

Semi-supervised Learning with the EM Algorithm: A Comparative Study between Unstructured and Structured Prediction

Wenchong He, Zhe Jiang, *Member, IEEE*,

Abstract—Semi-supervised learning aims to learn prediction models from both labeled and unlabeled samples. There has been extensive research in this area. Among existing work, generative mixture models with Expectation-Maximization (EM) is a popular method due to clear statistical properties. However, existing literature on EM-based semi-supervised learning largely focuses on unstructured prediction, assuming that samples are independent and identically distributed. Studies on EM-based semi-supervised approach in structured prediction is limited. This paper aims to fill the gap through a comparative study between unstructured and structured methods in EM-based semi-supervised learning. Specifically, we compare their theoretical properties and find that both methods can be considered as a generalization of self-training with soft class assignment of unlabeled samples, but the structured method additionally considers structural constraint in soft class assignment. We conducted a case study on real-world flood mapping datasets to compare the two methods. Results show that structured EM is more robust to class confusion caused by noise and obstacles in features in the context of the flood mapping application.

Index Terms—Semi-supervised learning; Expectation-Maximization (EM); Structured prediction

1 INTRODUCTION

Semi-supervised learning aims to learn prediction models based on both labeled and unlabeled samples. It is important when training data contains limited labeled samples but abundant unlabeled samples. In real world spatial prediction problems, input data often contains abundant explanatory features but very limited ground truth. For example, in earth image classification for land cover mapping, a large number of image pixels with spectral features are collected from remote sensing platforms, but only a limited number of pixels are labeled with ground truth land cover classes [1], [2]. The reason is that collecting ground truth is both expensive and time consuming by field crew or well-trained image interpreters.

The topic of semi-supervised learning has been extensively studied in the literature. According to a survey [3], techniques can be categorized into generative models with EM method, graph-based methods, label propagation, self-training and co-training, etc. In the EM-based method [4], unknown labels are considered as hidden variables, and both labeled and unlabeled samples are used to estimate parameters of a generative model of the joint distribution. Graph-based methods assume that samples connected by heavy edges tend to have the same label and thus aim to obtain a smooth label assignment over the graph [5]. In label propagation, the main idea is to propagate the labels of nodes throughout the network and form class communities [6]. Semi-supervised learning in structured output space focuses on learning dependency or some internal relationships between classes [7]. In self-training, a classifier

is first trained with a small amount of labeled samples and then is used to classify unlabeled samples [8]. The most confident predictions on unlabeled samples together with their predicted class labels are added to the training set. For co-training, it assumes that features can be split into two sets and each set is sufficient to train a good classifier separately [9]. Each classifier then classifies the unlabeled data and teaches the other classifier with the most confident predictions on unlabeled samples [3].

Among all the methods, semi-supervised learning with EM provides clear statistical properties. There are two types of EM algorithms: unstructured EM and structured EM. Unstructured EM assumes that data samples follow an identical and independent distribution and feature variables are independent [4]. It considers unknown class labels as hidden variables in generative mixture models (e.g., Gaussian mixture models). It then uses EM algorithm to learn model parameters and infer hidden classes at the same time. Structured EM assumes that either input samples or feature variables within samples follow structural dependency. Existing works on structured EM can be further categorized into two types: Bayesian structural EM and Hidden Markov Model (HMM). Bayesian structural EM uses a directed acyclic graph (DAG) to represent the conditional dependency between feature variables within a sample. It uses EM to learn the DAG structure from samples whose feature variables are partially unobserved (treated as hidden variables) [10]. HMM uses a chain [11] or tree [12] to represent the conditional dependency between samples. It uses EM to learn the model parameters and infer hidden classes at the same time. Structured EM is of particular interest because data in many real-world applications (e.g., earth science, biology, and material science) often show structural constraints. However, there is limited study that compares

- Corresponding author: Zhe Jiang, zjiang@cs.ua.edu
- W. He and Z. Jiang were with the Department of Computer Science, University of Alabama, Tuscaloosa, AL, 35487.

Manuscript received April 19, 2005; revised August 26, 2015.

structured and unstructured methods in EM-based semi-supervised learning. It remains unclear about their strengths and weaknesses.

To fill the gap, this paper provides a comparative study between structured methods and unstructured methods in EM-based semi-supervised learning for spatial classification problems [13]. For unstructured methods, we use Gaussian mixture models as an example. For structured methods, we use a recent model called *geographical hidden Markov tree (HMT)* [12] as a representative example. We compare the theoretical properties and conducted detailed case studies on real-world datasets. In summary, this paper makes the following contributions:

- We compared the theoretical properties of a representative unstructured methods (Gaussian Mixture Model) and structured methods (HMT) in EM-based semi-supervised learning.
- Through theoretical analysis, we found out that both EM-based methods can be considered as a generalization of self-training with soft class assignment of unlabeled samples. The difference is that structured methods additionally consider structure constraints in soft class assignment.
- We also empirically compared the performance of a representative unstructured EM (Gaussian mixture model) and structured EM (hidden Markov tree) in a case study on real-world flood mapping datasets. Results showed that in this particular application, unstructured EM without spatial feature could be impacted by feature noise and obstacles. Adding spatial feature into unstructured EM could alleviate but not fully resolve the issue. In contrast, structured EM could resolve the issue due to explicit structural constraint on this particular type of applications

2 PROBLEM FORMULATION

Suppose we have data set \mathcal{D} of N data samples, which is composed of an unlabeled subset \mathcal{D}_u and a labeled subset \mathcal{D}_l ($\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$). \mathcal{D}_u contains N_u data samples of inputs without class labels, $\mathcal{D}_u = \{(\mathbf{x}_n)\}_{n=1}^{N_u}$. The labeled subset \mathcal{D}_l contains N_l data samples of input-output pairs, $\mathcal{D}_l = \{(\mathbf{x}_n, y_n)\}_{n=N_u+1}^N$ where $\mathbf{x}_n \in \mathbb{R}^{m \times 1}$ is a vector of m explanatory features (including both non-spatial features and spatial contextual feature) with each element corresponding to one feature layer, and $y_n \in \{0, 1\}$ is a binary class label.

Explanatory features of labeled samples and unlabeled samples are noted as $\mathbf{X}_u = [\mathbf{x}_1, \dots, \mathbf{x}_{N_u}]^T$ and $\mathbf{X}_l = [\mathbf{x}_{N_u+1}, \dots, \mathbf{x}_N]^T$ respectively. The class layers are noted as $\mathbf{Y}_u = [y_1, \dots, y_{N_u}]^T$ and $\mathbf{Y}_l = [y_{N_u+1}, \dots, y_N]^T$. \mathbf{Y}_u is unknown and to be predicted. For example, in flood mapping from earth imagery, \mathbf{x}_n can be the spectral band values (and elevation) of each pixel, and y_n is the class label (flood, dry). We also assume $N_u \gg N_l$. Labeled subset \mathcal{D}_l and an unlabeled subset \mathcal{D}_u are training and test samples respectively. Note that both \mathbf{X}_l and \mathbf{X}_u are used in training (semi-supervised learning).

3 EM ALGORITHMS FOR SEMI-SUPERVISED LEARNING

This section compares theoretical properties of EM-based semi-supervised learning algorithms in unstructured and structured prediction respectively.

3.1 Example of Unstructured EM: Gaussian Mixture Models

EM-based semi-supervised learning algorithm in unstructured prediction does not consider structural dependency of sample classes and assumes that the sample classes are independent and identically distributed (i.i.d.).

