Accelerated Online Low-Rank Tensor Learning for Multivariate Spatio-Temporal Streams

Rose Yu QIYU@USC.EDU
Dehua Cheng DEHUA.CHENG@USC.EDU
Yan Liu YANLIU.CS@USC.EDU
Department of Computer Science, University of Southern California

Abstract

Low-rank tensor learning has many applications in machine learning. A series of batch learning algorithms have achieved great successes. However, in many emerging applications, such as climate data analysis, we are confronted with large-scale tensor streams, which pose significant challenges to existing solutions. In this paper, we propose an accelerated online low-rank tensor learning algorithm (ALTO) to solve the problem. At each iteration, we project the current tensor to a low-dimensional tensor, using the information of the previous low-rank tensor, in order to perform efficient tensor decomposition, and then recover the low-rank approximation of the current tensor. By randomly selecting additional subspaces, we successfully overcome the issue of local optima at an extremely low computational cost. We evaluate our method on two tasks in online multivariate spatio-temporal analysis: online forecasting and multi-model ensemble. Experiment results show that our method achieves comparable predictive accuracy with significant speed-up.

1. Introduction

Low-rank tensor learning enjoys a broad range of applications in practical machine learning problems (Kolda & Bader, 2009), ranging from signal processing, computer vision, to neuroscience. One classical example is learning a low-rank tensor for multivariate regression, for which a series of effective batch learning algorithms have been developed (De Lathauwer et al., 2000; Guo et al., 2012; Zhou et al., 2013; Bahadori et al., 2014). We notice that in many emerging applications, large-scale tensor data come in streams, such as the spatio-temporal climate observations in climate data analysis. Batch learning algorithms would suffer from computational bottleneck, especially facing the challenge of short response time. Therefore, effective and fast online learning algorithms are a must for enabling real-time large-scale tensor analysis.

Online learning of low-rank tensors aims to dynamically update a tensor while preserving the low-rank structure. While online low-rank matrix learning has been intensively studied, e.g. (Brand, 2002; Meka et al., 2008; Shalit et al., 2010), online tensor learning remains underexplored. The problem is extremely challenging due to the inherent complexity of tensor analysis (Hillar & Lim, 2013). Local solutions (Sun et al., 2008) have achieved wide success in real applications but lack rigorous theoretical understanding. For certain rank structures, we can use the nuclear norm as a convex surrogate for the rank constraint and solve the problem with off-the-shelf online low-rank matrix algorithms (Avron et al., 2012; Ouyang et al., 2013). Nevertheless, it is known that optimizing over convex surrogate loss may lead to sub-optimal solutions (Zhang et al., 2013). Moreover, solving an optimization problem with nuclear norm regularization itself is computationally expensive.

In this paper, we develop a novel framework, namely the **A**ccelerated **L**ow-rank **T**ensor **O**nline Learning (ALTO) algorithm to address the problem. Our solution follows a two-step procedure: we first solve an unconstrained tensor learning problem, and then adjust the solution tensor to satisfy the low-rank constraint. ALTO significantly accelerates the online learning process by keeping track of the low-rank components of the solution obtained at each iteration: It performs a dimension reduction of the tensor using previous low-rank components and tensor matrix multiplications, in order to avoid the expensive operations of singular value decomposition (SVD) on unfolded matrices of the full tensor. In addition, it employs randomization techniques to select additional dimensions so as to overcome the issue of local optima in existing incremental

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

tensor learning algorithms (Sun et al., 2008). Theoretical analysis shows that our randomization technique can significantly reduce the noise at a cost of very minor biases. As a side outcome, we also observe an interesting property: despite being non-convex, the low-rank space usually behaves like a convex set in its neighborhood.

We demonstrate the effectiveness of our algorithm via two machine learning tasks in spatio-temporal stream analysis: one is the classical forecasting problem (i.e., performing n-step ahead prediction from historical observations), and the other is the multi-model ensemble problem, a fundamental task in climatology to make predictions by combining the forecasting results from multiple simulation models. In these applications, the data often exhibit unique properties, such as spatial proximity, temporal periodicity and variable correlations, which can be captured naturally via the low-rank constraint. We conduct experiments on both synthetic and real-world application datasets, including a Foursquare check-in dataset, a daily weather dataset and a climate ensemble dataset. Experimental results show that our algorithm can achieve competitive prediction accuracy with significant speed-up. In addition, the low-rank tensor parameters learned by our algorithm from the climate dataset provide interesting insights into the underlying relationships between simulations models and physical processes.

2. Online Low-Rank Tensor Learning

2.1. Preliminaries

Across the paper, we use calligraphy font for tensors, such as $\mathcal{X}, \mathcal{Y}$, bold uppercase letters for matrices, such as $\mathbf{A}, \mathbf{B}$, and bold lowercase letters for vectors, such as $\mathbf{x}, \mathbf{y}$.

Rank-R Projection For any matrix $\mathbf{M}$, let $p\left(\mathbf{M}, R\right)$ be the projection of $\mathbf{M}$ to the top-R spectral spaces. It can be calculated using top-R truncated SVD: $\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$, $p\left(\mathbf{M}, R\right) = \mathbf{U}_R\boldsymbol{\Sigma}_R\mathbf{V}_R^\top$. The rank R might be omitted when the context is clear.

Tensor Unfolding Each dimension of a tensor is a mode. An n-mode unfolding of a tensor $\mathcal{A}$ along mode i transforms a tensor into a matrix $\mathcal{A}_{(i)}$ by treating i as the first mode of the matrix and cyclically concatenating other modes. The indexing follows the convention in (Kolda & Bader, 2009). It is also known as tensor matricization.

N-Mode Product The n-mode product between tensor $\mathcal{A}$ and matrix $\mathbf{U}$ on mode i is represented as $\mathcal{A} \times_i \mathbf{U}$ and is defined as $\left(\mathcal{A} \times_i \mathbf{U}\right)_{(i)} = \mathbf{U}\mathcal{A}_{(i)}$.

Tucker Decomposition Tucker decomposition factorizes a tensor $\mathcal{A}$ into $\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}_1 \cdots \times_n \mathbf{U}_n$, where $\{\mathbf{U}_n\}$ are all unitary matrices and the core tensor satisfies that $\mathcal{S}_{(i)}$ is row-wise orthogonal for all $i = 1, 2, \ldots, n$. Tucker decomposition can be computed by SVD on all possible unfolded matrices of a tensor. It is also known as *high order singular value decomposition* (HOSVD) (De Lathauwer et al., 2000).

