Change detection in remotely-sensed images using associative classification

Raghunath dey
Department of Computer Science and Engineering
Balasore College of Engineering and Technology, Sergarh, Balasore, Orissa

Jayashree Piri
Department of Information Technology
Balasore College of Engineering and Technology, Sergarh, Balasore, Orissa

Abstract - Change detection is the work of detecting the regions of changes in multi-spectral images taken over same geographical area, by observing it at different times. In this paper a supervised classification technique based on associative classification is presented for change detection in remotely-sensed multi-spectral and multi-temporal images. Though there are large numbers of change detection techniques exist in literature, no attempts have been made from the point of associative classification towards remotely-sensed image change detection. Classification based association technique seems to be an appropriate and realistic choice to predict the class labels of pixels across multi-temporal satellite images by using very less number of training samples with in a very small time. Potentiality of the presented method is justified from the experimental outcome on a number of satellite images.

Keywords: Remote sensing, Change detection, Multi-spectral images, Multi-temporal images, Class association rules, Associative classifier, Discretization, Confusion matrix.

1. INTRODUCTION

In remote sensing applications, change detection is the process of identifying differences in the state of an object or phenomenon by analyzing a pair of images acquired on the same geographical area at two different instants [26]. Change detection is useful in so many applications, such as land use change analysis, monitoring of shifting cultivation, assessment of deforestation, study of changes in vegetation, seasonal changes in pasture production, damage assessment, crop stress detection, disaster monitoring, snow-melt measurements, day/night analysis of thermal characteristics and other environmental changes. It is very difficult to manually handle data for change detection using sequential imagery. So, to automatically correlate and compare two sets of imagery taken of the same area at different times and display the changes and their locations to the interpreter, an automatic process for change detection is needed.

This change detection task can be viewed as a classification process, i.e. Changed class (can be more than one) and another is unchanged class. Classification and association rule discovery are similar except that classification involves prediction of one attribute, i.e. the class, while association rule discovery can predict any attribute in the datasets. Classification is a two-step process. In the first step, a classifier is built by the help of a predetermined set of data with their class labels. This is the learning step (or training phase). Second is the test step, made up of test tuples and their associated class labels. These tuples are randomly selected from the general dataset. They are independent of the training tuples, meaning that they are not used to construct the classifier [13]. In last few years, a new approach that integrates association rule mining with classification is found in CBA [19], CMAR[18], CPAR[30], MMAC[29]. There is a growing evidence that merging classification and association rule mining together can produce more accurate and efficient classification system than traditional rule-based classification techniques.

A decision tree [23] is a flowchart-like tree structure, where each internal node (non leaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The top most node in a tree is the root node. Here for a given tuple, for which the associated class label
is unknown, the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple. Decision trees can easily be converted to classification rules.

The decision tree output of Quinlan’s ID3 algorithm is one of its major weaknesses. Not only can it be incomprehensible and difficult to manipulate, but also its use in expert systems frequently demands irrelevant information to be supplied. This report argues that the problem lies in the induction algorithm itself and can only be remedied by radically altering the underlying strategy. PRISM [3] which, although based on ID3, uses a different induction strategy to induce rules which are modular, thus avoiding many of the problems associated with decision trees.

Partial tree (PART) is an algorithm for inferring rules by repeatedly generating partial decision trees [8], thus combining the two major paradigms for rule generation (creating rules from decision trees) and the separate-and-conquer rule (learning technique). It produces rule sets that are as accurate and of similar size to those generated by C4.5 [23], and more accurate than RIPPERs [4]. Moreover, it operates efficiently, and because it avoids post processing, does not suffer the extremely slow performance on pathological example sets for which the C4.5 method has been criticized.

Constructing fast and accurate change detector for remotely sensed images by the help of very few labeled samples is a difficult task. In this paper a change detection technique, that is based on association rule mining (ARM) [1] and rule based classification system [13], which is known as classification based on association [19], for remotely sensed image change detection is being presented. Traditional classification techniques often produce a small subset of rules, and therefore usually miss detailed rules that might play an important role in some case. More over many of the rules found by associative classification method cannot be found by traditional classification techniques. Here a set of images are provided, those are of same geographical area taken at different times. The goal is to identify the set of pixels that are significantly different between the last image of the sequence and the previous images [24]. This technique can be broken down as follows:

- Preprocessing upon the continuous valued attributes. (Discretization process)
- Adapt the apriori [1] algorithm to generate class association rules (CARs) [19] efficiently.
- Build a classifying model from CARs.
- Extend the model to produce the change detection map for remotely sensed multi-spectral and multi-temporal images.

Rest of this paper is organized as follows. Section 2 gives a short survey on remotely-sensed image change detection. Section 3 describes the presented method briefly. Data sets description, evaluation metrics and analysis of results are briefly described in Section 4. Finally, Section 5 concludes the paper.

2. BACKGROUND AND RELATED WORK

Change detection is a process, that observes the differences of an object between two remotely-sensed images which are taken at different times. Images of two dates are transformed into a new single image, which contains the changes. There are so many techniques present in literature [26] to detect the changes of remotely-sensed images, such as image differencing, image rationing, image regression, change vector analysis, multi-date principal component analysis, post-classification technique, etc. The resultant changed image must be further processed to assign the changes to each pixel. Here several methods for generating the change detection map in remotely-sensed images have been discussed. It has been generally agreed that change detection is a complicated and integrating process. No existing approach is optimal and applicable to all cases. Few of them are discussed here.

In case of histogram thresholding [26] the difference image can classified into two groups, namely, ‘changed’ and ‘unchanged’ classes. An optimal threshold is selected by the discriminant criterion, namely, so as to maximize the separability of the resultant classes in gray levels. Utilizing only the zeroth and the first-order cumulative moments of the gray-level histogram. The optimal threshold is determined by analyzing the behavior of the variances of changed and unchanged classes obtained assuming different threshold values. It is straightforward to extend the method to multi threshold problems as in otsu’s method [21]. Another method where optimal threshold is determined based on the concept of entropy [25]. Two probability distributions, one for unchanged (\(\omega_u\)) and the other for changed (\(\omega_c\)), are derived from the original grey level distribution of the difference image by assuming a threshold value as in kapur’s method [16].

