Accelerated Online Low-Rank Tensor Learning for Multivariate spatio-temporal Streams

Rose Yu
Dehua Cheng
Yan Liu

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 poses significant challenges to existing solution in terms of computational costs and limited response time. In this paper, we propose an online accelerated low-rank tensor learning algorithm (ALTO) to solve the problem. At each iteration, we project the current tensor to the subspace of low-rank tensors in order to perform efficient tensor decomposition, then recover the decomposition of the new tensor. By randomly glancing at additional subspaces, we successfully avoid local optima at negligible extra computational cost. We evaluate our method on two tasks in streaming multivariate spatio-temporal analysis: online forecasting and multi-model ensemble, which shows that our method achieves comparable predictive accuracy with significant boost in run time.

1. Introduction

Low-rank tensor learning enjoys a broad range of applications in practical machine learning problems, ranging from signal processing, to computer vision, and to neuroscience (Kolda & Bader, 2009). One classical example is learning a low-rank tensor for multivariate regression, for which a series of effective batch learning algorithms have been developed (Guo et al., 2012; Zhou et al., 2013; Bahadori et al., 2014). We notice that in many emerging applications, such as climate data analysis, large-scale tensor data, e.g., spatio-temporal climate observations, come in streams. Batch learning algorithms not only would suffer from computational bottleneck, but also face 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 (Brand, 2002; Meka et al., 2008; Shalit et al., 2010), tensor learning remains underexplored. The problem is extremely challenging because the rank constraint is non-convex. Local solutions (Sun et al., 2008) have achieved wide success in real applications but lack rigorous theoretical understanding. For certain rank structure, we can use the nuclear norm as a convex surrogate for the rank constraint and solve 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 nuclear norm regularized problem itself is computationally expensive.

In this paper, we will address the problem of online low-rank tensor learning. Our algorithm, Accelerated online low-rank Tensor learning (ALTO), follows a sequential two-stage procedure: we first solve an unconstrained tensor learning problem, then we adjust the solution tensor to satisfy the low-rank constraint. ALTO significantly accelerates the process of online low-rank tensor learning by keeping track of the low-rank components of the solution obtained at each iteration. It performs low-rank projection of the tensor using tensor matrix multiplications and low-rank matrix singular value decomposition (SVD) to avoid expensive full-rank matrix SVD. In addition, it employs randomization technique to glance at additional linear subspaces so as to avoid the local optima in existing incremental tensor learning algorithms (Sun et al., 2008). Theoretical analysis shows that our randomization technique can significantly reduce the noise at a cost of introducing mi-
nor biases. As a side outcome, we also reveal an interesting property of the low-rank space: despite being non-convex, it usually behaves like a convex set in its neighborhood.

We demonstrate the effectiveness of our framework via example tasks of analyzing multivariate spatio-temporal streams, which are common in climatology, sociology and biology. Multivariate spatio-temporal streams often have unique structures such as spatial proximity, temporal periodicity and variable correlations. In those cases, one would expect the low-rankness to capitalize on shared latent structures of the data. We focus on two important tasks in multivariate spatio-temporal streams: one is the classic stream forecasting problem, where we are interested in performing n-step ahead prediction using the most recent observations. The other is the multi-model ensemble problem in climatology, where we want to predict the actual climate observations using the current climate model outputs.

For empirical evaluation, we conduct experiments on synthetic data to show the effectiveness of our proposed algorithm. We also examine the performance of ALTO when applied to real world spatio-temporal applications for online forecasting and multi-model ensemble tasks. Our results on climate datasets and Foursquare datasets show that our algorithm can achieve competitive prediction accuracy with significant speed-up. The low-rank model parameter tensor learned from our algorithm also provides us with interesting insights into the underlying correlations between climate models and physical climatology processes.