2.2. General Framework

Consider the following low-rank tensor learning problem for regression with predictor tensor $\mathcal{Z} \in \mathbb{R}^{Q\times T\times M}$, and response tensor $\mathcal{X} \in \mathbb{R}^{P\times T\times M}$. Our goal is to learn a model parameter tensor $\mathcal{W} \in \mathbb{R}^{P\times Q\times M}$ whose rank is upper bounded by R.

$$\begin{aligned}\widehat{\mathcal{W}} = \operatorname{argmin}_{\mathcal{W}} \left\{\textstyle\sum_{t,m} \|\mathcal{W}_{:,:,m}\mathcal{Z}_{:,t,m} - \mathcal{X}_{:,t,m}\|_F^2\right\} \\ \text{s.t.} \qquad \operatorname{rank}(\mathcal{W}) \le R, \end{aligned} \tag{1}$$

where a single : is used to index entire rows or columns. Different from matrices, there are several meaningful ways to define the rank of a tensor (Kolda & Bader, 2009). The matrix rank of the mode-n unfolding $\operatorname{rank}(\mathcal{W}_{(n)})$ is called the n-rank of tensor $\mathcal{W}$. And the summation of n-rank $\sum_{n=1}^N \operatorname{rank}(\mathcal{W}_{(n)})$, namely the sum-$n$-rank of a tensor $\mathcal{W}$, is commonly applied because it is easy to compute (Hillar & Lim, 2013; De Lathauwer et al., 2000) than others (e.g., CP rank).

Bahadori et al. (2014) provides a greedy algorithm for computing the solution in the batch setting. It can also be solved by replacing the rank constraint with the nuclear norm constraint. However, optimizing over this convex surrogate loss may lead to sub-optimal solutions and is computationally expensive (Zhang et al., 2013). Therefore, we propose the **A**ccelerated **L**ow-rank **T**ensor **O**nline Learning (ALTO) algorithm, which solves the problem via a simple two-step approach: (1) solving the unconstrained optimization problem given the new data, (2) updating the solution with the low-rank constraint, i.e. projecting the solution to the space of low-rank tensors. As we will show later through theoretical analysis that while the first step yields approximately low-rank tensor, this two-step approach is *nearly optimal*. It is also computationally efficient since the unconstrained optimization problem has a closed form solution, which is preferable since in online settings, where the data stream arrives in mini-batches, we need to dynamically update the model tensor while preserving its low-rank structures.

2.3. Tensor Stream in Online Setting

In Step 1, the quadratic loss function in Equation 1 is equivalent to $\sum_{m=1}^M \|\mathcal{W}_{:,:,m}\mathcal{Z}_{:,:,m} - \mathcal{X}_{:,:,m}\|_F^2$ for $m = 1, 2, \cdots, M$, which is an ordinary linear regression.

In online setting, the predictor tensor $\mathcal{Z}$ and the response tensor $\mathcal{X}$ are updated over time. Especially for the application of our interest, i.e., the multivariate spatio-temporal stream analysis, both $\mathcal{Z}$ and $\mathcal{X}$ grow along the temporal dimension as time T increases. We define $\mathbf{W}_m = \mathcal{W}_{:,:,m}$ and similarly for others, the unconstrained optimization problem at time T can be written as $\min_{\mathbf{W}} \|\mathbf{W}\mathbf{Z}_{1:T} - \mathbf{X}_{1:T}\|_{\text{F}}^2$, where we omit the index m for simplicity. Suppose that at time stamp T, we receive a new batch of data of size b, we can update the parameter tensor in the k-th iteration $\mathcal{W}^{(k)}$ with two possible strategies: one is *exact update*, and the other is *increment update*.

Exact update Notice that we can obtain a closed-form solution of $W^{(k)}$ by using all the data from time stamp 1 to $T + b$ as follows:

$$\mathbf{W}^{(k)} = \mathbf{X}_{1:T+b}\mathbf{Z}_{1:T+b}^{\dagger}.$$

where † denotes matrix pseudo-inverse. Note that the pseudo-inverse can be computed efficiently via the Woodbury matrix identity (Woodbury, 1950). At each iteration, we can compute the inverse of the complete data covariance $(\mathbf{Z}_{1:T+b}\mathbf{Z}_{1:T+b}^{\top})^{-1}$ by inverting a smaller matrix constructed from the new data $\mathbf{Z}_{T+1:T+b}$ at a computational cost linear to the batch size b, with a small memory overhead to store the inverse of the previous covariance matrix $(\mathbf{Z}_{1:T}\mathbf{Z}_{1:T}^{\top})^{-1}$. We defer the details to Appendix B.1.

Increment update We can also incrementally update the value of $\mathbf{W}$ given the new data as follows:

$$\mathbf{W}^{(k)} = (1-\alpha)\mathbf{W}^{(k-1)} + \alpha\mathbf{X}_{T+1:T+b}\mathbf{Z}_{T+1:T+b}^{\dagger}.$$

The difference of the two updating scheme lies in the variables we store in memory. For *exact update*, we store the data statistics required to reconstruct the model. It gives an exact solution for the linear regression problem given all the historical observations. For *incremental update*, we store the previous model, compute the solution for current data only, and then take a convex combination of two models. Note that different statistical properties of these two updating scheme may require different theoretical analysis tools, but the low-rank projection of the solution is invariant to the updating strategy.

2.4. Online Low-Rank Tensor Approximation

In Step 2, we need to project the solution from Step 1 to the low-rank tensor space. In ALTO, we measure the rank with respect to the sum-n-rank of the tensor: We restrict the maximum n-rank of tensor $\mathcal{W}$ over all modes to be no larger than R. In order to obtain the n-rank projection, we resort to Tucker decomposition (De Lathauwer et al., 2000), which decomposes a tensor into a core tensor and a set of projection matrices. The dimensions of the core tensor are n-ranks of the tensor itself. The projection is generally time consuming, as it usually involves SVD on unfolded matrices at each mode of a full tensor. For the online setting, this operation needs to be repeated for each iteration, which is infeasible for large-scale applications. In ALTO, we utilize the projection results from the last iteration to approximate the current projection. It eliminates the need of SVD on unfolded matrices of a full tensor. Instead, it performs dimension reduction and computes the SVD on unfolded matrices of a low-dimensional tensor.