An unsupervised context-sensitive technique for change-detection in multi-temporal remote sensing images proposed in [9]. Here a modified self-organizing feature map neural network is used. The network is updated
depending on some threshold value and when the network converges, status of output neurons depicts a change-detection map. To select a suitable threshold of the network, a correlation based and an energy based criteria are suggested.

A context-sensitive change-detection technique based on semi-supervised learning with multilayer perceptron is proposed in [22]. The network is initially trained using these labeled data. The unlabeled patterns are iteratively processed by the already trained perceptron to obtain a soft class label. A new unified approach is presented that integrates SVM based classifier to change detection (SVMCCD) is proposed in [31]. Combined with the change detection task, a bootstrapping strategy is proposed to solve sample selection problem. Considering the relative simplicity of non-change patterns, one-class SVM based change detection method is also provided.

Land use and land cover (LULC) can usually indicate the change of the eco-environment for one area. In the paper [6], a knowledge based decision tree classification system based on the spectrums knowledge of ground objects and their spatial knowledge was developed.

Two techniques for change detection have been developed to deal with the more general scenario where illumination is not assumed to be constant as shown in the paper [27]. The derivative model method uses partial derivatives with respect to the pixel coordinates of a second order gray level surface model to compare regions and determine if a change has taken place.

Two algorithms for the detection of small changes in a pair of images in a low signal to clutter plus noise ratio (SCNR) environment is being proposed in the paper [17]. They both have the ability to track the non stationary image signals and suppress the clutter plus noise background. Both detectors are based on the adaptive correlation canceling technique. The first one uses an order recursive least squares (ORLS) lattice filter, while the second one is based on the two dimensional least mean square (TDLMS) algorithm.

The article [20] demonstrates a methodology for predicting those areas with the greatest propensity for deforestation based on natural and cultural landscape variables. Logistic regression analysis was used to determine variables, which most closely associated with deforestation. GIS analysis was then used to verify spatially the close statistical relationship between the dependent variable and each of the independent variables selected by the logistic regression modeling.

Two procedures were developed to make better use of multi spectral information from remotely sensed data for change detection as in [10]. Principal component analysis (PCA) was applied to difference image instead of applying to a combined data set of original multi spectral images. Operation based on fuzzy set theory were proposed to combine change information from different change images channels into a single image channel. Change areas could been extracted from this single image.

Optimum multi sensor data fusion is taken for image change detection based on the optimum likelihood ratio test for the statistical dependence of the luminance signals in additive Gaussian noise as shown in [14]. It is demonstrated that the information to be transmitted from the sensors to the fusion center is the maximum likelihood estimates of the correlation coefficients between pairs of consecutive image frames. It shows that the detection error decreases as the number of sensors and/or frames increases.

Two automatic techniques (based on the Bayes theory) for the analysis of the difference image is being proposed [2]. One allows an automatic selection of the decision threshold that minimizes the overall change detection error probability under the assumption that pixels in the difference image are independent of one another. The other analyzes the difference image by considering the spatial-contextual information included in the neighborhood of each pixel. In particular, an approach based on Markov Random Fields (MRFs) that exploits inter pixel class dependency contexts is presented.

3. PROPOSED METHODOLOGY

Here this paper presents an application of the classification based association [19] to multi-spectral, multi-temporal remotely-sensed images change detection. Rule mining using association and classification are two major techniques in data mining. Finding all possible rules satisfying minimum support and minimum confidence is done by association rule mining. Here the target is not pre-determined. In a large database discovering a small set of rules as a classifier is done by classification rule mining. Here is only one pre-determined target, i.e. the class. Both these association rule mining and classification rule mining are necessary to practical application. So integration of these two is known as Associative classification. There are so many rule based classification algorithms present in literatures [3, 23, 15, 8], but these traditional rule based classification techniques often produce a small subset of rules. More over many of the rules found by associative classification method can not be
found by traditional classification technique, which often plays an important role in some cases. There are so many algorithms present for associative classification. Among them classification based association [19] is one of them.

The complete change detection process consists of basic two steps. Those are namely (1) Generation of input patterns, and (2) Classification step. (for generating changed and unchanged class).

In the first step, required percentage of training sample is being selected from the difference image using reference map. The next classification step is based on the concept of associative classification which will generate the change detection map. The whole process is being shown in the block diagram 1 and later the steps are being described briefly.

3.1. Generation of input patterns

As this is a supervised change detection technique, we need few labeled training samples. The complete process for generation of labeled training patterns are described below.

3.1.1. Producing difference image

This technique to generate difference image exploits a simple vector subtraction operator to compare pixel-by-pixel of two images of different dates. The difference image is computed as the magnitude of spectral change vectors obtained for each pair of corresponding pixels [9]. The difference image obtained by as follows:

![Figure 1: Block diagram for change detection using associative classification of multi-spectral multi-temporal remotely-sensed images.](image)
where PDI(mn) is the gray value of the (m, n)th pixel (1 ≤ m ≤ p, 1 ≤ n ≤ q) in the difference image generated from corresponding pixels of the images X1 (of time t1) and X2 (of time t2) having γ bands i.e. b=1, 2,......γ. Generating multi spectral difference image corresponding to (m, n)th pixel is given by

\[ P_{D}(m,n) = [P_{D}(m,n,b)]_{b=1}^{γ} \]  \hspace{1cm} (2)

Here the attribute values required for the classification based association algorithm are in the form of difference values of the intensity labels of each band of images. That means if there are γ number of bands of images present, then there will be γ number of attributes for classification based association approach. These intensity values are continuous in nature and they lie in between ‘0’ to ‘255’.

3.1.2. Training sample selection
This process will provide us required number of labeled training patterns by the help of which the classifier can be built. From the set of difference images and the reference map, required percentage of training samples are chosen from both changed and unchanged regions randomly. The training samples are consisting of labeled patterns; those are from both changed class as well as unchanged class. After selecting the required number of labeled training samples the set of all difference images are provided to the classification step. These are unlabeled and used to calculate the accuracy of the classifier and to generate the change detection map.

3.2. Classification based association
Classification based on association work is done on the basis of following three steps:

- Discretization of continuous attributes,
- Rule generation, and
- Building classifier.

These processes are being described as follows.

3.2.1. Discretization of continuous attributes
The difference image consists of the difference values of grey values of each band of images. These values are continues in nature lies in between 0 to 255. For easy manipulation we need a discretizing process. Generally a continuous valued attribute is discretized by partitioning its range into two intervals. For the continuous valued attribute ‘A’ a threshold value T is to be found out. The values of A less than T are to be kept left of the T and values of A greater than T are to be kept right to T. So T is called as cut point. For each continuous valued attribute ‘A’ the best cut point TA is to select from its range of values.

