MUSCAT: Multi-Scale Spatio-Temporal Learning with Application to Climate Modeling

Jianpeng Xu[1], Xi Liu[1], Tyler Wilson[1], Pang-Ning Tan[1], Pouyan Hatami[2], Lifeng Luo[2],
[1] Department of Computer Science and Engineering, Michigan State University
[2] Department of Geography, Michigan State University
{xujianpe, liuxi4, wils1270, ptan, pouyanhb, lluo}@msu.edu

Abstract

In climate and environmental sciences, vast amount of spatio-temporal data have been generated at varying spatial resolutions from satellite observations and computer models. Integrating such diverse sources of data has proven to be useful for building prediction models as the multi-scale data may capture different aspects of the Earth system. In this paper, we present a novel framework called MUSCAT for predictive modeling of multi-scale, spatio-temporal data. MUSCAT performs a joint decomposition of multiple tensors from different spatial scales, taking into account the relationships between the variables. The latent factors derived from the joint tensor decomposition are used to train the spatial and temporal prediction models at different scales for each location. The outputs from these ensemble of spatial and temporal models will be aggregated to generate future predictions. An incremental learning algorithm is also proposed to handle the massive size of the tensors. Experimental results on real-world data from the United States Historical Climate Network (USHCN) showed that MUSCAT outperformed other competing methods in more than 70% of the locations.

1 Introduction

The spatio-temporal data obtained from climate and environmental sciences are often available at multiple spatial resolutions. For example, Table 1 shows examples of climate data generated by different global and regional climate models, which can have varying spatial scales, from tens to several hundred kilometers. Such multi-scale data provide useful information that can aid scientists in understanding the variability of the climate system in order to predict its future behavior. However, since the data at different scales are potentially correlated with each other, concatenating them together into a single feature vector may not be an effective strategy for building robust prediction models. Utilizing only the data from the finest resolution is also not a viable approach as the data at coarser resolutions may capture broad-scale effects that cannot be easily discerned from the finest resolution data. Finding an effective way to integrate the multi-scale data into a prediction modeling framework is thus a challenge that needs to be addressed [Miller *et al.*, 2015].

Table 1: Spatial resolutions for various climate datasets

Climate Dataset	Scale
NCEP North American Regional Reanalysis	32 km
Canadian Regional Climate Model	45 km
Weather Research & Forecasting Model	50 km
HadCM3 Global Climate Model	300 km

For applications such as climate modeling, in addition to the multi-scale nature of the data, the predictions must be made at multiple locations. Although a model can be trained to fit the training data at each location independently, the output predictions may not be spatially coherent as the models may not preserve the spatial autocorrelation of the data. Furthermore, the amount of training data may vary from one location to another, making it difficult to obtain accurate models for locations with limited training data. Recent works have demonstrated the advantages of applying multi-task learning to such multi-location prediction problems [Xu *et al.*, 2016a][Xu *et al.*, 2016b][Gonçalves *et al.*, 2016][Gonçalves *et al.*, 2017][Yu *et al.*, 2015]. These approaches consider the spatial relationship between different locations to jointly train the models. However, none of them are designed to handle multi-scale data. Furthermore, these approaches are mostly developed for batch learning algorithms, making it harder to scale them up to larger spatio-temporal datasets.

Finally, model interpretability is another important consideration for many spatio-temporal applications. For example, climate scientists are interested to understand the major driving factors that influence the climate variability at various locations. Some of the factors, such as the El Niño phenomenon, are well-known to the scientists, but there may be other broad-scale patterns governing the variability of the data. Thus, it would be useful to develop a framework that can shed light on these patterns in addition to generating accurate predictions.

To address these challenges, this paper presents a novel framework called MUSCAT (MUlti-SCAle Spatio-Temporal Learning) for the predictive modeling of multi-scale spatio-temporal data. MUSCAT represents the spatio-temporal data at each scale as a 3-dimensional tensor. The tensors are then jointly decomposed into a set of shared latent factors, representing the various patterns that can help summarize the variability observed in the spatio-temporal data. MUSCAT employs an ensemble of spatial and temporal prediction models to make its predictions, where the spatial models are trained to fit the climate response variable against the shared spatial latent factors of the tensors whereas the temporal models are trained to fit the respective temporal latent factors. More importantly, the multi-scale tensor decomposition and the fitting of spatial and temporal prediction models are performed simultaneously in a unified learning framework. As this can be computationally expensive due to the massive size of the spatio-temporal data, an incremental learning algorithm is proposed. The algorithm enables the latent factors and model parameters to be iteratively learned over space and time, thereby avoiding the need to rebuild the models from scratch each time there is new data available.

