USING LEARNING CELLULAR AUTOMATA FOR POST CLASSIFICATION SATELLITE IMAGERY

B. Mojaradi [a] , C.Lucas [b], M.Varshosaz [a]

[a] Faculty of Geodesy and Geomatics Eng., KN Toosi University of Technology, Vali_Asr Street, Mirdamad Cross, Tehran, Iran,
Mojaradi@alborz.kntu.ac.ir , varshosazm@kntu.ac.ir

[b] Dept. of Electrical and Computer science Engineering, University of Tehran ,Amirabad Cross,Tehran, Iran,
Lucas@imp.ir

KEY WORDS: Cellular Automata, Expert System, Entropy, Hyper Spectral, Information Extraction, Post Classification, Reliability, Uncertainty

ABSTRACT:

When classifying an image, there might be several pixels having near among probability, spectral angle or mahalanobis distance which are normally regarded as unclassified or misclassified. These pixels so called chaos pixels exist because of radiometric overlap between classes, accuracy of parameters estimated, etc. which lead to some uncertainty in assigning a label to the pixels. To resolve such uncertainty, some post classification algorithms like Majority, Transition matrix and Probabilistic Label Relaxation (PLR) are traditionally used. Unfortunately, these techniques are inflexible so a desired accuracy can not be achieved. Therefore, techniques are needed capable of improving themselves automatically.

Learning Automata have been used to model biological learning systems in computer science to find an optimal action offered by an environment. In this research, we have used pixels as the cellular automata and the thematic map as the environment to design a self-improving post classification technique. Each pixel interacts with the thematic map in a series of repetitive feedback cycles. In each cycle, the pixel chooses a class (an action), which triggers a response from the thematic map (the environment); the response can either be a reward or a penalty. The current and past actions performed by the pixel and its neighbours define what the next action should be. In fact, by learning, the automata (pixels) change the class probability and choose the optimal class adapting itself to the environment. For learning, tow criteria for local and global optimization, the entropy of each pixel and Producer's Accuracy of classes have been used.

Tests were carried out using a subset of AVIRIS imagery. The results showed an improvement in the accuracy of test samples. In addition, the approach was compared with PLR, the results of which suggested high stability of the algorithm and justified its advantages over the current post classification techniques.

1. INTRUDUCTION

There are many techniques for hyper spectral image analysis in order to extract information. Classification is one of these which are used frequently in remote sensing. Maximum Likelihood (ML), Spectral Angle Mapper, Linear Spectral Unmixing (LSU), fuzzy and binary encoding are conventional algorithms for multi spectral and hyperspectral image classification. These algorithms have their own accuracy which should be investigated. In order to produce thematic map it is necessary to performed post processing algorithms on the result of classification. There are many parameters that tend to make uncertainty in remote sensed data. These parameters arise from sensor system, complexity of the area that is covered by image, geometric and atmospheric distortions (Franciscus Johannes, 2000).Furthermore training data, size of sample data for estimating of statistic such as mean and standard deviation, statistical model for computing statistic parameters, radiometric overlap and also classification algorithms effect on label classified pixels.

These parameters cause to decrease accuracy of classification which should be improved in post processing stage. There are many conventional techniques such as, majority filter, Tomas's filter, transition matrix, Probability Label Relaxation (PLR) model which are used to improve accuracy of classification results. Most of these algorithms are limited and inflexible or need some background for using. Their accuracy depends on the knowledge; therefore, techniques are needed capable of improving themselves automatically and compensate the lack of complete knowledge. In this paper, at first we express different techniques of post processing, then we introduce components of learning automata and their structures. We followed by discussing about cellular learning automata and the way of learning cellular automata. As cellular learning automata is goal oriented and try to change its action with respect many parameters such as its experiments, action of its neighbours and the response of environment, it could be used for different purpose. In this research cellular learning automata is used for post processing of result of classification which performed by maximum likelihood and linear spectral unmixing algorithms. At the end the result of algorithm is compared with probability label relaxation.

