Endmember Extraction of Hyperspectral Remote Sensing Images Based on the Discrete Particle Swarm Optimization Algorithm

Paper Review

Miguel A. Veganzones

Grupo de Inteligencia Computacional
Universidad del País Vasco

2012-02-10

Paper

IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 49, NO. 11, NOVEMBER 2011 4173

Endmember Extraction of Hyperspectral Remote Sensing Images Based on the Discrete Particle Swarm Optimization Algorithm

Bing Zhang, Xun Sun, Lianru Gao, and Lina Yang

Motivation

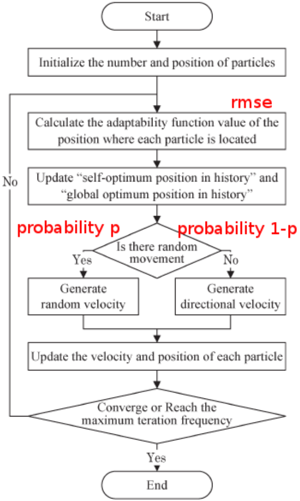
- Endmember induction using Discrete-PSO.
- Solution: a subset of the pixels, $E = \{\mathbf{e}_j\}_{j=1}^{m} \subset \{\mathbf{r}_i\}_{i=1}^{n}$, where:
  - m is the number of endmembers $\mathbf{e}_j$ in the solution.
  - n is the number of pixels $\mathbf{r}_i$ in the image.
- Criterium: minimize root-mean-squared-error $\mathrm{rmse}\left(\{\mathbf{r}_i\}_{i=1}^{n}, \{\hat{\mathbf{r}}_i\}_{i=1}^{n}\right)$, where:
  - $\hat{\mathbf{r}}_i$ is a remixing pixel using E and the abundancies obtained by FCLSU.

- Feasible solution space:

$$X_{n,m} = \left\{ (x_1, x_2, \ldots, x_n) \,|\, x_i \in \{0,1\}, \sum_{i=1}^{n} x_i = m \right\}$$

- $\mathbf{x}_k(t)$ and $\mathbf{v}_k(t)$ denote respectively the k-th particle's position and velocity at time t.
- $\mathbf{x}_{k,\mathrm{best}}(t)$ specify the k-th particle's self-optimum position at time t.
- $\mathbf{x}_{\mathrm{gbest}}(t)$ specify the global optimum position before time t.
- $\mathbf{x}_k(t), \mathbf{x}_{k,\mathrm{best}}(t), \mathbf{x}_{\mathrm{gbest}}(t) \in X_{n,m}$.

Algorithm


Start
Initialize the number and position of particles
rmse
Calculate the adaptability function value of the position where each particle is located
No
Update "self-optimum position in history" and "global optimum position in history"
probability p
probability 1-p
Is there random movement
Yes
No
Generate random velocity
Generate directional velocity
Update the velocity and position of each particle
Converge or Reach the maximum teration frequency
Yes
End

Updating functions

$$\mathbf{x}_k(t+1) = \mathbf{x}_k(t) + \mathbf{v}_k(t)$$

$$\mathbf{v}_k(t+1) = \begin{cases} T\left((\mathbf{x}_{k,best}(t) - \mathbf{x}_k(t)) \right. \\ \qquad \left. + (\mathbf{x}_{gbest}(t) - \mathbf{x}_k(t))\right) \\ R\left(\mathbf{x}_k(t)\right) \end{cases}$$

- $\mathbf{v}_k \in \{-1, 0, 1\}^n$.
- T and R are both random selection functions.
  - T: directional movement.
  - R: random movement.
- **Authors say nothing about how to ensure $\sum_{i=1}^{n} = m$.**

Experiments

- Real hyperspectral image: Cuprite, Nevada (U.S.A.).
  - 400×360 pixels and 50 bands.
- Compared to:
  - VCA and N-FINDR for different m and p values.
  - Spectra from the USGS library for $m = 15$ (same as virtual dimensionality) and $p = 0.2$ (best result).
- 20 particles.
- A-priori setting the number of endmembers: $m = \{5, 10, 15, 20\}$.
- Random selection probability: $p = \{0.1, 0.2, 0.5\}$.

The PSO is to search inside the feasible solution space. If the feasible solution space is too large, it will affect the search rate and reduce the computational efficiency, so the maximum noise fraction (MNF) algorithm was first used to reduce the dimensions of the images, and then, the PPI algorithm was used to obtain the 80 candidate endmembers which constitute a feasible solution space, but when calculating the value of the adaptability function, all the images still need to be processed with FCLS to calculate the rmse. In addition, the N-FINDR algorithm needs to reduce the dimensions of the images, so it extracts endmembers from the images after being converted by MNF, while VCA and D-PSO directly extract endmembers from the original images.

Results (I)

TABLE I
COMPARISON OF THE RMSE OF D-PSO, N-FINDR, AND VCA UNDER DIFFERENT PARAMETERS

Number of end-member	Random probability	D-PSO	N-FINDR	VCA	Time(sec)
5	0.1	4.273	8.890	8.380	8.330
	0.2	4.069			3.301
	0.5	4.045			56.567
10	0.1	2.806	7.101	7.043	10.632
	0.2	2.797			7.958
	0.5	2.734			291.430
15	0.1	2.273	5.648	5.537	13.935
	0.2	2.266			25.530
	0.5	2.237			651.534
20	0.1	1.931	4.838	5.232	18.509
	0.2	1.913			21.578
	0.5	1.884			1464.520

TABLE II
RESULTS OF ENDMEMBER COMPARISON

No.	Position	Mineral	Similarity
1	(394,48)	Jarosite	0.890
2	(134,97)	Palygorskite	0.913
3	(82,118)	Alunite	0.823
4	(85,135)	Alunite	0.805
5	(146,164)	Ferrihydrite	0.778
6	(327,170)	Hyalite	0.912
7	(327,182)	Chalcedony	0.903
8	(231,193)	Buddingtonite	0.847
9	(271,230)	Chert	0.890
10	(286,256)	Alunite	0.805
11	(39,266)	Kaolinite	0.772
12	(5,267)	Illite	0.828
13	(84,290)	Montmorillonite	0.911
14	(35,337)	Calcite	0.834
15	(368,348)	Niter	0.782