



Flood Inundation Mapping with Limited Observations Based on Physics-Aware Topography Constraint

Arpan Man Sainju¹, Wenchong He², Zhe Jiang^{2*}, Da Yan³ and Haiquan Chen⁴

¹Department of Computer Science, Middle Tennessee State University, Murfreesboro, TN, United States,

²Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, United States,

³Department of Computer Science, University of Alabama at Birmingham, Birmingham, AL, United States, ⁴California State University, Sacramento, CA, United States

OPEN ACCESS

Edited by:

Xun Zhou,
The University of Iowa, United States

Reviewed by:

Keli Xiao,
Stony Brook University, United States
Liang Zhao,
George Mason University,
United States

*Correspondence:

Zhe Jiang
zhe.jiang@ufl.edu

Specialty section:

This article was submitted to
Data Mining and Management,
a section of the journal
Frontiers in Big Data

Received: 11 May 2021

Accepted: 14 June 2021

Published: 26 July 2021

Citation:

Sainju AM, He W, Jiang Z, Yan D and
Chen H (2021) Flood Inundation
Mapping with Limited Observations
Based on Physics-Aware
Topography Constraint.
Front. Big Data 4:707951.
doi: 10.3389/fdata.2021.707951

Spatial classification with limited observations is important in geographical applications where only a subset of sensors are deployed at certain spots or partial responses are collected in field surveys. For example, in observation-based flood inundation mapping, there is a need to map the full flood extent on geographic terrains based on earth imagery that partially covers a region. Existing research mostly focuses on addressing incomplete or missing data through data cleaning and imputation or modeling missing values as hidden variables in the EM algorithm. These methods, however, assume that missing feature observations are rare and thus are ineffective in problems whereby the vast majority of feature observations are missing. To address this issue, we recently proposed a new approach that incorporates physics-aware structural constraint into the model representation. We design efficient learning and inference algorithms. This paper extends our recent approach by allowing feature values of samples in each class to follow a multi-modal distribution. Evaluations on real-world flood mapping applications show that our approach significantly outperforms baseline methods in classification accuracy, and the multi-modal extension is more robust than our early single-modal version. Computational experiments show that the proposed solution is computationally efficient on large datasets.

Keywords: limited observation, physical constraint, spatial classification, machine learning, flood mapping

1 INTRODUCTION

Given a spatial raster framework with explanatory feature layers, a spatial contextual layer (e.g., an elevation map), and a set of training samples with class labels outside the framework, the spatial classification problem aims to learn a model that can predict the class layer Du et al. (2019); Jiang (2019, 2020); Shekhar et al. (2015); Wang P. et al. (2018); Zhang et al. (2019); Karpadne et al. (2016); Jiang (2019). We particularly focus on spatial classification with limited feature observations, i.e., only limited pixel locations in the raster framework have explanatory feature data available. For example, observation-based flood inundation extent mapping aims to classify all pixels in the raster framework into flood and dry classes, even in the case whereby only a part of the pixels have corresponding spectral features. In this example, the elevation values are available for all pixels in the

framework, but only limited pixel locations have spectral data (e.g., a drone or aerial plane could not cover the entire region due to limited time during a flood disaster).

The problem is important in many applications such as flood extent mapping. Flood extent mapping is crucial for disaster management, national water forecasting, and energy and food security Jiang and Shekhar (2017); Jiang et al. (2019); Xie et al. (2018). For example, during hurricane floods, first responders needed to know where the floodwater is in order to plan rescue efforts. In national water forecasting, accurate flood extent maps can be used to calibrate and validate the NOAA National Water Model National Oceanic and Atmospheric Administration (2018b). In current practice, flood extent maps are mostly produced by forecasting models, whose accuracy is often unsatisfactory in a high spatial resolution Cline (2009); Merwade et al. (2008). Other ways to generate flood maps involve sending a field crew on the ground (e.g., recording high watermarks), but the process is both expensive and time-consuming. A promising alternative is to utilize Earth observation data from remote sensors. However, sensor observations often have limited spatial coverage due to only a subset of sensors being deployed at certain spots, making it a problem of spatial classification with limited feature observations. For example, during a flood disaster, a drone can only collect spectral images in limited areas due to time limit. Though we use flood mapping as a motivation example, the problem is general for other applications such as water quality monitoring Yang and Jin (2010) in river networks.

The problem poses several unique challenges that are not well addressed by traditional classification techniques. First, there are limited feature observations on samples in the raster framework due to only a subset of sensors being deployed in certain regions. In other words, only a subset of samples have complete explanatory feature values, making it hard to predict classes for all samples. Second, among the sample pixels with complete explanatory feature values, their feature values may contain rich noise and obstacles (e.g., clouds and shadows). Third, the explanatory features of image pixels can be subject to class confusion due to heterogeneity. For instance, pixels of tree canopies above flood water have the same spectral features as trees in dry areas, yet the classes of these pixels are different. Finally, the number of pixel locations can be very large for high-resolution data (e.g., hundreds of millions of pixels). This requires scalable algorithms.

Over the years, various techniques have been developed to address missing feature observations (or incomplete data) in classification García-Laencina et al. (2010). Existing methods can be categorized into data cleaning or imputation, extending classification models to allow for missing values, and modeling missing features as hidden variables in the EM algorithm. Data cleaning will remove samples that miss critical feature values. Data imputation focuses on filling in missing values either by statistical methods Little and Rubin (2019) (e.g., mean feature values from observed samples) or by prediction models (e.g., regression) based on observed samples Batista and Monard (2002); Bengio and Gingras (1996); Rubin (2004); Schafer (1997); Yoon and Lee (1999). There are also approaches that

handle missing values by the multi-task strategy (i.e., partition different patterns of missing values into different tasks) as in García-Laencina et al. (2013); Zhou et al. (2011); Wang et al. (2018a). Another approach focuses on classification models and algorithms that allow for missing feature values in learning and prediction without data imputation. For example, a decision tree model allows for samples with missing features in learning and classification Quinlan (1989, 2014); Webb (1998). During learning, for a missing feature value in a sample, a probability weight is assigned to each potential feature value based on its frequency in observed samples. During classification, a decision tree can explore all possible tree traversal paths for samples with missing features and select the final class prediction with the highest probability. Similarly, there are some other models that have been extended to allow for missing feature values, such as neural network ensembles Jiang et al. (2005), and support vector machine Chechik et al. (2007); Pelckmans et al. (2005); Smola et al. (2005). The last category is to model missing feature values as hidden variables and use the EM (Expectation-Maximization) algorithm for effective learning and inference Ghahramani Z. and Jordan M. I. (1994, 1994a); McLachlan and Krishnan (2007); Williams et al. (2007). Specifically, the joint distribution of all samples' features (both observed and missing features) can be represented by a mixture model with fixed but yet unknown parameters. In the EM algorithm, we can use initialized parameters and observed features to estimate the posterior distribution of hidden variables (missing features), and then further update the parameters for the next iteration. However, all these existing methods assume that incomplete feature observations are rare and thus cannot be effectively applied to our problem where the vast majority of samples have missing features (i.e., limited feature observations).

