WISDOM: Weighted Incremental Spatio-Temporal Multi-Task Learning via Tensor Decomposition

Jianpeng Xu, Jiayu Zhou, Pang-Ning Tan, Xi Liu
Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI
Email: {xujianpe, jiayuz, ptan, liuxi4}@msu.edu

Lifeng Luo
Department of Geography,
Michigan State University, East Lansing, MI
Email: lluo@msu.edu

***Abstract*—This paper presents a novel multi-task learning framework for the accurate prediction of spatio-temporal data at multiple locations. The framework encodes the data as a third-order tensor and performs supervised tensor decomposition to identify the latent factors that capture the inherent spatio-temporal variabilities of the data and their relationship to the target variable of interest. The framework is unique in that it trains both spatial and temporal prediction models from the latent factors of the decomposed tensor and aggregates their outputs to generate its final prediction. The latent factors and model parameters are simultaneously estimated by optimizing a joint objective function. We also develop an incremental learning algorithm called WISDOM to efficiently solve the optimization problem, in which the model is gradually updated with new data, either from a previously unobserved location or from its most recent time period. WISDOM can also incorporate known patterns from the application domain to guide the tensor decomposition. Finally, we showed that WISDOM outperforms several baseline algorithms in more than 75% of the locations when applied to a global-scale climate data.**

I. Introduction

Predictive modeling of geospatio-temporal data is an important task for many application domains, such as climatology [1], [19], [15], medicine [8], and crop sciences [3]. Such a task typically requires making robust predictions of a target variable at multiple geo-referenced locations based on their historical observation data and other predictor variables. For example, climate scientists are interested to obtain projections of the future climate for multiple locations by downscaling the coarse-scale outputs from regional or global climate models as predictor variables. The multi-location prediction problem can be naturally cast into a multi-task learning (MTL) framework, in which the time series prediction at each location can be regarded as a single learning task. Recent studies [27] have demonstrated the merits of performing a joint learning of the models for multi-location predictions using MTL instead of learning the model at each location independently.

While there have been several previous studies on modeling the predictions at various locations by taking into account the spatial autocorrelation [27] or spatial smoothness of the predictions [20], these methods are often developed for batch learning, thus hindering their applicability to large-scale spatio-temporal data. In addition, as many of the previous works have focused primarily on improving prediction accuracy, their resulting models are often too complicated for interpretation by the domain experts. Incorporating known patterns that drive the variability of the spatio-temporal data into a predictive modeling framework is also non-trivial. For example, it is well-known that the climate variability at a location can be influenced by broad-scale teleconnection patterns such as El Niño (see Fig. 1). How to seamlessly integrate such patterns into the predictive modeling framework and derive new, previously unknown patterns that could capture other variability in the spatio-temporal data are challenges that have not been sufficiently addressed in the literature [19], [15].

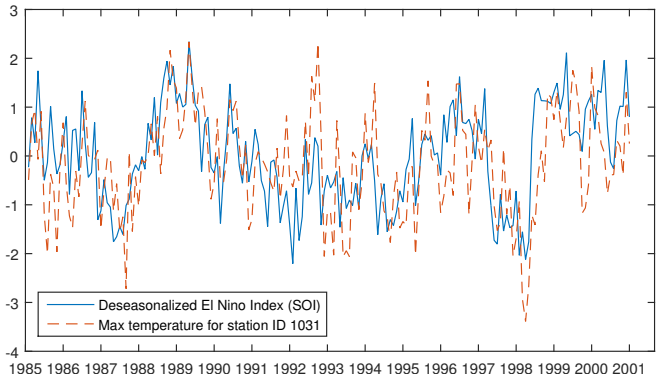


Fig. 1: The standardized monthly maximum temperature of a weather station in French Polynesia, which correlates strongly with the deseasonalized El-Nino Southern Oscillation Index.

To overcome these challenges, this paper presents a novel, incremental spatio-temporal learning algorithm called WISDOM (**W**eighted **I**ncremental **S**patio-temporal Multi-task Learning via Tensor **D**ec**om**position) for multi-location prediction. The algorithm represents the spatio-temporal data as a third-order tensor, where the dimensions (modes) of the tensor represent the temporal, spatial, and predictor variables of the data. By performing tensor decomposition, the latent factors that characterize the variability of the data along each of the three dimensions can be identified. For climate data, known temporal patterns such as El Niño can be directly integrated as a constraint on one of the temporal latent factors of the spatio-temporal tensor. Sparsity-inducing norms can also be added as additional constraints to avoid model overfitting and enhance model interpretability by the domain experts.

Our proposed tensor decomposition approach is supervised in that the latent factors of the tensor are estimated jointly

with the parameters of the prediction models in a unified learning framework. A unique aspect of our formulation is that it constructs two types of prediction models—spatial and temporal—by regressing on the spatial and temporal latent factors inferred from the data. This is a significant departure from conventional spatio-temporal prediction approaches [30], [20], [27], which typically learns a temporal model for each location from the historical observations and adds spatial constraints to guide the learning algorithm. An alternative approach is to develop a spatial prediction model such as Gaussian Markov Random Field and kriging that considers the spatial distribution of the target variable, where temporal dependencies are used to compute the covariance function of the model. In both types of approaches, the model is trained on one of the dimensions (space or time) while the other dimension is used as side information that constrains the modeling process. Instead, our formulation enables both space and time to be treated equally as it explicitly trains prediction models from both spatial and temporal latent factors. To make a prediction for location s at time t, we first apply the spatial model to the spatial latent features for s and the temporal model to the temporal latent features for t. We then compute their weighted average to determine the final prediction.

Another challenge is that the spatio-temporal data for many applications often grow rapidly over space and time. The prediction models have to be re-trained whenever new observation data become available, either for a new location (e.g., data from a newly deployed sensor) or as time progresses (when labeled data from the most recent time period are available to verify earlier predictions). Instead of performing the joint tensor factorization and model building steps repeatedly from scratch, which is computationally prohibitive due to the time and memory constraints, it would be desirable to develop a framework that can gradually update its previous latent factors and model parameters based on the newly observed data. Thus, we develop an incremental learning algorithm called WISDOM to solve the optimization problem associated with our proposed formulation.

In short, the main contributions of the paper are as follows:

1) We present a supervised tensor factorization framework for spatio-temporal predictive modeling. The framework is unique in that it constructs both spatial and temporal prediction models from the data and can incorporate known patterns from the domain.
2) We develop a scalable algorithm called WISDOM to effectively solve the optimization problem of the proposed framework. The algorithm can be applied to incremental learning over space, time, or both when new observation data become available.
3) We demonstrate the effectiveness of the proposed framework for multi-location time series prediction on a large-scale global climate data.

II. Related Work

This section presents a brief overview on some of the previous research related to this work.

A. Tensor decomposition

Existing tensor decomposition approaches can be categorized into unsupervised [12], [5], [7] and supervised methods [23], [18], [25], [30]. The former is designed to minimize reconstruction error whereas the latter considers the relationship between the predictor and response variables, and thus, is more suitable for predictive modeling problems.

Several implementations of supervised tensor decomposition approaches have been developed in recent years. For example, Wu et. al [25] proposed the SNTFM framework to map the representation of each sample from a tensor into a vector and build a predictive model on the new representation of the data. Bernardino et. al [18] and Kishan et. al [23] presented a MTL framework for data sets with multimodal structures using a supervised tensor decomposition approach. Similarly, Yu et. al [30] proposed a low-rank tensor learning approach for multivariate spatio-temporal data. These approaches encode the parameters of their predictive models as a tensor, which is assumed to have a low rank. Tensor decomposition was performed on the model parameters, unlike our proposed framework, which performs the decomposition on the spatio-temporal tensor data. This strategy allows us to provide meaningful interpretation of the latent factors in terms of their spatial, temporal, and feature dimensions.