Discretization procedure: Here in this paper, discretization of continuous attributes is done using the Entropy method in [7]. As the attribute values are continues in nature, the entropy based discretization technique [7] was applied to convert the ranges [28] into discrete values. By the help of which we can easily apply the classification based association approach.

Evaluation of Entropy: For every cut point data are partitioned into two subsets. If T partitions the set S into the subset S1 and S2, and If there are k classes C1,.....Ck, then class information entropy (E(A,T,S)) made by the cut point T is given by:

\[ E(A,T,S) = \frac{S1}{S} \text{Ent}(S1) + \frac{S2}{S} \text{Ent}(S2) \] \hspace{1cm} (3)

Where class entropy (Ent(S)) is given by
\[ \text{Ent}(S) = \sum_{c} p(c|s) \log(p(c|s)) \] ..................................................(4)

Then gain is given by

\[ \text{Gain}(A,T,S) = \text{Ent}(S) - E(A,T,S) \] ..................................................(5)

These steps continue recursively until a stopping condition is found.

Decision Criterion: Selecting the cut point and continuing partitioning till the following condition satisfy.

\[ \text{Gain}(A,T,S) > \frac{\log(N-1)}{N} + \frac{\log(T)}{N} \] ..................................................(6)

Where \( N \) is the number of tuples in \( S \), and \( T \) is the cut point and let \( k_1, k_2 \) is the number of classes in partition \( S_1, S_2 \) respectively, then

\[ \Delta(A,T,S) = \log_2(3^k - 2) - [k \text{Ent}(S) - k_1 \text{Ent}(S_1) - k_2 \text{Ent}(S_2)] \]

The complete discretization process is shown in Algorithm 1.

**Algorithm 1**: Discretization process

**Input**: A dataset ‘D’ with ‘γ’ continuous bands (features) and C target classes. SPLITPOINT[] is an array that stores all the selected split/cut points of a particular band (feature). SINDEX is the number of splitpoints.

1: for each feature \( i=1 \) to \( \gamma \) do
2: Initialize the array SPLITPOINT[] as 0 and SINDEX=0;
3: find the maximum and minimum value of feature \( i \);
4: split(D, i) (described in algorithm 2);
5: sort the array SPLITPOINT[] in ascending order;
6: print “minimum to SPLITPOINT[0];”;
7: for \( j=0 \) to SINDEX do
8: print “SPLITPOINT[j]:SPLITPOINT[j+1]”
9: end for
10: print “SPLITPOINT[SINDEX] to maximum;”
11: end for
Algorithm 2: split(D, i) function
1: Form a set of all distinct values of feature i of D in ascending order;
2: calculate the mid points of all adjacent pairs in the set;
3: calculate the class information entropy of each mid point using equation 3 and the smallest will be
   chosen as the potential cut point;
4: store the split point in SPLITPOINT[SINDEX];
5: SINDEX++;
6: Divide D into two partition D_1 and D_2 such that D_1 contains all the samples in D having feature
   i values ≤ splitpoint and D_2 contains all the samples in D having feature i value > split point;
7: calculate class entropy and gain using equation 4 and equation 5 respectively.
8: if equation 6 satisfies then
9:   split(D_1, i);
10:  split(D_2, i);
11: else
12:   return;
13: end if

3.2.2. Rule generation
By making multiple passes over the data the classification based association rule generation algo- rithm generates all the frequent ruleitems. The support of individual ruleitem is counted in the first pass and it is being determined whether it is frequent or not. In each subsequent pass, it starts with the seed set of ruleitems found to be frequent in the previous pass. To generate new possibly frequent ruleitems, it uses this seed set called candidate ruleitems. The support counts for these candidate ruleitems are calculated during the pass over the data. At the end of the pass, it determines which of the candidate ruleitems are actually frequent. From this set of frequent ruleitems, it produces the rules class association rules. Finally, rule pruning is performed on these rules. The function pruneRules uses the pessimistic error rate based pruning method in C4.5 [23].

Some basic idea on rule generation based on classification based association are given as follows:
ruleitems: This is represented by < condset, y >.
Each rule item basically represent a rule, i.e. condset → y
where condset is a set of items, and y ∈ Y is a class label.
The support count of the condset called condSupCnt is the number of cases in dataset D that contain the condset.
The support count of the ruleitem is called ruleSupCnt is the number of cases in D that contain the condset and are labeled with class y.
Support: Calculated by
(rulesupCnt/|D|) * 100% 
(7)
confidenced: Calculated by
(rulesupCnt/condsupCnt) * 100% 
(8)
Example:
Ruleitem: <|{(A, 1), (B, 1)}, (C, 1)>|
condSupCnt= 3 ruleSupCnt= 2
support= (2 / 10) * 100% = 20%
confidence: (2 / 3) * 100% = 66.7%.
Frequent ruleitems must satisfy the minSup. Accurate rule are rules satisfying the minConf.
Possible rule (PRs) are ruleitems with same condset, with highest confidence. CARs are the PRs (both frequent and accurate) [19].

Algorithm 3: Rule generation process
1: Determine frequent-1 (F1) ruleitems;
2: Generate CAR_1 and prune (optional);
3: for each subsequent k pass do
4: evaluate candidate ruleitem \( C_k \) from frequent \((F_k - 1)\) ruleitems ;
5: for each data case do
6: scan the data base and update condSupCnt and ruleSupCnt;
7: end for
8: Generate CAR\(_k\) and prune;
9: end for
10: Union all CARs and apply pruning;

where \( k \)-ruleitem denote a ruleitem whose condset has \( k \) items, \( C_k \) be the set of candidate \( k \)-ruleitems and \( F_k \) be the set of frequent \( k \)-ruleitems having ruleSup \( \geq \) minSup;

The candidate Generation task is similar to the algorithm Apriori [1]. The difference is that here it need to increment the support counts of the condset and the ruleitem separately, whereas in algorithm apriori only one count is updated. This allows us to compute the confidence of the ruleitem.

