Spatial Ensemble Learning for Heterogeneous Geographic Data with Class Ambiguity: A Summary of Results

Zhe Jiang
Department of Computer Science
University of Alabama
zjiang@cs.ua.edu

Yan Li, Shashi Shekhar
Department of Computer Science
University of Minnesota
{lixx4266,shekhar}@umn.edu

Lian Rampi, Joseph Knight
Department of Forest Resources
University of Minnesota
{ortiz073,jknight}@umn.edu

ABSTRACT

Class ambiguity refers to the phenomenon whereby samples with similar features belong to different classes at different locations. Given heterogeneous geographic data with class ambiguity, the spatial ensemble learning (SEL) problem aims to find a decomposition of the geographic area into disjoint zones such that class ambiguity is minimized and a local classifier can be learned in each zone. SEL problem is important for applications such as land cover mapping from heterogeneous earth observation data with spectral confusion. However, the problem is challenging due to its high computational cost (finding an optimal zone partition is NP-hard). Related work in ensemble learning either assumes an identical sample distribution (e.g., bagging, boosting, random forest) or decomposes multi-modular input data in the feature vector space (e.g., mixture of experts, multimodal ensemble), and thus cannot effectively minimize class ambiguity. In contrast, our spatial ensemble framework explicitly partitions input data in geographic space. Our approach first preprocesses data into homogeneous spatial patches and uses a greedy heuristic to allocate pairs of patches with high class ambiguity into different zones. Both theoretical analysis and experimental evaluations on two real world wetland mapping datasets show the feasibility of the proposed approach.

CCS CONCEPTS

• **Information systems → Geographic information systems; Data mining;** • **Computing methodologies → Ensemble methods;**

KEYWORDS

Spatial classification, class ambiguity, spatial heterogeneity, spatial ensemble, local models

ACM Reference format:
Zhe Jiang, Yan Li, Shashi Shekhar, and Lian Rampi, Joseph Knight. 2017. Spatial Ensemble Learning for Heterogeneous Geographic Data with Class Ambiguity: A Summary of Results. In *Proceedings of SIGSPATIAL'17, Los Angeles Area, CA, USA, November 7–10, 2017,* 10 pages.
https://doi.org/10.1145/3139958.3140044

1 INTRODUCTION

Classifying heterogeneous geographic data with class ambiguity, i.e., same feature values corresponding to different classes in different locations, is a fundamental challenge in machine learning [13, 14]. This kind of effect is also called "ecological fallacy" [25]. Figure 1 shows an example in a wetland mapping application. The goal is to classify remote sensing image pixels (Figure 1(a)) into wetland and dry land classes (Figure 1(b)). The two circled areas contain pixels that share very similar spectral values yet belong to two different classes (also called spectral confusion). As a result, decision tree and random forest classifiers learned from the entire image makes tremendous prediction errors as shown in Figure 1(c-d). The goal of spatial ensemble learning is to decompose the geographic area into zones so as to minimize class ambiguity and to learn a local model in each zone.

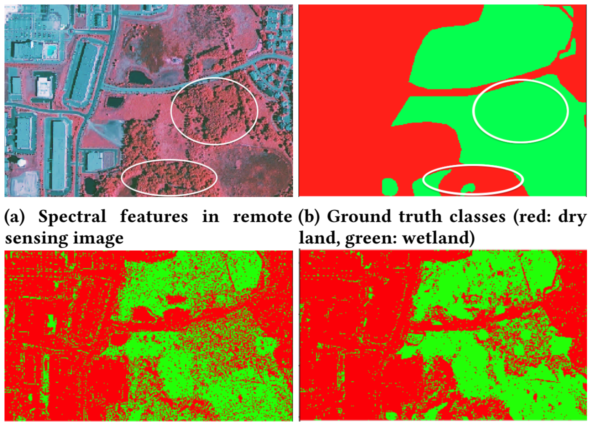

(a) Spectral features in remote sensing image (b) Ground truth classes (red: dry land, green: wetland)
(c) Decision tree predictions (d) Random forest predictions

Figure 1: Real world example of heterogeneous geographic data: class ambiguity exists in two white circles

Motivations: Spatial ensemble learning can be used in many applications where geographic data is heterogeneous with class ambiguity. For example, in remote sensing image classification, spectral confusion is a challenging issue [16, 21]. The issue is particularly important in countries where the type of auxiliary data that could reduce spectral confusion, such as elevation data, or imagery of high temporal and spatial resolution, is not available. In hydrologic scaling applications, the relationships between watershed characteristics and hydrologic responses is often spatially heterogeneous, driven by different physical control variables. In economic study, it may happen that old house age indicates high price in rural areas but low price in urban areas [8]. Thus, *age* can be an effective coefficient to classify house price in individual zones but ineffective in a global model. In cultural study, touching somebody during conversation is welcomed in France and Italy, but considered offensive in Britain unless in a sport field; the "V-Sign" gesture can mean "two" in America, "victory" in German, but "up yours" in Britain [24]. In these cases, spatial ensemble learning can provide a tool that captures heterogeneous relationships between factors (e.g., house age, gestures) and target phenomena (e.g., house price, culture meanings).

Challenges: The SEL problem is computationally challenging. First, there are a large number of spatial samples (pixels) to partition. Second, the objective measure of class ambiguity is non-distributive, i.e., the degree of class ambiguity in a zone cannot be easily computed from the degrees of class ambiguity in its subzones. Finally, given a geographic data, the number of candidate partitions is exponential to the number of spatial samples. It can be proved that finding an optimal zone partition is NP-hard.

Related work: Spatial ensemble learning belongs to a general category of ensemble learning problems [4, 29, 34], in which a number of weak models are combined to boost prediction accuracy. Conventional ensemble methods, including bagging [2], boosting [9], and random forest [3], assume an identical distribution of samples. Thus they cannot address heterogeneous geographic data with class ambiguity. Decomposition based ensemble methods (also called divide-and-conquer), including mixture of experts [15, 33] and multimodal ensemble [22], go beyond the identical and independent distribution assumption in that these methods can partition multi-modular input data and learn models in local partitions. Partitioning is usually conducted in feature vector space via a gating network, which can be learned simultaneously by an EM algorithm, or modeled by radius basis functions [32] or multiple local ellipsoids [27]. However, partitioning input data in feature vector space cannot effectively separate samples with class ambiguity because such samples are very "close" in non-spatial feature attributes. Other methods such as adding spatial coordinates into feature vectors can be ineffective since it creates geographic partitions whose zonal footprints are hard to interpret and can be too rigid to separate ambiguous zones with arbitrary shapes. There are other techniques for spatially heterogeneous data. A geographically weighted model [8] uses spatial kernel weighting functions to learn local models. However, it requires to learn a local model at every location, which is computationally very expensive, and it cannot allow arbitrary shapes of spatial zones for local models. Gaussian process [20] and multi-task learning [10] can also be used for heterogeneous geographic data, but they do not particularly focus on the class ambiguity issue. The mixture-of-experts approach has been used for *scene classification* on images via sub-blocks partitioning and learning local experts. But that problem is to classify an entire image (not individual pixels) [30].

Our contributions: To address limitations of related work, we formulate a spatial ensemble learning framework, which explicitly partitions input data in geographic space. Our approach first preprocesses data into homogeneous patches and then uses a greedy heuristic to group patches into contiguous zones while minimizing class ambiguity. A local model is learned from each zone to make predictions on samples in the same zone. We make the following contributions: (1) we formulate a novel spatial ensemble learning problem to classify heterogeneous geographic data with class ambiguity; (2) we propose effective and efficient algorithms, including constraint-based hierarchical clustering for homogeneous patch generation, as well as a bisecting algorithm to group patches into contiguous zones via greedy heuristics; (3) we provide theoretical analysis on the proposed algorithms; (4) we conduct experimental evaluations on the classification and computational performance of proposed approach on real world wetland mapping datasets.