To fill this gap, we recently proposed a new approach that incorporates physics-aware structural constraints into model representation Sainju et al. (2020b). Our approach assumes that a spatial contextual feature (elevation map) is fully observed for every sample location, and establishes the topography constraints (i.e., water flow directions across elevation contours) from the elevation map He and Jiang (2020); Jiang (2020). The advantage of such physical constraints is that it provides a global dependency structure of class labels (i.e., flood or dry) across locations beyond a spatial neighborhood. We design efficient algorithms for model parameter learning and class inference and conduct experimental evaluations to validate the effectiveness and efficiency of the proposed approach against existing works. Motivated by the observations that sample features can be heterogeneous with multiple modalities, this paper extends our recent approach by allowing for multi-modal feature distribution in each class. We also propose the parameter learning algorithms for the extended model. In summary, the paper makes the following contributions:

- We extend the model by allowing feature values of samples in each class to follow a multi-modal distribution.
- We evaluated the proposed model on two real-world hydrological datasets. Results show that the new multi-

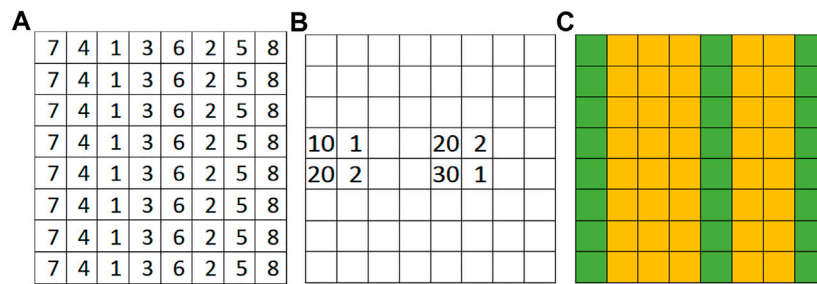


FIGURE 1 | An illustration problem example. **(A)** Spatial potential field (elevation) **(B)** Partially observed non-spatial feature values **(C)** Ground truth classes (green for dry, orange for flood).

modal solution is more robust than the previous single-modal version especially when feature distribution in training samples is multi-modal.

- Computation experiments show that the proposed solution is scalable to a large data volume.

2 PROBLEM STATEMENT

2.1 Preliminaries

Here we define several basic concepts that are used in the problem formulation.

A *spatial raster framework* is a tessellation of a 2D plane into a regular grid of N cells. The framework can contain m explanatory feature layers (e.g., spectral bands in satellite imagery), a potential field layer (e.g., elevation), and a class layer (e.g., *flood* or *dry*).

Each pixel in a raster framework is a *spatial data sample*, denoted by $\mathbf{s}_n = (\mathbf{x}_n, \phi_n, y_n)$, where $n \in \mathbb{N}$, $1 \leq n \leq N$, $\mathbf{x}_n \in \mathbb{R}^{m \times 1}$ is a vector of m non-spatial explanatory feature values with each element corresponding to a feature layer, $\phi_n \in \mathbb{R}$ is a pixel's potential field value, and $y_n \in \{0, 1\}$ is a binary class.

A raster framework with all samples is denoted by $\mathcal{F} = \{\mathbf{s}_n | n \in \mathbb{N}, 1 \leq n \leq N\}$, non-spatial explanatory feature matrix of all samples are denoted by $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$, the potential field vector is denoted by $\Phi = [\phi_1, \dots, \phi_N]^T$, and the class vector is denoted by $\mathbf{Y} = [y_1, \dots, y_N]^T$.

In a raster framework, it may happen that only a limited number of samples have non-spatial explanatory features being observed. We define \mathcal{O} as the set of indices for these fully observed samples. Samples with fully observed (complete) explanatory features are denoted by $\{\mathbf{x}_n | n \in \mathcal{O}\}$. Their feature matrix is denoted by \mathbf{X}_o .

2.2 Problem Definition

Given a raster framework with the explanatory features of a limited number of samples \mathbf{X}_o , the potential field layer of all samples in the framework $\Phi = [\phi_1, \dots, \phi_N]^T$, and a set of training pixels with class labels, the problem aims to learn a classifier f to predict the class layer $\mathbf{Y} = f(\mathbf{X}_o, \Phi)$. For example, in flood extent mapping, the explanatory feature layers are spectral bands of earth imagery; the spatial contextual layer is geographic terrains based on elevation, and the target class layer are classes of

flood or dry. We assume that the elevation values are available for all pixels in the framework (assuming that these values do not frequently change over time) but only limited pixel locations have spectral observations (e.g., a drone or aerial plane could not cover the entire region). **Figure 1** shows a toy example of a raster framework that consists of 64 samples with a one-dimensional explanatory feature and a potential field layer. There are only eight samples with observed explanatory features (four non-empty cells in **Figure 1B**). The goal is to learn a model that can predict the class layer in **Figure 1C**.

3 APPROACH

In this section, we introduce our proposed approach. We start with physics-aware structural constraints and then introduce our probabilistic model and its learning and inference algorithms. We will introduce our approach in the context of the flood mapping application, but the proposed method can be potentially generalized to other applications such as material science Wales et al. (1998), Wales (2003) and biochemistry Edelsbrunner and Koehl (2005); Günther et al. (2014).

3.1 Physics-Aware Structural Constraint

The main idea of our proposed approach is to establish a spatial dependency structure of sample class labels based on the physical constraint from the spatial potential field layers (e.g., water flow directions based on elevation). An illustration is provided in **Figure 2**. **Figure 2A** shows the elevation values of eight pixels in one dimensional space (e.g., pixels on a row in **Figure 1**). Due to gravity, water flows from high locations to nearby lower locations. If location 4 is flooded, locations 1 and 3 must also be flooded. Such a dependency structure can be established based on the topology of the potential field surface (e.g., elevation). **Figure 2B** shows a directed tree structure that captures the flow dependency structure. If any node is *flood*, then all sub-tree nodes must also be *flood* due to gravity. The structure is also called *split tree* in topology Carr et al. (2003); Edelsbrunner and Harer (2010), where a node represents a vertex on a mesh surface (spatial potential field) and an edge represents the topological relationships between vertices. We can efficiently construct the tree structure from a potential field map following the topological

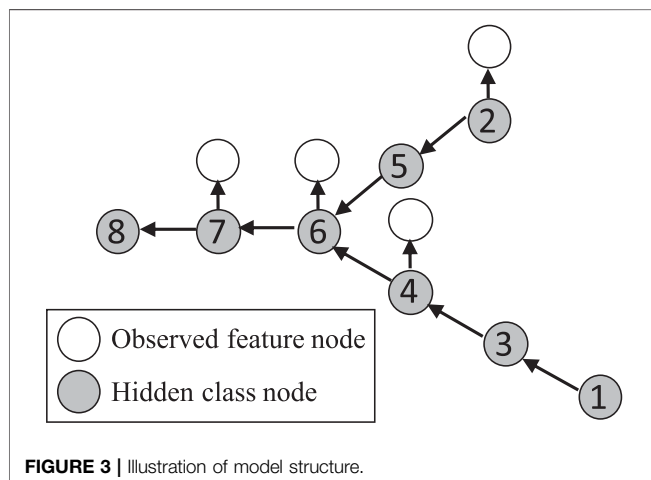
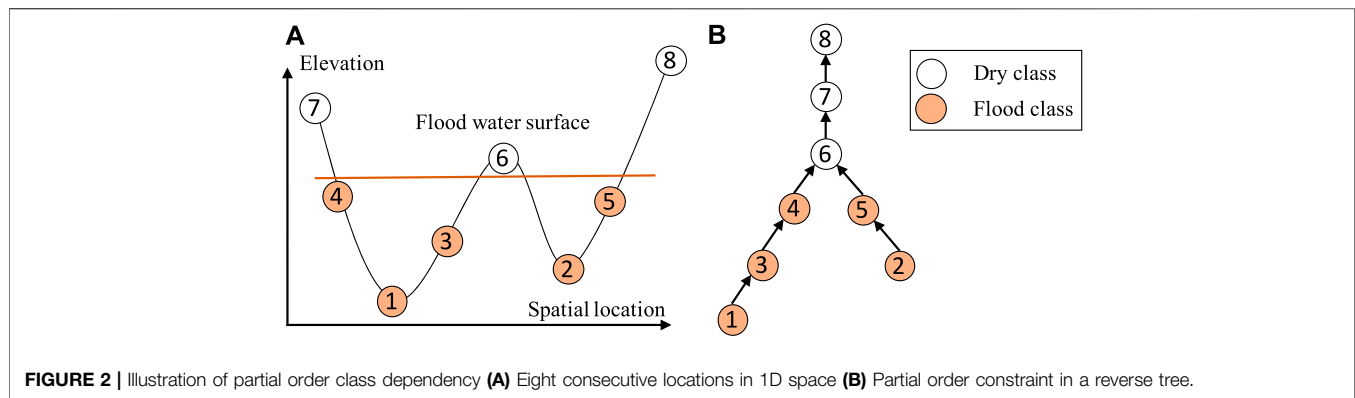


