A Hidden Markov Forest Model for Terrain-Aware Flood Inundation Mapping from Earth Imagery

Zhe Jiang [*]   Yupu Zhang [*]   Saugat Adhikari [†]   Da Yan [†]   Arpan Man Sainju [‡]

Xiaowei Jia [§]   Yiqun Xie [¶]

Abstract

Flood inundation mapping from Earth imagery plays a vital role in rapid disaster response and national water forecasting. However, the problem is non-trivial due to significant imagery noise and obstacles, complex spatial dependency on 3D terrains, spatial non-stationarity, and high computational cost. Existing machine learning approaches are mostly terrain-unaware and are prone to produce spurious results due to imagery noise and obstacles, requiring significant efforts in post-processing. Recently, several terrain-aware methods were proposed that incorporate complex spatial dependency (e.g., water flow directions on 3D terrains) but they assume that the inferred flood surface level is spatially stationary, making them insufficient for a large heterogeneous geographic area. To address these limitations, this paper proposes a novel spatial learning framework called hidden Markov forest, which decomposes a large heterogeneous area into local stationary zones, represents spatial dependency on 3D terrains via zonal trees (forest), and jointly infers the class map in different zonal trees with spatial regularization. We design efficient inference algorithms based on dynamic programming and multi-resolution filtering. Evaluations on real-world datasets show that our method outperforms baselines and our proposed computational refinement significantly reduces the time cost.

1 Introduction

Given Earth imagery with spectral features and a 3D terrain map (e.g., Digital Elevation Model), the flood mapping problem aims to classify the pixels into flood and dry classes. The problem is important in rapid disaster response and national water forecasting [1, 2, 3]. For example, during a hurricane flood event (e.g., Harvey in 2017), the first responders need to know where the flood water is in order to estimate the damage to

critical infrastructure and plan rescue efforts. Accurate flood extent maps derived from Earth observation imagery can significantly enhance the situational awareness of disaster management agencies. Another important application is national water forecasting. Currently, the NOAA hyper-resolution National Water Model [4] can forecast the flow of over 2.7 million river reaches in the entire continental U.S. [5], but the model is only validated and calibrated by observations at over 7000 river gauges. Observation-derived flood maps from Earth imagery can potentially fill the gap in validating and calibrating the national water forecasting models.

However, the problem poses several major technical challenges. First, Earth imagery features often contain significant noise and obstacles (e.g., shadows, clouds, tree canopies), making a classifier confused about whether a location is flooded or not. For example, the tree canopies in highly vegetated areas often obscure the view of flood water beneath them. Second, a complex spatial dependency structure exists in the target class map. Due to gravity, the spatial distribution of flood water is constrained by water flow directions on 3D terrains. Without incorporating such a spatial dependency structure, a classifier can produce spurious output patterns (e.g., a flooded location above a nearby dry location). Third, there is also the effect of spatial non-stationarity [6]. Specifically, the flood surface elevation level is not "flat" (constant) across a large geographic area. Therefore, the spatial dependency structure needs to be represented in multiple local zones with spatial continuity assumed across nearby zones. Finally, modeling such a complex spatial dependency structure for a large number of pixel locations leads to a high potential computational cost. In high-resolution imagery, we can easily have millions to billions of pixels for one city.

Existing relevant machine learning methods can be categorized into terrain-unaware methods and terrain-aware methods. The vast majority of methods are terrain unaware, including non-spatial classifiers that assume an identical and independent distribution (i.i.d.) of sample pixels, e.g., random forest [7], and spatial clas-

[*]University of Florida, (zhe.jiang@ufl.edu, y.zhang1@ufl.edu).

[†]University of Alabama at Birmingham, Birmingham, AL, 35294 (saugat@uab.edu,yanda@uab.edu).

[‡]Middle Tennessee State University (arpan.sainju@mtsu.edu).

[§]University of Pittsburgh (xiaowei@pitt.edu)

[¶]University of Maryland, College Park (xie@umd.edu)

sifiers that incorporate spatial autocorrelation but only at the neighborhood level [8, 9, 10, 11], e.g., deep image segmentation models [12, 13, 14]. These terrain-unaware methods are often vulnerable to the impact of significant noise and obstacles in imagery features. For example, the visual signatures of tree canopies in flood and dry areas are very similar, making it hard to distinguish the actual classes. One interesting relevant work [15] learns topological structure and outliers (or noises) detection in a common network embedding framework and has been evaluated in flood mapping. The difference from our work is that network embedding implicitly incorporates topological structure constraints while we focus on explicitly capturing the topography constraints within a model architecture. Recently, some terrain-aware spatial learning methods have been developed, e.g., geographic hidden Markov tree family [16, 17], which explicitly represent water flow directions on 3D terrains with a tree structure and learn a probabilistic graphical model to infer pixel classes (flood or dry). Results have shown that these methods can significantly reduce misclassification errors due to imagery noise and obstacles. However, they assume spatial stationarity and create a single global tree structure for the entire geographic area (i.e., assuming the flood surface is "flat" with an equal elevation). Thus, these methods are insufficient for a large geographic area with heterogeneous sub-zones.

