# Low-Rank Matrix Completion over Finite Abelian Group Algebras for Context-Aware Recommendation

Chia-An Yu, Tak-Shing Chan, Yi-Hsuan Yang
Research Center for IT Innovation, Academia Sinica, Taipei, Taiwan
b01201010@ntu.edu.tw,{takshingchan,yang}@citi.sinica.edu.tw

## ABSTRACT

The incorporation of contextual information is an important part of context-aware recommendation. Many context-aware recommendation systems adopt tensor completion to include contextual information. However, the symmetries between dimensions of a tensor induce an unreasonable assumption that users, items and contexts should be treated equally in recommender systems. In this paper, we address this by using matrices over finite abelian group algebra (AGA) to model context-aware interactions between users and items. Specifically, we formulate context-aware recommendation as a low-rank matrix completion problem over AGA (MC-AGA) and derive a new algorithm using the inexact augmented Lagrange multiplier method. We then test MC-AGA on two real-world datasets: one containing implicit feedback and one with explicit feedback. Experiment results show that MC-AGA outperforms not only existing tensor completion algorithms but also recommendation systems with other context-aware representations.

## CCS CONCEPTS

**•Information systems →Recommender systems;**

## KEYWORDS

Context-aware recommendation, low-rank modeling, group algebra, matrix/tensor completion

## 1 INTRODUCTION

The goal of a context-aware recommender system (CARS) is to find out items of interest for each user by taking into account their preferences, item properties and, most importantly, the context of the user where the recommendation is to be provided. While conventional recommendation relies only on user-item interactions, CARS exploits contextual information that can influence the relevance of items to each user. There are many ways to achieve context-awareness, including pre-filtering, post-filtering and contextual modeling [2]. Contextual modeling is more popular than the other two, because it takes context into account during the decision process and contextual information can thereby directly affect the recommendation results.

In traditional recommender systems, user-item interaction can be easily represented by a real-valued data matrix. However, things become complicated when we take contextual information into account, such as time, location, and user mood. Adding this information to the data matrix, we get a tensor $\mathcal{X} \in \mathbb{R}^{N \times M \times K_1 \times K_2 \times \cdots \times K_D}$, where each dimension corresponds to user, item or one contextual factor, $D$ is contextual dimension, i.e., the number of contextual factors, and $K_i$ is the number of possible values of the $i$-th contextual factor. Then, context-aware recommendation becomes a missing value estimation problem where the missing entry $\mathcal{X}_{nmk_1k_2\ldots k_D}$ corresponds to the preferences of user $n$ to item $m$ when the $i$-th contextual factor has the value $k_i$, for each $i \in [1, D]$.

This kind of tensor completion problem can be approached by latent factor learning or low-rank tensor completion. Given a fixed dimension $R$, *latent factor learning* finds an $R$-dimensional vector for each user, item, and context, such that interactions between vectors represent their counterparts in the real world. The learning process can usually be realized by some simple gradient descent algorithms [14]. On the other hand, *low-rank tensor completion* directly fills in the missing values without explicitly computing latent factors. Algorithms for low-rank tensor completion usually involve updating the entire tensor in each interaction, which makes them unsuitable for large data. However, taking the whole tensor into consideration provides a possibility to incorporate the notion of rank into the optimization objective. That is, the rank $R$, which plays a similar role as latent factor dimension, is determined by the algorithm adaptively. In addition, treating the whole tensor at once implies that we can make use of the algebraic structure of tensor algebra to theoretically study the quality of completion results [18]. In this paper, we focus only on low-rank tensor completion.

There are two most widely-used models for tensor completion, canonical polyadic (CP) decomposition and Tucker decomposition; both are extensions of matrix singular value decomposition (SVD). CP decomposition decomposes a tensor into a linear combination of rank-1 tensors. Tucker decomposition, a.k.a. higher-order singular value decomposition (HOSVD), is a more general model. It captures the multilinear nature of tensors and decomposes them into a core tensor and many factor matrices [16].

Both decompositions are proven to be effective in CARS [8, 19], but their performance might be limited by the underlying tensor representation. As every dimension of a tensor is processed in the same way, the model would treat users, items and contextual factors evenly in CARS. This assumption has been challenged in previous researches [10, 14], but no work has been done to devise a new matrix/tensor completion method that is free of this assumption.

