

Semi-supervised Learning with Multilayer Perceptron for Detecting Changes of Remote Sensing Images

Swarnajyoti Patra,¹ Susmita Ghosh^{1,*}, and Ashish Ghosh²

¹ Department of Computer Science and Engineering
Jadavpur University, Kolkata 700032, India
susmitaghoshju@gmail.com

² Machine Intelligence Unit and Center for Soft Computing Research
Indian Statistical Institute
203 B. T. Road, Kolkata 700108, India

Abstract. A context-sensitive change-detection technique based on semi-supervised learning with multilayer perceptron is proposed. In order to take contextual information into account, input patterns are generated considering each pixel of the difference image along with its neighbors. A heuristic technique is suggested to identify a few initial labeled patterns without using ground truth information. 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. Experimental results, carried out on two multispectral and multitemporal remote sensing images, confirm the effectiveness of the proposed approach.

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 different times [1]. Such a problem plays an important role in many different domains like studies on land-use/land-cover dynamics [2], burned area assessment [3], analysis of deforestation processes [4], identification of vegetation changes [5] *etc.* Since all these applications usually require an analysis of large areas, development of automatic change-detection techniques is of high relevance in order to reduce the effort required by manual image analysis.

In the literature [2,3,4,5,6,7,8,9,10], several supervised and unsupervised techniques for detecting changes in remote-sensing images have been proposed. The supervised methods need “ground truth” information whereas the unsupervised approaches perform change detection without using any additional information, besides the raw images considered. Besides these two methods of learning, another situation may arise where only a few training patterns are available. The semi-supervised learning [11] comes into play in such a situation. In this article

* Corresponding author.

we propose a context-sensitive semi-supervised change-detection technique based on multilayer perceptron (MLP) that automatically discriminates the changed and unchanged pixels of the difference image. In order to take care of the contextual information, the input patterns are generated considering each pixel of the difference image along with its neighbors. Initially the network is trained using a small set of labeled data. We suggest a technique to initially identify some labeled patterns automatically. The unlabeled patterns are iteratively processed by the MLP to obtain a soft class label for each of them.

2 Proposed Change Detection Technique

In this section we propose a context-sensitive semi-supervised technique that automatically discriminates the changed and unchanged pixels of the difference image. Few labeled patterns are identified by applying a suggested heuristic technique without using ground truth information. In the following subsections we will describe the steps involved in the proposed change detection technique.

2.1 Generation of Input Patterns

To generate the input patterns, we first produce the difference image by considering the multitemporal images in which the difference between the two considered acquisitions are highlighted. The most popular Change Vector Analysis (CVA) technique is used here to generate the difference image. Let us consider two co-registered and radiometrically corrected γ -spectral band images X_1 and X_2 , of size $p \times q$, acquired over the same geographical area at two times T_1 and T_2 , and let $D = \{l_{mn}, 1 \leq m \leq p, 1 \leq n \leq q\}$ be the difference image obtained by applying the CVA technique to X_1 and X_2 . Then

$$l_{mn} = (int) \sqrt{\sum_{\alpha=1}^{\gamma} \left(l_{mn}^{\alpha(X_1)} - l_{mn}^{\alpha(X_2)} \right)^2}.$$

Here $l_{mn}^{\alpha(X_1)}$ and $l_{mn}^{\alpha(X_2)}$ are the gray values of the pixels at the spatial position (m, n) in α^{th} band of images X_1 and X_2 , respectively.

After producing the difference image, the input patterns are generated corresponding to each pixel in the difference image D , considering its spatial neighborhood of order d . In the present case 2^{nd} order neighborhood ($d = 2$) is considered, and the input vectors contain nine components considering the gray value of the pixel and the gray values of its eight neighboring pixels. So the pattern set $U = \{\mathbf{u}(1), \mathbf{u}(2), \dots, \mathbf{u}(N)\}$ contains N ($N = p \times q$) pattern vectors in nine-dimension feature space. The patterns generated by the above technique help us to produce better change detection map as they take some spatial contextual information from the difference image.