The joint distribution of all samples is as below:

$$\begin{aligned} P(\mathbf{X}, \mathbf{Y}) &= P(\mathbf{X}_u, \mathbf{Y}_u, \mathbf{X}_l, \mathbf{Y}_l) \\ &= \prod_{n=1}^{N_u} P(\mathbf{x}_n, y_n) \prod_{n=N_u+1}^N P(\mathbf{x}_n, y_n) \end{aligned} \quad (1)$$

We assume that the sample features follow an i.i.d. Gaussian distribution in each class. The prior class probability $P(y_n = c)$ follows a Bernoulli distribution below.

$$P(y_n = c) = \pi_c, c = \{0, 1\}. \quad (2)$$

Feature distribution $P(\mathbf{x}_n | y_n)$ is as below,

$$P(\mathbf{x}_n | y_n = c) \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \quad (3)$$

We denote the entire set of parameters as $\boldsymbol{\Theta} = \{\pi_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c | c = 0, 1\}$. The labeled data subset \mathcal{D}_l is used to initialize parameters $\boldsymbol{\Theta}$. Then EM algorithm creates a function for the posterior expectation of log-likelihood evaluated using current estimate for the parameters (E-Step) and next updates parameters maximizing the expected log-likelihood for the next iteration (M-Step). The unknown class \mathbf{Y}_u is hidden variable. The posterior expectation of log likelihood of all samples is as below:

$$\begin{aligned} &\mathbb{E}_{\mathbf{Y}_u | \mathbf{X}, \mathbf{Y}_l, \boldsymbol{\Theta}_0} \log P(\mathbf{X}_u, \mathbf{Y}_u, \mathbf{X}_l, \mathbf{Y}_l | \boldsymbol{\Theta}) \\ &= \mathbb{E}_{\mathbf{Y}_u | \mathbf{X}, \mathbf{Y}_l, \boldsymbol{\Theta}_0} \log \left(\prod_{n=1}^{N_u} P(\mathbf{x}_n, y_n | \boldsymbol{\Theta}) \prod_{n=N_u+1}^N P(\mathbf{x}_n, y_n | \boldsymbol{\Theta}) \right) \\ &= \sum_{\mathbf{Y}_u} P(\mathbf{Y}_u | \mathbf{X}, \mathbf{Y}_l, \boldsymbol{\Theta}_0) \left(\sum_{n=1}^{N_u} \log P(\mathbf{x}_n, y_n | \boldsymbol{\Theta}) + \sum_{n=N_u+1}^N \log P(\mathbf{x}_n, y_n | \boldsymbol{\Theta}) \right) \\ &= \sum_{n=1}^{N_u} \sum_{y_n} P(y_n | \mathbf{x}_n, \boldsymbol{\Theta}_0) \log P(\mathbf{x}_n, y_n | \boldsymbol{\Theta}) \\ &\quad + \sum_{n=N_u+1}^N \sum_{y_n} P(y_n | \mathbf{x}_n, \mathbf{Y}_l, \boldsymbol{\Theta}_0) \log P(\mathbf{x}_n, y_n | \boldsymbol{\Theta}) \\ &= \sum_{n=1}^{N_u} \sum_{y_n} P(y_n | \mathbf{x}_n, \boldsymbol{\Theta}_0) \log P(\mathbf{x}_n, y_n | \boldsymbol{\Theta}) \\ &\quad + \sum_{n=N_u+1}^N \sum_c \mathbf{I}(y_n = c) \log P(\mathbf{x}_n, y_n = c | \boldsymbol{\Theta}) \end{aligned} \quad (4)$$

The posterior class distribution for unlabeled samples ($1 \leq n \leq N_u$) $P(y_n | \mathbf{x}_n, \boldsymbol{\Theta}_0)$ is

$$P(y_n = c | \mathbf{x}_n, \boldsymbol{\Theta}_0) = \frac{P(\mathbf{x}_n | y_n = c, \boldsymbol{\Theta}_0) P(y_n = c)}{\sum_{c'=0}^1 P(\mathbf{x}_n | y_n = c', \boldsymbol{\Theta}_0) P(y_n = c')}, c = 0, 1 \quad (5)$$

Taking the above into the posterior expectation of log likelihood, we can easily get the following formulas to update parameters that maximize the posterior expectation (M-Step). Note that $\mathbf{I}(y_n = c) = 1$ if $y_n = c$, and 0 otherwise.

$$\pi_c = \frac{\sum_{n=1}^{N_u} P(y_n = c | \mathbf{x}_n, \Theta_0) + \sum_{n=N_u+1}^N \mathbf{I}(y_n = c)}{\sum_{c'=0}^1 \sum_{n=1}^{N_u} P(y_n = c' | \mathbf{x}_n, \Theta_0) + \sum_{c'=0}^1 \sum_{n=N_u+1}^N \mathbf{I}(y_n = c')} \quad (6)$$

$$\mu_c = \frac{\sum_{n=1}^{N_u} P(y_n = c | \mathbf{x}_n, \Theta_0) \mathbf{x}_n + \sum_{n=N_u+1}^N \mathbf{I}(y_n = c) \mathbf{x}_n}{\sum_{n=1}^{N_u} P(y_n = c | \mathbf{x}_n, \Theta_0) + \sum_{n=N_u+1}^N \mathbf{I}(y_n = c)} \quad (7)$$

$$\Sigma_c = \frac{\sum_{n=1}^{N_u} P(y_n = c | \mathbf{x}_n, \Theta_0) (\mathbf{x}_n - \mu_c)(\mathbf{x}_n - \mu_c)^T + \sum_{n=N_u+1}^N \mathbf{I}(y_n = c) (\mathbf{x}_n - \mu_c)(\mathbf{x}_n - \mu_c)^T}{\sum_{n=1}^{N_u} P(y_n = c | \mathbf{x}_n, \Theta_0) + \sum_{n=N_u+1}^N \mathbf{I}(y_n = c)} \quad (8)$$

Class inference: After learning model parameters, we can infer hidden class variables by maximizing the log joint probability of unlabeled data.

$$\begin{aligned} \log P(\mathbf{X}_u, \mathbf{Y}_u) &= \log \prod_{n=1}^{N_u} P(\mathbf{x}_n | y_n) P(y_n) \\ &= \sum_{n=1}^{N_u} \log P(\mathbf{x}_n | y_n) P(y_n) \end{aligned} \quad (9)$$

To maximize the total log probability, we can maximize each term in Equation 9. For each sample n , we simply choose the class c that gives higher probability of $P(\mathbf{x}_n | y_n = c) P(y_n = c)$.

3.2 Example of Structured EM: Hidden Markov Tree

EM-based semi-supervised learning algorithm in structured prediction assumes a dependency structure between sample classes. We use a spatial classification model called *geographical hidden Markov tree (HMT)* [12]. It is a probabilistic graphical model that generalizes the common hidden Markov model (HMM) from a one-dimensional sequence to a partial order tree.

The joint distribution of unlabeled samples' features and classes can be formulated as Equation 10, where \mathcal{P}_n is the set of parent samples of the n th sample in the dependency tree ($\mathcal{P}_n = \emptyset$ for a leaf node), and $y_{k \in \mathcal{P}_n} \equiv \{y_k | k \in \mathcal{P}_n\}$ is the set of parent node classes of node n .

$$P(\mathbf{X}_u, \mathbf{Y}_u) = \prod_{n=1}^{N_u} P(\mathbf{x}_n | y_n) \prod_{n=1}^{N_u} P(y_n | y_{k \in \mathcal{P}_n}) \quad (10)$$

Similar to EM for unstructured prediction, the model assumes that features in each class follows an i.i.d. Gaussian distribution.

$$P(\mathbf{x}_n | y_n = c) \sim \mathcal{N}(\mu_c, \Sigma_c) \quad (11)$$

Class transitional probability follows the partial order flow dependency constraint [12], as shown in Table 1.