Incremental/online tensor decomposition methods have been developed for streaming data as well as for data sets that grow dynamically over time [22]. Current incremental methods can be divided into two categories, one is based on Tucker decomposition, while the other is based on CP decomposition. Existing incremental Tucker decomposition methods are mostly based on incremental SVD, applied to the matricization of the tensor for different modes [21][22][11]. The incremental SVD based methods assume orthogonality of the latent factors, which is somewhat restrictive for interpretability reasons. Zhou, et al. [33] developed an online CP decomposition approach, where the the latent factors are updated when there are new data. However, these methods are inapplicable to our problem setting since they were developed for unsupervised learning. Although there is a recent work on online supervised tensor decomposition [30], it is restricted to new observation data along the time dimension, whereas our WISDOM framework considers new observations in both space and time.

B. Multi-task learning

MTL assumes that the generalization performance for multiple prediction tasks can be enhanced by learning the related tasks jointly[4], [31]. Existing MTL methods can be classified in terms of the way in which the task relationships are defined. This includes methods based on low-rank representation [6], [2], explicit incorporation of a graph Laplacian [9], [32], sharing of common model parameters [10], [29], and hybrid methods [26], [28]. While most of the existing MTL approaches define the task relatedness via regularization on task models, the approach developed in this paper identifies the task relatedness from the spatio-temporal data itself, by performing tensor decomposition on the data.

C. Climate modeling

Data mining and machine learning techniques have been widely used to address various problems in climate data analysis, such as downscaling the coarse-scale climate projections [24], post-processing of ensemble forecasts [26], [16], discovering new climate indices [19], and analyzing teleconnections [13]. Special techniques such as distribution preserving regression [1] have also been developed to meet the specific needs of the domain. However, incorporating known climate patterns into the framework is non-trivial. This paper presents a novel approach that can easily incorporate such patterns and identify other patterns that can influence the variability in the climate data.

III. Preliminaries

We begin with the notations used in the paper. A scalar is denoted by a lowercase letter such as c whereas a vector is denoted by a boldface lowercase letter such as $\mathbf{x}$. We denote a matrix by a boldface capital letter, such as $\mathbf{A}$ and a tensor by a boldface Euler script letter, such as $\boldsymbol{\mathcal{X}}$. The symbol ":" is used to denote sub-arrays within a matrix or tensor, e.g., $\mathbf{A}_{:,i}$ denote the i-th column of matrix $\mathbf{A}$.

A tensor is a multidimensional array, whose order refers to the number of dimensions (or modes). Each dimension in the tensor can be referred to by its index. A **fiber** of a tensor is a vector obtained by fixing all the indices of the tensor except for one of them. For example, in a 3-dimensional tensor $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{X}}_{i,:,j}$ is the mode-2 fiber, which is obtained by fixing the mode-1 index to i and mode-3 index to j. A **slice** of a tensor refers to a matrix obtained by fixing all but two of the indices of the tensor. For example, $\boldsymbol{\mathcal{X}}_{:,:,i}$ is the i-th mode-3 slice of the tensor $\boldsymbol{\mathcal{X}}$ obtained by setting its mode-3 index to i.

Mode-n matricization is the process of reordering the elements of a tensor $\boldsymbol{\mathcal{X}} \in \Re^{p_1 \times \ldots \times p_N}$ into a matrix $\mathbf{X}_{(n)} \in \Re^{p_n \times q_n}$, where $q_n = \Pi_{k \neq n} p_k$. The mode-n matricization is obtained by arranging the mode-n fibers of the tensor so that each of them is a column of $\mathbf{X}_{(n)}$. This process is also known as mode-n unfolding.

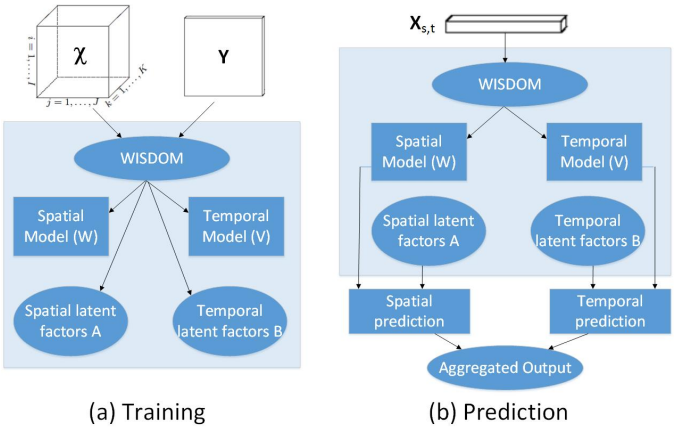
The **mode-n product** of a tensor $\boldsymbol{\mathcal{X}} \in \Re^{p_1 \times \ldots \times p_N}$ with a matrix $\mathbf{A} \in \Re^{q \times p_n}$ is defined as

$$(\boldsymbol{\mathcal{X}} \times_n \mathbf{A})_{i_1,\ldots,i_{n-1},j,i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{p_n} \boldsymbol{\mathcal{X}}_{i_1,\ldots,i_N} \mathbf{A}_{j,i_n}$$

which results in a new tensor of dimensions $p_1 \times \ldots \times p_{n-1} \times q \times p_{n+1} \times \ldots \times N$. Furthermore, if $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \ldots \times_N \mathbf{A}^{(N)}$ is an Nth-order tensor, then $\mathbf{Y}_{(n)} = \mathbf{A}^{(n)} \mathbf{X}_{(n)} \left(\mathbf{A}^{(N)} \otimes \ldots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \ldots \otimes \mathbf{A}^{(1)}\right)^T$ [14], where $\otimes$ denotes the Kronecker product.

The **Khatri-Rao product** of two matrices is equivalent to applying a Kronecker product columnwise to the matrices. For example, given matrices $\mathbf{A} \in \Re^{N \times K}$ and $\mathbf{B} \in \Re^{M \times K}$, then their Khatri-Rao product is given by:

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1, \mathbf{a}_2 \otimes \mathbf{b}_2, ..., \mathbf{a}_K \otimes \mathbf{b}_K]$$



(a) Training (b) Prediction

Fig. 2: Overview of the proposed WISDOM framework.

IV. WISDOM: An Incremental Spatio-Temporal Multi-Task Learning Framework

Let $\boldsymbol{\mathcal{D}} = (\mathcal{X}, \mathbf{Y})$ be a spatio-temporal data set, where $\mathcal{X} \in \Re^{S \times T \times d}$ denote the spatio-temporal tensor of predictor variables, $\mathbf{Y} \in \Re^{S \times T}$ denote the response variable for all the locations, S is the number of locations, T is the length of the time series, and d is the number of predictor variables. For incremental learning, the data is assumed to be periodically augmented with a new data chunk, $(\mathcal{X}_{\text{new}}, \mathbf{Y}_{\text{new}})$, where $\mathcal{X}_{\text{new}} \in \Re^{S \times 1 \times d}$ and $\mathbf{Y}_{\text{new}} \in \Re^{S \times 1}$ if the data is for a new time period or $\mathcal{X}_{\text{new}} \in \Re^{1 \times T \times d}$ and $\mathbf{Y}_{\text{new}} \in \Re^{1 \times T}$ if the data is from a new location.

A. Spatio-temporal Predictive Models

A standard approach to address the spatio-temporal prediction problem is to train a temporal prediction model for each location, $f_t(\mathbf{x}_{st}; \mathbf{w}_s)$, where $\mathbf{w}_s \in \Re^{d \times 1}$ is the model parameter for location s. Note that the temporal models can be trained independently or jointly using a multi-task learning approach such as [27] to predict values of the response variable at a future time t. In the latter case, the spatial information is typically used as constraints [20], [27] to guide the training of the temporal prediction model. Alternatively, one could also train a spatial prediction model for each time t, $f_s(\mathbf{x}_{st}; \mathbf{v}_t)$, where $\mathbf{v}_t \in \Re^{d \times 1}$ is the model parameter at time t and apply the model to predict the values of the response variable at a previously unobserved location s.

