HC-DT/SVM: A Tightly Coupled Hybrid Decision Tree and Support Vector Machines Algorithm with Application to Land Cover Change Detections

Jianting Zhang
Department of Computer Science
City College of New York
New York City, NY, 10031
jzhang@cs.ccny.cuny.edu

ABSTRACT

Change detection techniques have been widely used in satellite based environmental monitoring. Multi-date classification is an important change detection technique in remote sensing. In this study, we propose a hybrid algorithm called HC-DT/SVM, that tightly couples a Decision Tree (DT) algorithm and a Support Vector Machine (SVM) algorithm for land cover change detections. We aim at improving the interpretability of the classification results and classification accuracies simultaneously. The hybrid algorithm first constructs a DT classifier using all the training samples and then sends the samples under the ill-classified decision tree branches to a SVM classifier for further training. The ill-classified decision tree branches are linked to the SVM classifier and testing samples are classified jointly by the linked DT and SVM classifiers. Experiments using a dataset that consists of two Landsat TM scenes of southern China region show that the hybrid algorithm can significantly improve the classification accuracies of the classic DT classifier and improve its interpretability at the same time.

Categories and Subject Descriptors

H.2.8 [Database Applications] Data Mining

General Terms

Algorithms, Experimentation, Performance

Keywords

Hybrid Classifier, Decision Tree, SVM, Remote Sensing, Land Cover, Change Detection

1. INTRODUCTION

Change detection from remotely sensed images is a useful technology for detecting changes in large and rapidly changing area and is an important source for environmental monitoring. Many digital change detection techniques have been developed during the past few decades (Singh 1989, Lu et al 2004). The techniques can be grouped into three major categories: map algebra, direct multi-date classification and post-classification comparison. Image (band) differencing might be the most widely used method in the first category. While the techniques in the category are able to provide information on the possible existence of a change and the relative magnitude of the change, they do not identify the nature of the change (Im and Jensen 2005). In contrast, techniques in the later two categories have the capabilities of providing detailed information about the type of land cover change for every pixel and/or polygon under examination (Im and Jensen 2005). While the post-classification comparison based methods are straightforward, they were criticized for relying on the accuracy of the two individual classifications (Singh 1989). In this study, we propose a hybrid algorithm that tightly couples a Decision Tree (DT) algorithm and a Support Vector Machine (SVM) algorithm for land cover change detections that aims at improving the interpretability of the classification results and classification accuracies simultaneously. The proposed approach falls in the multi-date classification category.

Comparisons of different classification algorithms in the multi-date classification category have been extensively studied. For example, Chan et al (2001) compared four classifiers, namely Multi-Layer Perceptron (MLP), Learning Vector Quantization (LVQ), Decision Tree (DT) and Maximum-Likelihood Classifier (MLC). Seto and Liu (2003) compared ARTMAP neural network with MLC and observed that ARTMAP neural network classifiers were more accurate than MLC classifiers. Nemmour and Chibani (2006) has reported that Support SVM generally performed better than a two hidden-layer Artificial Neural Network (ANN) classifier using the standard back propagation rule with respect to classification accuracies. While a certain classifier may have higher classification accuracy for a particular dataset, it is hard to make a conclusion that some classifiers are always better than the rests when multiple criteria are used to evaluate the suitability of algorithms (Chan et al, 2001). Although in reality no classification algorithm can satisfy all evaluation requirements nor be applicable to all studies due to different environmental settings and datasets used (Lu and Weng 2007), hybridizing two or more classifiers with careful design may improve the suitability of classification algorithms for land cover change detections.

Hybrid classifier is a popular concept in classifying remotely sensed data. Various hybrid methods have been proposed since at least early 1990s (Kelly et al 2004). For example, the Iterative Guided Spectral Class Rejection (IGSCR) is a hybrid approach that combines unsupervised clustering and maximum likelihood (Wayman et al 2001) and have been successfully used in a few applications (Kelly et al 2004, Musy et al 2006, Wynne 2007). Techniques that hybridize clustering and classification algorithms for urban change analysis have

ACM SIGSPATIAL DMG Workshop, Nov 2, San Jose, CA, U.S.A.

been successfully applied to the Twin Cities (Minnesota) metropolitan area using multi-date Landsat data (Yuan et al 2005). In addition, the ensemble based techniques, such as bagging or boosting, can also be broadly considered as hybrid classifiers where a same base classifier is applied multiple times and the classifications are combined to generate the final results. In this case, the base classifiers are "hybridized" to themselves. Land cover classifications using boosting (Friedl et al 1999, de Colstoun and Walthall 2006) and bagging (DeFries and Chan 2000, Prasad et al 2006) on decision tree classifiers have been reported. More recently, Nemmour and Chibani (2006) applied multiple support vector machines for land cover change detection where multiple kernels were used to build multiple classifiers and the classification results were combined based on fuzzy integral and attractor dynamic rules. Most of existing hybrid classifiers require training and classifying the samples in a dataset multiple times independently and we term as loosely-coupled hybrid classifiers. The problem with such loosely-coupled hybrid classifiers are that the numbers of training and testing of the hybrid classifiers usually are proportional to the numbers of base classifiers that the hybrid classifiers are based on. Combining classification results from multiple independent classifiers beyond simple majority voting rule requires careful design of schemas of combination (e.g., Liu and Gopal 2004, Huang and Lees 2004) which is a non-trivial task. In addition, while it may be possible to visualize individual base classifiers for better interpretation, it may not be feasible to visualize the hybrid classifiers due to the composition complexities of the base classifiers.

