Classification of Soil and Vegetation by Fuzzy K-means Classification and Particle Swarm Optimization

M. Chapron

ETIS, ENSEA, UCP, CNRS, 6 avenue du ponceau

95014 Cergy-Pontoise, France

chapron@ensea.fr

Abstract.

Precision Agriculture is concerned with all sorts of within-field variability, spatially and temporally, that reduces the efficacy of agronomic practices applied in a uniform way all over the field. Because of these sources of heterogeneity, uniform management actions strongly reduce the efficiency of the resource input to the crop (i.e. fertilization, water) or for the agrochemicals used for pest control (i.e. herbicide). In particular, weed plants are one of these sources of variability for the crop, as they occur in patches in the field. Detecting the location, size and internal density of these patches, along with identification of main weed species involved, open the way to a site-specific weed control strategy, where only patches of weeds would receive the appropriate herbicide (type and dose). Herein, the first stage of recognition method of vegetal species, the classification of soil and vegetation, is described and is based upon the fuzzy k-mean classification (FKC) and on particle swarm optimization (PSO).

Key words: Fuzzy k-mean classification, particle swarm optimization, precision agriculture

1 Introduction

The pattern recognition or classification of vegetal species is one of the most relevant and difficult problems of precision agriculture. Instead of weed pulverization everywhere on the field, the farmer wants to pulverize only badly-infested sectors. It has a big impact on the chemical pollution due to herbicides especially in the water present in the soil which contains toxic products such as atrazine sprayed in big quantities and everywhere on cereal fields. The fast technological developments in electronic devices such as cameras and computers permit us to design real time techniques for recognizing maize and weeds. Therefore, herbicides will only be sprayed on patches of weeds in the field and not uniformly as performed in conventional agriculture. A lot of work has been carried out in this domain, see for instance [1] or [2] or [3]. In the following pages, we will present the FKC [4] and PSO [5,6,7,8] which classify the color pixels of input images in two classes: soil and vegetation. Finally, results of these two methods will be compared and interpreted and a conclusion will be proposed.

2 Fuzzy k-mean classification

Usually, this classification method is named fuzzy c-means classification, herein this technique will be termed fuzzy k-mean classification because of its link to the well known k-mean classification.
Bezdek [4] made a lot of work about this classification and extensions. The classical model is going to be described. Let us introduce the membership functions u_{ji} that corresponds to the degree that any pixel i of the whole input image belongs to the cluster or class j. This method is an extension of the usual k-mean classification technique and is associated to the subsequent optimization problem:

$$\begin{gathered}\text{Min } \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ji}^{r}\ |\mathbf{p}_i - \mathbf{m}_j|^2 \text{ according to the constraints:}\\ \sum_{j=1}^{C} u_{ji} = 1 \text{ with } 1\leq i\leq N \\ u_{ji} \geq 0 \text{ with } 1\leq i\leq N \text{ and } 1\leq j\leq C\end{gathered} \tag{1}$$

N is the number of image pixels, C is the number of classes, p_i is the value of pixel i and r (r>1) is a weight exponent which controls the fuzziness for pixel i of belonging to the cluster j represented by the following center of gravity $\mathbf{m}_j$:

$$\mathbf{m}_j = \frac{\sum_{i=1}^{N} \mathbf{p}_i u_{ji}^r}{\sum_{i=1}^{N} u_{ji}^r} \quad (2)$$

The basic algorithm is iterative and works as follows:

1. Give a value to r and randomly initialize the membership functions u_{ji} to a value between 0 and 1 for example
2. Compute the gravity centers of the clusters by equation (2)
3. Compute the distances d_{ij} between $\mathbf{p}_i$ and $\mathbf{m}j$
4. Compute the membership functions u_{ji} :
   if $d_{ic}=0$ for some class c then $u_{ci}=1$ and $u_{ji}=0$ for $j\neq c$
   else
   $u_{ji}=\frac{1}{\sum_{k=1}^{C}(\frac{d_{ij}}{d_{ik}})^{\frac{2}{r-1}}}$ (r>1)
5. Loop from 2. until convergence.

The membership functions u_{ji} converge either to 1 or 0 with this algorithm. In our case C is equal to 2.