The framework proposed in this study is novel in that it simultaneously learns the temporal and spatial prediction models using the latent factors derived from the spatio-temporal tensor, as shown in Fig. 2. Unlike other previous approaches, it builds separate models from the spatial and temporal latent factors and combines the output of both models to obtain the final prediction. Assuming a linear model, the framework predicts the value for a location s at time t as a weighted linear combination of its spatial and temporal models, i.e.,

$$\hat{y}_{s,t} = \mathbf{x}_{s,t}^T \left(\sum_k^K \mathbf{A}_{s,k} \mathbf{w}_k + \sum_k^K \mathbf{B}_{t,k} \mathbf{v}_k\right) \quad (1)$$

where $\mathbf{A}_{s,k}$ denote the weight of the k-th spatial latent feature for location s, $\mathbf{B}_{t,k}$ denote the weight of the k-th temporal latent feature for time t, $\mathbf{w}_k$ and $\mathbf{v}_r$ are the parameters for the corresponding spatial and temporal prediction models of the k-th latent feature. The model parameters can be represented in matrix form as $\mathbf{W} = [\mathbf{w}_1^T; \mathbf{w}_2^T; ...; \mathbf{w}_K^T] \in \Re^{K\times d}$ and $\mathbf{V} = [\mathbf{v}_1^T; \mathbf{v}_2^T; ...; \mathbf{v}_K^T] \in \Re^{K\times d}$ and are estimated by optimizing the following joint objective function:

$$\min_{\mathbf{W},\mathbf{V}} \sum_{s}^{S}\sum_{t}^{T} \mathcal{L}(\mathbf{x}_{s,t}, \mathbf{W}, \mathbf{V}, y_{s,t}) + \Omega_m(\mathbf{W}, \mathbf{V}) \quad (2)$$

where $\mathcal{L}(\mathbf{x}_{s,t}, \mathbf{W}, \mathbf{V}, y_{s,t})$ is the loss function and $\Omega(\mathbf{W}, \mathbf{V})$ is the regularizer for the model parameters. In this paper, we consider a squared loss function, $\mathcal{L}(\mathbf{x}_{s,t}, \mathbf{W}, \mathbf{V}, y_{s,t}) = (\hat{y}_{s,t} - y_{s,t})^2$, and define $\Omega_m(\mathbf{W}, \mathbf{V}) = \|\mathbf{W}\|_1 + \|\mathbf{V}\|_1$ to ensure sparsity of the models.

B. Supervised Tensor Decomposition

The formulation presented in Eq.(2) requires knowledge about the latent factors of the spatio-temporal tensor. These latent factors are represented by the loading matrices $\mathbf{A}$ and $\mathbf{B}$, which can be derived from the data using tensor decomposition techniques. There are two standard approaches for decomposing a tensor, namely, Tucker and CANDECOMP/PARAFAC (CP) decompositions [14]. Tucker decomposition factorizes a tensor into a core tensor and a product of its loading matrices along each mode. Though it provides a more general representation, the latent factors are harder to be interpreted as the number of latent factors along each mode does not have to be identical. In contrast, CP decomposition factorizes a tensor into a sum of rank-1 tensors, i.e., $\boldsymbol{\mathcal{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \sum_{k=1}^{K} \mathbf{a}_k \circ \mathbf{b}_k \circ \mathbf{c}_k$, where $\circ$ denote the outer product operation between two vectors while $\mathbf{a}_k$, $\mathbf{b}_k$ and $\mathbf{c}_k$ correspond to the vectors associated with the k-th latent factor. The vectors $\mathbf{a}_k$, $\mathbf{b}_k$ and $\mathbf{c}_k$ also denote the k-th columns of the loading matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, respectively.

In this work, we apply CP decomposition on the spatio-temporal tensor. Let $\boldsymbol{\mathcal{X}} \in \Re^{S\times T\times d}$ be a spatio-temporal tensor, where the (s,t)-th mode-3 fiber of $\boldsymbol{\mathcal{X}}$ corresponds to the feature vector for location s at time t, i.e., $\boldsymbol{\mathcal{X}}_{s,t,:} = \mathbf{x}_{s,t}$. To estimate the latent factors, the objective function for CP decomposition can be written as follows:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \frac{1}{2}\|\boldsymbol{\mathcal{X}} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]\|_F^2 + \Omega_d(\mathbf{A}, \mathbf{B}, \mathbf{C})$$

where $\|\boldsymbol{\mathcal{X}}\|_F = \sqrt{\sum_{ijk} \mathcal{X}_{ijk}^2}$ is the Frobenius norm of the tensor $\boldsymbol{\mathcal{X}}$ and $\Omega_d(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is a regularization term for the loading matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. To ensure model sparsity, the following regularization penalty can be used:

$$\Omega_d(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \|\mathbf{A}\|_1 + \|\mathbf{B}\|_1 + \|\mathbf{C}\|_1$$

Putting everything together, the objective function for our spatio-temporal MTL framework can be stated as follows:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{V},\mathbf{A},\mathbf{B},\mathbf{C}} \mathcal{F}(\mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{B}, \mathbf{C}) \\ = \;& \frac{1}{2}\sum_{s}^{S}\sum_{t}^{T}(\mathbf{x}_{s,t}^T(\mathbf{W}^T\mathbf{A}_s + \mathbf{V}^T\mathbf{B}_t) - y_{s,t})^2 \\ + \;& \frac{\lambda}{2}\|\boldsymbol{\mathcal{X}} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]\|_F^2 + \beta\|[\mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{B}, \mathbf{C}]\|_1 \quad (3) \end{aligned}$$

where we have used $\|[\mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{B}, \mathbf{C}]\|_1$ to denote the ℓ_1 norm for $\mathbf{W}$, $\mathbf{V}$, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, respectively.

C. WISDOM Algorithm

As the size of many spatio-temporal data sets can be very large, efficient algorithms are needed to learn the spatio-temporal predictive models of the data. To optimize the objective function in Eq.(3), we develop an incremental learning algorithm called WISDOM to learn the model parameters as well as the latent factors of the tensor. While most incremental learning algorithms consider only updating the parameters over time, for spatio-temporal data, there is also a need to update the parameters over space. For example, new observation data may be available from sensors deployed at a new location or when a scientific research is expanded to include a new study region. To support this, we present two implementations of WISDOM—one for incremental learning over space and the other for incremental learning over time. A hybrid approach that combines both strategies can be easily developed when new observations can be generated over space and time.

For incremental learning, our goal is to adapt the existing models without rebuilding the model from scratch as new data become available. To ensure that the model parameters and latent factors do not vary significantly from their previous values, a smoothness criterion can be added to the objective function. We reformulate the optimization problem for incremental learning as follows:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{V},\mathbf{A},\mathbf{B},\mathbf{C}} \mathcal{Q}(\mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \tilde{\mathbf{W}}, \tilde{\mathbf{V}}, \tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}) \\ = \;& \mathcal{F}(\mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{B}, \mathbf{C}) + \Gamma(\mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \tilde{\mathbf{W}}, \tilde{\mathbf{V}}, \tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}), \end{aligned}$$

where $\tilde{\mathbf{W}}$, $\tilde{\mathbf{V}}$, $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}$ are the previous values before the update, $\mathcal{F}(\mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{B}, \mathbf{C})$ is given by Eq. (3) and

$$\begin{aligned} &\Gamma(\mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \tilde{\mathbf{W}}, \tilde{\mathbf{V}}, \tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}) \\ = \;& \|\mathbf{W} - \tilde{\mathbf{W}}\|_F^2 + \|\mathbf{V} - \tilde{\mathbf{V}}\|_F^2 + \|\mathbf{A} - \tilde{\mathbf{A}}\|_F^2 \\ &+ \|\mathbf{B} - \tilde{\mathbf{B}}\|_F^2 + \|\mathbf{C} - \tilde{\mathbf{C}}\|_F^2. \quad (4) \end{aligned}$$

1) Incremental Learning over Space: First, we discuss WISDOM's approach for incremental learning over space, when data from a new location becomes available. Let T be the current time and S be the current number of locations. We assume that the new location has historical observation data from time t_0 to T. If the location has only one observation data, then $t_0 = T$. We further assume that the spatial latent features for other locations are unaffected by the addition of the new location, i.e., $\forall s : \tilde{\mathbf{A}}_s = \mathbf{A}_s$. However, the latent

features for other modes ($\mathbf{B}$ and $\mathbf{C}$) as well as the parameters of the prediction models ($\mathbf{W}$ and $\mathbf{V}$) can be affected by the addition of the new data, $\{\mathbf{x}_{S+1,t_0}, \mathbf{x}_{S+1,t_0+1}, \cdots, \mathbf{x}_{S+1,T}\}$. For brevity, we denote the feature vectors for the new location as $\mathcal{X}_{S+1}$, which is a tensor of size $1 \times (T - t_0 + 1) \times d$.