FIGURE 3 | Illustration of model structure.

order of pixels based on the union-find operator (its time complexity is $O(N\log N)$) Carr et al. (2003). We omit details due to space limit. It is worth noting that although our illustrative example in **Figure 2** is in one-dimensional space for simplicity, the structure is readily applicable to two-dimensional space Jiang and Sainju (2019). We can create a single tree structure for the entire elevation map in **Figure 1A**.

3.2 Model Probabilistic Formulation

Now we introduce our approach that integrates physics-aware structural constraint into the probabilistic model formulation to handle limited feature observations. The overall idea of the model structure is similar to Xie et al. (2018); Jiang et al. (2019); Jiang and Sainju (2019); Sainju et al. (2020a); Jiang and Sainju (2021). **Figure 3** illustrates the overall model structure. It consists of two layers: a hidden class layer with unknown sample classes (y_n) and an observation layer with limited sample feature vectors (\mathbf{x}_n). Each node is a spatial data sample (raster pixel). Edge directions represent a dependency structure based on physical constraints.

The joint distribution of all samples’ features and classes are in **Eq. 1**, where \mathcal{P}_n is the set of parent nodes in the dependency tree, and $y_{k \in \mathcal{P}_n} \equiv \{y_k | k \in \mathcal{P}_n\}$ is the set of parent classes. For a leaf node n , $\mathcal{P}_n = \emptyset$, and $P(y_n | y_{k \in \mathcal{P}_n}) = P(y_n)$.

TABLE 1 | Class transition probability and prior probability.

$\mathbf{P}(\mathbf{y}_n \mathbf{y}_{P_n})$	$\mathbf{y}_{P_n} = \mathbf{0}$	$\mathbf{y}_{P_n} = \mathbf{1}$
$y_n = 0$	1	$1 - \rho$
$y_n = 1$	0	ρ
	$\mathbf{P}(\mathbf{y}_n)$	
$y_n = 0$	$1 - \pi$	
$y_n = 1$	π	

$$P(X_o, \mathbf{Y}) = P(X_o|\mathbf{Y})P(\mathbf{Y}) = \prod_{n \in \mathcal{O}} P(\mathbf{x}_n|y_n) \prod_{n=1}^N P(y_n|y_{k \in \mathcal{P}_n}) \quad (1)$$

The sample feature distribution in each class is assumed i.i.d. Gaussian for simplicity, as shown in **Eq. 2**, where $\boldsymbol{\mu}_{y_n}$ and $\boldsymbol{\Sigma}_{y_n}$ are the mean and covariance matrix of feature vector \mathbf{x}_n for class y_n ($y_n = 0, 1$).

$$P(\mathbf{x}_n|y_n) \sim \mathcal{N}(\boldsymbol{\mu}_{y_n}, \boldsymbol{\Sigma}_{y_n}) \quad (2)$$

The class transitional probability follows a partial order. For instance, because of gravity, if any parent’s class is *dry*, the child’s class must be *dry*. On the other hand, if all parents’ classes are *flood*, then the current child’s class has a high probability of being *flood* too. Consider *flood* as class 1 and *dry* as class 0, then the previous assumption is actually conditioned on the product of parent classes $y_{\mathcal{P}_n} \equiv \prod_{k \in \mathcal{P}_n} y_k$, as expressed in **Table 1**, where π and ρ are parameters for class transitional probability and class prior probability.

3.3 Model Parameter Learning and Class Inference

Model parameters consist of the mean and covariance matrix of features in each class, the prior class probability, and the class transition probability. We denote the entire set of parameters as $\Theta = \{\rho, \pi, \mu_c, \Sigma_c | c = 0, 1\}$. Learning parameters poses two challenges: first, **Eq. 1** contains both unknown parameters and hidden class variables $\mathbf{Y} = [\gamma_1, \dots, \gamma_N]^T$ that are non-i.i.d.;

second, the number of samples (N) can be huge (e.g., millions of pixels).

We use an expectation-maximization (EM) algorithm together with message (belief) propagation. The main idea of the EM algorithm is to first initialize a parameter setting, and compute the posterior expectation of log-likelihood (Eq. 1) on hidden class variables (E-step). The posterior expectation is a function of unknown parameters. Thus, we can update parameters by maximizing the posterior expectation (M-step). The two steps can repeat iteratively until the parameter values converge. One remaining issue is the calculation of posterior expectation of log-likelihood on hidden class variables. This requires to compute the marginal posterior distribution of $P(y_n, y_{k \in \mathcal{P}_n} | \mathbf{O}, \Theta_0)$ and $P(y_n)$ for each node s_n . This is very challenging due to the high dimensionality of \mathbf{Y} . To address this challenge, we use message propagation. For message propagation, we use the sum and product algorithm Kschischang et al. (2001); Ronen et al. (1995). Propagation of message along tree nodes is similar to marginalizing out variables in the joint distribution in Eq. 1. Due to the space limit, we only show the major steps in the following discussion.

The message passing process is based on the sum-product algorithm, which involves tree traversal operations. After parameters learning, we can infer class variables by maximizing the joint probability. We use a dynamic programming algorithm called *max-sum* Rabiner (1989). It is similar to the above sum and product algorithm. The main difference is that we replace the *sum* operation with a *max* operation during message propagation.

3.4 Intuitions on How the Model Works

The main intuition behind how our model handles limited observations is that the model can capture physical constraints between sample classes. The spatial structural constraints are derived from the potential field layer that is fully observed on the entire raster framework, regardless of whether non-spatial features are available or not. The topological structure in a split tree is consistent with the physical law of water flow directions on a topographic surface based on gravity. In this sense, even though many samples in the raster framework do not have non-spatial explanatory features observed, we can still infer their classes based on information from the pixels in the upstream or downstream locations.

Another potential question is how our model can effectively learn parameters given very limited observations. This question can be answered from the perspective of how model learning works. The major task of model learning is to effectively update parameters of $P(\mathbf{x}_n | y_n)$ for observed pixels in the test region, so that we can infer the posterior class probabilities on these pixels and further infer hidden classes on other pixels. As long as the training samples could give the model a reasonable initial estimate of posterior class probabilities on the observed pixels (e.g., truly dry pixels having a higher probability of being dry), the update of parameters should be effective. This is because that parameter updates are largely weighted average of the sample mean and covariance matrices on fully observed pixels. The

corresponding weights are the posterior class probability of observed samples.