This rule generation algorithm counts the item and class occurrences to determine the frequent 1-ruleitems. From this set of 1-ruleitems, a set of CARs (called CAR1) is generated. CAR1 is subjected to a pruning operation. Its optional. The function pruneRules uses the pessimistic error rate based pruning method in C4.5 [23]. First, the frequent ruleitems \( F_{k-1} \) found in the \((k-1)\)th pass are used to generate the candidate ruleitems \( C_k \). scans the database and updates various support counts of the candidates in \( C_k \). After that new frequent rules have been identified to form \( F_k \). The algorithm then produces the rules CAR\(_k\). Finally, rule pruning is performed on these rules.

3.2.3. Building classifier

Out of the whole set of rules, the best classifier has to produce. It would involve evaluating all the possible subsets of it on the training data. This section presents the CBA-CB (Classifier building) [19] algorithm given in 4 for building a classifier using CARs (or pruned CARs). Selecting the subset with the right rule sequence that gives the least number of errors. Before going through the algorithm one thing is to be known, i.e. sorting of rules in precedence order.

Sorting in precedence order: Two rules, \( ri \) and \( rj \), \( ri \) precedes \( rj \), if confidence of \( ri \) is greater than \( rj \), or confidences same, but support of \( ri \) is greater than \( rj \), or Both confidences and supports same, but \( ri \) generated earlier than \( rj \). On the basis of these criteria the rules are being sorted according to their precedence.

The classifier is of the following format

\(< r_1 , r_2 , \ldots \ldots , r_n , \text{default class} >\)

where \( r_i \in R \), \( r_a > r_b \) if \( b > a \). Def aultClass is the default class.

In classifying an unseen case, the first rule that satisfies the case will classify it. If there is no rule that applies to the case, it takes on the default class.

To choose the highest precedence rules for the classifier first sort the set of generated rules according to precedence relation as discussed earlier. Take a rule from the rule set and if it can correctly classify at least one tuple of the Dataset D, then it will be a potential rule for the classifier. Those tuples covered by the rule are then removed from the dataset D. From the majority class in remaining
Algorithm 4: Building classifier process
1: sort generated rules according to precedence relation as in section 3.2.3;
2: for each rule do
3: if correctly classify at least one case of data base then
4: store inside the classifier;
5: those cases it covers, are then removed from data base D;
6: select a default class for the current classifier;
7: compute the total number of errors of current classifier;
8: end if
9: end for
10: discard those rules that do not improve accuracy of classifier;

dataset a default class will be chosen. Total error (sum of error done by the current classifier and the default class) is calculated. This process will stop when there will be no more rules or training case will left. Finally the first rule having least number of errors is to be chosen. All the rules after this rule can be discarded, because including them the accuracy of the classifier is not improving.
As sorting is performed upon the generated CARs, so that it will satisfy two conditions: Condition 1 - Each training case is covered [13] by the rule with the highest precedence among the rules that can cover the case. Condition 2 - Every rule in classifier correctly classifies at least one remaining training case when it is chosen.
Rule pruning: The prune Rules function uses the pessimistic error rate based pruning method in C4.5 [23]. It prunes a rule as follows: If rule r’s pessimistic error rate is higher than the pessimistic error rate of rule r’ (obtained by deleting one condition from the conditions of r), then rule r is pruned. This pruning can cut down the number of rules generated substantially.

3.3. Associative classification based change detection
Here a new way to build accurate classifiers for the change/unchanged pixels in remotely sensed images is being presented. Initially the difference images of respective bands of images are being found out. From this the required number of training samples are being selected from the changed as well as unchanged areas randomly and the corresponding class labels are being selected from the provided reference map. That means a particular row of the training sample set consists of γ number of difference values (if there are γ number of bands of images taken at time t1 and t2) with their resultant class label. The selected training samples are being discretized [7] as these are continuous in nature. Then classification based association [19] technique is being applied upon the discretized labeled training samples (patterns) to generate the class association rules (CARs) [19]. After generating CARs sorting is being performed as discussed earlier (in section 3.2.3). From these CARs the required classifier is being developed. A rule set pruning [23] is also applied to make a compact and accurate classifier. On the other hand the whole set of difference image (consists of only γ number of difference values of γ number bands of images) are being discretized by the help of discretizing labels found from the discretizing process applied on the labeled training samples. Then these discretized unlabeled patterns of all difference images are provided to the built classifier. From which the classifier classifies the whole image into two region (namely changed and unchanged area). Here the change region is represented as black (grey level intensity value is 0) and unchanged region as white (grey level intensity value is 255). For the purpose of evaluating the classifier, various measures (Miss alarm, False alarm, Overall Accuracy, Producer Accuracy, User Accuracy, Micro F1, Macro F1, Kappa Coefficient and Time taken for execution) are being calculated from the confusion matrix.
4. Experimental evaluation

In order to carry out the experimental analysis aimed to assess the effectiveness of the proposed approach, we considered various multi temporal remote sensing data sets corresponding to geographical areas of Mexico and Island of Sardinia, Italy and the Peloponnesian Peninsula, Greece. A detailed description of each of the data sets is given below.

4.1. Data sets Descriptions

4.1.1. Data set of Mexico area

The first data set used in the experiment is made up of two multi spectral images acquired by the Landsat Enhanced Thematic Mapper Plus (ETM+) sensor of the Landsat-7 satellite in an area of Mexico on 18th April 2000 and 20th May 2002. From the entire available Landsat scene, a section of 512 x 512 pixels has been selected as test site. Between the two aforementioned acquisition dates, a fire destroyed a large proportion of the vegetation in the region.

Figure 2(a) and 2(b) show channel 4 of the 2000 and 2002 images, respectively. In order to make a quantitative evaluation of the effectiveness of the presented approach, a reference map was manually defined in Figure 2(d) according to a detailed visual analysis of both the available multi
temporal images and the difference image Figure 2(c). Different color composites of these images were used to highlight all the portions of the changed area in the best possible way. This procedure resulted in a reference map containing 25,599 changed and 236,545 unchanged pixels. Analysis of the behavior of the histograms of multi-temporal images did not reveal any significant difference due to light and atmospheric conditions at the acquisition dates. Therefore, no radiometric correction algorithm was applied. The 2002 image was registered on the 2000 one using 12 ground control points. The procedure led to a residual average misregistration error on ground control points of about 0.3 pixels.