2. Related Work and Preliminaries

Most existing low-rank tensor learning algorithms are designed for the batch setting, such as HOSVD (De Lathauwer et al., 2000) for low-rank tensor decomposition and ALS estimators for low-rank tensor regression (Guo et al., 2012; Zhou et al., 2013). In online setting, incremental tensor analysis (Sun et al., 2008) reviews several heuristics for dynamically summarizing tensor streams. However, the theoretical aspect of those methods have not been fully examined. Many methods exist for online low-rank matrix learning, such as stochastic gradient on Riemannian manifold (Shalit et al., 2010), stochastic sub-gradient descent of nuclear norm regularization (Avron et al., 2012; Ouyang et al., 2013) and incremental singular value decomposition (Brand, 2002). Tensor generalization of these methods may yield sub-optimal solutions.

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, (Wiegnerinck & 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 relatedness not only on the task, i.e, variable level, but also on space and time. A recent study in multi-linear 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 samples from different tasks arrive one-at-a-time adversarially while in our setting, samples from multiple tasks all arrive at the same time.

Preliminaries

Across the paper, we use calligraphy font for tensors, such as \( \mathcal{A}, \mathcal{V} \), 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 \( p(\mathbf{M}, R) \) be the projection of \( \mathbf{M} \) to the top-\( R \) spectral spaces. It can be calculated using top-\( R \) truncated SVD: \( \mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^\top \), \( p(\mathbf{M}, R) = \mathbf{U}_R\Sigma_R\mathbf{V}_R^\top \). 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 \( i \) transform a tensor into a matrix \( \mathcal{A}_{(i)} \) by treating \( i \) as the first mode of the matrix and cyclically concatenate 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 \( (\mathcal{A} \times_i \mathbf{U})_{(i)} = \mathbf{U}\mathcal{A}_{(i)} \).

**Tucker Decomposition** Tucker decomposition factorize 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 core tensor \( \mathcal{S}_{(i)} \) is row-wise orthogonal for all \( i = 1, 2, \ldots, n \). Tucker decomposition can be computed by SVD on all possible unfolding of the tensor. It is also known as high order singular value decomposition (HOSVD) (De Lathauwer et al., 2000).
3. Online low-rank Tensor Learning

Consider the following low-rank tensor learning problem for regression at time stamp $T$:

$$
\hat{\mathcal{W}} = \arg \min_{\mathcal{W}} \left\{ \sum_{t=1}^{T} \| \mathcal{W} \mathcal{Z}_{t} - \mathcal{X}_{t} \|^2_F \right\}
\quad \text{s.t.} \quad \text{rank}(\mathcal{W}) \leq R,
$$

(1)

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 $\mathcal{W} \in \mathbb{R}^{P \times Q \times T \times M}$ whose rank is upper bounded by $R$. (Bahadori et al., 2014) provides a greedy algorithm for computing the solution in the batch setting. In this paper we focus on the online setting, where we assume the data stream arrives in mini-batches.

The problem in Equation 1 can be solved using the nuclear norm as a convex surrogate to rank constraint. However, optimizing over convex surrogate loss may lead to sub-optimal solutions and is computationally expensive. Our approach consists of two stages: (1) solve the unconstrained problem given the newly arrived data, (2) adapt the updated solution to satisfy the constraint, i.e., projecting the solution to the space of low-rank tensors. Such two-stage approach is nearly optimal, as we will show later in the theoretical analysis and experimental results section. It is also more computationally favorable, for the unconstrained optimization problem has a closed form solution.

For the unconstrained problem during the first stage, the quadratic loss function in Equation 1 is equivalent to $\sum_{m=1}^{M} \| \mathcal{W}_{1:m,m} \mathcal{Z}_{1:m,m} - \mathcal{X}_{1:m,m} \|^2_F$ for $m = 1, 2, \ldots, M$. Define $\mathcal{W}_m = \mathcal{W}_{1:m,m}$ and similarly for others, the unconstrained problem can be written as $\min_{\mathcal{W}} \| \mathcal{W} \mathcal{Z}_{1:T} - \mathcal{X}_{1:T} \|^2_F$, where we omit the index $m$ for simplicity. Suppose at time stamp $T$, we receive the new batch of data from $T + 1, T + 2, \ldots, T + b$ with batch size $b$. Denote the current iteration number as $k$, we can update the model parameter tensor $\mathcal{W}^{(k)}$ in the stream with one of the following schemes.