To fill the gap, this paper proposes a novel spatial machine learning framework called hidden Markov forest, which decomposes the complex spatial dependency on non-stationary (heterogeneous) terrains into zonal tree structures, and jointly infers the classes of pixel nodes from different zones with spatial regularization (to reflect the spatial autocorrelation between nearby zones). To speed up the inference for a large number of pixel locations, we also design efficient learning algorithms based on dynamic programming and multi-resolution filtering. In summary, we make the following contributions:

- We propose a novel hidden Markov forest model, which generalizes existing terrain-aware hidden Markov tree models from a single global topography tree to heterogeneous zonal tree structures.

- In the learning framework, we design an objective function that incorporate the spectral signatures learned from a base classifier (e.g., random forest, U-Net), the complex spatial dependency of water flow directions in zonal tree structures and the spatial regularization across nearby zones.

- To overcome the computational bottleneck, we design efficient algorithms for class inference, including using dynamic programming and multi-resolution filtering to prune out redundant computation.

(a) Eight consecutive sample locations in 1D
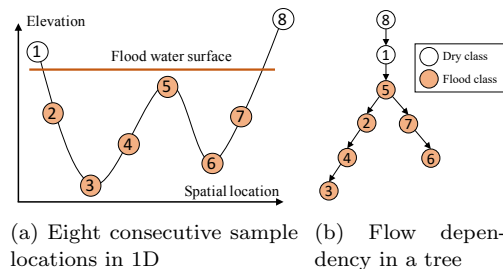
(b) Flow dependency in a tree

Figure 1: Illustration of partial order flow dependency

cluding using dynamic programming and multi-resolution filtering to prune out redundant computation.

- We evaluate the proposed model on real-world datasets for flood mapping. Results show that the proposed model outperforms multiple baseline methods in classification accuracy, and our proposed computational refinement methods significantly reduce the time cost.

2 Problem Statement

We start with some preliminaries and then formally define our problem.

A **spatial raster framework** is a tessellation of a 2D plane into a regular grid of N pixels. It can contain m non-spatial explanatory feature layers (e.g., spectral bands in Earth imagery), one elevation layer (i.e., 3D terrains), and one class layer (e.g., flood, dry). Each pixel in the raster framework is a **spatial data sample**, noted as $\mathbf{s_n} = (\mathbf{x}_n, \phi_n, y_n)$, 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 one feature layer, $\phi_n \in \mathbb{R}$ is a pixel's elevation value, and $y_n \in \{0, 1\}$ is a binary class label. A raster framework with all samples is noted as $\mathcal{F} = \{\mathbf{s_n} | n \in \mathbb{N}, 1 \leq n \leq N\}$, non-spatial explanatory features of all samples are noted as $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^T$, the elevation layer is noted as $\mathbf{\Phi} = [\phi_1, ..., \phi_N]^T$, and the class layer is noted as $\mathbf{Y} = [y_1, ..., y_N]^T$.

Spatial dependency exists between pixels based on the topography of 3D terrains [16]. Formally, a *flow dependency* $\mathbf{s}_j \rightsquigarrow \mathbf{s}_i$ exists if and only if there exists a connected path of pixels $< \mathbf{s}_i, \mathbf{s}_{p_1}, \mathbf{s}_{p_2}, ..., \mathbf{s}_{p_l}, \mathbf{s}_j >$ such that $\phi_i \leq \phi_j$, $\phi_{p_k} \leq \phi_j$, for any $1 \leq k \leq l$. In other words, water from $\mathbf{s}_j$ can flow to $\mathbf{s}_i$ since all locations along the path has a lower elevation. Figure 1(a) shows an illustrative example with eight spatially adjacent pixels in one-dimensional space (the numbers are node ids). Due to gravity, if node $\mathbf{s}_5$ in the middle is

flood, its nearby pixels with lower elevations including $\mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4, \mathbf{s}_6, \mathbf{s}_7$ should also be *flood*, even if their pixel features indicate otherwise. Thus, there exists flow dependency such as $\mathbf{s}_5 \leadsto \mathbf{s}_4$, $\mathbf{s}_5 \leadsto \mathbf{s}_2$.

Flow dependency across all pairs of pixels in a raster framework can be represented by a (reverse) tree structure, which is called *spatial dependency tree* or *flow tree*. The tree structure removed some redundant dependency between pixel locations. In the example of Figure 1(b), flow dependency $\mathbf{s}_2 \leadsto \mathbf{s}_3$ is redundant given dependency $\mathbf{s}_2 \to \mathbf{s}_4$ and $\mathbf{s}_4 \to \mathbf{s}_3$. Note that we use straight arrows to represent non-redundant (tree) edges. If $\mathbf{s}_1$ and $\mathbf{s}_8$ had an equal elevation, we can enforce an arbitrary order between them.

We now formally define the problem as follows.
Input:
• Spatial raster framework $\mathcal{F} = \{\mathbf{s}_n | n \in \mathbb{N}, 1 \le n \le N\}$
• Elevation values of all samples: $\mathbf{\Phi} = [\phi_1, ..., \phi_N]^T$
• Non-spatial explanatory features of all samples: $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^T$
Output: A spatial learning model $f : \hat{\mathbf{Y}} = f(\mathbf{X}, \mathbf{\Phi})$
Objective: minimize classification errors
Constraint:
• Features $\mathbf{X}$ contain rich noise and obstacles
• Sample classes follow flow dependency on 3D terrains
• Sample classes are binary, $y_n \in \{0, 1\}$

The spatial classification problem focuses on incorporating the complex spatial dependency structure on 3D terrains. The problem can be easily generalized to a dual post-processing formulation that takes initial class probabilities of all samples from a base classifier $\mathbf{Y} = [P(y_1|\mathbf{x}_1), ..., P(y_N|\mathbf{x}_N)]^T$. In this case, the feature representation of Earth imagery pixels has already been learned by an initial classifier (e.g., through a deep neural network [12]). The output model will be $f : \hat{\mathbf{Y}} = f(\mathbf{Y}, \mathbf{\Phi})$.

