Nonparametric Mixture of Sparse Regressions on Spatio-Temporal Data – An Application to Climate Prediction

Yumin Liu
Northeastern University
Boston, USA
yuminliu@ece.neu.edu

Junxiang Chen
Northeastern University
Boston, USA
jchen@ece.neu.edu

Auroop Ganguly
Northeastern University
Boston, USA
a.ganguly@neu.edu

Jennifer Dy
Northeastern University
Boston, USA
jdy@ece.neu.edu

ABSTRACT

Climate prediction is a very challenging problem. Many institutes around the world try to predict climate variables by building climate models called General Circulation Models (GCMs), which are based on mathematical equations that describe the physical processes. The prediction abilities of different GCMs may vary dramatically across different regions and time. Motivated by the need of identifying which GCMs are more useful for a particular region and time, we introduce a clustering model combining Dirichlet Process (DP) mixture of sparse linear regression with Markov Random Fields (MRFs). This model incorporates DP to automatically determine the number of clusters, imposes MRF constraints to guarantee spatio-temporal smoothness, and selects a subset of GCMs that are useful for prediction within each spatio-temporal cluster with a spike-and-slab prior. We derive an effective Gibbs sampling method for this model. Experimental results are provided for both synthetic and real-world climate data.

CCS CONCEPTS

• **Computing methodologies** → **Mixture modeling**; • **Applied computing** → **Environmental sciences**.

KEYWORDS

Markov random field, Dirichlet process, Spike-and-slab, Spatio-temporal, Climate

ACM Reference Format:
Yumin Liu, Junxiang Chen, Auroop Ganguly, and Jennifer Dy. 2019. Nonparametric Mixture of Sparse Regressions on Spatio-Temporal Data – An Application to Climate Prediction. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330692

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330692

1 INTRODUCTION

The impact of climate change is becoming more and more prominent. As a result, extreme weather events such as droughts, floods, heat waves, snow storms and hurricanes, are causing more and more damages to human society. This calls for an effective way to predict the impact of climate change in the future to forecast natural hazards and minimize damages. One way to achieve prediction is to build climate models based on physics and mathematics to simulate the dynamics of the atmosphere and produce predictive climate variables such as temperature, precipitation and so on. During past decades, climate models have been developed to forecast weather in days, weeks or months [27, 28].

General Circulation Models (GCMs) are climate models that simulate climate processes over ocean and/or atmosphere. As GCMs can simulate climate variables over a long time period (hundreds of years) and over the whole globe under different climate conditions (such as different carbon dioxide emission level scenarios), they have served as important tools to analyze climate change. GCMs can provide a reasonable prediction of climate change in a large scale (global scale) [14], and have advantages in simulating circulation patterns and long time projections [8, 32].

However, GCMs have some weaknesses. It is difficult to calibrate the simulated variables with observed variables. A tiny difference in boundary condition or initialization may cause a large difference in outcomes [19] due to the "butterfly effect". As a result, the GCM simulated data often do not agree with the observed data, sometimes even very far away. Thus it's not practical to use just one GCM. There are many GCMs developed by different research institutes, each with different models, boundary conditions and initialization, and therefore, with different specialties. While GCMs have consensus on large scale long time trend of climate change, they have discrepancies in smaller scales. Specifically, some GCMs maybe more accurate in some regions or time periods, but less accurate in others. To deal with this problem, existing work often divide the studied area into several regions and then deal with the regions and then do the regression within each region [13, 20, 34]. But this

division may be incorrect or improper. Besides, they assume that within each region, the relation between GCMs and observation is time invariant, which is questionable.

Motivated by the problem associated with GCMs above, we propose a method that combines Markov Random Fields (MRF), Dirichlet Process (DP) and spike-and-slab prior to predict precipitation based on the provided GCMs predictors. Our proposed model can automatically divide the spatio-temporal data into clusters based on the specialties of GCMs, automatically determine the number of clusters, and discover the most relevant GCMs that used for prediction in each cluster. The main advantage of our method over existing multi-GCM methods lies in three aspects. First, we don't have to manually divide the studied areas into several regions, instead we let the data speak for themselves. Second, we extend the spatial division (clustering) into spatio-temporal division (clustering), which can deal with time variant relation within regions. Third, unlike some clustering methods such as k-means that performs clustering based on just observation, our method performs the clustering based on the relation of GCM and observation, which is suitable for the problem we want to solve and more accurate. The climate is a dynamic system which changes over time. Two different locations may have different situations in the past but evolve to a similar situation in the future, or vice versa. Our proposed scheme of spatio-temporal clustering can handle this problem.

The rest of the paper is organized as follows: Section 2 presents an overview of the related work. In Section 3, we summarize our contributions. The building blocks of our model are introduced in Section 4. We describe our model in Section 5 and perform experiments and analysis in Section 6. Section 7 gives the conclusion.

2 RELATED WORK

There exist some statistical approaches and machine learning approaches that analyze GCMs for various purposes [2, 12, 13, 20, 22, 24, 25, 34]. O'Gorman and Dwyer [24] explored the implication of incorporating machine learning approaches in climate models. Smith et al. [34] proposed a Bayesian method to combine different GCMs and analyzed the uncertainty of climate model projections of temperature. Kumar et al. [20] compared the trend and variability of Coupled Model Intercomparison Project (a collaborative framework for studying GCMs) Phase 3 (CMIP3) and CMIP5 data by pairwise and multi-model comparisons for 11 models. Greene et al. [13] studied the regional temperature change projections using Bayesian linear model of multiple GCMs. O'Gorman and Schneider [25] analyzed the change of precipitation extremes using multiple models in CMIP3. Gonçalves et al. [12] proposed a multitask learning based method called Multi-task Sparse Structure Learning (MSSL) which regards each location as a task and estimate the sparse task parameters matrix and task relationship structure using L_1 regularization. Bahadori et al. [2] formulated the spatio-temporal data as tensors and imposed low rank regularization and spatial Laplacian regularization on the tensor to account for shared structures in variables. There exist some methods to combine multiple GCMs [2, 12, 34]; however, none address learning spatio-temporal clusters. Furthermore, our approach also simultaneously identifies which GCMs are more useful for each cluster.

We are inspired by the success of applying MRF and DP in the analysis of time series and spatial data. Basu et al. [3] used Hidden MRF in semi-supervised clustering to incorporate domain expert knowledge. Orbanz and Buhmann [26] proposed a method combining MRF and DP to impose spatial smoothness constraints in image segmentation and to automatically determine the number of segments. The method did not involve temporal constraint which needs a careful design together with spatial constraint. Ross and Dy [33] used MRF to represent must-link and cannot-link constraints in a disease subtyping problem. They also used DP to learn potentially meaningful disease trajectories in a nonparametric way. But it must specify pair-wise constraint for must-link and cannot-link data points. This maybe impossible or improper in the climate field. More recently, Prendes et al. [29] introduced a model based on DP combined with MRF to detect changes between heterogeneous images. There also exist time switching models [4, 5, 10] using Markov models. However, none of these models could simultaneously perform spatial and temporal clustering. Although [16] clustered data in both space and time by proposing a method called Multivariate Spatio-Temporal Clustering, it essentially just applied k-means in the data space. In addition, our model also incorporates the spike-and-slab technique, which is a Bayesian variable selection technique that has been widely used to select variables or features and render sparsity [11, 23].

