Learning Spatial Decision Tree For Geographical Classification: A Summary of Results

Zhe Jiang, Shashi Shekhar, Pradeep Mohan
Department of Computer Science & Engineering
University of Minnesota
zhe,shekhar,mohan@cs.umn.edu

Joseph Knight, Jennifer Corcoran
Department of Forest Resources
University of Minnesota
jknight,murph636@umn.edu

ABSTRACT

Given learning samples from a spatial raster dataset, the geographical classification problem aims to learn a decision tree classifier that minimizes classification errors as well as salt-n-pepper noise. The problem is important in many applications, such as land cover classification in remote sensing and lesion classification in medical diagnosis. However, the problem is challenging due to spatial autocorrelation. Existing decision tree learning algorithms, i.e. ID3, C4.5, CART, produce a lot of salt-n-pepper noise in classification results, due to their assumption that data items are drawn independently from identical distributions. In contrast, we propose a spatial decision tree learning algorithm, which incorporates spatial autocorrelation effect by a new spatial information gain (SIG) measure. The proposed approach is evaluated in a case study on a remote sensing dataset from Chanhassen, MN. Case study results show that the proposed approach outperforms the traditional approach in not only reducing salt-n-pepper noise but also improving classification accuracy.

Categories and Subject Descriptors

H.2.8 [**Database Management:**]: Database Applications—*Data mining*; J.2 [**Computer Applications:**]: Physical Science And Engineering—*Earth and atmospheric sciences*

General Terms

Algorithms, Experimentation

Keywords

spatial data mining, spatial autocorrelation aware decision tree , land cover classification

1. INTRODUCTION

Given learning samples from a spatial raster dataset, the geographical classification problem aims to learn a decision tree classifier that best characterizes the relationship between the explanatory features and ground truth class labels in the data, namely, minimizing classification errors and salt-n-pepper noise. Once a decision tree classifier is learned, it can be applied to future datasets where the ground truth class labels are unknown and need to be predicted.

Societal Application: The geographical classification problem is important for many applications ranging from land cover classification[5] to lesion classification in medical diagnosis[4], etc. For example, in the remote sensing community, land cover classification, especially wetland mapping, is critically important for natural resource management concerns[3][7]. In climate science domain, studies show that wetlands contribute to over half of the global emissions of methane, a powerful greenhouse gas; and accurate wetland mapping thus helps climate scientists to understand global warming process[14].

Challenges: The geographical classification problem is challenging due to the existence of spatial autocorrelation structure arising from phenomena such as "patches". For example, in wetland mapping applications, ground truth class labels (e.g., wetland) often form geographic "patches" (contiguous areas of the same class). In high resolution remote sensing images, the pixel size is much smaller than the patch size. Thus, a geographically contiguous patch may have many pixels with the same ground truth class label, leading to high spatial autocorrelation.

Related Work: Traditional decision tree learning algorithms include ID3[9], C4.5[10] and CART[11]. When applied to geographical classification, these algorithms implicitly assume the data items are independent, ignoring spatial autocorrelation effect. Thus, the classification result contains salt-n-pepper noise. In contrast, this paper proposes a spatial autocorrelation aware decision tree learning algorithm. Compared with traditional decision tree learning algorithm, its goal is to reduce salt-n-pepper noise, which may also improve classification accuracy.

Contributions: This paper makes the following contributions: (a)we propose a new spatial decision tree (SDT) model, including a spatial information gain interestingness measure; (b)we develope an SDT learning algorithm; (c)we conduct a case study to evaluate the proposed approach on a real world remote sensing dataset.

Scope: This paper focuses on decision tree classifiers. Other geographical classification techniques such as Bayesian classifier, SVM, neural network, logistic regression, GEOBIA (Geographical Object Based Image Analysis), etc. fall outside the scope of the study. The paper does not consider

ACM SIGSPATIAL GIS'12 November 6-9, 2012. Redondo Beach, CA, USA

ensembles of decision trees, e.g. bagging, boosting, etc.