TABLE 1: Class transition probability and prior probability

$P(y_n y_{\mathcal{P}_n})$	$y_{\mathcal{P}_n} = 0$	$y_{\mathcal{P}_n} = 1$		$P(y_n)$
$y_n = 0$	1	$1 - \rho$	$y_n = 0$	π_0
$y_n = 1$	0	ρ	$y_n = 1$	π_1

We denote the entire set of parameters as $\Theta = \{\rho, \pi_c, \mu_c, \Sigma_c | c = 0, 1\}$. The posterior expectation of log likelihood of unlabeled samples (E-step) is as below:

$$\begin{aligned} &\mathbb{E}_{\mathbf{Y}_u | \mathbf{X}_u, \Theta_0} \log P(\mathbf{X}_u, \mathbf{Y}_u | \Theta) \\ &= \mathbb{E}_{\mathbf{Y}_u | \mathbf{X}_u, \Theta_0} \log \left\{ \prod_{n=1}^{N_u} P(\mathbf{x}_n | y_n, \Theta) \prod_{n=1}^{N_u} P(y_n | y_{k \in \mathcal{P}_n}, \Theta) \right\} \\ &= \sum_{\mathbf{Y}_u} P(\mathbf{Y}_u | \mathbf{X}_u, \Theta_0) \\ &\quad \left\{ \sum_{n=1}^{N_u} \log P(\mathbf{x}_n | y_n, \Theta) + \sum_{n=1}^{N_u} \log P(y_n | y_{k \in \mathcal{P}_n}, \Theta) \right\} \\ &= \sum_{n=1}^{N_u} \sum_{y_n} P(y_n | \mathbf{X}, \Theta_0) \log P(\mathbf{x}_n | y_n, \Theta) \\ &\quad + \sum_{n=1}^{N_u} \sum_{y_n, y_{k \in \mathcal{P}_n}} P(y_n, y_{k \in \mathcal{P}_n} | \mathbf{X}, \Theta_0) \log P(y_n | y_{k \in \mathcal{P}_n}, \Theta) \end{aligned} \quad (12)$$

After computation of marginal class posterior distribution through forward and backward message propagation, we can get the parameter updating formula by maximizing the posterior expectation of log likelihood as below (M-Step).

$$\rho = \frac{\sum_{n | \mathcal{P}_n \neq \emptyset} \sum_{y_n} \sum_{y_{\mathcal{P}_n}} y_{\mathcal{P}_n} (1 - y_n) P(y_n, y_{\mathcal{P}_n} | \mathbf{X}, \Theta_0)}{\sum_{n | \mathcal{P}_n \neq \emptyset} \sum_{y_n} \sum_{y_{\mathcal{P}_n}} y_{\mathcal{P}_n} P(y_n, y_{\mathcal{P}_n} | \mathbf{X}, \Theta_0)} \quad (13)$$

$$\pi_1 = \frac{\sum_{n | \mathcal{P}_n = \emptyset} y_n P(y_n | \mathbf{X}, \Theta_0)}{\sum_{n | \mathcal{P}_n = \emptyset} \sum_{y_n} P(y_n | \mathbf{X}, \Theta_0)} \quad (14)$$

$$\mu_c = \frac{\sum_n \mathbf{x}_n P(y_n = c | \mathbf{X}, \Theta_0)}{\sum_n P(y_n = c | \mathbf{X}, \Theta_0)}, c = 0, 1 \quad (15)$$

$$\Sigma_c = \frac{\sum_n (\mathbf{x}_n - \mu_c)(\mathbf{x}_n - \mu_c)^T P(y_n = c | \mathbf{X}, \Theta_0)}{\sum_n P(y_n = c | \mathbf{X}, \Theta_0)}, c = 0, 1 \quad (16)$$

Class inference: After learning model parameters, we can infer hidden class variables by maximizing the overall joint probability.

$$\log P(\mathbf{X}, \mathbf{Y}) = \sum_{n=1}^{N_u} \log P(\mathbf{x}_n | y_n) + \sum_{n=1}^{N_u} \log P(y_n | y_{k \in \mathcal{P}_n}) \quad (17)$$

A naive approach that enumerate all combinations of class assignment is infeasible due to the exponential cost. We use a dynamic programming based method called *max-sum* [14].

TABLE 2: Comparison between unstructured EM (i.i.d.) and structured EM

	Gaussian mixture model	Hidden Markov tree
Parameter	π_c, μ_c, Σ_c	$\rho, \pi_c, \mu_c, \Sigma_c$
Posterior probability	$p(y_n x_n)$, from Bayes theorem	$p(y_n \mathbf{X})$, from message propagation.
Feature probability	$p(x_n y_n) \sim \mathcal{N}(\mu_c, \Sigma_c)$	$p(x_n y_n) \sim \mathcal{N}(\mu_c, \Sigma_c)$
Class probability	$p(y_n = c) = \pi_c$	$p(y_n = c) = \pi_c$ for leaf nodes, $p(y_n y_{k \in \mathcal{P}_n})$ for non-leaf nodes

3.3 Comparison

Table 2 summarizes the comparison of theoretical properties between the two EM methods. From the expression of joint probabilities, both methods assume that features in each class follows an i.i.d. Gaussian distribution. The difference lies in the class prior probability. For unstructured EM methods, the class prior probability follows i.i.d. Bernoulli distribution $p(y_n) \sim \mathbf{B}(1, \pi)$ while for the structured EM method, sample classes follow a dependency structure, which can be expressed by class transition probability $P(y_n|y_{k \in \mathcal{P}_n})$ [12]. Moreover, both methods have similar formulas for parameter update where sample mean and covariance are reweighted by each sample’s class posterior probability. The difference lies in the way they compute the class posterior probability. For unstructured model, it is from Bayes theorem; while for structured model it is from message propagation, considering the class dependency structure [12], [15], [16], [17].

When analysing the theoretical properties of the two EM algorithms, we found that EM algorithms can be considered as a generalization of self-training [6]. In self-training, a classifier is first trained with a small amount of labeled data. The classifier is then used to classify the unlabeled data. The most confident unlabeled points together with their predicted labels are added to the training set in the next iteration [3]. In contrast, EM algorithm first uses labeled samples to initialize model parameters, and then estimates the class posterior probability of each unlabeled sample. In the next iteration, it uses the class posterior probability as a weight to re-estimate model parameter (Equations 13, 14, 15, and 16). In summary, both self-training and EM algorithm use the labeled data to iteratively learn model parameters. The difference is that self-training makes a hard class assignment of unlabeled samples to retrain model in iterations while the EM algorithm uses the class posterior probability to make a soft class assignment for each unlabeled sample.

It is important to note that the local optimal problem exists for both approaches. If class estimation of unlabeled data is misleading, it may further hurt learning in iterations. This potential problem can be alleviated by a good initial parameters estimation from representative training samples. The labeled samples are used to initialize some model parameters, i.e., the mean vectors μ_c and covariance matrices Σ_c of features of samples in each class. These parameters could have been initialized randomly without labeled samples (unsupervised setting of Gaussian mixture model, also called EM clustering, as well as unsupervised

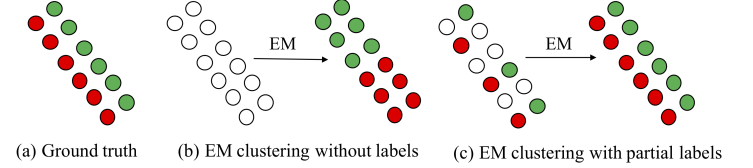


Fig. 1: Illustration of importance of labeled samples in Gaussian mixture models (red and green are two classes)

setting of hidden Markov models). However, randomly initialized parameters of mean and covariance matrix may not converge to good values after EM iterations, particularly when the feature clusters of samples in two classes are not well-separated from each other (as the case of our flood mapping application), as shown by the example in Figure 1(b). In this situation, labeled samples in each class can be used to estimate its corresponding mean vector and covariance matrix more accurately, as shown by Figure 1(c).