2.1 Contributions

In this paper we propose a novel model to cluster spatio-temporal data using MRF combined with DP and a spike-and-slab prior. The contributions of our work are: (1) Our model is able to automatically determine the number of clusters in a nonparametric way by embedding DP. (2) The model incorporates spatial and temporal constraints through MRF, which can be flexible to embed domain knowledge and induce smoothness both in space and time. (3) The weight vector of the linear regression within each cluster has a sparse pattern due to spike-and-slab prior, therefore the model can identify the most relevant features for each cluster. (4) We designed an energy function for the MRF to simultaneously incorporate spatial and temporal constraints. (5) We derive a Gibbs sampling method for this model. (6) We apply the model to learn spatio-temporal clusters of GCMs for predicting precipitation, and the results show efficacy of the model and provide new insights on GCMs for climate prediction.

3 BACKGROUND

In this section we will briefly introduce the three main building blocks of our model, *i.e.*, Markov Random Field (MRF), Dirichlet Process (in the view of the Chinese Restaurant Process(CRP)), and spike-and-slab prior for variable selection.

3.1 Markov Random Field

In the graph representation, MRF is an undirected graph consisting of a set of nodes and edges that satisfy the Markov property [30]. The nodes represent the random variables, while the edges represent dependencies between the random variables. Two nodes are called neighbours if they have an edge between them. One property of MRF is that a variable is conditionally independent of all other variables given all its neighbours. Let us denote S to be the set of

all random variables in the MRF,

$$S = \{s_n | n = 1, 2, ..., N\} \tag{1}$$

according to the Hammersley-Clifford theorem [21], an MRF can be equivalently characterized by a Gibbs distribution. The probability distribution is

$$p(S) = \frac{1}{Z}\exp\Big\{-\sum_{c\in C} H(S_c)\Big\} \tag{2}$$

where Z is a normalization constant called the partition function, and C is the set of all cliques. A clique c is defined as a subset of nodes in the MRF where every pair of distinct nodes are neighbours. $H(S_c)$ is the energy function or cost function over clique $c \in C$. The form of $H(S_c)$ depends on the local configuration on clique c.

3.2 Dirichlet Process via the Chinese Restaurant Process

There are several alternative approaches to view Dirichlet Process, including *the Pólya urn scheme*, *the stick-breaking process* and *the Chinese Restaurant Process (CRP)* [1]. Here we use CRP to explain DP.

Imagine a process in which customers go into a Chinese restaurant with an infinite number of tables, which are labeled as $1, 2, ..., k, ...$ according to the order they are occupied. Each table represents a cluster and each customer represents a data point. A customer sitting at a table means that the data point is assigned to the cluster. A new customer can either choose an empty table with probability proportional to a constant scalar α or an occupied table with probability proportional to the number of people that are already sitting at that table. Let K_n be the number of occupied tables (the existing clusters) just before the n-th customer arrives, and let s_n be the label of the table at which the n-th customer will sit at (*i.e.*, s_n is the cluster label of the n-th data point $\boldsymbol{x}_n$), then the probability of n-th customer to choose table k (*i.e.*, the probability of $\boldsymbol{x}_n$ to be in the cluster k) is:

$$p(s_n|s_1,\cdots,s_{n-1}) = \begin{cases} \sum_{k=1}^{K_n} \frac{n_k}{n-1+\alpha}\delta_k(s_n) & s_n \in \{1,\cdots,K_n\} \\ \frac{\alpha}{n-1+\alpha} & s_n = \underbrace{K_n+1}_{\text{new cluster}} \end{cases} \tag{3}$$

where n_k is the number of data points already in cluster k, we have $n-1 = \sum_{k=1}^{K_n} n_k$, and

$$\delta_k(s_n) = \begin{cases} 1 & s_n = k, \\ 0 & s_n \neq k, \end{cases} \quad k \in \{1, 2, \ldots, K_n\}. \tag{4}$$

The CRP exhibits a clustering effect. A new customer is more likely to choose a table which many customers have already sitting at. Thus the data will automatically form several clusters.

3.3 Spike-and-Slab

In a linear regression model $y = \boldsymbol{x}\boldsymbol{w} + \epsilon$, where $\boldsymbol{x} \in R^D$, $y \in R$, $\boldsymbol{w} \in R^D$ and $\epsilon \in R$ are predictor, target, weight and noise variables, respectively. The spike-and-slab prior assumes that each element w_d of the weight vector $\boldsymbol{w}$ comes from a mixture of two Gaussian distributions with 0 means and small/large variances (called "spike" and "slab", respectively). We represent whether each element w_d is sampled from either "spike" or "slab" component with a latent binary indicator vector $\boldsymbol{z} = [z_1, \cdots, z_d, \cdots, z_D]^T$. We assign each element of this vector $z_d \in \{0, 1\}$ a prior of Bernoulli distribution and

$$p(\boldsymbol{z}) = \prod_{d=1}^{D} p_d^{z_d}(1-p_d)^{1-z_d} \tag{5}$$

where p_d is defined as $p_d = p(z_d = 1) = 1 - p(z_d = 0)$. Given z_d, the distribution of w_d is defined as

$$w_d|z_d \sim (1-z_d)\mathcal{N}(0, \tau_d^2) + z_d\mathcal{N}(0, c_d^2\tau_d^2) \tag{6}$$

where $\mathcal{N}(0, \tau_d^2)$ and $\mathcal{N}(0, c_d^2\tau_d^2)$ are Gaussian distributions (the "spike" and "slab", respectively). τ_d is small such that, if $z_d = 0$, then w_d is likely to be so small that it can be regarded as 0; and c_d is large such that $c_d\tau_d$ would not be too small and w_d is non-zero if $z_d = 1$. The formula (6) can be reformulated in a multivariate form as follows:

$$\boldsymbol{w}|\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Lambda_z} R \boldsymbol{\Lambda_z}) \tag{7}$$

where $\boldsymbol{R}$ is the prior correlation matrix, and $\boldsymbol{\Lambda_z}$ is a diagonal matrix, $\boldsymbol{\Lambda_z} = diag[c_1^{z_1}\tau_1, \cdots, c_d^{z_d}\tau_d, \cdots, c_D^{z_D}\tau_D]$.

4 OUR FORMULATION

In this section we will describe our model. We assume that the spatio-temporal data can be divided into several clusters (the number of clusters is unknown beforehand), within each cluster there is a sparse linear relationship between the independent valuables and the dependent valuables. Specifically speaking, the relationships between the GCMs and the observations can vary in different locations and time. Some GCMs maybe more accurate in some regions and/or time periods, but less accurate in others. Those spatio-temporal spots with the same relationships form a cluster. We want to find out the clusters and fit a sparse linear regression model for each cluster. In this way we can understand which GCMs are more responsible for a spatio-temporal cluster and get a better prediction. Our model uses DP to automatically determine the number of clusters in a nonparametric way [37]. We incorporate the MRF in our model to impose spatio-temporal constraints for the data points. We also use spike-and-slab prior to attain sparsity for regression weights in each cluster. We use the spike-and-slab prior instead of L1 or L2 norm regularization to induce sparsity because it facilitates the sampling method we use for inference.

