Computers & Geosciences 35 (2009) 1827–1836

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

VDM-RS: A visual data mining system for exploring and classifying remotely sensed images

Jianting Zhang [a,*], Le Gruenwald [b,1], Michael Gertz [c,2]

[a] Department of Computer Science, The City College of the City University of New York, New York, NY 10031, USA
[b] School of Computer Science, The University of Oklahoma, Norman, OK 73019, USA
[c] Institute of Computer Science, University of Heidelberg, Heidelberg, Germany

ARTICLE INFO

Article history:
Received 3 April 2008
Received in revised form
29 January 2009
Accepted 20 February 2009

Keywords:
Data mining
Visualization
Remote sensing
Image classification
Decision tree

ABSTRACT

Remotely sensed imagery has become increasingly important in several applications domains, such as environmental monitoring, change detection, fire risk mapping and land use, to name only a few. Several advanced image classification techniques have been developed to analyze such imagery and in particular to improve the accuracy of classifying images in the context of such applications. However, most of the proposed classifiers remain a black box to users, leaving them with little to no means to explore and thus further improve the classification process, in particular for misclassified pixel samples. In this paper, we present the concepts, design and implementation of VDM-RS, a visual data mining system for classifying remotely sensed images and exploring image classification processes. The system provides users with two classes of components. First, visual components are offered that are specific to classifying remotely sensed images and provide traditional interfaces, such as a map view and an error matrix view. Second, the decision tree classifier view provides users with the functionality to trace and explore the classification process of individual pixel samples. This feature allows users to inspect how a sample has been correctly classified using the classifier, but more importantly, it also allows for a detailed exploration of the steps in which a sample has been misclassified. The integration of these features into a coherent, user-friendly system not only helps users in getting more insights into the data, but also to better understand and subsequently improve a classifier for remotely sensed images. We demonstrate the functionality of the system's components and their interaction for classifying imagery using a hyperspectral image dataset.

1. Introduction

It has been more than 35 years since the first multispectral imagery became available from civilian remote-sensing satellites in the early 1970s. Since then, advancements in remote-sensing technology and instrumentation have generated huge amounts of remotely sensed imagery from air- and spaceborne sensors. For example, NASA's Landsat 1–5 satellite series has resulted in a 165 TB archive of remotely sensed data over 31 years (Durbha and King, 2005). In recent years, numerous remote-sensing platforms for Earth observation with increasing spatial, temporal and spectral resolutions have been deployed by NASA, NOAA and the private sector. It has been estimated that remotely sensed imagery is acquired at the rate of several terabytes per day (Li and Bretschneider, 2007). As an example, NASA Landsat-7 has produced 269 TB images in four years since it was launched in 1999 (Durbha and King, 2005).

Processing, analyzing, and in particular classifying remotely sensed imagery has become imperative in many applications domains, where timely and accurate image classification is essential to decision making processes. This especially includes applications in environmental monitoring, such as wildfire detection and tracking environmental phenomena. The classification of remotely sensed images has long attracted the attention of the remote-sensing community (Lu and Weng, 2007). Over the past decades, several sophisticated techniques and algorithms have been developed to improve the classification accuracy for different types of applications. However, in most of the proposed approaches, image classifiers remain a black box to users, leaving them with little to no means to explore, and thus improve the classification process, particularly for misclassified samples.

In processing remotely sensed images, there are typically a number of band images (or bands for short) in an image dataset. A portion of the pixels in the image dataset are manually assigned class labels to be used as the training and testing samples. A pixel

* Corresponding author. Tel.: +1 212 650 6175.
E-mail addresses: jzhang@cs.ccny.cuny.edu, zjtnssl@yahoo.com (J. Zhang), ggruenwald@ou.edu (L. Gruenwald), gertz@informatik.uni-heidelberg.de (M. Gertz).
[1] Tel.: +1 405 325 3498.
[2] Tel.: +49 6221 545711.

0098-3004/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cageo.2009.02.006

sample (or sample for short), thus, consists of a number of values of the bands and a class label. A classifier accepts a certain number of training samples and outputs different measurements of accuracy before applying the classifier to the whole image dataset. An error matrix (or confusion table), whose elements express the numbers of sample units assigned to a particular class relative to the actual class, may provide useful information to understand the distribution of samples that are correctly or incorrectly classified. However, to the best of our knowledge, few attempts have been made to look into why and how each individual sample is correctly or incorrectly classified in the context of classifying remotely sensed images. Although a few commercial remote-sensing processing products (e.g., ENVI from ITT and ERDAS from Leica GeoSystems) provide visualization functions to display images, auxiliary data and classification results, they do not offer the functionality to visualize how an individual sample is classified by their built-in classifiers. A recent study shows that despite the considerable innovations in establishing and testing new classification methods, there has been no demonstrable improvement in classification performance over the past 15 years (Wilkinson, 2005). The lack of proper visualization tools that help exploring the structure of classifiers and in particular the classification processes of individual samples or groups of samples to gain better insights of both data and classifiers is likely the major bottleneck in solving these problems.

In this paper, we present the concepts and realization of a visual data mining system called VDM-RS we have developed to demonstrate how visualization techniques can be integrated into the classification processes of remotely sensed images. The system helps users trace the classification process of individual samples, i.e., how a sample is correctly classified according to a specific classifier. More importantly, it provides users with the functionality to explore classification steps that led to a misclassification of a sample, thus allowing users to test and evaluate local changes to the classification process and tailor the classifier to a specific image dataset. The prototype system is built on top of the WEKA open source data mining package (Witten and Frank, 2000) and the Prefuse graph visualization package (Jeffrey et al., 2005). It adopts a Coordinated Multiple Views (CMV) approach to integrate different visualization components for user interactions.

The rest of the paper is organized as follows. Section 2 discusses related work in the areas of visual data mining and classification of remotely sensed imagery. Section 3 presents the VDM-RS system architecture and discusses its components. Section 4 details the key concepts and implementation details of the visualization of a decision tree classifier and its coordination with other components of the VDM-RS system. In Section 5, experiments demonstrating the functionality of system are presented. Section 6 summarizes our approach and outlines future directions.

2. Background

The design of the proposed VDM-RS system is motivated by techniques in related fields such as exploratory geovisualization, visual data mining and coordinated multiple views. Exploratory geovisualization techniques are widely used in exploratory geospatial data analysis (Koua et al., 2006; Guo et al., 2006). The techniques typically apply unsupervised clustering algorithms (e.g., self-organizing maps) to generate clusters and then link the clusters with geographical maps to explore the relationships between clusters and spatial distributions of the sample images. While each cluster may be treated as a class, normally these clusters do not have one-to-one correspondences to the observed class labels (ground truth or ground reference) of the samples. Exploratory geovisualization techniques are also often applied to data managed in Geographical Information System (GIS). The focus of the systems built on top of a GIS is mainly on exploratory analysis but not on building classifiers and measuring classification accuracy in a rigorous manner. Under the framework of integrating information visualization and geospatial analysis, a number of works have been reported in visualizing remotely sensed data and their classification results, such as the ADaM tool (Rushing et al., 2005), visualizing class clusters in the 3D feature space plot based on a fuzzy classifier (Lucieer, 2004) and visualizing class distributions in comparison with the corresponding Gaussian distribution assumptions for individual band or pairs of bands based on the Maximum Likelihood Classifier (MLC) (Dai, 2005).

