Spatial Decision Tree: A Novel Approach to Land-Cover Classification

Zhe Jiang¹, Shashi Shekhar¹, Xun Zhou¹, Joseph Knight², Jennifer Corcoran²

¹Department of Computer Science & Engineering
²Department of Forest Resources

University of Minnesota, Twin Cities
Highlights

• Public engagement with science and technology:
  – Coursera MOOC, “From GPS and Google Maps to Spatial Computing”,
  – reached 21,844 participants across 182 countries

• Enhanced infrastructure for education
  – Interdisciplinary survey paper on spatiotemporal change footprint discovery
  – APA/IEEE Computing in Sc. & Eng. special issue on “Computing and Climate”

• Enhanced infrastructure for research
  – Spatial decision trees can help improve wetland maps for climate models
Highlights

• Understanding
  – Large semantic gap between Data Science and Climate Science
    • Data Science results are hard to interpret in Climate Science
  – Data Science assumptions violate laws of physics
    • unnecessary errors, e.g., salt and pepper noise

• Concepts:
  – Physics-Guided Data Mining concepts are potentially transformative
  – Ex. Spatial Decision Trees: explicit physics (e.g., continuity) to wetland mapping
  – Ex. Intervals of Persistent Change detection uses Physics (e.g., violation of continuity)
Spatial Decision Tree: Motivation

- Wetland mapping:
  - Climate Change: wetlands – major source of methane\(^1\)
  - manage natural disasters, defense against hurricanes, buffer of floods.
  - maintain biodiversity, habitats to wildlife species

\(^1\)Bryan Walsh, How Wetlands Worsen Climate Change, Time, Magazine, 2010
Wetland Mapping Example

Input:

(a) aerial photo  (b) aerial photo  (c) true classes

(d) DT prediction

Output:

Training samples: upper half
Test samples: lower half

Spatial neighborhood:

DT: decision tree

(aerial photo) wetland  (aerial photo) dry land
Challenges

- Spatial autocorrelation effect
  - samples violate i.i.d. assumption
  - salt-and-pepper noise (white circles)
- Spatial anisotropy
  - asymmetric spatial neighborhood (blue circle)
- Spatial heterogeneity
  - areas with the same features correspond to distinct class labels (white circle)
- High computational cost
  - large amount of focal computation with different spatial neighborhoods sizes
Problem Statement

• Given:
  – training & test samples from a raster spatial framework
  – spatial neighborhood, its maximum size

• Find:
  – a (spatial) decision tree

• Objective:
  – minimize classification error and salt-and-pepper noise

• Constraint:
  – training samples are contiguous patches
  – spatial autocorrelation, anisotropy, and heterogeneity exist
  – training dataset can be large with high computational cost
Example with Decision Tree

In this example, Gamma index $\Gamma_1$ on feature $f_1$ is unique. Most often, $\Gamma_1$ is computed on the fly.

<table>
<thead>
<tr>
<th>ID</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
<td>3</td>
<td>red</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>3</td>
<td>red</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>2</td>
<td>green</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>E</td>
<td>3</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>F</td>
<td>3</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>G</td>
<td>3</td>
<td>3</td>
<td>red</td>
</tr>
<tr>
<td>H</td>
<td>1</td>
<td>2</td>
<td>green</td>
</tr>
<tr>
<td>I</td>
<td>1</td>
<td>2</td>
<td>green</td>
</tr>
<tr>
<td>J</td>
<td>3</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>K</td>
<td>1</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>L</td>
<td>3</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>M</td>
<td>1</td>
<td>2</td>
<td>green</td>
</tr>
<tr>
<td>N</td>
<td>1</td>
<td>2</td>
<td>green</td>
</tr>
<tr>
<td>O</td>
<td>3</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>P</td>
<td>3</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>Q</td>
<td>3</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>1</td>
<td>red</td>
</tr>
</tbody>
</table>

**Input:**

**Output:**

---

**Gammaton: $\Gamma_1$**

This parameter is computed on the fly.

