Assessment of a Modified Version of the EM Algorithm for Remote Sensing Data Classification

Thales Sehn Korting, Luciano Vieira Dutra,
Guaraci José Erthal, and Leila Maria Garcia Fonseca

Image Processing Division
National Institute for Space Research – INPE
São José dos Campos – SP, Brazil
{tkorting,dutra,gaia,leila}@dpi.inpe.br

Abstract. This work aims to present an assessment of a modified version of the standard EM clustering algorithm for remote sensing data classification. As observing clusters with very similar mean vectors but differing only on the covariance structure is not natural for remote sensing objects, a modification was proposed to avoid keeping clusters whose centres are too close. Another modification were also proposed to improve the EM initialization by providing results of the well known K-means algorithm as seed points and to provide rules for decreasing the number of modes once a certain a priori cluster probability is very low. Experiments for classifying Quickbird high resolution images of an urban region were accomplished. It was observed that this modified EM algorithm presented the best agreement with a reference map ploted on the scene when compared with standard K-means and SOM results.

1 Introduction

Many supervised and unsupervised parametric classification methods usually follow a unimodal assumption for class conditional feature distribution. In general, this assumption is not suitable for remote sensing data, particularly for those of very high spatial resolution. One way to improve classification results is describing the class conditional distribution as a mixture of distributions.

The finite mixture model (FMM) is a useful tool for multimodal density estimation. Given the observed data X, an FMM $p(\mathbf{x};\Theta)$ where $\mathbf{x} \in X$ can be defined as

$$p(\mathbf{x};\Theta) = \sum_{j=1}^{M} p(\mathbf{x}|C_j;\theta_j)P_j \tag{1}$$

where M is the number of components, P_j is the jth mixing proportion, $p(\mathbf{x}|C_j)$ the corresponding component density and Θ denotes the parameter vector of the density. If the jth underlying density is the multivariate gaussian

I. Bloch and R.M. Cesar, Jr. (Eds.): CIARP 2010, LNCS 6419, pp. 476–483, 2010.

$p(\mathbf{x}|C_j) = \left(|2\pi\Sigma_j| \exp((\mathbf{x}-\mu_j)^T \Sigma_j^{-1}(\mathbf{x}-\mu_j))\right)^{-1/2}$, with mean vector μ_j and covariance matrix Σ_j, the model is refered to as the gaussian mixture model (GMM).

One way to estimate mixture models is to assume that data points have a "membership" to the unimodal components of data distributions and such membership is unknown. The objective is to estimate suitable parameters for the model, where the connection to the data points is represented as their membership in the individual model distributions.

In statistical pattern recognition, mixture models allow a formal approach to unsupervised learning [5]. A standard method to estimate FMM from observed data is the *Expectation-Maximization* (EM) algorithm, firstly proposed by [3].

Given a complete set $Z = (X, Y)$ where X is the observed data (the incomplete data) and Y the unobserved data, the joint probability density of Z is given as $p(X, Y; \Theta)$. The ML estimate of Θ is obtained by maximizing the incomplete-data log-likelihood function

$$L(\Theta; X) = \log p(X; \Theta) = \log \int p(X, Y; \Theta) dY \tag{2}$$

The incomplete data log-likelihood function is maximized through EM algorithm by iteratively maximizing the expectation of the complete data log-likelihood function given by

$$L_c(\Theta; Z) = \log p(X, Y; \Theta) \tag{3}$$

At $(t+1)$th iteration the E-step of the algoritm computes the expected complete data log-likelihood as follows:

$$Q(\Theta|\Theta(0)) = E[L_c(\Theta; Z)|X; \Theta(t)] \tag{4}$$

and the M-step calculates Θ by maximizing $Q(\Theta|\Theta(t))$.

EM is an iterative procedure which under mild conditions converges to a (local) maximum of $L(\Theta; X)$ depending on the initial solution $\Theta(0)$.