3 Approach

To solve the problem, we need to overcome several non-trivial technical challenges. First, in order to make the classified map robust to significant noise and obstacles in imagery features (reflected in erroneous initial class probabilities), the framework needs to incorporate complex spatial dependency constraints based on water flow directions on 3D terrains. Second, such terrain-guided spatial dependency constraints are spatially non-stationary. Thus, a single global tree for the entire area is insufficient. Third, according to the first law of geography, "everything is related to everything else, but near things are more related than distant things". Thus, the inferred class map should conform to spatial continuity across nearby regions (e.g., smoothness of flood surface levels). Finally, the potential computational cost is high
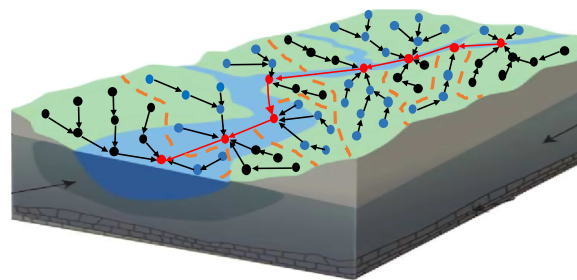

Figure 2: Illustration of zonal tree dependency structures on non-stationary 3D terrains. The red nodes are locations along a river channel. Dash lines are zone boundaries. (best in color)

due to the large number of pixels.

To address the above challenges, we partition the entire 3D terrain into different zones based on their target water drainage points and represent water flow directions by zonal tree structures (i.e., a forest). Based on the zonal tree structure, we provide a probabilistic formulation of all samples with a novel hidden Markov forest model. We design an objective function based on both the overall likelihood and a spatial regularization between nearby zones. Finally, in order to reduce computational bottlenecks in class inference, we design efficient algorithms based on dynamic programming and multi-resolution filtering. We next introduce the details for each component.

3.1 Representation of spatial dependency on non-stationary terrains in zonal tree structures
Here we want to design a structural representation of spatial dependency on 3D terrains. In our recent works [16, 17], we construct a global flow tree to represent the topography constraint for the entire area based on computational topology, assuming that the class map is spatially stationary (i.e., the flood surface is "flat" with an equal elevation level, as shown in Figure 1). However, such an assumption is not always true in reality due to spatial non-stationarity. Specifically, the flood surface elevation level tends to be higher upstream than downstream. Therefore, we need a structural representation that can capture the varying flood surface elevation levels along different sub-regions. In order to incorporate spatial non-stationarity, we partition the entire terrain into different zones based on their target drainage points along a river channel, leading to zonal tree structures. Specifically, nodes that flow to the same river location (drainage point) belong to the same zone because rising flood levels from this river node will first spill over to locations in this zone. Figure 2 shows an example of a 3D terrain surface with zone partition along river streams. The black arrows are flow dependency within zonal trees. The red dash lines show the zone

partition boundary. The solid red arrows show the direction of the river channel that links multiple zonal trees together (also illustrated in Figure 3).

In order to preprocess the 3D terrain map (elevation) into a zonal tree structure, we use a popular computational tool in hydrology called Terrain Analysis Using Digital Elevation Models (TauDEM) [18, 19]. The tool can establish water flow dependency by computing the shortest path routes on 3D terrains. We omit the details since the zonal tree construction can be considered as preprocessing. We focus on the probabilistic formulation and class inference.

3.2 Probabilistic formulation and objective function Based on the zonal tree dependency structure, we propose a probabilistic formulation of the distribution of all nodes (locations). We propose a novel probabilistic model called **hidden Markov forest**, which extends our recent hidden Markov tree [16] from a single global tree (assuming spatial stationarity) to zonal trees (incorporating spatial non-stationarity). Figure 3 provides an illustrative example. The model contains a series of hidden Markov trees, each of which has an observed feature layer (white nodes) and a hidden class layer (grey nodes). The hidden class nodes in grey follow the flow tree dependency structure within each zone (in this example, the tree topology is reverse of Figure 1 to highlight water flowing from different pixels towards the same stream location or drainage point). The overall probabilistic distribution of nodes from all zones can be decomposed into zonal (intra-zone) distributions, i.e., $P(\mathbf{X}, \mathbf{Y}) = \prod_k P(\mathbf{X}_k, \mathbf{Y}_k) = \prod_k \prod_i P(\mathbf{x}_{k_i}|y_{k_i})P(y_{k_i}|y_{\mathcal{P}_{k_i}})$, where k is the zone index, k_i is the index of the i-th sample in zone k, $y_{\mathcal{P}_{k_i}}$ is the set of parent classes of node k_i in the tree. The model can be parameterized with a similar assumption as in [16], that is, $P(\mathbf{x}_{k_i}|y_{k_i})$ follows a Gaussian distribution, and $P(y_{k_i}|y_{\mathcal{P}_{k_i}})$ follows a class transitional matrix with a partial order constraint [16]. For instance, we can assume $P(y_{k_i} = 0|y_{\mathcal{P}_{k_i}} = 0) = 1$ and $P(y_{k_i} = 1|y_{\mathcal{P}_{k_i}} = 1) = \rho$. For leaf nodes without parents, we can assume $P(y_{k_i} = 1) = \pi$. Here ρ and π are parameters.