Without the loss of generality, we elaborate ALTO via a third order tensor. Given the Tucker decomposition of $\mathcal{W} \in \mathbb{R}^{N\times N\times N}$ from the previous iteration:

$$\mathcal{W}^{(k-1)} = \mathcal{S}^{(k-1)} \times_1 \mathbf{U}_1^{(k-1)} \times_2 \mathbf{U}_2^{(k-1)} \times_3 \mathbf{U}_3^{(k-1)}.$$

we first augment each $\mathbf{U}_i^{(k-1)} \in \mathbb{R}^{N\times R}$ with K random column vectors for $i = 1, 2, 3$, which are drawn from a zero mean Gaussian distribution. These random column vectors are introduced as noise perturbation. Then we apply Gram-Schmidt process to create orthonormal augmented projection matrices $\mathbf{V}_i^{(k-1)} \in \mathbb{R}^{N\times(R+K)}$, which has K more columns than $\mathbf{U}_i^{(t-1)}$, for $i = 1, 2, 3$ respectively.

With augmented projection matrices $\mathbf{V}_i^{(k-1)}$, we project the tensor $\mathcal{W}^{(k)}$ to an augmented core tensor $\mathcal{S}^{'(k)}$ with dimension $(R + K) \times (R + K) \times (R + K)$.

$$\mathcal{S}^{'(k)} = \mathcal{W}^{(k-1)} \times_1 \mathbf{V}_1^{(k-1)\top} \times_2 \mathbf{V}_2^{(k-1)\top} \times_3 \mathbf{V}_3^{(k-1)\top}.$$

Then we compute the rank-R approximation of the augmented core by decomposing $\mathcal{S}^{'(k)}$:

$$\mathcal{S}^{'(k)} \approx \mathcal{S}^{(k)} \times_1 \mathbf{V}_1^{'(k)} \times_2 \mathbf{V}_2^{'(k)} \times_3 \mathbf{V}_3^{'(k)}$$

where $\mathcal{S}^{(k)}$ is the new core tensor with dimension $R\times R\times R$ and $\mathbf{V}_i^{'(k)}$ is of size $(R + K) \times R$. We update the new projection matrices as $\mathbf{U}_i^{(k)} = \mathbf{V}_i^{(k-1)}\mathbf{V}_i^{'(k)}$ for $i = 1, 2, 3$. And the final low-rank projection of the solution tensor of current iteration is given by

$$\mathcal{W}^{(k)} = \mathcal{S}^{(k)} \times_1 \mathbf{U}_1^{(k)} \times_2 \mathbf{U}_2^{(k)} \times_3 \mathbf{U}_3^{(k)}.$$

We summarize the workflow of ALTO in Algorithm 1. The rank-R approximation of the augmented core $\mathcal{S}^{'(k)}$ is computed by iterating over all the modes and sequentially mapping the unfolded tensor into the rank-R subspace. We name this procedure as *low-rank Tensor Sequential Mapping* (TSM), which is described in Algorithm 2.

ALTO is computationally efficient since the augmented core tensor $\mathcal{S}^{'(k)}$ has dimension $(R + K) \times (R + K) \times (R + K)$, which is much smaller than $\mathcal{W}^{(k)}$. At each iteration, the low-rank mapping procedure TSM only involves top-R SVD on matrices of size $(R + K) \times (R + K)^2$, in comparison to the expensive top-R SVD on $N \times N^2$ matrices in most existing low-rank tensor learning approaches.

Algorithm 1 Accelerated **L**ow-rank **T**ensor **O**nline Learning (ALTO)

```
$[\mathcal{W}^{\text{new}}, \mathbf{U}^{\text{new}}] = \text{ALTO}(\mathcal{W}, \mathbf{U}, R, K)$:
**Input:** original tensor $\mathcal{W}$ and projection matrices $\mathbf{U}_i, i = 1, 2, 3$. rank $R$, augmentation factor $K$
**Output:** updated tensor $\mathcal{W}^{\text{new}}$ and projection matrices $\mathbf{U}_i^{\text{new}}, i = 1, 2, 3$.
Augment, orthogonalize and normalize $\mathbf{U}_i, i = 1, 2, 3$ to $\mathbf{V}_i, i = 1, 2, 3$ with $R + K$ columns.
Project $\mathcal{W} \to \mathcal{S}' = \mathcal{W} \times_1 \mathbf{V}_1^\top \times_2 \mathbf{V}_2^\top \times_3 \mathbf{V}_3^\top$.
Find the rank-$R$ approximation to $\mathcal{S}'$ with TSM: $\text{TSM}(\mathcal{S}', R) = \mathcal{S} \times_1 \mathbf{V}'_1 \times_2 \mathbf{V}'_2 \times_3 \mathbf{V}'_3$.
Return $\mathbf{U}_i^{\text{new}} = \mathbf{V}_i \mathbf{V}'_i, i = 1, 2, 3$ and
    $\mathcal{W}^{\text{new}} = \mathcal{S} \times_1 \mathbf{U}_1^{\text{new}} \times_2 \mathbf{U}_2^{\text{new}} \times_3 \mathbf{U}_3^{\text{new}}$.
```

Algorithm 2 low-rank Tensor Sequential Mapping

```
$\mathcal{W}^{\text{new}} = \text{TSM}(\mathcal{W}, R)$:
**Input:**: tensor $\mathcal{W}$ and target rank $R$.
**Output:**: tensor $\mathcal{W}^{\text{new}}$.
Update $\mathcal{W}_{(1)} \leftarrow p\left(\mathcal{W}_{(1)}, R\right)$, where $p(\mathbf{M}, R)$ maps $\mathbf{M}$ to its top-$R$ singular spaces.
Update $\mathcal{W}_{(2)} \leftarrow p\left(\mathcal{W}_{(2)}, R\right)$.
Update $\mathcal{W}_{(3)} \leftarrow p\left(\mathcal{W}_{(3)}, R\right)$.
Return $\mathcal{W}^{\text{new}} = \mathcal{W}$.
```

The K random column vectors are introduced so that the algorithm can jump out of the same low-rank subspace. A heuristic algorithm called Streaming Tensor Analysis (STA) is explored in (Sun et al., 2006), where the new core tensor is simply computed by $\mathcal{S}^{(k)} = \mathcal{W}^{(k)} \times_1 (\mathbf{U}_1^{(k-1)})^\top \times_2 (\mathbf{U}_2^{(k-1)})^\top \times_3 (\mathbf{U}_3^{(k-1)})^\top$. However, since the projection restricts the tensor to a fixed subspace, STA suffers from local optima because even when the projection matrices are updated after one examines the core tensor, the space is still largely invariant. Our algorithm resolves this issue via the randomization technique.