In this study, we propose a new hybrid algorithm that adopts a tightly coupled strategy for base classifiers. The strategy first feeds all the training samples to a fast classifier that uses divide-and-conquer strategy (e.g. decision tree algorithms) and identifies ill-classified components in the divided classification space. The strategy then combines samples fall in the ill-classified components and sends them to a more sophisticated and computationally intensive classifier for further classification. A hybrid classifier that adopts the strategy actually is a chain of two types of base classifiers. One of the advantages of the new type of hybrid classifiers is the capacity to leverage fewer but more significant patterns resulting from the training samples and present them to users immediately in a compact form. For example, delivering decision rules from a resulting decision tree classifier that cover a larger number of training samples with high classification accuracies or few exceptions. In addition, the classifiers that adopt the divide-and-conquer strategy usually can be represented as a tree and can be easily visualized.

As a case study, we have developed a new hybrid algorithm that hybridizes a decision tree classifier and a SVM classifier. Different from previous hybrid techniques that mainly target at classification accuracies, the proposed hybrid algorithm also aims at interpretability of the trained classifier. We choose to hybridize the decision algorithm and the SVM algorithm for two main reasons. First, the decision tree algorithm has been widely used in land cover classification and its advantages, such as no presumption of data distribution and fast in training and execution, have been well recognized (Friedl and Brodley 1997, Friedl et al 2002). More importantly, it has the capabilities of generating human interpretable rules. The decision tree algorithm has been successfully applied to urban change detection as reported in (Chan et al 2001) and (Im and Jensen 2005). Second, recent studies on classifying remote sensing data have consistently reported that SVM classifiers have better classification accuracies than conventional MLC classifiers and ANN based classifiers for both multi-spectral (Huang et al 2002, Pal and Mather 2005) and hyperspectral images (Pal and Mather 2006) which suggests that SVM could be used as an accurate classifier for change detection that involves a large number of bands from multi-date images. Unfortunately, the resulting hyperplane in a SVM classifier is in a high-dimensional space, which makes visualizing SVM classifier for human interpretation very difficult if not impossible.

Following the strategy discussed previously, the proposed hybrid algorithm first applies a decision tree algorithm to the training samples to construct a DT classifier. The samples in the ill-classified branches of the resulting decision tree are used to construct a SVM classifier. The two classifiers are chained together through pointers and used for classification. The proposed approach is motivated by our previous work on devising a successive decision tree algorithm for classifying remotely sensed data where the samples in the ill-classified branches of a previous resulting decision tree are used to construct a successive decision tree (Zhang et al 2007). The rest of the paper is organized as follows. Section 2 introduces the basics of the DT classifier and the SVM classifier and presents the proposed hybrid algorithm. Section 3 provides details of software implementations of the HCC-DT/SVM algorithm. Section 4 is the experiments on the land cover change detections using a pair of TM images at two times in a southern China region. Finally Section 5 is the summary and conclusions.

2. The HC-DT/SVM Algorithm

Before going to the details of the hybrid algorithm, we first briefly introduce the two base classifiers, namely the decision tree classifier and the support vector machines classifier. The hybrid algorithm is then presented as a set of linked procedures.

2.1 The Decision Tree Algorithm

The decision tree method recursively partitions the data space into disjoint sections using impurity measurements (such as information gain and gain ratio). For the sake of simplicity, binary partition of feature space is usually adopted in implementations. Let $f(C_i)$ be the count of class i before the partition and $f(C_i^1)$ and $f(C_i^2)$ be the counts of class i in each of the two partitioned sections based on a partitioning value, respectively. Further let C be the total number of classes, $n = \sum_{i=1}^{C} f(C_i)$, $n_1 = \sum_{i=1}^{C} f(C_i^1)$, and $n_2 = \sum_{i=1}^{C} f(C_i^2)$, then the information entropy before the partition is defined as $e = -\sum_{i=1}^{C} \frac{f(C_i)}{n} * \log(\frac{f(C_i)}{n})$. Correspondingly the entropies of the two partitions are defined as $e_1 = -\sum_{i=1}^{C} \frac{f(C_i^1)}{n_1} * \log(\frac{f(C_i^1)}{n_1})$ and

$e_2 = -\sum_{i=1}^{C} \frac{f(C_i^2)}{n_2} * \log(\frac{f(C_i^2)}{n_2})$, respectively. The overall entropy after the partition is defined as the weighted average of e_1 and e_2, i.e.,

$$entropy\_partition = \frac{n_1}{n} * e_1 + \frac{n_2}{n} * e_2$$

The Information Gain then can be defined as:

$$entropy\_gain = e - entropy\_partition$$

The Gain Ratio is defined as:

$$gain\_ratio = \frac{entropy\_gain}{entropy\_partition}$$

Implementations may choose to use different criteria, such as information gain, gain ratio or their combinations. For example, the J48 module in the WEKA data mining package (Witten and Frank 2000) that implements the popular C4.5 algorithm uses the following procedure to determine the best partitioning attribute (band in remote sensing classification case) and the best partitioning value. First, for each attribute, a set of partitioning values is determined based on the minimum and maximum values of the attribute. Second, each of the partitioning values is used to partition the training samples into two subsets and the information gain and gain ratio are calculated. The partitioning value with the largest gain ratio among the partitioning values whose info gains are above the average is used as the attribute's partitioning value. Third, the process is repeated for all the attributes and the attribute with the largest gain ratio is chosen as the partitioning attribute.

The decision tree classifier adopts a divide-and-conquer strategy and is very fast in training and testing. More importantly, 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 suitable for human interpretation and evaluation. In addition, during the process of selecting partitioning attribute, the algorithm works on an attribute (band) at a time and do not need information from other attributes (bands). Thus band values come from multi-date images do not need rigid radiometric calibration before feeding to the decision tree algorithm. This is a significant advantage of using the algorithm for change detections that involve multi-date images when calibration is difficult.

