Applied Soft Computing 11 (2011) 5770–5781

Supervised and unsupervised landuse map generation from remotely sensed images using ant based systems

Anindya Halder[a], Ashish Ghosh[a,*], Susmita Ghosh[b]

[a] Center for Soft Computing Research, Indian Statistical Institute, Kolkata, India
[b] Department of Computer Science & Engineering, Jadavpur University, Kolkata, India

ARTICLE INFO

Article history:
Received 9 August 2010
Received in revised form 13 February 2011
Accepted 24 February 2011
Available online 21 March 2011

Keywords:
Landuse map
Pattern classification
Clustering
Ant colony
Aggregation pheromone

ABSTRACT

The landuse or land-cover map depicts the physical coverage of the Earth's terrestrial surface according to its use. Landuse map generation from remotely sensed images is one of the challenging tasks of remote sensing technology. In this article, motivated from group forming behavior of real ants, we have proposed two novel ant based (one supervised and one unsupervised) algorithms to automatically generate landuse map from multispectral remotely sensed images. Here supervised landuse map generation is treated as a classification task which requires some labeled patterns/pixels beforehand, whereas the unsupervised landuse map generation is treated as a clustering based image segmentation problem in the multispectral space. Investigations are carried out on four remotely sensed image data. Experimental results of the proposed algorithms are compared with corresponding popular state of the art techniques using various evaluation measures. Potentiality of the proposed algorithms are justified from the experimental outcome on a number of images.

1. Introduction

Remote sensing images provide a general reflection of the spatial characteristics for ground objects. Extraction of landuse or land-cover map information from multispectral or hyperspectral remotely sensed images is one of the important tasks of remote sensing technology. In order to automatically generate such landuse map from remotely sensed images, various pattern recognition techniques like classification and clustering can be adopted.

In real scenario, like remotely sensed images, the collection of training samples to design supervised classifier is not always possible due to the scarcity of available information and cost of acquisition of the ground truth information. A possible alternative to this, is unsupervised classification called clustering which does not require any labeled pattern. In order to generate the landuse map in unsupervised manner, remotely sensed image segmentation may be viewed as a clustering task.

Various attempts have been made for landuse prediction from remotely sensed data in supervised as well as unsupervised framework.

In the unsupervised manner, several clustering algorithms like split-and-merge [1], fuzzy k-means [2,3], neural network based methods [4,5] and scale space techniques [6] have been applied. Statistical methods like Markov Random Filed (MRF) based approach [7] and Expectation Maximization (EM) approach [8] have also been used for the purpose. Limitations of local search of the traditional clustering based methods have been improved by using genetic algorithm (GA) [9], particle swarm optimization (PSO) [10] and differential evolution [11] which exploits the global search concept.

Several attempts have also been made for remote sensing image analysis and classification using fuzzy sets [12–16], neural networks [17,18], support vector machines [19,20] and computational intelligence [21] in the supervised mode. Judicious combinations of some of these techniques are also proposed [22,23] for the purpose. Statistical models [24] are also adopted for the supervised landuse map generation.

In this article, in order to automatically generate landuse map of multispectral remotely sensed images, we have developed one supervised technique and one unsupervised (segmentation) algorithm based on aggregation pheromone density [25,26] which is inspired by the natural behavior of real ants and other social insects [27].

The social insects' behaviors such as finding the best food source, building of optimal nest structure, brooding, protecting the larva, guarding, etc. show intelligent behavior on the swarm level [27,28]. A swarms' behavior is determined not only by the behavior of itself, but also the interactions among individuals play a vital role in shaping the swarms' behavior [27,28]. Computational modeling of swarms' behavior is found to be useful in various

* Corresponding author.
E-mail address: ash@isical.ac.in (A. Ghosh).

application domains like function optimization [25,26], finding optimal routes [29], scheduling [29], image and data analysis [30]. Different applications have been originated from the study of different types of swarms. The most popular among them are ant colonies and bird flocks [27]. *Ant Colony Optimization* (ACO) [29] and *Aggregation Pheromone Systems* (APS) [25,26] are computational algorithms modeled on the behavior of ant colonies. ACO [29] algorithm is designed to emulate ants' behavior of laying pheromone on the ground (while moving) to solve optimization problems. Pheromone is a type of chemical emitted by an organism to communicate between members of the same species. Pheromone, which is responsible for clumping or clustering behavior in a species, brings individuals into closer proximity, is known as aggregation pheromone. Thus, aggregation pheromone causes individuals to aggregate around good positions which in turn produces more pheromone to attract individuals of the same species. In APS [25,26] (a variant of ACO) such behavior of ants is used to solve real parameter optimization problems. A model used for solving continuous optimization problems [31] was also proposed as an extension of ant colony optimization (ACO).

Though a large number of techniques exists for ant based unsupervised classification (i.e., clustering) in the literature [32], only few attempts have been made for (supervised) classification. *AntMiner* is the first of this kind, proposed by Parpinelli et al. [33] to extract '*if-then*' classification rule from categorical data. Liu et al. further extended the algorithm to reduce the computational complexity in *AntMiner2* [34] and to increase the classification accuracy in *AntMiner3* [35]. Later Martens et al. in *AntMiner+* [36] modified the existing versions of *AntMiner*. All the previously proposed ant based algorithms are based on the extraction of '*if-then*' classification rule.

As mentioned earlier, *Aggregation Pheromone Systems* [25,26] are used for continuous function optimization where aggregation pheromone density is defined by a function in the feature space. Inspired by the aggregation pheromone system found in ants and other similar agents, in some of our earlier works, attempts are made for solving clustering [37], classification [38], image segmentation [30] for gray label bechmark images with encouraging results. Motivated from the earlier research, in the present article supervised and unsupervised landuse maps are generated using aggregation pheromone based techniques [37,30,38].

The rest of the article is organized as follows. Section 2 describes the detailed descriptions of the proposed methodologies for supervised and unsupervised landuse map generation of remotely sensed images. Details of the experiments are given in Section 3, and finally conclusions are drawn in Section 4.

2. Proposed methodologies

In this section we will present how landuse map can be generated from a given multispectral remotely sensed image in supervised and unsupervised manner using the concept of aggregation pheromone density.

2.1. Supervised method

As mentioned in the previous section, aggregation pheromone brings individuals into closer proximity. This group formation nature of aggregation pheromone (found in natural behavior of real ants) is being used as the basis of the proposed technique. Here each pixel/data pattern is considered as an ant, and the training patterns (ants) form several colonies or homogeneous groups depending on the number of classes present in the data set. Each ant (in the group) emits pheromone around its local neighborhood. The intensity of pheromone (emitted by an individual ant) is maximum at the position where the ant is situated and it decays uniformly with distance from the said position. Hence pheromone intensity is modeled by the Gaussian function considering the position of the ant as the center of Gaussian. When a new test pattern (ant) comes into the system it tries to join to one of the existing colonies/groups. A new ant will move towards a colony for which average aggregation pheromone density (at the location of that new ant) is higher than that of the other colonies; and hence eventually the new ant will join that colony. Here average aggregation pheromone density of a colony is the average of the cumulative effect of pheromone intensity (at the location of the test ant) emitted by each individual ant belonging to that colony. Thereby each new individual ant will join a particular colony. This process continues until all the test patterns (ants) are assigned to some colonies. As opposed to the other existing ant based classification methods [33–36], here no '*if-then*' rule is extracted, rather the proposed algorithm classify each new test (pattern) ant by computing per colony average aggregation pheromone density deposited by the ants in the (already formed) colony (training set). Hence the algorithm is more suited and directly applicable to the data sets with continuous attributes. The proposed supervised aggregation pheromone density based classification (APC) algorithm for landuse map generation is described below.

2.1.1. Aggregation pheromone density based classification

Consider a data set with m classes which, by our assumption, forms m homogeneous groups/colonies of ants or training patterns. Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_{|C_i|}$ be the training data patterns in the class C_i. These patterns are considered as a population of $|C_i|$ number of ants represented as $a_1, a_2, a_3, \ldots, a_{|C_i|}$, respectively. These ants form a group/colony and labeled as C_i. Hence, an ant $a_j \in C_i$ represents the training data pattern $\boldsymbol{x}_j$. The intensity of pheromone emitted by an individual ant a_j (located at $\boldsymbol{x}_j$) decreases with its distance from $\boldsymbol{x}_j$. Thus the pheromone intensity at a point closer to $\boldsymbol{x}_j$ is more than those at other points that are farther from it. To achieve this, the pheromone intensity emitted by ant $a_j \in C_i$ is modeled by a Gaussian distribution. The pheromone intensity deposited at any location $\boldsymbol{x}$ by an ant a_j (located at $\boldsymbol{x}_j$) is thus computed as

$$\Delta\tau(a_j, \boldsymbol{x}) = \exp^{-(d(\boldsymbol{x}_j, \boldsymbol{x})^2)/2\delta^2} \tag{1}$$

where δ denotes the spread of Gaussian function and $d(\boldsymbol{x}_j, \boldsymbol{x})$ is the Euclidean distance between $\boldsymbol{x}_j$ and $\boldsymbol{x}$.