2. PROBLEM FORMULATION

Formally, the problem may be defined as follows:
Given: A spatial framework s, learning samples (i.e. training, test samples) located in s, each sample specifies explanatory features and a class label
Find: a decision tree classifier based on training set
Objective: minimize classification error on test set
Constraint:
(a): spatial framework is a regular grid
(b): spatial autocorrelation exists in ground truth class labels

3. PROPOSED APPROACH

This section introduces the proposed approach. First we explain some basic concepts and define our interestingness measure. We then describe the spatial decision tree learning algorithm with its pseudocode.

3.1 Basic Concepts

In the training (tree construction) phase, learning samples are recursively split by a discriminative feature with a test threshold (e.g., $F \leq 10$). The goal is to reduce the impurity of class distributions as much as possible. **Entropy** is a measure of the impurity of class probability distribution. Formally, $Entropy = -\sum_l p_l \log p_l$ where p_l is the probability of class l. The entropy after split, E', is the average of entropy in each split result, shown as follows: $E' = \frac{n_1}{n_1+n_2}E_1 + \frac{n_2}{n_1+n_2}E_2$, where n_1, n_2 and E_1, E_2 are the number and entropy of learning samples in each split result respectively. **Information gain** is the decrease of entropy, defined as: $IG = E - E'$. Traditional decision tree learning algorithm selects the feature and test threshold which purify the classes to the most, namely, with the maximum information gain.

3.2 Proposed Interestingness Measure

Spatial Autocorrelation: Spatial datasets often show *autocorrelation*, instead of being *independently identically distributed* as assumed by traditional decision tree learning algorithms. When evaluating the split of a candidate feature with a test threshold, the algorithm should not only consider the class purification (reflected by IG), but also account for change on spatial autocorrelation structure. Xiang Li et al[8] proposed a distance-based autocorrelation structure measure to reflect global clusterness of homogeneous samples, but spatial autocorrelation is often significant at local neighborhood level. Thus, we adopt the local gamma autocorrelation[1], formally defined as: $\Gamma_i = \sum_j a_{i,j} b_{i,j} = \sum_j w_{i,j}\delta_{i,j} = c_i$, where i, j are sample indices; $a_{i,j}$, $b_{i,j}$ are spatial similarity and class similarity, further represented by W-matrix $w_{i,j}$ and indicator function $\delta_{i,j}$ (value is 1 when same class); c_i is count of homogeneous neighbors.

Neighborhood Split Autocorrelation Ratio (NSAR): *NSAR per sample* measures how much autocorrelation structure remains after the neighborhood of a sample is split. Formally, NSAR of sample i is defined as: $NSAR_i = \frac{\Gamma'_i}{\Gamma_i}$, where Γ_i and Γ'_i are local gamma of sample i before and after split respectively. Its value ranges between 0 and 1 (in special case when $\Gamma_i = 0$, we could define $NSAR_i$ as 1). The higher the NSAR, the less salt-n-pepper noise a split makes. Figure 1 shows an example of a split of a neighborhood graph. Before the split in (a), the central sample (with F value 30) has four homogeneous neighbors. However, after the split by $F <= 10$, no homogeneous neighbors remain in the same subgraph (shown in subfigure (b)). Thus, the NSAR value of the central sample is $0/4 = 0$.

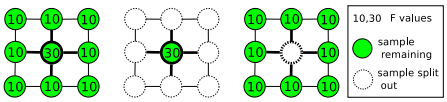


Figure 1: Example of neighborhood split $F \leq 10$: three graphs from left to right are samples before split, split result 1 and split result 2 respectively.

NSAR of all samples is defined as the average over NSAR per sample. Formally, $\overline{NSAR} = \frac{1}{m}\sum_{i=1}^{m} NSAR_i$, where i is the index of a sample, varying from 1 to m (m is the number of samples).

Spatial Information Gain (SIG): SIG is a balance between reduction of class impurity and maintenance of spatial autocorrelation structure (i.e., resistance to salt-n-pepper noise). More specifically, it is a weighted sum of traditional information gain (IG) and NSAR over all samples. Formally, $SIG = (1-\alpha)IG + \alpha\overline{NSAR}$, where α is a balancing parameter and can be learned from a validation set (more details are in *Evaluation* section).