2.2 The SVM Algorithm

For the sake of simplicity, we only introduce the basic SVM algorithm that handles two classes. For multi-class classification problem, either one against one or one against all strategy can be applied to decompose the multi-class classification problem into multiple two-class classification problems. The SVM classifier we use in this study is the Java version of the LIBSVM package (Chang and Lin 2001) which adopts the one against one class decomposition strategy. An n-class classification problem is decomposed into n*(n-1)/2 two-class classification problems. The results are merged through a majority vote.

For a two-class classification problem, given a set of samples N $\{x_i, y_i\}_{i=1}^{N}$, where $x_i \in \mathbf{R}^n$ is the *i*-th sample and $y_i \in \{-1, +1\}$ is the label of the sample, the SVM algorithm aims at finding a linear hyperplane that separate the data in a transformed space, i.e.,

$$y_i[w^T \phi(x_i) + b] \geq 1, i = 1..N$$

where function $\phi(x)$ is a mapping from the original space to a high dimensional space. In case of such separating hyperplane does not exist, a so called slack variable ξ_i is introduced such that

$$\begin{cases} y_i[w^T \phi(x_i) + b] \geq 1 - \xi_i, i = 1..N \\ \xi_i \geq 0 \end{cases} \quad (1)$$

SVM adopts the structural risk minimization principle and the risk bound is minimized by solving the following minimization problem:

$$\min_{w,\xi} J(w, \xi) = \frac{1}{2} w^T w + c \sum_{i=1}^{N} \xi_i \quad (2)$$

subjected to (1). To minimize (2), a Lagrangian function can be constructed as

$$L(w, b, \xi, \alpha, \beta) = J(w, \xi) - \sum_{i=1}^{N} \alpha_i \{y_i[w^T \phi(x_i) + b] - 1 + \xi_i\} - \sum_{i=1}^{N} \beta_i \xi_i \quad (3)$$

where $\alpha_i \geq 0,\ \beta_i \geq 0\ (i = 1, \ldots, N)$ are the Lagrangian multipliers of (2). Function (3) reaches its optimal value when the following conditions are met:

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \\ \frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0 \\ \frac{\partial L_1}{\partial \xi_i} = 0 \rightarrow c - \alpha_i - \beta_i = 0, i = 1..N \end{cases} \quad (4)$$

Substitute (4) for (3) we get the following quadratic programming problem

$$\max_{\alpha} Q(\alpha) = -\frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \sum_{i=1}^{N} \alpha_i \quad (5)$$

where $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ is called the kernel function. Solving this quadratic programming (QP) problem subject to constrains in (4), a decision hyperplane in the high dimensional space can be obtained and will be used in the subsequent classifications.

2.3 The Hybrid Algorithm

The basic idea of the hybrid algorithm is to keep classification branches of a resulting decision tree that have high classification accuracy (corresponds to a significant decision rule) while combining samples that are classified under branches with low classification accuracy into a new training dataset to use the SVM classifier. The modified decision tree classifier is responsible for constructing significant and compact decision rules for human interpretation and the SVM classifier is responsible for training the samples that do not fit in the decision rules of the resulting decision tree. By giving the ill-classified samples in the decision tree classifier a new chance in the SVM classifier, we expect the overall classification accuracy to be higher than using the decision tree classifier alone. The heuristics behind the expectation are as follows. In the decision tree classifier, there are samples in a multi-class training data set, although their patterns may be well perceived by human, they are small in sizes and are often assigned to various branches during the classification processes according to information entropy gain or gain ratio criteria. At some particular classification levels, the numbers of the samples may be below predefined thresholds in decision tree branches to be qualified as decision tree leaf nodes with high classification accuracies, thus the splitting processes stop and they are treated as noises. However, if we combine these samples into a new dataset and train a SVM classifier, since the distribution of the new dataset may be significantly different from the original one, new meaningful patterns may be discovered by the SVM classifier. The basic idea of the hybrid algorithm is illustrated in Fig. 1.

We next present the hybrid algorithm as a set of linked procedures. The overall control flow of the hybrid algorithm is shown in Fig 2. The process of building the modified decision tree classifier is shown in Fig. 3. The process of classifying a sample by the hybrid algorithm is shown in Fig. 4. Since we use a regular SVM classifier, the procedures for building a SVM classifier (Build_SVM) and classifying a sample using the SVM classifier (SVM_Classify) are omitted.

The function *Build_Tree* (Fig. 3) recursively partitions a data set into two and builds a decision tree by finding a partition attribute and its partition value based on the information gain and the gain ratio criteria as discussed previously. There are several parameters used in function *Build_Tree. Min_obj1* specifies the number of samples to determine whether the branches of a decision tree should stop or continue partitioning. *min_obj2* specifies the minimum number of samples for a branch to be qualified as having high classification accuracy. *Min_accuracy* specifies the percentage of samples of the dominating classes. While the purposes of setting *min_obj1* and *min_accuracy* are clear, the purpose of setting *min_obj2* is to prevent generating small branches with high classification accuracies in hope that the samples that fall within the branches can be used to generate more meaningful patterns in the subsequent SVM classifier.