The original GCM outputs and observational data are tensors with dimensions of longitude, latitude and time. We transform them into design matrix $\boldsymbol{X}$ and target vector $\boldsymbol{Y}$, respectively. Besides, we keep the location and time information to be used in the MRF constraint. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \ldots, \boldsymbol{x}_N]^T$ be the $N \times D$ design matrix where N and D are the number of data points and dimensions of features (number of GCMs), respectively. Each element of $\boldsymbol{X}$ is a GCM output variable in a particular space and time. To simplify notation, we will include the linear regression bias term (*i.e.*, constant 1s) into the first column of $\boldsymbol{X}$. Let $\boldsymbol{Y} = [y_1, \ldots, y_n \ldots, y_N]^T$ be the $N \times 1$ target vector corresponding to the design matrix. Each element of $\boldsymbol{Y}$ is an observational variable. Given $\boldsymbol{X}$ and $\boldsymbol{Y}$ and location and time information, our goal is to find out the cluster latent labels $\boldsymbol{S}$, the weights $\boldsymbol{W}$ and latent feature indicator $\boldsymbol{Z}$ as defined below.

Let $\boldsymbol{S} = [s_1, \ldots, s_n, \ldots, s_N]^T$ be the $N \times 1$ cluster label vector with elements s_n being one of an integer from 1 to ∞, indicating which cluster the n-th data point belongs to. Since we don't know the

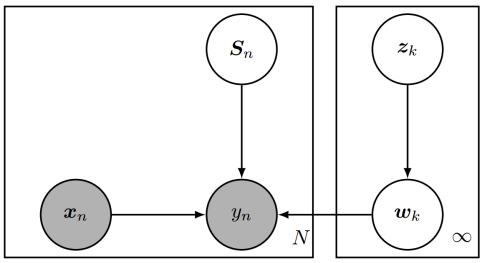


Figure 1: Graphical model representation of the model. The shaded nodes are observed variables while the blank nodes are unobserved variables. The arrows represent dependencies and the rectangular plates mean replica with number of times at the bottom-right corner.

number of clusters K beforehand, we utilize a nonparametric prior, where K is unbounded and allowed to be any value up to infinity. Given data, a finite K will be automatically learned. $\boldsymbol{S}$ has a prior of a DP mixture model combined with a spatio-temporal MRF such that, the model can automatically determine the number of clusters in a nonparametric way by the DP, while it can also incorporate empirical spatio-temporal constraints by the MRF.

Let $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k, \ldots, \boldsymbol{w}_\infty]^T$ be the $\infty \times D$ regression weight matrix, where each row $\boldsymbol{w}_k^T$ is a $1 \times D$ weight vector for the cluster k. We suppose that each $\boldsymbol{w}_k^T$ is independently generated from a spike-and-slab prior.

Let $\boldsymbol{Z} = [z_1, \ldots, z_k, \ldots, z_\infty]^T$ be a binary indicator matrix whose row $\boldsymbol{z}_k^T = [z_{k1}, \cdots, z_{kd}, \cdots, z_{kD}]$ indicates whether a "spike" or "slab" prior is assigned to each element in $\boldsymbol{w}_k$. If $z_{kd} = 0$, then w_{kd} is regarded as 0, meaning that the d-th feature does not contribute to the k-th cluster.

The graphical model of our formulation can be illustrated in Figure 1. Having defined these variables above, the joint distribution of our model is:

$$p(\boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{S}, \boldsymbol{Z}|\boldsymbol{X}) = p(\boldsymbol{Z})p(\boldsymbol{W}|\boldsymbol{Z})p(\boldsymbol{S})p(\boldsymbol{Y}|\boldsymbol{W}, \boldsymbol{S}, \boldsymbol{X}) \quad (8)$$

4.1 Formulation Decomposition

In this section we provide detailed expressions for the components of our joint model in Equation (8).

4.1.1 Priors. In order to get sparsity, we use the spike-and-slab prior introduced in Section 3.3 to induce the priors of $\boldsymbol{Z}$ and $\boldsymbol{W}$ in our model.

Latent Feature Indicator $\boldsymbol{Z}$. Assuming $\boldsymbol{z}_k$ is independent of each other, the prior for $\boldsymbol{Z}$ is the product of the prior for all $\boldsymbol{z}_k$. From Equation (5) we get

$$p(Z) = \prod_{k=1}^{\infty} p(\boldsymbol{z}_k) = \prod_{k=1}^{\infty} \prod_{d=1}^{D} p_{kd}^{z_{kd}} (1 - p_{kd})^{1 - z_{kd}} \quad (9)$$

where p_{kd} is chosen according to the probability that this feature should be included, larger p_{kd} means more features will be selected. We set $p_{kd} = 0.5$ for all k and d, which means we don't have a prior preference over inclusion or exclusion of a GCM.

Regression Weights $\boldsymbol{W}$. The prior for weight matrix $\boldsymbol{W}$ given $\boldsymbol{Z}$ is the product of the priors for the weight vectors of each cluster, following Equation (7) and setting $\boldsymbol{R} = \boldsymbol{I}$ (setting $\boldsymbol{R}$ to be an identity matrix means we assume components in $\boldsymbol{w}_k$ are independent of each other to simplify the model), we get

$$p(\boldsymbol{W}|\boldsymbol{Z}) = \prod_{k=1}^{\infty} p(\boldsymbol{w}_k|\boldsymbol{z}_k) = \prod_{k=1}^{\infty} \mathcal{N}(\boldsymbol{w}_k|\boldsymbol{0}, \Sigma_{0k}) \quad (10)$$

$$\Sigma_{0k} = \boldsymbol{\Lambda}_{\boldsymbol{z}_k} \boldsymbol{\Lambda}_{\boldsymbol{z}_k} \quad (11)$$

$$\boldsymbol{\Lambda}_{\boldsymbol{z}_k} = diag[c_1^{z_{k1}} \tau_1, \ldots, c_d^{z_{kd}} \tau_d, \ldots, c_D^{z_{kD}} \tau_D] \quad (12)$$

where c_d and τ_d are hyperparameters that control the sparsity for weights and as pointed out in Section 3.3 on spike-and-slab, τ_d should be set small so that if $z_d = 0$ then w_d will likely be small and be regarded as 0; and c_d is set large such that $c_d \tau_d$ will not be too small and w_d is non-zero. In our experiment for simplicity we set $c_d = 10.0$ and $\tau_d = 0.01$, which works well; alternatively, one may learn these hyperparameters via cross-validation or empirical Bayes.

Latent Cluster Label $\boldsymbol{S}$. The prior for $\boldsymbol{S}$ consists of two components and it can be written as follows:

$$p(\boldsymbol{S}) = p_1(\boldsymbol{S}) p_2(\boldsymbol{S}) \quad (13)$$

where $p_1(\boldsymbol{S})$ and $p_2(\boldsymbol{S})$ are the CRP term and the MRF term, respectively. The CRP term is given as:

$$p_1(\boldsymbol{S}) = \prod_{n=1}^{N} p_1(s_n|S_{1:n-1}) \quad (14)$$

where $p_1(s_n|S_{1:n-1})$ is the probability that the n-th data point belongs to cluster s_n given the cluster labels of all data points before it, and is given in Equation (3). This CRP term will help to automatically decide the number of clusters due to its clustering effect.