TABLE 3: Summary of unstructured EM and structured EM in unsupervised learning and semi-supervised learning.

Example	Unstructured EM	Structured EM
	<ul style="list-style-type: none">• Samples are identical and independent.• Feature variables are independent.• Unknown classes as hidden variables	<ul style="list-style-type: none">• Either samples or feature variables follow structural dependency.• Unknown classes and missing features are hidden variables.
Unsupervised Learning <ul style="list-style-type: none">• Randomly initialize model parameters.• Update model parameters without labels.	Generative mixture models without labels (EM clustering) [11].	Bayesian structural model [8].
Semi-supervised Learning <ul style="list-style-type: none">• Initialize model parameters with labeled samples.• Update model parameter with or without labeled samples.	Generative mixture models with partial labels [1].	Hidden Markov model (HMM) with extra labeled samples [10].

Table 3 compares unstructured EM versus structured EM in both unsupervised and semi-supervised settings in a broader perspective. Both unstructured and structured EM learn model parameters with the existence of missing variables (e.g., hidden class labels or missing feature observations). The difference is that structured EM incorporates structural dependency between samples or feature variables within a sample. In unsupervised setting, class labels are either fully unknown (e.g., EM clustering) or not of relevance (e.g., feature dependency learning in Bayesian structured EM). Thus, initialization and update of model parameters do not rely on class labels. In semi-supervised setting, class labels are partially available. These labels can be used in parameter initialization and potentially in parameter update as well. Specifically, in Gaussian mixture models with partial labels, both labeled and unlabeled samples are used in parameter initialization as well as class inference. In hidden Markov models, if extra labeled samples are available, these labels can be used to provide a reasonable initialization of some model parameters (e.g., mean and covariance matrix in each class). After this, model parameters are iteratively updated based on the features and dependency structure on test samples. The process belongs to transductive learning since a model is learned for a specific structure across test samples. It is worth to note that HMMs can also be

unsupervised with randomly initialized parameters without class labels. But the converged parameters in this case may be ineffective in discriminating two classes on test samples (see [12]).

4 EVALUATION

In this section, we compared the performance of unstructured EM prediction with structured EM prediction through case studies on real world datasets. Our goal is to get insights on how well different EM methods can handle class confusion due to noise and obstacles in features. Moreover, we also compare EM algorithm with other baseline methods. We chose Gradient Boost Machine and Random forest models as baseline because these models have well-tested source codes and have also shown superior performance over other models in the literature. The candidate classification methods are listed below.

- **Unstructured EM (Gaussian mixture model) w/o elevation feature:** We implemented our codes in Matlab.
- **Unstructured EM (Gaussian mixture model) w/ elevation feature :** We implemented our codes in Matlab.
- **Structured EM (HMT):** We implemented the HMT source code in C++.
- **Gradient Boost Model (GBM):** We used the GBM in R gbm packages on raw features together with elevation feature.
- **Random forest (RF) :** We used the random forest in R randomForest packages on raw features together with elevation feature.

Hyperparameter: For unstructured EM, the hyperparameter includes the parameter convergence threshold, and the cutoff threshold to decide the positive and negative classes. For structured EM, the hyper-parameter includes the parameter convergence threshold, and initial parameter values of ρ (class transitional probability) and π (class prior probability). The parameter convergence threshold was set as $1e-5$, and the cutoff threshold was set as 0.5. We set $\rho = 0.99$ and $\pi = 0.5$ based on earlier sensitivity study (see [12]). For random forest, the hyper-parameters include the number of trees N_1 , the number of variables randomly sampled as candidates at each split N_2 and the minimum size of terminal nodes N_3 . For N_1 , we tried values of 300, 350, 400, 450, 500. For N_2 , we tried values of 1, 2, 3. For N_3 , we tried values of 5, 10, 20, 40. When choosing the optimal value for one parameter, we kept other parameters constant. The optimal values are: $N_1 = 350$, $N_2 = 2$ and $N_3 = 10$. For gradient boosted model, the hyper-parameters include the number of trees T_1 , the maximum depth of each tree T_2 , and the shrinkage parameter T_3 that is used to reduce the impact of each additional fitted tree. For T_1 , we tried values of 1000, 1500, 2000, 2500, 3000. For T_2 , we tried values of 1, 2, 3, 4. For T_3 , we tried values of 0.1, 0.01, 0.001. The optimal hyper-parameters are: $T_1 = 2000$, $T_2 = 1$, and $T_3 = 0.01$.

Dataset description: We used two flood mapping datasets from Hurricane Harvey floods in Texas 2017 and Hurricane Mathew in North Carolina 2016 respectively. Non-spatial explanatory features include red, green, blue bands in aerial

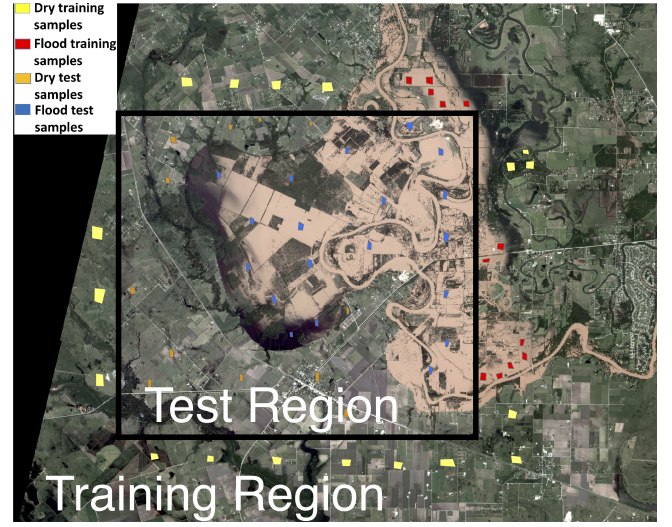


Fig. 2: Training and test polygons

imagery from PlanetLab Inc. and NOAA National Geodetic Survey [18] respectively. The spatial contextual feature was digital elevation map from Texas natural resource management department and the University of North Carolina Libraries [19] respectively. All data were resampled into 2 meter by 2 meter resolution. Figure 3 shows the entire input features in the Harvey dataset, including non-spatial features (RGB bands) in Figure 3(a) and spatial contextual feature (elevation) in Figure 3(b). From the images, we can see class confusion due to noise and obstacles in non-spatial features (there are pixels with tree colors in both flood and dry areas). Due to space limit, we put the results on the second dataset in Appendix.

