reformulated as a constrained optimization problem as;

$$\min\_[(U^N V^N T^N)\_(T_0^N \bar{V}, \bar{T})][\sum_{N=1}^{N}[f(U^N, V^N.T^N)$$

$$+g(U^N, V^N.T^N, T_0^N)]]$$

$$f(U^N, V^N.T^N) = \frac{1}{2}\sum_{i,j,k\in\ ,N}(X_{ijk}^N - <U_i^N, V_j^N, T_k^N>)^2$$

Denotes i, j, k indices of the values located in processes. Our motivation is that, several variables are independent and can be updated concurrently, thus updating one column at a time. It is important to select an appropriate parallel environment based on the scale of the recommender system. denotes the (i, j, k) located in the processes. our motivation is that, several variables are independent and can be updated concurrently, thus updating one column at a time

3. EXPERIMENTS

Coordinate descent is a classic and well-studied optimization technique [9], [10]. We employ a technique similar to that in [11] to resolve this issue through a coordinate descent framework. Per our partitioning strategy for parallelization, instead of being assigned to a fixed core, assignment is done in a manner that is dynamic based on the availability of each core. When a core finishes a small task in a sub-tensor, it can always start a new task without waiting for other cores. That means, partition I/N will refer to the indices assigned to the (r – th) core as a result of this strategic assignment. Such an approach can be also applied to update v and the residual X

$$X_{ijk} \leftarrow X_{ijk} + \bar{V}\ T_{ijk}\ V_{ijk}, i\ j\ k \in \Omega\ _m \quad (3)$$

$$\Omega M\mathrm{r} = \cup_{i\in M_r}\{(i,j,k) : j \in \Omega_i\} \quad (4)$$

Each core r then updates

$$\bar{V} \leftarrow \frac{\Sigma j \in \Omega, Muv_j}{\lambda + \Sigma j \in \Omega, \bar{T}_j} \quad (5)$$

Updating T can be parallelized in the same way with G =[fG1; : : : ; Gpg], which is a partition of row indices of T, Similarly, each core r

$$\bar{\mathrm{T}}_j \leftarrow \frac{\Sigma_i \Omega, M_{ij} Vg}{\lambda + \Sigma_i \in \Omega, \cup_i \bar{T}_i} \quad (6)$$

As all cores on the machine share a common memory space, no communication is required for each core to access the latest (V) and (T). After obtaining \(u*; v*\), we can also update the residual (M) and w¯ r t; h¯ r t in parallel by assigning core r to perform the update:

$$V_i, T_j \leftarrow (\bar{V}, \bar{T}) \quad (7)$$

$$M_{ijk} \leftarrow M_{ijk} - \bar{V}_i, \bar{T}_j, \forall (i,j) \in \Omega X_r \quad (8)$$

If the matrices (A), (W) and (H) in a single machine, Coordinate descent algorithm \cite{}can achieve significant speedup by utilizing all cores available on the machine. The key component in this proposition is that, parallelization is the computation to solve sub-problem. In the coordinate descent algorithm framework, the approximate solution to the sub-problem is obtained by updating (u), (v) and (t) alternately. When (v) is indexed, each variable (u) can be updated independently. Therefore, the update to (u) can be divided into (m) independent jobs which can be handled by different cores in parallel. This can be viewed as special cases of the above setting where different only act on a subset of variables. A mini-batch algorithm is also novely used to inferencing ensuring our algorithm SPTTF scalable.

4. RESULTS

Datasets and Parameter Settings: Public collaborative filtering datasets; Amazon dataset [15]. The hyper-parameters are all determined by k-fold cross validation. In order to study PTTD's parallel performance, we process the data into three 3rd-order tensors, where each mode correspond to Reviewers of Automotive (users), Asin (Items) and Time context, respectively.

Scalability: One of the metrics used to measure a parallel algorithm is the scalability. To study the scalability of SPTTF, we test our model on two datasets by varying the number of cores from 1 to 8. The result demonstrates that the running time is approximately reduced to a half when the number of cores gets doubled. Our comparison table shows the outcome.

Algorithm2. Parallelization

1: **Input:** Tensor $X \in R^{I \times J \times K}$, U, V, T, X, N, Iterate
2: Initialize $V = 0, R, U$
3: Core r constructs $\hat{R}$ using Eqn. (4)
4: $Iterate = 1, 2, \ldots, T$ **do**
5: Core r updates U
6: Core r updates α
7: **End for**
8: Core r updates $\bar{V}$ and $\bar{t}^{r}_{i-1}$
9: Core r updates R
10: **End for**

Table 1. Comparision table

MODELS	BPMF	PPMF	PTTF	CP	SPTTD
CORE1	0.920	0.950	0.890	0.8750	0.810
CORE2	0.910	0.900	0.880	0.75	0.711
CORE3	0.880	0.890	0.801	0.875	0.701

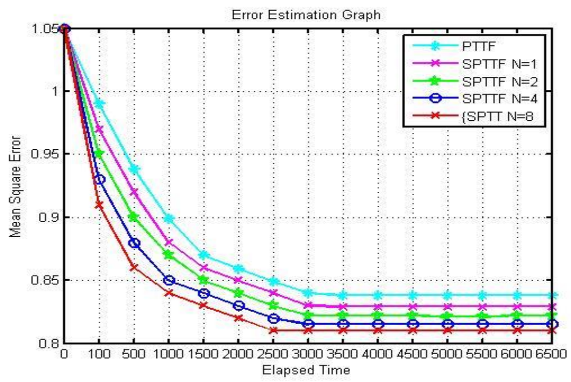

Figure 1. Core1 graph.