1. **[Exact Update]** the close form solution based on data from time 1 to $T + b$, i.e., $\mathcal{W}^{(k)} = \mathcal{X}_{1:T+b} \mathcal{Z}_{1:T+b}^{T}$. 

2. **[Incremental Update]** combination of previous model with the close form solution based on the newly arrived data only (Incremental Update), i.e., $\mathcal{W}^{(k)} = (1 - \alpha) \mathcal{W}^{(k-1)} + \alpha \mathcal{X}_{T+1:T+b} \mathcal{Z}_{T+1:T+b}^{T}$. 

The Exact Update can be computed efficiently via the Woodbury matrix identity:

$$(A + XX^T)^{-1} = A^{-1} - A^{-1}X(I + X^TA)^{-1}X^TA^{-1}.$$ 

where at each iteration, we can compute the inverse of the complete data covariance $(\mathcal{Z}_{1:T+b} \mathcal{Z}_{1:T+b}^{T})^{-1}$ by inverting a smaller matrix constructed from the newly arrived data $\mathcal{Z}_{T+1:T+b}$ at the computational cost on the scale of the batch size $b$, with some memory overhead to store the inverse of the previous inverse covariance matrix $(\mathcal{Z}_{1:T} \mathcal{Z}_{1:T}^{T})^{-1}$. Details of the calculation are deferred to Appendix B.1.

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 our algorithm is invariant to the update schemes.

In the second stage, projection into low-rank tensor space involves SVD of full-rank matrices for each mode, which is generally time consuming. For the online setting, this operation needs to be repeated for each iteration, which is unfeasible for large scale applications. To accelerate this process, we propose Accelerated Low rank Tensor Online Learning (ALTO) algorithm. ALTO utilizes the projection results from the last iteration to approximate the current projection. It reduces the cost of full-rank matrix SVD to that of low-rank matrix SVD. We conduct theoretical analysis of our method and prove its optimality.

3.1. ALTO: Accelerated Tensor Learning with Randomized Glance

ALTO performs low-rank projection with respect to tensor mode-$n$ rank. Mode-$n$ rank is the summation of the ranks of an unfolded tensor in all of its modes, mode-$n$ rank$(\mathcal{W}) = \sum_{n=1}^{N} \text{rank}(\mathcal{W}(n))$. We favor mode-$n$ rank over other tensor rank definitions such as CP rank because it is not only uniquely defined for tensors but also easy to compute.

We describe the projection of ALTO under the Tucker model. Tucker decomposition decomposes a tensor into a core tensor and a set of projection matrices. The dimension of the core tensor is equal to its mode-$n$ rank, thus a tensor with small core is guaranteed to be low-$n$-rank. Our algorithm projects the tensor to a subspace of low-rank tensors whose mode-$n$ rank is at most $R$.

Without the loss of generality, we elaborate ALTO for the third order tensor. Given the Tucker decomposition of $\mathcal{W}$ from the previous iteration:

$$\mathcal{W}^{(k)} = S^{(k)}U_1^{(k)} \times_1 U_1^{(k-1)} \times_2 U_2^{(k-1)} \times_3 U_3^{(k-1)}.$$ 

we first augment each $U_i^{(k-1)}$ with $K$ random column vectors for $i = 1, 2, 3$, which are drawn from a zero mean Gaussian distribution. These random column vectors aim at introducing noise as a perturbation. Then we apply Gram-
Schmidt process to create orthonormal augmented projection matrices $V_i^{(k-1)}$, which has $K$ more columns than $U_i^{(t-1)}$, for $i = 1, 2, 3$, respectively.