The objective function for incremental learning over space can be expressed as follows:

$$\begin{aligned}
&\min_{\mathbf{W},\mathbf{V},\boldsymbol{\mathcal{V}},\mathbf{A}_{S+1},\mathbf{B},\mathbf{C}} \mathcal{Q}(\mathbf{W},\mathbf{V},\mathbf{A}_{S+1},\mathbf{B},\mathbf{C},\tilde{\mathbf{W}},\tilde{\mathbf{V}},\tilde{\mathbf{B}},\tilde{\mathbf{C}}) \\
= \;& \frac{1}{2}\sum_{t=t_0}^{T}\left[\mathbf{x}_{S+1,t}^T(\mathbf{W}^T\mathbf{A}_{S+1} + \mathbf{V}^T\mathbf{B}_t) - y_{S+1,t}\right]^2 \\
&+ \frac{\lambda_1}{2}\|\boldsymbol{\mathcal{X}}_{S+1} - [\![\mathbf{A}_{S+1}^T, \mathbf{B}, \mathbf{C}]\!]\|_F^2 + \frac{\eta_1}{2}\Big[\|\mathbf{W} - \tilde{\mathbf{W}}\|_F^2 \\
&\qquad + \|\mathbf{V} - \tilde{\mathbf{V}}\|_F^2 + \|\mathbf{B} - \tilde{\mathbf{B}}\|_F^2 + \|\mathbf{C} - \tilde{\mathbf{C}}\|_F^2\Big] \\
&+ \beta_1\|\mathbf{W},\mathbf{V},\mathbf{A}_{S+1},\mathbf{B},\mathbf{C}\|_1 \qquad (5)
\end{aligned}$$

Note that $\mathbf{A}_{S+1}$ is a column vector that represents the spatial latent features for the new location and $\mathbf{x}_{S+1,t}$ denote the feature vector of the location at time t. The smoothness parameter η_1 determines the extent to which the previous model parameters should be retained.

We employ an alternating minimization strategy to solve the optimization problem. Since not all terms in the objective function are differentiable, we employ the proximal gradient descent method [17] to solve each subproblem. The method is applicable to non-differentiable objective functions that can be decomposed into a smooth part and a non-smooth part. Let $f(x) = g(x) + h(x)$, where $g(x)$ is a differentiable function and $h(x)$ is a non-differentiable function. For example, the loss function involving the Frobenius norm terms in our objective function is differentiable whereas the sparsity-inducing $L1$-norm terms are non-differentiable. The proximal gradient descent method updates its parameter as following:

$$x^{(k)} = \mathbf{prox}_{t_k,h}\left(x^{(k-1)} - t_k\nabla g(x^{(k-1)})\right)$$

where $x^{(k)}$ is the parameter to be estimated at step k. $\mathbf{prox}_{t_k,h}$ is the proximal operator for the nondifferentiable function h, $\nabla g(x^{(k-1)})$ is the gradient on the smooth function g w.r.t. $x^{(k-1)}$ and t_k is the step size for the gradient descent update. The proximal operator for ℓ_1 norm function is the soft-thresholding operator: $\mathbf{prox}_{\lambda,h}(v) = (v-\lambda)_+ - (-v-\lambda)_+$. The parameters are updated iteratively by calculating the gradient on the smooth part of the objective function, and then apply the soft-thresholding operator (proximal mapping function for ℓ_1 norm) to determine its next value. The step size can be found using a line search algorithm. In the following, we provide the gradient of the objective function for each alternating minimization step.

I. Solving for $\mathbf{A}_{S+1}$ by fixing W, V, B, C:

The objective function can be simplified to retain only terms involving $\mathbf{A}_{S+1}$ as follows:

$$\begin{aligned}
\min_{\mathbf{A}_{S+1}} \quad & \frac{1}{2}\sum_{t=t_0}^{T}(\mathbf{x}_{S+1,t}^T(\mathbf{W}^T\mathbf{A}_{S+1} + \mathbf{V}^T\mathbf{B}_t) - y_{S+1,t})^2 \\
&+ \frac{\lambda_1}{2}\|\mathbf{X}_{S+1_{(1)}} - \mathbf{A}_{S+1}^T(\mathbf{C}\odot\mathbf{B})^T\|_F^2 \\
&+ \beta_1\|\mathbf{A}_{S+1}\|_1 \qquad (6)
\end{aligned}$$

where $\mathbf{X}_{S+1_{(1)}}$ is the mode-1 unfolding of the tensor $\boldsymbol{\mathcal{X}}_{S+1}$.The gradient on the smooth part of the objective function w.r.t. $\mathbf{A}_{S+1}$ is

$$\begin{aligned}
&\sum_{t=t_0}^{T}\left(\mathbf{x}_{S+1,t}^T(\mathbf{W}^T\mathbf{A}_{S+1} + \mathbf{V}^T\mathbf{B}_t) - y_{S+1,t}\right)\mathbf{W}\mathbf{x}_{S+1,t} \\
&- \lambda_1\left([\mathbf{X}_{S+1_{(1)}} - \mathbf{A}_{S+1}^T(\mathbf{C}\odot\mathbf{B})^T](\mathbf{C}\odot\mathbf{B})\right)^T
\end{aligned}$$

II. Solving for B by fixing $\mathbf{A}_{S+1}$, W, V, C:

The terms in the objective function involving matrix $\mathbf{B}$ include

$$\begin{aligned}
\min_{\mathbf{B}} \quad & \frac{1}{2}\sum_{t=t_0}^{T}(\mathbf{x}_{S+1,t}^T(\mathbf{W}^T\mathbf{A}_{S+1} + \mathbf{V}^T\mathbf{B}_t) - y_{S+1,t})^2 \\
&+ \frac{\lambda_1}{2}\|\mathbf{X}_{S+1_{(2)}} - \mathbf{B}(\mathbf{C}\odot\mathbf{A}_{S+1}^T)^T\|_F^2 \\
&+ \frac{\eta_1}{2}\|\mathbf{B} - \tilde{\mathbf{B}}\|_F^2 + \beta_1\|\mathbf{B}\|_1 \qquad (7)
\end{aligned}$$

The gradient on the smooth part of the objective function w.r.t. $\mathbf{B}_t$ is given by

$$\begin{aligned}
&\left(\mathbf{x}_{S+1,t}^T(\mathbf{W}^T\mathbf{A}_{S+1} + \mathbf{V}^T\mathbf{B}_t) - y_{S+1,t}\right)\mathbf{V}\mathbf{x}_{S+1,t} \\
+ \;& \lambda_1\left([\mathbf{x}_{S+1,t}^T - \mathbf{B}_t^T(\mathbf{C}\odot\mathbf{A}_{S+1}^T)^T](\mathbf{C}\odot\mathbf{A}_{S+1}^T)\right)^T \\
- \;& \eta_1(\mathbf{B}_t - \tilde{\mathbf{B}}_t)
\end{aligned}$$