We propose to address this issue by using an algebraic structure called the finite abelian group algebra (AGA) $\mathbb{R}[G]$, where $G = \mathbb{Z}_{K_1} \times \mathbb{Z}_{K_2} \times \cdots \times \mathbb{Z}_{K_D}$. With AGA, we can think of a $(D+2)$-dimensional tensor as a matrix over AGA, $\boldsymbol{X} \in \mathbb{R}[G]^{N \times M}$. This way, all extended matrix operations and factorizations are intrinsically


*CIKM'17 , Singapore, Singapore*

DOI: 10.1145/3132847.3133057

context-aware, allowing us to model the context-dependent interactions between users and items directly over a user-item matrix of AGA numbers (i.e. $\boldsymbol{X}_{nm} \in \mathbb{R}[G]$). We exploit the mathematical properties of AGA to derive a factorization method that resembles matrix SVD, and propose an algorithm to solve the resulting convex optimization problem for CARS. Another advantage of AGA is that it uses the multidimensional circular convolutional multiplication, instead of ordinary matrix-vector product, to capture the complicated interaction between all the contexts. A number of experiments are conducted to confirm the validity of AGA on two public datasets for CARS. To our best knowledge, we are the first to use AGA in building recommender systems.

In what follows, we first give some background knowledge of AGA. Then, we introduce the formulation of matrix completion over AGA and the algorithm. We finally report and discuss the experiment results.

## 2 FINITE ABELIAN GROUP ALGEBRA

To simplify the discussion below, we will use circular indexing for tensors and numbers in group algebras, that is, $x_j = x_{j+J}$ where $J$ is the range of the index $j$.

Finite abelian group algebra (AGA), denoted by $\mathbb{R}[G]$, is a generalization of real numbers in an arbitrarily high dimension. When $G = \mathbb{Z}_2$, $\mathbb{R}[G]$ is known as split-complex or hyperbolic numbers. In general, $G$ can be written as $\mathbb{Z}_{K_1} \times \mathbb{Z}_{K_2} \times \cdots \times \mathbb{Z}_{K_D}$ according to the fundamental theorem of finite abelian groups.

Addition in $\mathbb{R}[G]$ is the same as ordinary tensor addition,

$$(\boldsymbol{x} + \boldsymbol{y})_{k_1 k_2 \ldots k_D} = \boldsymbol{x}_{k_1 k_2 \ldots k_D} + \boldsymbol{y}_{k_1 k_2 \ldots k_D} \,, \tag{1}$$

and the additive identity is the zero tensor. The natural multiplication on AGA is the multidimensional circular convolution

$$(\boldsymbol{x}\boldsymbol{y})_{k_1 k_2 \ldots k_D} = \sum_{l_1=1}^{K_1} \cdots \sum_{l_D=1}^{K_D} \boldsymbol{x}_{l_1 \ldots l_D} \boldsymbol{y}_{(k_1 - l_1) \ldots (k_D - l_D)} \,, \tag{2}$$

which is associative, commutative and distributes over addition. The multiplicative identity $\mathbf{1}$ is the tensor with all components equal to 0 except for $\mathbf{1}_{K_1 K_2 \ldots K_D} = 1$. Conjugation is $\overline{\boldsymbol{x}}_{k_1 k_2 \ldots k_D} = \boldsymbol{x}_{(K_1 - k_1)(K_2 - k_2) \ldots (K_D - k_D)}$ and absolute value is

$$|\boldsymbol{x}| = \sqrt{\sum_{k_1=1}^{K_1} \cdots \sum_{k_D=1}^{K_D} \left(\boldsymbol{x}_{k_1 k_2 \ldots k_D}\right)^2} \,. \tag{3}$$

While the computations of addition, conjugation and absolute value in AGA are efficient, the direct computation of multiplication can be slow. It can be sped up by using the convolution theorem:

$$\mathcal{F}(\boldsymbol{x}\boldsymbol{y}) = \mathcal{F}(\boldsymbol{x}) \circ \mathcal{F}(\boldsymbol{y}) \,, \tag{4}$$

where $\mathcal{F}(\cdot)$ is the multidimensional discrete Fourier transform (MDDFT) and $\circ$ is the component-wise product. We call this the "Fourier trick."

Let $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}[G]^{N \times M}$ be matrices with elements belonging to AGA. With the knowledge of $\mathbb{R}[G]$, we can directly extend basic matrix operations to $\mathbb{R}[G]^{N \times M}$: addition and multiplication are straightforward, $(\boldsymbol{X} + \boldsymbol{Y})_{nm} = \boldsymbol{X}_{nm} + \boldsymbol{Y}_{nm}$, $(\boldsymbol{X}\boldsymbol{Y})_{nm} = \sum_{i=1}^{r} \boldsymbol{X}_{nr} \boldsymbol{Y}_{rm}$, and conjugate transpose is similar except that conjugation is replaced by the above definition to yield $(\boldsymbol{X}^*)_{nm} = \overline{\boldsymbol{X}_{mn}}$. Other definitions such as identity matrix, diagonal matrix and unitary matrix can also be extended accordingly.