3.5 Extension to Multi-modal Feature Distribution

This subsection introduces an extension to our proposed model. In the conference version, the joint distribution of all sample features and classes follow a conditional independence assumption based on the tree structure derived from physical constraint (Eq. 1). Sample feature distribution in each class is assumed i.i.d. Gaussian (Eq. 2). In real-world datasets, the actual sample feature distribution in each class can be multi-modal, violating the earlier assumption on a single-modal Gaussian distribution. To account for this observation, we extend our model to allow for multi-modal sample feature distribution in each class. Specifically, we assume that sample features follow an i.i.d. mixture of Gaussian distribution.

For the extended model, the joint distribution of all observed samples' features and classes can be expressed the same way as Eq. 1. The joint probability can be decomposed into local factors, i.e., the conditional distribution of features in each class and class transitional probability. The assumption on class transitional probability for non-leaf nodes and prior class probability for leaf nodes remain the same as before (Table 1). What is different is the conditional probability of sample feature vector given its class. In the extended multi-modal model, sample feature in each class is assumed i.i.d. mixture Gaussian distribution, as shown in Eq. 3, where $\mu_{y_n}^i$ and $\Sigma_{y_n}^i$ are the mean and covariance matrix of the i -th Gaussian component of feature vector \mathbf{x}_n for class y_n ($y_n = 0, 1$), $\phi_{y_n}^i$ is the probability of a sample in class y_n belonging to the i -th Gaussian component $\mathcal{N}(\mu_{y_n}^i, \Sigma_{y_n}^i)$, K_{y_n} is the number of Gaussian components (modes) in the feature distribution of samples in class y_n .

$$P(\mathbf{x}_n | y_n) \sim \sum_{i=1}^{K_{y_n}} \phi_{y_n}^i \mathcal{N}(\mu_{y_n}^i, \Sigma_{y_n}^i), y_n = 0, 1 \quad (3)$$

Based on the extended probabilistic formulation, we can use the same EM algorithm with message propagation for parameter learning. The entire set of parameters can be denoted as $\Theta = \{\rho, \pi, \phi_c^i, \mu_c^i, \Sigma_c^i | 1 \leq i \leq K_c, c = 0, 1\}$. The EM algorithm involves iterations with each iteration consisting of two major steps: an E-step that calculates the marginal class posterior probabilities by message propagation based on old parameters, and an M-step that finds the optimal parameters to maximize the overall objective. We can use message propagation based on the sum and product algorithm Kschischang et al. (2001); Ronen et al. (1995). The main difference is that the calculation of messages related to the factor $P(\mathbf{x}_n | y_n)$ is based on the mixture of Gaussian distribution instead of a single-modal Gaussian distribution, as specified in Eq. 3.

After calculating the marginal posterior, we update model parameters by maximizing the posterior expectation of log-likelihood. Since we extend the probabilistic formulation of feature distribution to a mixture of Gaussian, we also need to revise the parameter update formula. The extended parameter

update formulas are shown by equations below. The symbol Θ_0 represents the old parameter values in the previous iteration of the EM algorithm, which is used in the calculation of messages and estimated posterior class probabilities. These calculated terms are then used to update the parameters in the formulas below to get a new Θ (the maximization step or M-step in the EM algorithm).

$$\rho = \frac{\sum_{n|P_n \neq \emptyset} \sum_{y_n} y_n (1 - y_n) P(y_n, y_{P_n} | \mathbf{X}, \Theta_0)}{\sum_{n|P_n \neq \emptyset} \sum_{y_n} y_n P(y_n, y_{P_n} | \mathbf{X}, \Theta_0)} \quad (4)$$

$$\pi = \frac{\sum_{n|P_n = \emptyset} \sum_{y_n} y_n P(y_n | \mathbf{X}, \Theta_0)}{\sum_{n|P_n = \emptyset} \sum_{y_n} P(y_n | \mathbf{X}, \Theta_0)} \quad (5)$$

$$\phi_c^i = \frac{\sum_{n \in \mathcal{O}} P(y_n = c | \mathbf{X}, \Theta_0) \gamma_c^i(x_n, \Theta_0)}{\sum_{i=1}^{K_c} \sum_{n \in \mathcal{O}} P(y_n = c | \mathbf{X}, \Theta_0) \gamma_c^i(x_n, \Theta_0)}, c = 0, 1 \quad (6)$$

$$\mu_c^i = \frac{\sum_{n \in \mathcal{O}} \mathbf{x}_n P(y_n = c | \mathbf{X}, \Theta_0) \gamma_c^i(x_n, \Theta_0)}{\sum_{n \in \mathcal{O}} P(y_n = c | \mathbf{X}, \Theta_0) \gamma_c^i(x_n, \Theta_0)}, c = 0, 1 \quad (7)$$

$$\Sigma_c^i = \frac{\sum_{n \in \mathcal{O}} (\mathbf{x}_n - \mu_c^i)(\mathbf{x}_n - \mu_c^i)^T P(y_n = c | \mathbf{X}, \Theta_0) \gamma_c^i(x_n, \Theta_0)}{\sum_{n \in \mathcal{O}} P(y_n = c | \mathbf{X}, \Theta_0) \gamma_c^i(x_n, \Theta_0)}, c = 0, 1, \quad (8)$$

$$\text{where } \gamma_c^i(\mathbf{x}_n, \Theta_0) = \frac{\phi_{0,c}^i \mathcal{N}(\mathbf{x}_n | \mu_{0,c}^i, \Sigma_{0,c}^i)}{\sum_{i=1}^{K_c} \phi_{0,c}^i \mathcal{N}(\mathbf{x}_n | \mu_{0,c}^i, \Sigma_{0,c}^i)}, c = 0, 1, \quad (9)$$

In the new parameter update formulas, the formulas for ρ and π are similar to the single-modal version. The main difference is related to parameters for feature distributions, such as ϕ_c^i , μ_c^i and Σ_c^i in Eqs 6–8. The key difference is the additional weighting terms $\gamma_c^i(x_n, \Theta_0)$ in Eq. 9. Intuitively, the intermediate variable $\gamma_c^i(x_n, \Theta_0)$ reflects the weight of a sample with feature \mathbf{x}_n belong to the i -th Gaussian component of class c . For example, in the parameter update formula for μ_c^i (Eq. 7), the updated mean for features in the i -th component of class c is based on the feature vector of each sample, averaged by the sample's weight on the i -th component of class c ($\gamma_c^i(x_n, \Theta_0)$). Similar interpretation can be given on the update formulas for ϕ_c^i and Σ_c^i .

Another issue is the parameter initialization before the EM iteration, the initial parameters of ϕ_c^i , μ_c^i and Σ_c^i can be initialized based on a small set of training samples (pixels) with known labels. This can be done by an EM clustering for feature values in each class. The cluster centroids can be initialized by randomly select K_c samples from the training samples in class c .

After parameter learning, we can use the same class inference algorithm based on message propagation (*max-sum* Rabiner (1989)). The calculation of messages is similar to before, except that the message related to $P(\mathbf{x}_n | y_n)$ are calculated based on the mixture of Gaussian distribution.

TABLE 2 | Dataset description.

Dataset	Training Set		Testing Set	
	Dry	Flood	Dry	Flood
Matthew, Kinston	5,000	5,000	48,071	47,967
Matthew, Grimesland	5,000	5,000	75,670	59,405

4 EXPERIMENTAL EVALUATION

4.1 Experiment Setup

In this section, we compared our proposed approach with baseline methods in related works on real-world datasets. Evaluation candidate methods are listed below. Note that we did not include data imputation methods (e.g., filling in mean feature values) due to its low capability of handling very limited observations. We used default parameters for open source tools. Experiments were conducted on a Dell workstation with Intel(R) Xeon(R) CPU E5-2687w v4 @ 3.00GHz, 64 GB main memory, and Windows 10.