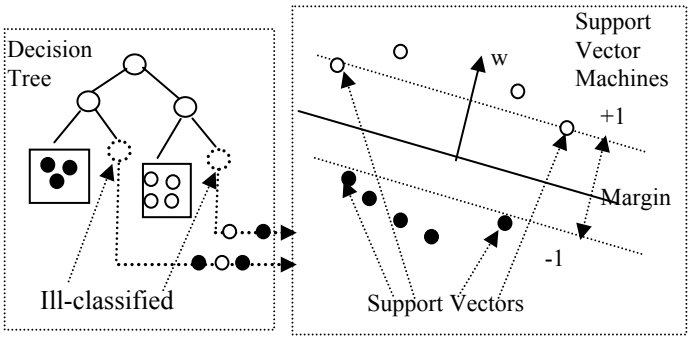

Fig. 1 Illustration of the Basic Idea of the Hybrid Algorithm

Algorithm Hybrid (P, *min_obj1, min_obj2, min_accuracy*)

Inputs:

(1)A training sample dataset (*P*) with *N* samples, each sample has *M* attributes (number of bands of the multi-date images to classify) and a class label

(2)Three thresholds for the modified decision tree algorithm: the number of samples to determine whether the branches of a DT should stop or continue partitioning (*min_obj1*), the minimum number of samples in a branch (*min_obj2*), and the percentage of the samples of classes in branches that can be considered as dominating (*min_accuracy*)

Output:

A hybrid classifier linking a decision tree classifier and a SVM classifier

1. Set dataset *D=P*, dataset *D'*={}
2. Call H.*T=Build_Tree* (*D, D',min_obj1, min_obj2, min_accuracy*)
3. Call H.V=Build_SVM(D')
4. Return H

Fig. 2 Overall Control Flow of the Hybrid Algorithm

Procedure Build_Tree (*D*, *D'*, *min_obj1*, *min_obj2*, *min_accuracy*)
Inputs:
D': new data set combining ill-classified samples
D, *min_obj1*, *min_obj2*, *min_accuracy*: same as in function *Hybrid* in Fig. 2
Output: The modified decision tree

1. Let *num_corr* be the number of samples of the dominating class in D
2. if(|*D*|< *min_obj1*)
   a. If (*num_corr*>|*D*|* *min_accuracy*) and |*D*|> *min_obj2*)
      i. Mark this branch as high accuracy branch (no need for further partitioning) and assign the label of the dominating class to the branch
      ii. Return NULL
   b. Else
      i. Mark this branch as low accuracy branch with "use_svm"
      ii. Merge *D* into *D'*
      iii. Return NULL
3. else
   a. if (*num_corr*>|*D*|* *min_accuracy*)
      i. Mark this branch as high accuracy branch (no need for further partition) and assign the label of the dominating class to the branch
      ii. Return NULL

//begin binary partition

4. For each of the attributes of *D*, find partition value using *entropy_gain* or *gain_ratio*
5. Find the partition attribute and its partition value that has largest *entropy_gain* or *gain_ratio*
6. Divide *D* into two partitions according to the partition value of the attribute, *D1* and *D2*
7. Allocate the tree structure to *T*
8. *T.left_child= Build_Tree*(*D1*, *D'*, *min_obj1*, *min_obj2, min_accuracy*)
9. *T.right_child= Build_Tree*(*D2, D'*, *min_obj1*, *min_obj2, min_accuracy*)
10. return *T*

Fig. 3 Procedure Build_Tree

If the *min_accuracy* value is set to a high percentage, many branches in the corresponding decision trees will not be able to be qualified as having high classification accuracy and samples that fall within these branches will need to be fed to the subsequent SVM classifier. On the other hand, using higher *min_accuracy* values generates decision branches that are higher in classification accuracies but smaller in numbers. For *min_obj1* and *min_obj2,* it is clear that *min_obj1* needs to be greater than *min_obj2.* The larger *min_obj1*, the earlier to check whether to further partition a decision tree branch. Once the number of samples is below *min_obj1*, the branch will be either marked as having high classification accuracy or marked as needing to be linked to the subsequent SVM classifier, depending on *min_accuracy* and *min_obj2.* A larger *min_obj1,* together with a higher *min_accuracy* makes the hybrid algorithm find larger but fewer decision branches that are high in classification accuracy (i.e., significant decision rules). The parameter *min_obj2* is more related to determining the granularity of "noises" of the decision tree. A smaller *min_obj2* means that fewer branches, the samples of which are almost of the same class (*>min_accuracy)* but are small in sizes, will be considered as unclassifiable in the decision tree classifier and need to be sent to the SVM classifier.

3. Software Implementation

We implement HC-DT/SVM on top of two Java open source data mining packages. The WEKA (Witten and Frank 1999) is a well-known general purpose data mining tool and has been successfully used in GESCONDA - an intelligent data analysis system for knowledge discovery and management in environmental databases (Gibert et al 2006). We use WEKA to provide input/output formatting and use its J48 implementation of the C4.5 algorithm as a skeleton for implementing the DT part of the HC-DT/SVM algorithm. LibSVM (Chang and Lin 2001) is a popular Java library for building SVM classifiers and a wrapper called WLSVM (WEKA LibSVM) has been provided to interface between LibSVM and WEKA (El-Manzalawy and Honavar 2005).

While these open source packages provide building blocks to implement HC-DT/SVM, the hybridization of the two algorithms and providing an integrated implementation is non-trivial for three reasons. First, the J48 code in the WEKA needs to be significantly revised to make it be aware of ill-classified branches. Second, a controlling mechanism needs to be implemented to gather training samples in the ill-classified branches and send them to a SVM classifier. Finally, a new classifier needs to be implemented to dispatch a testing sample to either the modified DT classifier or the SVM classifier and output the combined classification result.