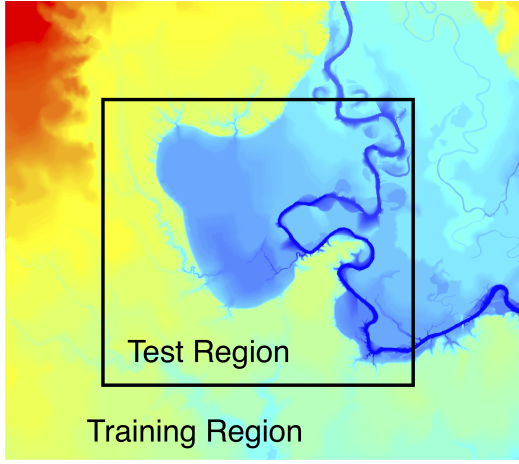
Training and test dataset split: We used simple validation. The separation of training and test sets is shown in Figure 2. We had a test region (highlighted by the black rectangle) with labeled polygons in both classes. The training region with training polygons in both classes was outside the test region. In the experiment, we randomly selected 10000 pixels from training polygons (5000 in flood and 5000 in dry) and 103374 pixels from test polygons (43972 in dry and 59402 in flood)

4.1 Parameter Convergence

Parameter iteration and convergence: Our convergence threshold was set to 0.001%. Figure 4 shows the iterations of μ_c, Σ_c ($c = 0$ for the dry class, $c = 1$ for the flood class) in unstructured EM without spatial contextual feature (elevation). For Σ_c , we only plotted the diagonal elements (i.e., the variance of each feature) and omitted covariance values due to space limit. From the results, we can see that the converged mean values of the two classes are well separated with μ_0 converged to 3500 to 5500 and μ_1 converged to 6300 to 6500. This is consistent with Figure 3(a) since flood areas have lighter color than dry areas. Note that the range of values in red, blue, and green bands are bigger than 256 due to a different imagery data type. We can also observe that the variance of the flood class Σ_1 converged to a lower range of values (around $1 \times 10^5 \sim 3 \times 10^5$). The reason is that unstructured EM without spatial contextual feature will regroup samples with class confusion in their feature values



(a) High-resolution satellite imagery in NC.



(b) Digital elevation

Fig. 3: RGB feature and spatial elevation feature

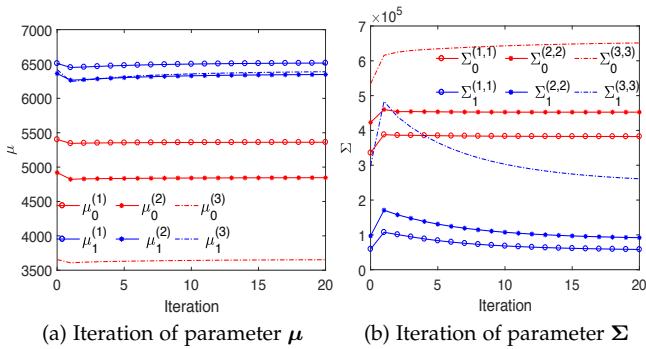


Fig. 4: Parameter iterations and convergence for unstructured EM without elevation feature

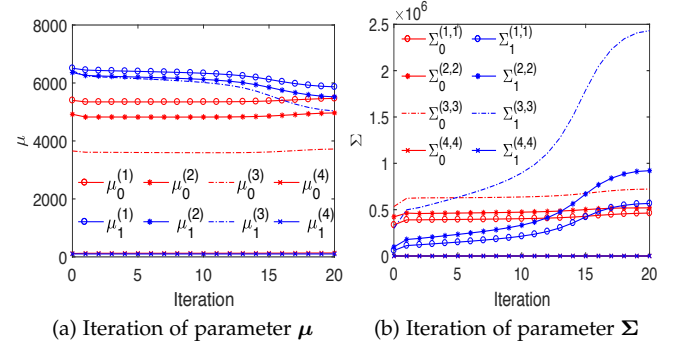


Fig. 5: Parameter iterations and convergence for unstructured EM with elevation feature

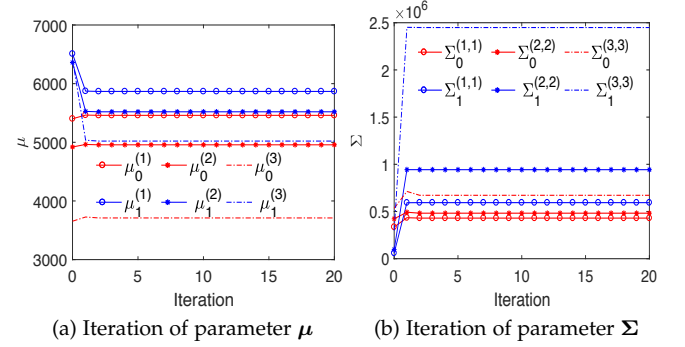


Fig. 6: Parameter iterations and convergence for structured EM algorithm

(e.g., tree pixels with the same color in both flood and dry areas) into the same class based on class posterior (Figure 7), which in turn will influence the update of parameters in each class.

For unstructured EM with spatial contextual feature (elevation), the parameter iteration and convergence are shown in Figure 5. Note that there is one more dimension for elevation in the plots. We can see that the converged mean values of two classes are less separately compared with Figure 4. μ_0 converges to a range of 3800 to 5500 and μ_1 converges to a range of 5000 to 6000. Another dramatical change is on the variance of flood class Σ_1 , which increases to a much larger range of values (0.5×10^6 to 2.5×10^6). This can be explained by the marginal class posterior probabilities in Figure 8, where samples with high posterior probability in the flood class (yellow color) are growing and include more tree pixels in the flood areas. Because of this, the variance of the flood class grows bigger and the mean of the flood class drops (tree pixels in water are darker than exposed flood water).

For structured EM, the converged parameter values are moderate in the middle of the above two cases, but more similar to unstructured EM with spatial contextual feature (elevation). The results can be explained by the posterior class probability in Figure 9, where samples with high posterior probability in the flood class (yellowish pixels) are moderately in between the previous two maps. The main difference is that μ_c and Σ_c converge in fewer (only 2) iterations compared with unstructured EM with elevation.