With augmented projection matrices $V_i^{(k-1)}$, we project the tensor $W_i^{(k)}$ to an augmented core tensor $S_i^{(k)}$ with dimension $(R + K) \times (R + K) \times (R + K)$.

$$S_i^{(k)} = W_i^{(k-1)} \times_1 V_i^{(k-1)\top} \times_2 V_2^{(k-1)\top} \times_3 V_3^{(k-1)\top}.$$  

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

$$S_i^{(k)} \approx S_i^{(k)} \times_1 V_i^{(k)} \times_2 V_2^{(k)} \times_3 V_3^{(k)}.$$  

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

$$W_i^{(k)} = S_i^{(k)} \times_1 U_i^{(k)} \times_2 U_2^{(k)} \times_3 U_3^{(k)}.$$  

We summarize the workflow of ALTO in Algorithm 1. The rank-$R$ approximation of the augmented core $S_i^{(k)}$ 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 $S_i^{(k)}$ has dimension $(R + K) \times (R + K) \times (R + K)$, which is much smaller than $W_i^{(k)}$. At each iteration, the low-rank mapping procedure TSM only involves top-$R$ SVD of rank $R + K$ matrices, in comparison to the expensive top-$R$ SVD of full rank matrices in most existing low-rank tensor learning approaches.

The augmentation using $K$ random column vectors serves the role of jumping 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 $S_i^{(k)} = W_i^{(k)} \times_1 (U_1^{(k-1)} \top) \times_2 (U_2^{(k-1)} \top) \times_3 (U_3^{(k-1)} \top)$. However, since the projection restricts the tensor to a fixed subspace, this heuristic suffers from severe performance issues. Even when the projection matrices are updated after one examines the core tensor, the space is largely invariant, which often leads to unsatisfactory local optima. Our algorithm avoids this by randomly glancing at some other subspaces.

When the augmentation factor $K$ is so large that $V_i$ becomes full rank, the algorithm turns into 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 the batch setting.

Algorithm 1 Accelerated Low rank Tensor Online Learning (ALTO)

$$|W_i^{\text{new}}, U_i^{\text{new}}| = \text{ALTO}(W_i, U_i, R, K):$$

**Input:** original tensor $W_i$ and projection matrices $U_i, i = 1, 2, 3$, rank $R$, augmentation factor $K$

**Output:** updated tensor $W_i^{\text{new}}$ and projection matrices $U_i^{\text{new}}, i = 1, 2, 3$.

1. Augment, orthonormalize and normalize $U_i, i = 1, 2, 3$ to $V_i, i = 1, 2, 3$ with $R + K$ columns.  
2. Project $W_i \rightarrow S_i' = W_i \times_1 V_i^{(1)} \times_2 V_2^{(1)} \times_3 V_3^{(1)}$.  
3. Find the rank-$R$ approximation to $S_i'$ with TSM: $\text{TSM}(S_i', R) = S_i \times_1 V_i^{(1)} \times_2 V_2^{(1)} \times_3 V_3^{(1)}$.  
4. Return $U_i^{\text{new}} = V_i V_i^{(1)}$, $i = 1, 2, 3$ and $W_i^{\text{new}} = S_i \times_1 U_i^{\text{new}} \times_2 U_2^{\text{new}} \times_3 U_3^{\text{new}}$.

Algorithm 2 Low-rank Tensor Sequential Mapping

$$W_i^{\text{new}} = \text{TSM}(W_i, R):$$

**Input:** tensor $W_i$ and target rank $R$.

**Output:** tensor $W_i^{\text{new}}$.

1. Update $W_i(1) \leftarrow p(W_i(1), R)$, where $p(M, R)$ maps $M$ to its top-$R$ singular spaces.
2. Update $W_i(2) \leftarrow p(W_i(2), R)$.
3. Update $W_i(3) \leftarrow p(W_i(3), R)$.
4. Return $W_i^{\text{new}} = W_i$.