- **Label propagation with structure (LP-Structure):** In the implementation of this baseline method, we used the maximum likelihood classifier (MLC) and gradient boosted model (GBM) respectively to pre-classify fully observed samples and then ran label propagation Wang and Zhang (2007) on the topography tree structure. We named them as **LP-Structure-MLC** and **LP-Structure-GBM**. The initial classifiers were from R packages.
- **EM with i.i.d. assumption (EM-i.i.d.):** In the implementation of this baseline method Ghahramani Z. and Jordan M. (1994), we treated missing features and unknown classes as latent variables and used the EM algorithm assuming that sample features follow i.i.d. Gaussian distribution in each class. Moreover, we assumed RGB (red, green, blue) features and elevation features are uncorrelated.
- **EM with structure:** This is our proposed approach. We treated unknown classes as latent variables and used the EM algorithm assuming that samples follow the topography tree dependency structure. The codes were implemented in C++. There are two configurations: single-modal feature distribution (**EM-Structure-Single**) and multi-modal feature distribution (**EM-Structure-Multi**).

Data Description: Our real-world data were collected from Kinston and Grimesland in North Carolina, 2016. The data include aerial imagery from NOAA National Geodetic Survey National Oceanic and Atmospheric Administration (2018a) with red, green, blue bands in a 2-m resolution and a digital elevation map from the University of North Carolina Libraries NCSU Libraries (2018). The test region size was 1743 by 1,349 pixels in Kinston and 2,757 by 3,853 pixels in Grimesland. The number of observed pixels was 31,168 in Kinston and 237,312 in Grimesland. The numbers of training and testing pixels are provided in Table 2.

TABLE 3 | Comparison on Mathew, Kinston flood data.

Classifiers	Class	Prec.	Recall	F	Avg. F
LP-Structure-GBM	Dry	0.91	0.56	0.69	0.74
	Flood	0.68	0.94	0.79	
LP-Structure-MLC	Dry	0.86	0.55	0.67	0.72
	Flood	0.67	0.91	0.77	
EM-i.i.d.	Dry	1.00	0.39	0.56	0.66
	Flood	0.62	1.00	0.76	
EM-Structure-Single	Dry	0.94	0.99	0.96	0.96
	Flood	0.99	0.94	0.96	

Evaluation Metrics: For classification performance evaluation, we used precision, recall, and F-score. For computational performance evaluation, we measured the running time costs in seconds.

4.2 Classification Performance Evaluation

We first compared methods on precision, recall, and F-score on the two real-world datasets. The results were summarized in **Tables 3, 4** respectively. On the Kinston dataset, EM algorithm with the i.i.d. assumption performed the worst with an average F-score of 0.66. The reason was that this method was not able to utilize the spatial structural constraint between sample classes. Its training process only updated the parameter of Gaussian feature distribution in each class. When predicting the classes of samples with only elevation feature, the method used only the learned Gaussian distribution of elevation feature on each class without considering spatial structure based on elevation values. On the same dataset, label propagation after pre-classification with the GBM model and the maximum likelihood classifier slightly outperformed the EM algorithm with the i.i.d. assumption. The main reason was that label propagation on the topography tree (split tree) structure utilized the physical constraint between sample classes when inferring the classes of unobserved samples without RGB features. However, label propagation still showed significant errors, particularly in the low recall on the dry class. Through analyzing the predicted map, we observed that the label propagation algorithm was very sensitive to the pre-classified class labels on the observed samples in the test region. Errors in the pre-classification phase may propagate into unobserved samples (those without RGB feature values). In label propagation methods, once the errors were propagated to unobserved samples, they were hard to be reverted. This was different from the EM algorithm, which could update the probabilities in iterations. We did not report the results of label propagation on a grid graph structure (only considering spatial neighborhood structure without physics-aware constraint) due to poor results. Our model based on the EM algorithm assuming structural dependency between class labels performed the best with an average F-score of 0.96. The main reason was that our model could leverage the physical constraint to infer unobserved samples, and also could effectively update sample probabilities during iterations with the EM algorithm. In our model, we used training samples to initialize the parameters of the Gaussian distribution of sample features in each class. Based on the

reasonable initial parameters, we can have a reasonable estimation of the posterior class probabilities of all samples in the test region. Based on the posterior class probabilities, the distribution parameters could be further updated. The representative training samples helped make sure that parameter iterations would converge in the right path.

Similar results were observed on the Grimesland dataset. In the label propagation method, pre-classification based on GBM performed worse than pre-classification based on MLC. The reason may be due to overfitting of GBM compared with MLC when predicting initial labels on the fully observed samples. The EM algorithm with the i.i.d. assumption performed slightly better on this dataset. The reason is likely that the final prediction of classes of the unobserved samples (with only elevation feature but without RGB features) was based on a slightly better fitted normal distribution. Our model showed the best performance with an F-score of 0.97.

4.2.1 The Effect of Model Initial Parameters

We also analyzed the sensitivity of our proposed model on different initial parameter settings. The parameters of μ_c and Σ_c were estimated from training data, but parameters ρ and π were from user input. Since ρ captured the transitional probability of a sample being flood given its parents were all flood, its value should be very high (close to 1) due to spatial autocorrelation. π is the initial class prior probabilities for samples without parent nodes (local lowest location). We could set it close to 0.5. We tested the sensitivity of our model to different initial values of ρ and π on the Kinston dataset. We first fixed ρ as 0.999 and varied the value of π from 0.1 to 0.9. Then we fixed π as 0.3 and varied the value of ρ from 0.9. The results were shown in **Figure 4**. We can see that the model was generally not very sensitive to the initial parameter values. For parameter ρ , as long as $1 - \rho$ was smaller than 0.01 (ρ greater than 0.99), the converged F-score was good. For parameter π , the results were consistently good for our model with an initial π between 0.1 and 0.9. The main reason was that π influenced only a small number of samples at the local lowest locations on the elevation map.

The parameter iterations of our model were shown in **Figure 5**. The model converged fast with only 20 iterations. Due to the space limit, we only show the iterations of parameters ρ and π .

TABLE 4 | Comparison on Mathew, Grimesland flood data.

Classifiers	Class	Prec.	Recall	F	Avg. F
LP-Structure-GBM	Dry	0.81	0.60	0.69	0.70
	Flood	0.61	0.82	0.70	
LP-Structure-MLC	Dry	0.90	0.75	0.82	0.81
	Flood	0.73	0.90	0.81	
EM-i.i.d.	Dry	0.83	0.74	0.78	0.77
	Flood	0.71	0.80	0.75	
EM-Structure-Single	Dry	0.99	0.96	0.97	0.97
	Flood	0.95	0.99	0.97	

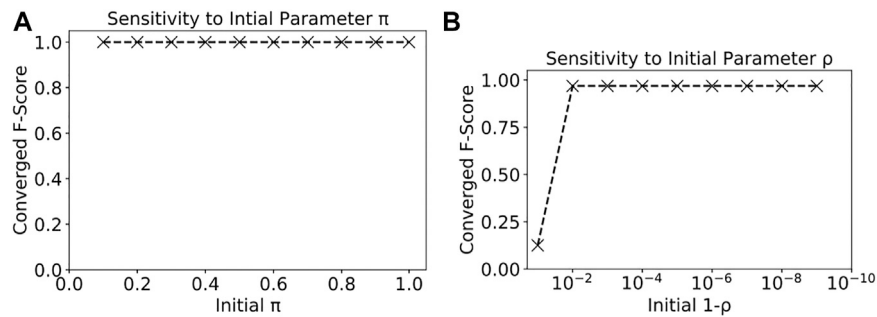


FIGURE 4 | Sensitivity of our model to initial parameters π and ρ .