In other words, EM is a general method of estimating the features of a given data set, when the data are incomplete or have missing values [1]. Finite mixture models are able to represent arbitrarily complex probability density functions [4]. This fact makes EM proper for representing complex likelihood functions. This algorithm has been used in several areas, such as image reconstruction, signal processing, and machine learning [9], [11].

Being an iterative procedure, EM presents high computational cost. This article presents a variation of the algorithm EM to improve the classification results, particularly for remote sensing applications. It is done first taking in account particularities of optical remote sensing data distribution, and providing the first set of parameters from K-means algorithm and by performing clustering validation techniques.

The paper is organized as follows. Section 2 describes the basic EM approach and its application to mixture models. In Section 3 we show our main contribution describing the improved EM. Section 4 presents some experimental

results for applying the proposed method to urban remote sensing images as well as a discussion about the classification method performance. Finally, Section 5 presents the conclusion.

2 The Standard EM for GMM

We assume that the algorithm will estimate M class distributions $C_j, j = 1, \ldots M$. For each of the N input vectors $\mathbf{x}_k \in X, k = 1, \ldots N$, the algorithm calculates the probability $P(C_j|\mathbf{x}_k)$ [12]. The highest probability will point to the vector's class.

To apply EM for remote sensing imagery analysis we have created the input vectors with one vector per pixel. The vector contains the pixel values for each spectral channel in the image. An image with l bands produces a $l-D$ attribute space.

The EM algorithm works iteratively by applying two steps: the E-step (*Expectation*) and the M-step (*Maximization*). Formally, $\hat{\Theta}(t) = \{\mu_j(t), \Sigma_j(t), P_j(t)\}$, $j = 1, \ldots M$ stands for successive parameter estimates. In the standard EM, $\hat{\Theta}(0)$ is randomly defined, and EM approximates $\hat{\Theta}(t)$ to the real data distribution when $t \to \infty$.

2.1 E-Step

This step calculates the conditional expectation of the complete *a posteriori* probability function. Each cluster probability, given a certain attribute-vector, is estimated as following:

$$P(C_j|\mathbf{x}) = \frac{|\Sigma_j(t)|^{-\frac{1}{2}} e^{\eta_j(t)} P_j(t)}{\sum_{k=1}^{M} |\Sigma_k(t)|^{-\frac{1}{2}} e^{\eta_k(t)} P_k(t)} \tag{5}$$

where

$$\eta_i(t) = -\frac{1}{2}(\mathbf{x} - \mu_i(t))^T \Sigma_i^{-1}(t)(\mathbf{x} - \mu_i(t))$$

2.2 M-Step

This step updates the parameter estimation $\hat{\Theta}(t)$. Given the cluster probabilities, the mean and covariance values for each cluster are estimated as

$$\mu_j(t+1) = \frac{\sum_{k=1}^{N} P(C_j|\mathbf{x}_k)\mathbf{x}_k}{\sum_{k=1}^{N} P(C_j|\mathbf{x}_k)} \tag{6}$$

$$\Sigma_j(t+1) = \frac{\sum_{k=1}^{N} P(C_j|\mathbf{x}_k)(\mathbf{x}_k - \mu_j(t))(\mathbf{x}_k - \mu_j(t))^T}{\sum_{k=1}^{N} P(C_j|\mathbf{x}_k)} \tag{7}$$

The overall probability for each cluster is also calculated in this step as:

$$P_j(t+1) = \frac{1}{N} \sum_{k=1}^{N} P(C_j|\mathbf{x}_k) \tag{8}$$

2.3 Convergence

Both steps, E and M, are performed until convergence, according to

$$\| \mathbf{\Sigma}(t+1) - \mathbf{\Sigma}(t) \|_F < \varsigma \tag{9}$$

where $\| \cdot \|_F$ stands for the Frobenius norm, the square root of the sum of the absolute squares of its elements [12], and ς is a threshold for convergence. The second stop criteria is given by

$$\| \mu(t+1) - \mu(t) \| < \varepsilon \tag{10}$$

where $\| \cdot \|$ is the Euclidean distance between vectors, and ε is second a convergence threshold. When both equations are true, the algorithm reaches convergence and Equation 5 is applied to classify the image.