We follow the structure of the weka.classifiers.trees.j48 package and modify the relevant components to implement HC-DT/SVM. First, in addition to *NoSplit* class that represents the leaf node in a constructed decision tree, the *NextSplit* class represents the ill-classified tree branches is added, both extend the *ClassifierSplitModel* class in the J48 package. The *C45ModelSelection* module is extended to handle the new category of decision tree nodes. The instances of the *NextSplit* class always return -1 when a sample is being

classified and thus training samples fall under the ill-classified decision tree branches can be gathered and sent to the linked SVM classifier. Similarly, the DT part of the hybrid classification algorithm returns -1 to indicate that a testing sample falls under an ill-classified decision tree branch and should use the linked SVM for final classification. Finally the hybrid classifier implements the *buildClassifier*, *classifyInstance* and *distributionForInstance* interface functions required by a WEKA classifier so that it can be used the same as other WEKA classifiers. By adding a simple component (*NextSplit)*, and slightly revising two existing components (*ClassifierSplitModel, C45ModelSelection),* our implementation of HC-DT/SVM maintains high compatibility with the J48 implementation of the C4.5 decision tree algorithm, which is desirable with respect to minimizing development cost, easy understanding and better usability.

Procedure *Hybrid_Classify*(H, *I*)
Input:
- A Hybrid Classifier
- A sample *I* with *M* attributes

Output:
- Class label of *I*

1. Set *label= DT*_classify*(H.DT,I)*
2. If label==”use_svm” return *SVM_Classify*(*H.SVM,I*)
3. Else Return *label*

Algorithm *DT_Classify* (*T*, *I*)
Input:
- A decision tree resulting from the modified decision tree classifier
- An instance *I* with *M* attributes

Output:
- Class label of *I*

1. If *T* is a leaf node
   a. If *T* is marked as a high classification confidence node
      i. Assign the class label of *T* to *I*
      ii. Return
   b. Else if *T* is marked as a low classification confidence node
      Return “use_svm”
2. Else
   a. Let *A* be the partitioning attribute and *V* be the partition value
   b. If(*I*[*A*]<=*V*) then
      Return *DT_Classify* (*T.left_child*, *I*)
   c. Else
      Return *DT_Classify* (*T.right_child*, *I*)

Fig. 4 Procedure *Hybrid_Classify*

We note that, while our implementation of HC-DT/SVM currently takes samples in WEKA’s data format only and cannot read data maintained by commercial remote sensing data processing systems, it is possible to use third party open source packages to generate training and testing samples from data managed by the commercial systems and feed the samples to HC-DT/SVM. For example, the StarSpan package developed at the Center for Spatial Technologies and Remote Sensing (CSTARS) at University of California at Davis (Rueda et al 2005). Given a set of images and Regions of Interests (ROIs), StarSpan can extract values and its label of pixels fall within the ROIs and exported them in a variety of data formats which can be further converted to the WEKA’s ARFF format.

4. Experiments

To validate the proposed hybrid algorithm, we use a dataset consists of two Landsat TM scenes of southern China acquired on 10 December 1988 (T1) and 03 March 1996 (T2). Preprocessing including geometric and atmospheric corrections of the dataset has been described elsewhere (Seto and Liu 2003). A total of 12 bands, i.e., TM bands 1-5 and band 7 for the two scenes, are used in the classification. Class labels and the numbers of training and testing samples for the classes are listed in Table 1. The six bands in the T1 image are numbered b0 through b5 and the six bands in the T2 image are numbered b7 through b11, respectively.

4.1 Tests of Accuracies

We use the following parameters in the hybrid classifier: min_obj1=200, min_obj2=100 and min_accuracy=95%. For the SVM parameters used in the hybrid classifier, we use a Radial Base Function (RBF) kernel and set G=1 and C=39 after fine tuning. The default parameters in the J48 decision tree implementation are used without fine tuning. For fair comparison, we use the same fine-tuned SVM parameters in the original SVM classifier for the hybrid classifier. The overall accuracy of the hybrid classifier is 89.87%. The classification accuracies for the original decision tree classifier and the SVM classifier are 81.25% and 90.31%, respectively. The error matrices for the three classifiers are listed in Table 2, Table 3 and Table 4, respectively. From the results we can see that the hybrid classifier has much higher accuracies than the classic DT classifier while slightly worse than the SVM classifier.

Table 1 Classes and the numbers of their training and testing samples

Class ID	Class Description	# of Training Samples	# of Testing Samples
1	Water	250	59
2	Natural vegetation	568	154
3	Agriculture	962	246
4	Urban	682	154
5	Water to Urban	544	84
6	Agriculture to Urban	1059	180
7	Vegetation to Urban	775	259
Total		4840	1136

Table 2 Error Matrix of the Hybrid Classifier (Overall Accuracy= 89.88%, Kappa= 0.8784)

	Reference Categories							Σ	UA
	1	2	3	4	5	6	7		
Water	58	0	1	0	0	0	0	59	98.31%
Natural vegetation	0	148	6	0	0	0	0	154	96.10%
Agriculture	0	31	199	1	0	12	3	246	80.89%
Urban	0	0	7	139	0	8	0	154	90.26%
Water to Urban	0	0	0	0	84	0	0	84	100.00%
Agriculture to Urban	0	0	12	3	1	154	10	180	85.56%
Vegetation to Urban	0	0	3	0	0	17	239	259	92.28%
Total								1136	

Table 3 Error Matrix of the Classic DT Classifier (Overall Accuracy=81.25%, Kappa=0.7750)

	Reference Categories							Σ	UA
	1	2	3	4	5	6	7		
Water	46	0	13	0	0	0	0	59	77.97%
Natural vegetation	0	146	8	0	0	0	0	154	94.81%
Agriculture	1	26	188	9	0	15	7	246	76.42%
Urban	0	0	6	134	0	8	6	154	87.01%
Water to Urban	0	0	0	0	84	0	0	84	100.00%
Agriculture to Urban	0	0	9	9	2	136	24	180	75.56%
Vegetation to Urban	0	4	8	0	0	58	189	259	72.97%
Total								1136	