The above formulation can be extended to a discriminative (post-processing) version that can be named *conditional Markov forest*. In this case, we directly formulate the conditional probability of classes given features, $P(\mathbf{Y}|\mathbf{X}) = \prod_k P(\mathbf{Y}_k|\mathbf{X}_k)$, whereby $P(\mathbf{Y}_k|\mathbf{X}_k) = \prod_i P(y_{k_i}|y_{\mathcal{P}_{k_i}}, \mathbf{x}_{k_i})$. We can further express this as $\prod_i \frac{P(y_{k_i}|\mathbf{x}_{k_i})P(y_{k_i}|y_{\mathcal{P}_{k_i}})}{P(y_{k_i})}$ based on an approximation that $P(y_{\mathcal{P}_{k_i}}|\mathbf{x}_{k_i}) = P(y_{\mathcal{P}_{k_i}})$. This approximation is due to the fact that the explanatory features of the current lo-

cation $\mathbf{x}_{k_i}$ is one node away from parent classes $y_{\mathcal{P}_{k_i}}$ [20], as in Figure 3. In this version, the final class predictions are based on initial class probabilities from a base classifier $P(y_{k_i}|\mathbf{x}_{k_i})$ together with class transitional probability on the trees.
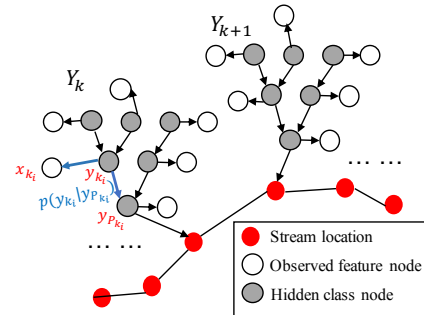

Figure 3: Our Hidden Markov forest model. (best viewed in color)

Inter-zone spatial regularization: Our probabilistic formulation of hidden Markov forest assumes statistical independence between zones for simplicity. In reality, spatial autocorrelation often exists. For flood mapping, the inferred class maps from nearby zones should be spatially smooth in their flood levels. We define the (*flood level*) in a zone as the average elevation of the inferred flood surface boundary. The flood boundary in a flow tree is the set of nodes whose inferred classes are flood but whose neighbors are dry. We denote flood level of zone k as ϕ_k, which is a function $\phi_k(\mathbf{Y}_k)$ on inferred class map $\mathbf{Y}_k$ in the zone. For example, in Figure 4, the flood levels of zone k and zone $k + 1$ are the vertical elevation of their frontier flood nodes (above their stream locations). Our new objective function is in Equation 3.1, where the first term is the log likelihood of node classes within individual zones and the second term is a spatial regularization to enforce spatial smoothness of inferred flood levels across nearby zones. Here λ is a weight with $\lambda = 0$ indicating completely independent zones and $\lambda = \infty$ enforcing strict smoothness.

(3.1)

$$Loss = -\sum_{k=1}^{K} \log p(\mathbf{Y}_k|\mathbf{X}_k) + \lambda \sum_{k=1}^{K-1} \|\phi_k(\mathbf{Y}_k) - \phi_{k+1}(\mathbf{Y}_{k+1})\|$$

3.3 Efficient class inference algorithms The goal of class inference is to infer the class labels for all locations (zonal tree nodes) such that the overall loss function in Equation 3.1 is minimized. There are several computational challenges. First, the number of nodes in each zonal tree can be very large (e.g., thousands). Second, the regularization term is based on inferred flood elevation level (a zonal function based on node classes), making it hard to apply eixsting efficient exact class in-
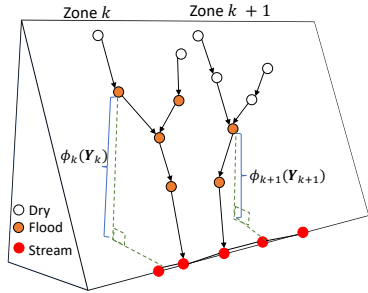
Figure 4: Illustration of zonal flood level Phi based on node classes

Figure 5: Incremental updates for the log likelihood of a rising flood frontier within a zonal tree

ference algorithms based on message propagation [16] (there is no closed-form solution). Third, the number of combinations of node class assignments for all zones is exponential to the number of zones. To address the challenges, we proposed several computational refinements to reduce the enumeration space, including using the partial order constraints on node classes within a zonal tree and using dynamic programming as well as a multi-resolution filter for multi-zone joint class inference.