2. CONVENTIONAL POST CLASSIFICATION ALGORITHM

2.1 Majority Filter

Majority filter is a logical filter which relabel centre pixel, if it is not a member of majority class; in other word the label of majority class is given to center pixel.This algorithm perform in the following expression .

$$\text{If } (n_i > n_j \;\&\&\; n_i > n_t \text{ for all } i = j) \text{ then } x \in \omega_j \quad (1)$$

where x = centre pixel,
ni & nj = the number of adjacent pixels belong to class i and j
nt = threshold

Usually a moving 3*3 window is used and threshold 5 applied for this purpose, the effect of this algorithm is to smooth the classified image by weeding out isolated pixels that were initially given labels that were dissimilar labels assigned to the surrounding pixels. (Mather, 1999)

2.2 Thomas Filter

Thomas (Thomas,1980)introduce a method based on proximity function which is described as follows:

$$f_j = \sum_i \frac{q_i q_5}{d_{i5}^2} \quad \text{if } x_i \in \omega_j \text{ then } q_i=2 \text{ else } q_i=0 \quad (2)$$

$$\text{if } x_5 \in \omega_j \text{ then } q_5=2 \text{ else } q_5=1 \quad (i=2,4,6,8) \quad (j=1,2,3,\ldots k)$$

where qi = weight of ith pixel
q5= center pixel
ωj = jth class
di52=distance between ith and center pixel.

As shown in figure1 this algorithm uses direct adjacent for its calculation. Like the majority filter, Tomas filter remove isolated pixels and relable considering direct neighbours. It might also reallocate a previously unclassified pixel that had been placed in the reject class by the classification algorithm. (mather, 1999)

	8	
6	5	2
	4	

Figure1: direct neighbor pixels

2.3 Transition Matrices

Transition Probability Matrices is an algorithm which uses temporal information and expresses the expectation that cover types will change during a particular period of time (Franciscus Johannes, 2000) Knowledge about the dependency of crops to seasons and their mutual sequences is valuable for defining the conditional probability as P(class ω_j at time t_2/ class ω_i at time t_1) . The statistical concept of marcov chains is closely related to this subject, as it describes the dependencies between a state at t_2 and the previous states ($t_1, t_0, t_{-1}, \ldots$) this algorithm concern to agriculture area.

2.4 Probability Label Relaxation

Probabilistic label relaxation is a postclassification context algorithm which begins by assuming that a classification based on spectral data alone has been carried out. This algorithm was introduced by hurries in 1985.This method is based on the key concepts of probability, compatibility coefficient, neighborhood function, and updating rule (Richards 1993).

2.4.1 Probabilities: Probabilities for each pixel describe the chance that the pixel belongs to each of the possible classes. In the initial stage, a set of probabilities could be computed from pixel based and subpixel classifiers. These algorithms performed by spectral data alone, maximum likelihood and linear spectral Unmixing are among these algorithms. In this research for LSU classification the fraction of each endemember is consider as initial stage.

$$\sum_{j=1}^{k} p_i(\omega_j) = 1 \qquad 0 \le p_i(\omega_j) \le 1 \quad (3)$$

where $p_i(\omega_j)$ = probabilities of pixel *i* belongs to class j

2.4.2 Compatibility Coefficient: A compatibility coefficient describes the context of the neighbour and how compatible it is to have pixel m classified as ω_i and neighbouring pixel n classified as ω_j.it is defind as

$$r_{ij}(w_k, w_l) = \log \frac{N_{ij}(w_k, w_l)}{\sum_{k=1}^{K} N_i(w_k, w_l) \sum_{l=1}^{K} N_j(w_k, w_l)} \quad (4)$$

where $N_{ij}(w_k, w_l)$ = the frequency of occurrence of class ω_k ω_l was neighbours at pixel i and j;

2.4.3 Neighbourhood Function: A neighborhood function is a function of the label probabilities of the neighboring pixels, compatibility coefficients, and neighborhood weights. It is defined as:

$$q_i^{(t)}(w_k) = \sum_{j=1}^{N_b} d_{ij} \sum_{l=1}^{Nc} r_{ij}(w_k, w_l) p_j^{(t)}(w_l) \quad (5)$$

where N_b= the number of neighbors considered for pixel i
d_{ij}=the weight factor of neighbors
N_c= number of classes
T=number of iteration

2.4.4 Updating Rule: A neighborhood function allows the neighborhoods to influence the possible classification of the center pixel and update the probabilities, by multiplying the label probabilities by the neighborhood function. These new values are divided by their sum in order to the new set of label probabilities sums to be one.