In short, the main contributions of this work are:

1. A novel framework called MUSCAT is proposed for the predictive modeling of multi-scale spatio-temporal data. The framework allows the latent spatial and temporal patterns shared by the multiple scales to be simultaneously derived using a multi-tensor decomposition approach. MUSCAT also trains an ensemble of spatial and temporal prediction models, whose outputs will be aggregated when making predictions for a new location or for a future time period.
2. To improve its scalability, an incremental learning algorithm over space and time is proposed.
3. Experiments performed on real-world data from the United States Historical Climate Network (USHCN) demonstrates the superiority of MUSCAT compared to other competing algorithms.

2 Related Works

This section reviews some of the previous works related to this research. In recent years, multi-task learning has been proven to be effective at learning models for predicting multiple tasks jointly by taking into account the relationships among the tasks [Caruana, 1997]. The success of multi-task learning for spatio-temporal data has also been been demonstrated in [Xu *et al.*, 2014][Xu *et al.*, 2016a][Xu *et al.*, 2016b][Zhao *et al.*, 2015][Yu *et al.*, 2015]. However, none of these approaches are designed for multi-scale data.

The term multi-scale learning has been used rather loosely in the literature to describe different classes of methods. For example, in traditional machine learning, it has referred to techniques based on multiple kernels [Bellocchio *et al.*, 2012], multi-covariance matrices [Walder *et al.*, 2008] or multi-basis functions [Nounou and Nounou, 2010], none of which are designed to handle multi-scale data. Instead, they were developed to extract multi-scale features from the given data. A closer related area is in deep learning, where multi-scale modeling approaches have been developed for computer vision applications. Here, the multi-scale data refer to different resolutions of an image [Bertasius *et al.*, 2015; Zhao and Du, 2016]. Other multi-scale learning approaches, such as those proposed in [Neverova *et al.*, 2014] and [Eigen *et al.*, 2014] do not consider the relationships between data at different scales.

Tensor decomposition has been widely used to explore the latent features of multi-dimensional data [Kolda and Bader, 2009]. This includes previous works on coupled tensor decomposition [Acar *et al.*, 2011; Ermiş *et al.*, 2015] for the joint analysis of data from multiple sources. However, such methods are mostly designed for batch learning, and thus, cannot efficiently handle the dynamic growth of data in different dimensions (e.g., spatial and temporal). As an alternative, incremental or online tensor decomposition approaches have been developed in recent years [Sun *et al.*, 2008; Zhou *et al.*, 2016]. These methods are unsupervised, and thus, are not as effective compared to the MUSCAT framework proposed in this paper. More recently, Yu, et al. [Yu *et al.*, 2015] also presented an online supervised tensor decomposition method. Unlike MUSCAT, the method employed a tensor to represent the model parameters instead of the data. As will be shown in our experiments, the strategy used by MUSCAT to represent data as tensors instead of model parameters as tensors is more effective for modeling the multi-scale spatio-temporal data investigated in this study.

3 Proposed MUSCAT Framework

MUSCAT is a supervised learning framework that jointly performs a coupled tensor decomposition on a multi-scale spatio-temporal dataset and fits the derived latent factors to the response variable of interest at multiple locations. This section presents the details of our proposed framework.

3.1 Preliminaries

Let $\boldsymbol{\mathcal{D}} = (\boldsymbol{\mathcal{X}}^{(1)}, ..., \boldsymbol{\mathcal{X}}^{(L)}, \mathbf{Y})$ be a multi-scale spatio-temporal data set, where $\boldsymbol{\mathcal{X}}^{(l)} \in \Re^{S \times T \times d_l}$ denote the spatio-temporal tensor of predictor variables at the l-th scale, $\mathbf{Y} \in \Re^{S \times T}$ denote the time series of the response variable for all locations, S is the number of locations, T is the length of time series, and d_l is the number of predictor variables associated with the l-th scale.

Inspired by previous work on discovering climate indices using SVD [Steinbach *et al.*, 2003], we employ tensor decomposition to extract broad-scale patterns from the spatio-temporal data. Specifically, the following CANDECOMP/PARAFAC (CP) decomposition [Kolda and Bader, 2009] is used to decompose a 3rd-order ten-

sor $\boldsymbol{\mathcal{X}}$ into its corresponding latent factors, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$:

$$\boldsymbol{\mathcal{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \sum_{k=1}^{K} \mathbf{a}_k \circ \mathbf{b}_k \circ \mathbf{c}_k, \quad (1)$$

where $\mathbf{a}_k$, $\mathbf{b}_k$ and $\mathbf{c}_k$ are column vectors corresponding to the k-th columns of matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, and $\circ$ denotes the outer product operation.