4.1.2. Data set of Sardinia, Italy

The second data set used in the experiment is composed of two multi-spectral images acquired by the Landsat Thematic Mapper (TM) sensor of the Landsat-5 satellite in September 1995 and July 1996. The test site is a section of 412 × 300 pixels of a scene including Lake Mulargia on the Island of Sardinia, Italy. Between the two aforementioned acquisition dates, the water level in the lake increased (see the lower central part of the image). Figure 3(a) and 3(b) shows channel 4 of the 1995 and 1996 images respectively. As done for the Mexico data set, in this case also a reference map was manually defined Figure 3(d) according to a detailed visual analysis of both the available multi-temporal images and the difference image (Figure 3(c)). In the end, 7,480 changed and 116,120 unchanged pixels were identified. As the histograms did not show any significant difference, no radiometric correction algorithm was applied to the multi-temporal images. The images were co-registered with 12 ground control points resulting in an average residual misregistration error of about 0.2 pixels on the ground control points.
4.1.3. Data set of Peloponnesian peninsula, Greece
The third data set used in the experiment is composed of two images acquired of the same area by a passive multispectral scanner installed on a satellite, that is, the Wide Field Sensor (WiFS) mounted on board the IRS-P3 satellite. The area shown in the two images is a section (492 × 492 pixels) of a scene acquired in the southern part of the Peloponnesian Peninsula, Greece, in April 1998 and September 1998.
As an example, Figure 4(a) and 4(b) shows channel 2 (i.e. near-infrared spectral channels) of both the images. As is readily apparent, various wild-fires destroyed a significant portion of the vegetation in the aforesaid area between the two dates. Like the previously mentioned data sets, a reference map was manually defined (see Figure 4(d)) to assess change-detection errors. This reference map contains 5197 changed and 256 947 unchanged pixels. The images were registered using the multi-spectral image acquired in April as a reference image. The analysis of the histograms of the April and September images did not reveal any significant difference in the light conditions at the two dates.

4.2. Evaluation Metrics
For performing the evaluation, both the quantitative and qualitative measurements are conducted. For the former, once the change detection map has been obtained by using change detection method, the following defined quantities are computed for comparing the computed change detection map against the ground truth change map.
Miss Alarm (MA): The number of actually changed pixels that were missed out on their detections and mistakenly determined as unchanged ones.

False Alarm (FA): The number of actually unchanged pixels that were incorrectly determined as changed ones.

Overall Error (OE): The total number of incorrect decisions made, which is the sum of the false alarms and Miss Alarm. i.e. \( OE = MA + FA \)

The complete description of the information that comes out from the comparison of the classification of test samples with the reference labeled data is given by the confusion (or error) matrix \( N \). \( N \) is a square matrix of size \( C \times C \) (where \( C \) is the number of information classes in the considered problem). The generic element \( n_{ij} \) of the matrix denotes the number of samples classified into category \( i \) (\( i = 1, \ldots, C \)) by the classifier that are associated with label \( j \) (\( j = 1, \ldots, C \)) in the reference data set. From the confusion matrix, different indices can be derived to summarize the information with a scalar value. Let us consider the sum of the elements of row \( i \), \( n_i = \sum_{j=1}^{C} n_{ij} \) (which is the number of samples classified into category \( i \) in the classification map), and the sum of the elements of column \( j \), \( n_j = \sum_{i=1}^{C} n_{ij} \) (which is the number of samples belonging to category \( j \) in the reference data set). Some commonly adopted measures such as Overall Accuracy, Micro F1, Macro F1, Producer Accuracy, User’s Accuracy and Kappa coefficient of accuracy are defined as \( [12] \)

\[
\text{Overall Accuracy (OA)} = \frac{\sum_{i=1}^{C} n_{ii}}{n} \tag{9}
\]

\[
\text{Kappa} = \frac{\sum_{i=1}^{C} n_{ii} - \frac{\sum_{i=1}^{C} n_i \cdot \sum_{j=1}^{C} n_j}{n}}{n - \frac{\sum_{i=1}^{C} n_i \cdot \sum_{j=1}^{C} n_j}{n}} \tag{10}
\]

Where \( n = \sum_{i=1}^{P} \sum_{j=1}^{P} n_{ij} \) is the total number of test samples. OA represents the ratio between the number of samples that are correctly recognized by the classification algorithm and the total number of test samples. The Kappa coefficient of accuracy is a measure based on the difference between the actual agreement in the confusion matrix (as indicated by the main diagonal) and the chance agreement, which is indicated by the row and column totals (i.e., the marginals). The Kappa coefficient is widely adopted, as it uses also off-diagonal elements of the error matrix and compensates for chance agreement. The value of Kappa coefficient lies in the range \([-1, +1]\). More close the value of Kappa to +1, better is the classification.

User’s Accuracy (UA) \([5]\): For a given class, how many of the pixels on the map are actually what they say they are? In otherwise it can be told as if the total number of correct pixels in a category is divided by the total number of pixels that were classified in that category. It is calculated as:

\[
\frac{\# \text{Patterns Correctly Identified In A Given Map Class}}{\# \text{Patterns Claimed To Be In That Map Class}} \tag{11}
\]

Producer’s Accuracy (PA) \([5]\): For a given class in reference plots, how many of the pixels on the map are labeled correctly? Otherwise it can be told as the total number of correct pixels in a category is divided by the total number of pixels of that category as derived from the reference data.
It is calculated as:

\[
\frac{\text{\# Patterns Correctly Identified In Ref Plots Of A Given Class}}{\text{\# Patterns Actually In That Reference Class}}
\]

(12)

To evaluate the performance of the classifier the commonly known F-measure metric is used, which is equal to the harmonic mean of recall and precision.

Macro averaged \(F_1\): Macro averaged \(F_1\) is derived from precision and recall. The precision of a class \(i\) is defined as:

\[
\text{precision}_i(p_i) = \frac{\text{\# Patterns Correctly Classified Into Class } i}{\text{\# Patterns Classified Into Class } i}
\]

(13)

and recall of class \(i\) is defined as

\[
\text{recall}_i(r_i) = \frac{\text{\# Patterns Correctly Classified Into Class } i}{\text{\# Patterns That Are Truly Present In Class } i}
\]

(14)

Then \(F_1\), the harmonic mean between precision and recall, of class \(i\) is defined as:

\[
F_1 = \frac{2 \times \text{precision}_i \times \text{recall}_i}{\text{precision}_i + \text{recall}_i}
\]

(15)

\(F_1\) measure gives equal importance to both precision and recall.