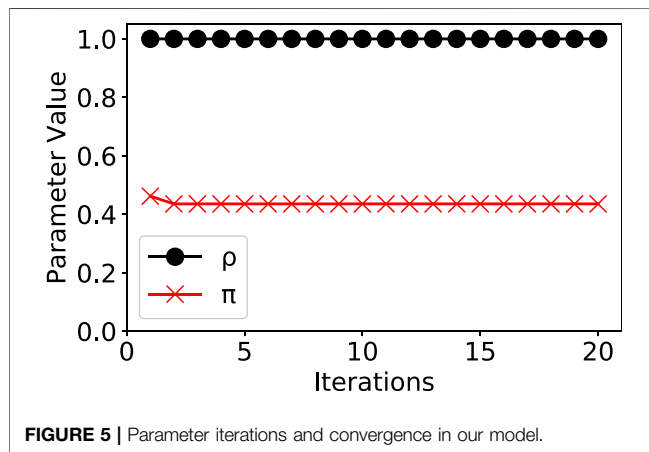


FIGURE 5 | Parameter iterations and convergence in our model.

4.3 Computational Performance Evaluation

We also evaluated the computational performance of our model learning and inference on different input data sizes. We used the Grimesland dataset to test the effect of different test region sizes. We varied the region size from around 2 million pixels to over 10 million pixels. The computational time costs of our model were shown in **Figure 6**. It can be seen that the time cost grows almost linearly with the size of the test region. This was because our learning and inference algorithms involve tree traversal operations with a linear time complexity on the tree size (the number of pixels on the test region). The model is computationally efficient. It classified around 10 million pixels in around 2 min.

We further analyzed the time costs of different components in our model, including split tree construction, model parameter learning, and class inference. The results are summarized in **Table 5**. We analyzed the results on both datasets (same as the settings in **Tables 3, 4**). Results showed that tree construction and class inference took less time than parameter learning. This was because the learning involves multiple iterations of message propagation (tree traversal operations).

4.4 Additional Comparison of EM Structured Between Single-modal and Multi-modal

This subsection provides additional evaluations on the comparison of our EM structured algorithms between the

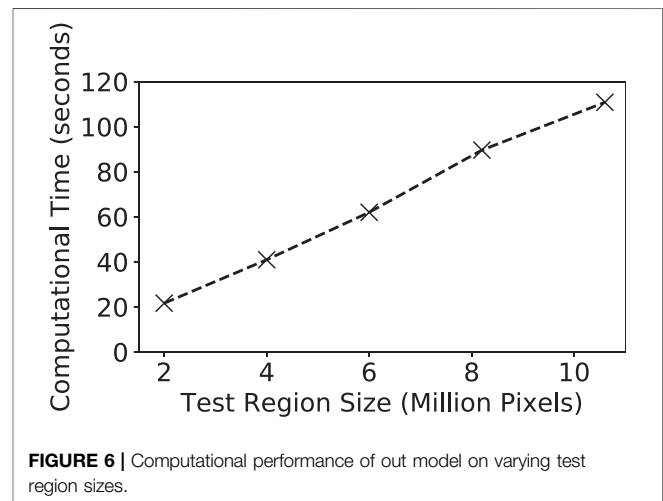


FIGURE 6 | Computational performance of our model on varying test region sizes.

single-modal feature distribution (conference version) and multi-modal feature distribution (journal extension). In previous experiments, we found that our EM structured (single-modal) outperformed several baseline methods when feature distribution among training samples in each class is single-modal. In this experiment, we used the same two test regions and observed polygons as previous experiments (**Table 2**), but collected training samples that exhibit multi-modal feature distributions in each class. The new training dataset still contains 5,000 samples for dry class and 5,000 samples for flood class.

For both single-modal and multi-modal, we fixed the parameter π (prior class probability for leaf nodes) as 0.5, ρ as 0.999 (class transitional probability between a node and its parents) and the maximum number of iterations in EM as 40. We compared different candidate methods on their precision, recall, and F-score on the two test regions.

The new results were summarized in **Tables 6, 7**. On Kinston flood data, we can see that the performance of the three baseline methods was poor (F-score below 0.72). Among these methods, GBM with label propagation (LP) was worse (F-score around 0.34), likely due to serious overfitting issues from the varying feature distributions between the training and test samples. The

TABLE 5 | Time costs of different components of our model on two datasets (seconds)

	Kinston	Grimesland
Tree construction part	3.2	8.39
Parameter learning part	25.74	86.79
Class inference part	3.8	15.62
Total time costs	32.74	110.80

performance of the proposed EM-structure model with single-modal feature distribution (conference version) significantly degraded (F-score dropped to 0.34). The reason was probably that the feature values of training samples follow a multi-modal distribution, violating the original assumption that features are single-modal Gaussian in each class. Thus, the parameter iterations were not ineffective during the learning iterations on observed samples in the test region. In contrast, the EM-structured with multi-modal distribution was more robust with the best performance (F-score around 0.97). Similar results were seen in the Grimesland dataset. The three baseline methods generally performed poorly, except for the label propagation with the maximum likelihood method achieving an F-score of 0.91. The reason was likely that the maximum likelihood classifier was simple and less prone to overfitting in initial label prediction. The EM-structured model with a single modal performed poorly (F-score around 0.28) due to the wrong assumption on feature distribution. In contrast, the EM-structured model with a multi-modal assumption performed the best with an F-score around 0.95.

We also visualize the predicted class maps on the Kinston dataset in **Figure 7** for interpretation. The input limited feature observations on spectral pixels in the test region are shown in **Figure 7A**. The observed pixels cover parts of the boundary of the flood region in the test region. The digital elevation image that was used to construct a tree structure based on physical constraint is shown in **Figure 7B**. From the topography of the area, we can see that the test region has a floodplain alongside a river channel that is spreading through the lower half of the image. The top area of the image have higher elevation. The predictions of label propagation on top of GBM (LP-Structure-GBM) and MLC (LP-Structure-MLC) are in **Figures 7C,D**. We can see a significant amount of misclassification (e.g., the flood area in

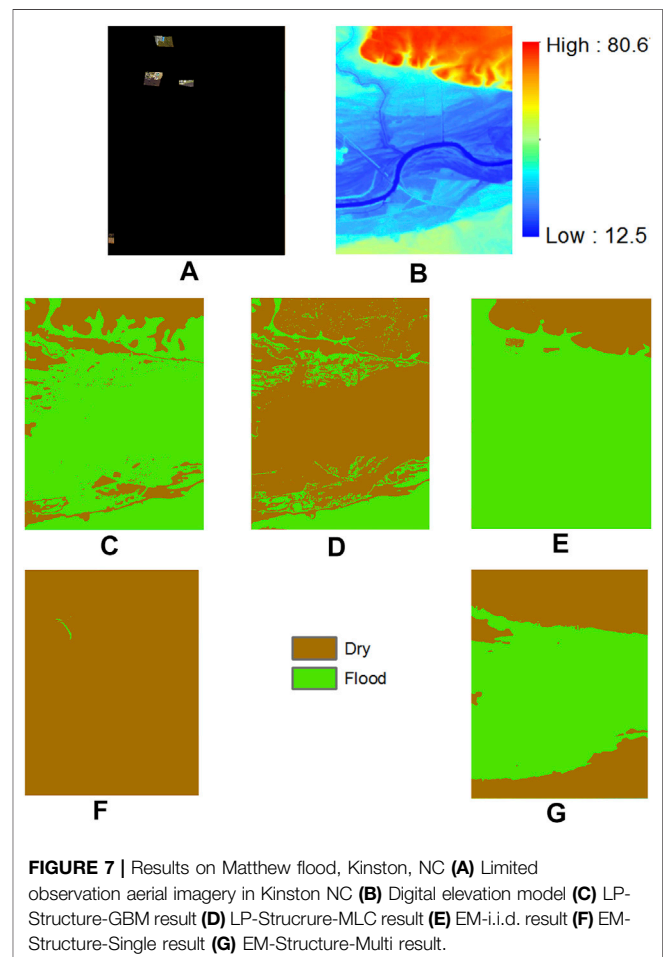
TABLE 6 | Comparison on Mathew, Kinston flood data.