The multiplication of matrices over AGA can be sped up and simplified by the Fourier trick. We perform MDDFT on each entry $\boldsymbol{X}_{nm}$ and represent the result $\mathcal{F}(\boldsymbol{X})$ as an order-$(D+2)$ tensor, i.e. $\mathcal{F}(\boldsymbol{X})_{nmk_1 k_2 \ldots k_D} = \mathcal{F}(\boldsymbol{X}_{nm})_{k_1 k_2 \ldots k_D}$. The multiplication of $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}[G]^{N \times M}$ is equivalent to ordinary matrix multiplication on each frontal slice of $\mathcal{F}(\boldsymbol{X})$ and $\mathcal{F}(\boldsymbol{Y})$, and then applying the inverse MDDFT. Here, a frontal slice of $\mathcal{F}(\boldsymbol{X})$ is a matrix defined by fixing the last $D$ indices $\mathcal{F}(\boldsymbol{X})_{:,:,k_1 k_2 \ldots k_D}$.

## 3 PROPOSED METHOD

### 3.1 Context-Aware Recommendation

We use $X \in \mathbb{R}^{N \times M}$ to represent the relations between $N$ users and $M$ items. The matrix $X$ is usually sparse and the non-zero entries correspond to observed data in our training set. It can be either users' rating on items for explicit feedback, or play counts/click counts for implicit feedback. The prediction for the unobserved entries can be formulated as the following low-rank matrix completion problem:

$$\underset{\widehat{X}}{\arg\min} \left\| \widehat{X} \right\|_*, \text{ s.t. } \pi\left(X - \widehat{X}\right) = 0 \,, \tag{5}$$

where $\|\cdot\|_*$ is the trace norm, the sum of all singular values. Minimizing the trace norm has been widely used as a convex surrogate for minimizing the rank of a matrix [9]. The operator $\pi(\cdot)$ sets unobserved entries to 0 and keeps the others unchanged, ensuring that the estimated $\widehat{X} \in \mathbb{R}^{N \times M}$ and $X$ agree on the observed entries.

Context-awareness can be achieved by extending Eq. (5), providing appropriate definition of the trace norm. For example, if we consider only one contextual factor, we can use an order-3 tensor or a polar $n$-complex matrix for $\mathcal{X} \in \mathbb{R}^{N \times M \times K_1}$. By defining the trace norm as the sum of the singular values of all frontal slices, a.k.a. the tensor nuclear norm, the tensor completion problem can be solved by the t-SVD algorithm [5, 18].

To enable the use of an arbitrary number of contextual factors in general, we propose a new formulation with $X$ being a matrix over AGA. We need to come up with a definition of the trace norm for a matrix over AGA, for this has not been attempted before. We first define the SVD of $\boldsymbol{X} \in \mathbb{R}[G]^{N \times M}$ as

$$\boldsymbol{X} = \boldsymbol{U} \Sigma \boldsymbol{V}^* \,, \tag{6}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary and $\Sigma$ is diagonal. Using the Fourier trick, we propose a simple way to compute Eq. (6): perform the ordinary SVD on each frontal slice in the Fourier domain $\mathcal{F}(\boldsymbol{X})_{:,:,k_1 k_2 \ldots k_D}$ using MDDFT, and then transform the result back to the original domain. The singular values of $\boldsymbol{X}$ are the diagonal entries $\boldsymbol{\sigma}_i \in \mathbb{R}[G]$ of $\Sigma$. Unlike real and complex matrices, the singular values of matrices over AGA lie in AGA and may not be real numbers in general. We therefore propose to define the trace norm as the sum of the absolute values of the singular values:

$$\|\boldsymbol{X}\|_* = \sum_i |\boldsymbol{\sigma}_i| \,, \tag{7}$$

leading to the following optimization problem for CARS:

$$\underset{\widehat{\boldsymbol{X}}}{\arg\min} \left\| \widehat{\boldsymbol{X}} \right\|_*, \text{ s.t. } \pi(\boldsymbol{X} - \widehat{\boldsymbol{X}}) = 0 \,, \tag{8}$$

where $X, \widehat{X} \in \mathbb{R}[G]^{N\times M}$. The predicted preference of user $n$ for item $m$ under the context $(k_1, k_2, \ldots, k_D)$ is $\left(\widehat{X}_{nm}\right)_{k_1 k_2 \ldots k_D}$.