The macro-averaged \(F_1\) measure is computed by first computing the \(F_1\) scores for each category (class) and then averaging these per-category scores to compute the global means [11]. Macro-averaged \(F_1\) (or MacroF1 in short) is defined as

\[
\text{macro averaged } F_1 = \frac{1}{M} \sum_{i=1}^{M} F_1
\]

(16)

where \(M\) is the number of category (class). Macro-averaged \(F_1\) gives equal weight to each category. The value of Macro-averaged \(F_1\) lies between 0 and 1. More close the value of macro-averaged \(F_1\) to 1, the better is the classification.

Micro-averaged \(F_1\) measure: The micro-averaged \(F_1\) measures are computed by first creating a global contingency table whose cell values are the sum of the corresponding cells in the per-category contingency tables [11]. Then use this global contingency table to compute the micro-averaged performance scores. Micro-averaged \(F_1\) gives equal weightage on each sample (test case). Micro-averaged \(F_1\) (or Micro \(F_1\) in short) is defined as:

\[
\text{micro averaged } F_1 = \frac{\text{\# Patterns Correctly Identified}}{\text{\# Patterns Actually Present In That Reference Class}}
\]

(17)

The value of Micro-averaged \(F_1\) lies between 0 and 1. More close the value of micro-averaged \(F_1\) to 1, the better is the classification.

4.3. Analysis of results

Table 1 shows that the change detection errors as well as accuracy produced using the presented method on various remotely-sensed images. The average confusion matrix of 20 trials on each data
set are shown in Table 2 (for Mexico data set), Table 3 (for Sardinia data set) and Table 4 (for Greece data set).

Figure 5: Change-detection maps obtained using CBA technique on Mexico area dataset by (a) 1% of training samples (b) 5% of training samples, (c) 10% of training sample, (d) 15% of training samples and (e) 20% of training samples.

Figure 6: Change-detection maps obtained using CBA technique on Sardinia area dataset by (a) 1% of training samples (b) 5% of training samples, (c) 10% of training sample, (d) 15% of training samples and (e) 20% of training samples.
<table>
<thead>
<tr>
<th>Data</th>
<th>Training</th>
<th>Miss</th>
<th>False</th>
<th>Overall Avg</th>
<th>Avg.</th>
<th>PA of UA of</th>
<th>F1-changed</th>
<th>F1-changed</th>
<th>Accuracy (%)</th>
<th>Value</th>
<th>PA of</th>
<th>UA of</th>
<th>Overall</th>
<th>Kappa</th>
<th>No. of Support</th>
<th>Confidence</th>
<th>Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>sets</td>
<td>Samples (%)</td>
<td>Alarms</td>
<td>Error</td>
<td>Macro</td>
<td>Micro</td>
<td>F1-changed</td>
<td>F1-changed</td>
<td>changed</td>
<td>un-changed</td>
<td>changed</td>
<td>un-changed</td>
<td>Accuracy (%)</td>
<td>Value</td>
<td>(%)</td>
<td>(%)</td>
<td>Time (sec.)</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3487</td>
<td>1494</td>
<td>4981</td>
<td>0.9440</td>
<td>0.9446</td>
<td>0.932</td>
<td>0.965</td>
<td>96.1</td>
<td>0.888</td>
<td>7</td>
<td>0.05</td>
<td>22</td>
<td>7.44</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2820</td>
<td>1953</td>
<td>4773</td>
<td>0.9474</td>
<td>0.9477</td>
<td>0.921</td>
<td>0.988</td>
<td>96.17</td>
<td>0.894</td>
<td>10</td>
<td>0.01</td>
<td>22</td>
<td>7.58</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mexico</td>
<td>10</td>
<td>2668</td>
<td>2065</td>
<td>0.9441</td>
<td>0.9435</td>
<td>0.913</td>
<td>0.988</td>
<td>96.19</td>
<td>0.896</td>
<td>11</td>
<td>0.005</td>
<td>14</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>2566</td>
<td>2077</td>
<td>0.9443</td>
<td>0.9463</td>
<td>0.917</td>
<td>0.989</td>
<td>96.22</td>
<td>0.898</td>
<td>11</td>
<td>0.005</td>
<td>20</td>
<td>8.20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>2630</td>
<td>2099</td>
<td>4639</td>
<td>0.9491</td>
<td>0.9494</td>
<td>0.915</td>
<td>0.988</td>
<td>96.22</td>
<td>0.898</td>
<td>13</td>
<td>0.004</td>
<td>27</td>
<td>8.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mexico</td>
<td>10</td>
<td>1267</td>
<td>2085</td>
<td>0.923</td>
<td>0.9235</td>
<td>0.830</td>
<td>0.888</td>
<td>96.31</td>
<td>0.847</td>
<td>5</td>
<td>0.1</td>
<td>25</td>
<td>3.54</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1110</td>
<td>1954</td>
<td>0.9232</td>
<td>0.9256</td>
<td>0.858</td>
<td>0.885</td>
<td>96.41</td>
<td>0.858</td>
<td>8</td>
<td>0.03</td>
<td>26</td>
<td>3.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sardinia</td>
<td>10</td>
<td>1126</td>
<td>829</td>
<td>0.926</td>
<td>0.9285</td>
<td>0.848</td>
<td>0.886</td>
<td>96.41</td>
<td>0.858</td>
<td>11</td>
<td>0.01</td>
<td>26</td>
<td>4.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1151</td>
<td>792</td>
<td>0.937</td>
<td>0.9256</td>
<td>0.846</td>
<td>0.890</td>
<td>96.42</td>
<td>0.858</td>
<td>12</td>
<td>0.005</td>
<td>25</td>
<td>3.84</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Greece</td>
<td>20</td>
<td>1126</td>
<td>798</td>
<td>1924</td>
<td>0.9340</td>
<td>0.9334</td>
<td>0.848</td>
<td>0.885</td>
<td>96.44</td>
<td>0.863</td>
<td>13</td>
<td>0.005</td>
<td>25</td>
<td>4.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4061</td>
<td>602</td>
<td>4663</td>
<td>0.6481</td>
<td>0.6932</td>
<td>0.218</td>
<td>0.648</td>
<td>96.22</td>
<td>0.299</td>
<td>7</td>
<td>0.05</td>
<td>25</td>
<td>7.53</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3958</td>
<td>602</td>
<td>4560</td>
<td>0.6647</td>
<td>0.7079</td>
<td>0.238</td>
<td>0.670</td>
<td>96.26</td>
<td>0.334</td>
<td>13</td>
<td>0.01</td>
<td>24</td>
<td>7.51</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>3948</td>
<td>593</td>
<td>4541</td>
<td>0.6622</td>
<td>0.7094</td>
<td>0.997</td>
<td>0.984</td>
<td>96.26</td>
<td>0.338</td>
<td>11</td>
<td>0.005</td>
<td>22</td>
<td>7.52</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>3921</td>
<td>605</td>
<td>4526</td>
<td>0.672</td>
<td>0.7144</td>
<td>0.997</td>
<td>0.984</td>
<td>96.27</td>
<td>0.344</td>
<td>13</td>
<td>0.005</td>
<td>22</td>
<td>8.43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>3700</td>
<td>732</td>
<td>4452</td>
<td>0.6945</td>
<td>0.7237</td>
<td>0.997</td>
<td>0.985</td>
<td>98.38</td>
<td>0.390</td>
<td>14</td>
<td>0.001</td>
<td>22</td>
<td>8.71</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 2: Confusion matrices for the change-detection maps obtained by the presented method on the Mexico data set.