3.2. Theoretical Analysis of ALTO

We provide theoretical analysis on the low-rank structure and the approximation accuracy of the solution obtained by ALTO. We defer all proofs to Appendix A.

**Lemma 1 (Low-rank Guarantee).** Given a tensor $W \in \mathbb{R}^{I \times J \times K}$ and a target rank $R$, then for $W' = \text{TSM}(W, R)$, we have that its mode $i$ rank is no greater than $R$ for any $i$.

Lemma 1 shows that the TSM routine guarantees the low-rankness of its output. This conclusion directly follows the results from HOSVD (De Lathauwer et al., 2000). Next we analyze the approximation accuracy of the solution tensor.

**Lemma 2 (Approximation Guarantee).** Given a tensor $W^* \in \mathbb{R}^{I \times J \times K}$ where its mode $i$ rank is no greater than $R$ for all $i$. If tensor $V \in \mathbb{R}^{I \times J \times K}$ satisfies $||W^* - V||_F \leq \epsilon$ and $W' = \text{TSM}(W, R)$, then $||W^* - W'||_F \leq 8\epsilon$.

Lemma 2 bounds the additional error incurred by TSM routine in the worst case scenario. Note that Lemma 2 holds without any statistical assumptions. And we will show that $||W' - W'||_F < ||W^* - W||_F$ holds with some statistical assumptions, which resembles the behavior as if the projected space is convex.

We state our observation for the low-rank matrix space. The results are also applicable to low-rank tensor space. Due to the non-convexity of the low-rank space, projection
may direct the result further away from the target even if the target itself lies within the low-rank matrix space. Though possible, this can be proved to be a rare event. Indeed, the space of rank-\(R\) matrix can be treated as “nearly-convex” in its neighborhood.

To simplify the notation, consider a matrix \(W\) with \(\|W - W^*\|_F < \epsilon\) and full SVD \(W = [U_1, U_2] \text{diag}(\Sigma_1, \Sigma_2) [V_1, V_2]^T\), the blocks with subscript 1 correspond to the top-\(R\) space. And \(W^*\) is the rank-\(R\) target matrix. Then we have

\[
\|W - W^*\|_F^2 - \|p(W) - W^*\|_F^2 \\
= \|\Sigma_2 - U_2^T W^* V_2\|_F^2 - \|U_2^T W^* V_2\|_F^2.
\]

By Wedin sin \(\theta\) theorem (Weden, 1972), we know that \(\|U_2^T W^* V_2\|_F \sim O(\epsilon^2)\), while \(\|\Sigma_2\|_F\) is very likely to be at the level of \(O(\epsilon)\). That is, the reduced noise is a first order quantity, while the newly introduced bias is of the second order. It explains the phenomenon we observed, i.e., the error is reduced after the greedy low-rank approximation \(- \|p(W) - W^*\|_F \leq \|W - W^*\|_F\). Moreover, even under the worst case, the error is still contained within a factor of 2, i.e., \(\|p(W) - W^*\|_F \leq 2\|W - W^*\|_F\). The following lemma rigorizes the above statement.

**Lemma 3.** Let \(W\) be an \(N \times N\) matrix with (1) \(\text{rank}(W) = R\), (2) \(\|W\|_F < C_w\), (3) \(\|W - W^*\|_F < \epsilon\). \(W'\) be an \(N \times N\) matrix such that \(\|W' - W\|_F \leq \epsilon\), \(E\) be a random matrix with (3) zero mean, (4) \(\sigma_N(E) \geq \sigma_c\), (5) \(\|E\|_F \leq \epsilon_e\), then we have that

\[
\|p(W' + E) - W^*\|_F^2 \leq \|W' + E - W\|_F^2,
\]

when

\[
(N - 2R) \geq \frac{8(\epsilon_e + \epsilon)^2}{\epsilon_e^2 \sigma_c^2}\text{ AND } \sigma_w \geq 4(\epsilon_e + \epsilon).
\]