Scope: This paper focuses on the class ambiguity issue in heterogeneous geographic data. Other recent advances that do not address class ambiguity, such as spatial-spectral classifiers [6], object-based image analysis [5, 23], and deep learning, fall outside the scope.

Outline: The paper is organized as follows. Section 2 defines basic concepts and formalizes the spatial ensemble learning problem. Section 3 introduces our approach. Experimental evaluations are in Section 4. Section 5 discusses some other relevant works. Section 6 concludes the paper with future work.

2 PROBLEM STATEMENT

2.1 Basic Concepts

Geographic raster framework: A geographic raster framework $\mathbf{F}$ is a tessellation of a 2-D plane into a regular grid. Each grid cell (or pixel) is a *spatial data sample*, defined as $s_i = (x_i, l_i, y_i)$, $1 \le i \le |\mathbf{F}|$, where x_i is a non-spatial feature vector, l_i is a 2-dimensional vector of spatial coordinates, and $y_i \in \{c_1, c_2, ..., c_p\}$ is a class label among p categories. All the samples in $\mathbf{F}$ can be divided into two disjoint subsets, a *labeled sample set* $\mathbf{L} = \{s_i = (x_i, l_i, y_i) \in \mathbf{F} | y_i \text{ is known}\}$ and *unlabeled sample set* $\mathbf{U} = \{s_i = (x_i, l_i, y_i) \in \mathbf{F} | y_i \text{ is unknown}\}$. In the example of Figure 2(a), $\mathbf{F}$ has 64 samples, including 14 labeled samples (colored in "training labels") and 50 unlabeled samples. Each sample has a 1-dimensional feature x and a class label (*red* or *green*).

Geospatial neighborhood relationship: It is a boolean function on two samples $\mathcal{R}(s_i, s_j)$, whose value is *true* if and only if s_i and s_j are spatially adjacent (i.e., two cells share a boundary).

Patch: A patch $\mathbf{P}$ is a spatially contiguous subset of samples, formally, $\mathbf{P} \subseteq \mathbf{F}$ such that for any two samples $s_i, s_j \in \mathbf{P}$, either $\mathcal{R}(s_i, s_j)$ is *true* or we can find a set of samples $s_{p_1}, s_{p_2}, ..., s_{p_L} \in \mathbf{P}$ such that $\mathcal{R}(s_i, s_{p_1})$, $\mathcal{R}(s_{p_k}, s_{p_{k+1}})$, and $\mathcal{R}(s_{p_L}, s_j)$ are all *true* for $1 \le k \le L-1$. For example, all samples with input feature value 3 in Figure 2(a) form a patch. A patch is *homogeneous* if its samples have similar feature vectors (e.g., by Euclidean distance) and its labeled samples, if exist, belong to only one class. For example, there are seven homogeneous patches highlighted in different gray scales in the first map of Figure 2(a).

Zone: A zone $\mathbf{Z}$ is a number of homogeneous patches that are spatially contiguous with each other. It is a set of spatially contiguous samples in a raster framework $\mathbf{Z} \subseteq \mathbf{F}$ with both labeled samples $\mathbf{L_Z} = \mathbf{L} \cap \mathbf{Z}$ and unlabeled samples $\mathbf{U_Z} = \mathbf{U} \cap \mathbf{Z}$. In the example of Figure 2(c), zone 1 consists of three homogeneous patches, while zone 2 consists of four homogeneous patches.

Class ambiguity refers to the phenomenon whereby samples with the same non-spatial feature vector belong to different classes, due

Table 1: A list of symbols and descriptions

Symbol	Description
$\mathbf{F}$	All samples in a raster framework
$\mathbf{L}$	All labeled samples in $\mathbf{F}$
$\mathbf{U}$	All unlabeled samples in $\mathbf{F}$
s_i	The ith spatial data sample
x_i	The vector of non-spatial features
l_i	The vector of two spatial coordinates
y_i	The class label of of sample s_i
$\mathcal{R}(s_i, s_j)$	Spatial neighborhood relationship
$\mathbf{P}$	A patch
$\mathbf{Z}$	A zone
$\mathbf{L_Z}$	All labeled samples in $\mathbf{Z}$
$N_k(s_i)$	Feature space neighborhood of s_i
$a(s_i)$	Per sample class ambiguity
$a(\mathbf{Z})$	Per zone class ambiguity

to spatial heterogeneity (e.g., heterogeneous terrains). For example, in Figure 2(a), the four samples labeled with feature value $x = 1$ belong to different classes (two *red* and two *green*). A global decision tree model makes erroneous predictions (Figure 2(b)). The degree of class ambiguity in a zone $\mathbf{Z}$ can be measured on its labeled samples $\mathbf{L_Z}$. We define the following three concepts to quantify class ambiguity:

Feature space neighborhood: Feature space neighborhood of a sample s_i among all labeled samples $\mathbf{L_Z}$ in zone $\mathbf{Z}$ is defined as $N_k(s_i) = \{s_j \in \mathbf{L_Z} | s_j \neq s_i, d(x_i, x_j) \text{ is the k smallest}\}$, where $d(x_i, x_j)$ is a metric function such as Euclidean distance. For example, for the red sample in the last column in the middle of Figure 2(a), its $N_2(s_i)$ can be any two labeled samples with $x = 1$ except the sample itself, including one *red* sample and two *green* samples. In this definition, we assume that labeled samples are locally dense in feature space to avoid the curse of dimensionality. In reality, this assumption is often satisfied due to the spatial autocorrelation effect (i.e.,nearby training samples often resemble each other).

Per sample class ambiguity on a labeled sample s_i among all labeled sample $\mathbf{L_Z}$ in zone $\mathbf{Z}$ is defined as the ratio of labeled samples in different class from s_i in its neighborhood $N_k(s_i)$. Formal definition is in Equation 1, where $I(\cdot)$ is an indicator function. For example, the class ambiguity of the red sample in the last column of Figure 2(a) is $\frac{1}{2} = 0.5$ if one *red* sample and one *green* sample (with feature $x = 1$) are selected as $N_2(s_i)$. Its value can also be $\frac{2}{2} = 1$ if both *green* samples with feature value 1 happen to be selected as $N_2(s_i)$.

$$a(s_i) = \frac{1}{k} \sum_{s_j \in N_k(s_i)} I(y_j \neq y_i), \tag{1}$$

The *per zone class ambiguity* of a zone is defined as the average of *per sample class ambiguity* over all labeled samples. It is formally defined in Equation 2. For example, in Figure 2(a-b), the class ambiguity in the zone of the entire raster framework is $(\frac{1}{2} \times 4 + \frac{1}{2} \times 4 + 0 \times 2 + 0 \times 4)/14 = 0.3$. Similarly, the per zone class ambiguity of $\mathbf{Z_1}$ or $\mathbf{Z_2}$ in Figure 2(c) is 0.

$$a(\mathbf{Z}) = \frac{1}{|\mathbf{L_Z}|} \sum_{s_i \in \mathbf{L_Z}} a(s_i), \tag{2}$$

A *spatial ensemble* is a decomposition of a raster framework $\mathbf{F}$ into m disjoint zones $\{\mathbf{Z_1}, \mathbf{Z_2}, .., \mathbf{Z_m}\}$ such that the average per zone class ambiguity is minimized. A local model can be learned in each zone $\mathbf{Z_i}$ based on its labeled (training) samples $\mathbf{L_{Z_i}}$, and then be used to classify unlabeled samples $\mathbf{U_{Z_i}}$ in the same zone. The concept of a local model in each zone can be generalized to a set of models (e.g., bagging, boosting, random forest) in the zone. In other words, spatial ensemble learning can be used together with traditional ensemble methods since they are orthogonal. Figure 2(c) shows an example of spatial ensemble with $m = 2$.

2.2 Problem Definition

The spatial ensemble learning problem is defined as follows:
Input:
• A geographic raster framework $\mathbf{F}$ with labeled samples $\mathbf{L}$ and unlabeled samples $\mathbf{U}$
• The number of zones in the spatial ensemble: m
• The parameter in feature space neighborhood: k
Output: A spatial ensemble with m contiguous zones such that:

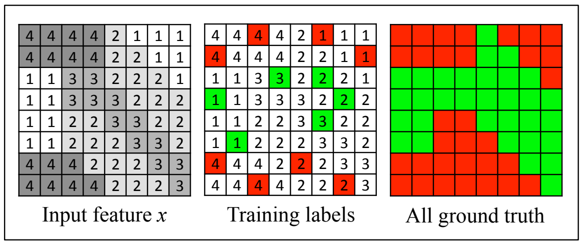
$$\begin{aligned} \underset{\mathbf{Z_1}, \mathbf{Z_2}, .., \mathbf{Z_m}}{\arg\min} \quad & \frac{1}{m} \sum_{i=1}^{m} a(\mathbf{Z_i}) \\ \text{subject to} \quad & (1) \mathbf{Z_i} \cap \mathbf{Z_j} = \emptyset \text{ for } i \neq j \\ & (2) \bigcup_{i=1}^{m} \mathbf{Z_i} = \mathbf{F} \end{aligned}$$

where $a(\mathbf{Z_i})$ is the per zone class ambiguity, and $f(\mathbf{Z_i})$ is the number of isolated patches.

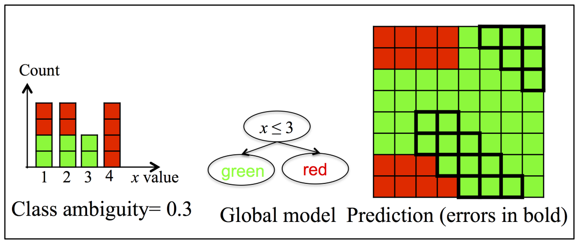
Figure 2 shows a problem example. Inputs include a geographic data with 64 samples, 14 labeled (training) and 50 unlabeled, with one feature x and two classes (*red*, *green*) (Figure 2(a)). The class ambiguity of the entire framework is $a(\mathbf{F}) = 0.3$, computed from the class histogram of training samples. A global decision tree makes prediction errors (Figure 2(b)). In contrast, a spatial ensemble with two zones in Figure 2(c) reduces per zone class ambiguity to zero. Predictions of local models show zero errors.

The spatial ensemble learning problem is formulated as a geographical partition problem because we assume that the underlying causes of class ambiguity is spatial heterogeneity. This phenomenon is also known as "ecological fallacy", or spatial Simpson's Paradox. Individual zones in spatial ensemble are contiguous to avoid overfitting (spatial regularization) and also to conform the first law of geography, "Everything is related to everything else, but nearby things are more relevant than distant things" [31]. There are several other assumptions in our problem formulation. First, we assume samples in the raster framework form homogeneous patches. This is often true due to the spatial autocorrelation effect, particularly when the pixel resolution is high. Second, we assume feature vectors of unlabeled (test) samples are given within the same raster framework of training samples. In other words, the problem belongs to transductive learning. This can limit the scope of the problem. Finally, we assume a pixel belongs to only one class, i.e., there is no class ambiguity within a pixel. The computational challenges of the problem is discussed in Theorem 2.1 below.

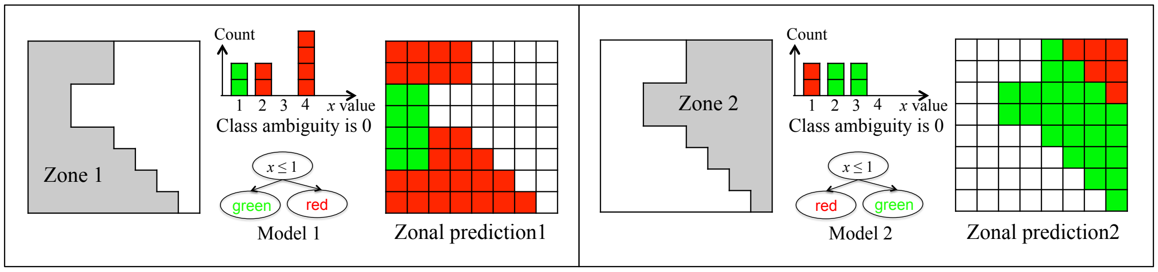
THEOREM 2.1. *The spatial ensemble learning problem is NP-hard.*



(a) Problem inputs



(b) Problem outputs for global model



(c) Problem outputs for spatial ensemble

Figure 2: Illustrative example of problem inputs and outputs (best viewed in color)

PROOF. Due to space limit, we only provide main ideas. First, our objective function of per zone class ambiguity is non-monotonic and non-distributive. Thus, we cannot compare one candidate zone partitioning against another without computing class ambiguity. Second, the number of possible zone partitioning is beyond polynomial. This can be derived from the NP-hardness of grid graph partitioning problems [7]. □

3 PROPOSED APPROACH

In this section, we present our algorithms to address computational challenges of the spatial ensemble learning problem. Our algorithms consist of two phases. First, input spatial data samples (both labeled and unlabeled) are clustered into homogeneous patches. We propose to use a constraint-based hierarchical spatial clustering approach (Section 3.1). After this, homogeneous patches are further grouped into contiguous zones through a recursive bisecting process (Section 3.2).

3.1 Preprocessing: Homogeneous Patches

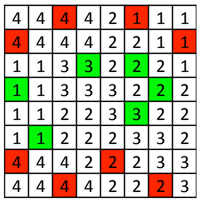
Given geographic data with all labeled and unlabeled samples, generating homogeneous patches can be considered as image segmentation [11] but with the constraint that labeled samples in the same patch, if exist, belong to the same class.

Algorithm 1 shows our bottom-up hierarchical method to generate homogeneous patches. First, each data sample is initialized as a patch (step 1). The algorithm then repeatedly merges pairs of *adjacent patches* (patches with samples that are spatial neighbors) in a greedy manner. Only patch pairs whose labeled samples belong to the same class can be merged (step 6). The patch pair whose samples have the smallest feature dissimilarity (step 7) are merged first (steps 10-11). Merging continues until the number of patches is reduced to a given number n. In implementation, we can

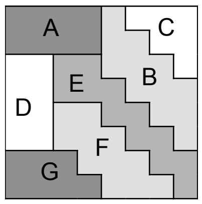
Algorithm 1 Homogeneous Patch Generation

```
Input:
  • All samples in the raster framework: $\mathbf{F}$
  • Spatial neighborhood relationship: $\mathcal{R}(\cdot,\cdot)$
  • The number of output patches: $n$, $n \ll |\mathbf{F}|$
Output:
  • A set of $n$ patches: $\mathbf{P} = \{\mathbf{P_1}, \mathbf{P_2}, ..., \mathbf{P_n}\}$
1:  Initialize a patch set $\mathbf{P} = \{\mathbf{P_i} = \{s_i\} | s_i \in \mathbf{F}\}$
2:  while number of patches $|\mathbf{P}| > n$ do
3:      for each adjacent pair $\mathbf{P_i}$ and $\mathbf{P_j}$ do
4:          if $d(\mathbf{P_i}, \mathbf{P_j})$ has been computed then
5:              Continue to next for iteration
6:          if $\mathbf{L_{p_i}}, \mathbf{L_{p_j}}$ either empty or same class then
7:              $d(\mathbf{P_i}, \mathbf{P_j}) \leftarrow \frac{1}{|\mathbf{P_i}||\mathbf{P_j}|} \sum_{s_i \in \mathbf{P_i}, s_j \in \mathbf{P_j}} d(x_i, x_j)$
8:          else
9:              $d(\mathbf{P_i}, \mathbf{P_j}) \leftarrow +\infty$
10:     Find $\mathbf{P_i}, \mathbf{P_j}$ with minimum dissimilarity $d(\mathbf{P_i}, \mathbf{P_j})$
11:     Merge these two patches: $\mathbf{P_i} \leftarrow \mathbf{P_i} \cup \mathbf{P_j}$, $\mathbf{P} \leftarrow \mathbf{P} \setminus \mathbf{P_j}$
12: return $\mathbf{P}$.
```

use a patch adjacency graph to efficiently find pairs of adjacent patches. The graph can be easily updated when two patches (nodes) are merged. Figure 3 shows a toy example. The input geographic data contains 64 samples with one feature and two classes (*red* and *green*). Adjacent samples with the same feature value are merged into a patch. For instance, all samples with feature value 4 in the upper left corner are merged into patch A. The final output is 7 homogeneous patches (shown by different shades: A to G).



(a) Input geographic data



(b) Output homogeneous patches

Figure 3: Illustration of homogeneous patch generation

Algorithm 1 has two major computational bottlenecks in its iterations: identification of the adjacent patch pair with the minimum dissimilarity on the entire map (step 10), and computation of dissimilarity values between new adjacent patch pairs (step 7). To address the first bottleneck, we propose to use a priority queue with adjacent patch pairs ordered by dissimilarity. To reduce the cost of patch dissimilarity computation, we reuse previously computed dissimilarity values when possible.

Details of these computational refinements are in Algorithm 2. The algorithm maintains a neighborhood graph where nodes are patches and edges are spatial adjacency between patches. Edge weights e_{ij} are dissimilarity values between adjacent patch pairs (v_i, v_j). Initially, the graph is a grid graph with each sample (pixel) as a node (patch) (steps 1-2). Then, the algorithm repeatedly merge two neighboring nodes with the minimum edge weight, until the total number of nodes (patches) are reduced to a required number n. In order to quickly find neighboring nodes with the minimum edge weight, we maintain a priority queue of all neighboring node pairs ordered by their edge weights (step 3), and extract the minimum element from the queue in each iteration (step 6). After extracted, the pair of nodes v_i, v_j are merged into a new node v_n (step 10), and the corresponding edges are also updated. When computing the weights of edges connected to the new node v_n, we reuse the weights of edges connected to nodes v_i, v_j (step 12) to avoid redundant computation (see definition of $d(\mathbf{P_i}, \mathbf{P_j})$ in steps 7 and 9 in Algorithm 1). The weights of new patch pairs are added to the priority queue (step 13). Once nodes v_i, v_j are merged, their corresponding elements in the priority queue become obsolete. Thus, we maintain a hash set of all obsolete nodes (steps 1 and 9) to ignore their elements in the priority queue (steps 7-8).

3.2 Group Homogeneous Patches into Zones

After samples are clustered into homogeneous patches, the second phase of our spatial ensemble learning method aims to divide these patches into several contiguous groups (zones) to minimize class ambiguity within each group (zone). This can be considered as a planar graph partition problem where nodes are patches and edges are spatial adjacency. In order to group patches (nodes) into multiple zones, we propose a bisecting algorithm (Algorithm 3). The algorithm starts with one zone containing the set of all patches (steps 1-2), and then keeps breaking down the current most ambiguous zone into two until the number of zones reaches a required number (steps 3-7). The critical question now becomes how to divide

Algorithm 2 Faster Homogeneous Patch Generation

```
Input:
    • All samples in the raster framework: $\mathbf{F}$
    • Spatial neighborhood relationship: $\mathcal{R}(\cdot,\cdot)$
    • The number of output patches: $n$, $n \ll |\mathbf{F}|$
Output:
    • A set of $n$ patches: $\mathbf{P} = \{\mathbf{P_1}, \mathbf{P_2}, ..., \mathbf{P_n}\}$
1: Initialize a patch set $\mathbf{P} = \{\mathbf{P_i} = \{s_i\} | s_i \in \mathbf{F}\}$
2: Initialize a neighborhood graph $G(V, E)$ with each patch as a node:
   $v_i = \mathbf{P_i} = \{s_i\}$ for $1 \le i \le |\mathbf{F}|$
   $e_{ij} = \begin{cases} d(x_i, x_j) & \text{if } s_i, s_j \text{ are neighbors, same class or unlabeled} \\ \infty & \text{otherwise} \end{cases}$
3: Create a priority queue $PQ$ with all neighbor pairs $(v_i, v_j, e_{ij})$
4: Initialize a set of obsolete nodes $O \leftarrow \emptyset$
5: while $|V| > n$ and $PQ$ not empty do
6:     $(v_i, v_j, e_{ij}) \leftarrow ExtractMin(PQ)$
7:     if $v_i \in O$ or $v_j \in O$ then
8:         Continue to next while iteration
9:     $O \leftarrow O \cup \{v_i\} \cup \{v_j\}$
10:    Create a new node $v_n$ merging $v_i, v_j$ in $G$ ($\mathbf{P_n} \leftarrow \mathbf{P_i} \cup \mathbf{P_j}$)
11:    for each other neighbor node $v_k$ of $v_i$ or $v_j$ do
12:
       $e_{k,n} = \begin{cases} \frac{e_{k,i}|\mathbf{P_k}|\cdot|\mathbf{P_i}|+e_{k,j}|\mathbf{P_k}|\cdot|\mathbf{P_j}|}{|\mathbf{P_k}|\cdot|(\mathbf{P_i}|+|\mathbf{P_j}|)} & \text{if } v_k \text{ neighbors both } v_i, v_j \\ \frac{d(\mathbf{P_k},\mathbf{P_i})|\mathbf{P_k}|\cdot|\mathbf{P_i}|+e_{k,j}|\mathbf{P_k}|\cdot|\mathbf{P_j}|}{|\mathbf{P_k}|\cdot|(\mathbf{P_i}|+|\mathbf{P_j}|)|} & \text{if } v_k \text{ neighbors } v_j \text{ only} \\ \frac{e_{k,i}|\mathbf{P_k}|\cdot|\mathbf{P_i}|+d(\mathbf{P_k},\mathbf{P_j})|\mathbf{P_k}|\cdot|\mathbf{P_j}|}{|\mathbf{P_k}|\cdot|(\mathbf{P_i}|+|\mathbf{P_j}|)} & \text{if } v_k \text{ neighbors } v_i \text{ only} \end{cases}$
13:        Add edge $(v_k, v_n, e_{k,n})$ to graph $G$
14:        Add $(v_k, v_n, e_{k,n})$ into priority queue $PQ$
15:    Remove obsolete nodes $v_i, v_j$ and their edges
16: return $\mathbf{P} = V$.
```

a zone (set of patches) into two to minimize class ambiguity. This is done via another subroutine called TwoZoneSpatialEnsemble (Algorithm 4) whose details are introduced below.

Since graph partitioning problems are generally computationally hard [7], in Algorithm 4, we propose a greedy heuristic that assign patches (graph nodes) into two zones maximizing *inter-zone* class ambiguity while minimizing *intra-zone* class ambiguity. To do that, the algorithm uses a seed growing process to expand two zones on a patch adjacency graph. At the beginning, all patches are marked as *unassigned* (step 2), and the class ambiguity of all patch pairs (whether adjacent or not) are computed (step 3). The algorithm finds the two patches with the highest class ambiguity as initial seeds, and assigns one patch to each zone respectively (steps 4-5). The algorithm also maintains a set of frontier nodes (unassigned spatially adjacent nodes) $\mathbf{F_1}, \mathbf{F_2}$ for each zone (steps 6-7). Next, the algorithm iteratively grows a zone by adding a node from its frontier until all nodes are *assigned* (i.e., two frontiers are empty).

When selecting a node from the frontier of a zone, we use a greedy heuristic that maximizes *inter-zone* class ambiguity while minimizes *intra-zone* class ambiguity. This is shown in the formula of A_k^1 and A_k^2 (steps 10 and 14). In the formula of A_k^1, the numerator

Algorithm 3 Bisecting Multi-zone Spatial Ensemble

```
Input:
    • A set of homogeneous patches: $\mathbf{P} = \{\mathbf{P_1}, \mathbf{P_2}, ..., \mathbf{P_n}\}$
    • The number of zones: $m$ ($m \ll n$)
    • The parameter in class ambiguity measure: $k$
    • The balancing parameter in our greedy heuristic: $\alpha$
Output:
    • A spatial ensemble of $m$ zones: $\mathbf{Z} = \{\mathbf{Z_1}, ..., \mathbf{Z_m}\}$
1: Initialize a zone with all input patches: $\mathbf{Z_1} \leftarrow \mathbf{P}$
2: Initialize a set of zones for outputs: $\mathbf{Z} \leftarrow \{\mathbf{Z_1}\}$
3: while $|\mathbf{Z}| < m$ do
4:     Find the zone with max class ambiguity:
       $\mathbf{Z_0} = \arg\max_{\mathbf{Z_i} \in \mathbf{Z}} a(\mathbf{Z_i})$
5:     Remove zone $\mathbf{Z_0}$ from result set: $\mathbf{Z} \leftarrow \mathbf{Z} \setminus \mathbf{Z_0}$
6:     $\{\mathbf{Z'_1}, \mathbf{Z'_2}\}$ = TwoZoneSpatialEnsemble($\mathbf{Z_0}$, $k$, $\alpha$)
7:     $\mathbf{Z} \leftarrow \mathbf{Z} \cup \{\mathbf{Z'_1}\} \cup \{\mathbf{Z'_2}\}$
8: return Z
```

Algorithm 4 Two Zone Spatial Ensemble

```
Input:
    • A set of homogeneous patches: $\mathbf{P} = \{\mathbf{P_1}, \mathbf{P_2}, ..., \mathbf{P_n}\}$
    • The parameter in class ambiguity measure: $k$
    • The weight parameter in our greedy heuristic: $\alpha$
Output:
    • A spatial ensemble of two zones: $\{\mathbf{Z_1}, \mathbf{Z_2}\}$
1: Create a spatial adjacency graph with patches as nodes
2: Initialize all nodes as unassigned
3: Compute class ambiguity $a_{ij} = a(\mathbf{P_i} \cup \mathbf{P_j})$ for any $i \neq j$
4: Find $\mathbf{P_i}, \mathbf{P_j}$ with max class ambiguity $a_{ij}$
5: Initialize $\mathbf{Z_1} \leftarrow \{\mathbf{P_i}\}$, $\mathbf{Z_2} \leftarrow \{\mathbf{P_j}\}$, mark $\mathbf{P_i}, \mathbf{P_j}$ as visited
6: Initialize $\mathbf{F_1}$ with all unassigned neighboring patches of $\mathbf{Z_1}$
7: Initialize $\mathbf{F_2}$ with all unassigned neighboring patches of $\mathbf{Z_2}$
8: while $\mathbf{F_1} \neq \emptyset$ or $\mathbf{F_2} \neq \emptyset$ do
9:     for each $\mathbf{P_k} \in \mathbf{F_1}$ do
10:        $A^1_k = \frac{1+\sup_{\mathbf{P_o} \in \mathbf{Z_2}} a(\mathbf{P_k}, \mathbf{P_o})}{1+\sup_{\mathbf{P_o} \in \mathbf{Z_1}} a(\mathbf{P_k}, \mathbf{P_o})}$ //class ambiguity avoidance
11:        $B^1_k \leftarrow$ SizeBalance($\mathbf{Z_1} \cup \{\mathbf{P_k}\}, \mathbf{Z_2}$) //zone size balance
12:        Compute overall score: $S^1_k \leftarrow \alpha A^1_k + (1-\alpha) B^1_k$
13:    for each $\mathbf{P_k} \in \mathbf{F_2}$ do
14:        $A^2_k = \frac{1+\sup_{\mathbf{P_o} \in \mathbf{Z_1}} a(\mathbf{P_k}, \mathbf{P_o})}{1+\sup_{\mathbf{P_o} \in \mathbf{Z_2}} a(\mathbf{P_k}, \mathbf{P_o})}$ //class ambiguity avoidance
15:        $B^2_k \leftarrow$ SizeBalance($\mathbf{Z_1}, \mathbf{Z_2} \cup \{\mathbf{P_k}\}$) //zone size balance
16:        Compute overall score: $S^2_k \leftarrow \alpha A^2_k + (1-\alpha) B^2_k$
17:    Find the $\mathbf{P_{k_0}} \in \mathbf{F_{f_0}}$ ($f_0 \in \{1, 2\}$) with max overall score
18:    $\mathbf{Z_{f_0}} \leftarrow \mathbf{Z_{f_0}} \cup \{\mathbf{P_{k_0}}\}$, mark $\mathbf{P_{k_0}}$ as visited
19:    $\mathbf{F_1} \leftarrow \mathbf{F_1} \setminus \{\mathbf{P_{k_0}}\}$, $\mathbf{F_2} \leftarrow \mathbf{F_2} \setminus \{\mathbf{P_{k_0}}\}$
20:    Expand $\mathbf{F_{f_0}}$ with all unassigned neighboring patches of $\mathbf{P_{k_0}}$
21: return $\{\mathbf{Z_1}, \mathbf{Z_2}\}$
```

Table 2: Patch pairs with non-zero class ambiguity

Patch $\mathbf{P_i}$	Patch $\mathbf{P_j}$	$a(\mathbf{P_i} \cup \mathbf{P_j})$
B	F	0.5
C	D	0.5
D	C	0.5
F	B	0.5

is the maximum class ambiguity between the candidate patch $\mathbf{P_k}$ and patches in the other zone $\mathbf{Z_2}$, reflecting inter-zone class ambiguity, while the denominator is the maximum class ambiguity between $\mathbf{P_k}$ and patches in its corresponding zone $\mathbf{Z_1}$, reflecting intra-zone class ambiguity. We add a value 1 in the formula for normalization. To avoid the case in which most patches are assigned to one single zone, we also add a size-balance factor B^1_k (B^2_k) to our heuristic. Size balance factor across two zones can be measured via the entropy $-r_1 \log r_1 - r_2 \log r_2$ where r_1 and r_2 are the ratio of the sizes (number of samples) of zone 1 and zone 2 to their total size. A higher entropy value indicates more size-balanced zones. We use a parameter α to weight the influence of two factors in our heuristic (step 12 and 16). The node with the maximum overall score $\mathbf{P_{k_0}}$ is selected, and is added to its corresponding zone $\mathbf{Z_{f_0}}$ (step 18). The node is then removed from frontiers. Its original frontier is expanded with the node's unassigned neighbors. Finally, all nodes are *assigned*, the frontiers become empty, and the two zones are returned (step 21).

Running example: Figure 4 shows a running example of Algorithm 4 with the same input data as the example in Figure 3. Assume $k = 2$, $\alpha = 0.5$, and $m = 2$. The adjacency graph of patches is shown in Figure 4(b). Patch pairwise class ambiguity is shown in Table 2. The two zones are shown by two different colors. Frontiers are shown by solid edges connected to zones. Initially, $\mathbf{Z_1} = \{C\}$ and $\mathbf{Z_2} = \{D\}$ (Figure 4(b)). The frontier of $\mathbf{Z_1}$ is $\{B\}$, while the frontier of $\mathbf{Z_2}$ is $\{A, E, F, G\}$. In the next iteration, all candidate nodes from the frontiers have zero class ambiguity avoidance score, but node B has the highest size balance score, so it is selected to grow $\mathbf{Z_1}$. Nodes F, A, G, E are then selected consecutively. The final output two zones are shown in Figure 4(i). This output is slightly different from our problem example in Figure 2, but both reduce class ambiguity to zero.

3.3 Theoretical Analysis

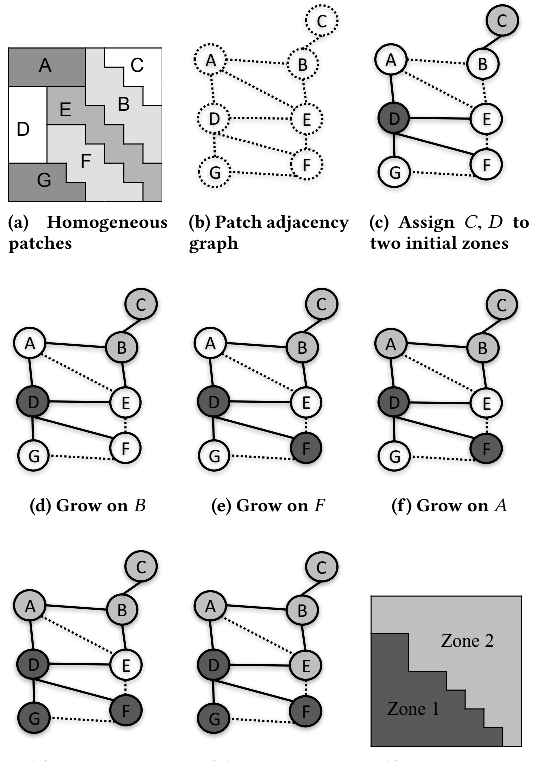
Theorem 3.1. *The expectation of our per zone class ambiguity measure is an upper bound of Bayesian error.*

Proof. We omit the detailed proof due to space limit. □

The Theorem 3.1 is important because Bayesian error rate is generally considered as the lowest possible error rate for any classifier in statistical classification. The fact that class ambiguity is an upper bound of Bayesian error means that minimizing class ambiguity in spatial ensemble learning can help reduce Bayesian error rate.

4 EXPERIMENTAL EVALUATION

The goal of the experiments was to:
• Evaluate the classification accuracy of spatial ensemble learning.



(a) Homogeneous patches (b) Patch adjacency graph (c) Assign C, D to two initial zones

(d) Grow on B (e) Grow on F (f) Grow on A

(g) Grow on G (h) Grow on E (i) Final 2 zones

Figure 4: A running example of Algorithm 3

- Test the sensitivity of spatial ensemble to its parameters.
- Evaluate the computational costs of spatial ensemble algorithms.

4.1 Experiment Setup

For classification performance evaluation, we compared spatial ensemble learning (learning models and making predictions within individual zones) with global model learning (i.e., learning models and making predictions on the entire study area). We used a single model, bagging, boosting, and random forest on both learning methods. For example, we can learn a random forest in each individual zone in our spatial ensemble learning method. We also tested the sensitivity of spatial ensemble learning to its parameters, including the number of zones m, the base classifier type, class ambiguity measure parameter k, and balancing parameter α in greedy heuristic. The number of patches n in preprocessing was determined via trying different values and visualizing the output homogeneous patches. For computational performance comparison, we compared our baseline and refined homogeneous patch generation algorithms (Algorithm 1 and 2). We also evaluated computational performance of bisecting spatial ensemble algorithm (Algorithm 3). All codes were implemented in Java and base classifiers were from Weka toolbox [1]. We conducted experiments on an iMac Desktop with 4 GHz Intel Core i7 processor and 32GB DDR3 main memory.

Dataset description: Our datasets were collected from two areas in Minnesota: *Chanhassen* and *Big Stone* [28]. We used eight

Table 3: Dataset Description

Scene	Training Samples		Test Samples	
	Dry	Wet	Dry	Wet
Chanhassen	6715	4323	40362	31254
Big Stone	45483	27138	345557	177762

Table 4: Results on Chanhassen ("SE" for spatial ensemble)

Ensemble Method	Confusion Matrix		F score
Global Single Model	36734	**3628**	0.76
	9640	21614	
Global Bagging	36497	**3865**	0.79
	8272	22982	
Global Boosting	35506	**4856**	0.79
	7646	23608	
Global Random Forest	36867	**3495**	0.79
	8349	22905	
SE with Single Model	37407	**2955**	0.92
	2073	29181	
SE with Bagging	37565	**2797**	0.93
	1871	29383	
SE with Boosting	37527	**2835**	0.93
	1851	29403	
SE with Random Forest	37609	**2753**	0.93
	1688	29566	

explanatory features, including four spectral bands (red, green, blue, near-infrared) in high resolution (3m by 3m) aerial photos from the National Agricultural Imagery Program during leaf-off season, and four corresponding texture on homogeneity [26]. Class labels (wetland and dry land) were collected from the updated National Wetland Inventory. The Chanhassen scene contains 221 by 374 pixels, and the BigStone scene contains 718 by 830 pixels. We used systematic clustered sampling to select training set, and used remaining pixels as test set (details in Table 3).

Evaluation metric: We evaluated the classification performance with confusion matrices, and F-score (harmonic mean of precision and recall) on the wetland class (wetland class is of more interest).

4.2 Classification Performance Evaluation

4.2.1 Comparison on Classification Accuracy. In Chanhassen data, we set parameter values as $n = 100$, $m = 6$, $k = 10$, $\alpha = 0.9$. In BigStone data, we set parameter values as $n = 800$, $m = 20$, $k = 10$, $\alpha = 0.9$ (n and m were set higher in BigStone than in Chanhassen because BigStone is larger). The classification accuracy results for the two datasets are summarized in Table 4, 5 respectively. In the confusion matrix displayed in each table, the first and second rows show *true* dry land and wetland samples respectively, and the first and second columns show *predicted* dry land and wetland samples respectively. We can see that in global models, bagging, boosting, and random forest slightly improve a single decision tree (overall F-score increases by 0.03), but significant errors remain. In contrast, the spatial ensemble of models improved the F-score of global models from around 0.8 to over 0.9 on Chanhassen data, and

Table 5: Results on BigStone ("SE" for spatial ensemble)

Ensemble Method	Confusion Matrix		F score
Global Single Model	305172	**40385**	0.75
	46760	131002	
Global Bagging	313625	**31932**	0.77
	45586	132176	
Global Boosting	307866	**37691**	0.76
	44813	132949	
Global Random Forest	317777	**27780**	0.78
	46263	131499	
SE with Single Model	316201	**29356**	0.85
	25300	152462	
SE with Bagging	316908	**28649**	0.86
	23162	154600	
SE with Boosting	315817	**29740**	0.85
	23397	154365	
SE with Random Forest	318009	**27548**	0.86
	22926	154836	

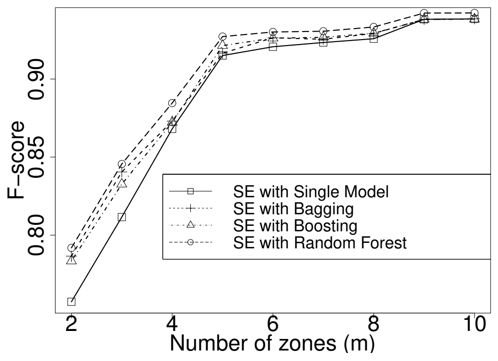
from 0.76 to 0.86 on BigStone data. The improvements can be seen in the reduction on the number of false negatives in the confusion matrices (lower left corner) (around 80% reduction on Chanhassen data, and around 50% reduction on BigStone data).

4.2.2 Effect of the Number of Zones m. To test the effect of the number of zones m in spatial ensemble learning, we fixed the other parameters the same as Section 4.2, but varying the number of zones m from 2 to 10 in Chanhassen data, and from 2 to 40 in BigStone data. We measured the overall F-score of the four spatial ensemble learning models over m. Results are summarized in Figure 5. As can be seen, as the number of zones m increases, the classification accuracy of all spatial ensemble learning models improves, and then reach to a plateau. Similar trends are shown on both datasets. In practice, the parameter m can be determined based on the size and homogeneity of the study area. The bigger and more heterogeneous a study area is, a larger m value is needed.