3.2 Supervised Tensor Decomposition for Multi-Scale Data

For multi-scale spatio-temporal data, the tensor decomposition can be performed at each scale l as follows:

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

where $\mathbf{A} \in \Re^{S \times K}$ denote the spatial latent factors and $\mathbf{B} \in \Re^{T \times K}$ denote the temporal latent factors. Our framework assumes that the spatial and temporal latent factors $\mathbf{A}$ and $\mathbf{B}$ are invariant across all scales. The normalization factor $vol(l) = 2(S \times T \times d_l)$ ensures that the decomposition at each scale does not depend on its tensor size. Finally, $\Omega_d(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is a regularization term used to enforce sparsity of the latent factors:

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

Although the latent factors derived via multi-tensor decomposition may capture the spatial and temporal variabilities of the data, they may not be well-suited for predictive modeling applications. To overcome this problem, MUSCAT incorporates the multi-tensor decomposition process directly into a supervised learning framework. Specifically, it assumes that the prediction for location s at time t is given by the weighted average of the predictions generated from models trained on data at different scales, i.e., $\hat{y}_{s,t} = \sum_l^L \alpha_l \hat{y}_{s,t}^{(l)}$, where α_l is the weight associated with the prediction function of the l-th scale. The weights are assumed to form a probability simplex that satisfies $\alpha_l \geq 0$ and $\sum_l^L \alpha_l = 1$.

Furthermore, the prediction model at a given scale l is given by an ensemble of spatial and temporal models:

$$\hat{y}_{s,t}^{(l)} = {\mathbf{x}_{s,t}^{(l)}}^T \Big[\sum_k^K \mathbf{A}_{s,k} \mathbf{w}_k^{(l)} + \sum_k^K \mathbf{B}_{t,k} \mathbf{v}_k^{(l)} \Big], \quad (2)$$

where $\mathbf{x}_{s,t}^{(l)}$ denotes the feature vector for location s and time t at scale l, $\mathbf{w}_k^{(l)}$ and $\mathbf{v}_k^{(l)}$ are parameters of the spatial and temporal prediction models for the k-th latent factor, while $\mathbf{A}_{s,k}$ and $\mathbf{B}_{t,k}$ are the scalar coefficients for the k-th spatial and temporal latent factors associated with location s and time t, respectively. Note that we enforce the constraint that the spatial and temporal latent factors are invariant across all scales, i.e., $\forall l : \mathbf{A}^{(l)} = \mathbf{A}, \mathbf{B}^{(l)} = \mathbf{B}$.

Our framework to simultaneously uncover the latent factors and derive the parameters for the spatial and temporal prediction models can be formalized as the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha},\mathbf{W},\mathbf{V},\mathbf{A},\mathbf{B},\mathbf{C}} \quad & \mathcal{F}(\boldsymbol{\alpha}, \mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{B}, \mathbf{C}) \\ = \quad & \frac{1}{2} \Big\| \sum_l^L \alpha_l \hat{\mathbf{Y}}^{(l)} - \mathbf{Y} \Big\|_F^2 \\ + \quad & \frac{\lambda}{2} \sum_l^L \frac{1}{vol(l)} \Big\| \boldsymbol{\mathcal{X}}^{(l)} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}^{(l)}]\!] \Big\|_F^2 \\ + \quad & \beta \sum_l^L \Big\| [\{\mathbf{W}^{(l)}, \mathbf{V}^{(l)}, \mathbf{C}^{(l)}\}, \mathbf{A}, \mathbf{B}] \Big\|_1 \quad (3) \\ \text{s.t.} \quad & \forall l : \alpha_l \geq 0 \ \text{ and } \sum_l^L \alpha_l = 1 \end{aligned}$$

where $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{C}$ denote the set of $\mathbf{W}^{(l)}$, $\mathbf{V}^{(l)}$ and $\mathbf{C}^{(l)}$ for $l = 1, ..., L$ respectively for notation simplicity, and $\hat{\mathbf{Y}}^{(l)}$ corresponds to the predictions generated by the spatio-temporal models at scale l, as shown in Equation (2). We also use $\|[\{\mathbf{W}^{(l)}, \mathbf{V}^{(l)}, \mathbf{C}^{(l)}\}, \mathbf{A}, \mathbf{B}]\|_1$ to denote the ℓ_1 norm regularization term for $\mathbf{W}^{(l)}$, $\mathbf{V}^{(l)}$, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}^{(l)}$, respectively.

3.3 Incremental Learning

Optimizing Eq. (3) can be very expensive for large spatio-temporal data. The problem is further exacerbated by the fact that the data may grow over time. Learning the model from scratch whenever there are new data available is not a feasible solution. In this section, we present an efficient algorithm to learn the model parameters and latent factors incrementally over space or time without requiring the old data to reside in memory.