Total aggregation pheromone density at $\boldsymbol{x}$ deposited by the entire population of $|C_i|$ ants belonging to the colony C_i is computed as

$$\Delta\tau_i(\boldsymbol{x}) = \sum_{x_j \in C_i} \exp^{-(d(\boldsymbol{x}_j, \boldsymbol{x})^2)/2\delta^2}. \tag{2}$$

Now a new (test pattern) ant a_t at $\boldsymbol{x}_t$ appears in the system. The average aggregation pheromone density (at the location of that new ant a_t) by the colony C_i is given by

$$\Delta\bar{\tau}_i(\boldsymbol{x}_t) = \frac{1}{|C_i|} \sum_{x_j \in C_i} \exp^{-(d(\boldsymbol{x}_j, \boldsymbol{x}_t)^2)/2\delta^2}. \tag{3}$$

The new ant a_t will move towards a colony for which the average aggregation pheromone density (at the location of that new ant) is higher than that of the other colonies. Hence finally the said ant will join the colony that will be governed by the following equation.

$$\mathit{ColonyLabel}(\boldsymbol{x}_t) = \operatorname*{argmax}_i(\Delta\bar{\tau}_i(\boldsymbol{x}_t)). \tag{4}$$

If ties occur, i.e. if the $(\max_i(\Delta\bar{\tau}_i(\boldsymbol{x}_t))$ is the same for more than one class (colony) i then the test pattern ant $\boldsymbol{x}_t$ is arbitrarily assigned to any of the colony for which $\Delta\bar{\tau}_i(\boldsymbol{x}_t)$ is *maximum*.

Thus each of the test ant will join a colony and the corresponding label of the colony will be the class label of that test pattern (ant). The proposed supervised aggregation pheromone density based classification (APC) algorithm is given in Algorithm 1.

Algorithm 1: Aggregation pheromone density based classifier
1: **for** each new (test) ant a_t located at $\boldsymbol{x}_t$ **do**
2: **for** each colony C_i **do**
3: Calculate the average aggregation pheromone density at location $\boldsymbol{x}_t$ due to all ants in colony C_i using Eq. (3).
4: **end for**
5: Compute the *ColonyLabel*($\boldsymbol{x}_t$) of the ant a_t by Eq. (4).//Ties are broken arbitrarily.
6: **end for**

2.2. Unsupervised method

We have considered unsupervised land-cover map generation as a segmentation problem of multispectral remotely sensed images where segmentation is treated as clustering (grouping) of pixels in a multidimensional space. Pixels belonging to a particular cluster are, therefore, spectrally similar.

Clustering is a popular technique for image segmentation [39]. As mentioned in the introduction, aggregation pheromone brings individuals into closer proximity. This group formation nature of aggregation pheromone is being used as the basic idea of the proposed algorithm. Here each ant represents a pixel of the input image. The ants move virtually with an aim to create homogenous groups of data. The amount of virtual movement of an ant towards a point is governed by the intensity of aggregation pheromone deposited by all other ants at that point. This gradual movement of ants in due course of time will result in formation of groups or clusters of homogeneous pixels (segments). The proposed technique has two parts. In the first part, from the pixels of the input image, clusters of homogeneous pixels (segments) are formed based on ants' property of depositing aggregation pheromone. The number of segments (clusters), thus formed, might be more than the desired number. So, to obtain the desired number of clusters, in the second part, agglomerative *average linkage* clustering algorithm is applied on these already formed clusters. Clusters so formed represent different homogeneous segments of an image.

While performing image segmentation for a given multispectral remotely sensed image, we group similar pixels together to form a set of spectrally similar coherent image regions. Similarity of pixels can be measured based on feature vectors. Different gray level values associated with each band corresponding to a pixel represent the feature vector of a pixel. Clustering is then performed on this set of feature vectors so as to group them. Finally, clustering result is mapped back to the original spatial domain to obtain segmented image (landuse map). The proposed unsupervised aggregation pheromone density based clustering (APC) for landuse map generation is described below.

2.2.1. Aggregation pheromone density based clustering/segmentation

Consider a data set of n patterns $\boldsymbol{x}_1, \boldsymbol{x_2}, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_n$ and a population of n-ants $a_1, a_2, a_3, \ldots, a_n$ where an ant a_i represents the data pattern $\boldsymbol{x_i}$. Here each pixel of the input image is assumed as a data point, and hence as an individual ant. Each individual ant emits pheromone around its neighborhood. Like the classification case, the pheromone intensity emitted by ant a_i is modeled by a Gaussian distribution. Hence the pheromone intensity $\Delta\tau(a_i, \boldsymbol{x})$ deposited at $\boldsymbol{x}$ by an ant a_i (located at $\boldsymbol{x_i}$) is computed by Eq. (1).

The total aggregation pheromone density at $\boldsymbol{x}$ deposited by the entire population of n ants is computed as

$$\Delta\tau(\boldsymbol{x}) = \sum_{i=1}^{n} \exp^{-\frac{d(\boldsymbol{x}_i,\boldsymbol{x})^2}{2\delta^2}}. \tag{5}$$

Now, an ant a_i which was initially at location $\boldsymbol{x}_i$ moves to the new location $\boldsymbol{x}_i'$ (which is computed using Eq. (6)) if the total aggregation pheromone density at $\boldsymbol{x}_i'$ is greater than that at $\boldsymbol{x}_i$. The movement of an ant is governed by the amount of pheromone deposited at different points in the feature space; and is defined as

$$\boldsymbol{x}_i' = \boldsymbol{x}_i + \eta \cdot \frac{Next(a_i)}{n} \tag{6}$$

where

$$Next(a_i) = \sum_{j=1}^{n} (\boldsymbol{x}_j - \boldsymbol{x}_i) \cdot \exp^{-\frac{d(\boldsymbol{x}_j,\boldsymbol{x}_i)^2}{2\delta^2}} \tag{7}$$

with η (a proportionality constant) as the step size. This process of finding a new location continues until an ant finds a location where the total aggregation pheromone density is more than its neighboring points. Once the ant a_i finds out such a point $\boldsymbol{x}_i'$, then the point $\boldsymbol{x}_i'$ is assumed to be a new potential cluster center, say $\boldsymbol{Z}_{C+1}$ (C being number of already formed clusters; note that, C is initially set to zero when no cluster exists); and the data point with which the ant was associated earlier (i.e., $\boldsymbol{x}_i$) is assigned to the cluster so formed with center $\boldsymbol{Z}_{C+1}$. Also the data points which are within a distance of $\delta/2$ from $\boldsymbol{Z}_{C+1}$ are assigned to the newly formed cluster. On the other hand, if the distance between $\boldsymbol{x}_i$ and the existing cluster center $\boldsymbol{Z}_j$ (j = 1,2,. . .,C) is less than 2δ and the ratio of their densities is greater than *threshold_density* (a predefined parameter), then the data point $\boldsymbol{x}_i$ is allocated to the cluster having cluster center $\boldsymbol{Z}_j$. Higher value of density ratio indicates that the two points are of nearly similar density and hence should belong to the same cluster. The proposed unsupervised aggregation pheromone based clustering (APC) algorithm is given in Algorithm 2.

Algorithm 2: Aggregation pheromone density based clustering
1: Initialize δ, *threshold_density*, η, C = 0
2: **for** i = 1 to n **do**
3: **if** (the data pattern $\boldsymbol{x}_i$ is not already assigned to any cluster) **then**
4: Compute $\Delta\tau(\boldsymbol{x}_i)$ using Eq. (5).
5: **label 1:**
6: Compute new location $\boldsymbol{x}_i'$ using Eq. (6)
7: Compute $\Delta\tau(\boldsymbol{x}_i')$.
8: **End of label 1**
9: **if** ($\Delta\tau(\boldsymbol{x}_i') > \Delta\tau(\boldsymbol{x}_i)$) **then**
10: Update the location of ant a_i (at $\boldsymbol{x}_i$) to $\boldsymbol{x}_i'$ and **goto label 1**.
11: **end if**
12: **if** ($C == 0$) **then**
13: //If no cluster exist
14: Consider $\boldsymbol{x}_i'$ as cluster center $\boldsymbol{Z}_1$ and increase C by one.
15: Assign all the data points within a distance of $\delta/2$ from $\boldsymbol{x}_i'$ to the newly formed cluster with center $\boldsymbol{Z}_1$.
16: **else**
17: **for** j = 1: C **do**
18: **if** ($\min(\Delta\tau(\boldsymbol{x}_i'), \Delta\tau(\boldsymbol{Z}_j))/\max(\Delta\tau(\boldsymbol{x}_i'), (\Delta\tau(\boldsymbol{Z}_j)) >$ *threshold_density* and $d(\boldsymbol{x}_i', \boldsymbol{Z}_j) < 2\delta$) **then**
19: Assign $\boldsymbol{x}_i'$ to $\boldsymbol{Z}_j$.//$\boldsymbol{Z}_j$ already exists.
20: **else**
21: Assign $\boldsymbol{x}_i'$ as a new cluster center say, $\boldsymbol{Z}_{C+1}$ and increase C by one.
22: Assign all the data points that are within a distance of $\delta/2$ from $\boldsymbol{x}_i'$ to the newly formed cluster with center Z_{C+1}.
23: **end if**
24: **end for**
25: **end if**
26: **end if**
27: **end for**

2.2.2. Merging of clusters/segments

In the proposed method (described in Section 2.2.1), we have applied the APC algorithm on the whole data set in only one pass (iteration). Depending on the parameter values, the number of clusters produced, may be more than the desired number of clusters. To obtain the desired number of clusters, we applied the average linkage agglomerative hierarchical clustering algorithm (average linkage, in short) [40] for merging them.