Supervised classification techniques of samples derived from remotely sensed images share many commonalities with data mining approaches. Quite a few unconventional classification algorithms are widely used in classifying remotely sensed imagery (Lu and Weng, 2007), e.g., decision trees and K-Nearest Neighbor (KNN). Over the past few years, there has been an increasing interest in research on visual data mining, e.g., Keim (2002) and De Oliveira and Levkowitz (2003). Ideally, a visual data mining system not only should have the capability to visualize both the data to be classified and the classification results, but also should provide functions in support of investigating the data mining processes to help users interact with the classifier. Ankerst (2001) distinguishes among three categories of visual data mining approaches: (1) applying visualization techniques independent of data mining algorithms, (2) using visualization techniques for data mining results and (3) tightly integrating visualization and data mining algorithms, so that intermediate steps of a data mining algorithm can be visualized. Most of the existing works on visual data mining fall into the first two categories, e.g., Barlow and Neville (2001) and Schulz et al. (2006). A very few systems truly support the features suggested for the third category. In addition, although there are some visual data mining packages for relational databases, e.g., DEVise from University of Wisconsin-Madison (Livny et al., 1997), only a very few of them support the analysis of scientific data such as remotely sensed imagery.

The concept of coordinated multiple views is a powerful visualization technique. It is used in some exploratory geovisualization and visual data mining packages, such as GeoVista (Takatsuka and Gahegan, 2002; Gahegan et al., 2002) and XmdvTool (Rundensteiner et al., 2002). Multiple views provide the opportunity to visualize different aspects of the dataset being studied, and the coordination may reveal new relationships in the data that might remain hidden otherwise (Boukhelifa and Rodgers, 2003). For example, suppose a view displaying an error matrix and another view displaying the training samples. If these two views are coordinated by highlighting all the samples corresponding to a particular element in the error matrix, i.e., the samples of class i that have been incorrectly classified as class j, one might find that these samples may be spectrally very similar to the samples of class j. If the sample view is further coordinated with a decision tree classifier, one even might find that these class i samples follow similar classification paths as some of the class j samples that lead to their incorrect classification. One can also link these samples to a map view to examine the geographical locations of the samples. If their locations are indeed close to some class j samples, one might need to check whether there have been some sampling errors.

3. Architecture and components of the VDM-RS system

The design of VDM-RS aims at providing an integration framework that tightly couples data mining algorithms with

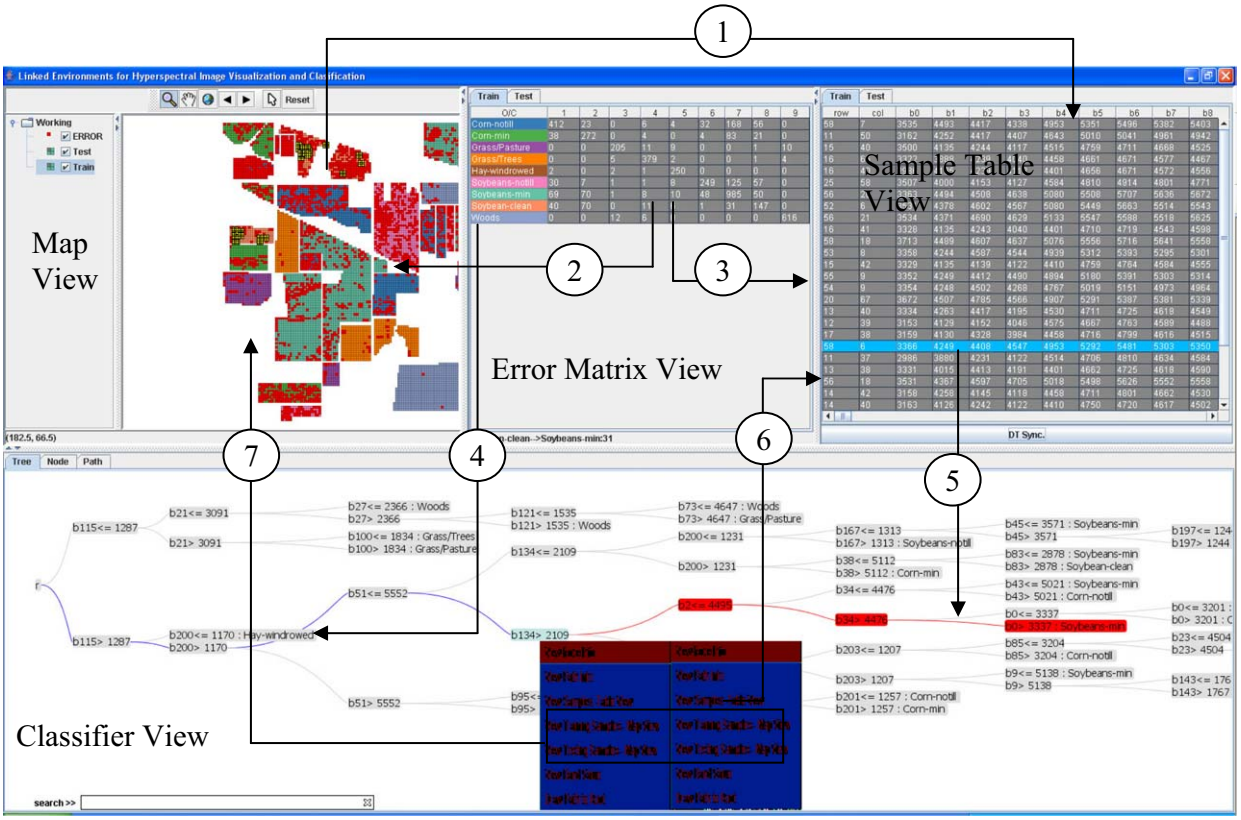

Fig. 1. Architecture, view components and coordination flows of VDM-RS. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

visualization techniques for classifying remotely sensed images. To this end, we identify intermediate results of a data mining algorithm that are important for users to understand the data mining process and use state-of-the-art visualization techniques to visualize such results and processes. Finally, we develop coordination techniques to link the multiple visualization components and provide users with a dynamic and visual environment for analyzing remotely sensed imagery. The current implementation of VDM-RS has two main components: the views and the coordination among views. A snapshot of the prototype system showing the visual components (views) and their coordination is presented in Fig. 1.

3.1. Views

VDM-RS includes four views, namely the *Map View*, the *Error Matrix View*, the *Sample Table View* and the *Classifier View*. The *Map View* displays the spatial locations of the training and testing samples. The samples are color-coded according to their class labels. The view provides basic GIS functions, such as zoom in, zoom out, zoom to full extent and to select samples interactively. After the underlying classifier is built, the system puts all the incorrectly classified samples (from both the training and the testing samples) into a separate layer (the ERROR layer colored in red in Fig. 1) and brings it to the top of the layer stack for highlighting purposes. This also gives users an opportunity to examine the spatial distribution patterns of the incorrectly classified samples and examine whether there are systematic errors. Users can interactively select samples from the *Map View*, and the samples will be added to the *Sample Table View* for further operations (Coordination 1). VDM-RS automatically detects whether the selected samples are the training or testing samples, and then puts them into the corresponding components in the *Sample Table View*.

The *Error Matrix View* displays the error matrices (also called confusion tables) for both training and testing samples. Different from static error matrices that are output by conventional systems for image processing, the class labels and the cells in the error matrices are coordinated with the *Map View*, the *Sample Table View* and the *Classifier View*. Specifically, the first columns of the matrices showing the class labels are colored using the same color table as in the *Map View*. When a cell of the error matrices is selected, the samples corresponding to the matrix cell will be highlighted in the *Map View* (Coordination 2) and displayed in the *Sample Table View* (Coordination 3). A status bar is also provided to help users keep track of the most recently selected matrix element by showing the observed class label and the classified class label as well as the number of samples covered by the element of the error matrix. When a class label in the first column of the error matrix is chosen, the classes in the visual components in the *Classifier View* (which will be described shortly in this section) will be highlighted or colored (Coordination 4).

The *Sample Table View* provides a tabular display of the training and the testing samples with each row for a sample and each column for an image band. The table displayed in the view is sortable in the sense that the samples can be sorted based on the values of one or more of the table's columns. Users can further select all or a subset of the samples in the *Sample Table View*, which can be either the training samples or the testing samples, depending on which component (implemented as a tab page in the system) is active. When a synchronization action is initiated from the *Sample Table View*, the selected samples will be sent to the classifier and visualized in the *Classifier View* (Coordination 5),

as described below. VDM-RS also allows users to manually change the band values of samples to form new samples and to test the sensitivity of the classifier used in the *Classifier View*, which is introduced next. The function is very useful to answer "what-if" types of questions.

The *Classifier View* is the most important component in the system. There are three major functions provided by this component. First, the *Classifier View* visualizes the structure of the classifier to provide users a general idea of the structure and complexity of the underlying classifier in the view. VDM-RS supports typical view operations, such as zoom in/out and pan, to allow users explore parts of the classifier of interest. Second, the *Classifier View* also accepts a sample or a set of samples from the *Sample Table View* and visualizes how the individual samples are being classified. Finally, the *Sample Table View* coordinates the visual components representing the summarization or intermediate results of the underlying classifier with the other views in VDM-RS for a better understanding of the existing classifier or to stimulate the formulation of a hypothesis that may potentially lead to the improvements of the classifier. While we defer the discussion of classifier-specific operations, such as operations for nodes and paths in a tree-like classifier, to Section 4, the *Classifier View* links the training and testing samples that fall within a subspace of the whole classification space with both the *Sample Table View* (Coordination 6) and the *Map View* (Coordination 7) for different representations. VDM-RS currently supports the Decision Tree (DT) classifier, which is being used in many applications for classifying remotely sensed imagery (see the next section for details).

In this study, we use an open source GIS package called JUMP[3] as an embedded GIS to implement the *Map View*. The *Error Matrix View* and the *Sample Table View* are implemented using java swing, which is included in JDK distributions. Compared to the *Map View*, the *Error Matrix View* and the *Sample Table View*, implementing a *Classifier View* is more difficult. Details of the implementation of the DT *Classifier View* are provided in Section 4.

3.2. Coordination

The design of the prototype follows the "Context+Focus" visualization principle, i.e., allowing users to focus on some detail without losing the context (Herman et al., 2000). In the prototype, the *Map View* and the *Error Matrix View* are the two representations of the whole sample dataset, one in geographical space and one in classification space. The *Map View* can serve as the context of the samples corresponding to an element in an error matrix. The *Sample Table View* serves as the focus of the *Map View* and the *Error Matrix View*, where the detailed information (band values and class labels) of the selected subset of samples (either from the *Map View* or from the *Error Matrix View*) are displayed. The samples displayed in the *Sample Table View* and one or multiple samples selected from the table can serve as another level of "Context+Focus". The relationship between the selected samples in the *Sample Table View* and the classifier in the *Classifier View* may be treated as the reverse of "Context+Focus", i.e., from focus to context. In this case, the trace of the classification process is the focus while the visual representation of the classifier is the context. In the context of the DT *Classifier View*, the view itself can also be treated as the context of the nodes, paths and sub-trees of the resulting decision tree displayed in the view. Unlike the *Map View* and the *Error Matrix View*, where samples are mutually exclusive and collectively exhaustive (in geographical space and classification space, respectively), the samples associated with the parent decision tree nodes include all the samples associated with their child nodes. In addition, in a DT *Classifier View*, the leaf nodes are labeled by their classes. Thus, a subset of the classes of interests can serve as the "Focus" and use the whole decision tree as the "Context". The "Focus" and "Context" in each of the four views are listed in Table 1 for comparison purposes. The values of the "type" column in Table 1 are determined as follows. In the *Error Matrix View*, only a single row or matrix element is allowed to be selected as the focus and coordinated with other views. Similarly in the DT *Classifier View*, only a single node can be chosen. These two views are categorized as "single". In contrast, the *Map View* and the *Sample Table View* allow users to choose one or multiple units to coordinate with other views.

Table 1
Summary of coordination characteristics of the four views in VDM-RS.

View	Context	Focus	Type
Map	Layer	Pixel	Single/multiple
Error matrix	Table	Row/cell	Single
Sample table	Table	Row	Single/multiple
DT classifier	Tree	Node/path	Single

We would like to point out that while the samples are the basic units in VDM-RS for the view coordination (as in many other visualization systems), the coordination at the class level and between a class in one view and samples in another view (and vice versa) is unique to VDM-RS. Among the seven coordination flows shown in Fig. 1, Coordination flows 1, 5, 6 and 7 are at the sample level and Coordination 4 is at the class level. Coordination flows 2 and 3 are from class level to sample level while Coordination 5 is from sample level to class level. In addition, the total of samples in the off-diagonal elements in the *Error Matrix View* and the pixels in the ERROR layer of the *Map View* are the two representations of the same subset of the samples. The diverse presentations of samples and coordination flows make VDM-RS capable of exploring patterns and correlations between samples and class labels that are central to remotely sensed image classification.

4. Realization of the decision tree classifier view

In this section, we first introduce the motivation for and the design of a DT *Classifier View* in VDM-RS. We then provide technical details on the implementation along with a presentation of the functionality. Finally, we briefly discuss the relationship of our design and implementation with existing work on decision tree visualization approaches.

4.1. Motivation and design

Decision tree is a popular type of classifier for classifying remotely sensed images (Friedl and Brodley, 1997; De Fries and Chan, 2000; Friedl et al., 2002; de Colstoun and Walthall, 2006) for a number of reasons. First of all, unlike many other classifiers (such as the maximum likelihood classifier) that assume samples in a training dataset to obey a certain distribution, there is no presumption of the data distribution in DT. Second, since DT adopts a divide-and-conquer strategy, it is fast in training and execution. Most importantly, the resulting classification rules are presented in the form of a tree. Paths from the root to leaf nodes can easily be transformed into decision rules (such as if $a > 10$ and $b < 20$ then Class 3), which is a suitable representation for human interpretation and evaluation.

[3] Vividsolutions, Unified Mapping Platform (JUMP), http://www.vividsolutions.com/jump/.

The goals of the DT classifier view in VDM-RS are threefold. In general, they aim at opening the black box of a classifier in traditional systems for classifying remotely sensed imagery. First, since a DT classifier partitions the data space into mutually exclusive and collectively exhaustive subspaces, a sample should follow a path in the decision tree when being classified. Thus, the primary goal of the DT *Classifier View* is to visualize the classification process of an individual sample as a path from the root to a leaf node in the decision tree. Second, when a sample is misclassified, the decision tree node at which the sample might have been misclassified should be identified. Third, important information regarding the construction of the decision tree for a classifier should be visualized properly, and thus should help users to understand the underlying classification process and algorithm. The visualization components in the DT *Classifier View* should provide capabilities to interact with users and to coordinate with other views. We next show how these design goals and objectives are realized in VDM-RS.