## 3.2 Algorithm: MC-AGA

**Algorithm 1** Matrix Completion over Finite Abelian Group Algebra (MC-AGA)

**Input:** $X \in \mathbb{R}[G]^{N\times M}$, $G = \mathbb{Z}_{K_1} \times \mathbb{Z}_{K_2} \times \cdots \times \mathbb{Z}_{K_D}$
**Output:** $\widehat{X}$

1: Initialize $Y$, $\Lambda$, $\mu$
2: **while** not converged **do**
3: $\quad \widehat{X} \leftarrow \arg\min_{\widehat{X}} \left\|\widehat{X}\right\|_* + \frac{\mu}{2} \left\|\frac{1}{\mu}\Lambda + X - \widehat{X} - Y\right\|_F^2$
4: $\quad Y \leftarrow \arg\min_Y \left\|\frac{1}{\mu}\Lambda + X - \widehat{X} - Y\right\|_F^2$, s.t. $\pi(Y) = 0$
5: $\quad \Lambda \leftarrow \Lambda + \mu\left(X - \widehat{X} - Y\right)$
6: $\quad$ Update $\mu$
7: **end while**

We use the inexact augmented Lagrange multiplier (IALM) method to solve the minimization problem (8). IALM was first applied in the context of low-rank matrix completion via trace norm minimization by Lin *et al.* [9], and was recently adapted to fit into the framework of polar $n$-complex matrix principal component pursuit [5]. In order to apply IALM, we first rewrite the formula (8) as

$$\arg\min_{\widehat{X}} \left\|\widehat{X}\right\|_*, \text{ s.t. } X = \widehat{X} + Y,\ \pi(Y) = 0\,, \tag{9}$$

from which we derive MC-AGA (Algorithm 1) using IALM. Here the Frobenius norm $\|X\|_F^2$ is defined as $\sum_{n,m} |X_{nm}|^2$, and $\mu > 0$ is a penalty parameter for the equality constraint. Explicit solutions to the optimization problems in lines 3 and 4 are given by the following theorems. Due to space limit, we will provide the proofs as online supplementary material, along with the MATLAB reference code (`http://github.com/versesrev/MC-AGA`).

Theorem 1. *The solution to line 3 in Algorithm 1 is*

$$\widehat{X} = U \cdot \mathcal{S}(\Sigma) \cdot V^*\,, \tag{10}$$

*where* $U\Sigma V^*$ *is the SVD of the matrix over AGA,* $\frac{1}{\mu}\Lambda + X - Y$, *and* $\mathcal{S}(\Sigma)_{nm} = \left(1 - \frac{\mu^{-1}}{|\Sigma_{nm}|}\right)_+ \Sigma_{nm}$ *is the soft thresholding operator,* $(x)_+ = \max(x, 0)$ *[9].*

Theorem 2. *The solution to line 4 in Algorithm 1 is*

$$Y = \frac{1}{\mu}\Lambda + X - \widehat{X} - \pi\left(\frac{1}{\mu}\Lambda + X - \widehat{X}\right)\,. \tag{11}$$

Next, we analyze the time complexity of MC-AGA. According to Theorem 1, the update of $\widehat{X}$ in line 3 requires one SVD over AGA. Such SVD consists of three steps: MDDFT, ordinary SVD on each frontal slice, and inverse MDDFT. The complexity of MDDFT and inverse MDDFT is $O(NM\,|G| \log |G|)$, and the complexity of SVD on all frontal slices is $O\left(|G| \times \min\left(N^2M, NM^2\right)\right)$. $|G| = K_1 K_2 \ldots K_D$ is the total number of different contexts and $\min\left(N^2M, NM^2\right)$ is the cost of an ordinary SVD. The other updates, including line 4, can all be computed in linear time. Therefore, the total complexity is $O(NM\,|G|\,(\min(N, M) + \log|G|))$. This relates only to the size of the matrix and is independent of the sparsity of the data.

The algorithm can be sped up in many ways. In the real-life setting, usually an approximation with very low rank $R$ is sufficient to obtain a comparable result. Therefore, instead of a full SVD, we can do a truncated SVD, whose complexity is only $O(NM\log(R) + (N+M)R^2)$ [6]. The other bottleneck, Fourier transform, is hard to improve directly, but we can further use a Monte-Carlo algorithm to avoid Fourier transform of a full tensor [17].

**Table 1: Dataset statistics. 'Cx' contextual dimension (i.e. $D$); 'Obs' number of observed entries.**

| Name | Type | Users | Items | Cx. | Obs. |
|---|---|---|---|---|---|
| Frappe [4] | implicit | 953 | 4,073 | 1 | 95,889 |
| Sushi [7] | explicit | 5,000 | 100 | 6 | 50,000 |

# 4 EXPERIMENTS

We evaluate the performance of the proposed method on the two datasets listed in Table 1. **Frappe** [4] is an dataset that contains 95,889 implicit feedback for 4,073 smart phone Apps by 953 users in a variety of 42 possible contexts, including different locations, day times, and day types (weekday or weekend). Most existing work [14] represents these contexts by one contextual dimension and accordingly treats the data as an order-3 tensor, with $D = 1$ and $K_1 = 42$. **Sushi** [7] is an explicit feedback dataset, comprising 50,000 ratings of 100 types of sushi by 5,000 users, along with 6 attributes of sushi, including style, group, oiliness, popularity, price, and availability. Following [11], we treat these attributes as different contextual factors. As the last four factors are numerical, we quantize them into binary values (high or low) by using the midpoint of their scales as the thresholds.