III. Solving for C by fixing $\mathbf{A}_{S+1}$, B, W, V:

Similarly, we can simplify the objective function to include only terms involving the matrix $\mathbf{C}$:

$$\begin{aligned}
\min_{\mathbf{C}} \quad & \frac{\lambda_1}{2}\|\mathbf{X}_{S+1_{(3)}} - \mathbf{C}(\mathbf{B}\odot\mathbf{A}_{S+1}^T)^T\|_F^2 \\
&+ \frac{\eta_1}{2}\|\mathbf{C} - \tilde{\mathbf{C}}\|_F^2 + \beta_1\|\mathbf{C}\|_1 \qquad (8)
\end{aligned}$$

The gradient on the smooth part of the objective function w.r.t. $\mathbf{C}$ is given by

$$-\lambda_1\left(\mathbf{X}_{S+1_{(3)}} - \mathbf{C}(\mathbf{B}\odot\mathbf{A}_{S+1}^T)^T\right)(\mathbf{B}\odot\mathbf{A}_{S+1}^T) + \eta_1(\mathbf{C} - \tilde{\mathbf{C}})$$

IV. Solving for W by fixing V, $\mathbf{A}_{S+1}$, B, C:

The terms in the objective function involving the model parameter $\mathbf{W}$ is

$$\begin{aligned}
\min_{\mathbf{W}} \quad & \frac{1}{2}\sum_{t=t_0}^{T}(\mathbf{x}_{S+1,t}^T(\mathbf{W}^T\mathbf{A}_{S+1} + \mathbf{V}^T\mathbf{B}_t) - y_{S+1,t})^2 \\
&+ \frac{\eta_1}{2}\|\mathbf{W} - \tilde{\mathbf{W}}\|_F^2 + \beta_1\|\mathbf{W}\|_1 \qquad (9)
\end{aligned}$$

The gradient on the smooth part of the objective function w.r.t. $\mathbf{W}$ is given by

$$\sum_{t=t_0}^{T} \mathbf{x}_{S+1,t} \left(\mathbf{x}_{S+1,t}^T (\mathbf{W}^T \mathbf{A}_{S+1} + \mathbf{V}^T \mathbf{B}_t) - y_{S+1,t} \right) \mathbf{A}_{S+1}^T + \eta_1 (\mathbf{W}^T - \tilde{\mathbf{W}}^T)$$

V. Solving for V by fixing W, $\mathbf{A}_{S+1}$, B, C:
Finally, the objective function can be simplified for terms involving $\mathbf{V}$ as follows:

$$\min_{\mathbf{V}} \quad \frac{1}{2} \sum_{t=t_0}^{T} (\mathbf{x}_{S+1,t}^T (\mathbf{W}^T \mathbf{A}_{S+1} + \mathbf{V}^T \mathbf{B}_t) - y_{S+1,t})^2 + \frac{\eta_1}{2} \|\mathbf{V} - \tilde{\mathbf{V}}\|_F^2 + \beta_1 \|\mathbf{V}\|_1 \quad (10)$$

The gradient on the smooth part of the objective function w.r.t. $\mathbf{V}$ is given by

$$\sum_{t=t_0}^{T} \mathbf{x}_{S+1,t} (\mathbf{x}_{S+1,t}^T (\mathbf{W}^T \mathbf{A}_{S+1} + \mathbf{V}^T \mathbf{B}_t) - y_{S+1,t}) \mathbf{B}_t^T + \eta_1 (\mathbf{V} - \tilde{\mathbf{V}})$$

2) Incremental Learning over Time: Next, we examine WISDOM's strategy for incremental learning over time. Let S be the number of locations and T be the current time. Similar to other online learning schemes, we assume the availability of the feature vectors of predictor variables for all S locations at time $T+1$. This information will be used to determine the temporal latent factor $\mathbf{B}_{T+1}$ for the new time period. Similar to the strategy used for incremental learning over space, we assume the new data for time $T+1$ does not affect previous temporal latent factors $\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_T$.

Our strategy for incremental learning over time is to perform the following two steps: first, we learn the temporal latent factor $\mathbf{B}_{T+1}$ based on the values of the predictor variables for the new time period. Next, the model parameters and latent factors for other modes are updated when the target variable for the new time period is observed for all the locations.

Step 1: Updating the temporal latent factor $\mathbf{B}_{T+1}$ before observing target variable. The objective function for updating the temporal latent factor is given below:

$$\min_{\mathbf{B}_{T+1}} \mathcal{Q}(\mathbf{B}_{T+1}) = \frac{\lambda_2}{2} \|\boldsymbol{\mathcal{X}}_{T+1} - [\![\mathbf{A}, \mathbf{B}_{T+1}^T, \mathbf{C}]\!]\|_F^2 \quad (11)$$

Note that the loading matrices $\mathbf{A}$ and $\mathbf{C}$ correspond to the values obtained from the previous update.

Step 2: Updating model parameters and other latent factors after observing target variable. After the new observation for target variable at time $T+1$, we can derive the update formula by optimizing the following objective function:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{V},\mathbf{A},\mathbf{B}_{T+1},\mathbf{C}} \mathcal{Q}(\mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{B}_{T+1}, \mathbf{C}, \tilde{\mathbf{W}}, \tilde{\mathbf{V}}, \tilde{\mathbf{A}}, \tilde{\mathbf{C}}) \\ = \; & \frac{1}{2} \sum_{s}^{S} (\mathbf{x}_{s,T+1}^T (\mathbf{W}^T \mathbf{A}_s + \mathbf{V}^T \mathbf{B}_{T+1}) - y_{s,T+1})^2 \\ + \; & \frac{\lambda_2}{2} \|\boldsymbol{\mathcal{X}}_{T+1} - [\![\mathbf{A}, \mathbf{B}_{T+1}^T, \mathbf{C}]\!]\|_F^2 \\ + \; & \frac{\eta_2}{2} (\|\mathbf{W} - \tilde{\mathbf{W}}\|_F^2 + \|\mathbf{V} - \tilde{\mathbf{V}}\|_F^2 + \|\mathbf{A} - \tilde{\mathbf{A}}\|_F^2 \\ + \; & \|\mathbf{C} - \tilde{\mathbf{C}}\|_F^2) + \beta_1 (\|\mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{B}_{T+1}, \mathbf{C}\|_1) \end{aligned} \quad (12)$$

Solving Eq. (12) to obtain the update formula for $\mathbf{W}$, $\mathbf{V}$, $\mathbf{A}$, $\mathbf{B}_{T+1}$ and $\mathbf{C}$ is similar to the approach described for incremental learning over space. We omit their details due to lack of space.

3) Incremental Learning over Space-Time: The approaches described in the previous subsections can be combined to create a hybrid approach for incremental learning over both space and time. Specifically, the WISDOM algorithm can be initially applied to a subset of the locations at a given starting time. As time progresses, it will apply the model update approach for incremental learning over time to the newly acquired observation data. Similarly, when data from a new location becomes available, it will then invoke the update strategy for incremental learning over space.

V. Experimental Evaluation

We applied WISDOM to a global-scale climate data set and compared its performance against several baseline algorithms.

A. Dataset Description

The climate data was obtained from two sources. First, we downloaded the monthly climate observation data from the *Global Surface Summary of Day* (GSOD)[1] website. These monthly values of total precipitation (prcp), maximum (tmax), minimum (tmin), and average (tmean) temperature are used to define the target/response variable for our prediction tasks. We created 4 data sets, one for each response variable, to evaluate the performance of WISDOM. Though the four response variables can be jointly modeled in a multi-task learning framework, this is beyond the scope of the current paper.

The second source corresponds to a coarse-scale gridded climate data from NCEP reanalysis[2]. We use the data to define the predictor variables for our climate prediction task. Although there are hundreds of variables available in the NCEP reanalysis data, we selected 13 of them as our predictor variables with the help of our domain expert. A detailed description of the selected features is given in Table I.