3. Experimental evaluations

To study the performance of the proposed methodologies, three multispectral remotely sensed image data have been used. They are described below along with the details of the experiments carried out and analysis of results.

3.1. Data sets used

The SATIMAGE data set [41] was generated from Landsat Multi Spectral Scanner image data listed as Statlog (Landsat Satellite). The data patterns used for the present investigation are a sub-area of a scene of 82×100 pixels. Each pixel value contains information from four spectral bands. The aim is to predict six different landuse classes present in the data set. The data set contains 6435 patterns with 36 attributes (4 spectral bands multiplied by 9 pixels in each neighborhood). In our experiment we have used four features (17–20) only as recommended by the database designer (i.e., the four spectral values of each pixel). As the data is completely labeled we can use this label information for evaluating the algorithms. Note that, this remotely sensed data is given in random order and certain lines of data have been removed by database designer. As a result reconstructing the original image from this data set is not possible.

IRS-1A Calcutta image (of dimension 512×512 pixels) was acquired by Indian Remote Sensing Satellite (IRS) [42]. The image used in this article is taken from Linear Imaging Self Scanner (LISS-II). LISS-II has a spatial resolution of 36.25 m × 36.25 m and the wavelength range 0.45–0.86 μm. The whole spectrum range is decomposed into four spectral bands, namely, blue, green, red, and near infrared (NIR) corresponding to bands 1, 2, 3, and 4, having wavelength 0.45–0.56 μm, 0.52–0.59 μm, 0.62–0.68 μm, and 0.77–0.86 μm, respectively. The image covers an area around the city of Calcutta (now Kolkata) with six major land-cover classes: pure water (PW), turbid water (TW), concrete area (CA), habitation (HAB), vegetation (VEG), and open spaces (OS). Few regions like Dum Dum Airport, Hooghly river, Salt Lake area, Fisheries are also labeled in the original image (Fig. 1(a)) for typical illustration.

The SPOT Calcutta image (of dimension 512×512 pixels) was obtained by Systeme Pour d'Observation de la Terre (SPOT) satellite [43] which carries an imaging device High Resolution Visible (HRV) that uses the wavelength range 0.50–0.89 μm. The whole spectrum range is decomposed into three spectral bands, namely, green band (band1), red band (band2) and near infrared band (band3) of wavelengths 0.50–0.59 μm, 0.61–0.68 μm and 0.79–0.89 μm, respectively. In this case also the same six different classes for the landuse classification of the SPOT image are considered. These are pond or fishery water (PW), turbid water (TW), concrete area (CA), habitation (HAB), vegetation (VEG) and open spaces (OS) as mentioned above. In this image also few landmark regions such as, Hooghly river (TW), Rabindra Sarobar lake (PW), Gardenrich (PW), Santragachi lake (PW), Khidderpore dock, Race Course area are labeled.

The IRS image of Bombay (of dimension 512×512 pixels) was also acquired by the LISS-II sensor. Similar to IRS-1A Calcutta image, the IRS Bombay image is also available in four bands, viz., blue, green, red, and NIR. The image covers an area of city Bombay (now Mumbai) with six major land-cover classes as follows. They are concrete area (CA), habitation (HAB), vegetation (VEG) and open spaces (OS) and turbid water 1 (TW1) and turbid water 2 (TW2). Since the water body of Arabian Sea has two distinct regions with different spectral signature, we considered so two types of turbid water.

The elongated city area is surrounded on three sides by the Arabian Sea. Towards the bottom right of the image, there are several

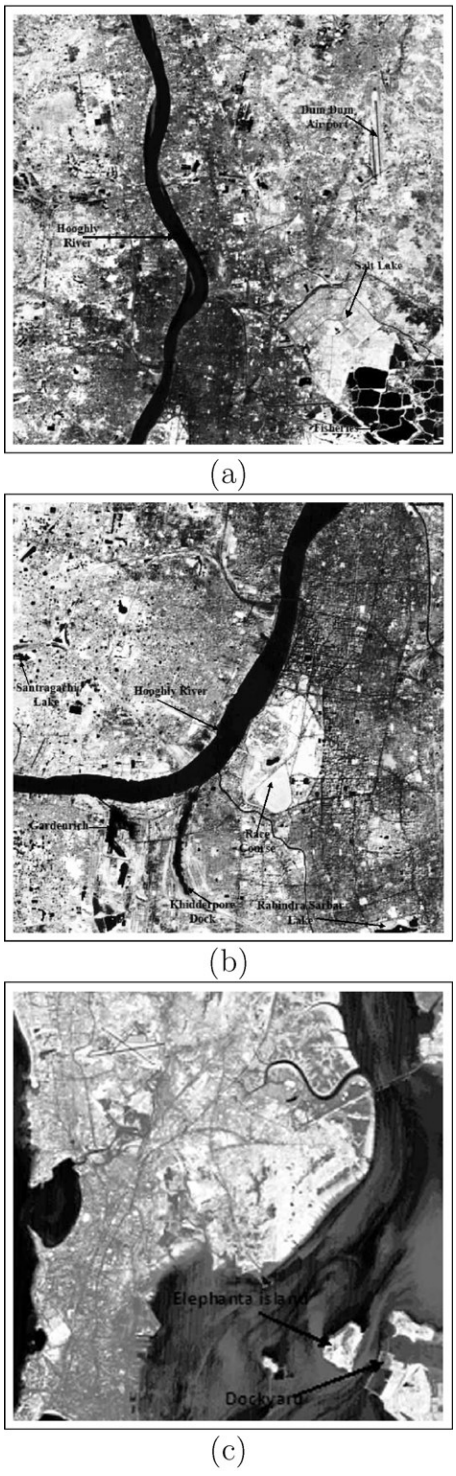

Fig. 1. Histogram equalized images of: (a) IRS-1A Calcutta in NIR (band4), (b) SPOT Calcutta in NIR (band3), and (c) IRS Bombay in NIR (band4).

Table 1
Experimental results for SATIMAGE data obtained by supervised methods (classification) with 10% training data.

Methods used	% OA	Kappa	*S_dbw*	β	Time
APC (δ = 5.2)	**84.49**	**0.81**	0.31	**11.81**	**1.99**
MLP ($N1$ = 6, $N2$ = 5)	77.44	0.73	0.39	8.85	39.97
SVM (σ = 0.35, C = 500)	81.80	0.77	**0.29**	11.57	42.02

Table 2
Experimental results for IRS-1A Calcutta image obtained by supervised methods (classification).

Methods used	Training with labeled data			10-Fold cross-validation		
	S_dbw	β	Time	% OA	Kappa	Time
APC (δ = 5)	**0.33**	**8.73**	**36.82**	**92.14**	**0.90**	**5.20**
MLP ($N1$ = 7, $N2$ = 6)	0.35	7.59	42.82	89.87	0.86	11.59
SVM (σ = 0.27, C = 325)	0.38	7.98	48.62	90.52	0.88	14.88

islands including the well-known Elephanta islands. The dockyard is situated on the south eastern part of Bombay which can be seen as a set of three finger like structure.

The original images of (IRS-1A Calcutta, SPOT Calcutta and IRS Bombay) have poor illumination, very low contrast and are not properly visible. Therefore, we have shown the corresponding (histogram equalized based) enhanced spectral (NIR band) images in Fig. 1 instead of showing their original versions. However, all the investigations are carried out using (feature values of the) original images.

3.2. Description of experiments

The experiments carried out on the four remotely sensed data/images are described below.

3.2.1. Experiments with supervised methods

It is mentioned earlier that the SATIMAGE data is completely labeled. We have randomly chosen 10%, 20%, and 30% training data for the classifier and the rest as test data in three different simulations. For typical illustration of the experimental outcome, we have reported only the results using 10% training data. However, we have observed similar findings considering 20% and 30% training data also.