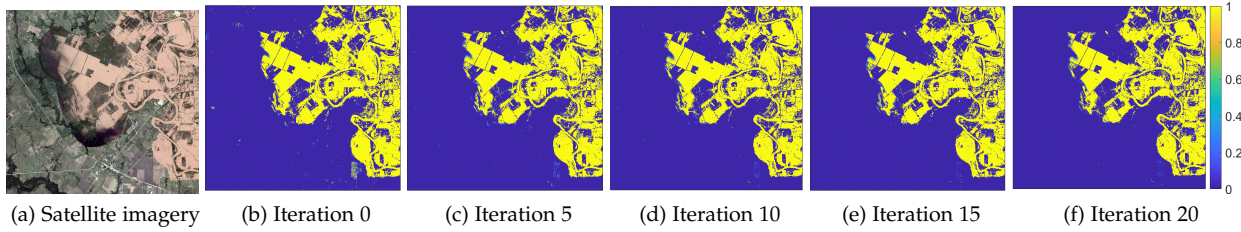


Fig. 7: Posterior Probability of Unstructured EM without elevation

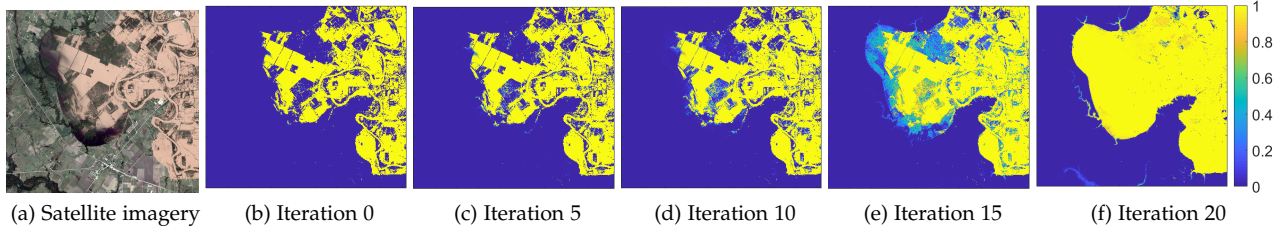


Fig. 8: Posterior Probability of Unstructured EM with elevation

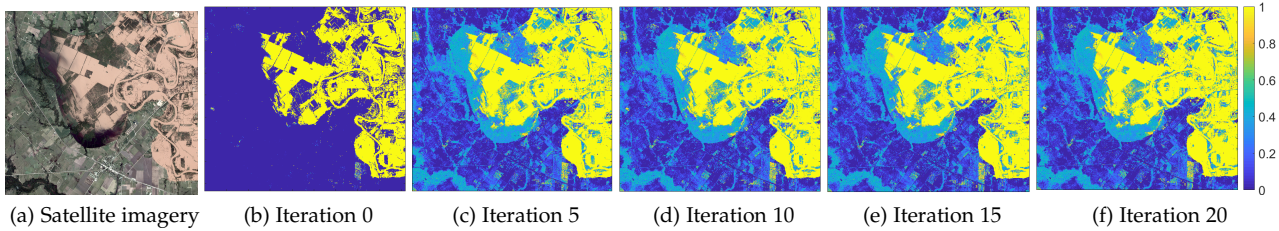


Fig. 9: Posterior Probability of structured EM

4.2 Posterior Class Probability

Figures 7, 8, and 9 show the iterations of posterior probabilities of samples in the flood class for unstructured EM without and with elevation features as well as structured EM respectively. Sample class posterior probability is important in understanding how the EM algorithm works because it shows how much a sample contribute to the parameter update for the next iteration in each class (e.g., Equations 6, 7, 8, and Equations 13, 14, 15, 16). For unstructured EM, posterior class probability is estimated based on the Bayes Theorem with an i.i.d. assumption. Thus, samples with class confusion in feature values will be estimated towards the same class. This explains why pixels correspond to trees in the flood have low posterior probability in the flood class in Figure 7 (same as the trees in dry areas). In contrast, unstructured EM with the elevation feature could separate confused tree pixels in flood and dry areas since their elevation values differ. This is shown in Figure 8(f). Finally, for structured EM, posterior class probability is estimated based on both local class likelihood from non-spatial features (RGB colors) and class dependency structure. This explains why the class posterior in Figure 9 is moderate compared with Figure 7 and Figure 8. It is worth noting that posterior class probability for individual pixels are not the same as final class prediction in structured EM (final class prediction could be smoother due to jointly predicting all classes with dependency).

4.3 Prediction Result

Table 4 and Figure 11 show the final classification results of the three methods. We can see that unstructured EM

TABLE 4: Comparison on Harvey, Texas flood data

Classifiers	Class	Prec.	Recall	F	Avg. F
GBM	Dry	0.88	0.98	0.93	0.87
	Flood	0.96	0.70	0.81	
RF	Dry	0.70	0.99	0.82	0.82
	Flood	0.99	0.69	0.81	
Unstructured EM w/o elev.	Dry	0.68	0.99	0.81	0.75
	Flood	0.99	0.53	0.70	
Unstructured EM w/ elev.	Dry	0.99	0.99	0.99	0.99
	Flood	0.99	0.99	0.99	
EM Structured	Dry	0.99	0.99	0.99	0.99
	Flood	0.99	0.99	0.99	

without spatial contextual feature (elevation) performed poorly with class confusion. Unstructured EM with spatial contextual feature performs significantly better with less class confusion, but also produces some salt-and-pepper noise errors since spatial features are used with an i.i.d. assumption. In contrast, the structured EM method can both address the class confusion issue and show a smooth class map due to explicitly considering spatial dependency structure.

Moreover, we plot the ROC curve and calculate the AUC of ROC curve for unstructured EM, GBM and RF classifier. As shown in Figure 10 and Table 5, the unstructured EM with elevation. gives the best ROC curve with AUC of 0.996, while the unstructured EM without elevation. shows the less significant result with AUC of 0.786. It means the spatial features play an important role in the unstructured EM classifier. The two baseline methods random forest and GBM show better results than unstructured EM without elevation classifier.

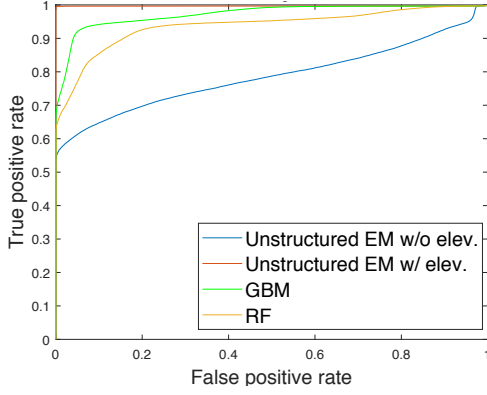


Fig. 10: ROC curve on Harvey, Texas data

TABLE 5: Comparison on AUC of ROC curve

Classifiers	AUC
GBM	0.975
RF	0.946
Unstructured EM w/o elev.	0.786
Unstructured EM w/ elev.	0.996

4.4 Salt-and-pepper Noise

TABLE 6: Comparison on the total number of salt-and-pepper noise (The total number of pixels is 19,167,008)

Classifiers	Number of Salt-and-pepper noise
GBM	62,846
RF	69,354
Unstructured EM w/o elev.	77,625
Unstructured EM w/ elev.	23,803
EM Structured	3,740

We used a spatial autocorrelation statistic called Gamma index [20] to quantify the salt-and-pepper noise level. Gamma index measures the similarity between the attribute

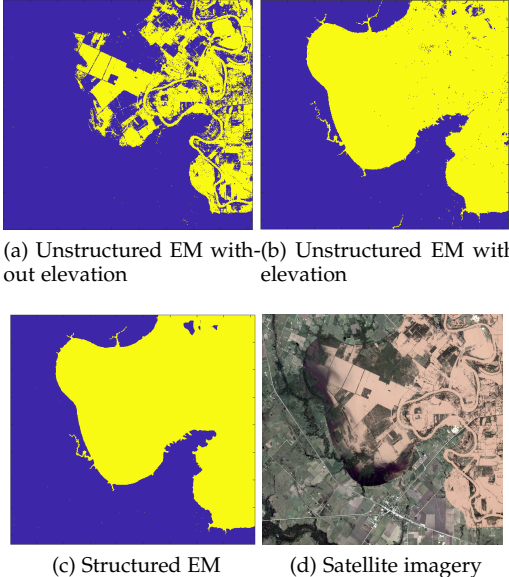


Fig. 11: Comparison of class prediction

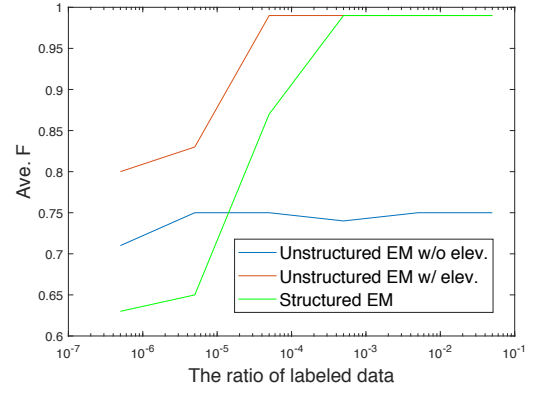


Fig. 12: Sensitivity to the ratio of labeled samples

values of a location and those of its neighbors. It is defined as

$$\Gamma_i = \frac{\sum_j W_{i,j} I_i I_j}{\sum_j W_{i,j}} \quad (18)$$

where i and j are locations, $W_{i,j}$ is 1 if j is neighbor, and 0 otherwise, I_i is 1 if pixel i is flood pixel and -1 otherwise. We define a salt-and-pepper noise pixel as a pixel that has a negative local Gamma index ($\Gamma_i < 0$). We calculated the total number of salt-and-pepper noise pixels across all locations in the predicted class map. The total numbers of salt-and-pepper noise pixels of the five classifiers are summarized in Table 6. We can see that RF, GBM, and EM i.i.d. without elevation have the highest salt-and-pepper noise level (above 60,000). EM i.i.d. with elevation is much better with only 23,803 salt-and-pepper noise. The EM structured method has the lowest number of salt-and-pepper noise, i.e., 3,740, about one order of magnitude lower than other classifiers.