Lemma 3 shows that when the target matrix is low-rank and it has reasonable condition number, then in its neighborhood, we can conduct low-rank projection and expect the error to be reduced. The theoretical claim relies on the approximately low-rank assumption of the estimation from the first stage. If the data is generated from a low-rank model, the estimator will be approximate low-rank by the standard maximum likelihood estimation analysis. We will show that empirically we do observe that many spatio-temporal applications satisfy this assumption.

We provide discussions on random glance technique. We view the random glance on tensor \(W \in \mathbb{R}^{P \times Q \times M}\) in its mode-\(n\) unfolding. For instance, the mode-1 unfolding of tensor \(W = S \times_1 U_1 \times_2 U_2 \times_3 U_3\) can be represented as

\[
W_{(1)} = U_1 S_{(1)} (U_3 \odot U_2)^T,
\]

where \(\odot\) is the matrix Khatri-Rao product. The matrix \(U_3 \odot U_2\) is also an orthonormal matrix. And when \(U_i\) is augmented with \(K\) additional dimension to \(V_i\), the corresponding \(V_3 \odot V_2\) is also augmented with \(K\). This connection essentially allows us to study the tensor problem from the matrix perspective.

To understand the random glance, we start with two extreme cases, \(K = 0\) and \(K = \max\{P, Q, M\} - R\), and 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 information are kept, so that the analysis is exactly as it in Lemma 3. For \(K \geq 0\), the potential bias caused by the information loss during the projection, as analyzed in the “nearly-convexity” section, is in the order of \(O(\epsilon^2)\). From rank \(R + K\) to rank \(R\), the projection step will introduce additional bias that is proportional to the order of \(O(\epsilon^2)\), but the noise reduced is likely to be in the order of \(O(\frac{K}{R + K}\epsilon)\), which dominates the extra bias. This also indicates that we should set \(K\) to a larger value when \(R\) increases.

### 3.3. 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 time step \(t = 1, 2, \ldots\), we observe a set of measurements \(X_{p,t,m}\) for \(p \in \{1, 2, \ldots, P\}\) and \(m \in \{1, 2, \ldots, M\}\). Suppose we also know the geographical coordinates of \(P\) locations. The task of online forecasting is to predict the value of \(X_{p,t+1,m}, X_{p,t+1,m}, \ldots, X_{p,t+1,m}\) 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

\[
X_{t,m} = W_{t,m}X_{t,m} + \epsilon_{t,m},
\]

for \(m = 1, \ldots, M\) and \(t = L + 1, \ldots, T\). Here \(X_{t,m} = [X_{t-1,m}, \ldots, X_{t-L,m}]^T\) denotes the concatenation of \(L\)-lag historical data before time \(t\).

We learn a model coefficient tensor \(W \in \mathbb{R}^{P \times P \times L \times M}\) to forecast multiple variables simultaneously, where \(W_{:,m} = [W_{1,m}, W_{2,m}, \ldots, W_{K,m}] \in \mathbb{R}^{P \times LP}\). The inherent shared structure of spatio-temporal data leads us to the two consistency principals: Local consistency which assumes the data in adjacent locations are likely to be similar; Global consistency which assumes the data on the shared latent structure, i.e, space, time and variable, are similar. We enforce the local consistency via the spatial Laplacian matrix, where the Laplacian matrix is defined as...
\( L = D - A \). Here \( A \) is a kernel matrix constructed by pairwise similarity and diagonal matrix \( D = \sum_j (A_{i,j}) \). One example of the similarity matrix can be based on the geographical distances of the locations. We enforce the global consistency via the low-rank constraint.