4.2. Implementation details

In our approach, we use the J48 implementation of the C4.5 decision tree classifier (Quinlan, 1993) from the WEKA open source data mining package (Witten and Frank, 2000). Although WEKA comes with a decision tree visualization component, this component is primarily designed for visualizing the structure of the resulting decision tree and its functionality is very limited. To meet the above design goals, a natural choice is to use a tree/graph visualization package with sophisticated layout algorithms to deal with large decision trees. In VDM-RS, we use an open source Java package called Prefuse (Jeffrey et al., 2005) for this purpose. There were a few technical issues that needed to be addressed before mapping the decision tree structure resulting from the J48 algorithm to the Prefuse tree structure. First, important key properties of the J48 decision tree classifier are designed to be non-public but they contain critical information for users to look into the classifier. Parsing this information through the summary API that returns limited information of the classifier as a string, as WEKA's built-in decision tree visualization component does, is far beyond the scope of our design objectives. Second, the resulting J48 decision tree classifier, while it returns a class label for each given sample, does not provide intermediate information that helps users understand how the sample is correctly/incorrectly classified and from which node along the classification path the sample is incorrectly classified. Third, some of the classification information that is important for users to understand the classifier does not exist in the native J48 classifier and needs to be derived.

To overcome these problems, we have developed a wrapper on top of the WEKA native J48 classifier to interact with Prefuse tree/graph visualization modules. The wrapper has the same structure as the resulting J48 decision tree with additional information fields in its nodes for visualization purposes and interactions with other views. The wrapper constructs a modified J48 decision tree classifier that allows it to expose its hierarchy and node-split models to other modules. A subset of the properties (e.g., information gain and gain ratio) associated with J48 decision tree nodes is adopted in the wrapper, and another set of properties that we believe are important to remote-sensing classification are also added to the wrapper. In particular, the error matrices for both the training and testing samples are associated with each node of the wrapper. Both the training and testing samples are then fed into the resulting J48 classifier to identify the paths along which the samples are classified. The identifiers of the samples are recorded in the wrapper and the error matrices of the leaf nodes in the wrapper are updated during the process. Finally, the wrapper is mapped to a Prefuse tree structure for visualization purposes. For memory efficiency purposes, the error matrices associated with non-leaf nodes in the wrapper are not stored. Instead, they are computed on the fly by accumulating the corresponding matrix elements from the error matrices associated with the leaf nodes under a selected non-leaf node.

As indicated previously, identifying the split nodes in the decision tree that lead to an incorrect classification of a sample is important to open the black box of the decision tree classifier. By testing whether the true class label of the sample being classified is in the set of the class labels corresponding to the leaf nodes under an intermediate decision node, the prototype system is able to determine from which decision node the sample is being incorrectly classified. For example, in Fig. 2, the highlighted path from the root to a leaf node in the decision tree shows how a sample is classified. Starting from the node $b100 <= 1834$ along the path, the true class label "Grass/Trees" cannot be found in any of its leaf nodes under the decision node $b100 <= 1834$. Here attribute $b100$ refers to band 100 and 1834 is the pixel value at the band. Splitting the path and coloring the two segments differently (in blue and red, respectively) provides users with useful information about how a sample is incorrectly classified and starting from which node in the tree the sample is incorrectly classified.

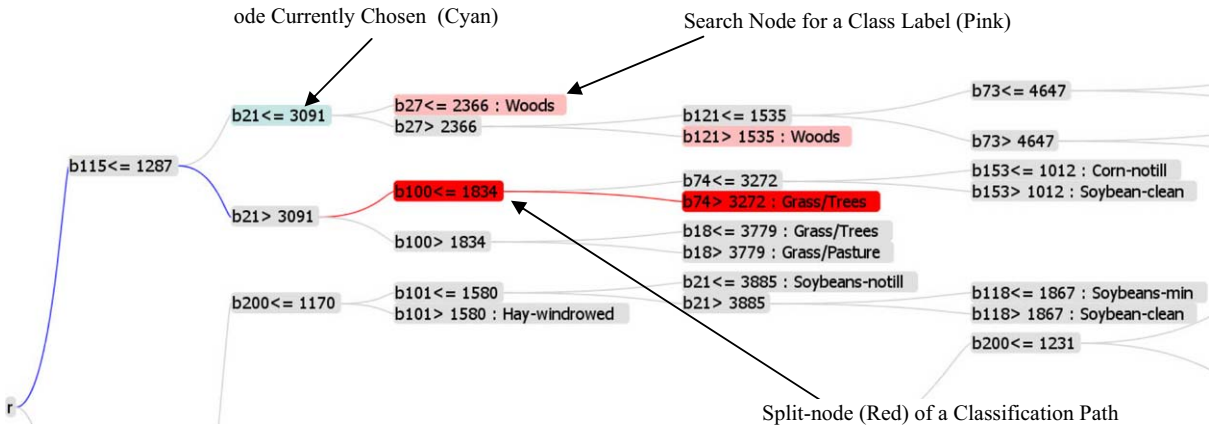

Fig. 2. Visualizing classification path of an individual sample in VDM-RS and identifying breaching point for an incorrectly classified sample. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

To help users gain insights into how training samples are divided into groups while a decision tree is being constructed, the following three types of information are associated with each decision tree node. First, the information gain and gain ratio as well as the error matrices are managed. In the decision tree, the information gain parameter measures the change of information entropy before and after partitioning a dataset into two or more subsets. The gain ratio parameter is the ratio between the information gain and the information entropy after the partitioning. These two values indicate how well the chosen partitioning attribute and its partitioning value are in separating the training samples. By comparing the error matrices associated with the node and the global error matrices in the *Error Matrices View*, normally users will recognize that the partitioned subspaces represented by the decision tree nodes are gradually specialized in distinguishing certain classes. If a small number of elements in an error matrix for a decision tree node are diagonally concentrated, compared to the global error matrix for either training or testing, it indicates that the subspace represented by the tree node is good in separating the classes from the rest.

Second, the information gains and gain ratios for all attributes of the nodes along the path from the root to a chosen node are managed. Note that the values are computed using the training samples associated with the respective nodes along the path. The reason for exposing this information to users is to help them understand how the partitioning attributes are chosen and whether there are good alternatives in choosing the partitioning attributes. This is better illustrated in Fig. 3. In this particular example, by sorting the "Full" column in Fig. 3(a), we can see that attribute b115 has the largest gain ratio value (before/) with the corresponding information gain (after/) greater than the average of all the attributes, and therefore b115 is chosen as the partitioning attribute (the partitioning value is 1287). Similarly, in Fig. 3(b), by sorting the "b115 > 1287" column, one can see that band b200 has the largest gain ratio value (0.8969), and it is selected as the partitioning attribute. On the other hand, users may see that the two neighboring bands, b114 and b201 in the two cases, have gain ratios and information gains similar to those of the bands b115 and b200 and they could have been used as the partitioning attributes. This may indicate that the resulting decision tree might be non-unique due to redundant attributes in the dataset. Furthermore, the function may inform users that while b200 is the best attribute in partitioning the dataset associated with node b115 > 1287 (Fig. 3(b)), it is not the best attribute for partitioning the whole dataset. The reason is that the distribution of the samples has been changed significantly after partitioning.

Finally, the samples associated with a decision tree node are managed. They can be visualized in other views through inter-view coordination. To display the samples in text format, VDM-RS reuses the *Sample Table View* (i.e., through Coordination 6 in Fig. 1) and highlights the incorrectly classified samples, so that users can easily identify these samples and trace their classification paths through Coordination 5, as illustrated in Fig. 1. Different from the regular display in the *Sample Table View*, the attributes (or bands) of the subset samples are reordered according to their path sequence from the root to the selected node. These attributes are followed by the class label attribute and the rest of the attributes in the dataset. A typical operation that users can perform on the sample table is to first sort the table based on the class labels of the samples and then sort it based on the values of attribute of interest. This way, they can see how the values of one or more bands differ among classes. Similarly, the samples can be highlighted in the *Map View* (Coordination 7 in Fig. 1).

The search function in the classifier view (at the very bottom of Fig. 1), which has been used in Coordination 4, plays an important role in gaining insights into the decision tree classifier. While the zoom/pan and expand/collapse functions allow users to explore the structure of a decision tree classifier, when the resulting decision tree is very large, it is very difficult if not impossible for users to explore all the nodes/paths of the decision tree at the same time. Alternatively, users can search for a class label, and the leaf nodes of the decision tree containing the class label will be highlighted. If searching a class label results in only a few leaf nodes, this case suggests that only a few decision rules (each rule can be derived by concatenating the labels of the nodes along the path from the root to a leaf node) are needed to cover the majority of the samples of the class. On the other hand, if a large number of leaf nodes are returned as the search result, such a case might suggest that the samples of the class have complex relationships between spectral values and class labels, and care must be taken to interpret the resulting decision rules. In addition to searching for class labels, users can also search for band (attribute) names and examine their distribution in the decision tree. If a band

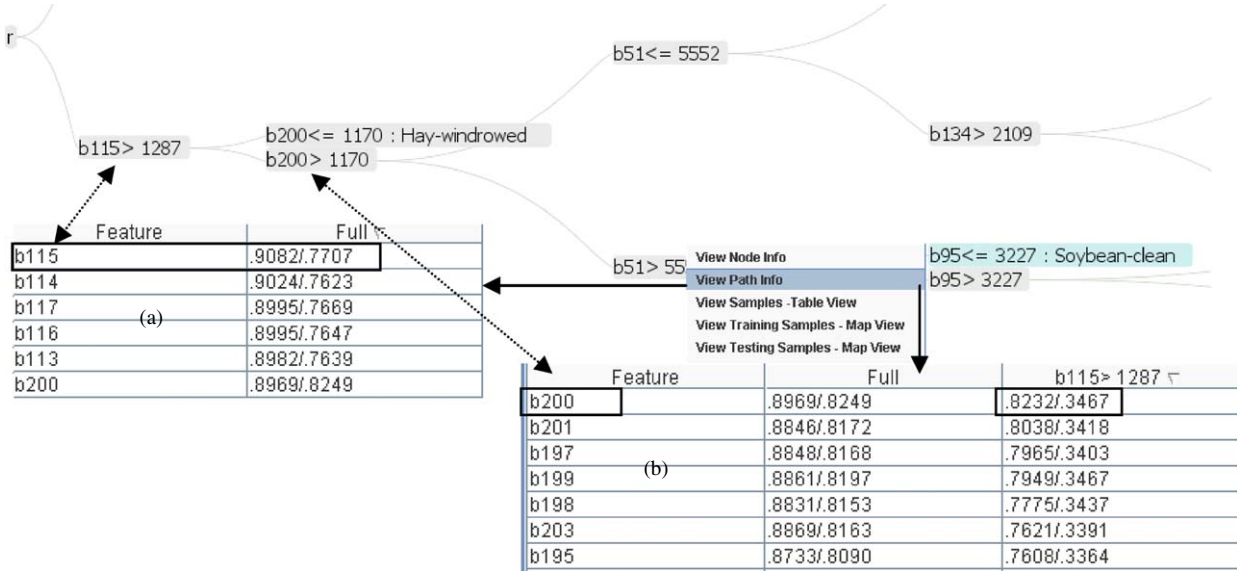

Fig. 3. Viewing and sorting information gain and gain ratio information from root to a selected node: understanding how partitioning attributes are chosen by decision tree classifier in VDM-RS. Subpanels: information gains and gain ratios for all the attributes using training samples under the root node (a) and node b115 > 1287 (b).

appears in multiple nodes and is used more frequently than other bands, this might suggest that the band plays an important role in classifying the remotely sensed image.

4.3. Discussion

A few studies related to visual aspects of decision trees have been published, and some of them focus on visual/interactive constructions of decision trees (Teoh and Ma, 2003; Liu and Salvendy, 2007). While the current implementation of the DT *Classifier View* in VDM-RS focuses on exposing the internal information of a DT classifier and coordination with other types of views, we plan to allow users to modify the automatically constructed decision trees, thus building on the experience from visual/interactive constructions of decision trees. The techniques proposed by Song et al. (2004) can be incorporated in VDM-RS to visualize large decision trees, and thus allow users to focus on multiple selections while maintaining context. The work most closely related to ours with respect to visualizing constructed decision trees is the work reported by Barlow and Neville (2001). Compared to their approach, in addition to providing more information at each decision node, the DT *Classifier View* of VDM-RS also provides information about the paths from the root to a decision node of interest (Fig. 3). Furthermore, VDM-RS is specifically designed for classifying remotely sensed images with the addition of visual components, such as *Map View* and *Error Matrix View*. In addition, different from all the above work, the VDM-RS DT classifier view allows users to visually trace the classification processes for individual samples, an aspect that has been proven very useful.

5. Experiments

5.1. Data and setup

To test the functionality of VDM-RS, we use a publicly available hyperspectral image dataset called the Indian Pine dataset (http://dynamo.ecn.purdue.edu/~biehl/MultiSpec/documentation.html). The dataset is a segment of one AVIRIS data scene taken at the NW Indiana's Indian Pine test site in 1992. The dataset representing the hyperspectral image has 145×145 pixels and about half of them have ground truths (class labels). We use nine classes of samples derived from the image. The nine classes are Corn-notill, Corn-min, Grass/Pasture, Grass/Trees, Hay-windrowed, Soybeans-notill, Soybeans-min, Soybean-clean and Woods and their sample sizes range from 497 to 2468 pixels with a total of 9345 samples. Following standard practices in classifying remotely sensed images, we randomly divide the samples into training and testing datasets. We extract the samples from their binary native dataset and transform them into the Attribute-Relation File Format (ARFF), which is supported by WEKA (Witten and Frank, 2000). The dataset or its subset has been used in quite a few previous studies (Jimenez and Landgrebe, 1999; Melgani and Bruzzone, 2004; Huang and He, 2005; Wang et al., 2007). While these studies report different classification accuracies using different classification algorithms, the causes of the misclassifications were neither analyzed nor visualized. In this section, we will demonstrate how VDM-RS can provide useful information that can help users to understand the samples and the decision tree classifier better. The number of minimum objects in a leaf tree node is set to 10 and the resulting decision tree has 126 leaf nodes. The overall classification accuracy for the testing samples is 73.60%.

5.2. Exploration of classification results

The distribution of the incorrectly classified samples (including training and testing samples) resulting from the *Map View* is shown in Fig. 4(b). For comparison purposes, the distribution of the samples in the whole dataset is also shown in Fig. 4(a). The samples are color-coded based on their class labels. From Fig. 4, one can see that the spatial distribution of the incorrectly classified samples is uneven. For example, circled regions in Fig. 4(b) have dense incorrectly classified samples. By comparing Fig. 4(a) and (b), it is easy to see that this is mostly because samples are clustered by classes in the dataset and some classes are more likely to be incorrectly classified. The observation can be further supported by the error matrices shown in Fig. 5 (resulting from the *Error Matrix View*). For example, among the 733 testing samples of class Corn-notill, 127 were misclassified as Soybeans-min. Similarly, among the 1227 testing samples of class Soybeans-min, 129 of them were misclassified as Corn-notill. In contrast, there were no misclassifications between Corn-notill and Woods and between Soybeans-min and Woods (Fig. 5(b)).

5.3. Deriving significant decision rules

The paths that lead to the classifications of Grass/Pasture, Grass/Trees, Woods and Hay-windrowed are shown in Fig. 6. The leaf nodes corresponding to the four classes are highlighted (colored in pink) and the paths from the leaf nodes to the root are

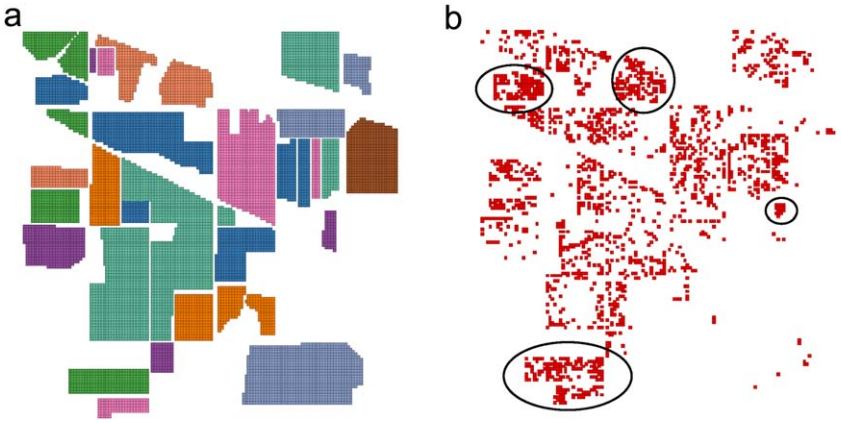

Fig. 4. Spatial distributions of all samples (a) and incorrectly classified samples (b).

a

Train | Test

O/C	1	2	3	4	5	6	7	8	9
Corn-notill	564	8	0	2	0	39	65	23	0
Corn-min	19	321	0	1	0	7	44	30	0
Grass/Pasture	1	0	205	12	5	0	2	3	8
Grass/Trees	0	1	2	384	0	2	0	0	2
Hay-windrowed	2	0	1	2	250	0	0	0	0
Soybeans-notill	16	5	1	1	0	388	53	14	0
Soybeans-min	72	22	1	7	0	42	1073	24	0
Soybean-clean	21	17	0	0	1	12	26	229	0
Woods	0	0	6	8	0	0	0	0	620

b

Train | Test

O/C	1	2	3	4	5	6	7	8	9
Corn-notill	473	31	0	1	1	54	127	46	0
Corn-min	43	249	0	0	0	8	60	52	0
Grass/Pasture	7	0	201	13	3	8	1	2	26
Grass/Trees	1	0	3	347	0	5	0	0	0
Hay-windrowed	3	0	2	1	228	0	0	0	0
Soybeans-notill	37	8	0	1	1	276	146	21	0
Soybeans-min	129	53	2	11	1	95	881	55	0
Soybean-clean	41	32	0	1	0	29	54	151	0
Woods	0	0	10	11	0	0	0	0	639

Fig. 5. Color-coded error matrices for training samples (a) and testing samples (b).

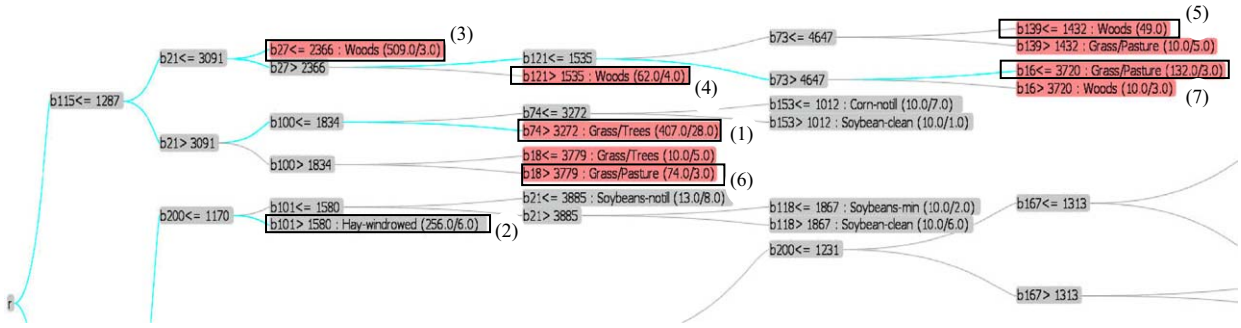

Fig. 6. Discovering significant decision rules from decision tree classifier for samples corresponding to pixel (53, 83) and pixel (52, 84). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

highlighted (colored in cyan) as well in Fig. 6. From the figure, we can easily derive the following significant decision rules:

(1) B115 $<=$ 1287 and b21 $>$ 3091 and b100 $<=$ 1834 $\rightarrow$ Grass/Trees. The rule covers 407 training samples with 28 exceptions. The number of training samples for Grass/Trees is 391 (which is the sum of numbers in the corresponding row in Fig. 5(a)).
(2) B115 $>$ 1287 and b200 $<=$ 1170 and b101 $>$ 1580 $\rightarrow$ Hay-windrowed. The rule covers 256 samples with 6 exceptions. The number of training samples for Hay-windrowed is 255.
(3) B115 $<=$ 1287 and b21 $<=$ 3091 and b27 $<=$ 2366 $\rightarrow$ Woods. The rule covers 509 samples with 3 exceptions. The number of training samples for Woods is 634.

The first and the second rules cover almost all of the samples with the corresponding classes. Besides the third rule, there are two additional significant rules for Woods with longer decision paths:

(1) B115 $<=$ 1287 and b21 $<=$ 3091 and b27 $>$ 2366 and b121 $>$ 1535 $\rightarrow$ Woods. The rule covers 62 samples with 4 exceptions
(2) B115 $<=$ 1287 and b21 $<=$ 3091 and b27 $>$ 2366 and b121 $<=$ 1535 and b73 $<=$ 4647 and b139 $<=$ 1432 $\rightarrow$ Woods. The rule covers 49 samples without exceptions.
(3) For Grass/Pasture, the number of training samples is 236. Two decision rules (rule (6) and rule (7)) can be derived as shown below. However, the rule that covers more samples has a longer decision path. The percentage of the samples covered by the significant decision rules is lower than those of the above three classes, which might indicate that the class is more difficult to classify.
(4) B115 $<=$ 1287 and b21 $>$ 3091 and b100 $>$ 1834 and b18 $>$ 3779 $\rightarrow$ Grass/Pasture. The rule covers 74 samples with 3 exceptions.
(5) B115 $<=$ 1287 and b21 $<=$ 3091 and b27 $>$ 2366 and b121 $<=$ 1535 and b73 $>$ 4647 and b16 $<=$ 3720 $\rightarrow$ Grass/Pasture. The rule covers 132 samples with 3 exceptions.

While some of the significant decision rules can be derived from the resulting decision tree for the remaining five classes, they only cover a small percentage of the training samples of the classes. We next demonstrate how VDM-RS can help users trace the decision tree classification process and understand how a sample is incorrectly classified.

5.4. Tracing classification processes of individual samples

As a demonstrative example, we have selected two neighboring pixels in the testing samples, pixel (53, 83) and pixel (52, 84). They both represent soybeans-notill instances, as determined by the ground truth. However, pixel (53, 83) was correctly classified (highlighted in green in Fig. 7) while pixel (52, 84) was incorrectly classified as soybeans-min (colored in red in Fig. 7), following two classification paths with significant overlaps. The classification paths and the pixel values of the bands used by the classification paths of the two samples are shown in Fig. 7. The values of pixel (52, 84) are given first, where pairs of values are given in the boxes in Fig. 7. The split point of the incorrectly classified path for pixel (52, 84) is node b9 $>$ 5138. Since the value of pixel (52, 84) at b73 is 4025, it follows the wrong classification path and eventually is classified as soybeans-min. Even if pixel (52, 84) takes the b73 $<=$ 3983 branch, because its value at b52 is 5247, it will be classified as soybeans-clean. Furthermore, if pixel (52, 84) takes b52 $<$ 5523 branch, it will be classified as corn-notill, because its value at b15 is 5584, which is greater than the partitioning value 5533 at b15. Finally, assuming pixel (52, 84) takes the b15 $<=$ 5533 path, it is still incorrectly classified as soybeans-min, because its value at b206 (1173) is larger than the split value of the attribute (1172). However, if one assumes a 2% measurement error, by taking the lower bounds, the new values will be b73 $=$ 3944.5, b52 $=$ 5142.06, b15 $=$ 5472.32 and

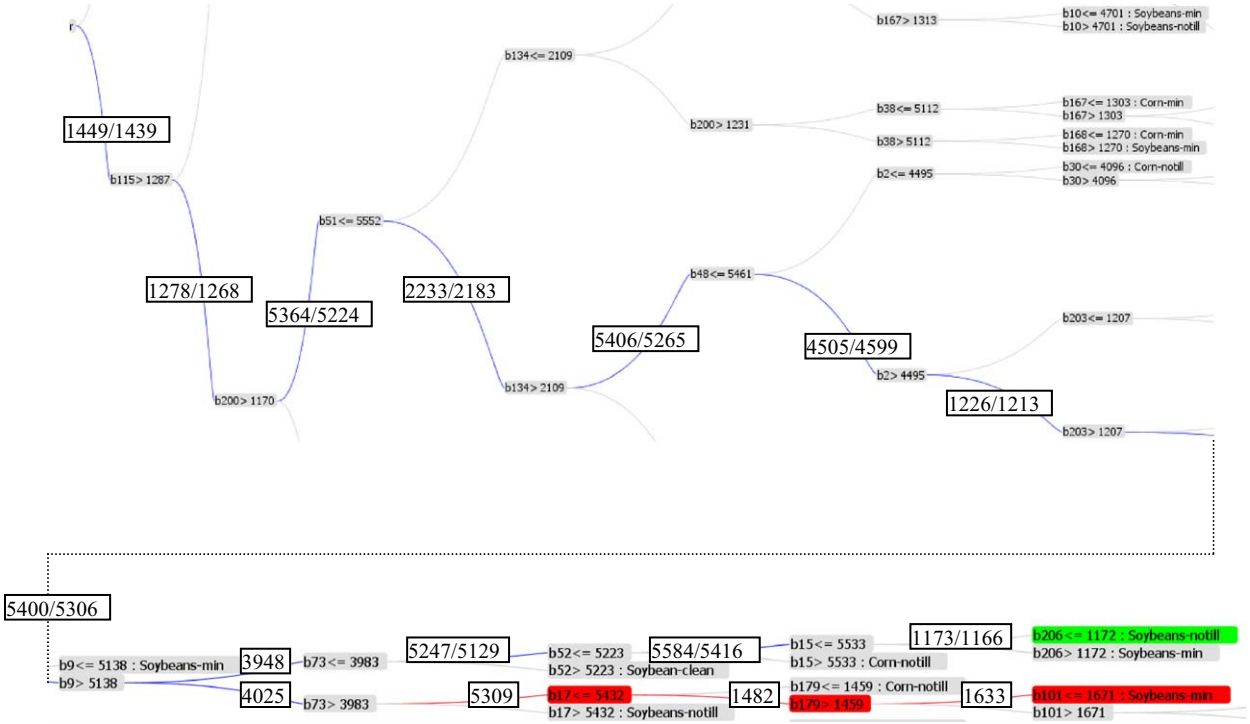

Fig. 7. Classification paths and band values along the paths for samples corresponding to pixel (53, 83) and pixel (52, 84). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

b15 = 1149.54. The new values of pixel (52, 84) will lead to the correct classification by following the path of b73 < = 3983 and b52 < = 5223 and b15 < = 5533 and b206 < = 1172. The visualization as well as the additional what-if test results suggest that the decision tree classifier is sensitive to possible measurement errors for pixel (52, 84). With a possible 2% measurement error, the pixel can be classified as any of the four classes (soybeans-min, soybeans-notill, coybeans-clean and corn-notill) under node b203 > 1207. The results also suggest that the four classes may have subtle differences and that it is difficult to distinguish them from each other, which can be verified from the error matrices shown in Fig. 5. The visualization process of the two samples also suggests that neighboring pixels that belong to the same class can have quite different band values. For example, the values of the two pixels are 5364 and 5224 at band 51, 5406 and 5256 at band 48, and 5584 and 5416 at band 15. This may affect distance-based classifiers (such as KNN) significantly and further investigations are needed.

6. Discussion and future work

In this paper, we have presented the concepts and realization of a visual data mining prototype system called VDM-RS for classifying remotely sensed images. The four views implemented in the prototype, i.e., the *Map View*, the *Error Matrix View*, the *Sample Table View* and the *Classifier View*, allow users to visualize sample datasets from different perspectives. The coordination among the views can provide useful information that is otherwise hidden. We have demonstrated the capabilities of the prototype system using a publicly available and extensively used hyperspectral image dataset.

Tightly coupling data mining algorithms and visualization techniques to build an integrated visual data mining system for classifying remotely sensed images provides new potentials for better understanding datasets and classifiers and to further improve classification accuracies. However, the advantages do not come without a price. First, not all classifiers can be easily visualized, and new visualization techniques are needed for these classifiers. Second, there is significant coding work to implement visual components of a classifier and link them with data visualization components. Third, a visual data mining system provides much more information to users, and thus requires a significant amount of user interaction. Compared to existing black-box type image classification systems, there might be a learning curve for users to adapt to such a new system.

Remotely sensed data is a special type of multi- or high-dimensional data. There are some existing techniques to visualize such data and which may be applicable to remotely sensed data, for example, the Parallel Coordinate Plot (PCP) (Edsall, 2003) techniques. We do not include PCP in VDM-RS due to some negative evaluation results from a previous study (Koua et al., 2006). However, these results should be carefully re-evaluated in the remote-sensing classification context before making a definite conclusion.

VDM-RS currently supports the decision tree-based classifiers only. We plan to investigate how other classifiers, such as the K-Nearest Neighbor, Support Vector Machines (SVM) and Maximum Likelihood Classifier can be integrated into VDM-RS. In addition, for hyperspectral image data classification, dimensionality reduction techniques are usually applied before classification. A research question that arises is why and how dimensionality reduction techniques affect classification accuracies. We plan to integrate our previous work on incorporating visualization for high-dimensional data mining processes (Zhang and Gruenwald, 2006) into VDM-RS to enhance its functionality.

References

Ankerst, M., 2001. Visual data mining with pixel-oriented visualization techniques. In: Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) Workshop on Visual Data Mining, San Francisco.

Barlow, T., Neville, P., 2001. Case study: visualization for decision tree analysis in data mining. In: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01), San Diego, CA, pp. 149–152.

Boukhelifa, N., Rodgers, P.J., 2003. A model and software system for coordinated and multiple views in exploratory visualization. Information Visualization 2, 258–269.

Dai, X., 2005. Integrated approach for the exploration of geospatial datasets: the interaction of concepts, methods and data. Ph.D. Dissertation, the Pennsylvania State University, USA, 249 pp.

De Colstoun, E.C.B., Walthall, C.L., 2006. Improving global scale land cover classifications with multi-directional POLDER data and a decision tree classifier. Remote Sensing of Environment 100, 474–485.

De Fries, R.S., Chan, J.C.W., 2000. Multiple criteria for evaluating machine learning algorithms for land cover classification from satellite data. Remote Sensing of Environment 74, 503–515.

de Oliveira, M.C.F., Levkowitz, H., 2003. From visual data exploration to visual data mining: a survey. IEEE Transactions on Visualization and Computer Graphics 9, 378–394.

Durbha, S.S., King, R.L., 2005. Semantics-enabled framework for knowledge discovery from Earth observation data archives. IEEE Transaction on Geoscience and Remote Sensing 43, 2563–2572.

Edsall, R.M., 2003. The parallel coordinate plot in action: design and use for geographic visualization. Computational Statistics & Data Analysis 43, 605–619.

Friedl, M.A., Brodley, C.E., 1997. Decision tree classification of land cover from remotely sensed data. Remote Sensing of Environment 61, 399–409.

Friedl, M.A., McIver, D.K., Hodges, J.C.F., Zhang, X.Y., Muchoney, D., Strahler, A.H., Woodcock, C.E., Gopal, S., Schneider, A., Cooper, A., Baccini, A., Gao, F., Schaaf, C., 2002. Global land cover mapping from MODIS: algorithms and early results. Remote Sensing of Environment 83, 287–302.

Gahegan, M., Takatsuka, M., Wheeler, M., Hardisty, F., 2002. Introducing GeoVISTA studio: an integrated suite of visualization and computational methods for exploration and knowledge construction in geography. Computers, Environment and Urban Systems 26, 267–292.

Guo, D.S., Chen, J., MacEachren, A.M., Liao, K., 2006. A visualization system for space-time and multivariate patterns (VIS-STAMP). IEEE Transactions on Visualization and Computer Graphics 12, 1461–1474.

Herman, I., Melancon, G., Marshall, M.S., 2000. Graph visualization and navigation in information visualization: a survey. IEEE Transactions on Visualization and Computer Graphics 6 (1), 24–43.

Huang, R., He, M.Y., 2005. Band selection based on feature weighting for classification of hyperspectral data. IEEE Geoscience and Remote Sensing Letters 2, 156–159.

Jeffrey, H., Stuart, K.C., James, A.L., 2005. Prefuse: a toolkit for interactive information visualization. In: Proceedings of the ACM Special Interest Group on Computer–Human Interaction (SIGCHI) conference on Human factors in computing systems, Portland, Oregon, USA, pp. 421–430.

Jimenez, L.O., Landgrebe, D.A., 1999. Hyperspectral data analysis and supervised feature reduction via projection pursuit. IEEE Transactions on Geoscience and Remote Sensing 37, 2653–2667.

Keim, D.A., 2002. Information visualization and visual data mining. IEEE Transactions on Visualization and Computer Graphics 8, 1–8.

Koua, E.L., Maceachren, A., Kraak, M.J., 2006. Evaluating the usability of visualization methods in an exploratory geovisualization environment. International Journal of Geographical Information Science 20, 425–448.

Li, Y.K., Bretschneider, T.R., 2007. Semantic-sensitive satellite image retrieval. IEEE Transactions on Geoscience and Remote Sensing 45, 853–860.

Liu, Y., Salvendy, G., 2007. Design and evaluation of visualization support to facilitate decision trees classification. International Journal of Human-Computer Studies 65, 95–110.

Livny, M., Ramakrishnan, R., Beyer, K., Chen, G., Donjerkovic, D., Lawande, S., Myllymaki, J., Wenger, K., 1997. DEVise: integrated querying and visual exploration of large datasets. In: Proceedings of the 1997 ACM Special Interest Group on Management of Data (SIGMOD) International Conference, Tucson, Arizona, United States, pp. 301–312.

Lu, D., Weng, Q., 2007. A survey of image classification methods and techniques for improving classification performance. International Journal of Remote Sensing 28, 823–870.

Lucieer, A., 2004. Uncertainties in segmentation and their visualisation. Ph.D. Dissertation, International Institute for Geo-Information Science and Earth Observation, Enschede, The Netherlands, 199 pp.

Melgani, F., Bruzzone, L., 2004. Classification of hyperspectral remote sensing images with support vector machines. IEEE Transactions on Geoscience and Remote Sensing 42, 1778–1790.

Quinlan, J.R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA, 302pp.

Rundensteiner, E.A., Ward, M.O., Yang, J., Doshi, P.R., 2002. XmdvTool: visual interactive data exploration and trend discovery of high-dimensional data sets. In: Proceedings of the 2002 ACM Special Interest Group on Management of Data (SIGMOD) International Conference. Wisconsin, Madison, p. 631.

Rushing, J., Ramachandran, R., Nair, U., Graves, S., Welch, R., Lin, H., 2005. ADaM: a data mining toolkit for scientists and engineers. Computers & Geosciences 31, 607–618.

Schulz, H.-J., Nocke, T., Schumann, H., 2006. A framework for visual data mining of structures. In: Proceedings of the 29th Australasian Computer Science Conference, Hobart, Australia, pp. 157–166.

Song, H., Curran, E.P., Sterritt, R., 2004. Multiple foci visualisation of large hierarchies with FlexTree. Information Visualization 3, 19–35.

Takatsuka, M., Gahegan, M., 2002. GeoVISTA studio: a codeless visual programming environment for geoscientific data analysis and visualization. Computers & Geosciences 28, 1131–1144.

Teoh, S.T., Ma, K.-L., 2003. PaintingClass: interactive construction, visualization and exploration of decision trees. In: Proceedings of the ninth ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) International Conference, Washington, D.C., pp. 667–672.

Wang, L.G., Jia, X.P., Zhang, Y., 2007. A novel geometry-based feature-selection technique for hyperspectral imagery. IEEE Geoscience and Remote Sensing Letters 4, 171–175.

Wilkinson, G.G., 2005. Results and implications of a study of fifteen years of satellite image classification experiments. IEEE Transactions on Geoscience and Remote Sensing 43, 433–440.

Witten, I.H., Frank, E., 2000. Data Mining: Practical Machine Learning Tools with Java Implementations. Morgan Kaufmann, San Francisco, CA, 416pp.

Zhang, J., Gruenwald, L., 2006. Opening the black box of feature Extraction: incorporating visualization into high-dimensional data mining processes. In: Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM), Hong Kong, China, pp. 1188–1192.