Let $\mathcal{D}^{\text{new}} = (\mathcal{X}^{\text{new}}, \mathbf{Y}^{\text{new}})$ denote the set of new observations and $\tilde{\boldsymbol{\Pi}} = \{\tilde{\alpha}, \tilde{\mathbf{W}}, \tilde{\mathbf{V}}, \tilde{\mathbf{C}}, \tilde{\mathbf{A}}, \tilde{\mathbf{B}}\}$ be the previous set of model parameters and latent factors estimated by the algorithm. For incremental learning, the model parameters and latent factors are updated by optimizing the following objective function:

$$\min_{\boldsymbol{\Pi}} \ \mathcal{Q}(\boldsymbol{\Pi}, \tilde{\boldsymbol{\Pi}}) \quad = \quad \mathcal{F}(\boldsymbol{\Pi}; \mathcal{D}^{\text{new}}) + \eta \Gamma(\boldsymbol{\Pi}, \tilde{\boldsymbol{\Pi}})$$

where $\mathcal{F}(\boldsymbol{\Pi}; \mathcal{D}^{\text{new}})$ is given by Eq.(3) and

$$\begin{aligned} \Gamma(\boldsymbol{\Pi}, \tilde{\boldsymbol{\Pi}}) \quad = \quad & \frac{1}{2} \sum_l^L \Big((\alpha_l - \tilde{\alpha_l})^2 + \|\mathbf{W}^{(l)} - \tilde{\mathbf{W}}^{(l)}\|_F^2 \\ + \quad & \|\mathbf{V}^{(l)} - \tilde{\mathbf{V}}^{(l)}\|_F^2 + \|\mathbf{C}^{(l)} - \tilde{\mathbf{C}}^{(l)}\|_F^2 \Big) \\ + \quad & \|\mathbf{A} - \tilde{\mathbf{A}}\|_F^2 + \ \|\mathbf{B} - \tilde{\mathbf{B}}\|_F^2, \end{aligned}$$

is a regularization term to ensure smoothness of the model by controlling the amount of information to be retained from the previous model. Our formulation enables the update formula to be applied when augmented with data from new locations or when data for a new time period become available. We consider the former as *incremental learning over space* and the latter as *incremental learning over time*.

Incremental Learning over Space

Let T be the current time step and S be the number of locations with training data at time T. Without loss of generality, we assume that the model is updated with new data one location at a time. Furthermore, the historical data for the new location is assumed to be available from time t_0 to T. If the location has only one historical observation, then $t_0 = T$. For brevity, we denote the multi-scale tensor data for the new location as $\boldsymbol{\mathcal{X}}_{S+1}^{(l)} \in \Re^{1\times(T-t_0+1)\times d_l}$ for $l = 1, ..., L$, and $d = \sum_{l=1}^{L} d_l$. With the addition of the new data, the spatial latent factors need to be updated from its previous matrix $\tilde{\mathbf{A}}_S \in \Re^{S\times K}$ to $\mathbf{A}_{S+1} \in \Re^{(S+1)\times K} = [\mathbf{A}_S;\ \mathbf{a}_{S+1}^T]$. We further assume that the spatial latent factors for other locations are unaffected by the addition of the new location, i.e., $\mathbf{A}_S = \tilde{\mathbf{A}}_S$. However, the latent factors for other modes of the tensor ($\mathbf{B}$ and $\mathbf{C}^{(l)}$) as well as the parameters of the prediction models (α_l, $\mathbf{W}^{(l)}$ and $\mathbf{V}^{(l)}$) can be affected by the addition of the new data.

Let $\epsilon_{S+1,t} = \sum_l^L \alpha_l \mathbf{x}_{S+1,t}^{(l)}{}^T(\mathbf{W}^{(l)}{}^T\mathbf{a}_{S+1} + \mathbf{V}^{(l)}{}^T\mathbf{b}_t) - y_{S+1,t}$ be the prediction error at time t. Our objective function for incremental learning over space is:

$$\begin{aligned}
&\min_{\mathbf{\Pi}}\ \mathcal{Q}(\mathbf{\Pi}, \tilde{\mathbf{\Pi}}) \\
=\ & \frac{1}{2}\sum_{t=t_0}^{T}\epsilon_{S+1,t}^2 + \frac{\eta_1}{2}\Gamma(\mathbf{\Pi}, \tilde{\mathbf{\Pi}}) \\
+\ & \frac{\lambda_1}{2}\sum_{l}^{L}\frac{1}{Td_l}\left\|\boldsymbol{\mathcal{X}}_{S+1}^{(l)} - [\![\mathbf{a}_{S+1}^T, \mathbf{B}, \mathbf{C}^{(l)}]\!]\right\|_F^2 \\
+\ & \beta_1\|[\{\mathbf{W}^{(l)}, \mathbf{V}^{(l)}, \mathbf{C}^{(l)}\}, \mathbf{a}_{S+1}, \mathbf{B}]\|_1 \\
\text{s.t.}\quad & \forall l : \alpha_l \geq 0 \ \text{ and } \sum_l^L \alpha_l = 1
\end{aligned}$$

where $\mathbf{a}_{S+1}$ is a column vector that represents the spatial latent factors for the new location. The smoothness parameter η_1 determines the extent to which the previous model parameters should be retained.