GSOD provides climate data from more than 30,000 monitoring sites worldwide, spanning a time period from 1942 to the present time. We use the monthly data from January 1985 to November 2015 (for a total of 371 months) in our

[1] https://data.noaa.gov/dataset/global-surface-summary-of-the-day-gsod

[2] http://www.esrl.noaa.gov/psd/data/gridded/data.ncep.reanalysis.derived.html

TABLE I: List of predictor variables selected from NCEP reanalysis data.

Variable	Description
cprat.sfc	Monthly mean convective precipitation rate at surface
dlwrf.sfc	Monthly mean longwave radiation flux at surface
dswrf.sfc	Monthly mean solar radiation flux at surface
prate.sfc	Monthly mean of precipitation rate
tmax.2m	Monthly mean maximum temperature at 2 m
tmin.2m	Monthly mean minimum temperature at 2 m
lftx.sfc	Monthly mean surface lifted index
omega.sig995	Monthly mean omega at sigma level 0.995
pr_wrt.eatm	Monthly mean of precipitable water content
rhum.sig995	Monthly mean relative humidity at sigma level 0.995
slp	Sea level pressure
thick_1000500	Monthly mean of thickness for 1000-500mb
thick_850500	Monthly mean of thickness for 850-500mb

experiment. During preprocessing, we remove the sites that have missing values as well as sites that are co-located in the same grid as another previously chosen site, i.e., we restrict each grid to contain only one GSOD site. This reduces the number of sites in our data set to 1,110. The variables are deseasonalized by subtracting each monthly value from the average value of the given month and then standardized to obtain their Z-scores. The dimensionality of the resulting spatio-temporal tensor after preprocessing is $1110 \times 371 \times 13$.

B. Experimental Setup

We use the data from the first 20 years (1985 - 2004) for training and the rest (2005 - 2015) for testing the efficacy of the prediction models. The last 10 years of the training data are used as validation set to determine the model parameters. We set the number of latent factors in WISDOM to $k = 5$ and randomly select 100 sites as our initial starting locations. For incremental learning, the new observation may come from a new location or for a new time point. In the former case, we perform incremental learning over space for the new location and update the prediction models for other locations. In the latter case, we perform incremental learning over time to update the models for all the locations. The mean absolute error (MAE) for all locations are used as our evaluation metric.

We compared WISDOM against the following two baseline algorithms. The order in which new observations are introduced over space and time is the same for all the algorithms:

1) **STL** (Single Task Learning): Each location has its own local (linear) model that is incrementally updated using a gradient descent approach when it has a new observation data. When a new location is introduced, its parameters are randomly initialized and updated only when new observation data for the location becomes available.
2) **ALTO**: This is an adaptation of the method in [30], which assumes the model parameters for multiple response variables are in the form of a tensor. To extend ALTO to our problem setting, we make the following changes: First, the tensor is reduced to a matrix $\mathbf{W}$ since each data set has only one response variable. Second, we replace tensor decomposition with singular value decomposition and apply it to the noise-perturbed weight matrices to obtain the updated model parameters. We have also extended ALTO to perform incremental learning over space: when data from a new location is available, we compute the model for the new location using linear regression and adds the estimated parameters as a new row in $\mathbf{W}$. The updated $\mathbf{W}$ is then projected into its low-rank matrix representation.

In addition to the two baseline methods, we also consider the following two variations of WISDOM:

3) **WISDOM-S**: This baseline considers only the spatial component of the framework. Specifically, we remove all terms related to $\mathbf{V}$ in Eq. (5) and (12).
4) **WISDOM-T**: This baseline considers only the temporal component of the framework. We remove all terms related to $\mathbf{W}$ in Eq. (5) and (12).

C. Comparison against Baseline Methods

We first present the results comparing WISDOM against the two baseline algorithms, STL and ALTO. Table II shows the average MAE for the various methods whereas Table IV shows the number of locations in which each method outperforms another. The results suggest that WISDOM outperform STL and ALTO in more than 75% of the locations for all 4 data sets evaluated. The percentage is even higher ($> 90\%$) when compared against ALTO on the three temperature data sets. By outperforming STL, this suggests the importance of incorporating spatial autocorrelation into the learning framework. WISDOM also outperforms ALTO, which is another online tensor learning approach for spatio-temporal data. There are two possible reasons for this. First, ALTO performs the following simple update to its weight matrix each time new observation data is available[3]: $\mathbf{W}^{(k)} = (1-\alpha)\mathbf{W}^{(k-1)} + \alpha\mathbf{X}\mathbf{Z}^{\dagger}$[30]. The single-step update may not be sufficient to learn the right weights of the prediction model. In contrast, WISDOM learns the optimal weights that minimize an incrementally updated objective function. Second, ALTO performs a low-rank decomposition on a perturbed weight matrix whereas WISDOM decomposes the data tensor itself. The results suggest that the latter strategy is more effective as the observation data is potentially noisy.

Next, we compare WISDOM against its variants in Tables III and IV. Observe that WISDOM and WISDOM-S outperform WISDOM-T on all four data sets, which suggest the importance of incorporating a predictive model from the spatial latent factors. Furthermore, WISDOM performs better than WISDOM-S especially for precipitation prediction. This makes sense as precipitation has less spatial autocorrelation compared to temperature, which is why temporal autocorrelation plays a more significant role in improving its prediction.

TABLE II: MAE for WISDOM and other baseline methods

	tmax	tmin	tmean	prcp
WISDOM	**0.4751**	**0.5016**	**0.4438**	**0.5700**
STL	0.5580	0.5670	0.5233	0.6930
ALTO	0.6824	0.6598	0.6570	0.6087

[3]We use incremental update of the weight matrix instead of exact update since the latter requires the entire data to be available in memory.

TABLE III: MAE for WISDOM and its variants

	tmax	tmin	tmean	prcp
WISDOM	0.4751	**0.5016**	0.4438	**0.5700**
WISDOM-S	**0.4685**	0.5030	**0.4380**	0.5824
WISDOM-T	0.4832	0.5285	0.4607	0.6075

TABLE IV: Comparison between the number of locations (out of 1,100) in which one method outperforms another

Variable		WISDOM	WISDOM-S	WISDOM-T	ALTO	STL
tmax	WISDOM	0	621	842	1031	904
	WISDOM-S	489	0	901	1036	968
	WISDOM-T	268	209	0	976	848
	ALTO	79	74	134	0	163
	STL	206	142	262	947	0
tmin	WISDOM	0	642	823	1007	883
	WISDOM-S	468	0	869	1016	901
	WISDOM-T	287	241	0	956	792
	ALTO	103	94	154	0	192
	STL	227	209	318	918	0
tmean	WISDOM	0	621	767	1033	898
	WISDOM-S	489	0	779	1044	951
	WISDOM-T	343	331	0	1003	869
	ALTO	77	66	107	0	131
	STL	212	159	241	979	0
prcp	WISDOM	0	756	946	838	990
	WISDOM-S	354	0	789	685	997
	WISDOM-T	164	321	0	651	910
	ALTO	272	425	495	0	852
	STL	120	113	200	258	0

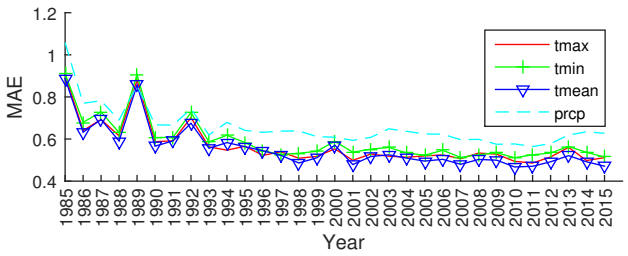


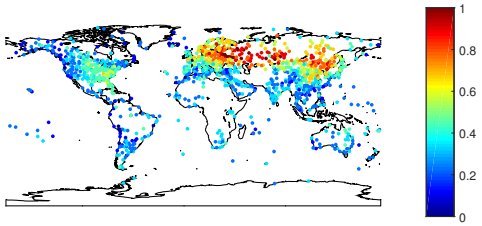
Fig. 3: Changes in MAE over time for WISDOM

D. Convergence Analysis of WISDOM

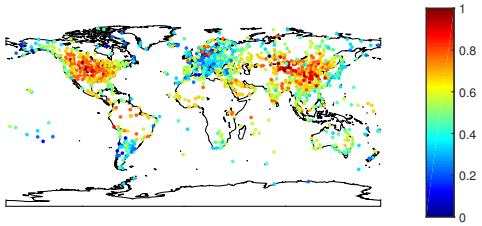
To demonstrate its convergence, Fig. 3 shows the average MAE of WISDOM for all locations across time. A location is included into the average MAE calculation only after the data for the location becomes available. Although there are some instabilities in its performance during the first 8 years, WISDOM begins to converge after the first 10 years, which is our initial training period, on all four data sets.