For each dataset, we randomly pick 75%, 5% and 20% of the data as the training, validation, and test set, respectively. For Frappe, we first generate a recommendation list based on the predicted preference, and then evaluate the result in terms of mean average precision (MAP) and precision, recall at different ranks. For Sushi, we use mean absolute error (MAE) and root mean square error (RMSE) as the metrics. For normalization, we shift all the ratings of Sushi by making the average rating in the training set zero.

We compare the performance of the proposed **MC-AGA** algorithm against the following baseline methods.

- **Pop**ularity-based method is a non-personalized, naïve method that generates recommendation lists based on the item popularity observed in the training set, measured by the number of users interacted with the item.
- Bayesian personalized ranking (**BPR**) [12] is a widely-used learning-to-rank based method to generate context-independent recommendation for implicit data.
- Tensor factorization for mean average precision (**TFMAP**) [15] is a learning-to-rank based method that directly optimizes the MAP criterion for CARS.
- **CARS$^2$-P**air [14] is a factorization-based method that was recently shown to outperform various methods for CARS, including TFMAP and factorization machines (FM) [13]. It uses a pairwise ranking loss function to directly optimize the context-aware representations to be learned.

- **MC-Split** is a pre-filtering method that splits data into different categories based on their contextual information and perform recommendation using low-rank matrix completion independently for each category. For example, 42 models are trained for Frappe, one for each context.
- **Const**: A simple RS that gives constant prediction. The constant is set as the average rating of the training data.
- Tensor completion (**CP-TC**) based on CP decomposition [1], implemented in the MATLAB Tensor Toolbox [3]. The tensor rank is tuned using the validation set.

Table 2(a) presents the results on Frappe, excluding the CP-based method for it did not work well for this implicit feedback dataset. We see that MC-AGA obtains the best results in all the five performance metrics. The performance comparison between MC-AGA and MC-Split shows the importance of jointly modeling different contexts. The comparison between MC-AGA and CARS$^2$-P further suggests the advantages of using circular convolution product instead of matrix-vector product for contextual modeling.

Table 2(b) shows the results on Sushi. MC-AGA still outperforms MC-Split, as expected. More importantly, we see that MC-AGA performs better than the tensor completion method, supporting the argument that AGA provides a better algebraic structure in modeling contextual information.

As for the execution time, MC-AGA is the longest among all methods tested on Sushi dataset. Performance profiling indicates that half of the time is taken by ordinary SVD, and that MDDFT and inverse MDDFT almost use up the other half. This result is consistent to the time complexity discussed in Section 3.2.

## 5 CONCLUSION

In this paper, we have proposed the idea of modeling the relations between users, items and contexts in CARS using an algebraic structure called the finite abelian group algebra (AGA), which is seldom used in the literature of recommender systems. This algebraic structure enables modeling context-aware interactions between users and items directly over a user-item matrix. We have also proposed a new algorithm that achieves context-aware recommendation by minimizing the trace norm of a matrix over AGA. Our experiments on two datasets showed that the proposed algorithm outperforms a number of existing learning-to-rank or tensor completion based methods in various performance metrics. For future work, we like to test the robustness of the algorithm to noisy feedback in the training data, and to investigate other algebraic structures, such as non-commutative algebras, for recommendation problems. We also want to improve the efficiency of MC-AGA by adopting approximate and probabilistic algorithms. Another possible direction is to integrate AGA with latent factor learning, whose complexity depends only on the amount of observed data, which is usually much smaller than the size of the data matrix.

**Table 2: Performance comparison of different methods. For (a) Frappe, higher values are better; for (b) Sushi, lower values are better. Bold font indicates the best results. Asterisks indicate that MC-AGA has a higher mean than the other methods on that metric (p-value < 0.01) — we use non-paired t-test for Frappe as we only have aggregate results for BPR, TFMAP, and CARS$^2$-P; for Sushi, we use the paired t-test.**

(a) Frappe (implicit dataset)

| Methods | MAP | P@5 | R@5 | P@10 | R@10 |
|---|---|---|---|---|---|
| Pop | 0.076 | 0.043 | 0.121 | 0.028 | 0.153 |
| BPR [12] | 0.121 | 0.061 | 0.160 | 0.037 | 0.184 |
| TFMAP [15] | 0.127 | 0.061 | 0.159 | 0.039 | 0.195 |
| CARS$^2$-P [14] | 0.140 | 0.068 | 0.183 | 0.044 | 0.226 |
| MC-Split | 0.097 | 0.057 | 0.128 | 0.040 | 0.165 |
| MC-AGA | **0.155*** | **0.089*** | **0.199*** | **0.057*** | **0.241*** |

(b) Sushi (explicit dataset)

| Methods | MAE | RMSE | Time (mins.) |
|---|---|---|---|
| Const | 1.073 | 1.257 | < 1 |
| CP-TC [1] | 0.994 | 1.279 | 2 |
| MC-Split | 1.052 | 1.241 | 19 |
| MC-AGA | **0.950*** | **1.147*** | 67 |

## REFERENCES


[1] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup. 2011. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems* 106, 1 (Mar. 2011), 41–56.
[2] G. Adomavicius and A. Tuzhilin. 2015. Context-aware recommender systems. In *Recommender Systems Handbook*. Springer, 191–226.
[3] B. W. Bader, T. G. Kolda, and others. 2015. MATLAB Tensor Toolbox Version 2.6. (2015). http://www.sandia.gov/~tgkolda/TensorToolbox/
[4] L. Baltrunas, K. Church, A. Karatzoglou, and N. Oliver. 2015. Frappe: Understanding the usage and perception of mobile app recommendations in-the-wild. *arXiv:1505.03014* (Jun. 2015).
[5] T.-S. T. Chan and Y.-H. Yang. 2016. Polar $n$-complex and $n$-bicomplex singular value decomposition and principal component pursuit. *IEEE Transactions on Signal Processing* 64, 24 (Dec. 2016), 6533–6544.
[6] N. Halko, P.-G. Martinsson, and J. A. Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review* 53, 2 (May 2011), 217–288.
[7] T. Kamishima. 2003. Nantonac collaborative filtering: Recommendation based on order responses. In *Proceedings of SIGKDD*. 583–588.
[8] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. 2010. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of RecSys*. 79–86.
[9] Z. Lin, M. Chen, and Y. Ma. 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv:1009.5055* (Sep. 2010).
[10] X. Liu and W. Wu. 2015. Learning context-aware latent representations for context-aware collaborative filtering. In *Proceedings of SIGIR*. 887–890.
[11] T. V. Nguyen, A. Karatzoglou, and L. Baltrunas. 2014. Gaussian process factorization machines for context-aware recommendations. In *Proceedings of SIGIR*. 63–72.
[12] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of UAI*. 452–461.
[13] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. 2011. Fast context-aware recommendations with factorization machines. In *Proceedings of SIGIR*. 635–644.
[14] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, and A. Hanjalic. 2014. CARS$^2$: Learning context-aware representations for context-aware recommendations. In *Proceedings of CIKM*. 291–300.
[15] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver. 2012. TFMAP: Optimizing MAP for top-N context-aware recommendation. In *Proceedings of SIGIR*. 155–164.
[16] A. Sobral and E.-h. Zahzah. 2016. Matrix and tensor completion algorithms for background model initialization: A comparative evaluation. *Pattern Recognition Letters* (Dec. 2016). https://doi.org/10.1016/j.patrec.2016.12.019
[17] D. A. Tarzanagh and G. Michailidis. 2017. Fast Monte Carlo algorithms for tensor operations. *arXiv:1704.04362* (Apr. 2017).
[18] Z. Zhang and S. Aeron. 2017. Exact tensor completion using t-SVD. *IEEE Transactions on Signal Processing* 65, 6 (Mar. 2017), 1511–1526.
[19] C. Zheng, H. E, M. Song, and J. Song. 2016. CMPTF: Contextual modeling probabilistic tensor factorization for recommender systems. *Neurocomputing* 205 (Sep. 2016), 141–151.

## APPENDIX

In this appendix we will solve $\arg\min_{\boldsymbol{X}} \lambda \|\boldsymbol{X}\|_* + \frac{1}{2}\|\boldsymbol{Z}-\boldsymbol{X}\|_F^2$. To derive the solution, we need a definition and some properties of inner product on $\mathbb{R}[G]^{N\times M}$.

DEFINITION 1. $\langle \boldsymbol{X}, \boldsymbol{Z}\rangle := \operatorname{tr}(\boldsymbol{X}\boldsymbol{Z}^*)_{K_1K_2\ldots K_D}$ *, i.e. the "real part" of the trace of* $\boldsymbol{X}\boldsymbol{Z}^*$.

PROPOSITION 1. $\|\boldsymbol{X}\|_F^2 = \langle \boldsymbol{X}, \boldsymbol{X}\rangle$ .