IRS-1A Calcutta, SPOT Calcutta and IRS Bombay images considered here are labeled partially. 375, 362 and 152 pixels only are labeled for IRS-1A Calcutta, SPOT Calcutta and IRS Bombay images respectively. Labeled pixels are distributed in the six landuse classes mentioned earlier. These labeled pixels act as training patterns and the rest are considered as test patterns for the first kind of experiments (shown as Training with labeled data in Tables 2–4). In the second kind of experiments (shown as 10-fold cross validation in Tables 2–4) the labeled data sets are randomly divided into 10 mutually exclusive and (nearly) equal sized subsets. For each subset, considered as the test set, the classifiers are trained on the union of all other subsets. Then cross validation is run (10 times) for each training and test set pair and average results are reported.

Table 3
Experimental results for SPOT Calcutta image obtained by supervised methods (classification).

Methods used	Training with labeled data			10-Fold cross-validation		
	S_dbw	β	Time	% OA	Kappa	Time
APC (δ = 5.2)	**0.23**	10.04	**30.39**	94.14	0.92	**4.96**
MLP ($N1$ = 5, $N2$ = 4)	0.27	9.50	32.39	87.87	0.84	10.45
SVM (σ = 0.15, C = 300)	0.35	**12.78**	52.07	**94.30**	**0.92**	13.80

Table 4
Experimental results for IRS Bombay image obtained by supervised methods (classification).

Methods used	Training with labeled data			10-Fold cross-validation		
	S_dbw	β	Time	% OA	Kappa	Time
APC (δ = 5.2)	**0.23**	10.74	**28.14**	**92.94**	**0.88**	**5.26**
MLP ($N1$ = 6, $N2$ = 5)	0.29	**11.22**	32.39	89.87	0.83	10.17
SVM (σ = 0.25, C = 328)	0.42	9.73	52.07	88.54	0.80	13.96

(i) *Evaluation measures:* To evaluate the results quantitatively we have used some evaluation measures and they are provided below.

Performance of the algorithms obtained using supervised methods are compared using percentage of overall accuracy (OA) [44], Kappa measures [44] and two internal cluster validity indices namely *S_dbw* [45] and β [2] for the SATIMAGE data. OA and Kappa measures require the class label information of all pixels/patterns. Higher the value of these measures, better is the land-cover prediction. *S_dbw* and β are unsupervised measures, do not require the class label information; rather they measure the fit between the partition imposed by an algorithm and data itself. Lower the value of *S_dbw* and higher the value of the β index, better is the partitioning.

For other three data sets (IRS-1A Calcutta, SPOT and IRS Bombay images) we have used *S_dbw* [45] and β [2] for the experiments requiring training or label data. In the experiments with 10-fold cross validation (done on the training set) OA and Kappa measure have been used.

(ii) *Comparison with other supervised methods:* We have compared the proposed (supervised) classifier APC with two other supervised state of the art techniques namely, Multilayer Perceptron (MLP) using back propagation algorithm [46] and Support Vector Machine (SVM) [47]. For MLP we have taken two hidden layers and experimentally determined the number of hidden neurons in each layer to get the optimum result. The number of neurons in the first and the second hidden layers (denoted as N1 and N2) are put within bracket in Tables 1–4. The initial synaptic weights are randomly assigned in the range [−1,1]. For SVM, *libsvm* package [48] has been used. In SVM we have taken radial basis function. The kernel radius σ and C (controlling parameter for the tradeoff between model complexity and training error) of the SVM are determined experimentally as follows. For complectly labeled SATIMAGE data the parameter values are chosen such that the overall accuracy (OA) is maximum. For other three data sets, parameters are determined for which OA is maximum in 10-fold cross validation, and these parameter values are set for other experiments training with 10%, 20%, 30% training data. Note that for the present investigation one against all multi class decision criteria has been used. The selected parameters C and σ are shown in bracket in Tables 1–4. The value of the only parameter δ (shown in bracket in Tables 1–4) used in the proposed algorithm APC is determined experimentally for which OA is maximum in 10 fold cross validation and that δ value is set for other experiments.

(iii) *Experimental results and analysis:* For SATIMAGE data the performance of the proposed supervised method (APC), MLP and SVM are summarized in Table 1. It is seen from the table that APC performs better with all evaluation indices except *S_dbw*. In terms of *S_dbw*, SVM outperformed the other methods.

The landuse maps of IRS-1A Calcutta image generated by the APC, MLP and SVM are shown in Fig. 2(a)–(c), respectively.

Careful observation of the generated landuse maps reveals that there are misclassified pixels in the Hooghly river when MLP and SVM methods are used. In case of APC there seems to be no or very

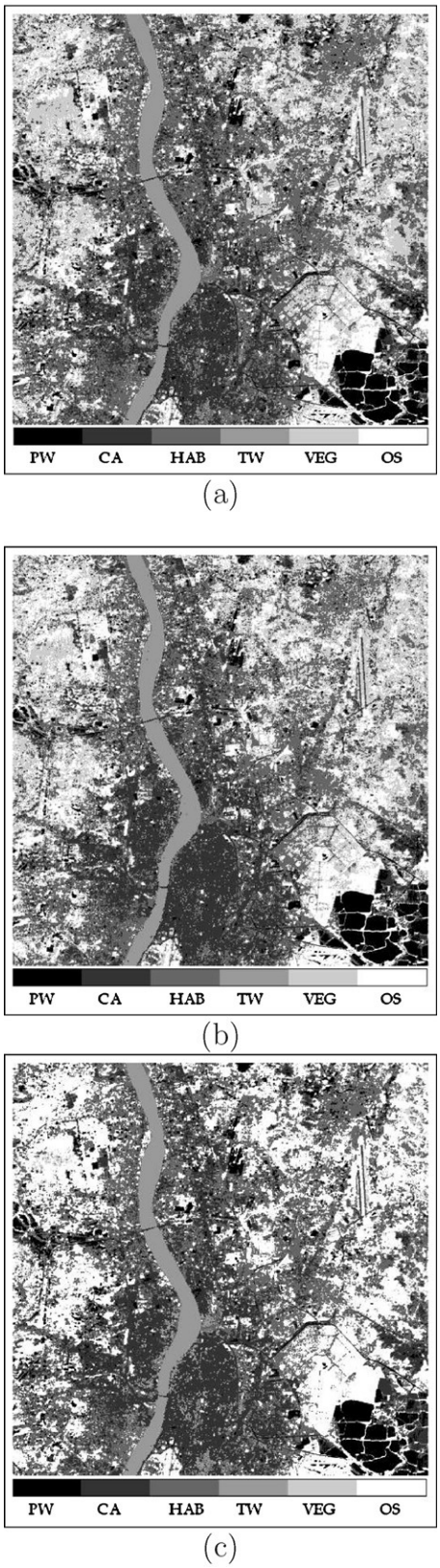

Fig. 2. Landuse maps of IRS-1A Calcutta image generated by supervised methods: (a) APC, (b) MLP, and (c) SVM.

less such misclassification. In general all the supervised algorithms, particularly APC, and MLP appeared to perform well. However, contrary to the actual Calcutta image, open space (OS) detected by the SVM is predominant, and vegetation (VEG) area is infrequent.

For IRS-1A Calcutta image, the performance of the supervised methods in terms of evaluation measures and execution time (in s) are also shown in Table 2. From the quantitative measures also, APC is seen to outperform the other two in both kinds of experiments (Training with labeled data and 10-fold cross validation).

The landuse maps of SPOT Calcutta image generated by supervised methods are shown in Fig. 3(a)–(c). Though all the algorithms seem to perform equally good, minute observations reveal that MLP and SVM fail to detect various concrete area (CA); whereas APC can detect such portions. For example, in the upper left portion of the SPOT Calcutta image the concrete area (CA) determined by the APC seems to be better than those obtained using MLP and SVM. Also the misclassification around the Hooghly river bank is lesser in APC and SVM as compared to MLP. Different regions nearby the Race Course area seem to be better identified by APC.

The performance of the supervised methods for SPOT Calcutta image in terms of various performance measures and execution time (in seconds) are put in Table 3. Performance of the APC method is the best in terms of *S_dbw* measures. However for all other cases SVM marginally outperforms the APC.

The landuse map of IRS Bombay image generated by supervised methods are shown in Fig. 4(a)–(c). From the experimental observations it is seen that there are lots of misclassification in the turbid water (TW1 and TW2), particularly in the bottom portion of the Arabian sea region when MLP (Fig. 4(b)) and SVM (Fig. 4(c)) are used. The problem is even predominant for the SVM method. On the other hand in case of APC, (Fig. 4(a)), the misclassification is very less in such region.

The performance of the supervised methods for IRS Bombay image in terms of various performance indices and execution time (in s) are shown in Table 4. Performance of the APC method is better (for both kinds of experiments) compared to other methods in terms of all the performance measures except β. Execution time is also the least for the APC.

3.2.2. Experiments with unsupervised methods

Here, the same remotely sensed data/images, as used in supervised methods, are considered for investigation purpose.