4.5 Effect of the ratio of labels

We analyzed the sensitivity of three candidate EM methods to the ratio of training labels. Specifically, we increased the ratio of labeled samples from 5×10^{-7} to 5×10^{-2} . The results were summarized in Figure 12. We can see that as the ratio of labeled samples increase, the F-scores of all three methods first improved and then converged to an optimal value. Specifically, EM i.i.d. (represented by Gaussian mixture model) achieved the lowest peak F-score (around 0.75), while EM i.i.d. with elevation feature and EM structured model (represented by HMT) have a much better peak F-score (around 0.99). We also observe that when the ratio of labeled sample is small (e.g., below 10^{-5}), EM structured model has a poor performance (with an F-score below 0.65). This is probably because more labeled samples are needed to initialize good parameters for our representative EM structured model (HMT).

5 CONCLUSIONS AND FUTURE WORK

This paper makes a comparative study between unstructured and structured EM in semi-supervised learning. We compare the two methods in their theoretical properties and find that EM-based semi-supervised learning can be considered as a generalization of self-training method with

soft class assignment on unlabeled samples. A case study on flood mapping datasets shows that unstructured EM method can be significantly impacted by noise and obstacles in sample features. Adding spatial contextual features in unstructured EM method can reduce the impact of noise and obstacles but will still produce salt-and-pepper noise errors. Finally, structured EM can better address the issue compared with the other methods in this flood mapping applications. In future work, we plan to conduct the comparison studies on more types of datasets and applications to see if the conclusion can hold in general.

ACKNOWLEDGEMENT

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

REFERENCES

- [1] Z. Jiang and S. Shekhar, "Spatial big data science," *Schweiz: Springer International Publishing AG*, 2017.
- [2] S. Shekhar, Z. Jiang, R. Y. Ali, E. Eftelioglu, X. Tang, V. Gunturi, and X. Zhou, "Spatiotemporal data mining: a computational perspective," *ISPRS International Journal of Geo-Information*, vol. 4, no. 4, pp. 2306–2338, 2015.
- [3] X. Zhu, "Semi-supervised learning literature survey," 2005.
- [4] Z. Ghahramani and M. Jordan, "Learning from incomplete data (tech. rep. no. aim-1509)," 1994.
- [5] G. Camps-Valls, T. V. B. Marsheva, and D. Zhou, "Semi-supervised graph-based hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 10, pp. 3044–3054, 2007.
- [6] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," 2002.
- [7] U. Brefeld and T. Scheffer, "Semi-supervised learning for structured output variables," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 145–152.
- [8] I. Dópido, J. Li, P. R. Marpu, A. Plaza, J. M. B. Dias, and J. A. Benediktsson, "Semisupervised self-learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 7, pp. 4032–4044, 2013.
- [9] Y. Hong and W. Zhu, "Spatial co-training for semi-supervised image classification," *Pattern Recognition Letters*, vol. 63, pp. 59–65, 2015.
- [10] N. Friedman, "The bayesian structural em algorithm," *arXiv preprint arXiv:1301.7373*, 2013.
- [11] C. Damian, Z. Eksi, and R. Frey, "Em algorithm for markov chains observed via gaussian noise and point process information: Theory and case studies," *Statistics & Risk Modeling*, vol. 35, no. 1-2, pp. 51–72, 2018.
- [12] M. Xie, Z. Jiang, and A. M. Sainju, "Geographical hidden markov tree for flood extent mapping," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '18. ACM, 2018, pp. 2545–2554.
- [13] Z. Jiang, "A survey on spatial prediction methods," *IEEE Transactions on Knowledge and Data Engineering*, 2018.
- [14] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [15] Z. Jiang, M. Xie, and A. M. Sainju, "Geographical hidden markov tree," *IEEE Transactions on Knowledge and Data Engineering*, 2019.
- [16] Z. Jiang and A. M. 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*, 2019, pp. 804–813.
- [17] A. M. Sainju, W. He, and Z. Jiang, "A hidden markov contour tree model for spatial structured prediction," *IEEE Transactions on Knowledge and Data Engineering*, 2020.
- [18] National Oceanic and Atmospheric Administration, "Data and imagery from noaa's national geodetic survey," <https://www.ngs.noaa.gov>.
- [19] NCSU Libraries, "LIDAR Based Elevation Data for North Carolina," <https://www.lib.ncsu.edu/gis/elevation>, 2018.
- [20] L. Anselin, "Local indicators of spatial association," *Geographical analysis*, vol. 27, no. 2, pp. 93–115, 1995.



Wenchong He is a Ph.D. student in the department of Computer Science at the University of Alabama. He received his B.S. degree from University of Science and Technology of China (USTC) in 2017, and Master degree from the College of William Mary in 2019. His research interests include machine learning, data mining and deep learning.



Zhe Jiang is an assistant professor in the department of Computer Science at the University of Alabama. He received his Ph.D. from the University of Minnesota, Twin Cities in 2016, and B.E. from the the University of Science and Technology of China in 2010. His research interests include spatial and spatio-temporal data mining, spatial database, and geographic information system. He is a member of IEEE.

APPENDIX

This appendix shows empirical evaluation results on the Hurricane Mathew flood in North Carolina. Figure 13, 14, and 15 show the parameter iterations of model parameters in unstructured EM without elevation, unstructured EM with elevation feature, and structured EM respectively. The overall trends look similar to the results on the Harvey dataset. The corresponding class posterior probabilities of pixels in three methods are shown in Figure 17, 18, and 19. Comparison on class predictions are shown in Table 7 and Figure 16.

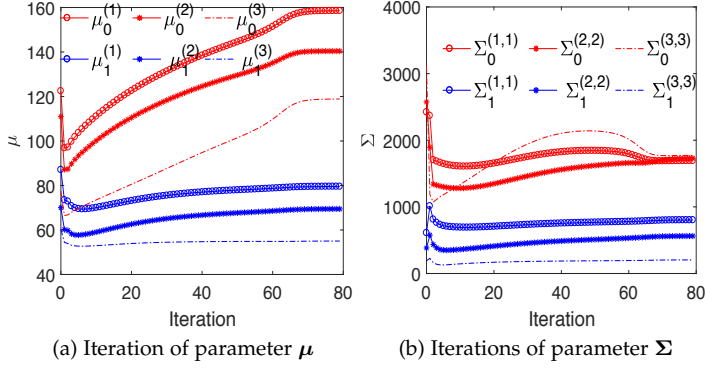


Fig. 13: Parameter iterations and convergence for unstructured EM without elevation feature

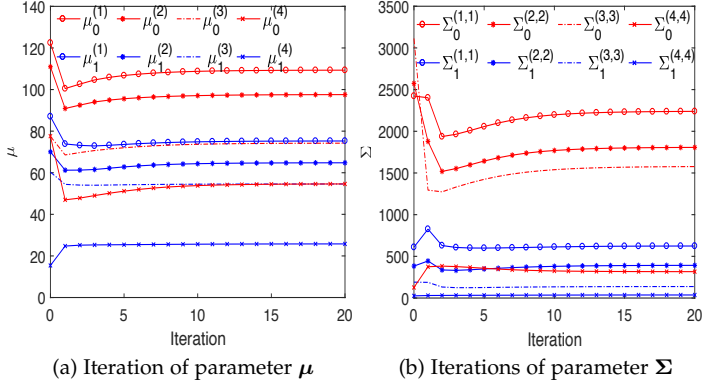


Fig. 14: Parameter iterations and convergence for unstructured EM with elevation feature