The MRF term uses the pairwise constraints to incorporates spatially and temporally smoothness for the clustering pattern of the data points. From Equation (2) we have

$$p_2(\boldsymbol{S}) = \frac{1}{Z_2} \exp\Big\{ - \sum_{(i,j) \in C} H(s_i, s_j) \Big\} \quad (15)$$

where Z_2 is a normalization constant, and C is the set of two connected data points, *i.e.*, $C = \{(i, j) | i \text{ and } j \text{ are connected (neighbours)}\}$. We define the neighbors of a data point as the data points that are spatially and temporally next to it. $H(\cdot)$ can be viewed as a cost function which penalizes cluster inconsistency among neighbours. We penalize more for points that are closer in space and time. This enforces the clusters to be smooth both spatially and temporally.

$$H(s_i, s_j) = \begin{cases} 0 & s_i \neq s_j \\ -\rho \exp(-\beta d_{ij} - \gamma l_{ij}) & s_i = s_j \end{cases} \quad (16)$$

where β, γ and ρ are scaling factors for space, time and composition, respectively. d_{ij} and l_{ij} are the geological distance and time lag between data points i and j, respectively. The values of ρ, β and γ are chosen by cross-validation for simplicity, with the details described in Section 5.

4.1.2 Likelihood. The linear regression model for a data point $(\boldsymbol{x}, y)$ is $y = \boldsymbol{w}^T\boldsymbol{x} + \epsilon$ where $\boldsymbol{w}$ is one of $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k, \ldots\}$, depending on which cluster the data point belongs to. Assuming that $\epsilon \sim \mathcal{N}(0, \sigma^2)$, where σ is a parameter to be estimated, the linear regression above is equivalent to $y \sim \mathcal{N}(y|\boldsymbol{w}^T\boldsymbol{x}, \sigma^2)$. Although different clusters can have different noise variances, for simplicity we assume all clusters share the same noise variance σ^2. We choose the value of σ^2 by cross-validation. The overall likelihood is

$$p(Y|\boldsymbol{W}, \boldsymbol{S}, \boldsymbol{X}) = \prod_{n=1}^{N}\prod_{k=1}^{\infty}\Big[\mathcal{N}(y_n|\boldsymbol{w}_k^T\boldsymbol{x}_n, \sigma^2)\Big]^{\mathbf{1}(s_n=k)} \tag{17}$$

where the label expression $\mathbf{1}(s_n = k) = 1$ if $s_n = k$; and $\mathbf{1}(s_n = k) = 0$ if $s_n \neq k$.

4.1.3 Joint Distribution. Substituting Equations (9), (10), (13) (14), (15) and (17) to Equation (8), we get the joint distribution of our model as:

$$\begin{aligned}
&p(\boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{S}, \boldsymbol{Z}|\boldsymbol{X})\\
=&p(\boldsymbol{Z})p(\boldsymbol{W}|\boldsymbol{Z})p(\boldsymbol{S})p(\boldsymbol{Y}|\boldsymbol{W}, \boldsymbol{S}, \boldsymbol{X})\\
=&\prod_{k=1}^{\infty}\prod_{d=1}^{D}p_{kd}^{z_{kd}}(1-p_{kd})^{1-z_{kd}}\prod_{k=1}^{\infty}\mathcal{N}(\boldsymbol{w}_k|\boldsymbol{0}, \Sigma_{0k})\prod_{n=1}^{N}p_1(s_n|S_{1:n-1})\\
&\frac{1}{Z_2}\exp\Big\{-\sum_{(i,j)\in C}H(s_i, s_j)\Big\}\prod_{n=1}^{N}\prod_{k=1}^{\infty}\Big[\mathcal{N}(y_n|\boldsymbol{w}_k^T\boldsymbol{x}_n, \sigma^2)\Big]^{\mathbf{1}(s_n=k)}
\end{aligned} \tag{18}$$

4.2 Gibbs Sampling

We use a Markov Chain Monte Carlo (MCMC) sampling method for inference. In particular we use Gibbs Sampling [7] where we alternatively sample each parameter given all other parameters. It is suitable for our model because conditional distributions of all parameters are available in a simple form. In the following we will give the conditional distributions for $\boldsymbol{S}$, $\boldsymbol{W}$ and $\boldsymbol{Z}$. Due to page limitations, here we omit the derivation details and provide the conditional distributions directly.

Conditional Distribution for S. The conditional distribution for each element s_n in $\boldsymbol{S}$ with $n = 1, \cdots, N$ is:

$$p(s_n = k|\boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{X}, \boldsymbol{S}^{(-n)}) \propto \begin{cases} \frac{n_k^{(-n)}}{N-1+\alpha}B_k & s_n = k \in \{1, \cdots, K_n\} \\ \frac{\alpha}{N-1+\alpha}C_k & s_n = \underbrace{K_n + 1}_{\text{new cluster}} \end{cases} \tag{19}$$

where $\boldsymbol{S}^{(-n)} = \{s_1, \cdots, s_{n-1}, s_{n+1}, \cdots, s_N\}$, and $n_k^{(-n)}$ is the number of data points in the k-th cluster excluding the n-th data point, and

$$B_k = \frac{1}{\sqrt{2\pi}\sigma}\exp\Big\{-\sum_{(i,j)\in C}H(s_i, s_j) - \frac{1}{2\sigma^2}(y_n - \boldsymbol{w}_k^T\boldsymbol{x}_n)^2\Big\} \tag{20}$$

$$C_k = \frac{1}{\sqrt{2\pi}\sigma}\frac{|\Sigma|^{1/2}}{|\Sigma_{0k}|^{1/2}}\exp\Big\{-\frac{1}{2}\big(\frac{y_n^2}{\sigma^2} - \bar{\boldsymbol{w}}^T\Sigma^{-1}\bar{\boldsymbol{w}}\big)\Big\} \tag{21}$$

where $\bar{\boldsymbol{w}} = \frac{y_n}{\sigma^2}\Sigma\boldsymbol{x}_n$, $\Sigma^{-1} = \Sigma_{0k}^{-1} + \frac{1}{\sigma^2}\boldsymbol{x}_n\boldsymbol{x}_n^T$, and Σ_{0k} and $H(s_i, s_j)$ are in Equations (11) and (16), respectively. When updating s_n, it will take on one of the existing cluster labels (i.e., $1, 2, \cdots, K_n$) more likely than taking on a new cluster label $(K_n + 1)$ since $n_k^{(-n)}B_k$ is likely to be larger than αC_k, thus forming clusters.