(i) *Evaluation measures:* As in the supervised methods, here also for all the data sets we have used two internal cluster validity indices namely, *S_dbw* [45] and β [2]. In addition to these internal measures we have used two external cluster validity indices, namely, *Rand* and *Jaccard* [40] for (completely labeled) SATIMAGE data. Note that, range of the *Rand* and *Jaccard* is [0,1]. More is the value of *Rand* or *Jaccard*, better is the partition.

(ii) *Comparison with other unsupervised methods:* We have compared the proposed method with two popular clustering based image segmentation techniques namely *k*-means (KM) method [40] and mean shift (MS) [49] method. In the present investigation MS algorithm presented in [30] is adopted. Here we have used Epanechnikov kernel [50]. Note that the procedure automatically detects the number of segments depending upon the value of the two parameters '*bandwidth*' (radius) and '*stop_threshold*'. For uniform comparison in this article we have set the '*bandwidth*' and '*stop_threshold*' for each data, such that the number of detected clusters (segments) is the same as that of other methods. We have reported the average (quantitative) results of 10 runs and one typical image with the same parameter setting. Selected '*andwidth*' and '*stop_threshold*' are put within bracket with the MS method in Tables 5–8

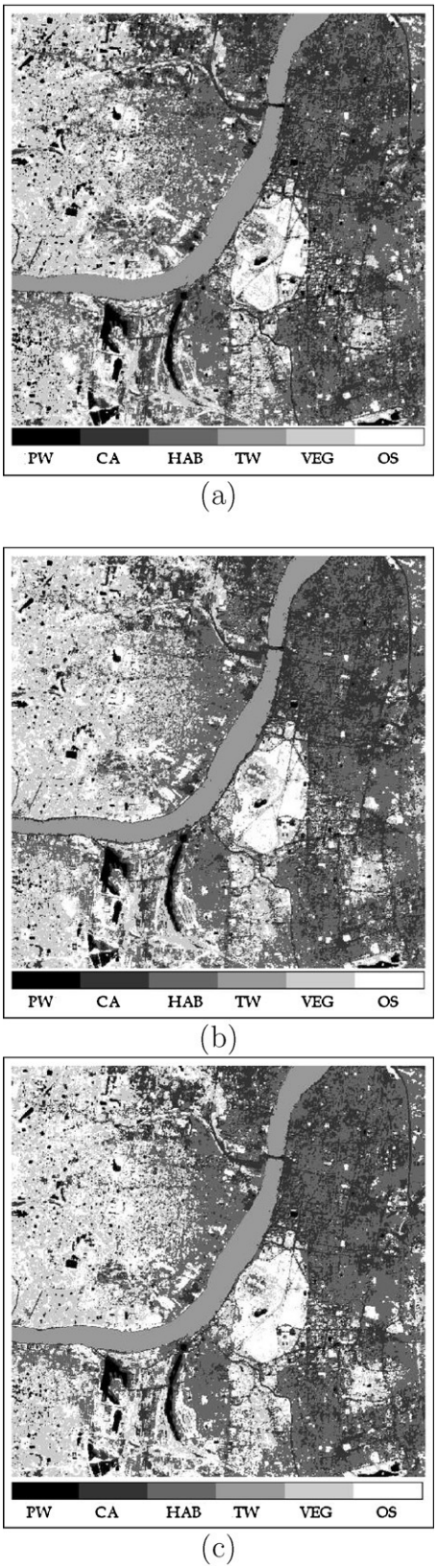

Fig. 3. Landuse maps of SPOT Calcutta image generated by supervised methods: (a) APC, (b) MLP, and (c) SVM.

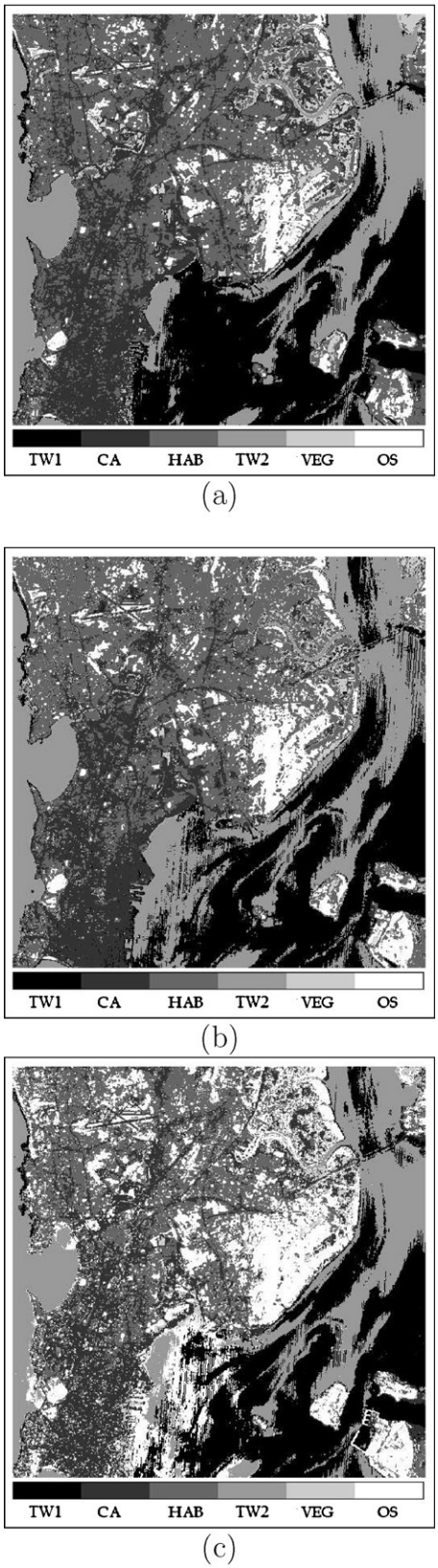

Fig. 4. Landuse maps of IRS Bombay image generated by supervised methods: (a) APC, (b) MLP, and (c) SVM.

Table 5
Experimental results for SATIMAGE data obtained by unsupervised methods (clustering).

Methods used	Rand	Jaccard	*S_dbw*	β	Time
APC ($\delta = 0.37$)	**0.85**	**0.39**	**0.13**	6.43	1.06
KM	0.83	0.38	0.16	7.79	**0.89**
MS (bandwidth = 0.25; *stop_threshold* = 3.17)	0.79	0.28	0.26	**8.27**	4.71

Table 6
Experimental results for IRS-1A Calcutta image obtained by unsupervised methods (clustering).

Methods used	*S_dbw*	β	Time
APC ($\delta = 0.11$)	**0.25**	**8.58**	7.28
KM	0.40	4.80	**5.04**
MS (bandwidth=0.15; *stop_threshold* = 2.52)	0.54	3.92	80.62

Table 7
Experimental results for SPOT Calcutta image obtained by unsupervised methods (clustering).

Methods used	*S_dbw*	β	Time
Proposed APC ($\delta = 0.15$)	**0.15**	**12.85**	**1.01**
KM	0.26	5.00	1.85
MS (bandwidth = 0.15; *stop_threshold* = 2.74)	0.22	8.52	10.26

(iii) *Selection of parameters*: It is evident from Algorithm 2 that the proposed method has three parameters namely η, *threshold_density* and δ. Here η is the step size. The smaller the step size, more will be the time taken to explore the feature space. The performance of the algorithm in terms of validity measures is found to remain almost constant for a wide range [0.1–1.9] of η. We have reported the results of the experiments with step size $\eta = 1$, as the performance is found to be constant over a wide range around it. If the ratio of the pheromone density of a data point and an already formed cluster center (within distance 2δ) is higher than the *threshold_density* then the data point is assigned to the said cluster. This assumes that two closer points having nearly similar pheromone densities should belong to the same cluster. High *threshold_density* value indicates that pheromone densities of two points (within distance 2δ) should be very similar (assign to the same cluster) whereas, less *threshold_density* value indicates that the two closer points may reside in the same cluster even if their pheromone densities are not very similar. If the *threshold_density* value is high, it is likely to form large number of clusters in the initial phase (before merging the clusters); and if it is less, the number of clusters thus formed (in the initial phase) may be less. We have executed the algorithm considering different values of *threshold_density* over the range [0.5–0.9] and on an average the value of 0.9 has been found to be a suitable one. The algorithm is executed for different δ (spread of the Gaussian) values in the range (0–0.9]; and experimentally determined δ (shown within bracket in Tables 5–8 with APC method) is used to produce the optimum results in terms of validity measures.

Table 8
Experimental results for IRS Bombay image obtained by unsupervised methods (clustering).