3 Modifications to the Standard EM Algorithm

In this section we explain our main contributions to the EM algorithm. Figure 1 illustrates our method, which is composed by three main modules. The first one describes the data initialization, followed by the probabilities estimation, and finished by the data classification. The "Initialization" and "Probabilities estimation" modules were adjusted to carry out the improvements in the results.

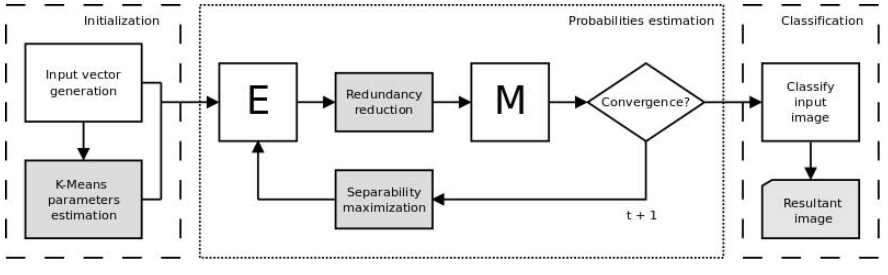

Fig. 1. The improved EM diagram

3.1 Initialization

The instance set $\mathbf{x}$ is built with the pixels of each image spectral channel. [6] used agglomerative hierarquical clustering based on the classification likelihood to estimate the initial parameters for EM. Besides this approach, [10] have also suggested the parameter estimation from K-means. This work employs K-means algorithm for producing the first set of unknown parameters Θ, *i.e.* when $t = 0$. It is important to point out that K-means defines its initial parameters randomly, and provides to our algorithm the clusters means. Therefore, in the beginning the set of covariance matrix is created with identity matrices. By applying this to the EM approach, we reduce the number of iterations, thus reducing computational time.

3.2 Probabilities Estimation

This module performs the iterative procedure for probabilities estimation. We suggest to remove redundant clusters and to maximize the separability between them to correct the number of clusters.

The approach performs cluster exclusion when some cluster presents low probability, according the equation 8. In Figure 1, such operation is defined by the "Redundancy reduction" module. Through a threshold η, the cluster exclusion is defined as:

$$\text{if } P_i(t) < \eta \text{ then exclude cluster } C_i \tag{11}$$

As observing clusters with very similar mean vectors but differing only on the covariance structure is not natural for remote sensing objects, another modification was proposed to avoid keeping clusters whose centres are too close. If a cluster center is approaching another cluster center, one of them has its parameters randomly changed. We define a module called "Separability maximization", as the following equation:

$$\text{if } ||\mu_i(t) - \mu_j(t)|| < \zeta \text{ then } \mu_j(t) = \vartheta \text{ and } \Sigma_j(t) = I \tag{12}$$

where ζ is a threshold, ϑ is a random vector, and I is the identity matrix.

3.3 Classification

After convergence is acchieved, the algorithm classifies each pixel k in the image. The vector $\mathbf{x}_k$ is associated to one class with higher probability. The algorithm finds $P(C_j|\mathbf{x}_k) > P(C_i|\mathbf{x}_k), j \neq i$ and classify $\mathbf{x}_k$ as C_j.

Given the classified image, the next step includes the clusters labeling phase, which is performed manually, and stands for associating the generated clusters to the classes of interest.

4 Results

Figure 2a shows a color composition (R3G2B1) image acquired in January 2004 by the Quickbird satellite, and covers and urban area of São José dos Campos – Brazil. By visual inspection, we can identify four main classes, namely *Shadow, Vegetation, Ground*, and *Roofs*. To analyze the results and compare the obtained agreement with reference regions, we also classified using well known unsupervised methods K-means and Self-Organizing Maps (SOM) [8].

The initial number of clusters was set to 15, a number that was big enough to consider all possibilities of class definitions in the test image. For EM and K-means we set $k = 15$, and for SOM, we created a map with 3×5 neurons. As the algorithm initializations are random we performed 10 classifications, using the same image for each algorithm, trying to avoid sub-optimal solutions. Considering all tests, the minimum detected number of Gaussians, considered

as elements of the mixture, was 9. After classification, we manually assigned the clusters to one of the four reference classes.