3 Particle Swarm Optimization and Classification

PSO algorithms are inspired from social behavior of bird flocks where birds are replaced by particles in the solution space. Each particle of the swarm represents a potential solution of the classification problem considered as a cost function to be minimized. In a PSO framework, any particle navigates in the multidimensional search space, adjusting its position according to its own trajectory and to those of its neighbor particles. The particles try to fly to a minimum of the fitness function which must be minimized. x_i denotes the current position of the particle i while v_i corresponds to the current velocity of this particle i and y_i the best position of this particle up to now.
If f is the objective function or criteria then the best position of each particle is updated as below:

$$\mathbf{y}_i(t+1)=\begin{cases} \mathbf{y}_i(t) \text{ if } f(\mathbf{x}_i(t+1)) \geq f(\mathbf{y}_i(t)) \\ \mathbf{x}_i(t+1) \text{ if } f(\mathbf{x}_i(t+1)) < f(\mathbf{y}_i(t)) \end{cases} \quad (3)$$

Two extreme strategies exist, the first one *lbest* takes into account the behavior of the neighbor particles and the second one *gbest* integrates the dynamics of the whole swarm. *gbest* is carried out at the beginning of the computing process and *lbest* is chosen at the end. In the context of *gbest*, the vector $\hat{y}$ is the position of the best particle:

$$\hat{\mathbf{y}}(t+1) = \text{arg\_min } (f(\mathbf{y}_0(t)), f(\mathbf{y}_1(t)), .., f(\mathbf{y}_s(t)) \quad (4)$$

with s+1 the number of particles of the swarm.
The particle flock is divided into overlapped neighborhoods in the model *lbest*. Let N_j the particle neighborhood of the current particle j, the best position $\hat{\mathbf{y}}_j(t+1)$ of N_j particles is such that:

$$f(\hat{\mathbf{y}}_j(t+1)) = \min\left(f(\mathbf{y}_i(t))\right), \forall i \in N_j \quad (5)$$

If N_j is the entire swarm, $\hat{\mathbf{y}}_j(t+1)$ is obviously $\hat{\mathbf{y}}(t+1)$.
Other parameters and equations of particle dynamics are presented below:

$$\mathbf{v}_i(t+1)=wv_i(t)+c_1r_1(t)(\mathbf{y}_i(t)-x_i)+c_2r_2(t)(\hat{\mathbf{y}}_i(t) - \mathbf{x}_i(t)) \quad (6)$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (7)$$

where w is the inertia weight, c_1 and c_2 are acceleration constants and $r_1(t)$, $r_2(t)$ parameters which follow the uniform distribution U(0,1) in the interval [0,1]. The above iterative procedure consists in calculating at each step $\mathbf{v}_i(t+1)$, $\mathbf{x}_i(t+1)$ and $\hat{\mathbf{y}}_j(t+1)$. w is settled in order to apply *gbest* strategy at the beginning of the computation and *lbest* strategy after. The algorithm finishes when the velocity $\mathbf{v}_i(t+1)$ is close to 0 and the solution is given by $\hat{\mathbf{y}}_j(t+1)$.

3.1 PSO classification of soil and vegetation

In our image classification, any particle $\mathbf{x_i}$ encapsulates the concatenation of the mean vectors $\mathbf{m_{i1}}$,$\mathbf{m_{i2}}$ of the 2 respective clusters soil and vegetation such that $\mathbf{x_i=(m_{i1},m_{i2})}$. The objective function to optimize for each particle is:

$$f(\mathbf{x}_i,Z)=w_1\bar{d}_{max}(\mathbf{Z},\mathbf{x}_i) + w_2(z_{max} - d_{min}(\mathbf{x}_i)) \quad (8)$$

where $\mathbf{Z}$ is a matrix which associates the pixels to the 2 classes soil and vegetation. z_{max} is equal to 2^l-1 for a l-bit image. The elements of Z corresponds to a pixel $\mathbf{z}_p$ which belongs to the cluster C_{ij} of particle i, the constants w_1 and w_2 are empirical. $\bar{d}_{max}(\mathbf{Z},\mathbf{x}_i)$ is the maximum average Euclidean distance of particles to their associated classes:

$$\bar{\mathrm{d}}_{\max}(\mathbf{Z},\mathbf{x}_i) = \max_{j=1,2}\{\textstyle\sum_{\forall \mathbf{z_p}\in \mathbf{C}_{ij}} \mathrm{d}(\mathbf{z}_p,\mathbf{m}_{ij})/|C_{ij}|\} \tag{9}$$

$|C_{ij}|$ is the cardinality of the set C_{ij}. This fitness function f maximizes the distance between the clusters soil and vegetation by minimizing $-d_{min}(\mathbf{x}_i)$ and minimizes the intra-distance between pixels and their cluster centers of gravity calculated by $\bar{\mathrm{d}}_{\max}(\mathbf{Z},\mathbf{x_i})$ and $d_{min}(\mathbf{x}_i)=\min\{d(\mathbf{m}_{i1},\mathbf{m}_{i2})\}$ the minimum Euclidean distance between any pair of clusters.