2.2 Description of the MLP

The multilayer perceptron [12] has one input, one output and one or more hidden layers. Neurons/nodes in one layer of the network are connected to all the neurons in the next layer. Let S be the number of layers in the network and $y_j^r(n)$ denote the output signal of the j^{th} neuron in the r^{th} layer for an input pattern $\mathbf{u}(n)$, where $n = 1, 2, \dots, N$ and w_{ij}^r be the connection strength between the i^{th} neuron in the $(r-1)^{th}$ layer and j^{th} neuron in the r^{th} layer. For an input pattern vector $\mathbf{u}(n)$, the output value of neuron j in the input layer is defined as $y_j^0(n) = u_j(n)$, which is sent to the first hidden layer as an input signal. A neuron j in the r^{th} ($r \geq 1$) layer takes input signal $v_j^r(n) = \sum_i y_i^{r-1}(n) \cdot w_{ij}^r + w_{0j}^r$, where w_{0j}^r is the connection strength between a fixed unit (bias) to neuron j , and produces an output $y_j^r(n) = f(v_j^r(n))$. The activation function $f(\cdot)$ mapped the output sigmoidally between 0 to 1. The network is trained using backpropagation algorithm [12] that iteratively adjusts coupling strengths (weights) in the network to minimize the sum-square error $\sum_{n=1}^N \sum_{j=1}^C (y_j^{S-1}(n) - t_j(n))^2$, where $y_j^{S-1}(n)$ and $t_j(n)$ are the predicted and desired value of the output layer neuron j for input pattern $\mathbf{u}(n)$, respectively. As the generated patterns have nine features and belong to either changed class or unchanged class, the architecture of the MLP used here has nine neurons in the input layer and two neurons in the output layer.

2.3 Labeled (Training) Pattern Generation

MLP needs labeled patterns for learning. In this section we suggest a technique to automatically identify some patterns which either belong to changed or unchanged class without using ground truth information.

As component values of the generated pattern vectors contain gray values of pixels in the difference image, the patterns whose component values are very low belong to unchanged class and the patterns whose component values are very high belong to changed class. To identify these patterns automatically, K-means (K=2) clustering algorithm [13] is used. Let lc and uc be the two centroid obtained by K-means algorithm in nine-dimensional feature space. We also considered two other points $lb(0, \dots, 0)$, the possible minimum component values of the patterns near to lc and $ub(255, \dots, 255)$, the possible maximum component values of the patterns which is near to uc in the same feature space (see Fig. 1). A pattern is assigned to the unchanged class if it is inside the hypersphere whose center is at lb and radius is the distance between lb and lc ; it is assigned to the changed class if it is inside the hypersphere whose center is at ub and radius is the distance between ub and uc else it is considered as unlabeled. The pattern set U is represented as $U = \{(\mathbf{u}(n), \mathbf{t}(n)), n = 1, 2, \dots, N\}$, where $\mathbf{u}(n)$ is the n^{th} input pattern vector and $\mathbf{t}(n)$ is the target vector of the corresponding input pattern. The target vector $\mathbf{t}(n)$ where $\mathbf{t}(n) = \{[t_1(n), t_2(n)] \mid t_i(n) \in (0, 1), \forall n\}$ represents the changed class when $\mathbf{t}(n) = [1, 0]$, unchanged class when $\mathbf{t}(n) = [0, 1]$ and unlabeled pattern when $\mathbf{t}(n) = [0, 0]$.

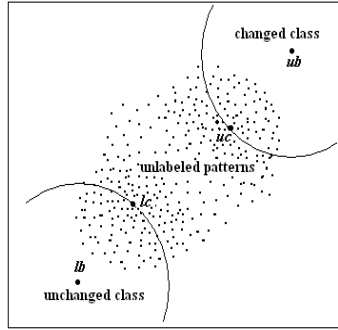


Fig. 1. Labeled and unlabeled patterns in two-dimensional feature space

2.4 Labeling Unknown Patterns

We train the network first using the labeled patterns that are automatically identified. The decision classes are considered as fuzzy sets [14] and we assume that the network’s output values provide degree of membership to the fuzzy sets. Let us suppose the n^{th} input pattern is presented to the network. The membership value $\mu_j(\mathbf{u}(n))$ of the n^{th} input pattern $\mathbf{u}(n)$ to the j^{th} fuzzy set is then given by the output $y_j^2(\mathbf{u}(n))$ of the j^{th} ($j = 1, 2$) output neuron. Contrast within the set of membership values $\mu_j(\mathbf{u}(n))$ is then increased [15] as follows:

$$\mu_j(\mathbf{u}(n)) = \begin{cases} 2[\mu_j(\mathbf{u}(n))]^2, & 0 \leq \mu_j(\mathbf{u}(n)) \leq 0.5 \\ 1 - 2[1 - \mu_j(\mathbf{u}(n))]^2, & 0.5 < \mu_j(\mathbf{u}(n)) \leq 1.0 \end{cases} \quad (1)$$

Now we find out the k_{nn} nearest neighbors for each unlabeled pattern. Finding out the k_{nn} nearest neighbors considering all patterns is a time consuming task. To reduce this time complexity, instead of considering all the patterns we found out k_{nn} nearest neighbors for an unlabeled pattern corresponding to a pixel of the difference image by considering only the patterns generated by its surrounding pixels. Let M^n be the set of indices of the k_{nn} nearest neighbors of the unlabeled pattern $\mathbf{u}(n)$. Then the target vector $\mathbf{t}(n)$ for the unlabeled pattern $\mathbf{u}(n)$ is computed by

$$\mathbf{t}(n) = \left[\frac{\sum_{i \in M^n} t_1(i)}{k_{nn}}, \frac{\sum_{i \in M^n} t_2(i)}{k_{nn}} \right] \quad (2)$$

where for unlabeled pattern $\mathbf{t}_j(n) = \mu_j(\mathbf{u}(n))$ and for labeled pattern $\mathbf{t}_j(i) = 0$ or 1 , $j = 1, 2$.

2.5 Learning Algorithm

The learning technique used here is inspired by the principle used in [15] and is given bellow.

Learning algorithm of the network

- Step 1: Train the network using labeled patterns only.
- Step 2: Assign soft class labels to each unlabeled or softly labeled pattern by passing it through the trained network.
- Step 3: Re-estimate these soft class labels using Eq. (1) and Eq. (2).
- Step 4: Train the network using both the labeled and the softly labeled patterns.
- Step 5: If the sum of square error obtained from the network becomes constant or the number of iterations exceeds some given number then goto Step 6; else goto Step 2.
- Step 6: Stop.

3 Description of the Data Sets

In our experiments we used two data sets representing areas of Mexico and Sardinia Island of Italy. For Mexico data multispectral images (of size 512×512) acquired by the Landsat Enhanced Thematic Mapper Plus (ETM+) sensor of the Landsat-7 satellite, in an area of Mexico on April 2000 and May 2002 were considered. Between the two acquisition dates, fire destroyed a large portion of the vegetation in the considered region. As band 4 is more effective to locate the burned area, we generated the difference image by considering only spectral band 4. Sardinia data is related to Mulargia lake of Sardinia Island. It consists of Landsat-5 TM images (of size 412×300) acquired on September 1995 and July 1996, respectively. Between the two acquisition dates the water level in the lake is changed. We applied the CVA technique [1] on spectral bands 1, 2, 4, and 5 of the two multispectral images, as preliminary experiments had demonstrated that the above set of channels contains useful information of the change in water level. The available ground truth concerning the location of the changes was used to prepare the “reference map” (Fig. 2) which was useful to assess change-detection errors.