Efficiently enumerate the candidate class assignments within a zonal tree. The goal here is to efficiently enumerate the log likelihood values for different candidate node class assignments within a zonal tree, i.e., $\log p(\mathbf{Y}_k|\mathbf{X}_k)$ for different $\mathbf{Y}_k$. We propose to leverage the order constraint on node classes within a tree, i.e., if any node is flooded (class 1), then all sub-tree nodes must be flooded as well (otherwise flood water will inundate there due to gravity). Based on this, we can enumerate all feasible class assignments within a zonal tree by raising the elevation level of the flood frontier and incrementally updating the log likelihood. This is shown in Figure 5, where y_c is the child class of node n. When increasing the flood level from node n to its child, only the probability terms related to node n, its parents, and its children are impacted, as expressed in Equation 3.2. The time complexity of enumerating the log likelihood values of all candidate class assignment is $O(N_k)$, where N_k is the number of tree nodes.

$$(3.2)$$
$$\Delta \leftarrow \log \left(P(y_{k_i}|\mathbf{x}_{\mathbf{k_i}})P(y_{k_i}|y_{\mathcal{P}_{k_i}})P(y_c|y_{k_i})/P(y_{k_i}) \right) \Big|_{y_{k_i}=0}^{y_{k_i}=1}$$

Efficiently search the optimal combination of class frontier levels across zones through dynamic programming. Because the regularization term is on the vertical elevation of zonal class frontiers $\phi_k(\mathbf{Y}_k)$, there is no closed-form solution. We have to
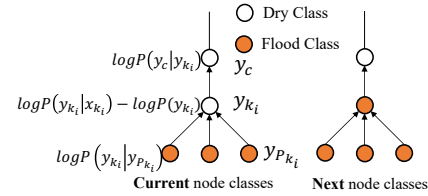
enumerate through all combinations of class frontiers from different zones, the number of which is exponential. We propose to use dynamic programming to reduce the exponential search space (similar to the Viterbi algorithm [21]). Figure 6(a) provides an example with three zonal trees in black (the internal node degree is one for simplicity) and each tree has four nodes (four candidate flood frontier levels, illustrated by node vertical positions). The edge weight $e_{i,j}^{(k)}$ is the loss component in Equation 3.1 related to zone k. Specifically, $e_{i,j}^{(k)} = -\log p(\mathbf{Y}_k|\mathbf{X}_k) + \lambda\|\phi_k(\mathbf{Y}_k) - \phi_{k+1}(\mathbf{Y}_{k+1})\|$ with $\mathbf{Y}_k$ and $\mathbf{Y}_{k+1}$ assigned to flood levels i and j respectively. For example, $e_{3,4}^{(1)}$ is calculated by assuming that zone 1 class assignment $\mathbf{Y}_1$ (the left-most tree) is with a frontier level at node 3 (the second topmost node) and that zone 2 class assignment $\mathbf{Y}_2$ is with a frontier level at node 4 (the topmost node). In this way, inferring the optimal class combinations from all zones becomes searching the path with the minimum total edge weights. This can be done by dynamic programming (i.e., the Viterbi algorithm). The time complexity of this process is $O(\sum_k N_k N_{k+1}) = O(KN^2)$, where N_k is the number of nodes in zone k, N is the largest size among all zones, and K is the total number of zones.
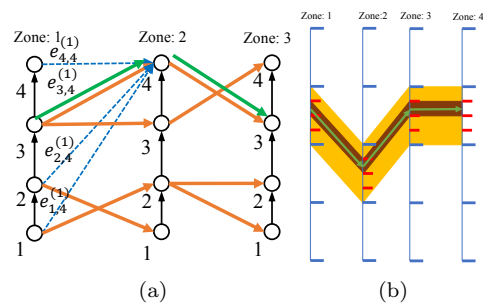

Figure 6: (a) dynamic programming for multi-zone joint learning (b) multi-resolution filtering to reduce enumeration space

Further reduce the search space across zones through multi-resolution filtering. Even though

the dynamic programming approach reduces the time complexity of a naive approach, the overall time complexity is still quadratic to the number of nodes in each zone. In practice, the number of nodes in a zone can reach millions, still making the computation prohibitively expensive. To further reduce this bottleneck, we propose to use **multi-resolution filtering**. The main idea is to divide the number of class frontier levels in each zone into m equal bins ($m \ll N$). Instead of enumerating through all frontier levels, we only enumerate through the bins (at a coarser resolution). This reduce the time complexity from $O(KN^2)$ to $O(Km^2)$. After we find the optimal path through dynamic program, we can further refine the bin at a finer resolution, e.g., by further dividing the optimal bin from each zone into m smaller bins (if possible) and search for the optimal path within this filtered space. The total number of iterations can be $O(\log_m N)$, where N is the number of nodes in a zone. So the total time complexity is $O(Km^2 \log_m N)$ within the optimal bins from each zone. Figure 6(b) provides an illustrative example with four zones and $m = 4$. The initial optimal path is shown in yellow, the further refined path is shown in brown, and the final optimal path is shown in green.

4 Evaluation

The goal is to compare our proposed method with baseline methods in classification performance on two real world datasets. We also evaluated the computational performance of our method with the proposed multi-resolution filter. Experiments were conducted on a Linux machine with an AMD 64-Core Processor 2GHz CPU and 128GB main memory. Below are the candidate methods in comparison. Unless specified otherwise, we used default parameters in open source tools for baseline methods.

- **Terrain-unaware i.i.d. classifiers**: We tested random forest (**RF**) in R randomForest package.

- **Terrain-unaware spatial classifier**: We used **UNet** [12] implemented in Tensorflow [22].

- **Terrain aware global model:** We use a global hidden Markov tree (**HMT**) as a post-processor from [17]. For two initial classifiers, we have random forest **HMT-RF** and UNet **HMT-UNET**.