Conditional Distribution for W. The conditional distribution for $\boldsymbol{W}$ can be expressed as follows:

$$p(\boldsymbol{W}|\boldsymbol{Y}, \boldsymbol{S}, \boldsymbol{X}) = \prod_{k=1}^{\infty}\mathcal{N}(\boldsymbol{w}_k|\bar{\boldsymbol{w}}_k, \Sigma_k) \tag{22}$$

where $\bar{\boldsymbol{w}}_k = \Sigma_k\boldsymbol{b}_k, \boldsymbol{b}_k = \frac{1}{\sigma^2}\sum_{n=1}^{N}y_n\boldsymbol{x}_n$ and $\Sigma_k^{-1} = \Sigma_{0k}^{-1} + \frac{1}{\sigma^2}\sum_{n=1}^{N}\mathbf{1}(s_n = k)\boldsymbol{x}_n\boldsymbol{x}_n^T$.

Conditional Distribution for Z. The conditional distribution for each element z_{kd} is a Bernoulli distribution given the other elements $\boldsymbol{z}_k^{(-d)} = \{z_{k1}, \cdots, z_{k(d-1)}, z_{k(d+1)}, \cdots, z_{kD}\}$ and regression weights $\boldsymbol{w}_k$.

$$p(z_{kd} = 1|\boldsymbol{w}_k, \boldsymbol{z}_k^{(-d)}) = \frac{a_k}{a_k + b_k} \tag{23}$$

where $b_k = p(\boldsymbol{w}_k|\boldsymbol{z}_k^{(-d)}, z_{kd} = 0)(1 - p_{kd})$, $a_k = p(\boldsymbol{w}_k|\boldsymbol{z}_k^{(-d)}, z_{kd} = 1)p_{kd}$, and $p(\boldsymbol{w}_k|z_k)$ is given in Equation (7).

The Gibbs sampling procedure involves two steps:

Step 1: Initialize $\boldsymbol{S}$ with a random integer vector, $\boldsymbol{W}$ with a random matrix whose elements are drawn from a normal distribution, and $\boldsymbol{Z}$ with an all-one vector.

Step 2: Alternatively sample $\boldsymbol{S}$, $\boldsymbol{W}$ and $\boldsymbol{Z}$ from their conditional distributions (Equations (22),(23) and (19)) until the Markov Chain has reached its stationary distribution. We use the tail sample values as our estimation.

5 EXPERIMENTS

In this section we perform experiments on both synthetic and real world data sets. To maintain causality we always keep the chronological ordering of the data points for the DP ordering, which means that the states of the data points in "earlier" time will be updated first while the states of the data points in "later" time will be updated after its predecessors.

5.1 Synthetic Data

We first test on synthetic data to verify that our model can learn the known underlying spatio-temporal clusters and features. The synthetic data are constructed as follows: we created six different predictor tensors $\mathcal{X}^1, \mathcal{X}^2, \ldots, \mathcal{X}^6$ analogous to GCM outputs, and one target tensor $\mathcal{Y}$ analogous to observations. The dimensions of each tensor are $24 \times 4 \times 36$, where the three dimensions can be regarded as longitude, latitude and month, respectively. We assume the grid resolution is 1 degree by 1 degree ($\sim$ 100km by 100km geometrically). Each element of $\mathcal{X}^m$ $(m = 1, \cdots, 6)$ is independently and identically generated from $\mathcal{N}(0.0, 5.0)$.

We divide the tensors into three clusters of equal size along the longitude dimension. Within each cluster, the element of $\mathcal{Y}$ is a sparse linear combination of the corresponding elements of $\mathcal{X}^m$. We use sparse weights such that only a few features are relevant in each cluster. Specifically, $w_1 = [1, 0, 0, 0, 0, 0]^T$, $w_2 = [0, 0.5, 0.5, 0, 0, 0]^T$ and $w_3 = [0, 0, 0, 0.3, 0.3, 0.4]^T$. Then we merge the predictor tensors together and rearrange the two tensors into 3456×6 predictor matrix $\boldsymbol{X}$ and 3456×1 target matrix $\boldsymbol{Y}$. We also keep the location and neighbourhood information of each data point to be used in MRF. We use the first 30 months as training data and the last 6 months as the test data, that is, 2880 training data and 576 test data. Two metrics are used to evaluate the performance of the model on the synthetic data set. One is the Root-Mean-Square Error (RMSE)

Table 1: Results on synthetic data. K is the number of clusters. See text for the definition of NMI. Bold numbers represent best performance.

methods	K	RMSE	NMI
kmeans+OLS	1	0.826	0.000
	2	1.089	0.011
	3	0.969	0.011
	4	1.048	0.021
	5	1.056	0.023
kmeans+Lasso	1	0.826	0.000
	2	1.096	0.009
	3	0.984	0.011
	4	1.062	0.021
	5	1.071	0.024
MSSL	-	0.066	-
our method	3	**0.063**	**0.9841**

of the test data, and the other is the Normalized Mutual Information (NMI) [36]. Suppose we have N_{test} number of test data points, the RMSE is calculated as $RMSE = \sqrt{\frac{1}{N_{test}} \sum_{n=1}^{N_{test}} (y_n - \hat{y}_n)^2}$ where y_n and $\hat{y}_n$ are the ground-truth value and the predicted value of the n-th test data point, respectively. The NMI is calculated as follows: let S_{true} represent the ground truth cluster labels and S be the cluster labels determined by our model for all the data points, then $NMI = \frac{H_e(S_{true}) - H_e(S_{true}|S)}{\sqrt{H_e(S_{true}) H_e(S)}}$, where $H_e(S_{true})$ and $H_e(S)$ are the entropy of S_{true} and S, respectively. The value of NMI is between 0 and 1 inclusively, and a larger value means a better consistancy between the two clustering assignments.

We apply our model on the synthetic data with parameter $\sigma = 0.05$ and compare it with k-means methods and the MSSL method [12]. The results are shown in Table 1 and Figure 2. k-means is not able to find out the true clusters and performs worse with high RMSE. This is expected because k-means cluster the data only by using the dependent valuable (i.e., Y) while the ground truth clusters are based on the relationship between the independent and dependent valuables (i.e., X and Y), therefore k-means cannot capture the clustering patterns. While both MSSL and our model have low RMSE, our model can learn the true clusters. Within each cluster we recover the relevant features correctly. The results confirm that our method is able to discover the correct clusters as reflected by high NMI values.

5.2 Real World Data

We test whether our model can predict precipitation, which is very important yet very challenging in climate analysis. We downloaded 18 GCMs data sets of CMIP5 from the National Aeronautics and Space Administration (NASA) database [9] [1]. Each GCM data set contains a three dimensional tensor (i.e., latitude, longitude and time) $\mathcal{X}^m (m = 1, \cdots, 18)$ of monthly mean precipitation in space and time simulated by different climate research institutes. The