E. Analysis of Spatial Latent Factors

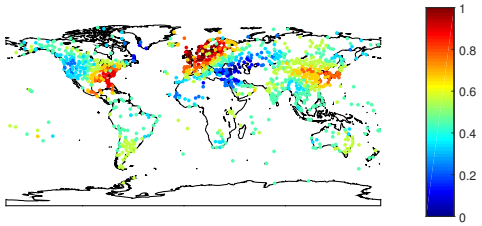
Next, we investigate the spatial latent factors derived by WISDOM. Each spatial latent factor is a vector whose elements represent the membership of each location to the given latent factor. Figure 4 shows the spatial distribution of the latent factors for prcp. The figure shows that the latent factors have varying spatial distributions, which suggests that they capture different aspects of the spatial variability in the data. For example, the first latent factor is dominant in Europe and north of China whereas the second latent factor emphasizes more in US and east of China.



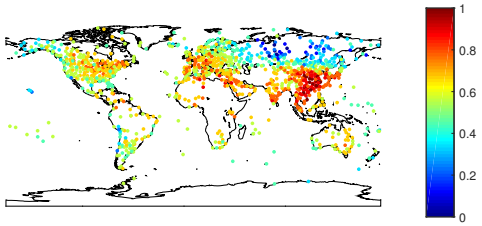
(a) Factor 1



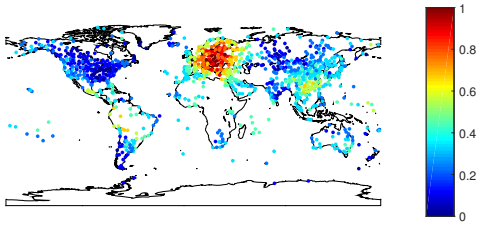
(b) Factor 2



(c) Factor 3



(d) Factor 4



(e) Factor 5

Fig. 4: Spatial distribution of the spatials factor learned by WISDOM for prcp. (Figure is best viewed in color).

WISDOM utilizes the spatial latent factors to perform incremental learning over space. To further demonstrate the benefit of incremental learning over space, we compare the average annual MAE for the first 100 randomly chosen locations when the model is updated with and without incremental learning

TABLE V: List of the climate indices used to correlate with the temporal factors learned from WISDOM.

Climate Index	Description
AOI	Arctic Oscillation Index
NAO	North Atlantic Oscillation
WPI	West Pacific Pattern
QBO	Quasi-Biennial Oscillation
PDO	Pacific Decadal Oscillation
SOI	Southern Oscillation Index

TABLE VI: Comparing MAE of WISDOM and WISDOM-KP

	tmax	tmin	tmean	prcp
WISDOM	0.4751	**0.5016**	0.4438	**0.5700**
WISDOM-KP	**0.4678**	0.5037	**0.4343**	0.5725

over space. Specifically, in the latter case, no new locations are added into the data set as time progresses. Indeed, as shown in Fig 5, adding data from new locations helps to improve the MAE of the first 100 randomly chosen locations.

F. Analysis of Temporal Latent Factors

Each temporal latent factor derived by WISDOM can be represented as a time series. To understand their significance, we correlate the temporal latent factors against the known climate indices given in Table V. Fig. 6 shows the resulting correlation for the tmean and prcp data sets. Though the temporal latent factors for both data sets are different, we found some of the factors correlate highly (over 0.6) with the existing climate indices. This result suggests that the temporal latent factors may capture some of the previously known climate phenomena, represented by the climate indices such as AOI and NAO. For each temporal latent factor and climate index, we also calculate the percent of locations whose temperature or precipitation has a correlation above 0.3. The results in Fig. 7 suggest that (1) not all climate indices have a significant number of locations highly correlated with them and (2) some latent factors have significant correlation with a relatively large number of locations, comparable to the known indices. More importantly, as some of the latent factors do not correlate highly with the known indices, this suggests that our framework can potentially discover new indices that capture the climate variability for many locations.

Surprisingly, none of the temporal latent factors were found to correlate highly with SOI, which is a surrogate time series for El Niño. One of the strengths of WISDOM is its ability to incorporate known domain patterns as additional constraints for its formulation. In order to incorporate known patterns such as SOI, we simply fix one of the columns in the temporal latent factor matrix $\mathbf{B}$ to be the time series of SOI and learn the remaining spatial and temporal latent factors using WISDOM. We denote this approach as **WISDOM-KP**. The MAE results comparing WISDOM against WISDOM-KP is shown in Table VI. The results suggest that WISDOM-KP achieves comparable results as WISDOM in terms of their average MAE. In addition, we also compared the number of locations where WISDOM-KP outperforms WISDOM. For tmax, tmean, and prcp, the MAE for WISDOM-KP is lower than WISDOM in at least 49% of the locations whereas for tmin, the percentage is around 43%. This is not surprising as we do not expect SOI to accurately capture the climate variability for all locations. Instead, there are locations that are expected to benefit from using SOI as one of the temporal latent factors. To identify such locations, we plot a map of the locations in which WISDOM-KP is better than WISDOM, and vice-versa, for predicting t-mean in Fig. 8. The results suggest that by incorporating SOI, an improved predictive performance is observed in areas such as Australia, part of South America, northeast of North America, and locations around Arctic Ocean. Some of these locations are consistent with the results of previous studies [19].

VI. Conclusion

This paper presents a spatio-temporal learning framework for multi-location prediction based on supervised tensor decomposition. Unlike conventional methods, the proposed framework constructs both spatial and temporal models of the data and aggregates their output to obtain the final prediction. A novel incremental learning algorithm called WISDOM is developed to simultaneously extracts the latent factors of the spatio-temporal data and learns the spatial and temporal prediction models. We show that WISDOM outperforms several baseline algorithms and can easily accommodate known patterns from the spatio-temporal domain.

Acknowledgments

This research is partially supported by NOAA Climate Program office through grant #NA12OAR4310081, NASA Terrestrial Hydrology Program through grant #NNX13AI44G, and NSF grants #IIS-1565596, IIS-1615597 and IIS-1615612.

References

[1] Z. Abraham, M. Liszewska, Perdinan, P.-N. Tan, J. Winkler, and S. Zhong. Distribution regularized regression framework for climate modeling. In *SDM*, pages 333–341, 2013.

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, Dec. 2008.

[3] S. Boriah, V. Kumar, M. Steinbach, C. Potter, and S. Klooster. Land cover change detection: A case study. In *KDD*, pages 857–865, 2008.

[4] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[5] K.-W. Chang, W.-T. Yih, B. Yang, and C. Meek. Typed tensor decomposition of knowledge bases for relation extraction. In *EMNLP*, 2014.

[6] J. Chen, J. Zhou, and J. Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *KDD*, pages 42–50, 2011.

[7] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *Signal Processing Magazine*, 32(2):145–163, 2015.