- **Terrain aware zonal model:** Our proposed hidden Markov forest. We denote HMF without spatial regularization as **HMF** ($\lambda = 0$) and HMF with spatial regularization as **HMFSR** ($\lambda > 0$). With two base classifiers, we have **HMF-RF**, **HMF-UNET**, **HMFSR-RF** and **HMFSR-UNET**. For

Table 1: Classification on real data in Grifton, NC

Classifiers	Class	Precision	Recall	F1	Avg F1
RF	Dry	0.67	0.58	0.62	0.65
	Flood	0.63	0.72	0.67	
UNET	Dry	0.46	0.60	0.52	0.44
	Flood	0.44	0.31	0.36	
HMT-RF	Dry	0.39	0.13	0.19	0.40
	Flood	0.48	0.81	0.60	
HMT-UNET	Dry	0.68	0.92	0.78	0.74
	Flood	0.88	0.58	0.70	
HMF-RF	Dry	0.89	0.59	0.72	0.76
	Flood	0.69	0.93	0.79	
HMF-UNET	Dry	0.75	0.99	0.85	0.82
	Flood	0.99	0.66	0.80	
HMFSR-RF	Dry	0.99	0.99	0.99	0.99
	Flood	0.99	0.99	0.99	
HMFSR-UNET	Dry	0.98	0.99	0.99	0.99
	Flood	0.99	0.98	0.99	

HMT and HMF, we set $\rho = 0.999$, $\pi = 0.2$ for RF base and $\pi = 0.5$ for UNet base.

Dataset description: We used two flood mapping datasets from the cities of Grifton and Greenville in North Carolina during Hurricane Matthew in 2016. Explanatory features were red, green, blue bands in aerial imagery from NOAA National Geodetic Survey [23]. The digital elevation map was from the University of North Carolina Libraries [24]. All data were resampled into a 2 meter resolution. The test area size was 5722 by 7143 in Grifton and 9310 by 9519 in Greenville. We randomly drew 45000 test samples per class from manually annotated ground truth polygons in both regions. For random forest, we used 20000 training samples per class. Note that for the deep learning U-Net, we had to provide extra training set with 143 contiguous 224 by 224 patches for training and 28 patches for validation. We also added normalized elevation as an additional input channel for U-Net. For HMF, the specific zone partition details are in the supplementary materials.

4.1 Classification Performance Comparison
The overall classification performance of different methods on two datasets are summarized in Table 1 and Table 2 respectively. From Table 1, we can see that RF classifier achieved overall F1-score of 0.65. The deep learning UNet achieved overall F1-score of 0.44. The main reason for this poor performance is that the region contains a large area of tree canopies, which obscure the view of flood water beneath them (UNet cannot distinguish tree canopies in flood areas from those in dry areas and thus classify all of them into dry). Using global HMT as a post-processor degraded the RF's F1 to 0.40 but enhanced the U-Net from 0.44 to 0.74. In contrast, our proposed model HMF

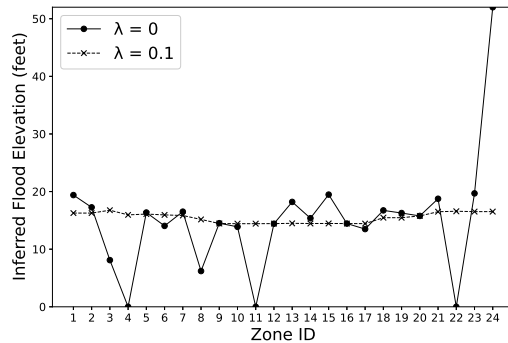Table 2: Classification on real data in Greenville, NC

Classifiers	Class	Precision	Recall	F1	Avg F1
RF	Dry	0.72	0.62	0.67	0.69
	Flood	0.67	0.76	0.71	
UNET	Dry	0.57	0.88	0.69	0.58
	Flood	0.73	0.34	0.46	
HMT-RF	Dry	0.55	0.16	0.25	0.45
	Flood	0.51	0.87	0.64	
HMT-UNET	Dry	0.66	0.92	0.77	0.71
	Flood	0.86	0.52	0.65	
HMF-RF	Dry	0.84	0.57	0.67	0.72
	Flood	0.67	0.89	0.77	
HMF-UNET	Dry	0.71	0.99	0.83	0.78
	Flood	0.99	0.59	0.74	
HMFSR-RF	Dry	0.98	0.99	0.99	0.99
	Flood	0.99	0.98	0.99	
HMFSR-UNET	Dry	0.99	0.99	0.99	0.99
	Flood	0.99	0.99	0.99	

(a) Zone wise inferred flood elevation levels in Grifton, NC

(b) Zone wise inferred flood elevation levels in Greenville, NC

Figure 7: Inferred flood surface elevations at individual zones by HMF-UNet and HMFSR-UNet