3.3 Spatial Decision Tree Learning Algorithm

The spatial decision tree learning algorithm (Algorithm 1) takes a neighborhood graph of training pixels as inputs and constructs a spatial decision tree model. Now we introduce the algorithm step by step:

Algorithm 1 SDT-Learning($\mathcal{G}$,$\mathcal{F}$,$\mathcal{C}$, α,c_0)

Input:
- $\mathcal{G}$: neighborhood graph of training pixels
- $\mathcal{F}$: feature set of neighborhood graph nodes
- $\mathcal{C}$: class label set of neighborhood graph nodes
- α: weight for autocorrelation term in SIG measure
- c_0: minimum decision tree node size

Output:
- root of a spatial decision tree model

```
if |G| < c0 or G.C in the same class then
    Create a leaf node L from G
    Return L
end if
for each feature f ∈ F do
    Sort f values of all G nodes in ascending order, as G.f
    for each distinct value δ of G.f do
        Create candidate test f ≤ δ
        Scan and split G into G1 and G2 based on f ≤ δ
        Calculate IG, NSAR
        Compute current SIG as (1 − α)IG + α NSAR
    end for
end for
Find candidate test f0 ≤ δ0 with largest SIG
Create internal node I with test f0 ≤ δ0
Split G into G1 and G2 based on f0 and δ0
I.LeftChild=SDT-Learning(G1, F, C, α, c0)
I.RightChild=SDT-Learning(G2, F, C,α, c0)
Return I
```

Steps 1-4 check the stop criteria: if all training pixels are in the same class or their number is smaller than the minimum decision tree node size, then a leaf labeled with the majority class is returned.
Steps 5-13 evaluate the SIG measure of all candidate features and possible splitting thresholds. More specifically, step 6 generates candidate test thresholds. Steps 9 and 11 evaluate each candidate feature and each threshold.
Steps 14-16 make a greedy choice. The feature and threshold with the largest SIG are selected, an internal node is created, and the neighborhood graph is then split into two subgraphs. This is the "divide" part.
Steps 17-18 call the SDT-Learning function recursively on each subgraph. This is the "conquer" part.

4. EVALUATION

We evaluated the proposed spatial decision tree learning algorithm in a case study on a real world remote sensing dataset. The goal of the case study was to answer the following questions:

- Does incorporating spatial autocorrelation effect into decision tree learning algorithm help to improve classification accuracy?
- Does incorporating spatial autocorrelation effect into decision tree learning algorithm help to reduce salt-n-pepper noise in classification?
- How may one choose α, the balancing parameter for the Spatial Information Gain interestingness measure?

4.1 Dataset and Settings

Dataset description: The real world high resolution (3m by 3m) remote sensing dataset was collected from the city of Chanhassen, Minnesota (shown in Figure 2). Explanatory features consisted of multi-temporal (2003, 2005, 2008) spectral information (e.g. R, G, B, NIR bands of aerial photos) and topographical derivatives (e.g. slope and curvature). Ground truth class labels (wetland and upland cover types) were depicted by wetland delineation field crew and trained photo interpreters. The training area was in the central east, and the test area was in the northeast.

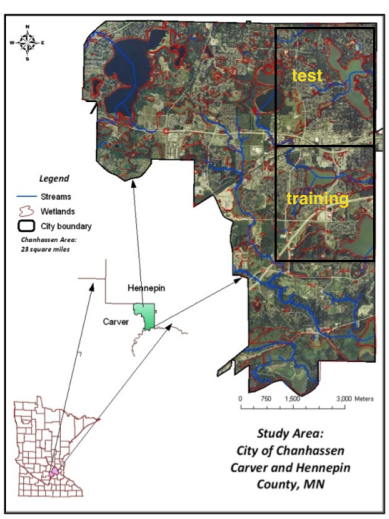


Figure 2: Study area: Chanhassen, MN. (best viewed in color) Courtesy of Lian Rampi.