We obtained an agreement coefficient with a reference classification map and the best results for each clustering procedure. The best classification results are shown in Figure 2, and the obtained agreement matrices are displayed on Table 1. The overall agreement with reference regions for EM was 70.58% of correct matches, whereas K-means obtained 68.12% and SOM obtained 65%. Kappa indices for every algorithm were $\kappa = 0.557$ for EM, $\kappa = 0.483$ for K-means and $\kappa = 0.474$ for SOM.

EM algorithm presents some drawbacks. Being a local method, it is sensitive to the initialization because the mixture model likelihood function is not unimodal [5]. This was the main reason for using K-means as first set of parameters. For certain mixture types, it may converge to the parameter space boundary, leading to meaningless estimates. It would be expected to get better results for EM

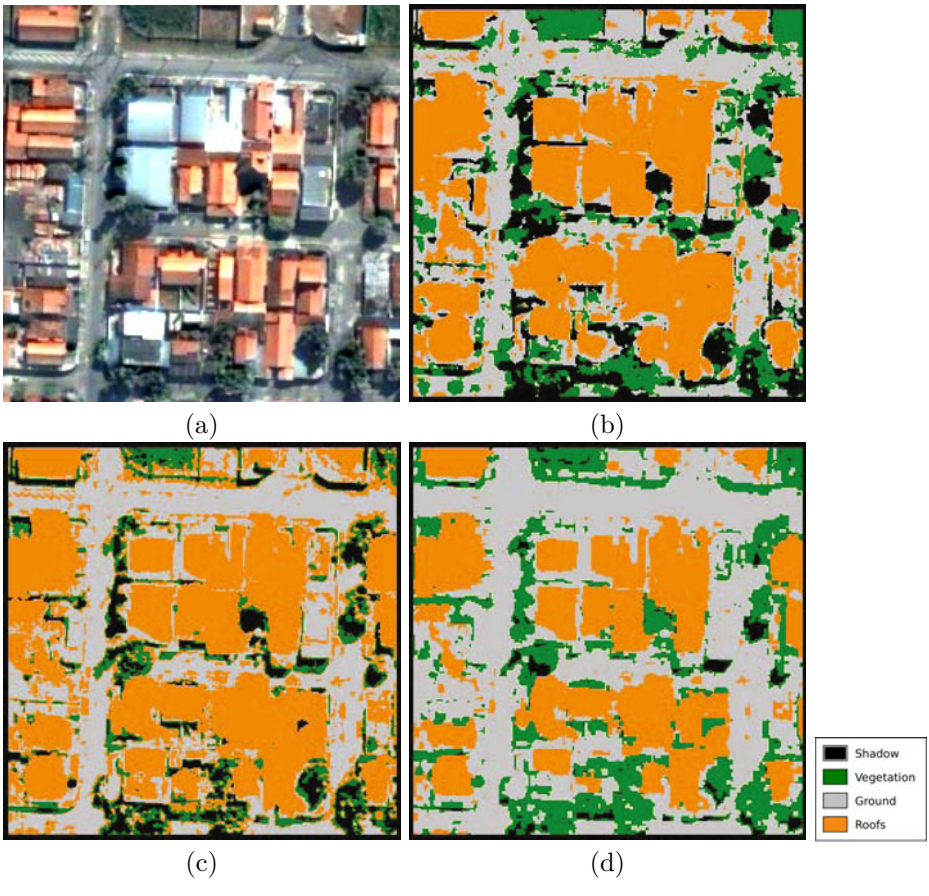

Fig. 2. a) Color composition R3G2B1 of QuickBird scene from São José dos Campos – Brazil. b) Improved EM result, c) K-means result, and d) SOM result.

Table 1. Agreement Matrices. The reference data are displayed in the rows.