**Decision Tree:**

- $f_1 \leq 1$

  - **Yes:**
    - green
  - **No:**
    - red

**Predicted Map:**

- A B C D E F
- G H I J K L
- M N O P Q R

**Salt-and-Pepper Noise:**

- Pixel K is identified as noise from the decision tree.
## Related Work Summary

<table>
<thead>
<tr>
<th></th>
<th>Existing Work</th>
<th>Proposed Work</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tree</strong></td>
<td>local feature test &amp; information gain:</td>
<td>focal feature test &amp; spatial information gain:</td>
</tr>
<tr>
<td><strong>Ensemble</strong></td>
<td>bootstrap sampling:</td>
<td>geographic space partitioning:</td>
</tr>
</tbody>
</table>

- Single decision tree
- Traditional decision tree **spatial decision tree**
- Random forest ensemble **spatial ensemble**
- Ensemble of decision trees
Proposed Approach – Focal Test

- Focal feature test
  - Test both local and focal (neighborhood) information
  - focal test uses local autocorrelation statistics, e.g., Gamma index
Proposed Approach - 2

- tree traversal direction depends on both local and focal (neighborhood) information
- focal test uses local autocorrelation statistics, e.g., Gamma index (Γ)
- neighborhood

\[
\Gamma_i = \frac{\sum_j S_{i,j} W_{i,j}}{\sum_j W_{i,j}}
\]

where:
- i, j: pixel locations
- \(S_{i,j}\): similarity between location i and j
- \(W_{i,j}\) is adjacency matrix element
Example – Focal Tests

### traditional decision tree

**inputs:** table of records

<table>
<thead>
<tr>
<th>ID</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$\Gamma_1$</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>green</td>
</tr>
<tr>
<td>H</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>green</td>
</tr>
<tr>
<td>I</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>green</td>
</tr>
<tr>
<td>K</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>red</td>
</tr>
<tr>
<td>M</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>green</td>
</tr>
<tr>
<td>N</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>green</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>red</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>E</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>F</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>G</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>J</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>L</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>O</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>P</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>red</td>
</tr>
<tr>
<td>Q</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>red</td>
</tr>
</tbody>
</table>

**predicted map**

### spatial decision tree

**inputs:** feature maps, class map

**predicted map**

**focal function $\Gamma_1$**

$$ (f_1 \leq 1) \ast (\Gamma_1 \geq 0) $$

**predicted map**
Evaluation: Case Study

• Questions to answer:
  – SDT v.s. DT - classification accuracy
  – SDT v.s. DT - salt-and-pepper noise
  – Computational scalability of SDT

• Dataset:
  – Chanhassen, MN (wetland mapping)
    • 2 classes: wetland, dry land
    • features: high resolution (3m*3m) aerial photos (RGB, NIR, NDVI) in 2003, 2005, 2008
    • Training set: randomly select circular patches; Test set: remaining pixels on the scene; Three scenes are used.
    • Max neighborhood size: 11 pixels by 11 pixels
Wetland Mapping Comparison – Scene 1

Input:
(a) aerial photo  (b) aerial photo  (c) true classes

Training samples: upper half
Test samples: lower half
Spatial neighborhood:

Output:
(d) DT prediction  (e) SDT prediction

DT: decision tree
SDT: spatial decision tree
(11x11 neighborhood)
Classification Performance – Scene 2

Trends:
1. DT: salt-and-pepper noise
2. SDT improve accuracy, salt-and-pepper noise levels
Evaluation: Classification Performance

Classification accuracy and salt-and-pepper noise level

<table>
<thead>
<tr>
<th>Model</th>
<th>Confusion Matrix</th>
<th>Prec.</th>
<th>Recall</th>
<th>F measure</th>
<th>Autocorrelation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>99,141</td>
<td>10,688</td>
<td>0.81</td>
<td>0.75</td>
<td>0.78</td>
</tr>
<tr>
<td></td>
<td>15,346</td>
<td>45,805</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDT</td>
<td>99,390</td>
<td>10,439</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
</tr>
<tr>
<td></td>
<td>10,618</td>
<td>50,533</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Significance test between confusion matrices:

<table>
<thead>
<tr>
<th>Model</th>
<th>Khat</th>
<th>Khat Variance</th>
<th>Z-score</th>
<th>significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>0.66</td>
<td>3.6*10^{-6}</td>
<td>26.8</td>
<td>significant</td>
</tr>
<tr>
<td>SDT</td>
<td>0.73</td>
<td>3.0*10^{-6}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Spatial decision tree reduces salt-and-pepper noise and misclassification errors, compared with decision trees.
Computational Bottleneck Analysis

Analysis:
1. focal computation takes the vast majority of the time
2. focal computation cost increases faster with the training set size

Focal computation is the bottleneck!
Incremental Update Approach

**Key idea:** reduce redundant focal computation by reusing results across candidate test thresholds $\Gamma(f < \delta)$

<table>
<thead>
<tr>
<th>Candidate $\delta$</th>
<th>Feature Values</th>
<th>Indicators, Focal Values for $\delta=1$</th>
<th>Indicators, Focal Values for $\delta=2$</th>
<th>Indicators, Focal Values for $\delta=3$</th>
<th>Indicators, Focal Values for $\delta=4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${1, 2, 3, 4, 5, 6, 7, 8}$</td>
<td><img src="image1" alt="Table" /></td>
<td><img src="image2" alt="Table" /></td>
<td><img src="image3" alt="Table" /></td>
<td><img src="image4" alt="Table" /></td>
<td><img src="image5" alt="Table" /></td>
</tr>
</tbody>
</table>

$$\Gamma_i = \frac{\sum_j W_{i,j} I_i I_j}{\sum_j W_{i,j}}$$
The refined algorithm significantly reduces computational cost.

Notation of symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F$</td>
<td># of features (12)</td>
</tr>
<tr>
<td>$N$</td>
<td># of samples</td>
</tr>
<tr>
<td>$N_d$</td>
<td># of distinct feature values</td>
</tr>
<tr>
<td>$S_{max}$</td>
<td>max neigh size</td>
</tr>
<tr>
<td>$N_0$</td>
<td>min node size</td>
</tr>
</tbody>
</table>
Conclusions

• Ignoring auto-correlation leads to errors, e.g., salt-n-pepper noise
• Proposed a novel spatial decision tree approach with focal tests
• Evaluation shows that proposed method reduced salt-n-pepper noise
  – And improved classification accuracy
• Designed computational refinements to improve scalability
Publications on Spatial Decision Trees


Challenges Revisited

- **Spatial autocorrelation effect**
  - samples violate i.i.d. assumption
  - salt-and-pepper noise (white circles)

- **Spatial anisotropy**
  - asymmetric spatial neighborhood (blue circle)

- **Spatial heterogeneity**
  - areas with the same features correspond to distinct class labels (white circle)

- **High computational cost**
  - large amount of focal computation with different spatial neighborhoods sizes
Future Work

• Key idea I: focal feature test
  – tree traversal direction depends on both *local* and *focal* (neighborhood) information
  – focal test uses local autocorrelation statistics, e.g., Gamma index

• Key idea II: spatial information gain (SIG)
  – SIG = Info. Gain * α + Spatial Autocorrelation * (1 − α)
  – tree node test selection depends on both *class purification* and *autocorrelation structure*

• Key idea III: spatial ensemble of local trees
  – geographic space partitioning, learn local classifiers
Proposed Approach: Spatial Ensemble

**traditional ensemble**
(random forest)

1. assume i.i.d. distribution
2. bootstrap sampling
3. learn a tree from one sampling with random feature subsets

**spatial ensemble**
(spatial forest)

1. assume spatial heterogeneity
2. spatial partitioning
3. learn local tree model in each partition

Diagram:
- **f₁ ≤ 1**
  - yes
  - red
  - green
  - no

- **f₁ ≤ 1**
  - yes
  - red
  - green
  - no