Note that when the augmentation factor K is so large that $\mathbf{V}_i$ becomes full rank, ALTO turns into an iterative singular value thresholding procedure, where the solution obtained from each iteration is directly projected to the space of low-rank tensor via top-R truncated SVD. Similar idea has been examined in (Jain et al., 2010) for the low-rank matrix learning in batch settings.

2.5. Theoretical Analysis of ALTO

We provide theoretical analysis of ALTO in terms of low-rankness, approximation accuracy as well as the behavior of the randomized projection technique. We summarize the main results and defer the detailed proofs to Appendix A.

Let $\mathcal{W}$ be the tensor before TSM procedure. The following proposition guarantees that the output $\mathcal{W}^{\text{new}}$ has low n-rank.

Proposition 1 (Low-Rank Guarantee)**.** *Given a tensor $\mathcal{W} \in \mathbb{R}^{I \times J \times K}$ and a target rank R, then for $\mathcal{W}^{new} = TSM(\mathcal{W}, R)$, its i-rank is no greater than R for any i.*

The result directly follows the conclusions from HOSVD (De Lathauwer et al., 2000).

We denote $\mathbf{W}^*$ as the rank-R target matrix and consider a matrix $\mathbf{W}$ in its neighborhood with $\|\mathbf{W} - \mathbf{W}^*\|_{\text{F}} < \epsilon$. The following proposition guarantees the approximation accuracy of TSM.

Proposition 2. *Let $\mathbf{W}^*$ be an $N \times N$ matrix with (1) rank$(\mathbf{W}^*) = R$, (2)$\|\mathbf{W}^*\|_{\text{F}} < C_w$, (3)$\sigma_k(\mathbf{W}^*) > \sigma_w$, $\mathbf{W}$ be an $N \times N$ matrix such that $\|\mathbf{W} - \mathbf{W}^*\|_{\text{F}} \leq \epsilon$, $\mathbf{E}$ be a random matrix with (3) zero mean, (4) $\sigma_N(\mathbf{E}) \geq \sigma_e$, (5)$\|\mathbf{E}\|_{\text{F}} \leq \epsilon_e$, then we have that*

$$\|p(\mathbf{W} + \mathbf{E}) - \mathbf{W}^*\|_{\text{F}}^2 \leq \|\mathbf{W} + \mathbf{E} - \mathbf{W}^*\|_{\text{F}}^2,$$

when

$$(N - 2R) \geq \frac{8(\epsilon+\epsilon_e)^4 C_w^2}{\sigma_w^4 \sigma_e^2} \quad \text{AND} \quad \sigma_w \geq 4(\epsilon_e + \epsilon).$$

Proposition 2 shows that when the target matrix is low-rank and it has reasonable condition number, then in its neighborhood, we can conduct low-rank mapping and expect the error to be reduced.

Due to the non-convexity of the low-rank space, the low-rank mapping may push the output further away from the target even if the target itself lies within the low-rank matrix space. Luckily, we prove that this is a rare event and that the space of rank-R matrix can be treated as "nearly-convex" in its neighborhood.

In order to see this, denote the full SVD of $\mathbf{W}$ as $\mathbf{W} = [\mathbf{U}_1, \mathbf{U}_2] \operatorname{diag}(\mathbf{\Sigma}_1, \mathbf{\Sigma}_2) [\mathbf{V}_1, \mathbf{V}_2]^\top$, the blocks with subscript 1 correspond to the top-R space. Then we have

$$\begin{aligned} &\|\mathbf{W} - \mathbf{W}^*\|_{\text{F}}^2 - \|p(\mathbf{W}) - \mathbf{W}^*\|_{\text{F}}^2 \\ &\quad = \left\|\mathbf{\Sigma}_2 - \mathbf{U}_2^\top \mathbf{W}^* \mathbf{V}_2\right\|_{\text{F}}^2 - \left\|\mathbf{U}_2^\top \mathbf{W}^* \mathbf{V}_2\right\|_{\text{F}}^2. \end{aligned} \quad (2)$$

By Wedin $\sin\theta$ theorem (Wedin, 1972), we know that $\left\|\mathbf{U}_2^\top \mathbf{W}^* \mathbf{V}_2\right\|_{\text{F}} \sim \mathcal{O}(\epsilon^2)$, while $\|\mathbf{\Sigma}_2\|_{\text{F}}$ is very likely to be at the level of $\mathcal{O}(\epsilon)$. That is, the reduced noise is a first order quantity, while the newly introduced bias is of the second order. It justifies that the approximation error is reduced after the low-rank sequential mapping: $\|p(\mathbf{W}) - \mathbf{W}^*\|_{\text{F}} \leq \|\mathbf{W} - \mathbf{W}^*\|_{\text{F}}$.

Note that error reduction in Proposition 2 holds under certain statistical assumptions. However, the error is still contained well within a factor of 2 without any statistical assumptions: $\|p(\mathbf{W}) - \mathbf{W}^*\|_{\text{F}} \leq 2\|\mathbf{W} - \mathbf{W}^*\|_{\text{F}}$. For tensors, the error is upper bounded by a factor of 8 in the worst case scenario, as described by the following proposition:

Proposition 3 (Approximation Guarantee)**.** *Given a tensor $\mathcal{W}^* \in \mathbb{R}^{I\times J\times K}$ where its i-rank is no greater than R for all i. If tensor $\mathcal{W} \in \mathbb{R}^{I\times J\times K}$ satisfies $\|\mathcal{W} - \mathcal{W}^*\|_\mathrm{F} \le \epsilon$ and $\mathcal{W}^{new} = TSM(\mathcal{W}, R)$, then $\|\mathcal{W}^{new} - \mathcal{W}^*\|_\mathrm{F} \le 8\epsilon$.*

The guarantees above rely on the low-rank assumption of the solution before TSM. If the data is generated from a low-rank model, the estimator can be proved to be approximately low-rank via the standard maximum likelihood estimation analysis. For many real applications, as we will show later in Section 3, we do observe the low-rank structures from the data.