An alternating minimization strategy is used to solve the optimization problem, and each subproblem is solved by the proximal gradient descent method [Parikh and Boyd, 2014]. The parameters are updated iteratively by calculating the gradient on the smooth part of the objective function, and then apply the soft-thresholding operator to determine its next value. The step size can be found using a line search algorithm. We omit the details of the gradient calculation due to space limitation.

Incremental Learning over Time

MUSCAT performs incremental learning over time when there are new observations available at the new timestamp. Let S be the number of locations and T be the current time. We denote the newly acquired data for all S stations at time $T+1$ as $\{\boldsymbol{\mathcal{X}}_{T+1}^{(1)}, \boldsymbol{\mathcal{X}}_{T+1}^{(2)}, \cdots, \boldsymbol{\mathcal{X}}_{T+1}^{(L)}\}$, where each $\boldsymbol{\mathcal{X}}_{T+1}^{(l)} \in \Re^{S\times 1\times d_l}$ corresponds to the data at scale l and $d = \sum_{l=1}^{L} d_l$. We use this information to update the temporal latent factors from $\tilde{\mathbf{B}}_T \in \Re^{T\times K}$ to $\mathbf{B}_{T+1} \in \Re^{(T+1)\times K} = [\mathbf{B}_T;\ \mathbf{b}_{T+1}^T]$. Similar to the strategy for incremental learning over space, we assume the new data for time $T+1$ does not affect previous temporal latent factors: $\mathbf{B}_T = \tilde{\mathbf{B}}_T$.

Incremental learning over time is implemented via the following two steps. First, we learn the temporal latent factor $\mathbf{b}_{T+1}$ based on the predictor variables for the new time period before the target variable is observed. Next, the 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 $\mathbf{b}_{T+1}$. The objective function for updating $\mathbf{b}_{T+1}$ is:

$$\min_{\mathbf{b}_{T+1}} \mathcal{Q}(\mathbf{b}_{T+1}) = \frac{\lambda_2}{2}\sum_{l}^{L}\frac{1}{Sd_l}\|\boldsymbol{\mathcal{X}}_{T+1}^{(l)} - [\![\mathbf{A}, \mathbf{b}_{T+1}^T, \mathbf{C}^{(l)}]\!]\|_F^2$$

Note that $\mathbf{A}$ and $\{\mathbf{C}^{(l)}\}$ correspond to the matrices obtained from the previous update. The predictions for the target variable are performed using $\mathbf{b}_{T+1}$ and other model parameters from previous update.

Step 2: Updating model parameters and latent factors after observing target variable. After observing the true values of the target variable at time $T+1$, the model parameters and other latent factors are updated by minimizing the following objective function:

$$\begin{aligned}
&\min_{\mathbf{\Pi}}\ \mathcal{Q}(\mathbf{\Pi}, \tilde{\mathbf{\Pi}}) \\
=\ & \frac{1}{2}\sum_{s}^{S}\Bigg[\sum_{l}^{L}\alpha_l \mathbf{x}_{s,T+1}^{(l)}{}^T(\mathbf{W}^{(l)}{}^T\mathbf{a}_s + \mathbf{V}^{(l)}{}^T\mathbf{b}_{T+1}) \\
& \qquad -y_{s,T+1}\Bigg]^2 + \frac{\eta_2}{2}\Gamma(\mathbf{\Pi}, \tilde{\mathbf{\Pi}}) \\
+\ & \frac{\lambda_2}{2}\sum_{l}^{L}\frac{1}{Sd_l}\|\boldsymbol{\mathcal{X}}_{T+1}^{(l)} - [\![\mathbf{A}, \mathbf{b}_{T+1}^T, \mathbf{C}^{(l)}]\!]\|_F^2 \\
+\ & \beta_2(\|\{\mathbf{W}^{(l)}, \mathbf{V}^{(l)}, \mathbf{C}^{(l)}\}, \mathbf{A}, \mathbf{b}_{T+1}\|_1) \\
\text{s.t.}\quad & \forall l : \alpha_l \geq 0 \ \text{ and } \sum_l^L \alpha_l = 1
\end{aligned}$$

4 Experimental Evaluation

This section describes the extensive experiments performed to demonstrate the effectiveness of MUSCAT when applied to a multi-scale climate dataset.

4.1 Climate Data

The climate data used in our experiments has three spatial resolutions. At the finest scale, monthly climate data are obtained for more than 300 weather stations from the United States Historical Climatology Network (USHCN)[1]. Four variables—maximum (tmax), minimum (tmin), mean (tmean) temperature and precipitation (prcp)—are selected as response variables for

[1] `http://cdiac.ornl.gov/epubs/ndp/ushcn/ushcn.html`

Table 2: Number of weather stations and grid cells for each response variable.