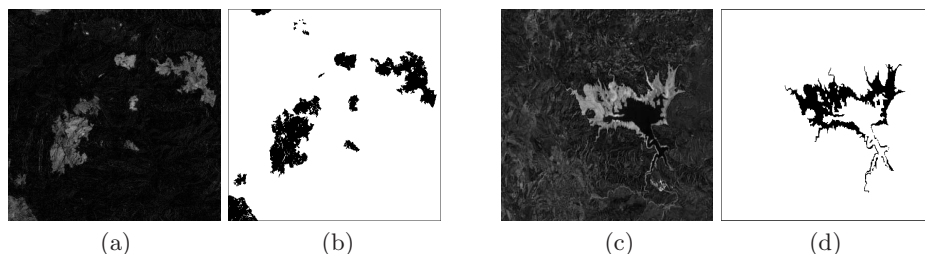


Fig. 2. Mexico data: (a) Difference image and (b) Corresponding reference map; Sardinia data: (c) Difference image and (d) Corresponding reference map

4 Experimental Results

In order to establish the effectiveness of the proposed technique, the present experiment compares the change-detection result provided by the proposed method with a context-insensitive Manual Trial and Error Thresholding (MTET) technique, the K-means clustering [13] technique and a context sensitive technique presented in [3] based on the combined use of the EM algorithm and Markov Random Fields (MRF) (we refer to it as EM+MRF technique). The MTET technique generates a minimum error change-detection map under the hypothesis of spatial independence among pixels by finding a minimum error decision threshold for the difference image. The minimum error decision threshold is obtained by computing change-detection errors (with the help of the reference map) for all values of the decision threshold. K-means clustering algorithm is applied on the generated patterns (as described in Sec. 2.1) with $K = 2$. Comparisons were carried out in terms of both overall change-detection error and number of false alarms (i.e., unchanged pixels identified as changed ones) and missed alarms (i.e., changed pixels categorized as unchanged ones).

In the present experiment the architecture of MLP is 9 : 8 : 2 i.e., the network has 9 input neurons, 8 hidden neurons in a single hidden layer and 2 output neurons. To find out k_{nn} nearest neighbors for each input pattern we have taken 50×50 window and the value of k_{nn} is taken as 8.

4.1 Analysis of Results

Table 1 shows that the overall error produced (using both Mexico and Sardinia data) by the proposed technique is much smaller than that produced by the MTET technique. Figs. 3 and 4 depict the change-detection maps. A visual comparison points out that the proposed approach generates a more smooth change-detection map compared to the MTET procedure. From the tables one can also see that the proposed MLP based technique generates better change-detection results than the result produced by the K-means technique. The best result obtained by existing EM+MRF technique with a specific value of β of MRF is also close to the result obtained by the proposed technique.

Table 1. Overall error, missed alarms and false alarms resulting by MTET, K-means, EM+MRF, and the proposed technique

Techniques	Mexico data			Sardinia Island data		
	Missed alarms	False alarms	Overall error	Missed alarms	False alarms	Overall error
MTET	2404	2187	4591	1015	875	1890
K-mean	3108	665	3773	637	1881	2518
EM+MRF	946	2257	3203	592	1108	1700
Proposed	2602	703	3305	1294	303	1597

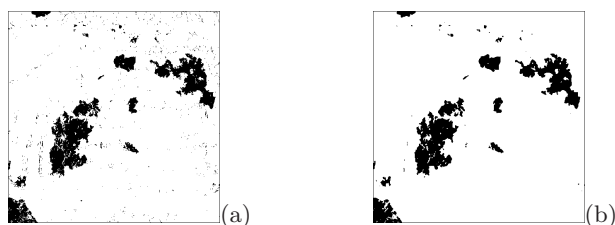


Fig. 3. Change-detection maps obtained for the data set related to the Mexico area using (a) MTET technique, (b) proposed technique

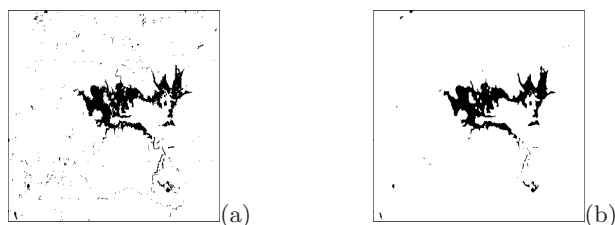


Fig. 4. Change-detection maps obtained for the data set related to the Sardinia area using (a) MTET technique, (b) proposed technique

5 Discussion and Conclusions

In this article, a semi-supervised and automatic context-sensitive technique for change detection in multitemporal images is proposed. The technique discriminates the changed and unchanged pixels in the difference image by using a multilayer perceptron. The number of neurons in the input layer is equal to the dimension of the input patterns and the number of neurons in the output layer is two. The input patterns are generated considering each pixel in the difference image along with its neighboring pixels, in order to take into account the spatial contextual information. On the basis of the characteristics of these input patterns, a heuristic technique is suggested to automatically identify a few input patterns that have very high probability to belong either to changed or to unchanged class. Depending on these labeled patterns and assuming that the less dense region can act as separator between the classes, a semi-supervised learning algorithm based on MLP is used to assign the soft class label for each unlabeled pattern.