<table>
<thead>
<tr>
<th>True class</th>
<th>Changed class</th>
<th>Unchanged class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changed class</td>
<td>22112</td>
<td>1494</td>
</tr>
<tr>
<td>Unchanged class</td>
<td>3487</td>
<td>235051</td>
</tr>
</tbody>
</table>

Training sample = 1%, MA = 3487, FA = 1494, OE = 4981, Avg. Micro F1 = 0.9446, Avg. Macro F1 = 0.9440, PA(changed) = 0.863, PA(unchanged) = 0.993, UA(changed) = 0.937, UA(unchanged) = 0.985, OA = 98.1%, Kappa = 0.888, No. of rules = 7 and Execution time = 7.44 sec. (a)

<table>
<thead>
<tr>
<th>True class</th>
<th>Changed class</th>
<th>Unchanged class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changed class</td>
<td>22779</td>
<td>1953</td>
</tr>
<tr>
<td>Unchanged class</td>
<td>2820</td>
<td>234592</td>
</tr>
</tbody>
</table>

Training sample = 5%, MA = 2820, FA = 1953, OE = 4773, Avg. Micro F1 = 0.9477, Avg. Macro F1 = 0.9474, PA(changed) = 0.889, PA(unchanged) = 0.991, UA(changed) = 0.921, UA(unchanged) = 0.988, OA = 98.17%, Kappa = 0.894, No. of rules = 10 and Execution time = 7.58 sec. (b)

<table>
<thead>
<tr>
<th>True class</th>
<th>Changed class</th>
<th>Unchanged class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changed class</td>
<td>22931</td>
<td>2065</td>
</tr>
<tr>
<td>Unchanged class</td>
<td>2668</td>
<td>234480</td>
</tr>
</tbody>
</table>

Training sample = 10%, MA = 2668, FA = 2065, OE = 4733, Avg. Micro F1 = 0.9483, Avg. Macro F1 = 0.9481, PA(changed) = 0.895, PA(unchanged) = 0.991, UA(changed) = 0.918, UA(unchanged) = 0.988, OA = 98.19%, Kappa = 0.896, No. of rules = 11 and Execution time = 8 sec. (c)

<table>
<thead>
<tr>
<th>True class</th>
<th>Changed class</th>
<th>Unchanged class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changed class</td>
<td>23033</td>
<td>2077</td>
</tr>
<tr>
<td>Unchanged class</td>
<td>2566</td>
<td>234468</td>
</tr>
</tbody>
</table>

Training sample = 15%, MA = 2566, FA = 2077, OE = 4643, Avg. Micro F1 = 0.9493, Avg. Macro F1 = 0.9493, PA(changed) = 0.899, PA(unchanged) = 0.991, UA(changed) = 0.917, UA(unchanged) = 0.989, OA = 98.22%, Kappa = 0.898, No. of rules = 11 and Execution time = 8.20 sec. (d)

<table>
<thead>
<tr>
<th>True class</th>
<th>Changed class</th>
<th>Unchanged class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changed class</td>
<td>23069</td>
<td>2009</td>
</tr>
<tr>
<td>Unchanged class</td>
<td>2630</td>
<td>234536</td>
</tr>
</tbody>
</table>

Training sample = 20%, MA = 2630, FA = 2009, OE = 4639, Avg. Micro F1 = 0.9491, Avg. Macro F1 = 0.9491, PA(changed) = 0.896, PA(unchanged) = 0.991, UA(changed) = 0.919, UA(unchanged) = 0.988, OA = 98.22%, Kappa = 0.898, No. of rules = 13 and Execution time = 8.8 sec. (e)
Table 3: Confusion matrices for the change-detection maps obtained by the presented method on Sardinia data set.

<table>
<thead>
<tr>
<th>True class</th>
<th>Estimated class</th>
<th>Changed class</th>
<th>Unchanged class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changed class</td>
<td>6213</td>
<td>817</td>
<td></td>
</tr>
<tr>
<td>Unchanged class</td>
<td>1267</td>
<td>115303</td>
<td></td>
</tr>
</tbody>
</table>

Training sample=1%, MA= 1267, FA= 817, OE=2085, Avg. Micro F1= 0.9245, Avg. Macro F1=0.923, PA(changed)=0.830, PA(unchanged)=0.992, UA(changed)=0.888, UA(unchanged)=0.989, OA=98.31%, Kappa= 0.847, No. of rules=5 and Execution time=3.54 sec (a)

<table>
<thead>
<tr>
<th>True class</th>
<th>Estimated class</th>
<th>Changed class</th>
<th>Unchanged class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changed class</td>
<td>6370</td>
<td>844</td>
<td></td>
</tr>
<tr>
<td>Unchanged class</td>
<td>1110</td>
<td>115276</td>
<td></td>
</tr>
</tbody>
</table>

Training sample=5%, MA=1110, FA= 844, OE=1954, Avg. Micro F1= 0.9296, Avg. Macro F1=0.9292, PA(changed)= 0.851, PA(unchanged)=0.992, UA(changed)=0.885, UA(unchanged)=0.990, OA=98.41%, Kappa=0.858, No. of rules=8 and Execution time=3.6 sec (b)