	USHCN	NARR	NCEP
tmax	357	350	146
tmin	341	336	144
tmean	333	328	143
prcp	790	635	159

our prediction task. We train a prediction model for each response variable separately.

We use two gridded climate datasets, NARR and NCEP reanalysis, to create the predictor variables. NARR[2], which stands for the North American regional reanalysis dataset, has a spatial resolution of 0.3° (32 km) whereas NCEP reanalysis[3] has a coarser resolution of 2.5°. Nine variables from NARR (acpcp, air.2m, dlwrf, dswrf, lftx4, prate, prmsl, pr_wtr, and rhum) along with seven variables from NCEP reanalysis (cprat.sfc, dlwrf.sfc, dswrf.sfc, prate.sfc, tmax.2m, tmin.2m, and lftx.sfc) are chosen as predictor variables.

The monthly climate data span a 30-year period between January 1985 to November 2015. Table 2 shows the number of stations and grid cells, which may vary for each response variable since we discard stations with missing values. The time series for the predictor and response variables are deseasonalized (by subtracting each monthly value from the mean value of its corresponding month) and subsequently standardized.

4.2 Experimental Setup

The 30-year climate dataset is divided into 3 partitions. We first incrementally build the models using training data from the first 10 years (1985-1994) and then apply the models to validation data from the next 10 years (1995-2004). After tuning the model hyperparameters using the validation set, we apply the chosen models to data from the last 10 years (2005-2015), which serve as our test set. Note that the number of weather stations associated with the training, validation, and test sets may vary depending on when each station was introduced into the incremental learning process. Initially, we randomly choose 100 weather stations as our starting locations. At each step of the incremental learning process, we randomly add either a new station or a new time period to update the latent factors and the ensemble of prediction models.

We used the mean absolute error (MAE) metric to evaluate the performance of various algorithms:

$$\text{MAE} = \frac{\sum_{s=1}^{S}\sum_{t=t_s}^{T}|y_{s,t} - \hat{y}_{s,t}|}{\sum_{s=1}^{S}\sum_{t=t_s}^{T}1}$$

Note that the MAE is calculated for each station s starting from its corresponding test period t_s. For example, if a station was introduced into the incremental learning formulation during the training or validation period, then its MAE is calculated for the entire 10-year test period. However, if the station was first introduced during the testing period, its MAE is computed starting from the time it was introduced. The incremental learning process is repeated 10 times, each time with a different initialization and random ordering of weather stations. Results are reported based on the average performance over the 10 trials.

4.3 Baseline Algorithm

We compare the performance of MUSCAT against the following three incremental learning algorithms.

1. **STL** (Single Task Learning): A linear model is incrementally trained at each location using the stochastic gradient descent algorithm. When a new weather station is introduced during the incremental learning process, its model parameters are initialized randomly and subsequently updated as new observation data become available.
2. **ALTO**: This is variation of the spatio-temporal multi-task learning algorithm proposed in [Yu *et al.*, 2015] which was designed to build models for multiple response variables simultaneously.
3. **WISDOM**: This is a recent spatio-temporal multi-task learning approach [Xu *et al.*, 2016b] that applies tensor decomposition on the data but does not distinguish between variables from different scales.

4.4 Experimental Results

We designed our experiments to (1) compare the performance of MUSCAT against the baseline methods, (2) assess the value of using multi-scale data, and (3) analyze the significance of the latent factors.

Comparison against Baseline Methods

Table 3 shows the mean and standard deviation of MAE after 10 trials for each method. The results suggest that WISDOM and MUSCAT significantly outperform STL and ALTO in all four datasets, which suggests the advantages of using a multi-task learning approach based on tensor decomposition. In addition, MUSCAT outperforms WISDOM on all four datasets, which shows the benefits of using our approach to factorize the multi-scale tensors jointly, instead of factorizing a single tensor with concatenated features from all scales, which is the approach used in WISDOM. A more detailed analysis given in Table 5 shows that MUSCAT outperforms all other competing methods in at least 70% of the stations for all 4 response variables.