PROOF.

$$\begin{aligned}\langle \boldsymbol{X}, \boldsymbol{X}\rangle &= \sum_{n,m} \left(\boldsymbol{X}_{nm}\overline{\boldsymbol{X}}_{nm}\right)_{K_1K_2\ldots K_D} \\ &= \sum_{n,m} |\boldsymbol{X}_{nm}|^2 = \|X\|_F^2\end{aligned}$$

□

PROPOSITION 2. $\langle \boldsymbol{X}, \boldsymbol{X}\rangle = \langle \boldsymbol{\Delta}, \boldsymbol{\Delta}\rangle$ *, where* $\boldsymbol{X} = \boldsymbol{P}\boldsymbol{\Delta}\boldsymbol{Q}^*$ *is the SVD in AGA.*

PROOF.

$$\begin{aligned}\langle \boldsymbol{X}, \boldsymbol{X}\rangle &= \langle \boldsymbol{P}\boldsymbol{\Delta}\boldsymbol{Q}^*, \boldsymbol{P}\boldsymbol{\Delta}\boldsymbol{Q}^*\rangle \\ &= \operatorname{tr}(\boldsymbol{P}\boldsymbol{\Delta}\boldsymbol{Q}^*\boldsymbol{Q}\boldsymbol{\Delta}^*\boldsymbol{P}^*)_{K_1K_2\ldots K_D} \\ &= \operatorname{tr}(\boldsymbol{\Delta}\boldsymbol{\Delta}^*)_{K_1K_2\ldots K_D} = \langle \boldsymbol{\Delta}, \boldsymbol{\Delta}\rangle\end{aligned}$$

□

LEMMA 1. $\langle \boldsymbol{X}, \boldsymbol{Z}\rangle \le \langle \boldsymbol{\Delta}, \boldsymbol{\Sigma}\rangle$ *, where* $\boldsymbol{X} = \boldsymbol{P}\boldsymbol{\Delta}\boldsymbol{Q}^*$ *and* $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ *are the SVD in AGA.*

PROOF. We enumerate the components of $\mathcal{F}(\boldsymbol{x})$ and the frontal slices of $\mathcal{F}(\boldsymbol{X})$, and denote the $k$-th one as $\mathcal{F}(\boldsymbol{x})_k$ and $\mathcal{F}(\boldsymbol{X})_{:,:,k}$. Then,

$$\begin{aligned}\mathcal{F}(\operatorname{tr}(\boldsymbol{X}\boldsymbol{Z}^*))_k &= \operatorname{tr}\left(\mathcal{F}\left(\boldsymbol{X}\boldsymbol{Z}^*\right)_{:,:,k}\right) \\ &= \operatorname{tr}\left(\mathcal{F}\left(\boldsymbol{X}\right)_{:,:,k}\mathcal{F}\left(\boldsymbol{Z}^*\right)_{:,:,k}\right) \ .\end{aligned} \tag{1}$$

We expand and rewrite the inner product in the Fourier domain, using properties of Fourier transform on the second line and (1) on the last line.

$$\begin{aligned}\langle \boldsymbol{X}, \boldsymbol{Z}\rangle &= \operatorname{tr}(\boldsymbol{X}\boldsymbol{Z}^*)_{K_1K_2\ldots K_D} \\ &= \frac{1}{G}\sum_k \Re\left(\mathcal{F}\left(\operatorname{tr}(\boldsymbol{X}\boldsymbol{Z}^*)\right)_k\right) \\ &= \frac{1}{G}\sum_k \Re\left(\operatorname{tr}\left(\mathcal{F}\left(\boldsymbol{X}\right)_{:,:,k}\mathcal{F}\left(\boldsymbol{Z}^*\right)_{:,:,k}\right)\right)\end{aligned} \tag{2}$$

Our last step is to observe that

$$\left|\operatorname{tr}\left(\mathcal{F}\left(\boldsymbol{X}\right)_{:,:,k}\mathcal{F}\left(\boldsymbol{Z}^*\right)_{:,:,k}\right)\right| \le \operatorname{tr}\left(\mathcal{F}\left(\boldsymbol{\Delta}\right)_{:,:,k}\mathcal{F}\left(\boldsymbol{\Sigma}^*\right)_{:,:,k}\right) , \tag{3}$$

which follows from von Neumann's trace inequality and the definition of SVD over AGA. The lemma is proved by combining (2) and (3), using the facts that the real part is always not greater than the absolute value of a complex number and that $\mathcal{F}(\boldsymbol{\Delta}), \mathcal{F}(\boldsymbol{\Sigma})$ have non-negative entries. □

LEMMA 2. $\|\boldsymbol{X}-\boldsymbol{Z}\|_F^2 \ge \|\boldsymbol{\Delta}-\boldsymbol{\Sigma}\|_F^2$ *, where* $\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Delta}, \boldsymbol{\Sigma}$ *are as in Lemma 1.*

PROOF.

$$\begin{aligned}\|\boldsymbol{X}-\boldsymbol{Z}\|_F^2 &= \langle \boldsymbol{X}-\boldsymbol{Z}, \boldsymbol{X}-\boldsymbol{Z}\rangle \\ &= \langle \boldsymbol{X}, \boldsymbol{X}\rangle - 2\langle \boldsymbol{X}, \boldsymbol{Z}\rangle + \langle \boldsymbol{Z}, \boldsymbol{Z}\rangle \\ &\ge \langle \boldsymbol{\Delta}, \boldsymbol{\Delta}\rangle - 2\langle \boldsymbol{\Delta}, \boldsymbol{\Sigma}\rangle + \langle \boldsymbol{\Sigma}, \boldsymbol{\Sigma}\rangle \\ &= \|\boldsymbol{\Delta}-\boldsymbol{\Sigma}\|_F^2\end{aligned}$$

□

LEMMA 3. $\left(1-\frac{\lambda}{|\boldsymbol{z}|}\right)_+ \boldsymbol{z} = \arg\min_{\boldsymbol{x}} \lambda|\boldsymbol{x}| + \frac{1}{2}|\boldsymbol{x}-\boldsymbol{z}|^2$*, where* $(x)_+ = \max(x, 0)$.