According to the relative values of w_1 and w_2 one of these 2 sub costs ($\bar{\mathrm{d}}_{\max}$ and dmin) is more relevant.

The PSO image classification algorithm follows the subsequent stages:

1. Initialize each particle by the 3 color components (red, green and blue) of the soil and vegetation, so each particle $\mathbf{x}_i$ is composed of 6 components divided in $\mathbf{m}_{i1}=(R_{i1},G_{i1},B_{i1})$ and $\mathbf{m}_{i2}=(R_{i2},G_{i2},B_{i2})$.
2. For t=1 to t_{max} do

    2.1 for each particle i do

    for each pixel $\mathbf{z}_p$ do

    compute $d(\mathbf{z}_p, \mathbf{m}_{ij})$ for all cluster C_{ij}

    end for

    assign $\mathbf{z}_p$ to C_{ij} with:

    $\mathrm{d}(\mathbf{z}_p,\mathbf{m}_{ij}) = \min_{c=1,2}\{\, d(\mathbf{z}_p, \mathbf{m}_{ic})\}$

    Compute the fitness $f(x_i(t),Z)$

    end for

    2.2 Find the global best solution $\hat{\mathbf{y}}_j(t+1)$.

    2.3 Update the class centroids using equations () and ()

    End For

The parameter w is chosen such that $w>0.5(c_1+c_2)-1$

t_{max} is the maximum number of iterations, w(t) can follow the different decreasing laws [6]:

. Linear decreasing rule:

$w(t)=(w(0)-w(n_t))(\frac{n_t-t}{n_t}) + w(n_t)$

. Nonlinear decreasing rules

$w(t+1)=\frac{(w(t)-0.4)(n_t-t)}{n_t+0.4}$

$w(t+1)=a\, w(t) \quad (0<a<1)$

$w_i(t+1)=w(0)+(w(n_t)-w(0))(\frac{e^{m_i(t)}-1}{e^{m_i(t)}+1})$

with $m_i(t)=\frac{f(\hat{y}_1(t))-f(x_i(t))}{f(\hat{y}_1(t+1))+f(x_i(t)}$

n_t is the maximal number of iterations of the algorithm, w(0) is the initial inertia weight, w(t) is the weight at step t, $w(n_t)$ is the final weight. $w(0)>w(n_t)$. As t increases, w(t) decreases and each particle $\mathbf{x_i}$ is more and more influenced by the dynamics of the best particle.

4 Results

Let $\mathbf{\Phi}$ be the mapping between the set P of pixels and K the space of vectors of their color components red, green and blue:

$$\mathbf{\Phi} : P\rightarrow \ K = \{\begin{pmatrix} r \\ g \\ b \end{pmatrix}, (r,g,b) \in \{0,1, \ldots,255\}^3\}.$$

Several trials have been tested on the colors red R, green G ,blue B and also on the normalized red r and green g which are less sensitive to the lightning quality. The normalized red and green are given by the simple following formulas :

$$r=255R/(R+G+B),\ g=255G/(R+G+B) \tag{10}$$

Other more complex objective functions have been carried out such as computing the sum of the largest eigen-values λ_{+1} and λ_{-1} of the 2 covariance matrices below instead of $\bar{\mathrm{d}}_{\max}(Z,x_i)$ in the objective function, λ which represents the dispersion of data with respect to the principal axis, is computed by power algorithm. This change of the first sub objective implies an increase of CPU time but the results are very good. The covariance matrices are given by the following formulas for the classes +1 (vegetation) and -1 (soil):

$$\boldsymbol{MCV_{+1}} = \frac{1}{m_{+1}}\sum_{(x_i,+1)\in z}\begin{pmatrix} (r_i - M_{r_{+1}})^2 & (r_i - M_{r_{+1}})(g_i - M_{g_{+1}}) & (r_i - M_{r_{+1}})(b_i - M_{b_{+1}}) \\ (g_i - M_{g_{+1}})(r_i - M_{r_{+1}}) & (g_i - M_{g_{+1}})^2 & (g_i - M_{g_{+1}})(b_i - M_{b_{+1}}) \\ (b_i - M_{b_{+1}})(r_i - M_{r_{+1}}) & (b_i - M_{b_{+1}})(g - M_{g_{+1}}) & (b_i - M_{b_{+1}})^2 \end{pmatrix} \quad (11)$$

$$\boldsymbol{MCV_{-1}} = \frac{1}{m_{-1}}\sum_{(x_i,-1)\in z}\begin{pmatrix} (r_i - M_{r_{-1}})^2 & (r_i - M_{r_{-1}})(g_i - M_{g_{-1}}) & (r_i - M_{r_{-1}})(b_i - M_{b_{-1}}) \\ (g_i - M_{g_{-1}})(r_i - M_{r_{-1}}) & (g_i - M_{g_{-1}})^2 & (g_i - M_{g_{-1}})(b_i - M_{b_{-1}}) \\ (b_i - M_{b_{-1}})(r_i - M_{r_{-1}}) & (b_i - M_{b_{-1}})(g - M_{g_{-1}}) & (b_i - M_{b_{-1}})^2 \end{pmatrix} \quad (12)$$

Another experiment has been performed: always in the first sub objective we have introduced the gaussian kernel GK defined by :

$$\mathrm{GK}(x, y) = e^{\frac{-|x-y|^2}{2\sigma^2}} \quad (13)$$

in the calculation of distances $d(z_p, m_{ij})$ but this does not bring an actual improvement. We tried different laws for w(t) and adapted the different parameters in order to get the convergence before t_{max}=20.

Photo 1: Input Image

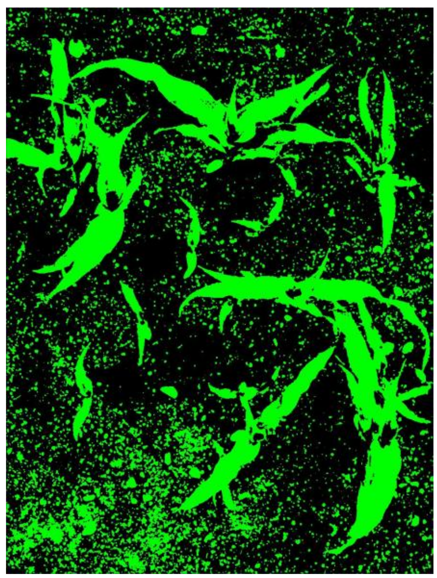

Photo 2: Result image of FKC

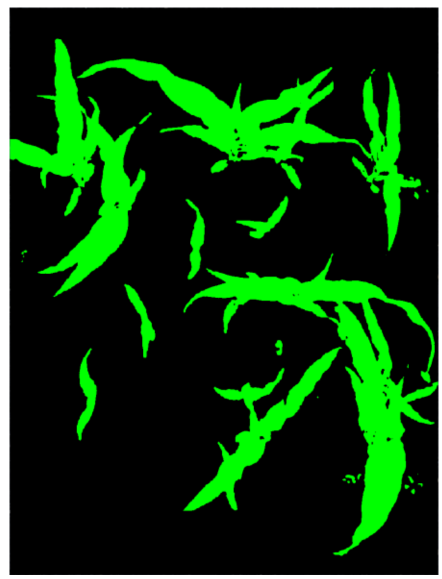

Photo 3: Result image of PSO

5 Conclusion

The PSO based classification method gives very good results compared to the ones obtained from the fuzzy k-mean classification if we take into account the number of false positives and true negatives pixels. These methods have been tested on different images more or less difficult due to the variations of lightings and soil types and each time the PSO classification provides better results than the FKC and needs more CPU time than FKC. The stones are in the same class as soil which is right for the application. Other algorithms based on artificial ants [9] will be adapted to this classification problem.

References

[1] CHAPRON M., ASSEMAT L., BOISSARD, P., HUET P., *Weed and maize recognition using 2D and 3D data fusion*. Third European Conference Precision Agriculture, 2001.

[2] MANH A.G., RABATEL G., ASSEMAT L., ALDON M.J., *In field classification of weed leaves by machine vision using deformable templates* Third European Conference Precision Agriculture, 2001.

[3] TIAN L., SLAUGHTER D. C., NORRIS R. F., *Machine Vision Identification of Tomato Seedlings for automated weed control*, 2000

[4] BEZDEK J. et al, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Springer**,** 2005

[5] OMRAN M., SALMAN A., ENGELBRECHT A.P., *Image Classification Using Particle Swarm Optimization*, Fourth Asia Pacific on Simulated Evolution and Learning, p 370-374, 2002.

[6] OMRAN M., ENGELBRECHT A.P., SALMAN A., *Dynamic Clustering using Particle Swarm Optimization with Application in Unsupervised Image Classification*, IEEE Transaction on Engineering, Computing and Technology, vol 9, november 2005.

[7] Hung C. C., Purnawan H., Gelbukh A., Morales E., *A Hybrid Rough K-mean Algorithm and Particle Swarm Optimization for Image Classifiocation*, Advanced in Artificial Intelligence, p585-593, Springer,2008

[8] ENGELBRECHT A.P., *Computational Intelligence*, Second Edition, John Wiley, 2007.

[9] BONMARCHE N., GUINARD F., SIARRY P., *Artificial Ants*, ISTE-John Wiley, sept 2010.