Methods used	*S_dbw*	β	Time
Proposed APC ($\delta = 0.14$)	**0.1772**	**11.3306**	**1.25**
KM	0.3094	4.8271	1.65
MS (bandwidth = 0.23; *stop_threshold* = 2.5)	0.2850	5.4356	11.37

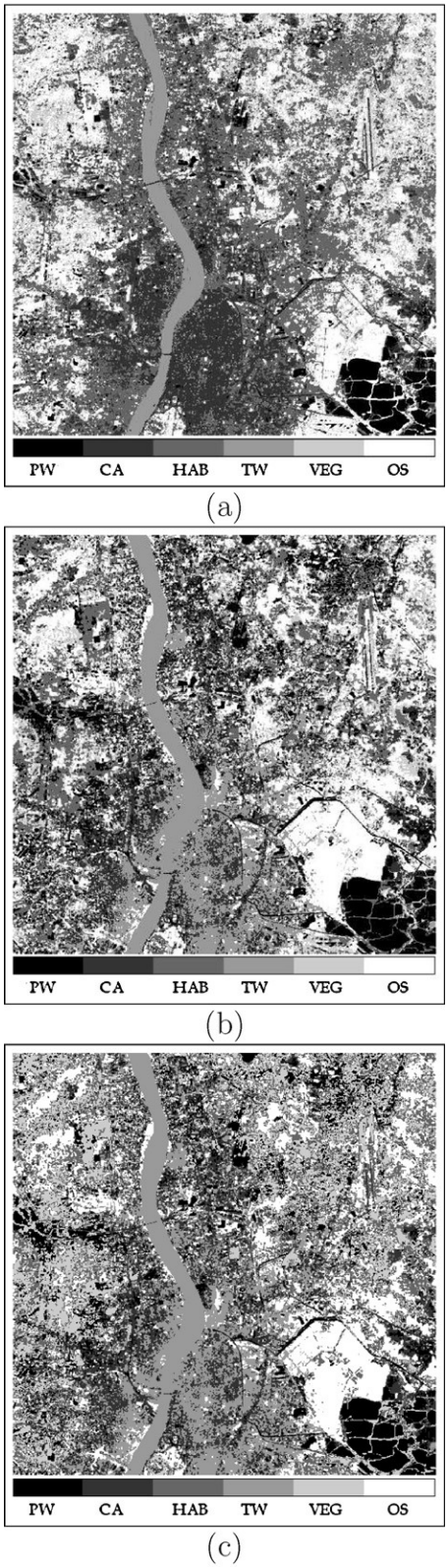

Fig. 5. Landuse maps of IRS-1A Calcutta image generated by unsupervised methods: (a) APC, (b) KM, and (c) MS.

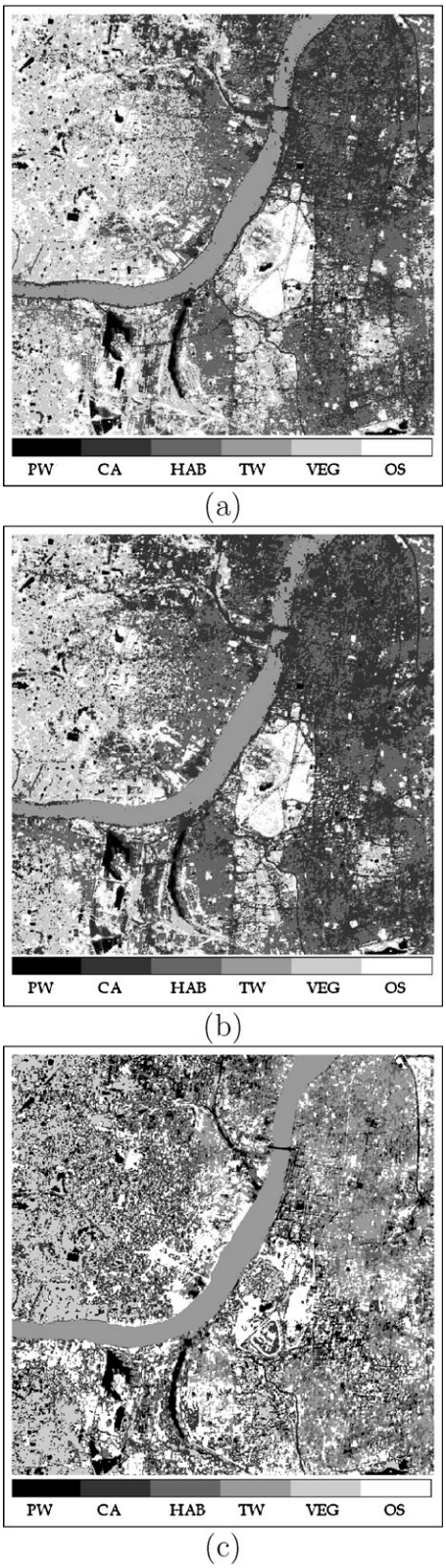

Fig. 6. Landuse maps of SPOT Calcutta image generated by unsupervised methods: (a) APC, (b) KM, and (c) MS.

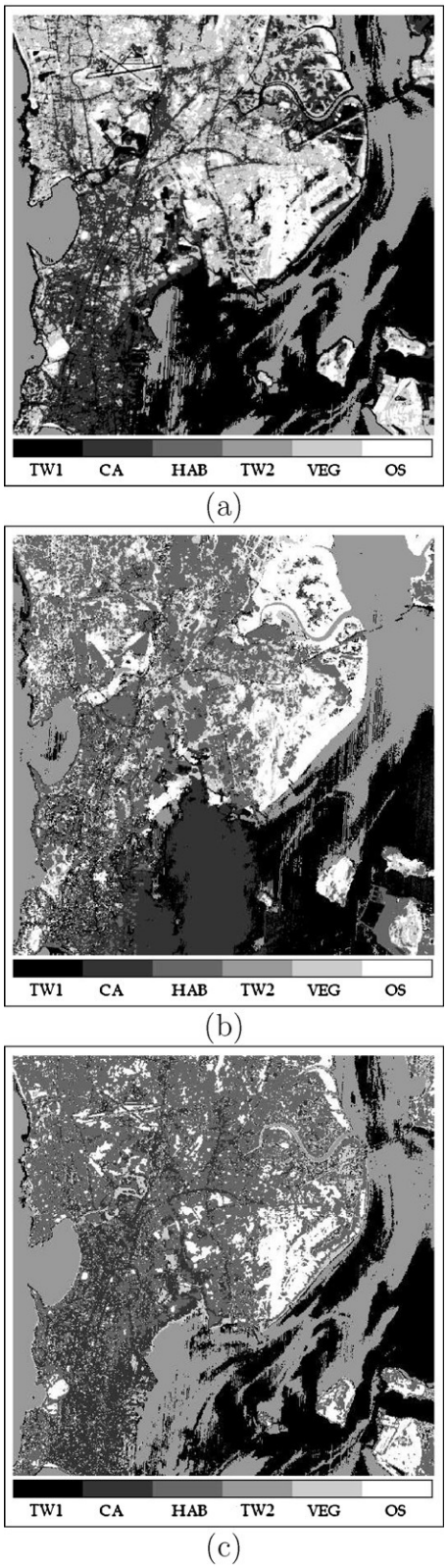

Fig. 7. Landuse maps of IRS Bombay image generated by unsupervised methods: (a) APC, (b) KM, and (c) MS.

(iv) *Experimental results and analysis:* Experimental results on SATIMAGE data obtained by the proposed unsupervised method (APC) compared to KM and MS methods are summarized in Table 5. Among all the evaluation measures, except β index, performance of APC is found to be better compared to other algorithms. In terms of β, MS method outperformed others.

For visual illustration, the landuse maps of IRS-1A Calcutta image generated by APC, KM and MS methods are shown in Fig. 5(a)–(c), respectively. From the generated landuse maps it is seen that different regions are better identified by the proposed APC method. For example, runway of Dum Dum Airport (CA), two bridges in the Hooghly river (CA), and Hooghly river (TW) itself is clearly identified by the APC method as compared to those using KM and MS. There are lots of misclassification in the landuse map; particularly concrete area (CA) near the Hooghly river are wrongly identified as turbid water (TW) by KM and MS methods; whereas APC detected these regions properly.

The performance of the proposed unsupervised method (for IRS-1A Calcutta image) in terms of evaluation measures and execution time (in s) are summarized in Table 6. Superiority of the proposed APC algorithm is also seen from the table. However, computational time requirement of the KM method is the least among the three.

The landuse maps of SPOT Calcutta image generated by APC, KM and MS methods are shown in Fig. 6(a)–(c), respectively. From the landuse maps, it is seen that like earlier images lots of misclassification occurred in the Hooghly river by the KM method; whereas it is very less by APC and MS methods. However, MS method failed to detect the Race Course region properly, but APC and KM could detect it. Also the concrete area (CA) near the Hooghly river seems to be infrequently detected by MS.

The performance of the proposed unsupervised method APC as compared to KM and MS methods (summarized in Table 7) in terms of evaluation measures and execution time (in s) are also found to be better.

The landuse maps of IRS Bombay image generated by APC, KM and MS methods are shown in Fig. 7(a)–(c), correspondingly. From the landuse maps, it is seen that there are lots of misclassification in both kinds of turbid water (TW1 and TW2); particularly at the bottom portion of the sea when KM and MS methods are used. The misclassification is less in case of APC method.