Classifiers	Class	Prec.	Recall	F	Avg. F
LP-Structure-GBM	Dry	0.44	0.73	0.55	0.34
	Flood	0.26	0.09	0.14	
LP-Structure-MLC	Dry	0.85	0.56	0.67	0.72
	Flood	0.67	0.90	0.77	
EM-i.i.d.	Dry	1.00	0.40	0.57	0.67
	Flood	0.62	1.00	0.77	
EM-Structure-Single	Dry	0.50	1.00	0.66	0.34
	Flood	1.00	0.01	0.01	
EM-Structure-Multi (Our extension)	Dry	0.94	0.99	0.97	0.97
	Flood	0.99	0.94	0.97	

TABLE 7 | Comparison on Mathew, Grimesland flood data.

Classifiers	Class	Prec.	Recall	F	Avg. F
LP-Structure-GBM	Dry	0.74	0.53	0.62	0.71
	Flood	0.75	0.88	0.81	
LP-Structure-MLC	Dry	0.84	0.96	0.90	0.91
	Flood	0.97	0.89	0.93	
EM-i.i.d.	Dry	0.57	0.87	0.69	0.70
	Flood	0.88	0.590	0.70	
EM-Structure-Single	Dry	0.38	1.00	0.55	0.28
	Flood	1.00	0.00	0.00	
EM-Structure-Multi (Our extension)	Dry	0.91	0.96	0.94	0.95
	Flood	0.97	0.94	0.96	

the middle of **Figure 7C** is mistakenly predicted as dry, the dry area at the bottom of **Figure 7B** is mistakenly classified as flood). The reason is probably that initial class predictions by these models are noisy and these noisy labels further spread out during label propagation. The EM i.i.d. algorithm also shows significant errors in the bottom part of the image, where the elevation is lower. The reason is probably that the learned decision boundary on the elevation feature in the EM i.i.d. algorithm is inaccurate



due to limited sample observations. The EM structured model with single-modal feature distribution performs poorly, classifying almost the entire area as dry. The reason is likely that the feature distribution in the model during parameter learning iterations is wrong, making the inferred classes largely wrong. In contrast, the EM structured model with multi-modal feature distribution identified the complete flood boundary.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we investigate the flood extent mapping application of spatial classification in a case whereby samples have limited feature observations. We extend our recent approach that incorporates physics-aware structural constraints (e.g., water flow directions on geographic terrains) into model structural representation. We propose efficient algorithms for model parameter learning and class inference. The extended model allows for multi-modal feature distribution with the mixture Gaussian model. Evaluations on flood mapping datasets show that the proposed approach outperformed existing methods in classification accuracy.

In future work, we plan to extend our proposed model to address other problems such as integrating noisy and incomplete observations such as volunteered geographic information (VGI).

REFERENCES

- Batista, G. E., and Monard, M. C. (2002). A Study of K-Nearest Neighbour as an Imputation Method. *HIS* 87, 48.
- Bengio, Y., and Gingras, F. (1996). "Recurrent Neural Networks for Missing or Asynchronous Data," in *Advances in Neural Information Processing Systems*, 395–401.
- Carr, H., Snoeyink, J., and Axen, U. (2003). Computing Contour Trees in All Dimensions. *Comput. Geometry* 24, 75–94. doi:10.1016/s0925-7721(02)00093-7
- Chechik, G., Heitz, G., Elidan, G., Abbeel, P., and Koller, D. (2007). "Max-margin Classification of Incomplete Data," in *Advances in Neural Information Processing Systems*, 233–240. doi:10.7551/mitpress/7503.003.0034
- Cline, D. (2009). *Integrated Water Resources Science and Services: An Integrated and Adaptive Roadmap for Operational Implementation*. National Oceanic and Atmospheric Administration.
- Du, J., Zhang, Y., Wang, P., Leopold, J., and Fu, Y. (2019). "Beyond Geo-First Law: Learning Spatial Representations via Integrated Autocorrelations and Complementarity," in 2019 IEEE International Conference on Data Mining (ICDM) (New York: IEEE), 160–169. doi:10.1109/icdm.2019.00026
- Edelsbrunner, H., and Harer, J. (2010). *Computational Topology: An Introduction*. Providence, RI: American Mathematical Society.
- Edelsbrunner, H., and Koehl, P. (2005). The Geometry of Biomolecular Solvation. *Comb. Comput. geometry* 52, 243–275.
- García-Laencina, P. J., Sancho-Gómez, J.-L., and Figueiras-Vidal, A. R. (2013). Classifying Patterns with Missing Values Using Multi-Task Learning Perceptrons. *Expert Syst. Appl.* 40, 1333–1341. doi:10.1016/j.eswa.2012.08.057
- García-Laencina, P. J., Sancho-Gómez, J.-L., and Figueiras-Vidal, A. R. (2010). Pattern Classification with Missing Data: A Review. *Neural Comput. Applic* 19, 263–282. doi:10.1007/s00521-009-0295-6
- Ghahramani, Z., and Jordan, M. I. (1994b). "Supervised Learning from Incomplete Data via an Em Approach," in *Advances in Neural Information Processing Systems*, 120–127.
- Ghahramani, Z., and Jordan, M. (1994a). *Learning from Incomplete Data* (Tech. Rep. No. Aim-1509).
- Günther, D., Boto, R. A., Contreras-Garcia, J., Piquemal, J.-P., and Tierny, J. (2014). Characterizing Molecular Interactions in Chemical Systems. *IEEE Trans. Vis. Comput. Graphics* 20, 2476–2485. doi:10.1109/tvcg.2014.2346403
- He, W., and Jiang, Z. (2020). Semi-supervised Learning with the Em Algorithm: A Comparative Study between Unstructured and Structured Prediction. *IEEE Trans. Knowledge Data Eng.* doi:10.1109/tkde.2020.3019038
- Jiang, K., Chen, H., and Yuan, S. (2005). "Classification for Incomplete Data Using Classifier Ensembles," in 2005 International Conference on Neural Networks and Brain (Piscataway, NJ: IEEE), 559–563.
- Jiang, Z. (2019). A Survey on Spatial Prediction Methods. *IEEE Trans. Knowledge Data Eng.* 31 (9), 1645–1664. doi:10.1109/TKDE.2018.2866809
- Jiang, Z., and Sainju, A. M. (2021). A Hidden Markov Tree Model for Flood Extent Mapping in Heavily Vegetated Areas Based on High Resolution Aerial Imagery and Dem: A Case Study on hurricane Matthew Floods. *Int. J. Remote Sensing* 42, 1160–1179. doi:10.1080/01431161.2020.1823514
- Jiang, Z., and Sainju, A. M. (2019). "Hidden Markov Contour Tree," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, August 4–8, 2019 ((ACM), KDD '19). doi:10.1145/3292500.3330878
- Jiang, Z., and Shekhar, S. (2017). *Spatial Big Data Science*. Berlin: Springer International Publishing. doi:10.1007/978-3-319-60195-3
- Jiang, Z. (2020). Spatial Structured Prediction Models: Applications, Challenges, and Techniques. *IEEE Access* 8, 38714–38727. doi:10.1109/access.2020.2975584
- Jiang, Z., Xie, M., and Sainju, A. M. (2019). Geographical Hidden Markov Tree. *IEEE Trans. Knowledge Data Eng.* doi:10.1109/tkde.2019.2930518
- Karpatne, A., Jiang, Z., Vatsavai, R. R., Shekhar, S., and Kumar, V. (2016). Monitoring Land-Cover Changes: A Machine-Learning Perspective. *IEEE Geosci. Remote Sens. Mag.* 4, 8–21. doi:10.1109/mgrs.2016.2528038
- Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor Graphs and the Sum-Product Algorithm. *IEEE Trans. Inform. Theor.* 47, 498–519. doi:10.1109/18.910572
- Little, R. J., and Rubin, D. B. (2019). *Statistical Analysis with Missing Data*. New York, NY: John Wiley & Sons.