<table>
<thead>
<tr>
<th>True class</th>
<th>Estimated class</th>
<th>Changed class</th>
<th>Unchanged class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changed class</td>
<td>6354</td>
<td>829</td>
<td></td>
</tr>
<tr>
<td>Unchanged class</td>
<td>1126</td>
<td>115291</td>
<td></td>
</tr>
</tbody>
</table>

Training sample=10%, MA= 1126, FA=829, OE=1955, Avg. Micro F1=0.9295, Avg. Macro F1=0.9290, PA(changed)=0.849, PA(unchanged)=0.992, UA(changed)=0.886, UA(unchanged)=0.990, OA=98.41%, Kappa=0.858, No. of rules=11 and Execution time=4.4 sec. (c)

<table>
<thead>
<tr>
<th>True class</th>
<th>Estimated class</th>
<th>Changed class</th>
<th>Unchanged class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changed class</td>
<td>6329</td>
<td>792</td>
<td></td>
</tr>
<tr>
<td>Unchanged class</td>
<td>1151</td>
<td>115328</td>
<td></td>
</tr>
</tbody>
</table>

Training sample=15%, MA=1151, FA=792, OE=1943, Avg. Micro F1=0.9296, Avg. Macro F1=0.9291, PA(changed)=0.846, PA(unchanged)=0.993, UA(changed)=0.890, UA(unchanged)=0.990, OA=98.42%, Kappa=0.858, No. of rules=12 and Execution time=3.94 sec (d)

<table>
<thead>
<tr>
<th>True class</th>
<th>Estimated class</th>
<th>Changed class</th>
<th>Unchanged class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changed class</td>
<td>6354</td>
<td>798</td>
<td></td>
</tr>
<tr>
<td>Unchanged class</td>
<td>1126</td>
<td>115322</td>
<td></td>
</tr>
</tbody>
</table>

Training sample=20%, MA=1126, FA=798, OE=1924, Avg. Micro F1=0.9304, Avg. Macro F1=0.9300, PA(changed)=0.849, PA(unchanged)=0.993, UA(changed)=0.889, UA(unchanged)=0.990, OA=98.44%, Kappa=0.86, No. of rules=13 and Execution time=4.12 sec (e)
Table 4: Confusion matrices for the change-detection maps obtained by the presented method on Greece data set.

<table>
<thead>
<tr>
<th>True class</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changed class</td>
<td>1536</td>
<td>602</td>
</tr>
<tr>
<td>Unchanged class</td>
<td>4061</td>
<td>256345</td>
</tr>
</tbody>
</table>

Training sample=1%, MA=4061, FA=602, OE=4663, Avg. Micro F1=0.6952, Avg. Macro F1=0.6481, PA(changed)=0.218, PA(unchanged)=0.997, UA(changed)=0.648, UA(unchanged)=0.984, OA=98.22%, Kappa=0.299, No. of rules=7 and Execution time=7.33 sec. (a)

<table>
<thead>
<tr>
<th>True class</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changed class</td>
<td>1239</td>
<td>602</td>
</tr>
<tr>
<td>Unchanged class</td>
<td>3958</td>
<td>256345</td>
</tr>
</tbody>
</table>

Training sample=5%, MA=3958, FA=602, OE=4560, Avg. Micro F1=0.7079, Avg. Macro F1=0.6657, PA(changed)=0.238, PA(unchanged)=0.997, UA(changed)=0.679, UA(unchanged)=0.984, OA=98.26%, Kappa=0.334, No. of rules=13 and Execution time=7.91 sec. (b)

<table>
<thead>
<tr>
<th>True class</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changed class</td>
<td>1249</td>
<td>593</td>
</tr>
<tr>
<td>Unchanged class</td>
<td>3948</td>
<td>256354</td>
</tr>
</tbody>
</table>

Training sample=10%, MA=3948, FA=593, OE=4541, Avg. Micro F1=0.7094, Avg. Macro F1=0.6682, PA(changed)=0.240, PA(unchanged)=0.997, UA(changed)=0.984, UA(unchanged)=0.682, OA=98.26%, Kappa=0.338, No. of rules=11 and Execution time=7.82 sec. (c)

<table>
<thead>
<tr>
<th>True class</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changed class</td>
<td>1276</td>
<td>605</td>
</tr>
<tr>
<td>Unchanged class</td>
<td>3921</td>
<td>256342</td>
</tr>
</tbody>
</table>

Training sample=15%, MA=3921, FA=605, OE=4526, Avg. Micro F1=0.7114, Avg. Macro F1=0.6712, PA(changed)=0.997, PA(unchanged)=0.245, UA(changed)=0.984, UA(unchanged)=0.684, OA=98.27%, Kappa=0.344, No. of rules=13 and Execution time=8.43 sec. (d)

<table>
<thead>
<tr>
<th>True class</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changed class</td>
<td>1497</td>
<td>732</td>
</tr>
<tr>
<td>Unchanged class</td>
<td>3700</td>
<td>256215</td>
</tr>
</tbody>
</table>

Training sample=20%, MA=3700, FA=732, OE=4432, Avg. Micro F1=0.7237, Avg. Macro F1=0.6945, PA(changed)=0.997, PA(unchanged)=0.288, UA(changed)=0.985, UA(unchanged)=0.674, OA=98.30%, Kappa=0.390, No. of rules=14 and Execution time=8.71 sec. (e)
Figure 7: Change-detection maps obtained using CBA technique on Greece area dataset by (a) 1% of training samples (b) 5% of training samples, (c) 10% of training sample, (d) 15% of training samples and (e) 20% of training samples.

Figure 5, 6, 7 depict the change-detection maps. A visual comparison points out that the presented approach generates a smooth change-detection map. From the Table 1 one can also see that the presented CBA based technique generates a clear change detection results using very less number of labeled training samples.

5. Conclusions and Future works

Here in this paper, an algorithm based on classification based association for change detection in multi-spectral, multi-temporal images remotely-sensed images is presented. To evaluate the performance of the presented algorithm, it tested on few remote-sensed image data sets with various measures discussed earlier. Note that in case of the presented method a few number of parameters to be set, such as support and confidence [13], depending upon the number of training samples. Experimental results show that the presented method performs fairly well both in terms of the classification quality and execution time. In future this method can also be implemented by semi-supervised method and the presented algorithms can be tested on other kinds of (hyperspectral) remotely sensed image data. The results can be compared with some other existing methodologies.

References