The performance of the proposed unsupervised method APC for IRS Bombay image as compared to KM and MS methods (summarized in Table 8) in terms of evaluation measures and execution time (in s) are also found to be better.

4. Conclusions

Motivated from group forming behavior of real ants, in order to automatically generate landuse maps from multispectral remotely sensed images, we have proposed two novel ant colony based algorithms. One is a supervised method, treating land use map generation as pattern classification problem assuming few labeled pixels are available from different landuse regions. The other one is an unsupervised technique considering landuse map generation as clustering based image segmentation problem. The proposed supervised APC algorithm is compared with two other popular classification algorithms, MLP and SVM. One multispectral remotely sensed data (SATIMAGE) and three remotely sensed images (IRS-1A, SPOT Calcutta and IRS Bombay) are used for the investigation purpose. Results are quantified with overall accuracy (OA), Kappa measure for SATIMAGE, and & *S_dbw* [45] & β [2] indices for all the data sets. From experimental results, it is observed that supervised APC method outperforms the other two algorithms, MLP and SVM. The proposed unsupervised APC method is compared with KM and MS using Rand, Jaccard (for comletely labeled SATIMAGE data), and two external evaluation indices *S_dbw* and β (for all the data sets). Superiority of the proposed unsupervised APC algorithm for landuse map generation/prediction is justified from experimental outcome. In summary, from the experimental observations it is seen that both supervised as well as unsupervised APC algorithms are useful for automatic landuse map generation from mutispectral remotely sensed images. In future we plan to test the proposed algorithms on other kinds of (hyperspectral) remotely sensed data as well as compare the results with some other existing methodologies.

Acknowledgements

Support of the Department of Science and Technology, Govt. of India to the Center for Soft Computing Research is thankfully acknowledged by Mr. Anindya Halder, Research Scholar of the Center, Indian Statistical Institute, Kolkata.

Appendix A.

A.1. Formulas of evaluation indices

- Overall accuracy (OA) and Kappa coefficient (Kappa): With respect to statistical measures for accuracy evaluation, 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 [44] *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 supervised 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). Two commonly adopted indices are the overall accuracy (OA) [44] and the kappa coefficient of accuracy (Kappa) [44], which are defined as

$$\mathrm{OA} = \frac{\sum_{i=1}^{C} n_{ii}}{n} \tag{A.1}$$

$$\mathrm{Kappa} = \frac{n\sum_{i=1}^{C} n_{ii} - \sum_{i=1}^{C} n_{i+}n_{+i}}{n^2 - \sum_{i=1}^{C} n_{i+}n_{+i}} \tag{A.2}$$

where $n = \sum_{i=1}^{C}\sum_{j=1}^{C} n_{ij}$ 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.*

- S_dbw: *S_Dbw* index with *C* number of clusters is based on the cluster compactness in terms of intra-cluster variance and inter-cluster density [45]. It is defined as

$$S\_Dbw(C) = Scat(C) + Den(C), \tag{A.3}$$

where $Scat(C)$ represents the intra-cluster variance and is defined as

$$Scat(C) = \frac{1}{C}\sum_{i=1}^{C} ||\sigma(\mathbf{Z_i})||/||\sigma(X)||; \tag{A.4}$$

the term $\sigma(\boldsymbol{X})$ is the variance of the data set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ and $\sigma(\mathbf{Z_i})$ is the variance of cluster C_i. Inter-cluster density, $Den(C)$, is defined as

$$Den(C) = \frac{1}{C(C-1)}\sum_{i=1}^{C}\left(\sum_{i=1, i\neq j}^{C}\frac{den(\mathbf{u}_{ij})}{\max\{den(\mathbf{Z_i}), den(\mathbf{Z_j})\}}\right) \tag{A.5}$$

where $\mathbf{Z_i}$ and $\mathbf{Z_j}$ are centers of clusters C_i and C_j, respectively and $\mathbf{u_{ij}}$ is the mid point of the line segment joining $\mathbf{Z_i}$ and $\mathbf{Z_j}$. The term $den(\mathbf{u})$ is defined as

$$den(\mathbf{u}) = \sum_{x\in C_i \cup C_j} f(\mathbf{x}, \mathbf{u}). \tag{A.6}$$

The function $f(\mathbf{x}, \mathbf{u})$ is defined as

$$f(\mathbf{x}, \mathbf{u}) = \begin{cases} 0, & \text{if } d(\mathbf{x}, \mathbf{u}) > stdev; \\ 1, & \text{otherwise;} \end{cases} \tag{A.7}$$

where *stdev* is the average standard deviation of *C* clusters and is defined as

$$stdev = \frac{1}{C}\sqrt{\sum_{i=1}^{C}||\sigma(\mathbf{Z_i})||} \tag{A.8}$$

and $d(\mathbf{x}, \mathbf{u})$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{u}$.

Lower the value of S_Dbw, better is the clustering.

- Beta index (β): It computes the ratio of total variation and within class variation [2], and is defined as

$$\beta = \frac{\sum_{i=1}^{C}\sum_{j=1}^{n_i}(\boldsymbol{X}_{ij} - \bar{\boldsymbol{X}})^2}{\sum_{i=1}^{C}\sum_{j=1}^{n_i}(\boldsymbol{X}_{ij} - \bar{\boldsymbol{X}}_i)^2} \tag{A.9}$$

where $\bar{\boldsymbol{X}}$ is the mean of all the data points and $\bar{\boldsymbol{X}}_i$ is the mean of the data points that belong to cluster C_i, $\boldsymbol{X}_{ij}$ is the jth data point of ith cluster and n_i is the number of data points in cluster C_i. Since the numerator is a constant for a given data set, the value of β is dependent only on the denominator. The denominator decreases with homogeneity in the formed clusters. Therefore, for a given data set, higher the value of β, better is the clustering.

- Rand coefficient (R): It determines the degree of similarity between the known correct solution reflecting its class label (group) and the solution obtained by a clustering algorithm [40]. It is defined as

$$R = \frac{SS + DD}{SS + SD + DS + DD}. \tag{A.10}$$

SS, *SD*, *DS*, *DD* represent the number of possible pairs of data points *i* and *j* where,

SS: both the data points belong to the same cluster and same group.
SD: both the data points belong to the same cluster but different groups.
DS: both the data points belong to different clusters but same group.
DD: both the data points belong to different clusters and different groups.

Value of *R* is in the range [0,1] and *higher the value of R, better is the clustering.*

- Jaccard coefficient (J): It is the same as rand coefficient except that it excludes *DD* and is defined as

$$J = \frac{SS}{SS + SD + DS}. \tag{A.11}$$

Value of *J* lies in the interval [0,1] and *higher the value of J, better is the clustering.*

A.2. Typical results in terms of confusion matrix for SATIMAGE using supervised methods (classification)

Here we provide some typical results in terms of confusion matrix for SATIMAGE using supervised methods (proposed APC, MLP, and SVM). Please note that the reported results are obtained using the same set of training (with 10%) and test set pair for every method (APC, MLP, and SVM). Obtained results are reported below.

$$\text{by APC} = \begin{bmatrix} 1318 & 5 & 4 & 3 & 50 & 0 \\ 0 & 565 & 0 & 0 & 9 & 0 \\ 30 & 0 & 1091 & 94 & 3 & 17 \\ 8 & 19 & 124 & 425 & 22 & 309 \\ 24 & 42 & 0 & 2 & 503 & 32 \\ 0 & 2 & 4 & 40 & 50 & 1000 \end{bmatrix}$$

% of OA and Kappa coefficient for the particular confusion matrix given above are 84.59 and 0.81, respectively.

$$\text{by MLP} = \begin{bmatrix} 963 & 6 & 1 & 1 & 47 & 0 \\ 0 & 554 & 0 & 0 & 6 & 0 \\ 123 & 0 & 1099 & 96 & 13 & 19 \\ 105 & 34 & 119 & 407 & 35 & 284 \\ 128 & 32 & 0 & 1 & 432 & 18 \\ 61 & 7 & 4 & 59 & 104 & 1037 \end{bmatrix}$$

% of OA and Kappa coefficient for the particular confusion matrix given above are 77.51 and 0.72, respectively.

$$\text{by SVM} = \begin{bmatrix} 1202 & 5 & 1 & 2 & 61 & 0 \\ 0 & 559 & 0 & 0 & 7 & 0 \\ 56 & 0 & 1111 & 96 & 6 & 21 \\ 28 & 26 & 107 & 413 & 29 & 295 \\ 92 & 38 & 0 & 1 & 462 & 18 \\ 2 & 5 & 4 & 52 & 72 & 1024 \end{bmatrix}$$

% of OA and Kappa coefficient for the particular confusion matrix given above are 82.33 and 0.78, respectively.