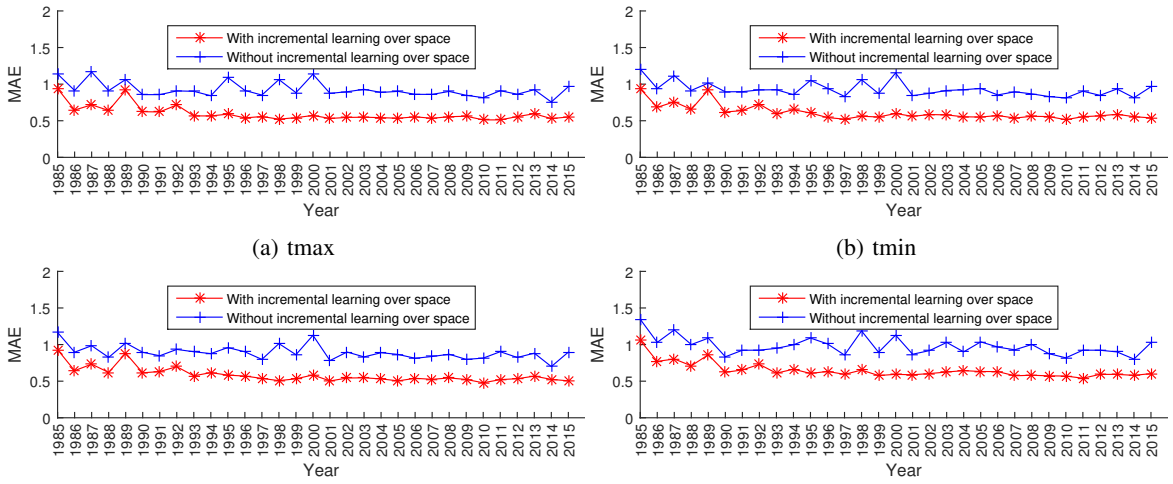
[8] G. Davis, N. Sevdalis, and L. Drumright. Spatial and temporal analyses to investigate infectious disease transmission within healthcare settings. *Journal of Hospital Infection*, 86:227–243, 2014.

[9] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *JMLR*, 6:615–637, Dec. 2005.

[10] T. Evgeniou and M. Pontil. Regularized multi-task learning. In *KDD*, pages 109–117, 2004.

[11] W. Hu, X. Li, X. Zhang, X. Shi, S. Maybank, and Z. Zhang. Incremental tensor subspace learning and its applications to foreground segmentation and tracking. *IJCV*, 91(3):303–327, 2011.

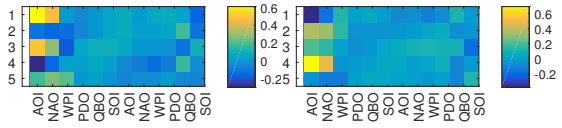
[12] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *RecSys 2010*, pages 79–86, 2010.



(a) tmax (b) tmin

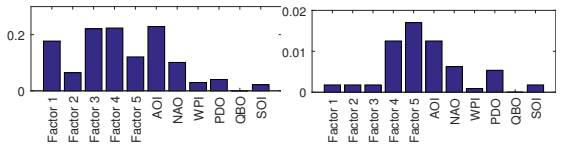
(c) tmean (d) prcp

Fig. 5: Average annual MAE comparison between WISDOM with incremental learning over space and WISDOM without incremental learning over space for the 100 initially chosen locations



(a) tmean (b) prcp

Fig. 6: Correlations between the climate indices and the temporal factors learned from WISDOM for tmean and prcp



(a) tmean (b) prcp

Fig. 7: Percentage of locations whose response variables has a correlation above 0.3 with the temporal factors and climate indices learned from WISDOM for tmean and prcp

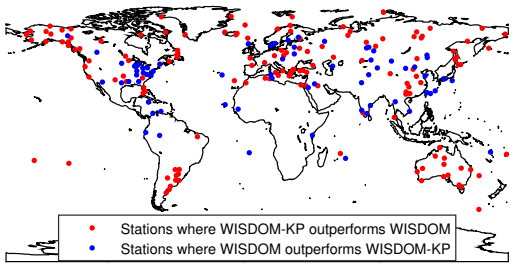


Fig. 8: Stations where WISDOM-KP outperforms WISDOM more than 0.05 in MAE evaluation for tmean and vise versa.

[13] J. Kawale, S. Chatterjee, D. Ormsby, K. Steinhaeuser, S. Liess, and V. Kumar. Testing the significance of spatio-temporal teleconnection patterns. In *KDD*, pages 642–650, 2012.

[14] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, Aug. 2009.

[15] C. Monteleoni, G. Schmidt, and S. McQuade. Climate informatics: Accelerating discovering in climate science with machine learning. *Computing in Science Engineering*, 15(5):32–40, Sept 2013.

[16] C. Monteleoni, G. A. Schmidt, and S. Saroha. Tracking climate models. In *CIDU*, pages 1–15, 2010.

[17] N. Parikh and S. Boyd. Proximal algorithms. *Found. Trends Optim.*, 1(3):127–239, Jan. 2014.

[18] B. Romera-Paredes, H. Aung, N. Bianchi-Berthouze, and M. Pontil. Multilinear multitask learning. In *ICML*, pages 1444–1452.

[19] M. Steinbach, P.-N. Tan, V. Kumar, S. Klooster, and C. Potter. Discovery of climate indices using clustering. In *KDD*, pages 446–455, 2003.

[20] K. Subbian and A. Banerjee. Climate multi-model regression using spatial smoothing. In *SDM*, pages 324–332, 2013.

[21] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: Dynamic tensor analysis. In *KDD*, pages 374–383, 2006.

[22] J. Sun, D. Tao, S. Papadimitriou, P. S. Yu, and C. Faloutsos. Incremental tensor analysis: Theory and applications. *TKDD*, 2(3):11:1–11:37.

[23] K. Wimalawarne, M. Sugiyama, and R. Tomioka. Multitask learning meets tensor factorization: task imputation via convex optimization. In *NIPS*, pages 2825–2833. 2014.

[24] J. Winkler, G. Guentchev, Perdinan, P.-N. Tan, S. Zhong, M. Liszweska, Z. Abraham, T. Niedzwiedz, and Z. Ustrnul. Climate scenario development and applications for local/regional climate change impact assessments: An overview for the non-climate scientist. part i: Scenario development using downscaling methods. *Geography Compass*, 5(6):275–300, 2011.

[25] F. Wu, X. Tan, Y. Yang, D. Tao, S. Tang, and Y. Zhuang. Supervised nonnegative tensor factorization with maximum-margin constraint. In *AAAI 2013*, 2013.

[26] J. Xu, P.-N. Tan, and L. Luo. ORION: Online Regularized multI-task regressiON and its application to ensemble forecasting. In *ICDM*, pages 1061–1066, 2014.

[27] J. Xu, P.-N. Tan, L. Luo, and J. Zhou. Gspartan: a geospatio-temporal multi-task learning framework for multi-location prediction. In *SDM 2016*, 2016.

[28] J. Xu, J. Zhou, and P.-N. Tan. Formula: Factorized multi-task learning for task discovery in personalized medical models. In *SDM*, pages 496–504, 2015.

[29] K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *ICML*, pages 1012–1019, 2005.

[30] R. Yu, D. Cheng, and Y. Liu. Accelerated online low rank tensor learning for multivariate spatiotemporal streams. In *ICML 2015*, volume 37, pages 238–247, 2015.

[31] J. Zhou, J. Chen, and J. Ye. *MALSAR: Multi-tAsk Learning via StructurAl Regularization*. Arizona State University, 2011.

[32] J. Zhou, L. Yuan, J. Liu, and J. Ye. A multi-task learning formulation for predicting disease progression. In *KDD*, pages 814–822, 2011.

[33] S. Zhou, X. V. Nguyen, J. Bailey, Y. Jia, and I. Davidson. Accelerating Online CP Decompositions for Higher Order Tensors. In *KDD*, 2016.