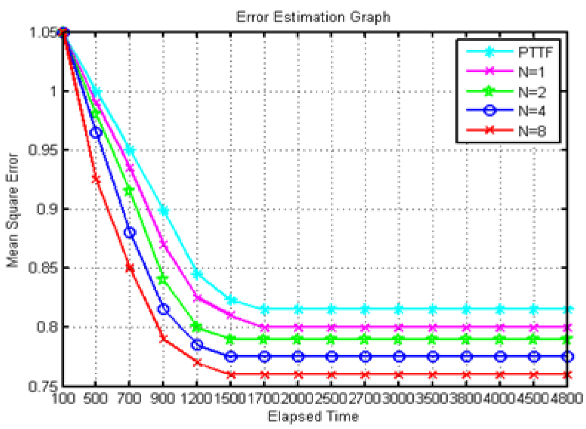

Figure 2. Core2 graph.

5. CONCLUSION

In this paper, we present SPTTF by deriving a mini-batch stochastic coordinate descent algorithm to calculate the latent factors of probabilistic time contest tensors. We propose a new data tensor partitioning to divide the big data problem into several independent sub-problems along the user dimension. Then we use the parallel coordinate descent algorithm framework to decompose these sub-tensors in parallel. Experiments on real world data sets demonstrate that our SPTTF model outperforms the traditional CP decomposition, PTTF and PPMF model in terms of efficiency and scalability.

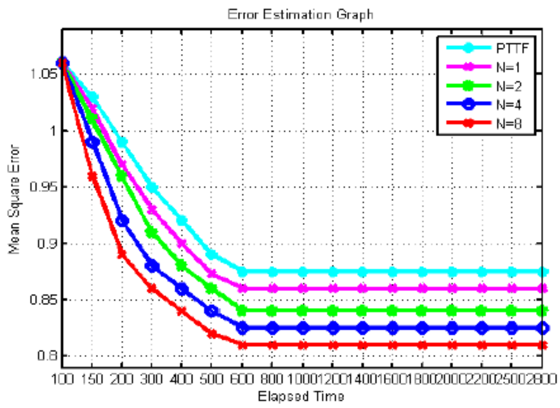

Figure 3. Core3 graph.

Our framework SPTTD shows improved performance in all the cores.

6. REFERENCES

[1] Abbas, A., Zhang, L. and Khan, S.U., 2015. A survey on context-aware recommender systems based on computational intelligence techniques. Computing, 97(7), pp.667-690.

[2] Cichocki, A., 2014. Tensor networks for big data analytics and large-scale optimization problems. arXiv preprint arXiv:1407.3124..

[3] Cichocki, A., Lee, N., Oseledets, I.V., Phan, A.H., Zhao, Q. and Mandic, D., 2016. Low-Rank Tensor Networks for Dimensionality Reduction and Large-Scale Optimization Problems: Perspectives and Challenges PART 1. arXiv preprint arXiv:1609.00893.

[4] Tan, W., Cao, L. and Fong, L., 2016, May. Faster and cheaper: Parallelizing large-scale matrix factorization on gpus. In Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing (pp. 219-230). ACM.

[5] Yujie, Z. and Licai, W., 2010, August. Some challenges for context-aware recommender systems. In Computer Science and Education (ICCSE), 2010 5th International Conference on (pp. 362-365). IEEE.

[6] Xiong, L., Chen, X., Huang, T.K., Schneider, J. and Carbonell, J.G., 2010, April. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In Proceedings of the 2010 SIAM International Conference on Data Mining (pp. 211-222). Society for Industrial and Applied Mathematics.

[7] Liu, Q., Wu, S., Wang, L. and Tan, T., 2016, February. Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts. In AAAI (pp. 194-200).

[8] Kefalas, P. and Manolopoulos, Y., 2017. A time-aware spatio-textual recommender system. Expert Systems with Applications, 78, pp.396-406.

[9] Hsieh, C.J., Chang, K.W., Lin, C.J., Keerthi, S.S. and Sundararajan, S., 2008, July. A dual coordinate descent method for large-scale linear SVM. In Proceedings of the 25th international conference on Machine learning (pp. 408-415). ACM.

[10] Wright, S.J., 2015. Coordinate descent algorithms. Mathematical Programming, 151(1), pp.3-34.

[11] You, Y., Lian, X., Liu, J., Yu, H.F., Dhillon, I.S., Demmel, J. and Hsieh, C.J., 2016. Asynchronous parallel greedy coordinate descent. In Advances in Neural Information Processing Systems (pp. 4682-4690).