EM				
	52	6	0	3
	20	79	13	1
	32	47	245	52
	11	22	101	363

K-means				
	36	14	0	7
	17	38	12	22
	16	26	223	108
	7	20	78	402

SOM				
	16	40	4	1
	3	71	22	0
	1	49	278	19
	5	45	161	285

than for K-means, since it provides the first set of parameters, and improved EM adjust them in a better way, including also the estimation of covariance matrices.

Therefore, to test the better performance of our method we performed several tests using the original EM and the modified EM approach. The experimental tests took into account the processing time until convergence, for both approaches. We used 5 different images, regardless the image used in the previous experiment, with different parameters. Table 2 shows the results considering the image size, number of classes, and computational time until convergence.

Calculating the average values for time speed up, showed at the line $\frac{\Delta t_1}{\Delta t_2}$ of the table, we reach the value 3.35, *i.e.* our improved approach is around $3\times$ faster than the original one. However, even becoming faster than the original approach, EM is still more expensive in terms of processing time than the other methods. It performs calculations of inverse matrix and determinant at each iteration for the whole set of data.

Table 2. Comparison between original and improved approaches

	Image1	Image2	Image3	Image4	Image5
Size	512×512	512×512	200^2	512×384	264×377
# of classes	4	4	5	6	5
Δt_1 original EM	$467s$	$467s$	$103s$	$402s$	$202s$
Δt_2 improved EM	$140s$	$148s$	$29s$	$105s$	$70s$
$\frac{\Delta t_1}{\Delta t_2}$	3.33	3.15	3.55	3.82	2.88

Images classified by pixel-based methods (non region-based) generally present a noisy appearance because of some isolated pixels that are misclassified [7]. As observed in the agreement matrix, the class *Roofs* presented the worst classification results. This was due to the fact that such class varies a lot and some parts of the roofs are very similar to roads, leading to misclassifications.

5 Conclusion

This work has presented improvements to the EM clustering method, by using K-means results as input, and some changes in the "Probabilities estimation" module.

Afterall, one of the main conclusions drawn from the experiments is that mixture models seems to be the best way to characterize the distributions for high resolution images, since the minimum number of detected modes was 9 for a 4 class problem, considering all tested methods. When compared with standard clustering approaches like K-means and SOM, the modified EM algorithm presented the best agreement with the reference map. This fact suggests that the proposed EM algorithm can be adopted as a standard choice for this task. Future works include a complete assessment of our method comparing it with other algorithms such as the original implementation of EM, hierarchical clustering, and fuzzy approaches.

Wrong initial parameters might result in meaningless classification, therefore initial estimation from K-means increased the resultant agreement. We have implemented the algorithm using TerraLib library [2], which is available for free download at `http://www.terralib.org/`.

References

1. Bilmes, J.: A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. TR-97-021, International Computer Science Institute (1998)
2. Câmara, G., et al.: TerraLib: An Open Source GIS Library for Large-scale Environmental and Socio-economic Applications. Open Source Approaches in Spatial Data Handling, 247–270 (2008)
3. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood estimation from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 39 (1977)
4. Figueiredo: Lecture Notes on the EM Algorithm. Tech. rep., Institute of Telecommunication (May 19, 2004)
5. Figueiredo, M., Jain, A.: Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(3), 381–396 (2002)
6. Fraley, C.: How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. The Computer Journal 41(8), 578–588 (1998)
7. Guo, L., Moore, J.: Post-classification Processing For Thematic Mapping Based On Remotely Sensed Image Data. In: Geoscience and Remote Sensing Symposium. IGARSS, vol. 4 (1991)
8. Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (2001)
9. McLachlan, G., Krishnan, T.: The EM Algorithm and Extensions, 1st edn. Wiley Interscience, Hoboken (1997)
10. McLachlan, G., Peel, G., Basford, K., Adams, P.: The EMMIX software for the fitting of mixtures of normal and t-components. Journal of Statistical Software 4 (1999)
11. Moon, T.: The expectation-maximization algorithm. IEEE Signal Processing Magazine 13(6), 47–60 (1996)
12. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Academic Press, London (2003)