Value of Multi-scale Data

To determine the value of using multi-scale data, we consider the following two variations of MUSCAT: (1) **MUSCAT-S1**, which uses only predictor variables from NCEP, and (2) **MUSCAT-S2**, which uses only predictor variables from NARR. Once again, the results

[2] `https://www.esrl.noaa.gov/psd/data/gridded/data.narr.html`

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

Table 3: Performance comparison between MUSCAT and the baseline methods on 4 response variables.

	tmax	tmin	tmean	prcp
STL	0.4422 ± 0.0016	0.4412 ± 0.0020	0.4141 ± 0.0018	0.5446 ± 0.0012
ALTO	0.5854 ± 0.0064	0.5687 ± 0.0031	0.5656 ± 0.0053	0.5806 ± 0.0051
WISDOM	0.3543 ± 0.0155	0.4001 ± 0.0075	0.3850 ± 0.0236	0.4212 ± 0.0054
MUSCAT	**0.3212** ± 0.0074	**0.3454** ± 0.0065	**0.2844** ± 0.0112	**0.4115** ± 0.0023

Table 4: Comparison between two variations of MUSCAT that utilize data from a single scale only.

	tmax	tmin	tmean	prcp
WISDOM	0.3543 ± 0.0155	0.4001 ± 0.0075	0.3850 ± 0.0236	0.4212 ± 0.0054
MUSCAT-S1	0.5492 ± 0.0700	0.4328 ± 0.0115	0.4543 ± 0.0393	0.6194 ± 0.0026
MUSCAT-S2	0.3910 ± 0.0183	0.4094 ± 0.0186	0.4350 ± 0.0532	0.4208 ± 0.0051
MUSCAT	**0.3212** ± 0.0074	**0.3454** ± 0.0065	**0.2844** ± 0.0112	**0.4115** ± 0.0023

Table 5: Number of weather stations that MUSCAT outperforms other methods for each response variable.

	# stations	STL	ALTO	WISDOM
tmax	357	355	315	350
tmin	341	330	321	339
tmean	333	331	313	333
prcp	790	780	739	553

shown in Table 4 suggest that MUSCAT outperforms WISDOM, MUSCAT-S1, and MUSCAT-S2 on all four datasets. For precipitation prediction, MUSCAT-S2 outperforms WISDOM, which indicates that augmenting the predictor variables from the coarsest scale may degrade the model performance. Nonetheless, MUSCAT still achieves the lowest MAE because it can learn the appropriate weight (α) for combining the predictions from NARR and NCEP reanalysis datasets. To measure the relative influence of data at different scales on the performance of MUSCAT, we examine the mean values of the parameters α_1 and α_2 over the 10 trials. The results given in Table 6 suggest that α_2, which is the weight associated with the finer-level predictors, has a consistently higher weight than α_1, the weight associated with coarser-level predictors on all four datasets.

Table 6: Mean of α_1 and α_2 for the climate datasets over 10 trials.

	tmax	tmin	tmean	prcp
α_1	0.2876	0.3651	0.3360	0.0933
α_2	**0.7124**	**0.6349**	**0.6640**	**0.9067**

4.5 Analysis of Latent Factors

Figure 1 shows the spatial distribution of the latent factors for precipitation data. For each spatial latent factor, we plot its top 20% most influential stations on the map. The results suggest that the latent factors exhibit some spatially coherent patterns. For example, the first latent factor is more dominant on the eastern part of the United States, while the third and fifth latent factors are more influential on the northwest part of the country.

We also examine the temporal latent factors derived by MUSCAT by computing their pairwise absolute correlation against some of the well-known climate indices. The results are shown in Figure 2. The correlation values

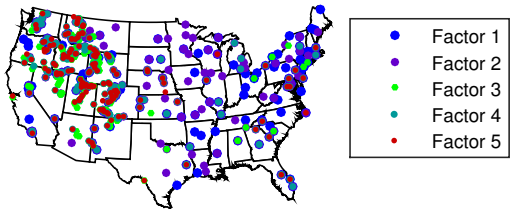
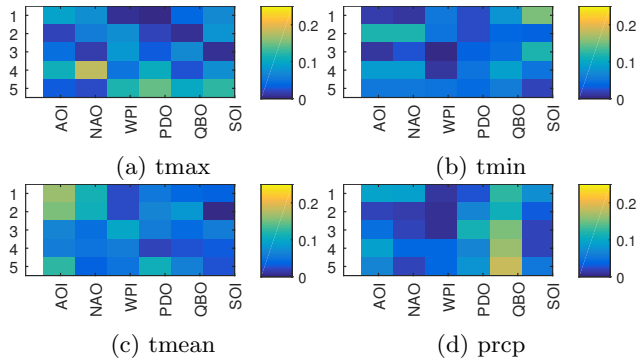


Figure 1: Spatial distribution of the spatial latent factors learned by MUSCAT for precipitation data (Figure is best viewed in color)

are not that high, which is not surprising as the study region is limited to the United States only.



(a) tmax (b) tmin

(c) tmean (d) prcp

Figure 2: Correlations between known climate indices and the temporal latent factors derived by MUSCAT .

5 Conclusion

This paper presents a multi-scale multi-task learning framework for geospatio-temporal data by employing a supervised multi-tensor decomposition approach. The framework enables the multi-scale relationships to be harnessed by enforcing a constraint on the consistency between the spatial and temporal latent factors derived from the multi-scale geospatio-temporal data. An incremental learning algorithm over space and time is then proposed to efficiently learn the weights of the model. Experiments performed on a real-world multi-scale climate dataset demonstrate the effectiveness of proposed method compared to several baseline algorithms.