Next, we further discuss the randomized projection technique. We examine the random projection on tensor $\mathcal{W} \in \mathbb{R}^{P\times Q\times M}$ at its mode-n unfolding. For instance, the mode-1 unfolding of tensor $\mathcal{W} = \mathcal{S} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$ can be represented as

$$\mathcal{W}_{(1)} = \mathbf{U}_1 \mathcal{S}_{(1)} \left(\mathbf{U}_3 \otimes \mathbf{U}_2\right)^\top,$$

where $\otimes$ is the matrix Kronecker product. The matrix $\mathbf{U}_3 \otimes \mathbf{U}_2$ is also an orthonormal matrix. And when $\mathbf{U}_i$ is augmented with K additional dimension to $\mathbf{V}_i$, the corresponding $\mathbf{V}_3 \otimes \mathbf{V}_2$ is augmented with $2K$ degrees of freedom. This connection essentially allows us to study the tensor problem from the matrix perspective.

To understand the random projection, we start with cases when $K = \max\{P, Q, M\} - R$ and $K \ge 0$. We show that setting K in the middle provides a trade-off between the amount of induced bias and the reduced noise. The case with $K = \max\{P, Q, M\} - R$ projects the tensor to the whole space, i.e., all the information are kept, so that the analysis is exactly as it in Lemma 2. For $K \ge 0$, the potential bias introduced by the information loss during the projection, as analyzed in the "nearly-convexity" section, is in the order of $\mathcal{O}(\epsilon^2)$. From rank $R + K$ to rank R, the projection step will introduce additional bias that is proportional to the order of $\mathcal{O}(\epsilon^2)$, but the noise reduction is likely to be in the order of $\mathcal{O}(\frac{K}{R+K}\epsilon)$, which dominates the extra bias if K is not too small comparing with R. This also indicates that we should set K to a larger value when R increases.

2.6. Applications for Multivariate Spatio-Temporal Streams

Tensor provides a concise representation of multivariate spatio-temporal data. We formulate two important tasks of multivariate spatio-temporal stream as tensor learning problems, which can be efficiently solved with ALTO.

Online Forecasting

We are given access to M climate variables of P locations. At each time step $t = 1, 2, \cdots, T$, we observe a set of measurements $X_{p,t,m}$ for $p \in \{1, 2, \cdots, P\}$ and $m \in \{1, 2, \cdots, M\}$. Suppose we also know the geographical coordinates of P locations. The task of online forecasting is to predict the value of $(X_{p,t+1,m}, X_{p,t+1,m}, \cdots,)$ for all variables and locations given their historical measurements.

We use the classic Vector Auto-regressive (VAR) model of lag L to describe the multivariate time series data, where we assume the generative process as $\mathcal{X}_{:,t,m} = \mathcal{W}_{:,:,m}\mathbf{X}_{t,m} + \mathcal{E}_{:,t,m}$, for $m = 1, \ldots, M$ and $t = L+1, \ldots, T$. Here $\mathbf{X}_{t,m} = [\mathcal{X}_{:,t-1,m}^\top, \ldots, \mathcal{X}_{:,t-L,m}^\top]^\top$ denotes the concatenation of L-lag historical data before time t. We learn a model coefficient tensor $\mathcal{W} \in \mathbb{R}^{P\times PL\times M}$ to forecast multiple variables simultaneously, where $\mathcal{W}_{:,:,m} = [W_{1m}, W_{2m}, \cdots, W_{Km}] \in \mathbb{R}^{P\times LP}$.

In order to achieve good prediction performance, we note two properties of spatio-temporal data: one is local smoothness, which assumes that the data in adjacent locations are likely to be similar, and the other is shared latent structures, i.e., the data lie in some shared latent structures across space, time and variables. We achieve the local smoothness via a spatial Laplacian matrix, where the Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Here $\mathbf{A}$ is a kernel matrix constructed by pairwise similarity and diagonal matrix $\mathbf{D}_{i,i} = \sum_j(\mathbf{A}_{i,j})$ (one example of the similarity matrix can be based on the geographical distances of the locations). We can achieve the global latent structures via the low-rank constraint. Therefore the online forecasting problem can be formulated as follows.

$$\widehat{\mathcal{W}} = \operatorname*{argmin}_{\mathcal{W}} \left\{ \|\widehat{\mathcal{X}} - \mathcal{X}\|_F^2 + \mu \sum_{m=1}^{M} \mathrm{tr}(\widehat{\mathcal{X}}_{:,:,m}^\top \mathbf{L} \widehat{\mathcal{X}}_{:,:,m}) \right\}$$

$$\text{s.t. } \widehat{\mathcal{X}}_{:,t,m} = \mathcal{W}_{:,:,m}\mathbf{X}_{t,m}, \; \sum_{n=1}^{N} \mathrm{rank}(\mathcal{W}_{(n)}) \le R$$

where $\mathbf{X}_{t,m} = [\mathcal{X}_{:,t-1,m}^\top, \ldots, \mathcal{X}_{:,t-L,m}^\top]^\top$ denotes the concatenation of L-lag historical data before time t.

Multi-model Ensemble

The multi-model ensemble problem arises in climate modeling. In the past decades, numerous climate models have been developed to generate large simulation data sets of future climate projections (Tebaldi & Knutti, 2007). Sophisticated physical models share similar representations of the ocean-atmosphere and land-ice processes but have different parameter uncertainty levels. Learning the correlation between model outputs and the actual observations can help quantify uncertainty in climate models and prompt the design of more accurate models. The multi-model ensemble task seeks a way to learn such correlation. It aims to combine climate model outputs into a more accurate description of the observations. While classic methods such as model coupling (Van den Berge et al., 2011) has been used in existing work, we provide an alternative way to automatically learn the ensemble model and make predictions.

Suppose we have gathered the model simulation outputs from S models of M climate variables in P locations over time period T. At the same time, we are given access to the actual observations of the same variables, locations and time. As in the forecasting problem setting, we can represent the observation measurements using a three-mode tensor $\mathcal{X} \in \mathbb{R}^{P \times T \times M}$. Similarly, we encode the model outputs with a four-mode tensor $\mathcal{Y} \in \mathbb{R}^{P \times T \times M \times S}$. Those model outputs serve as "experts" for the climate prediction. Incorporating those experts' advice can reduce the uncertainty of the forecasts.