Table 4 Error Matrix of the Classic SVM Classifier (Overall Accuracy =90.32%, Kappa=0.8838)

	Reference Categories							Total	UA
	1	2	3	4	5	6	7		
Water	58	0	1	0	0	0	0	59	98.31%
Natural vegetation	1	149	4	0	0	0	0	154	96.75%
Agriculture	0	27	202	0	0	10	7	246	82.11%
Urban	0	0	4	145	0	5	0	154	94.16%
Water to Urban	0	0	0	0	84	0	0	84	100.00%
Agriculture to Urban	0	0	12	7	1	153	7	180	85.00%
Vegetation to Urban	0	0	2	0	0	22	235	259	90.73%
Total								1136	

Table 5 Accuracy Comparisons of the Hybrid, the Classic DT and the SVM Classifiers

	Accuracies			Hybrid~DT Test		Hybrid~SVM Test	
	Hybrid	DT	SVM	Z-Score	Significance Level	Z-Score	Significance Level
Water	98.31%	77.97%	98.31%	3.4162	P<0.001	/	/
Natural vegetation	96.10%	94.81%	96.75%	0.5471		-0.307	
Agriculture	80.89%	76.42%	82.11%	1.2104		-0.3483	
Urban	90.26%	87.01%	94.16%	0.8977		-1.2754	
Water to Urban	100.00%	100.00%	100.00%	/	/	/	/
Agriculture to Urban	85.56%	75.56%	85.00%	2.397	P<0.01	0.1487	
Vegetation to Urban	92.28%	72.97%	90.73%	5.7982	P<0.001	0.6304	
Overall	89.88%	81.25%	90.32%	5.8499	P<0.001	-0.3512	

To further compare the classification accuracies at the individual class level, the accuracies for the hybrid, classic DT and SVM classifiers for each of the seven classes are listed in Tables 2-4 as well. Z-statistics between the accuracies of the hybrid classifier and the classic DT classifier and Z-statistics between the accuracies of the hybrid classifier and the SVM classifier for the classes are also calculated and listed in Table 5. For classifications using two classifiers and having the same accuracies, it is not possible to calculate Z-statistics and the correspondingly Z-scores and confidence levels are marked with "/". From the results it is clear that the hybrid classifier outperforms the classic decision tree classifier for all classes

except *Water to Urban* where both classifiers achieve full (100%) classification accuracies. More specifically, the hybrid classifier outperforms the classic DT classifier for classes *Agriculture to Urban* at p<0.01 significance level and *Water* and *Vegetation to Urban* at p<0.001 significance level. Table 5 also shows that while the SVM classifier achieves slightly better with respect to the overall classification accuracy than the hybrid classifier, SVM is not always better than the hybrid classifier at the class level. In fact, the SVM classifier performs better only for four out of the seven classes and none of them are statistically significant at the significance level p<0.1.

4.2 Test of Interpretability

While the hybrid classifier achieves much higher classification accuracies than the classic DT classifier and comparable classification accuracies to the SVM classifier, the most significant advantage of the hybrid classifier is its capability to generate concise and human interpretable decision rules. Among the 4840 training samples, the hybrid classifier generalizes 2141 samples and creates a compact decision tree (Fig. 5). The resulting decision tree has eight leaves which can be easily translated into decision rules. In contrast, the decision tree resulting from the original decision classifier has 214 leaves and is too big to fit in a page for presentation. In addition, we find that the significant decision rules resulting from the classic DT classifier are mixed with insignificant decision rules and it is hard for users to interpret. Thus the hybrid classifier has the capacity to leverage the most significant decision rules with high classification accuracies and present them to users for immediate validations.

```
b4 <= 10
|  b11 <= 8
|  |  b9 <= 12: Water (165.0)
|  b11 > 8
|  |  b3 <= 17: Water to Urban (387.0/6.0)
b4 > 10
|  b8 <= 40
|  |  b0 <= 66
|  |  |  b7 <= 29: Natural vegetation (342.0/9.0)
|  |  b0 > 66
|  |  |  b6 > 68
|  |  |  |  b5 <= 17: Agriculture (387.0/14.0)
|  |  |  |  b5 > 17
|  |  |  |  |  b9 > 38: Agriculture (222.0/9.0)
|  b8 > 40
|  |  b2 <= 51
|  |  |  b3 > 36
|  |  |  |  b7 > 38
|  |  |  |  |  b1 <= 30
|  |  |  |  |  |  b10 > 111
|  |  |  |  |  |  |  b1 <= 28
|  |  |  |  |  |  |  |  b5 <= 18: Vegetation to Urban (185.0/5.0)
|  |  |  |  |  b1 > 30
|  |  |  |  |  |  b3 <= 43
|  |  |  |  |  |  |  b3 <= 40: Agriculture to Urban (116.0/1.0)
|  |  b2 > 51: Urban (337.0/5.0)
```

Fig. 5 The Resulting Compact Decision Tree

The resulting compact decision tree from the hybrid classifier is quite meaningful. The first decision rule, b4<=10 and b11<=8 and b9<=12➔*Water* generalizes 165 out of the 250 training samples of the class (66.0%). The second rule, b4<=10 and b11>8 and b3<=17 ➔ *Water to Urban* generalizes 387 out of the 544 training samples (69.3%) with six exceptions. The exceptions are allowed because the min_accuracy is set to 95% and there could be up to 5% exceptions. By comparing the two rules, it is easy to derive the following interpretations. Class *Water* has low values in both band 5 at time T1 image (b4) and band 7 at time T2 image (b11). While class *Water to Urban* has low values in band 5 at time T1 image (b4), it has high values in band 7 at time T2 image (b11). The derived rules match domain knowledge very well - urban samples (pixels) have higher values than water samples. This can be further explained by the rule derived from the very bottom branch of the decision tree in Fig. 5 related to class *Urban*: b4>10 and b8>40 and b2>51➔*Urban*. The rule generalizes 337 out of the 682 samples (49.4%) with just five exceptions.