enhanced both RF and UNet even without spatial regularization, indicating that the zonal partition better reflects the spatial heterogeneity. Finally, our HMF with spatial regularization performed the best. We observed similar results on the Greenville dataset. The visualization of the predicted flood maps is in the supplementary materials.
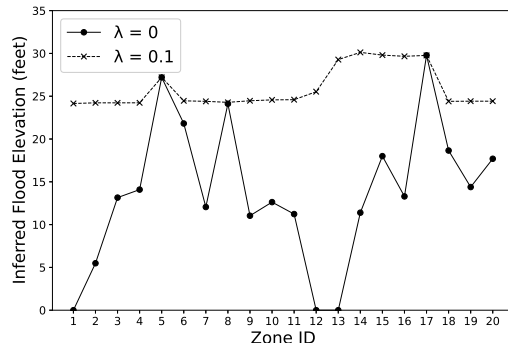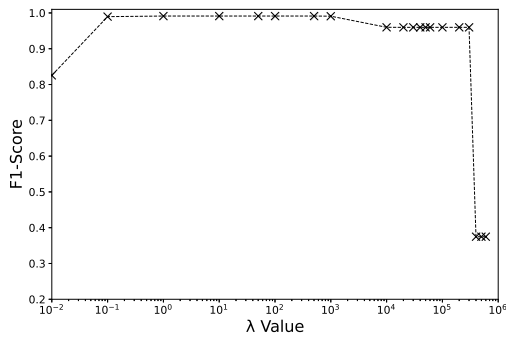
To see the effect of spatial regularization in multizone joint inference, we plotted the inferred flood surface elevation levels in individual zones in each test region (with and without regularization), as shown in Figure 7. The results show that without spatial regularization, HMF tends to infer different class surface levels across neighboring zones, making the results spurious (especially when we use the results to analyze flood depth). In contrast, the spatial regularization significantly reduced inconsistency of flood surface levels across zones. It's worth noting that the inferred flood surface elevation levels in different zones here are all normalized (i.e., substracted by the corresponding river node elevation to reflect the "thickness" of water along a river). Therefore, the relatively flat curve in this figure does not mean that we could easily use a global tree for the entire area.

Sensitivity of HMFSR to λ: We conducted sensitivity of our HMF model to different regularization weight λ from 10^{-2} to 10^6. The results on the Grifton dataset are shown in Figure 8(a). We see that when λ is very small (e.g., 10^{-2}), the F1 score is lower (due to little regularization effect). When λ is very large (close to 10^6), the results significantly degraded due to over smoothing. The results also show that our HMFSR model persistently performed well for a wide range of λ (e.g., from 0.1 to 1000, even near 10^5), making it easy to select a reasonable value.

Sensitivity of HMFSR to the number of

intervals in multi-resolution filtering: We varied the number of intervals k from 500 to 10000. The results of F1 score in Figure 8(b) show that the performance of HMF was stable. This is likely due to the fact that the loss function changes smoothly with a rising flood class frontier level. Thus, our multi-resolution filtering did not miss the optimal solution.
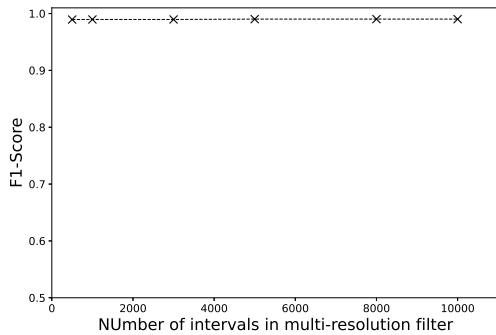
4.2 Computational Performance Comparison
We first evaluated the computational time costs of our HMF model. The time costs of two major components in HMF with multi-resolution filtering, i.e., incremental log likelihood calculation and joint class inference, are shown in Figure 9(a). We can see that the log likelihood calculation took the vast majority of the time costs in HMF. This is because the time complexity of log likelihood calculation is $O(N)$, while the time complexity of joint class inference has been reduced from $O(KN^2)$ to $O(Km^2 \log_m N)$ (thanks to multi-resolution filtering, see Section 3.3).

To evaluate the effect of multi-resolution filtering, we evaluated the time costs of HMF class inference with different number of intervals m from 500 to 30000, shown in Figure 9(b). We can see that as m in-

(a) Sensitivity to λ

(b) Sensitivity to the number of intervals

Figure 8: Sensitivity analysis of HMF on Grifton, NC

(a) Time cost decomposition.

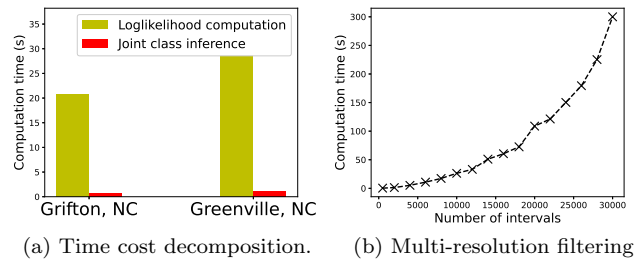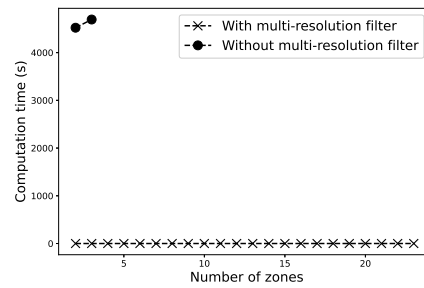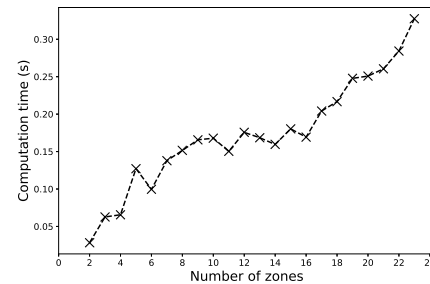(b) Multi-resolution filtering

Figure 9: The time cost analysis of HMF.