References

[1] R. Laprade, Split-and-merge segmentation of aerial photographs, Computer Vision Graphics and Image Processing 48 (1) (1988) 77–86.
[2] S. Pal, A. Ghosh, B. Uma Shankar, Segmentation of remotely sensed images with fuzzy thresholding and quantitative evaluation, International Journal on Remote Sensing 21 (11) (2000) 2269–2300.
[3] R. Cannon, R. Dave, J. Bezdek, M. Trivedi, Segmentation of a thematic mapper image using fuzzy c-means clustering algorithm, IEEE Transactions on Geoscience and Remote Sensing 24 (1) (1986) 400–408.

[4] A. Baraldi, F. Parmiggiani, A neural network for unsupervised categorization of multivalued input pattern: an application to satellite image clustering, IEEE Transactions on Geoscience and Remote Sensing 33 (2) (1995) 305–316.
[5] Z. Zhou, S. Wei, X. Zhang, X. Zhao, Remote sensing image segmentation based on self organizing map at multiple scale, in: Proceedings of SPIE Geoinformatics: Remotely Sensed Data and Information, USA, 2007, pp. 122–126.
[6] Y. Wong, E. Posner, A new clustering algorithm applicable to polarimetric and sar images, IEEE Transactions on Geoscience and Remote Sensing 31 (3) (1993) 634–644.
[7] A. Baraldi, F. Parmiggiani, A multiscale random field model for bayesian image segmentation, IEEE Transactions on Image Processing 3 (2) (1994) 162–177.
[8] S. Pal, P. Mitra, Multispectral image segmentation using the rough set initialized EM algorithm, IEEE Transactions on Geoscience and Remote Sensing 40 (11) (2002) 2495–2501.
[9] U. Maulik, S. Bandyopadhyay, Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification, IEEE Transactions on Geoscience and Remote Sensing 41 (5) (2003) 1075–1081.
[10] S. Bedawi, M. Kamel, Segmentation of very high resolution remote sensing imagery of urban areas using particle swarm optimization algorithm, in: Image Analysis and Recognition, Vol. 6111 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2010, pp. 81–88.
[11] U. Maulik, I. Saha, Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery, Pattern Recognition 42 (9) (2009) 2135–2149.
[12] C. Chen, Fuzzy training data for fuzzy supervised classification of remotely sensed images, in: Proceedings of 20th Asian Conference on Remote Sensing (ACRS 1999), 1999, pp. 460–465.
[13] A. Ghosh, S. Meher, B.U. Shankar, A novel fuzzy classifier based on product aggregation operator, Pattern Recognition 41 (6) (2008) 961–971.
[14] F. Maselli, A. Rodolfi, C. Copnese, Fuzzy classification of spatially degraded thematic mapper data for the estimation of sub-pixel components, International Journal of Remote Sensing 17 (3) (1996) 537–551.
[15] F. Melgani, B.A. Hashemy, S. Taha, An explicit fuzzy supervised classification method for multispectral remote sensing images, IEEE Transaction on Geoscience and Remote Sensing 38 (1) (2000) 287–295.
[16] F. Wang, Linear spectral mixture models and support vector machines for remote sensing, IEEE Transactions on Geosciences and Remote Sensing 28 (2) (1990) 194–201.
[17] N. Mulder, L. Spreeuwers, Neural networks applied to the classification of remotely sensed data, in: Proceedings of IGARSS, Espoo, Finland, 1991, pp. 2211–2213.
[18] Z.K. Liu, J.Y. Xiao, Classification of remotely-sensed image data using artificial neural networks, International Journal of Remote Sensing 12 (11) (1991) 2433–2438.
[19] M. Brown, H. Lewis, S. Gunn, Linear spectral mixture models and support vector machines for remote sensing, IEEE Transactions on Geosciences and Remote Sensing 38 (5) (2000) 2346–2360.
[20] C. Huang, L. Davis, J. Townshend, An assessment of support vector machines for land cover classificafition, International Journal of Remote Sensing 23 (2002) 725–749.
[21] D. Stathakis, A. Vasilakos, Comparison of computational intelligence based classification techniques for remotely sensed optical image classification, IEEE Transactions on Geoscience and Remote Sensing 44 (8) (2008) 2305–2318.
[22] S. Lee, J. Han, K. Chi, J. Suh, H. Lee, M. Miyazaki, K. Akizuki, A neuro-fuzzy classifier for land cover classification, in: Proceedings of IEEE International Conference on Fuzzy System, 1999, pp. 1063–1068.
[23] S. Meher, B.U. Shankar, A. Ghosh, Wavelet-feature-based classifiers for multispectral remote-sensing images, IEEE Transactions on Geoscience and Remote Sensing 45 (6) (2007) 1881–1886.
[24] G. Storvik, R. Fjørtoft, A.H. Solberg, A bayesian approach to classification of multiresolution remote sensing data, IEEE Transactions on Geosciences and Remote Sensing 43 (3) (2005) 539–547.
[25] S. Tsutsui, Ant colony optimization for continuous domains with aggregation pheromones metaphor, in: Proceedings of the 5th International Conference on Recent Advances in Soft Computing, United Kingdom, December, 2004, pp. 207–212.
[26] S. Tsutsui, A. Ghosh, An extension of ant colony optimization for function optimization, in: Proceedings of the 5th Asia Pacific Conference on Simulated Evolution and Learning, Pusan, Korea, 2004.
[27] A. Englebrecht, Computational Intelligence: An Introduction, Wiley, New York, USA, 2007.
[28] J. Kennedy, R. Eberhart, Swarm Intelligence, Morgan Kaufman Publishers, San Francisco, CA, USA, 2001.
[29] M. Dorigo, T. Stützle, Ant Colony Optimization, Prentice Hall of India Private Limited, New Delhi, India, 2005.
[30] S. Ghosh, M. Kothari, A. Halder, A. Ghosh, Use of aggregation pheromone density for image segmentation, Pattern Recognition Letters 30 (10) (2009) 939–949.
[31] K. Socha, M. Dorigo, Ant colony optimization for continuous domains, European Journal of Operational Research 185 (3) (2008) 1155–1173.
[32] J. Handl, B. Meyer, Ant-based and swarm-based clustering, Swarm Intelligence 1 (2007) 95–113.
[33] R. Parpinelli, H. Lopes, A. Freitas, Data mining with an ant colony optimization algorithm, IEEE Transactions on Evolutionary Computing 6 (4) (2002) 321–332.
[34] B. Liu, H. Abbass, B. McKay, Density-based heuristic for rule discovery with ant-miner, in: Proceedings of 6th Australasia-Japan Joint Workshop on Intelligence Evololution System, 2002, pp. 180–184.
[35] B. Liu, H. Abbass, B. McKay, Classification rule discovery with ant colony optimization, in: Proceedings of IEEE/WIC International Conference Intellelligent Agent Technology, IEEE Computer Society, 2003.
[36] D. Martens, M. Backer, R. Haesen, J. Vanthienen, M. Snoeck, B. Baesens, Data mining with an ant colony optimization algorithm, IEEE Transactions on Evolutionary Computing 11 (5) (2007) 651–665.
[37] A. Ghosh, A. Halder, M. Kothari, S. Ghosh, Aggregation pheromone density based data clustering, Information Sciences 178 (13) (2008) 2816–2831.
[38] A. Halder, A. Ghosh, S. Ghosh, Aggregation pheromone density based pattern classification, Fundamenta Informaticae 92 (4) (2009) 345–362.
[39] R. Gonzalez, R. Woods, Digital Image Processing, 3rd ed., Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.
[40] S. Theodoridis, K. Koutroumbas, Pattern Recognition, 3rd ed., Academic Press, New York, 2006.
[41] D. Newman, S. Hettich, C. Blake, C. Merz, UCI Repository of Machine Learning Databases, University of California, Irvine, Department of Information and Computer Sciences, 1998, http://www.ics.uci.edu/~mlearn/MLRepository.html.
[42] I. Data Users Hand Book, Document No. IRS/NRSA/NDC/HB-02/89, NRSA, Hyderabad, India, 1989.
[43] J. Richards, X. Jia, Remote Sensing Digital Image Analysis: An Introduction, Springer-Verlag, Berlin, Heidelberg, 2006.
[44] R. Congalton, K. Green, Assessing the Accuracy of Remotely Sensed Data, Lewis Publications, Boca Raton, FL, 1999.
[45] M. Halkidi, M. Vazirgiannis, Clustering validity assessment: finding the optimal partitioning of a data set, in: Proceedings of ICDM, CA, USA, 2001, pp. 187–194.
[46] D. Rumelhart, G. Hinton, L. McLelland, Parallel Distributed Processing Exploration in the Microstructure of Cognition, vol. 1, MIT, Cambridge, 1986.
[47] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.
[48] C. Chang, C. Lin, LIBSVM: A Library for Support Vector Machines, 2001, http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[49] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (8) (1995) 233–260.
[50] B Silverman, Density Estimation for Statistics and Data Analysis, Chapman and Hall, New York, 1986.