[1] https://nex.nasa.gov/nex/resources/348/, last accessed May 2019

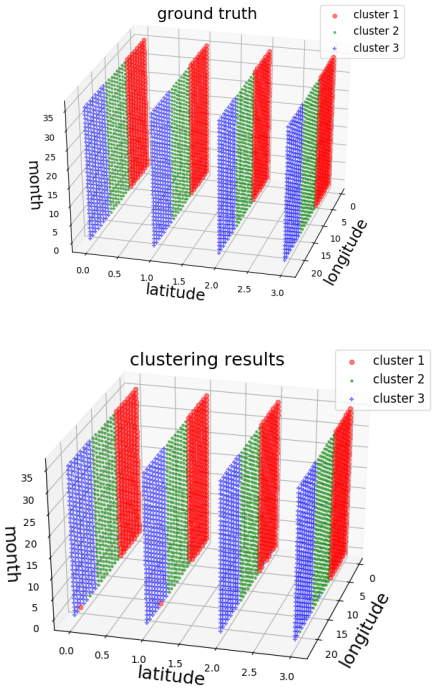


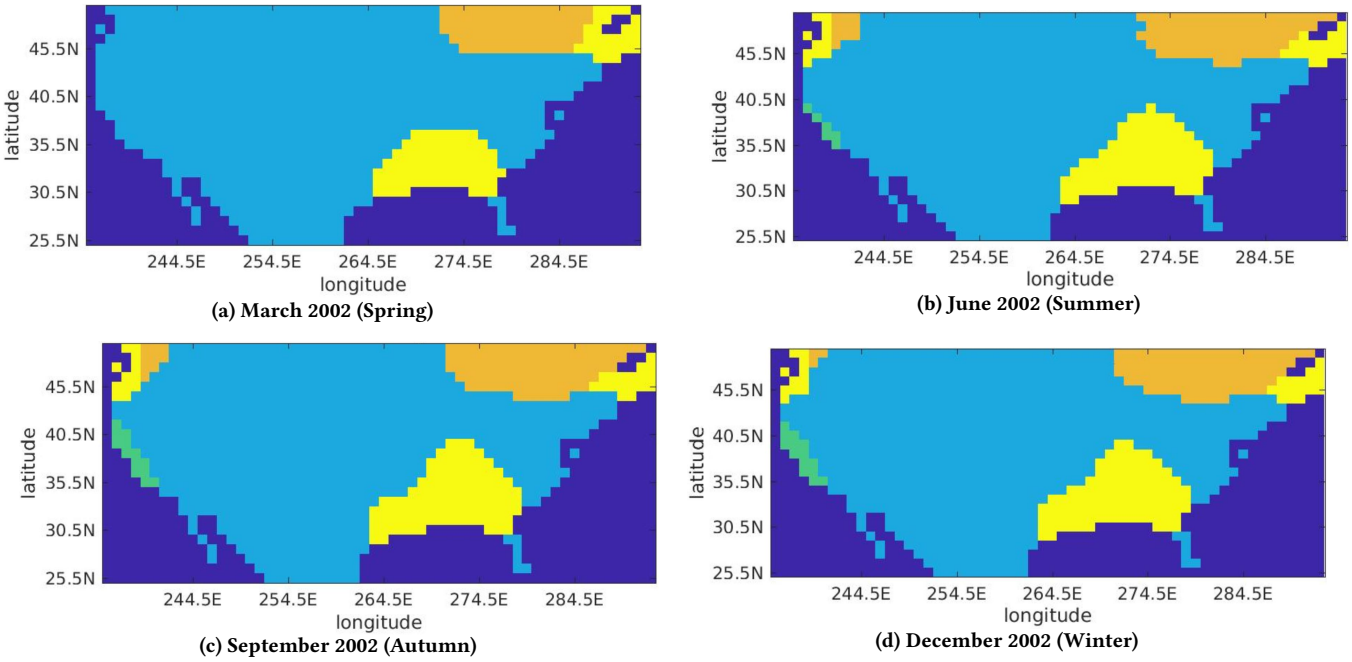
Figure 2: Clustering results of synthetic data. Top: ground truth. Bottom: results of our model. Better view in color.

Table 2: GCM names and their corresponding indices in the feature vector

index	GCM name	index	GCM name
1	giss-e2-h-cc	10	cnrm-cm5
2	access1-0	11	cesm1-cam5
3	canesm2	12	inmcm4
4	hadcm3	13	bcc-csm1-1-m
5	fgoals-g2	14	mri-cgcm3
6	csiro-mk3-6-0	15	miroc5
7	noresm1-m	16	bnu-esm
8	mpi-esm-lr	17	ccsm4
9	fio-esm	18	ipsl-cm5a-lr

resolution for the grid is 1 degree by 1 degree (~100km by 100km geometrically). The indices and the GCM names are shown in Table 2. For the target variable, we use the reanalysis data set by the University of Delaware from the National Oceanic and Atmospheric Administration (NOAA) [38]. The reanalysis data combine sparse on-site observation with other sources such as remote sensing and satellite images to produce high resolution data which are often used to represent the true observations. The raw data contains a three dimensional tensor $\mathcal{Y}$ of monthly mean precipitation in space and time.

We focus our model on the monthly mean precipitation over the continental United States (latitude from 25.5N to 49.5N by 1 degree,



(a) March 2002 (Spring)

(b) June 2002 (Summer)

(c) September 2002 (Autumn)

(d) December 2002 (Winter)

Figure 3: Clustering results for different time and seasons. The four clusters colored in cyan, green, yellow and orange are denoted as cluster 1, cluster 2, cluster 3 and cluster 4, respectively. The areas in blue color are areas that we don't have observational data. Please view in color version.

longitude from 235.5E to 292.5E by 1 degree) for 72 consecutive months, from January 1997 to December 2002. The original data are tensors with dimensions longitude, latitude and month. We first preprocess the data to align the grids between the predictor and the target, then merge and rearrange the tensors to the design matrix and the target vector in the same way as for the synthetic data. As a result, we get a predictor design matrix $\boldsymbol{X}$ of $N \times D$ and a target vector $\boldsymbol{Y}$ of $N \times 1$. Specifically, $N = 72864$ and $D = 19$.

We select our parameters $\{\beta, \gamma, \rho, \sigma, \alpha\}$ by using 5-fold cross- validation. The first 60 months are divided into 5 folds sequentially (1st to 36th month, 7th to 42nd month, $\cdots$, 25th to 60th month). Each fold contains 36 months including 30 months for training and 6 months for validating. The training set has enough number of months to cover the potential effect of seasonality. The optimal parameters are selected based on the average performance of the 5 folds, *i.e.*, the parameters that corresponds to the minimum average RMSE on validation data of the 5 folds are regarded as the optimal parameters. The optimal parameters chosen are $\beta = 0.01$, $\gamma = 0.01$, $\rho = 8.0$, $\sigma = 0.05$, $\alpha = 0.01$. After that, we re-train the model on all the first 60 months data and test it on the remaining 12 months data using the optimal parameters.

For the real world data, we do not have the ground truth cluster labels. We compare our model with three other methods. Two of them are based on k-means, which is a standard method used in climate science [15, 16, 35]. The first one is the baseline method in which we first apply k-means to $\boldsymbol{Y}$ to get the clusters and then fit each cluster with Ordinary Least Squares (OLS) regression. The second one is the same as the baseline except that we use Lasso regression instead of OLS regression in order to also select the features (GCMs) in each cluster. The third method is the Multi-task Sparse Structure Learning (MSSL) method [12] which regards each location as a task and estimate the sparse task parameters matrix and task relationship structure using L_1 regularization. All methods use the same training and testing procedures and select optimal parameters by cross-validation. To keep the results manageable, we choose the hyperparameters that result in a small number of clusters, and only the most discernible regions will be clustered into different clusters. In the experiment, there are four clusters discovered by our model.