As opposed to the forecasting task, we only focus on the current time stamp correlation between model outputs and observations. We start with a simple linear model as $\mathcal{X} = \mathcal{W}_{:,:,m}\mathbf{Y}_{t,m}$, where $\mathbf{Y}_{t,m} = [\mathcal{Y}_{:,t,m,1}^\top, \dots, \mathcal{Y}_{:,t,m,S}^\top]^\top$ denotes the concatenation of S model outputs at time t for variable m, and $\mathcal{W} \in \mathbb{R}^{P \times PS \times M}$ characterizes the "importance" of various models in climate predictions. We formulate the multi-model ensemble task as follows:

$$\widehat{\mathcal{W}} = \underset{\mathcal{W}}{\operatorname{argmin}} \left\{ \|\widehat{\mathcal{X}} - \mathcal{X}\|_F^2 + \mu \sum_{m=1}^{M} \operatorname{tr}(\widehat{\mathcal{X}}_{:,:,m}^\top \mathbf{L} \widehat{\mathcal{X}}_{:,:,m}) \right\}$$

$$\text{s.t. } \widehat{\mathcal{X}}_{:,t,m} = \mathcal{W}_{:,:,m}\mathbf{Y}_{t,m}, \ \sum_{n=1}^{N} \operatorname{rank}(\mathcal{W}_{(n)}) \le R$$

Where the Laplacian matrix $\mathbf{L}$ serves similar role as in the forecasting task to account for the spatial proximity of observations. With change of variables, both the online forecasting and the multi-model ensemble problem can be reformulated into the low-rank tensor learning framework in Equation 1. Details are deferred to Appendix B.2.

2.7. Discussion

A plethora of excellent work have been conducted for analysis of multivariate spatio-temporal data streams. For online forecasting task, time series models such autoregressive (AR), and autoregressive moving average (ARMA) models fail to capture the complex shared structure of the spatio-temporal data. Classic state-space models (Cressie & Wikle, 2011) often require high-level domain knowledge and manual work to specify the parametric form of the covariance functions. For multimodel ensemble task, (Wiegerinck & Selten, 2011) learns a super model whose dynamics are a convex combination of the individual model components. Unfortunately, learning the parameters of those statistical models is computationally expensive, making them infeasible for large-scale applications.

Our work has connection to the common practice of imposing low-rank constraint to capture the task relatedness (Ando & Zhang, 2005; Argyriou et al., 2008). However, the nature of multi-variate spatio-temporal requires us to capture the correlations not only between tasks (or features), but also between space and time. A recent study in multilinear multitask learning (Romera-Paredes et al., 2013) describes the multi-linear commonality of the data with low-rank tensor. They consider Tucker and PARAFAC tensor decomposition in the batch setting. They use alternating minimization method for tensor learning, which converges slow in practice and easily yields local optima. Another line of work in online multitask learning (Abernethy et al., 2007; Cavallanti et al., 2010; Saha et al., 2011) considers a different setting where data points from different tasks arrive one-at-a-time adversarially while in our setting, the data from multiple tasks all arrive at the same time.

3. Experiments

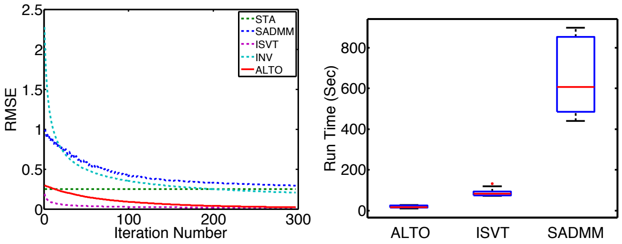
We conduct experiments on synthetic data as well as real world application data in climate and social networks. Empirical studies for tensor learning in batch settings have already been conducted in many existing work, such as (Bahadori et al., 2014). Therefore we compare our algorithm with the following online learning baselines:

- INV: closed form solution of *Exact Update* for VAR model without low-rank constraint.
- SADMM: stochastic alternating direction method of multipliers (Ouyang et al., 2013) adapted for tensor nuclear norm regularizer.
- ISVT: iterative singular value thresholding (Jain et al., 2010) generalized to tensor mode-n rank constraint.
- GREEDY: greedy sequential rank-1 approximation (Bahadori et al., 2014) for low-rank tensor learning in batch setting.

3.1. Synthetic Datasets

We generate the synthetic data stream of 30000 time stamps according to the VAR(2) model $\mathcal{X}_{:,t,m} = \mathcal{W}_{:,:,m}\mathbf{X}_{t,m} + \mathcal{E}_{:,t,m}$ for $m = 1, \dots, M$ and $t = K + 1, \dots, T$, where parameter tensor $\mathcal{W} \in \mathbb{R}^{30 \times 60 \times 20}$ is randomly drawn from standard normal distribution. We project $\mathcal{W}$ with tensor sequential mapping of rank 2. The noise at each time is independently standard normal distributed. We set the initial batch size to 200, the mini-batch size to 100, and repeat the experiment for 10 times. Figure 1 compares the average parameter estimation RMSE and the run time for ALTO and baselines over 10 random runs. We measure the run time on a machine with a 6-core 12-thread Intel Xenon 2.67GHz processor and 12GB memory.

As the true tensor is low-rank, low-rank tensor learning algorithms ISVT and ALTO outperform INV at each iteration in terms of parameter estimation accuracy. SADMM



(a) Estimation RMSE (b) Run Time

Figure 1. (a) Average parameter estimation RMSE (b) Overall run time comparison over 10 random runs for ALTO and baselines on the synthetic dataset.

outperforms INV in first few iterations, but later converges to a sub-optimal solution, since it utilizes a surrogate loss function. We also adapt Streaming Tensor Analysis (STA) (Sun et al., 2008) for our experiment. We observe that STA stays at a local optimal point and the performance barely improves after the initial iteration, which demonstrates the benefit of adding random projection in ALTO.

3.2. Spatio-Temporal Application Datasets

We conduct experiments on real world applications of multivariate spatio-temporal streams, online forecasting and multi-model ensemble respectively.

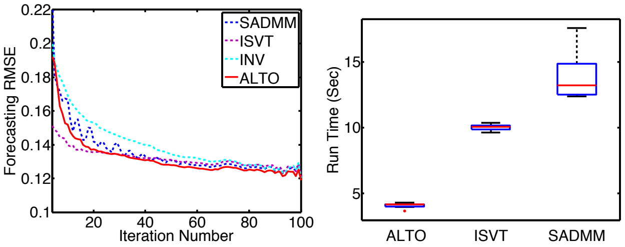
Online Forecasting

We use following two data sets for online forecasting:

Foursquare The Foursquare dataset contains the users' check-in records in the Pittsburgh area from Feb 24 to May 23, 2012, categorized by different venue types such as Art & Entertainment, College & University, and Food. The dataset records the number of check-ins by 121 users in each of the 15 categories of venues over 1200 time intervals, as well as their friendship network.