Learning Samples Creation:
Training samples and validation samples were generated by systematic stratified cluster sampling. Firstly, a coarse grid was drawn over the training area and each coarse cell (98 pixel by 98 pixel) was a primary sample unit (PSU). Then, we randomly selected eight dryland PSUs and eight wetland PSUs for training, two dryland PSU and two wetland PSU for validation, identified by the majority class. Within each PSU, we further randomly selected 1000 pixels. Thus, training set consists of 16000 pixels and validation set consists of 4000 pixels. *Test samples* consists of the pixels in the entire test area.

Parameter settings: The distance threshold for neighborhood relationship between data samples was 18 meters (6 pixels). Neighbor-wise tree node test ratio θ was 0.6. Minimum tree node size c_0 was 100 pixels. The balancing parameter α in SIG was 0.26 (details are in subsection 4.4).

Evaluation candidates:
(1)the traditional C4.5 decision tree learning algorithm(**DT**)
(2)the spatial decision tree learning algorithm(**SDT**)

4.2 Does incorporating spatial autocorrelation improve classification accuracy?

Classification accuracy of the two candidate algorithms on test area is shown in the confusion matrices in Table 1 and Table 2. In a confusion matrix, each row counts pixels in a class and each column counts pixels classified as a class. By comparison of bold numbers (misclassification) in the confusion matrices, we observe that SDT algorithm reduces over one fifth false wetland pixels, and over ten percents of false dryland pixels.

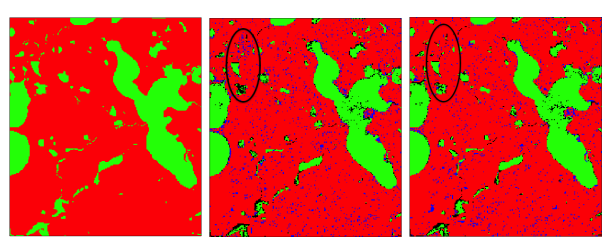
Table 1: Confusion matrix of DT

	Classified Dry	Classified Wet
Truth Dry	826,844	**47,173**
Truth Wet	**26,385**	188,814

Table 2: Confusion matrix of SDT

	Classified Dry	Classified Wet
Truth Dry	836,961	**37,056**
Truth Wet	**23,565**	191,634

Corresponding confusion maps are in Figure 3, where each color represents one cell of a confusion matrix. For example, the blue color in Figure 3(a) corresponds to the cell with value $47,173$ in the Table 1. The areas circled in Figure 3 (b) and (c) illustrate where SDT algorithm reduces false wetland pixels (in blue color) and false dryland pixels (in black color).



(a) Truth (b) DT (c) SDT

Figure 3: Confusion maps and ground truth: green for true wetland, red for true dryland, blue for false wetland, black for false dryland (best viewed in color)

4.3 Does incorporating spatial autocorrelation reduce salt-n-pepper noise?

We evaluated the salt-n-pepper level by the BB join count[12] on wetland class. $JC = \sum_{i,j} w_{i,j} x_i x_j$, where w is a w-matrix and x is one only when the pixel is a wetland class. The join count measure is then normalized. The higher the normalized joint count, the more autocorrelated the pixels are, thus the less salt-n-pepper noise there is. From Table 3, we see the amount of salt-n-pepper noise is reduced in the spatial decision tree learning algorithms.

Table 3: BB Join Count

	DT	SDT
Observed JC	404,454	402,586
Expected μ	102,158	95,938.2
Variance σ^2	1.24e+08	1.16e+08
$(JC - \mu)/\sigma$	**27.18**	**28.45**

4.4 How may one choose α, the balancing parameter for SIG interestingness measure?

This subsection describes how to automatically choose the value of α, the most important parameter in the SDT learning algorithm. The approach is to learn α via the independent validation set drawn from training area. More specifically, we first fix the other parameters as described in parameter setting part. Then, under each α value in a candidate set (i.e. 0.02, 0.04, 0.06, to 1.00), a spatial decision tree was learned from the training samples and then evaluated on validation samples. Validation errors with different α values were plotted in a curve, as indicated by Figure 4. Then the α value with minimum validation error, i.e., $\alpha = 0.26$, was selected. This method is also applicable to other study areas.