$$p_i^{(t+1)}(w_k) = \frac{p_i^{(t)}(w_k)\left[1+q_i^{(t)}(w_k)\right]}{\sum_{l=1}^{k} p_i^{(t)}(w_k)\left[1+q_i^{(t)}(w_k)\right]} \tag{6}$$

where $P_i^{(t)}(\omega_k)$= the probability of pixel i belongs to class ω_k of the t-th iteration
$q_i^{(t)}(\omega_k)$= neighborhood function of pixel i belongs to class ω of the t-th iteration;

Therefore relaxation is an iterative technique which probabilities of neighbouring pixels are used iteratively to update the probabilities for a given pixel based on a relation between the pixel labels specified by compatibility coefficient. This approach is computationally intensive and robust to image noise (zur Erlangung, 1999).

3. LEARNING ATOMATA AND ENVIRONMENT

The goal of many intelligent problem-solving systems is to be able to make decisions without a complete knowledge of the consequences of the various choices available. In order for a system to perform well under conditions of uncertainty, it has to be able to acquire some knowledge about the consequences of different choices. This acquisition of the relevant knowledge can be expressed as a learning problem.

Learning Automata is a model of computer learning which has been used to model biological learning systems and to find the optimal action which is offered by a random environment. Learning automata has found applications in system that process incomplete knowledge about the environment in which they operate. These applications includes parameter optimization, statistical decision making, telephone routing, pattern recognition, game playing, natural language processing, modelling biological learning systems, and object partitioning(Oommen1, 2003). The learning loop involves two entities: the environment and learning automata; the actual process of learning is represented as a set of interactions between the environment and the learning automata the learning automata is limited to choosing only one of actions at any given time from a set of actions$\{\alpha_1, ..., \alpha_r\}$ which are offered by the environment. Once the learning automata decide on an action a_i, this action will serve as input to the environment. The environment will then respond to the input by either giving a reward, or a penalty, based on the penalty probability c_i associated with α_i . This response serves as the input to the automata. Based upon the response from the environment and the current information accumulated so far, the learning automata decide on its next action and the process repeats. The intention is that the learning automata gradually converge toward an ultimate goal.

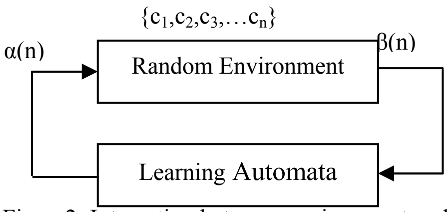

Figure2: Interaction between environment and automata

3.1 Fixed Structure Learning Automata

Fixed structure automata exhibit transition and output matrices which are time invariant. A=$\{\alpha,\beta,F,G,\varphi\}$ is a fixed structure automata which $\alpha= \{\alpha_1, ..., \alpha_r\}$ is the set of r actions offered by the environment that the learning automata must choose from, $\beta= \{0, 1\}$ is the set of inputs from the environment, φ is set of inner state of automata, F is set of updating inner state automata based on exist state automata and penalty and reward of environment, G is choosing action function based on new state of automata

3.2 Variable Learning Automata

Variable structure automata exhibit transition and output matrices which are change with time, a variable learning automata can be formally defined as a quadruple (Oommen1, 2003):

$$A= \{\alpha, P, b, T\} \tag{7}$$

where, $\alpha = \{\alpha_1, ..., \alpha_r\}$ is the set of r actions offered by the environment that the LA must choose from.
$P = [p_1(n), ..., p_r(n)]$ is the action probability vector where pi represents the probability of choosing action α_i at the nth time instant.
$\beta = \{0, 1\}$ is the set of inputs from the environment where '0' represents a reward and '1' a penalty.
$T: P \times \beta \rightarrow P$ is the updating scheme. and defines the method of updating the action probabilities on receiving an input from the environment.

If(β=1&& α_i is chosen) then $P_i(n+1)=P_i(n)+\alpha[1- P_i(n)]$
If(β=1&& α_i is chosen) then $P_j(n+1)=(1-a)P_j(n)$ $i \neq j \forall j$ (8)

If(β=0&& α_i is chosen) then $P_i(n+1)=(1-b)P_i(n)$
If(β=0&& α_i is chosen) then $P_j(n+1)=b/(r-1)+(1-b)P_j(n)$ $i \neq j \forall j$

According to equation 8 if a and b be equal the learning algorithm will be known as linear reward penalty. If b<<a the learning algorithm will known as linear reward epsilon penalty and if b=0 the learning algorithm will be a linear reward inaction.

4. LEARNING CELLULAR ATOMATA

Learning cellular automata A and its environment E are defined as follows (Fei Qian 2001).

$$A = \{U,X, Y, Q,N, \xi, F,O, T\} \tag{9}$$
$$E = \{Y,C, r\} \tag{10}$$

where, $U = \{u_j, j = 1, 2, . . . , n\}$ is the cellular space.
$X = \{xj , 0 \leq j < \infty\}$ is the set of inputs
$Y = \{yj , 0 \leq j < \infty\}$ is the set of outputs
$N = \{n1, \cdots , n|N|\}$ is the list of neighborhood relations.
$Q = \{qj , 0 \leq j < \infty\}$ is the set of internal states.
$\xi : U \rightarrow \Omega, \Omega \subset U$ is the neighborhood state configuration function

$F : Q \times X \times r \rightarrow Q$ is the stochastic state transition function
$O : Q \rightarrow Y$ is the stochastic output function
$Q(t + 1) = T(Q(t))$ is the reinforcement scheme.
$C = \{cj\ ,\ 0 \leq j < \infty\}$ is the penalty probability distribution.
$r = \{rj\ ,\ 0 \leq j < \infty\}$ is the reinforcement signal.

5. USING CELLULAR LEARNING ATOMATA FOR POSTCLASSIFICATOIN

In order to use cellular learning automata for improving classification accuracy, a cellular learning with 8 neighbour structures is considered, and the following steps which include choosing an action by automata, compute penalty probability by environment, updating neighbour functions and updating inner state are considered.

5.1.1 Action: Action $\boldsymbol{\alpha}_i$ is choosing one of two classes which have more probability; at initial state it choose randomly by automata.

5.1.2 Penalty probability: penalty probability c_i is associated with action $\boldsymbol{\alpha}_i$ which is chosen by environment. The environment considers two criteria for evaluating action automata: pixel entropy for local optimization and omission error for global optimization of each class. Once the automata choose an action that lead to increase the entropy of pixel, environment gives it penalty. After each iteration if the omission error decreased the environment will give reward to the automata's action. Amount of reward and penalty is compute as follows:

$$C_i = a*C_{1i}+b*C_{2i} \tag{11}$$

where C2i =omission error 0<a,b<1 , a+b=1

$$C_{1i} = -\sum_{i=1}^{n} p(\omega = \omega_i / x) * \log_2 p(\omega = \omega_i / x) \tag{12}$$

The amount of C_i maps to 0 and 1 as follows:

$$\text{If } (C_i \leq 0.5) \text{ then } \beta=1 \text{ else } \beta=0 \tag{13}$$

5.1.3 Neighbour function between automata: in order to compute the inner state of automata it should compute neighbour function between automata. We use equation 5 in which way that the C_i affects on neighbour function between automata.

5.1.4 Computing inner state of automata: in this stage, at first the local probabilities of pixels based on two stage of percipience memory of neighbour pixel which refers to penalty probability are computed. Then, an updating probability role which depends on local probabilities, initial inner state and neighbor function was introduced. After that, inner state of automata is computed by probability role.
The algorithm executes the steps mentioned already and continues until reach to a best situation; the best situation is a state where pixels have less entropy with the classes having less omission error.

6. EVALUATION AND EXPERIMENT RESULTS

In order to evaluate the algorithm of post processing, a subset image (Figure 3) which is a portion of the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) of hyperspectral data is used. This image was taken over an agricultural area of California, USA in 1994. This data has 220 spectral bands about 10 nm apart in the spectral region from 0.4 to 2.45 μm with a spatial resolution of 20 m. The subset image is 145 by 145 pixels and its corresponding ground truth map is shown in Figure 4 .the image area has 12 classes.

Figure 3. AVIRIS image

Figure 4. Grand truth of area with 12 classes

At first some noisy bands were put away. In order to separate noise, and to extract original signal from image bands the minimum noise fraction transform was performed. Based on eigenvalue of components we chose components which had high variance; therefore the original image dimension was reduced. We used 46 components which contain high percent of original image content information. These features are used for classification. In order to compute classification parameters and endemember selection the training sample with proper schema were introduced; also test sample for computing overall accuracy and classification assessment were picked. At the end, the image was classified by ML (figure 5) and LSU algorithms. The results of classification were sequence, rule images and fraction images which used for post classification.

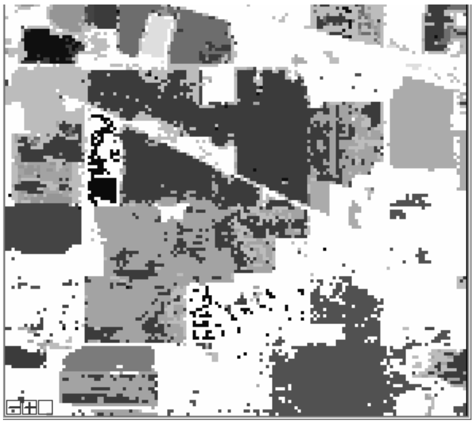

Figure 5. Maximum likelihood classification

We implemented two algorithms of cellular learning automata and probability label relaxation. These two algorithms were used for post classification of the result of MLC and LSU classification which following results (table 1) for test samples were obtained.

Classification Algorithm	ML			LSU		
Post processing algorithm	Without post process	PLR	CA	Without post process	PLR	CA
Overall accuracy	68.01	72.2	84.3	74.40	78.3	87.5

Table 1. The result of post processing algorithm

Noticing to the result, it could be realised that the accuracy of images was increased, and the cellular learning automata adapted itself better to the environment as compared of probability relaxation algorithm. Another interesting result is that the result of CLA for two classification algorithms is close together. Therefore it could be said that the CLA could overcome the result of poor classification such as MLC in Hyperspectral images. And also CLA could be used for transit fraction images computed by LSU to image classified and it is useful for accuracy assessment of sub pixel classifier. Another advantage of LA is that the CLA compensates the poor result of classification algorithm and it isn't so sensitive to initial probability (state) but PLR is too sensitive to the initial probability. In addition the result of CLA algorithm is almost independent from initial probabilities and with respect to two parameters of entropy and omission error the CLA algorithm tries to optimize these parameters and to reach a global optimization. However in PLR it is possible to algorithm satisfied in local optimization. The two parameters of a and b in equation 11 affect to the results of post processing and it depends on our bias to local or global optimization. The prosperous of algorithm depends on schema which designs environment so active in which way, response actually to the action of automata and compute penalty and reward in a real way. One of the disadvantage which experiments showed was that the CLA takes more time as compared with PLR and we should control it by the number of iterations.

7. CONCLOSION

The algorithm developed is so flexible that can change label pixels to reach an agreement between neighbour pixels and decrease chaos in environment of image classified. Therefore cellular automata have good potential for dealing with problems which need to find the best choice until transiting from chaos environment to order environment.

REFERENCES

References from Books:

Mather, paul M 1999. Computer processing of remotely-sensed images, john wiely& sons,pp.202-203

Richards, A. John,1993. Remote sensing digital image analysis, Third edition, Springer- Verlag Berlin Heidelberg, Printed in Germany, pp .196-201

References from websites:

Franciscus Johannes Maria van der Wel, 2000. Assessment and visualisation of uncertainty in remote sensing land cover classifications Faculteit Ruimtelijke Wetenschappen Universiteit Utrecht Netherlands
www.library.uu.nl/digiarchief/dip/diss/1903229/ref.pdf
(accessed 20 sep. 2003)

Fei Qian ,Yue Zhao Hironori Hirata, 2001. Learning Cellular Automata for Function Optimization Problems T.IEE Japan, Vol. 121-C, No.1.
http://www.iee.or.jp/trans/pdf/2001/0101C_261.pdf
(accessed 20 sep. 2003)

Oommen1,B. John and T. Dale Roberts,2003 continuous learning automata solutions to the capacity assignment problem, School of Computer Science Carleton University Ottawa;
http://www.scs.carleton.ca/~oommen/papers/CascaJn.PDF
(accessed 20 sep. 2003)

zur Erlangung des Grades, 1999. A New Information Fusion Method for Land-Use Classification Using High Resolution Satellite Imagery An Application in Landau, Germany Dissertation
archimed.uni-mainz.de/pub/2000/0004/diss.pdf
(accessed 20 sep. 2003)