AWS The AWS dataset is provided by AWS Convergence Technologies, Inc. of Germantown, MD. It consists of 76 daily maximum values of 4 variables: surface wind speed (mph) and gust speed (mph), temperature and precipitation. We choose 153 weather stations located on a grid laying in the 35N-50N and 70W-90W block.

Figure 2 shows the forecasting RMSE per iteration and the run time on the Foursquare dataset. The superior performance of SADMM, ISVT and ALTO in forecasting accuracy over INV justify the low-rank assumptions. Compared with SADMM or ISVT, ALTO requires less computational time while achieving more accurate solutions.



(a) Forecasting RMSE (b) Run Time

Figure 2. (a) Forecasting RMSE with respect to iteration number (b) Per iteration run time comparison for ALTO and baselines on Foursquare dataset.

Table 1. Forecasting RMSE (top) and overall run time (bottom) comparison for ALTO and baselines on Foursquare and AWS datasets with 90 % training data with respect to different lag of the VAR model.

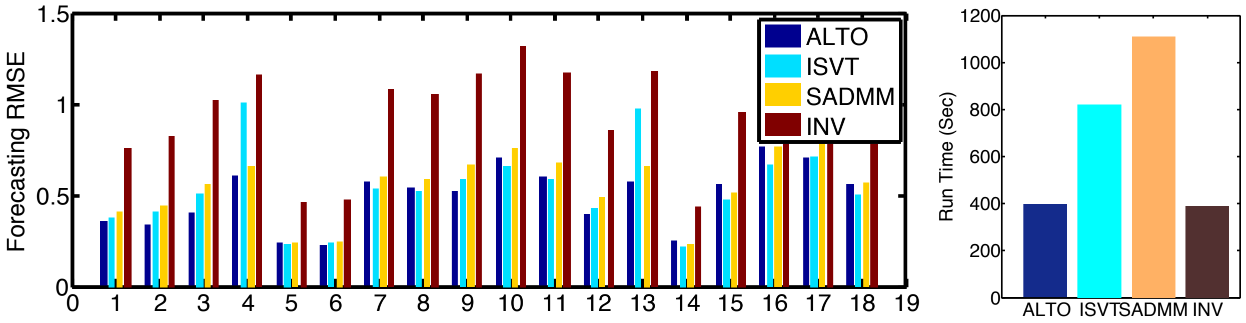
Lag	ALTO	ISVT	SADMM	INV	Greedy
			Foursquare		
1	0.1239	0.1285	0.1240	0.1394	0.1246
2	0.1244	0.1244	0.1234	0.1357	0.1225
3	0.1241	0.1240	0.1242	0.1362	0.1223
			AWS		
1	0.9318	1.0055	0.9441	1.4707	0.8951
2	0.9285	0.9182	0.9447	1.0853	0.9131
3	0.9303	0.9297	0.9485	0.9840	0.9166

Data set	ALTO	ISVT	SADMM
Foursquare	16 (s)	65 (s)	119 (s)
AWS	20 (s)	64 (s)	126 (s)

Table 1 shows the forecasting RMSE and the overall run time with 90 % training data on both datasets for VAR model with different lags. We present the results from state-of-art batch algorithm GREEDY as a reference. In general, the forecasting performance of online low-rank tensor learning algorithms significantly outperforms INV, and is comparable to that of batch algorithms. ALTO obtains accurate forecasting results with much faster speed. We also vary the rank and evaluate the performance of the ALTO algorithm. Figure 4(a) shows that both ISVT and ALTO achieves slight increase in accuracy as the rank decreases, but the difference is marginal.

Multimodel Ensemble

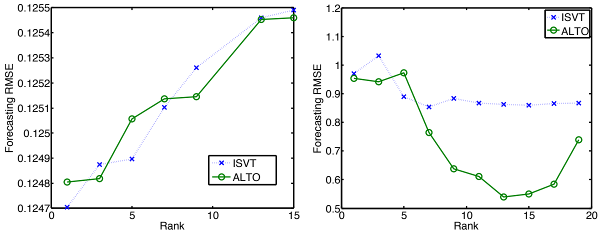
We evaluate our method on the multimodel ensemble task. For observation series, we collect the monthly measurements from NCEP-DOE Reanalysis 2 (Jones, 1999). For model outputs, 7 different model simulation data are taken from the World Climate Research Programme's (WCRP's) CMIP3 multi-model dataset and processed with CDO soft-



(a) RMSE (b) Runtime(Sec)

Figure 3. Per variable forecasting RMSE for 18 variables (a) and overall run time (b) comparison of multi-model ensemble for ALTO and baselines using 90 % training data, with 7 different models over 20 years.

ware. [1] We align the variables of observation series with the model output series. 19 variables are selected with 252 time points from 1979 to 1999. (See Appendix B.3 for details of dataset processing).



(a) Foursquare (b) Multi-model Ensemble

Figure 4. Forecasting RMSE using 90 % data with the rank value for (a) Foursquare forecasting and (b) multi-model ensemble.

We use model outputs to predict the observation measurements. 90% of the time series are used for online training. Figure 4(b) describes the forecasting RMSE for all variables with respect to the rank value. ALTO selects rank 13 as its optimal rank while ISVT chooses rank 7. We also examine the forecasting error for each variable separately using the learned model. Figure 3 shows the forecasting RMSE for 18 of the 19 variables and overall run time in second. ALTO is not only more accurate but also much faster than baselines.

Multimodel ensemble accounts for the different uncertainties in climate models. This difference is partially due to the geographical configuration of the research institutes. To see this, we aggregate the parameters of the learned tensors of all variables and color-code the models. Figure 5 shows the area where a particular model is most influential (i.e., corresponding to the largest value of the aggregated parameters).