The presented technique shows the following advantages: (i) it is distribution free, i.e., like EM+MRF model presented in [3] it does not require any explicit assumption on the statistical model of the distributions of classes of changed and unchanged pixels, (ii) it does not require human efforts to identify the labeled patterns, i.e., the labeled patterns are identified heuristically without using any ground truth information. Like other semi-supervised cases, the time requirement of this technique is little more. It is worth noting that for the considered kind of application it is not fundamental to produce results in real time.

Experimental results confirm the effectiveness of the proposed approach. The presented technique significantly outperforms the standard optimal-manual context-insensitive MTET technique and K-means technique. The proposed technique provides comparable overall change detection error to the best result achieved with the context-sensitive EM+MRF technique.

Acknowledgements

Authors like to thank the Department of Science and Technology, India and University of Trento, Italy, the sponsors of the ITPAR program and Prof. L. Bruzzone, the Italian collaborator of this project, for providing the data.

References

1. Singh, A.: Digital change detection techniques using remotely sensed data. *Int. J. Remote Sensing* 10, 989–1003 (1989)
2. Liu, X., Lathrop, R.G.: Urban change detection based on an artificial neural network. *Int. J. Remote Sensing* 23, 2513–2518 (2002)
3. Bruzzone, L., Prieto, D.F.: Automatic analysis of the difference image for unsupervised change detection. *IEEE Trans. Geosci. Remote Sensing* 38, 1171–1182 (2000)
4. Hame, T., Heiler, I., Miguel-Ayanz, J.S.: An unsupervised change detection and recognition system for forestry. *Int. J. Remote Sensing* 19, 1079–1099 (1998)
5. Chavez, P.S., MacKinnon, D.J.: Automatic detection of vegetation changes in the southwestern United States using remotely sensed images. *Photogramm. Eng. Remote Sensing* 60, 1285–1294 (1994)
6. Radke, R.J., Andra, S., Al-Kofahi, O., Roysam, B.: Image change detection algorithms: A systematic survey. *IEEE Trans. Image Processing* 14, 294–307 (2005)
7. Canty, M.J.: *Image Analysis, Classification and Change Detection in Remote Sensing*. CRC Press, Taylor & Francis (2006)
8. Kasetkasem, T., Varshney, P.K.: An image change-detection algorithm based on Markov random field models. *IEEE Trans. Geosci. Remote Sensing* 40, 1815–1823 (2002)
9. Ghosh, S., Bruzzone, L., Patra, S., Bovolo, F., Ghosh, A.: A context-sensitive technique for unsupervised change detection based on Hopfield-type neural networks. *IEEE Trans. Geosci. Remote Sensing* 45, 778–789 (2007)
10. Patra, S., Ghosh, S., Ghosh, A.: Unsupervised change detection in remote-sensing images using modified self-organizing feature map neural network. In: *Int. Conf. on Computing: Theory and Applications (ICCTA-2007)*, pp. 716–720. IEEE Computer Society Press, Los Alamitos (2007)
11. Seeger, M.: *Learning with labeled and unlabeled data*. Technical report, University of Edinburgh (2001)
12. Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Pearson Education, Fourth Indian Reprint (2003)
13. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. Wiley, Singapore (2001)
14. Klir, G.J., Yuan, B.: *Fuzzy sets and Fuzzy Logic - Theory and Applications*. Prentice Hall, New York (1995)
15. Verikas, A., Gelzinis, A., Malmqvist, K.: Using unlabelled data to train a multilayer perceptron. *Neural Processing Letters* 14, 179–201 (2001)