TABLE 7: Comparison on Mathew, Kinston flood data

Classifiers	Class	Prec.	Recall	F	Avg. F
EM i.i.d.(Converged)	Dry	0.91	0.18	0.30	0.51
	Flood	0.55	0.98	0.70	
EM i.i.d.(20 iterations)	Dry	0.83	0.51	0.63	0.70
	Flood	0.65	0.89	0.76	
EM i.i.d.+ elev.	Dry	0.96	0.81	0.88	0.89
	Flood	0.84	0.96	0.90	
EM-Structure	Dry	0.94	0.99	0.96	0.96
	Flood	0.99	0.94	0.96	

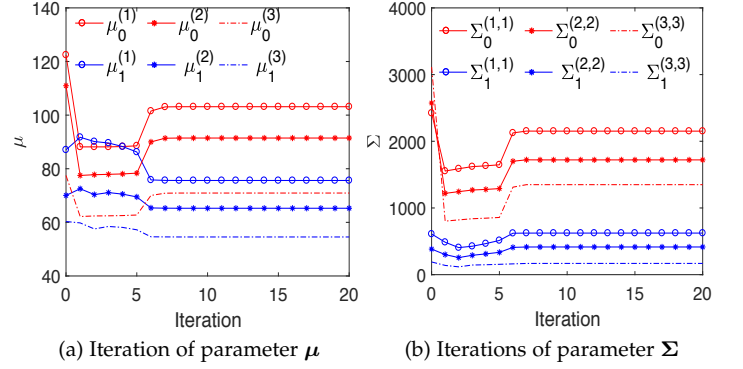


Fig. 15: Parameter iterations and convergence for structured EM algorithm

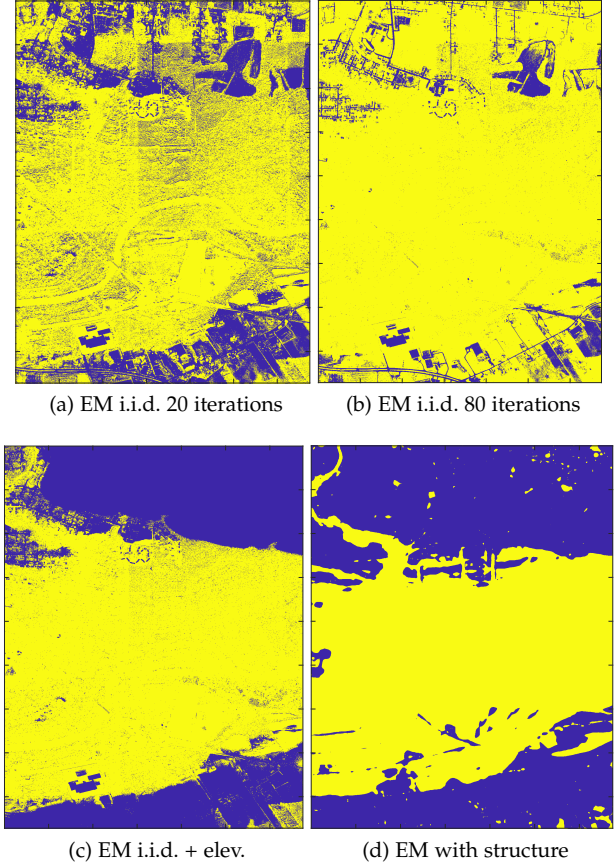


Fig. 16: Prediction result

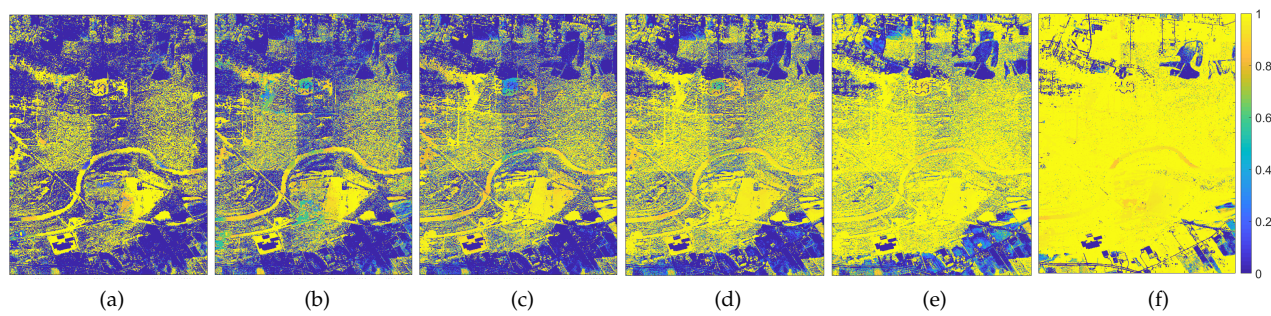


Fig. 17: Posterior Probability of Unstructured EM without elevation

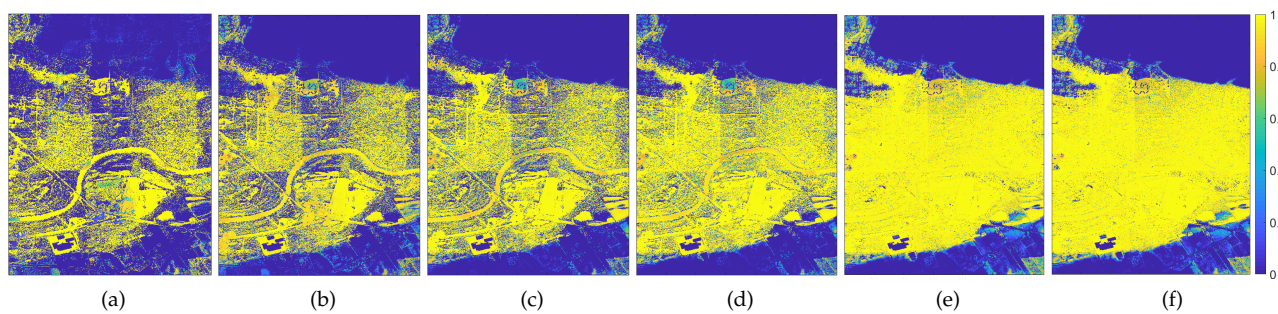


Fig. 18: Posterior Probability of Unstructured EM with elevation

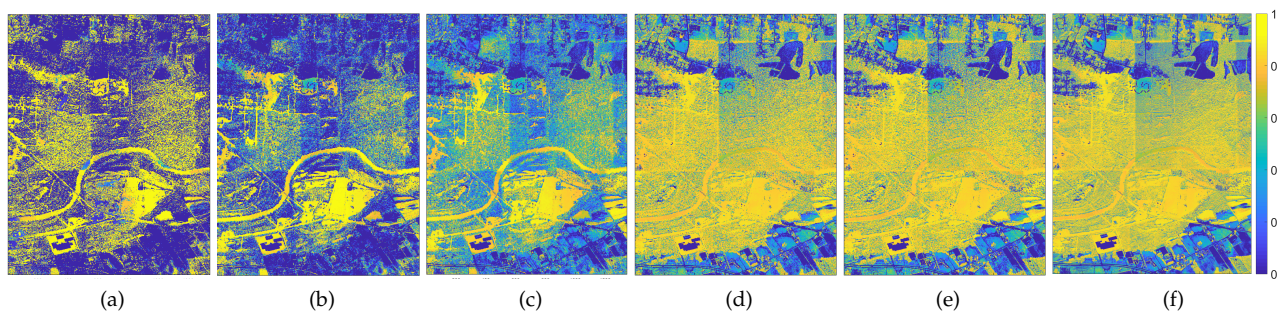


Fig. 19: Posterior Probability of structured EM