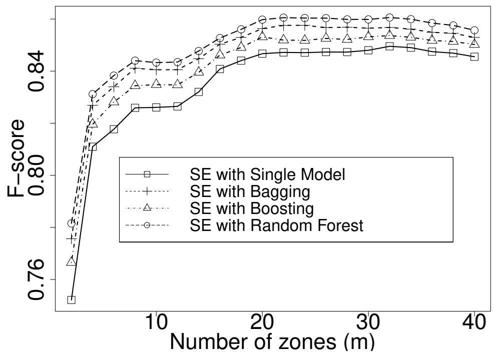
4.2.3 Effect of Base Classifier Type. The parameters were the same as Section 4.2. We chose several different base classifier types including decision tree (DT), SVM, neural network (NN), and logistic regression (LR). Results on Chanhassen data are shown in Figure 7. We can see that spatial ensemble learning consistently outperforms global model learning on different base classifier types. Trends were similar on BigStone data.

4.2.4 Effect of the Parameter k in Class Ambiguity Measure. We fixed the same parameters as Section 4.2 except that we varied the parameter k from 5 to 25. Results of our four spatial ensemble models on different parameter k are summarized in Figure 8. From the results, we can see that the spatial ensemble learning algorithm is not sensitive to the value of k. In practice, we can select $k = 5$. Similar trend was also observed on the BigStone data.

4.2.5 Effect of the Parameter α in Greedy Heuristic. We fixed the same parameters as Section 4.2 except that we varied the balancing parameter in our greedy heuristic α from 0 to 1. A higher α means a higher weight on class ambiguity avoidance than on



(a) Results on Chanhassen



(b) Results on BigStone

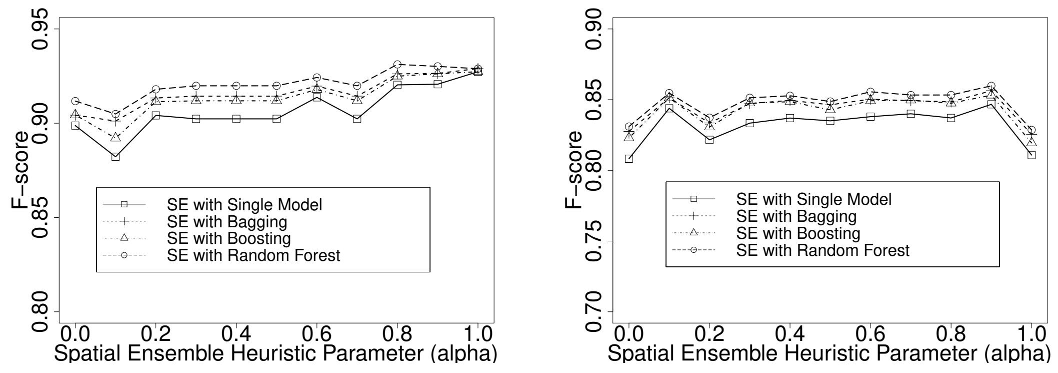
Figure 5: Effect of the number of zones *m*

zone size balance. Results are summarized in Figure 6. We can see that the spatial ensemble learning results are generally stable. In Chanhassen data, classification performance slightly improves as α increases. In BigStone data, classification performance stays stable except for the extreme cases $alpha = 0$ and $\alpha = 1$. In practice, we can determine the value of α based on cross-validation.

4.3 Computational Performance Evaluation

We now discuss the computational time costs of our spatial ensemble learning algorithms, including the homogeneous patch generation phase, and the bisecting spatial ensemble phase.

To evaluate the time costs of homogeneous patch generation phase, we compared our baseline algorithm (Algorithm 1) and refined algorithm (Algorithm 2) on different parameter values of n (the number of patches). We used the Chanhassen data with 82,654 total input samples. We varied the values of parameter n from 82,000 to 100. Results are shown in Figure 9. We can see that as n decreases, the time costs of both algorithms increase (due to more merging operations), but the cost of baseline is far higher than the refined algorithm. The growth rate of time cost in the baseline algorithm gradually gets lower with decreasing n. The reason is that as the patch adjacency graph gets smaller, the cost of finding the best patch pair with the minimum dissimilarity is also lower. In contrast,



(a) Results on Chanhassen (b) Results on BigStone

Figure 6: Effect of balancing parameter α in spatial ensemble

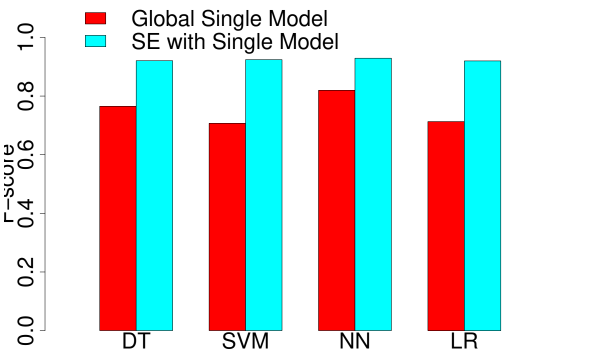


Figure 7: Effect of the base classifier type on Chanhassen data

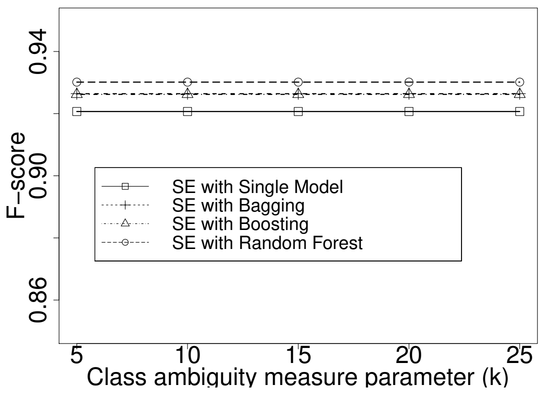


Figure 8: Effect of class ambiguity measure parameter k on Chanhassen data

the growth rate in the refined algorithm gets higher. The reason is that as patches get larger, the cost computing dissimilarity is more

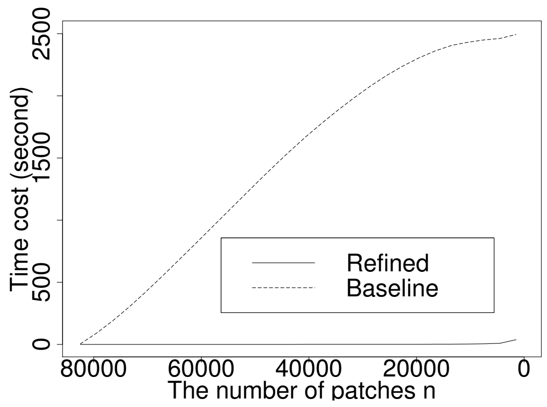


Figure 9: Time costs of baseline and refined algorithms in homogeneous patch generation

expensive (finding the patch pair with the minimum dissimilarity from a priority queue is very fast).

We measured the time costs of homogeneous patch generation (the refined algorithm) and bisecting spatial ensemble on the two datasets we used. The parameter settings were the same as Section 4.2. Results are summarized in Table 6. The time costs are averages of five runs. The reported time does not include local model learning time. Our algorithms can process over half a million samples within several minutes.