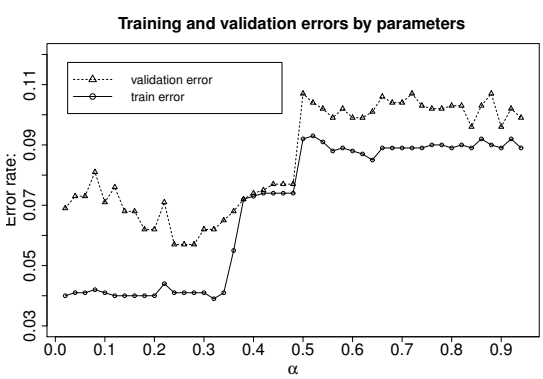


Figure 4: Learning balancing parameter α

5. CONCLUSIONS AND FUTURE WORK

This paper proposes a spatial decision tree learning algorithm for geographical classification problem. The problem is important in many applications but is challenging due to the presence of spatial autocorrelation. Results of a case study show that the proposed spatial decision tree learning algorithm outperforms traditional counterpart in not only higher classification accuracy but also less salt-n-pepper noise.

D. Stojanova et al.[13] propose a definition of spatial entropy based on global and local autocorrelation in a predictive cluster tree. The difference is that their approach is designed for geographical regression.

In future work, we plan to test the sensitivity of other parameters and conduct accuracy evaluation on multiple datasets. We may also compare our spatial decision tree learning algorithm with other relevant techniques in geographical classification, e.g., GEOBIA (Geographical Object Based Image Analysis)[6], pre-processing, and post processing.

6. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1029711, III-CXT IIS-0713214, IGERT DGE-0504195, CRI:IAD CNS-0708604, and USDOD under Grant No. HM1582-08-1-0017, HM1582-07-1-2035, and W9132V-09-C-0009. We would like to thank Kim Koffolt and the members of the Spatial Database and Spatial Data Mining research group at the University of Minnesota for their helpful comments.

7. REFERENCES

[1] L. Anselin. Local indicators of spatial association-lisa. *Geographical analysis*, 27(2):93–115, 1995.

[2] A. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[3] R. Brooks, D. Wardrop, and C. Cole. Inventorying and monitoring wetland condition and restoration potential on a watershed basis with examples from spring creek watershed, pennsylvania, usa. *Environmental Management*, 38(4):673–687, 2006.

[4] M. Celebi, H. Kingravi, Y. Aslandogan, and W. Stoecker. Detection of blue-white veil areas in dermoscopy images using machine learning techniques. In *Proc. of SPIE Vol*, volume 6144, pages 61445T–1. Citeseer, 2006.

[5] M. A. Friedl and C. E. Brodley. Decision tree classification of land cover from remotely sensed data. *Remote Sensing of Environment*, 61(3):399–409, 1997.

[6] G. Hay and G. Castilla. Geographic object-based image analysis (geobia): A new name for a new discipline. *Object-based image analysis*, pages 75–89, 2008.

[7] R. Hearne. Evolving water management institutions in the red river basin. *Environmental Management*, 40(6):842–852, 2007.

[8] X. Li and C. Claramunt. A spatial Entropy-Based decision tree for classification of geographical information. *Transactions in GIS*, 10(3):451–467, 2006.

[9] J. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[10] J. Quinlan. *C4. 5: programs for machine learning*. Morgan kaufmann, 1993.

[11] B. Ripley. Classification and regression trees. *R package version*, pages 1–0, 2005.

[12] O. Schabenberger and C. Gotway. *Statistical methods for spatial data analysis*, volume 64. CRC Press, 2005.

[13] D. Stojanova, M. Ceci, A. Appice, D. Malerba, and S. Džeroski. Global and local spatial autocorrelation in predictive clustering trees. In *Discovery Science*, pages 307–322. Springer, 2011.

[14] B. Walsh. How wetlands worsen climate change. *Time*, Jan. 2010.