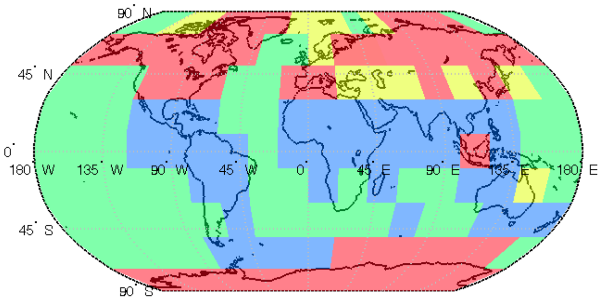


Figure 5. Climate models and their influential areas. Different color denotes different models. The influence is computed by aggregating the model parameters.

Japan Center for Climate System Research (Red) has a dominating area in Asia. Norway Bjerknes Centre for Climate Research (Yellow) is most influential in Europe. Other interesting findings reveal that Japan Meteorological Research Institute (Blue) is more accurate in the south hemisphere. Russia Institute for Numerical Mathematics (Green) shows most expertise in oceans.

4. Conclusion

In this paper, we propose a simple and efficient algorithm, namely ALTO, to accelerate the process of online low-rank tensor learning. We introduce randomized projection technique in ALTO to overcome the local optimal issue and provide theoretical justifications. We formulate two classical tasks in multivariate spatio-temporal data streams: online forecasting and multi-model ensemble, via the tensor learning framework. We demonstrate that our algorithm can produce accurate predictions and significantly reduce the computational costs. For future work, we are interested in examining broader applications and relaxing the assumptions of ALTO for better theoretical properties.

[1] CDO 2015: Climate Data Operators.
Available at: http://www.mpimet.mpg.de/cdo

5. Acknowledgement

The authors would like to thank Mohammad Taha Bahadori for insightful discussions and the anonymous reviewers for their constructive feedback. The authors are grateful to Tanachat Nilanon for his assistance in processing multi-model data. The research was sponsored by NSF research grants IIS-1134990, IIS-1254206 and U.S. Defense Advanced Research Projects Agency (DARPA) under Social Media in Strategic Communication (SMISC) program, Agreement Number W911NF-12-1-0034. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agency, or the U.S. Government.

References

Abernethy, Jacob, Bartlett, Peter, and Rakhlin, Alexander. Multitask learning with expert advice. In *Learning Theory*, pp. 484–498. Springer, 2007.

Ando, Rie Kubota and Zhang, Tong. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*, 6:1817–1853, 2005.

Argyriou, Andreas, Evgeniou, Theodoros, and Pontil, Massimiliano. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

Avron, Haim, Kale, Satyen, Sindhwani, Vikas, and Kasiviswanathan, Shiva P. Efficient and practical stochastic subgradient descent for nuclear norm regularization. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 1231–1238, 2012.

Bahadori, Mohammad Taha, Yu, Qi Rose, and Liu, Yan. Fast multivariate spatio-temporal analysis via low rank tensor learning. In *Advances in Neural Information Processing Systems*, pp. 3491–3499, 2014.

Brand, Matthew. Incremental singular value decomposition of uncertain data with missing values. In *Computer VisionECCV 2002*, pp. 707–720. Springer, 2002.

Cavallanti, Giovanni, Cesa-Bianchi, Nicolo, and Gentile, Claudio. Linear algorithms for online multitask classification. *The Journal of Machine Learning Research*, 11: 2901–2934, 2010.

Cressie, Noel and Wikle, Christopher K. *Statistics for spatio-temporal data*. John Wiley & Sons, 2011.

De Lathauwer, Lieven, De Moor, Bart, and Vandewalle, Joos. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4): 1253–1278, 2000.

Guo, Weiwei, Kotsia, Irene, and Patras, Ioannis. Tensor learning for regression. *Image Processing, IEEE Transactions on*, 21(2):816–827, 2012.

Hillar, Christopher J and Lim, Lek-Heng. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60 (6):45, 2013.

Jain, Prateek, Meka, Raghu, and Dhillon, Inderjit S. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, pp. 937–945, 2010.

Jones, Philip W. First-and second-order conservative remapping schemes for grids in spherical coordinates. *Monthly Weather Review*, 127(9):2204–2210, 1999.

Kolda, Tamara G and Bader, Brett W. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Meka, Raghu, Jain, Prateek, Caramanis, Constantine, and Dhillon, Inderjit S. Rank minimization via online learning. In *Proceedings of the 25th International Conference on Machine learning*, pp. 656–663. ACM, 2008.

Ouyang, Hua, He, Niao, Tran, Long, and Gray, Alexander. Stochastic alternating direction method of multipliers. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 80–88, 2013.

Romera-Paredes, Bernardino, Aung, Hane, Bianchi-Berthouze, Nadia, and Pontil, Massimiliano. Multilinear multitask learning. In *Proceedings of The 30th International Conference on Machine Learning*, pp. 1444–1452, 2013.

Saha, Avishek, Rai, Piyush, Venkatasubramanian, Suresh, and Daume, Hal. Online learning of multiple tasks and their relationships. In *International Conference on Artificial Intelligence and Statistics*, pp. 643–651, 2011.

Shalit, Uri, Weinshall, Daphna, and Chechik, Gal. Online learning in the manifold of low-rank matrices. In *Advances in neural information processing systems*, pp. 2128–2136, 2010.

Sun, Jimeng, Tao, Dacheng, and Faloutsos, Christos. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 374–383. ACM, 2006.

Sun, Jimeng, Tao, Dacheng, Papadimitriou, Spiros, Yu, Philip S, and Faloutsos, Christos. Incremental tensor analysis: Theory and applications. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(3):11, 2008.

Tebaldi, Claudia and Knutti, Reto. The use of the multi-model ensemble in probabilistic climate projections. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365 (1857):2053–2075, 2007.

Van den Berge, LA, Selten, FM, Wiegerinck, WAJJ, and Duane, GS. A multi-model ensemble method that combines imperfect models through learning. *Earth System Dynamics*, 2:161–177, 2011.

Wedin, Per-Åke. Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics*, 12(1):99–111, 1972.

Wiegerinck, W and Selten, F. Supermodeling: Combining imperfect models through learning. 2011.

Woodbury, Max A. Inverting modified matrices. *Memorandum report*, 42:106, 1950.

Zhang, Hongyang, Lin, Zhouchen, and Zhang, Chao. A counterexample for the validity of using nuclear norm as a convex surrogate of rank. In *Machine Learning and Knowledge Discovery in Databases*, pp. 226–241. Springer, 2013.

Zhou, Hua, Li, Lexin, and Zhu, Hongtu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.