The results of comparing our model with other methods are shown in Table 3. K is the number of clusters which is manually set in k-means but learned automatically in our model. R^2 is the coefficient of determination, which is the proportion of variance in the target that can be explained by the predictor. It is a measure of how well the data are fitted to the regression line and the larger the value the better. The R^2 gain is the gain compared to the value of R^2 of the *k-means+OLS* with K=1. The MSSL regards each location as a task and does not do clustering. From the table we can see that our method has the smallest RMSE and largest R^2 among the three methods. Furthermore, k-means is not able to discover meaningful clusters and its performance degrades as K increases. The reason is the same with that of synthetic data. Here k-means cluster the spatio-temporal data only by using observed precipitation (i.e.,$\boldsymbol{Y}$), which is not enough for capturing the relationship between the GCMs simulated precipitation and the observed precipitation. The reason why MSSL performs even worse than the k-means+OLS/Lasso on the real data maybe that MSSL assumes that all data points belonging to the same location are in the same task, which maybe incorrect. Because the climate system is a time varying system, the relationships between GCMs and observation

Table 3: Comparision between our method and others

methods	K	RMSE	R^2	R^2 gain (%)
kmeans+OLS	1	0.8149	0.3359	baseline
	2	0.8461	0.2841	-15.43
	3	0.8972	0.1950	-41.94
	4	0.8828	0.2207	-34.31
	5	0.9316	0.1321	-60.67
kmeans+Lasso	1	0.8116	0.3413	1.60
	2	0.8456	0.2850	-15.17
	3	0.8983	0.1931	-42.53
	4	0.8842	0.2182	-35.05
	5	0.9329	0.1297	-61.39
MSSL	–	0.8716	0.2403	−28.46
our method	4	**0.7811**	**0.3900**	**16.06**

will change over time. And that is why we need to introduce a spatio-temporal clustering method instead of just spatial clustering method. Our method cluster the spatio-temporal data based on the relationships between GCMs and observed precipitation and can discover better clustering patterns and achieve better prediction performance. Figure 3 shows the clustering results for four different months over the continental United States. The clusters are spatially smooth with nearby locations being assigned to the same cluster. The results are partially consistent with the Köppen climate classification system [17, 18], which is one of the most widely used climate classification system. For example, in Figure 3(a), the cluster in the orange color is around the Great Lakes area, where the terrain and evaporation characteristics are different from other areas. These Great Lakes cause a phenomenon called *Lake Effect* which affects the precipitation [6]. This area is regarded as type *Dfb* in the Köppen system [17]. In Figure 3(b)-(d) the cluster in the green color is the *Central Valley* in California that has a Mediterranean climate, which is dry during the summer and damp in winter. According to Köppen [17] and Köppen et al. [18], this area has a climate type *Csa* which is unique in the US. For the cluster in yellow color, a large portion of it lies in the downstream area of the Mississippi River where it is very rich in both waters and vegetation, which distinguishes itself from other areas in the US.

We also discover some teleconnections among far away regions. For example, in Figure 3(c)-(d), the northeastern corner and the northwestern corner have the same cluster (yellow), which is interesting considering that they are both coastal areas in the north. Further research is needed to study the teleconnections.

Comparing Figure 3(a)-(d), we can see that the cluster pattern has some continuity in the temporal aspect, but also evolves as time goes by. For example, cluster in the green color appeared since June 2002, while clusters in the yellow and orange emerged and then shrunk again in the northwestern corner of the US.

Figure 4 shows the regression weights for the clusters. Each row represents a weight vector for one cluster. The horizontal axis is the index number of the weight vector (the first index is for the bias term), the vertical axis is the value for each element of the weight vector, an absence means the element is 0 and thus the corresponding GCM has no contribution to the observation in the cluster. From the figure, we can see that the weights exhibit some sparsity in each cluster, meaning that for a specific cluster, some

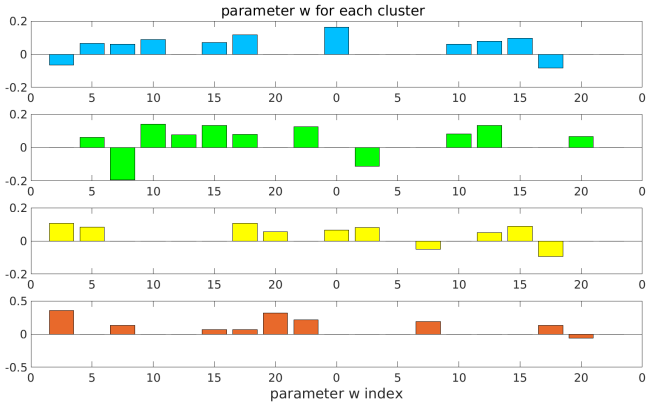


Figure 4: Regression weights of clusters for real world data – precipitation prediction. There are 4 rows, each represents a weight vector for a cluster. From top to bottom are the weights for cluster 1, cluster 2, cluster 3 and cluster 4.

GCMs are not important while others are more useful in predicting the observational precipitation, as expected.

6 CONCLUSION

We introduce a novel method to cluster spatio-temporal data and perform sparse regression within each cluster. This method incorporates spatio-temporal constraints by Markov Random Field, automatically learns the number of clusters in a nonparametric way through Dirichlet Process, and selects features using spike-and-slab prior. We apply our method on both synthetic and real-world GCM precipitation data and show that we outperform competing methods in terms of prediction performance. We further learn interesting clusters and which GCMs are important within each spatio-temporal cluster. The reason behind this may be an interesting topic for climatologists. The results on precipitation data is partially consistent with the widely used Köppen climate classification system. The learned clusters agree with the terrain and geological characteristics such as forests, lakes and valleys. Furthermore, we also discover teleconnections and cluster evolution through time. These results provide new insights to the data.

There are several possible directions for future work. One can extend the model to be fully Bayesian, adding priors to the hyperparameters. In this work we focus on monthly mean precipitation of the continental United State. In order to get a more comprehensive understanding, this can be extended to a larger scale with more computational resources such as high performance computers. Also, with the success of deep learning in many fields, the climate science community has embraced deep learning approaches in recent years; for example Reichstein et al. [32] and Rasp et al. [31]. In our ongoing research, we are exploring ways to reformulate the problem using deep learning approaches.

ACKNOWLEDGMENTS

This work was supported by National Science Foundation CyberSEES under grant number: NSF CCF-1442728. We thank Thomas Vandal, Yi Li and Yale Chang for their help.

REFERENCES

[1] David J Aldous. 1985. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII—1983*. Springer, 1–198.

[2] Mohammad Taha Bahadori, Qi Rose Yu, and Yan Liu. 2014. Fast multivariate spatio-temporal analysis via low rank tensor learning. In *Advances in neural information processing systems*. 3491–3499.

[3] Sugato Basu, Mikhail Bilenko, Arindam Banerjee, and Raymond J Mooney. 2006. Probabilistic semi-supervised clustering with constraints. *Semi-supervised learning* (2006), 71–98.

[4] Luc Bauwens, Jean-François Carpantier, and Arnaud Dufays. 2017. Autoregressive moving average infinite hidden Markov-switching models. *Journal of Business & Economic Statistics* 35, 2 (2017), 162–182.