(a) Both with and without filtering

(b) With filtering in detail

Figure 10: Computation time costs of HMF with and without multi-resolution filtering.

creases, the time costs also increases. The trend is consistent with our theoretical time complexity analysis ($O(Km^2 \log_m N)$). We further compared the time costs of HMF class inference without multi-resolution filtering and that with multi-resolution filtering. Specifically, we increased the number of zones within the Grifton dataset from 2 to 24. Results in Figure 10(a) show that the time cost of class inference for HMF with filtering (crossing marks) is far lower than HMF without filtering (solid circles). In fact, without multi-resolution filtering, the time cost is so high that it took over one hour for two zones only. The time costs of HMF with filtering looks flat. To see the details, we plotted the HMF with filtering separately in Figure 10(b). We can see that as the number of zone increases, the time cost also increases (not exactly linearly since the zones have different sizes). The total time costs was below 1 second.

5 Conclusion and Future Work

This paper proposes a hidden Markov forest framework for spatial classification on 3D terrain data. The framework is motivated by a real-world application of observation-based flood inundation mapping based on Earth imagery. The framework addresses the limitations of existing methods by incorporating the spa-

tial non-stationarity of class dependency on terrains through a zonal tree structure and conducting effective and efficient class inference. Evaluations on real-world datasets show that our method outperforms multiple baselines and our proposed computational refinement significantly reduces the time cost.

In future work, we plan to conduct more experimental evaluations on multiple base classifiers (e.g., other image segmentation models). We also plan to explore other potential applications of the proposed framework, such as potential energy landscape in biochemistry.

Acknowledgement

References

[1] Y Wang, JD Colby, and KA Mulcahy. An efficient method for mapping flood extent in a coastal floodplain using landsat tm and dem data. *International Journal of Remote Sensing*, 23(18):3681–3696, 2002.

[2] Wen Yang, Hui Song, Xiaojing Huang, Xin Xu, and Mingsheng Liao. Change detection in high-resolution sar images based on jensen–shannon divergence and hierarchical markov model. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(8):3318–3327, 2014.

[3] Jessica V Fayne, John D Bolten, Colin S Doyle, Sven Fuhrmann, Matthew T Rice, Paul R Houser, and Venkat Lakshmi. Flood mapping in the lower mekong river basin using daily modis observations. *International journal of remote sensing*, 38(6):1737–1757, 2017.

[4] National Oceanic and Atmospheric Administration. National Water Model: Improving NOAA's Water Prediction Services. `http://water.noaa.gov/documents/wrn-national-water-model.pdf`, 2018.

[5] Don Cline. Integrated water resources science and services: an integrated and adaptive roadmap for operational implementation. Technical report, National Oceanic and Atmospheric Administration, 2009.

[6] Zhe Jiang. A survey on spatial prediction methods. *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[7] Quanlong Feng, Jiantao Liu, and Jianhua Gong. Urban flood mapping based on unmanned aerial vehicle remote sensing and random forest classifier—a case of yuyao, china. *Water*, 7(4):1437–1455, 2015.

[8] Luc Anselin. *Spatial econometrics: methods and models*, volume 4. Springer Science & Business Media, 2013.

[9] Raymond H Chan, Chung-Wa Ho, and Mila Nikolova. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *Image Processing, IEEE Transactions on*, 14(10):1479–1485, 2005.

[10] Anne Puissant, Jacky Hirsch, and Christiane Weber. The utility of texture analysis to improve per-pixel classification for high to very high spatial resolution imagery. *International Journal of Remote Sensing*, 26(4):733–745, 2005.

[11] Yuliya Tarabalka, Jón Atli Benediktsson, and Jocelyn Chanussot. Spectral–spatial classification of hyperspectral imagery based on partitional clustering techniques. *IEEE Transactions on Geoscience and Remote Sensing*, 47(8):2973–2987, 2009.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[13] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[14] Liangpei Zhang, Lefei Zhang, and Bo Du. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 4(2):22–40, 2016.

[15] Jiongqian Liang, Peter Jacobs, Jiankai Sun, and Srinivasan Parthasarathy. Semi-supervised embedding in attributed networks with outliers. In *Proceedings of the 2018 SIAM international conference on data mining*, pages 153–161. SIAM, 2018.

[16] Miao Xie, Zhe Jiang, and Arpan Man Sainju. Geographical hidden markov tree for flood extent mapping. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 2545–2554. ACM, 2018.

[17] Zhe Jiang and Arpan Man Sainju. Hidden markov contour tree: A spatial structured model for hydrological applications. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 804–813. ACM, 2019.

[18] Kornelijus Survila, Ting Li, Yan Y Liu, David G Tarboton, and Shaowen Wang. A scalable high-performance topographic flow direction algorithm for hydrological information analysis. In *Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale*, pages 1–7, 2016.

[19] David Tarboton, Yan Liu, Nazmus Shams Sazib, and Shaowen Wang. Accelerating taudem for extracting hydrology information from national-scale high resolution topographic dataset. In *Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale*, pages 1–2, 2016.

[20] Arpan Man Sainju, Wenchong He, and Zhe Jiang. A hidden markov contour tree model for spatial structured prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[21] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[22] ZFTurbo. ZF_UNET_224 Pretrained Model. `https://github.com/ZFTurbo/ZF_UNET_224_Pretrained_Model`, 2018.

[23] National Oceanic and Atmospheric Administration. Data and imagery from noaa's national geodetic survey. `https://www.ngs.noaa.gov`.

[24] NCSU Libraries. LIDAR Based Elevation Data for North Carolina. `https://www.lib.ncsu.edu/gis/elevation`, 2018.