Similarly, the following five rules can be derived for the rest four classes:

1) B4>10 and b8<=40 and b0<=66 and b7<=29➔*Natural Vegetation*. The rule generalizes 342 out of the 568 samples (60.2%) with 9 exceptions.

2) B4>10 and b8<=40 and b0>66 and b6>68 and b5<=17➔*Agriculture*. The rule generalizes 342 out of the 962 samples (40.2%) with 14 exceptions

3) B4>10 and b8<=40 and b0>66 and b6>68 and b5>17 and b9>38➔*Agriculture*. The rule generalizes 222 out of the 962 samples (23.1%) with 9 exceptions

4) B4>10 and b8>40 and b2<=51 and b3>36 and b7>38 and b1<=30 and b1<=28 and b5<=18➔ *Vegetation to Urban*. The rule generalizes 185 out of the 775 samples (23.9%) with 5 exceptions.

5) B4>10 and b8>40 and b2<=51 and b3>36 and b7>38 and b1>30 and b3<=40➔ *Agriculture to Urban*. The rule generalizes 116 out of the 1059 samples (11.0%) with 1 exception.

Rule 2 and rule 3 are related to the same class (*Agriculture*). If we group the two rules then 564 out of the 962 samples (58.6%) can be generalized with 23 exceptions. The two rules for the *Agriculture* class are pretty similar and fall in the same decision tree branch (B4>10 and b8<=40 and b0>66). The breaching point for class *Agriculture* and class *Natural Vegetation* is b0=66 which indicates that natural vegetation has lower pixel values than agriculture at band 1 in time T1 image (b0). Compared with the rules of the other five classes, rules representing the two change classes *Vegetation to Urban* and *Agriculture to Urban* (Rule 4 and Rule 5) are less well represented since lower percentages of the samples of the classes can be generalized by the rules. This might indicate that these two classes are more complex and their sample values may not fit linear classifiers (such as decision tree) very well. Similar to characterizing the differences between class *Water* and class *Water to Urban* (and class *Natural Vegetation* and class *Agriculture* as well), from the resulting decision tree it is clear that, the difference between the samples that are generalized by Rule 4 (*Vegetation to Urban*) and Rule 5 (*Agriculture to Urban*) is that *Vegetation to Urban* has smaller values at band 2 of time 1 image (b1) than these of class *Agriculture to Urban*. The breaching point is b1=30. However, since the samples generalized by the two decision rules are only a fraction of the total samples

of the two classes, cautions are needed to validate this interpretation.

The resulting decision tree also naturally generates a hierarchy of the seven classes in the change detection dataset. From Fig. 5 we can see that water related classes (*Water* and *Water to Urban*) are first separated from the rest. The clustering process is followed by grouping urban related classes (*Vegetation to Urban*, *Agriculture to Urban* and *Urban*) as one cluster and *Natural Vegetation*/*Agriculture* as another cluster. Among the cluster for urban related classes, the two changing classes (*Vegetation to Urban* and *Agriculture to Urban*) are naturally grouped again. The class hierarchy can be used for knowledge transfer (Rajan and Ghosh 2006) and to refine the classification process. For example, decomposing a multi-class problem into multiple binary classifications based on the class hierarchy.

5. Conclusions and Future Work

In this study, we have proposed a hybrid algorithm that tightly integrates a decision tree algorithm and a SVM algorithm to classify multi-date images for land cover change detections. Experimental results show that the hybrid algorithm significantly improves the accuracies of the classic decision tree based classifier and achieves comparable classification accuracy to classic SVM based classifier. In addition, the hybrid algorithm leverages the most significant decision rules with high classification confidences and presents them to user for immediate evaluations.

The proposed hybrid algorithm represents a framework of hybridizing existing classification algorithms for classifying remotely sensed images. Due to the fuzzy and vague nature of classes defined by human and the inaccuracy introduced by the sampling process, the existence of samples that are difficult to classify is inevitable. Instead of using a single complex classifier for all the samples, it is more beneficial to use simple classifiers for "easy" samples and generate human interpretable knowledge from the classifiers through visualization (Zhang et al 2009) while leave the "difficult" samples for more sophisticated classifiers where visualization is usually not available. We also would like to point out that, while this work originates from the multi-date land cover change detection research, HC-DT/SVM is generic enough to be applied to a variety of types of environmental data analysis where supervised classifications are involved.

For future work, first, we would like to incorporate the hybrid classification algorithm into our VDM-RS (Visual Data Mining for Remote Sensing) prototype system (Zhang et al 2009) and help users gain more insights into the data, classification algorithm and results through visualization, interaction and exploration. Second, we would like to compare the HC-DT/SVM algorithm with other approaches discussed in the introduction section and test them on additional datasets. Finally, we plan to release the implementation as an open source package after proper documentation.

6. Acknowledgements

The author thanks Dr. Weiguo Liu for providing the test dataset and constructive discussions.