PROOF. Let $f(\boldsymbol{x}) = \lambda|\boldsymbol{x}| + \frac{1}{2}|\boldsymbol{x}-\boldsymbol{z}|^2$. We think of it as a multivariate function with $N = K_1K_2\ldots K_D$ variables, and we find out the minimum by looking for zeros of its partial derivatives $\frac{\partial f}{\partial x_i} = \lambda\frac{x_i}{|x|} + x_i - z_i$ for all $i \in [1, N]$. This leads to the equation

$$\left(1+\frac{\lambda}{|x|}\right)x = z \ . \tag{4}$$

If $|z| > \lambda$, we have $|x| = |z| - \lambda$ by taking the absolute value on both sides of (4). Then,

$$x = \left(1-\frac{\lambda}{|z|}\right)z.$$

If $|z| \le \lambda$, we prove that $f$ attains its minimum at $x = 0$, i.e.

$$f(0) = \frac{1}{2}|z|^2 \le \lambda|x| + \frac{1}{2}|x-z|^2 = f(x), \text{ for all } x \ .$$

The inequality holds because

$$\begin{aligned}|z|^2 - 2\lambda|x| &\le |z|^2 - 2|z||x| \\ &\le (|x|-|z|)^2 \\ &\le |x-z|^2 ,\end{aligned}$$

where the last line follows from the triangle inequality. Combining the result of both cases, we conclude that $f$ is minimized at $\left(1-\frac{\lambda}{|\boldsymbol{z}|}\right)_+ \boldsymbol{z}$. □

THEOREM 1. *The solution to*

$$\arg\min_{\boldsymbol{X}} \lambda\|\boldsymbol{X}\|_* + \frac{1}{2}\|\boldsymbol{Z}-\boldsymbol{X}\|_F^2 \tag{5}$$

*is*

$$\boldsymbol{X} = \boldsymbol{U}\cdot\mathcal{S}(\boldsymbol{\Sigma})\cdot\boldsymbol{V}^* , \tag{6}$$

*where* $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ *is the SVD of the matrix* $\boldsymbol{Z}$ *over AGA, and* $\mathcal{S}(\boldsymbol{\Sigma})_{nm} = \left(1-\frac{\lambda}{|\boldsymbol{\Sigma}_{nm}|}\right)_+ \boldsymbol{\Sigma}_{nm}$ *is the soft thresholding operator,* $(x)_+ = \max(x, 0)$.

PROOF. Let $\boldsymbol{P}\boldsymbol{\Delta}\boldsymbol{Q}^*$ be the SVD of $\boldsymbol{X}$.

$$\begin{aligned}\min_{\boldsymbol{X}} \lambda\|\boldsymbol{X}\|_* + \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{Z}\|_F^2 &\ge \min_{\boldsymbol{\Delta}} \lambda\|\boldsymbol{\Delta}\|_* + \frac{1}{2}\|\boldsymbol{\Delta}-\boldsymbol{\Sigma}\|_F^2 \\ &= \min_{\boldsymbol{\Delta}} \sum_i \lambda|\boldsymbol{\delta}_i| + \frac{1}{2}|\boldsymbol{\delta}_i-\boldsymbol{\sigma}_i|^2 \\ &= \sum_i \min_{\boldsymbol{\delta}_i} \lambda|\boldsymbol{\delta}_i| + \frac{1}{2}|\boldsymbol{\delta}_i-\boldsymbol{\sigma}_i|^2\end{aligned}$$

When $\boldsymbol{X} = \boldsymbol{U}\cdot\mathcal{S}(\boldsymbol{\Sigma})\cdot\boldsymbol{V}^*$, the inequality above becomes equality and the minimum is achieved for all $i$. This proves the theorem. □

THEOREM 2. *The solution to*

$$\arg\min_{\boldsymbol{X}} \|\boldsymbol{Z} - \boldsymbol{X}\|_F^2 \,, s.t.\; \pi(\boldsymbol{X}) = 0 \tag{7}$$

*is*

$$\boldsymbol{X} = \boldsymbol{Z} - \pi\left(\boldsymbol{Z}\right) . \tag{8}$$

PROOF. This should be easy. ☺