We also plan to explore incorporating deep learning into our framework to learn more complex feature distributions.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://geodesy.noaa.gov/storm_archive/storms/matthew/index.html.

AUTHOR CONTRIBUTIONS

AS is the first author, who conducted most of the research work. He is a co-author who contributed to some experiments. ZJ is the PI of the project who helps in the idea discussions and writing. DY and HC are collaborators who contributed ideas and writing.

FUNDING

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. IIS-1850546, IIS-2008973, and CNS-1951974, and the National Oceanic and Atmospheric Administration (NOAA).

- McLachlan, G., and Krishnan, T. (2007). *The EM Algorithm and Extensions*. New York, NY: John Wiley & Sons.
- Merwade, V., Olivera, F., Arabi, M., and Edleman, S. (2008). Uncertainty in Flood Inundation Mapping: Current Issues and Future Directions. *J. Hydrol. Eng.* 13, 608–620. doi:10.1061/(asce)1084-0699(2008)13:7(608)
- National Oceanic and Atmospheric Administration (2018a). Data and Imagery from NOAA's National Geodetic Survey. Available at: <https://www.ngs.noaa.gov>.
- National Oceanic and Atmospheric Administration (2018b). National Water Model: Improving NOAA's Water Prediction Services. Available at: <http://water.noaa.gov/documents/wrn-national-water-model.pdf>.
- NCSU Libraries (2018). LIDAR Based Elevation Data for North Carolina. Available at: <https://www.lib.ncsu.edu/gis/elevation>.
- Pelckmans, K., De Brabanter, J., Suykens, J. A. K., and De Moor, B. (2005). Handling Missing Values in Support Vector Machine Classifiers. *Neural Networks* 18, 684–692. doi:10.1016/j.neunet.2005.06.025
- Quinlan, J. R. (2014). *C4.5: Programs for Machine Learning*. Amsterdam: Elsevier.
- Quinlan, J. R. (1989). "Unknown Attribute Values in Induction," in Proceedings of the sixth international workshop on Machine learning (Amsterdam: Elsevier), 164–168. doi:10.1016/b978-1-55860-036-2.50048-5
- Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEEE* 77, 257–286. doi:10.1109/5.18626
- Ronen, O., Rohlicek, J. R., and Ostendorf, M. (1995). Parameter Estimation of Dependence Tree Models Using the Em Algorithm. *IEEE Signal. Process. Lett.* 2, 157–159. doi:10.1109/97.404132
- Rubin, D. B. (2004). *Multiple Imputation for Nonresponse in Surveys*. New York, NY: John Wiley & Sons.
- Sainju, A. M., He, W., and Jiang, Z. (2020a). A Hidden Markov Contour Tree Model for Spatial Structured Prediction. *IEEE Trans. Knowledge Data Eng.* doi:10.1109/TKDE.2020.3002887
- Sainju, A. M., He, W., Jiang, Z., and Yan, D. (2020b). "Spatial Classification with Limited Observations Based on Physics-Aware Structural Constraint," in Proc. AAAI Conf. Artif. Intell (Cambridge, MA: AAAI), 1–8. doi:10.1609/aaai.v34i01.5436
- Schafer, J. L. (1997). *Analysis of Incomplete Multivariate Data*. Boca Raton, FL: Chapman and Hall/CRC.
- Shekhar, S., Jiang, Z., Ali, R., Eftelioglu, E., Tang, X., Gunturi, V., et al. (2015). Spatiotemporal Data Mining: a Computational Perspective. *Ijgi* 4, 2306–2338. doi:10.3390/ijgi4042306
- Smola, A. J., Vishwanathan, S., and Hofmann, T. (2005). "Kernel Methods for Missing Variables," in AISTATS (Bridgetown, Barbados: Citeseer).
- Wales, D. (2003). *Energy Landscapes: Applications to Clusters, Biomolecules and Glasses*. Cambridge: Cambridge University Press.
- Wales, D. J., Miller, M. A., and Walsh, T. R. (1998). Archetypal Energy Landscapes. *Nature* 394, 758–760. doi:10.1038/29487
- Wang, F., and Zhang, C. (2007). Label Propagation through Linear Neighborhoods. *IEEE Trans. Knowledge Data Eng.* 20, 55–67.
- Wang, J., Gao, Y., Züfle, A., Yang, J., and Zhao, L. (2018a). "Incomplete Label Uncertainty Estimation for Petition Victory Prediction with Dynamic Features," in 2018 IEEE International Conference on Data Mining (ICDM) (Piscataway, NJ: IEEE), 537–546. doi:10.1109/icdm.2018.00069
- Wang, P., Fu, Y., Zhang, J., Li, X., and Lin, D. (2018b). Learning Urban Community Structures. *ACM Trans. Intell. Syst. Technol.* 9, 1–28. doi:10.1145/3209686
- Webb, G. I. (1998). "The Problem of Missing Values in Decision Tree Grafting," in Australian Joint Conference on Artificial Intelligence (Berlin: Springer), 273–283. doi:10.1007/bfb0095059
- Williams, D., Liao, X., Xue, Y., Carin, L., and Krishnapuram, B. (2007). On Classification with Incomplete Data. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 427–436. doi:10.1109/tpami.2007.52
- Xie, M., Jiang, Z., and Sainju, A. M. (2018). "Geographical Hidden Markov Tree for Flood Extent Mapping," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York (ACM), KDD '18), 2545–2554. doi:10.1145/3219819.3220053
- Yang, X., and Jin, W. (2010). Gis-based Spatial Regression and Prediction of Water Quality in River Networks: A Case Study in Iowa. *J. Environ. Manage.* 91, 1943–1951. doi:10.1016/j.jenvman.2010.04.011
- Yoon, S.-Y., and Lee, S.-Y. (1999). Training Algorithm with Incomplete Data for Feed-Forward Neural Networks. *Neural Process. Lett.* 10, 171–179. doi:10.1023/a:1018772122605
- Zhang, Y., Fu, Y., Wang, P., Li, X., and Zheng, Y. (2019). "Unifying Inter-region Autocorrelation and Intra-region Structures for Spatial Embedding via Collective Adversarial Learning," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1700–1708. doi:10.1145/3292500.3330972
- Zhou, J., Yuan, L., Liu, J., and Ye, J. (2011). "A Multi-Task Learning Formulation for Predicting Disease Progression," in Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 814–822. doi:10.1145/2020408.2020549

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Sainju, He, Jiang, Yan and Chen. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.