Table 6: Computational time costs of spatial ensemble

	Chanhassen	BigStone
Number of samples	82,654	595,940
Patch Generation	45 second	238 seconds
Bisecting Spatial Ensemble	6 seconds	190 seconds

5 DISCUSSION

There are other relevant works to our problem. Geographic Object Based Image Analyze (GEOBIA) [12] is a popular technique for earth imagery classification. GEOBIA first segments earth imagery into objects and then treats objects as minimum classification units. Segmentation can be done by software tools (e.g., eCognition) based on feature similarity (e.g., color, texture) often semi-automatically with human in the loop. Results are promising (e.g., reducing salt-and-pepper noise) particularly on high-resolution earth imagery. The main difference from our work is on the goal of space partition: GEOBIA partitions image based on feature similarity to recognize objects, while our spatial ensemble approach partitions space into zones to minimize class ambiguity. To consider class ambiguity in existing GEOBIA, extra manual efforts are often needed such as adding object features like “distance to roads". In fact, image segmentation in GEOBIA can be used in the preprocessing step of our approach (Algorithm 1-2) to generate homogeneous patches. After this, our spatial ensemble algorithms (Algorithm 3-4) can be applied to assign patches (or image segments) into different zones to minimize class ambiguity. There are other spatial classification methods that address spatial autocorrelation, including spatial decision trees [17–19]. These methods are orthogonal and complementary to spatial ensemble learning.

6 CONCLUSION

This paper investigates the spatial ensemble learning problem for heterogeneous geographic data with class ambiguity. We proposed spatial ensemble learning algorithms that consists of two phases: generating homogeneous patch from input spatial data samples, and grouping homogeneous patches into different zones to reduce class ambiguity via a greedy heuristic. We analyzed the theoretical properties of proposed algorithms on effectiveness. Evaluations on real world datasets show that our spatial ensemble approach outperforms global models in classification accuracy. Computational experiments also show that proposed computational refinements are effective in reducing time cost.

In future work, we plan to evaluate proposed algorithms on other applications such as spatial modeling in hydrological data. We can also investigate inductive spatial ensemble learning (spatial transfer learning), whereby test samples can be from a different spatial framework from the training samples.

ACKNOWLEDGMENT

This project is supported by the NSF under Grant No. 1029711, IIS-1320580, 0940818 and IIS-1218168, the USDOD under Grant No. HM1582-08-1-0017 and HM0210-13-1-0005, the OVPR U-Spatial and MSI at the University of Minnesota.

REFERENCES

[1] 2016. Weka 3: Data Mining Software in Java. "http://www.cs.waikato.ac.nz/ml/weka/". (2016).
[2] Leo Breiman. 1996. Bagging predictors. *Machine learning* 24, 2 (1996), 123–140.
[3] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.
[4] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *Multiple classifier systems*. Springer, 1–15.
[5] Jian Dong, Wei Xia, Qiang Chen, Jianshi Feng, Zhongyang Huang, and Shuicheng Yan. 2013. Subcategory-aware object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 827–834.
[6] Mathieu Fauvel, Yuliya Tarabalka, and et al. 2013. Advances in spectral-spatial classification of hyperspectral images. *Proc. IEEE* 101, 3 (2013), 652–675.
[7] Andreas Emil Feldmann. 2013. Fast balanced partitioning is hard even on grids and trees. *Theoretical Computer Science* 485 (2013), 61–68.
[8] A Stewart Fotheringham, Chris Brunsdon, and Martin Charlton. 2003. *Geographically weighted regression*. John Wiley & Sons, Limited.
[9] Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55, 1 (1997), 119–139.
[10] André R Gonçalves, Fernando J Von Zuben, and Arindam Banerjee. 2015. Multi-label structure learning with ising model selection. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 3525–3531.
[11] Robert M Haralick and Linda G Shapiro. 1985. Image segmentation techniques. In *Technical Symposium East*. International Society for Optics and Photonics, 2–9.
[12] GJ Hay and G Castilla. 2008. Geographic Object-Based Image Analysis (GEOBIA): A new name for a new discipline. In *Object-based image analysis*. Springer, 75–89.
[13] Tin Kamo Ho and Mitra Basu. 2002. Complexity measures of supervised classification problems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24, 3 (2002), 289–300.
[14] Tin Kam Ho, Mitra Basu, and Martin Hiu Chung Law. 2006. Measures of geometrical complexity in classification problems. In *Data complexity in pattern recognition*. Springer, 1–23.
[15] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation* 3, 1 (1991), 79–87.
[16] Zhe Jiang and Shashi Shekhar. 2017. *Spatial Big Data Science: Classification Techniques for Earth Observation Imagery*. Springer.
[17] Zhe Jiang, Shashi Shekhar, Pradeep Mohan, Joseph Knight, and Jennifer Corcoran. 2012. Learning spatial decision tree for geographical classification: a summary of results. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 390–393.
[18] Zhe Jiang, Shashi Shekhar, Xun Zhou, Joseph Knight, and Jennifer Corcoran. 2013. Focal-test-based spatial decision tree learning: A summary of results. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE, 320–329.
[19] Zhe Jiang, Shashi Shekhar, Xun Zhou, Joseph Knight, and Jennifer Corcoran. 2015. Focal-test-based spatial decision tree learning. *IEEE Transactions on Knowledge and Data Engineering* 27, 6 (2015), 1547–1559.
[20] Goo Jun and Joydeep Ghosh. 2013. Semisupervised learning of hyperspectral data with unknown land-cover classes. *Geoscience and Remote Sensing, IEEE Transactions on* 51, 1 (2013), 273–282.
[21] Anuj Karpatne, Zhe Jiang, Ranga Raju Vatsavai, Shashi Shekhar, and Vipin Kumar. 2016. Monitoring land-cover changes: A machine-learning perspective. *IEEE Geoscience and Remote Sensing Magazine* 4, 2 (2016), 8–21.
[22] Anuj Karpatne, Ankush Khandelwal, and Vipin Kumar. 2015. Ensemble Learning Methods for Binary Classification with Multi-modality within the Classes. In *Proceedings of the SIAM International Conference on Data Mining, 2015*. SIAM, 730–738. https://doi.org/10.1137/1.9781611974010.82
[23] Dengsheng Lu and Qihao Weng. 2007. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing* 28, 5 (2007), 823–870.
[24] Barbara Pease and Allan Pease. 2006. *The Definitive Book of Body Language*. Bantam.
[25] Steven Piantadosi, David P Byar, and Sylvan B Green. 1988. The ecological fallacy. *American journal of epidemiology* 127, 5 (1988), 893–904.
[26] Anne Puissant, Jacky Hirsch, and Christiane Weber. 2005. 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 (2005), 733–745.
[27] Viswanath Ramamurti and Joydeep Ghosh. 1996. Advances in using hierarchical mixture of experts for signal classification. In *Acoustics, Speech, and Signal Processing. IEEE International Conference on*, Vol. 6. IEEE, 3569–3572.
[28] Lian P Rampi, Joseph F Knight, and Keith C Pelletier. 2014. Wetland mapping in the upper midwest United States. *Photogrammetric Engineering & Remote Sensing* 80, 5 (2014), 439–448.
[29] Ye Ren, Le Zhang, and PN Suganthan. 2016. Ensemble Classification and Regression-Recent Developments, Applications and Future Directions [Review Article]. *Computational Intelligence Magazine, IEEE* 11, 1 (2016), 41–53.
[30] Martin Szummer and Rosalind W Picard. 1998. Indoor-outdoor image classification. In *Content-Based Access of Image and Video Database. Proceedings., 1998 IEEE International Workshop on*. IEEE, 42–51.
[31] Waldo R Tobler. 1970. A computer movie simulating urban growth in the Detroit region. *Economic geography* 46 (1970), 234–240.
[32] Lei Xu, Michael I Jordan, and Geoffrey E Hinton. 1995. An alternative model for mixtures of experts. *NIPS* (1995), 633–640.
[33] Seniha Esen Yuksel, Joseph N Wilson, and Paul D Gader. 2012. Twenty years of mixture of experts. *Neural Networks and Learning Systems, IEEE Transactions on* 23, 8 (2012), 1177–1193.
[34] Zhi-Hua Zhou. 2012. *Ensemble methods: foundations and algorithms*. CRC Press.