References

[Acar *et al.*, 2011] Evrim Acar, Tamara G. Kolda, and Daniel M. Dunlavy. All-at-once optimization for coupled matrix and tensor factorizations. In *MLG'11: Proceedings of Mining and Learning with Graphs*, August 2011.

[Bellocchio *et al.*, 2012] F. Bellocchio, S. Ferrari, V. Piuri, and N.A. Borghese. Hierarchical approach for multiscale support vector regression. *IEEE TNNLS*, 23(9):1448–1460, September 2012.

[Bertasius *et al.*, 2015] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *CVPR*, pages 4380–4389, 2015.

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

[Eigen *et al.*, 2014] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, pages 2366–2374, 2014.

[Ermiş *et al.*, 2015] Beyza Ermiş, Evrim Acar, and A. Taylan Cemgil. Link prediction in heterogeneous data via generalized coupled tensor factorization. *Data Mining and Knowledge Discovery*, 29(1):203–236, Jan 2015.

[Gonçalves *et al.*, 2016] André R. Gonçalves, Fernando J. Von Zuben, and Arindam Banerjee. Multi-task sparse structure learning with gaussian copula models. *JMLR*, 17(1):1205–1234, January 2016.

[Gonçalves *et al.*, 2017] André R. Gonçalves, Arindam Banerjee, and Fernando J. Von Zuben. Spatial projection of multiple climate variables using hierarchical multitask learning. In *AAAI*, pages 4509–4515, 2017.

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

[Miller *et al.*, 2015] Bradley A. Miller, Sylvia Koszinski, Marc Wehrhan, and Michael Sommer. Impact of multi-scale predictor selection for modeling soil properties. *Geoderma*, 239240:97 – 106, 2015.

[Neverova *et al.*, 2014] Natalia Neverova, Christian Wolf, Graham W. Taylor, and Florian Nebout. Multi-scale deep learning for gesture detection and localization. In *ECCV Workshops*, pages 474–490, 2014.

[Nounou and Nounou, 2010] Mohamed N. Nounou and Hazem N. Nounou. Multiscale latent variable regression. *International Journal of Chemical Engineering*, 2010, 2010.

[Parikh and Boyd, 2014] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, January 2014.

[Steinbach *et al.*, 2003] Michael Steinbach, Pang-Ning Tan, Vipin Kumar, Steven Klooster, and Christopher Potter. Discovery of climate indices using clustering. In *KDD*, pages 446–455, 2003.

[Sun *et al.*, 2008] Jimeng Sun, Dacheng Tao, Spiros Papadimitriou, Philip S. Yu, and Christos Faloutsos. Incremental tensor analysis: Theory and applications. *TKDD*, 2(3):11:1–11:37, 2008.

[Walder *et al.*, 2008] Christian Walder, Kwang In Kim, and Bernhard Schölkopf. Sparse multiscale gaussian process regression. In *ICML*, pages 1112–1119, 2008.

[Xu *et al.*, 2014] Jianpeng Xu, Pang-Ning Tan, and Lifeng Luo. ORION: Online Regularized multI-task regressiON and its application to ensemble forecasting. In *ICDM*, pages 1061–1066, 2014.

[Xu *et al.*, 2016a] Jianpeng Xu, Pang-Ning Tan, Lifeng Luo, and Jiayu Zhou. Gspartan: a geospatio-temporal multi-task learning framework for multi-location prediction. In *SDM*, pages 657–665, 2016.

[Xu *et al.*, 2016b] Jianpeng Xu, Jiayu Zhou, Pang-Ning Tan, Xi Liu, and Lifeng Luo. Wisdom: Weighted incremental spatio-temporal multi-task learning via tensor decomposition. In *IEEE Big Data*, pages 522–531, Dec 2016.

[Yu *et al.*, 2015] Rose Yu, Dehua Cheng, and Yan Liu. Accelerated online low rank tensor learning for multivariate spatiotemporal streams. In *ICML*, volume 37, pages 238–247, 2015.

[Zhao and Du, 2016] Wenzhi Zhao and Shihong Du. Learning multiscale and deep representations for classifying remotely sensed imagery. *JPRS*, 113:155 – 165, 2016.

[Zhao *et al.*, 2015] Liang Zhao, Qian Sun, Jieping Ye, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Multi-task learning for spatio-temporal event forecasting. In *KDD*, pages 1503–1512, 2015.

[Zhou *et al.*, 2016] Shuo Zhou, Xuan Vinh Nguyen, James Bailey, Yunzhe Jia, and Ian Davidson. Accelerating Online CP Decompositions for Higher Order Tensors. In *KDD*, pages 1375–1384, 2016.