7. References

1. Chan, J. C. W., Chan, K. P. and Yeh, A. G. O., 2001. Detecting the nature of change in an urban environment: A comparison of machine learning algorithms. Photogrammetric Engineering and Remote Sensing 67(2): 213-225.
2. Chang, C. and Lin, C. 2001. LIBSVM: a Library for Support Vector Machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm.
3. De Colstoun, E. C. B. and 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(4): 474-485.
4. De Fries, R. S. and Chan, J. C. W., 2000. Multiple criteria for evaluating machine learning algorithms for land cover classification from satellite data. Remote Sensing of Environment 74(3): 503-515.
5. EL-Manzalawy, Y. and Honavar, V. 2005. WLSVM: Integrating LibSVM into Weka Environment, http://www.cs.iastate.edu/~yasser/wlsvm.
6. Friedl, M. A. and Brodley, C. E., 1997. Decision tree classification of land cover from remotely sensed data. Remote Sensing of Environment 61(3): 399-409.
7. Friedl, M. A., Brodley, C. E. and Strahler, A. H., 1999. Maximizing land cover classification accuracies produced by decision trees at continental to global scales. IEEE Transactions On Geoscience and Remote Sensing 37(2): 969-977.
8. 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. and Schaaf, C., 2002. Global land cover mapping from MODIS: algorithms and early results. Remote Sensing of Environment 83(1-2): 287-302.
9. Gibert, K., Sanchez-Marre, M. and Rodriguez-Roda, I., 2006. GESCONDA: An intelligent data analysis system for knowledge discovery and management in environmental databases. Environmental Modelling & Software 21(1): 115-120.
10. Huang, C., Davis, L. S. and Townshend, J. R. G., 2002. An assessment of support vector machines for land cover classification. International Journal of Remote Sensing 23(4): 725-749.
11. Huang, Z. and Lees, B. G., 2004. Combining non-parametric models for multisource predictive forest mapping. Photogrammetric Engineering and Remote Sensing 70(4): 415-425.
12. Im, J. and Jensen, J. R., 2005. A change detection model based on neighborhood correlation image analysis and decision tree classification. Remote Sensing of Environment 99(3): 326-340.
13. Kelly, M., Shaari, D., Guo, Q. H. and Liu, D. S., 2004. A comparison of standard and hybrid classifier methods for mapping hardwood mortality in areas affected by "sudden oak death". Photogrammetric Engineering and Remote Sensing 70(11): 1229-1239.
14. Liu, W. G., Gopal, S. and Woodcock, C. E., 2004. Uncertainty and confidence in land cover classification using a hybird classifier approach. Photogrammetric Engineering and Remote Sensing 70(8): 963-971.

15. Lu, D., Mausel, P., Brondizio, E. and Moran, E., 2004. Change detection techniques. International Journal of Remote Sensing 25(12): 2365-2407.
16. Lu, D. and Weng, Q., 2007. A survey of image classification methods and techniques for improving classification performance. International Journal of Remote Sensing 28(5): 823-870.
17. Nemmour, H. and Chibani, Y., 2006. Multiple support vector machines for land cover change detection: An application for mapping urban extensions. Isprs Journal of Photogrammetry and Remote Sensing 61(2): 125-133.
18. Pal, M. and Mather, P. M., 2005. Support vector machines for classification in remote sensing. International Journal of Remote Sensing 26(5): 1007-1011.
19. Pal, M. and Mather, P. M., 2006. Some issues in the classification of DAIS hyperspectral data. International Journal of Remote Sensing 27(14): 2895-2916.
20. Prasad, A. M., Iverson, L. R. and Liaw, A., 2006. Newer classification and regression tree techniques: Bagging and random forests for ecological prediction. Ecosystems 9(2): 181-199.
21. Rajan, S., Ghosh, J. and Crawford, M. M., 2006. Exploiting class hierarchies for knowledge transfer in hyperspectral data. IEEE Transactions on Geoscience and Remote Sensing 44(11): 3408-3417.
22. Rueda, C. A., Greenberg, J. A. and Ustin, S. L., 2005. StarSpan: A Tool for Fast Selective Pixel Extraction from Remotely Sensed Data. Center for Spatial Technologies and Remote Sensing (CSTARS), University of California at Davis, Davis, CA.
23. Seto, K. C. and Liu, W. G., 2003. Comparing ARTMAP neural network with the maximum-likelihood classifier for detecting urban change. Photogrammetric Engineering and Remote Sensing 69(9): 981-990.
24. Singh, A., 1989. Digital Change Detection Techniques Using Remotely-Sensed Data. International Journal of Remote Sensing 10(6): 989-1003.
25. Witten, I. H. and Frank, E., 2000. Data Mining: Practical machine learning tools with Java implementations. San Francisco, CA, Morgan Kaufmann.
26. Wynne, R. H., Joseph, K. A., Browder, J. O. and Summers, P. M., 2007. Comparing farmer-based and satellite-derived deforestation estimates in the Amazon basin using a hybrid classifier. International Journal of Remote Sensing 28(6): 1299-1315.
27. Yuan, F., Sawaya, K. E., Loeffelholz, B. C. and Bauer, M. E., 2005. Land cover classification and change analysis of the Twin Cities (Minnesota) Metropolitan Area by multitemporal Landsat remote sensing. Remote Sensing of Environment 98(2-3): 317-328.
28. Zhang, J., Liu, W. and Gruenwald, L. (2007). A Successive Decision Tree Approach to Mining Remotely Sensed Image Data. Knowledge Discovery and Data Mining: Challenges and Realities. X. Zhu and I. Davidson, Idea Group Publishing, Inc: 98-112.
29. Zhang J.,Gruenwald, L. and Gertz. M (2009). VDM-RS: A Visual Data Mining System for Exploring and Classifying Remotely Sensed Images. Computers & Geosciences 35(9), 1827-1836