[5] Marco Bazzi, Francisco Blasques, Siem Jan Koopman, and Andre Lucas. 2017. Time-varying transition probabilities for Markov regime switching models. *Journal of Time Series Analysis* 38, 3 (2017), 458–478.

[6] Lee Botts and Bruce Krushelnicki. 1987. *The Great Lakes. An Environmental Atlas and Resource Book*. ERIC.

[7] George Casella and Edward I George. 1992. Explaining the Gibbs sampler. *The American Statistician* 46, 3 (1992), 167–174.

[8] FJ Doblas-Reyes, R Hagedorn, and TN Palmer. 2006. Developments in dynamical seasonal forecasting relevant to agricultural management. *Climate Research* 33, 1 (2006), 19–26.

[9] Karl E. Taylor, Stouffer Ronald, and Gerald Meehl. 2011. An overview of CMIP5 and the Experiment Design. *Bulletin of the American Meteorological Society* 93 (11 2011), 485–498. https://doi.org/10.1175/BAMS-D-11-00094.1

[10] Emily Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. 2009. Nonparametric Bayesian learning of switching linear dynamical systems. In *Advances in Neural Information Processing Systems*. 457–464.

[11] Edward I George and Robert E McCulloch. 1993. Variable selection via Gibbs sampling. *J. Amer. Statist. Assoc.* 88, 423 (1993), 881–889.

[12] André R Gonçalves, Puja Das, Soumyadeep Chatterjee, Vidyashankar Sivakumar, Fernando J Von Zuben, and Arindam Banerjee. 2014. Multi-task sparse structure learning. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 451–460.

[13] Arthur M Greene, Lisa Goddard, and Upmanu Lall. 2006. Probabilistic multimodel regional temperature change projections. *Journal of Climate* 19, 17 (2006), 4326–4343.

[14] Jonathan Gregory, Ronald J Stouffer, Mario Molina, Amnat Chidthaisong, Susan Solomon, Graciela Raga, Pierre Friedlingstein, Nathaniel L Bindoff, Hervé Le Treut, Matilde Rusticucci, et al. 2007. Climate change 2007: the physical science basis. (2007).

[15] William W Hargrove and Forrest M Hoffman. 2004. Potential of multivariate quantitative methods for delineation and visualization of ecoregions. *Environmental management* 34, 1 (2004), S39–S60.

[16] Forrest M Hoffman, William W Hargrove Jr, David J Erickson III, and Robert J Oglesby. 2005. Using clustered climate regimes to analyze and compare predictions from fully coupled general circulation models. *Earth Interactions* 9, 10 (2005), 1–27.

[17] Wladimir Köppen. 1884. Die Wärmezonen der Erde, nach der Dauer der heissen, gemässigten und kalten Zeit und nach der Wirkung der Wärme auf die organische Welt betrachtet. *Meteorologische Zeitschrift* 1, 21 (1884), 5–226.

[18] Wladimir Köppen, Esther Volken, and Stefan Brönnimann. 2011. The thermal zones of the earth according to the duration of hot, moderate and cold periods and to the impact of heat on the organic world (Translated from: Die Wärmezonen der Erde, nach der Dauer der heissen, gemässigten und kalten Zeit und nach der Wirkung der Wärme auf die organische Welt betrachtet, Meteorol Z 1884, 1, 215-226). *Meteorologische Zeitschrift* 20, 3 (2011), 351–360.

[19] Devashish Kumar and Auroop R Ganguly. 2018. Intercomparison of model response and internal variability across climate model ensembles. *Climate dynamics* 51, 1-2 (2018), 207–219.

[20] Devashish Kumar, Evan Kodra, and Auroop R Ganguly. 2014. Regional and seasonal intercomparison of CMIP3 and CMIP5 climate model ensembles for temperature and precipitation. *Climate dynamics* 43, 9-10 (2014), 2491–2518.

[21] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).

[22] Scott McQuade and Claire Monteleoni. 2012. Global climate model tracking using geospatial neighborhoods. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[23] Toby J. Mitchell and John J. Beauchamp. 1988. Bayesian variable selection in linear regression. *J. Amer. Statist. Assoc.* 83, 404 (1988), 1023–1032.

[24] Paul A O'Gorman and John G Dwyer. 2018. Using Machine Learning to Parameterize Moist Convection: Potential for Modeling of Climate, Climate Change, and Extreme Events. *Journal of Advances in Modeling Earth Systems* 10, 10 (2018), 2548–2563.

[25] Paul A O'Gorman and Tapio Schneider. 2009. The physical basis for increases in precipitation extremes in simulations of 21st-century climate change. *Proceedings of the National Academy of Sciences* 106, 35 (2009), 14773–14777.

[26] Peter Orbanz and Joachim M Buhmann. 2008. Nonparametric Bayesian image segmentation. *International Journal of Computer Vision* 77, 1-3 (2008), 25–45.

[27] TN Palmer, FJ Doblas-Reyes, A Weisheimer, and MJ Rodwell. 2008. Toward seamless prediction: Calibration of climate change projections using seasonal forecasts. *Bulletin of the American Meteorological Society* 89, 4 (2008), 459–470.

[28] Tim N Palmer, A Alessandri, U Andersen, P Cantelaube, M Davey, Pascale Delécluse, Michel Déqué, E Diez, Francisco Javier Doblas-Reyes, Henrik Feddersen, et al. 2004. Development of a European multimodel ensemble system for seasonal-to-interannual prediction (DEMETER). *Bulletin of the American Meteorological Society* 85, 6 (2004), 853–872.

[29] Jorge Prendes, Marie Chabert, Frédéric Pascal, Alain Giros, and Jean-Yves Tourneret. 2015. Change detection for optical and radar images using a Bayesian nonparametric model coupled with a Markov random field. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1513–1517.

[30] Simon JD Prince. 2012. *Computer vision: models, learning, and inference*. Cambridge University Press.

[31] Stephan Rasp, Michael S Pritchard, and Pierre Gentine. 2018. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences* 115, 39 (2018), 9684–9689.

[32] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, et al. 2019. Deep learning and process understanding for data-driven Earth system science. *Nature* 566, 7743 (2019), 195.

[33] James Ross and Jennifer Dy. 2013. Nonparametric mixture of Gaussian processes with constraints. In *International Conference on Machine Learning*. 1346–1354.

[34] Richard L Smith, Claudia Tebaldi, Doug Nychka, and Linda O Mearns. 2009. Bayesian modeling of uncertainty in ensembles of climate models. *J. Amer. Statist. Assoc.* 104, 485 (2009), 97–116.

[35] Karsten Steinhaeuser, Nitesh V Chawla, and Auroop R Ganguly. 2011. Complex networks as a unified framework for descriptive analysis and predictive modeling in climate science. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 4, 5 (2011), 497–511.

[36] Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research* 3, Dec (2002), 583–617.

[37] Yee W Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2005. Sharing clusters among related groups: Hierarchical Dirichlet processes. In *Advances in neural information processing systems*. 1385–1392.

[38] Cort J Willmott. 2000. Terrestrial air temperature and precipitation: Monthly and annual time series (1950-1996). *WWW url: http://climate. geog. udel. edu/~ climate/html_pages/README. ghcn_ts. html* (2000).