The online forecasting problem can be formulated as follows.

\[
\hat{W} = \arg\min_{\hat{W}} \left\{ \| \hat{\mathbf{X}} - \mathbf{X} \|_F^2 + \mu \sum_{m=1}^M \text{tr}(\hat{\mathbf{X}}^\top_{:,m} \mathbf{L} \hat{\mathbf{X}}_{:,m}) \right\}
\]

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

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

**MULTI-MODEL ENSEMBLE**

The multi-model ensemble problem arises in climatology modeling. In the past decades, numerous climate models have been run 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 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 \( \mathbf{X}_{t,m} = \mathcal{W}_{:,m} \mathbf{Y}_{t,m}, \mathbf{Y}_{t,m} = [\mathcal{Y}_{t,m,1}, \ldots, \mathcal{Y}_{t,m,S}]^\top \) 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 the following optimization problem.

\[
\hat{W} = \arg\min_{\hat{W}} \left\{ \| \hat{\mathbf{X}} - \mathbf{X} \|_F^2 + \mu \sum_{m=1}^M \text{tr}(\hat{\mathbf{X}}^\top_{:,m} \mathbf{L} \hat{\mathbf{X}}_{:,m}) \right\}
\]

s.t. \( \hat{X}_{:,t} = \mathcal{W}_{:,m} Y_{t,m}, \sum_{n=1}^N \text{rank}(\mathcal{W}(n)) \leq R \)

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 referred to Appendix B.2.

### 4. Experiments

We conduct experiments on synthetic data as well as real world applications of forecasting and multi-model ensemble in climate and social network datasets. We compare with the following 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.

Note that we compare mostly with online learning algorithms. Comparison against other batch learning methods have already been done in (Bahadori et al., 2014), which we take for granted here.

#### 4.1. Synthetic

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 \( m = 1, \ldots, M \) and \( t = K + 1, \ldots, T \), 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
outperforms INV at first few iterations, but later converges to a sub-optimal solution, due to the convex 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. This illustrates the benefit of using random glance of ALTO.

### 4.2. Real World spatio-temporal Applications

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 per iteration forecasting RMSE and run time for the Foursquare dataset. The superior performance of SADMM, ISVT and ALTO in forecasting accuracy over INV justify the existence of the low-rank structure in the data. Compared with SADMM or ISVT, ALTO needs less time to compute while produce more accurate solution.

Table 1 shows the forecasting RMSE and 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 the batch algorithm. ALTO obtains accurate forecasting results with much faster speed than other online baselines. We also vary the value of rank and evaluate the performance of the ALTO algorithm. Figure 4(a) shows this behavior of ISVT and ALTO. Both algorithms see a slight increase in accuracy as rank decreases, but the difference is very small.

#### Multimodel Ensemble

We evaluate our method on the multimodel ensemble task. The observation data are monthly data taken from NCEP-DOE Reanalysis 2 (Jones, 1999). 7 different model data are taken from the World Climate Research Programme’s (WCRP’s) CMIP3 multi-model dataset and processed with
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).

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.

CDO software. 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 not only achieves more accurate predictions 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 model tensor of all variables and color code the models. Figure 5 shows the area where a particular model is most influential (has the largest value of the aggregated 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.

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

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

5. Conclusion

We propose ALTO, a simple and efficient algorithm to accelerate the process of online low-rank tensor learning. We introduce random glance in ALTO to avoid the local optimal issue and provide theoretical analysis. We formulate two classic tasks in multivariate spatio-temporal data streams: online forecasting and multi-model ensemble, into our tensor learning framework. We demonstrate that our algorithm can produce accurate online prediction results, and significantly reduce the computational cost on both synthetic and real world spatio-temporal applications.
6. 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


Tebaldi, Claudia and Knutti, Reto. The use of the multi-model ensemble in probabilistic climate projections. Philosophical Transactions of the Royal Society A:


