

# SUPERVISED GAUSSIAN MIXTURE MODEL BASED REMOTE SENSING IMAGE CLASSIFICATION

---

ONUWA OKWUASHI, ETIM EYO, AND ANIEKAN EYOH

(Received 27, June 2011; Revision Accepted 23, August 2011)

## ABSTRACT

Software like ILWIS and GRASS GIS can be employed for remote sensing image processing and geographic information systems applications. The modules of the aforementioned image processing software are based on conventional multi-class classifiers/algorithms such as maximum likelihood classifier. These conventional multi-class classifiers/algorithms are usually written in programming languages such as C, C++, and python. The objective of this research is to experiment the use of the parametric Gaussian mixture model multi-class classifier/algorithm for multi-class remote sensing task, implemented in MATLAB. MATLAB is a programming language just like C, C++, and python. In this research, a computer program implemented in MATLAB is used to experiment the Gaussian mixture model algorithm. Using the supervised classification technique, both simulated and empirical satellite remote sensing data are used to train and test the Gaussian mixture model algorithm. For the purpose of validating the experiment, the resulting classified satellite image is compared with the ground truth data. For the simulated modelling, twenty-five pixels are used for the modelling, out of which six pixels are used for training while nineteen pixels are used for testing. All the nineteen tested pixels are correctly classified. For the empirical modelling, some of the pixels are wrongly classified; the computed overall accuracy is 85.35%; which indicates substantial agreement between the classification result and the reference data.

**KEY WORDS:** Gaussian Mixture Model; Image Classification; Remote Sensing

## Preamble

The process of relating pixels in a satellite image to known land cover is called “image classification.” The algorithms used to effect the classification process are called “image classifiers” (Mather, 1987). The extraction of land cover information from satellite images using image classifiers has been the subject of intense interest and research in the remote sensing community (Foody & Mather, 2004). Some of the traditional hard classifiers such as minimum distance to means and the box classifiers have been in use in remote sensing studies (Peddle et al., 1994; Rogan et al., 2002; Li et al., 2003; Mahesh & Mather, 2003). Because of the strong desire to maximize the degree of land cover information extracted from remotely sensed data research into new methods of classification has continued (Foody & Mather, 2004). Recently soft classification algorithms such as artificial neural network, k nearest neighbour, and Gaussian Mixture Model (GMM) have become part of the mainstream classification algorithms. The application of GMM to remote sensing image classification problems is uncommon. The objective of this research therefore is to illustrate how the GMM algorithm can be applied to solving multi-class problems in remote sensing image classification. GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMM is commonly used as a parametric model of the probability distribution of continuous measurements.

### Gaussian mixture model

GMM presumes that the patterns  $x_i$  originate from a probability density  $p(x)$  (Bishop, 1995). This density is a linear combination of Gaussian functions  $p(x | j)$ ,

$$p(x | j) = \frac{1}{N_j} \exp\left(-\frac{1}{2}(x - c_j)^T A_j (x - c_j)\right). \quad (1)$$

The normalisation constant  $N_j$  is selected such that the integral of  $p(x | j)$  equals one (a necessary condition for a probability density). The negative exponent is a weighted squared distance (called Mahalanobis distance) between  $x$  and the centre  $c_j$ ; the corresponding weights are given by the symmetric matrix  $A_j$ . The boundary that has a Mahalanobis distance to the centre  $c_j$  equal to one is a hyper-ellipsoid. The density  $p(x)$  is a weighted sum of the local densities  $p(x | j)$  (Bishop, 1995),

$$p(x) = \sum_{j=1}^m P(j) p(x | j). \quad (2)$$

To normalise  $p(x)$ , the weights  $P(j)$  must sum to one,  $\sum_{j=1}^m P(j) = 1$ . Therefore,  $P(j)$  can be interpreted as the probability that patterns originate from the unit  $j$ . It is called "prior probability." The goal of the mixture model is to find the unknown parameters  $c_j$ ,  $A_j$ , and the priors  $P(j)$  for each unit  $j$  such that the likelihood,  $L = \prod_{i=1}^n p(x_i)$ , to obtain the distribution  $\{x_i\}$  given the density  $p(x)$  is maximal (Bishop, 1995). To solve this optimization problem it is common to use a variant of the expectation maximization algorithm (Bishop, 1995). It consists of two steps which iterate until convergence is reached. In the expectation step, the soft-assignment  $P(j | x_i)$  for all  $j$  and  $i$  is computed based on a given estimate of the parameters  $c_j$ ,  $A_j$ , and  $P(j)$ ;  $P(j | x_i)$  is called "posterior probability." It is computed using Bayes' theorem<sup>1</sup>,

$$P(j | x_i) = \frac{p(x_i | j)P(j)}{p(x_i)}. \quad (3)$$

In the special case of uniform Gaussians that all have the same width and weight  $P(j)$ , is the same as the Gibbs distribution. In the maximization step, the Gaussian's parameters  $c_j$ ,  $A_j$ , and  $P(j)$  that maximize the likelihood given all  $P(j | x_i)$  can be directly computed (Bishop, 1995). The result is that the centre  $c_j$  is the weighted mean of the set  $\{x_i\}$ ,

given, and  $P(j|x)$  is the probability of  $j$  under the condition that  $x$  is given. Reorganizing equation b1 about  $P(j|x)$  yields the Baye's theorem: 
$$P(j|x) = \frac{p(x|j)P(j)}{p(x)}$$

$$C_j = \frac{\sum_{i=1}^n P(j|x_i)x_i}{\sum_{i=1}^n P(j|x_i)} \quad (4)$$

and the matrix  $A_j$  is the inverse of the weighted covariance matrix  $C_j$ ,

$$C_j = \frac{\sum_{i=1}^n P(j|x_i)x_i}{\sum_{i=1}^n P(j|x_i)}. \quad (5)$$

The inverse can be computed by extracting all eigenvectors of  $C_j$ . Thus, the axes of the mentioned hyper-ellipsoid are the principal components of the local data distribution. The size of this hyper-ellipsoid is given by the eigenvalues  $\lambda_j^l$  from the Principle Component Analysis (PCA) (the semi-axis length of unit  $j$  in the direction  $l$  equals  $\sqrt{\lambda_j^l}$ ). Finally, the result for the prior probabilities is (Bishop, 1995),

$$P(j) = \frac{1}{n} \sum_{i=1}^n P(j|x_i). \quad (6)$$

It can be shown that alternating these expectation and maximization steps increases the likelihood  $L$  in each iteration step (Bishop, 1995). However, local maxima are not avoided. Further, single isolated data points (outliers) can make the algorithm unstable (Archambeau et al., 2003). If just one pattern is assigned to a unit (that is, the other patterns have almost zero  $P(j|x_i)$ ) the variance of the local Gaussian collapses to zero.

### Simulated modelling

Given a simulated ground truth data (Table 1) with a matrix size of 5 x 5, and equivalent simulated satellite remote sensing multi-spectral data that consist of three spectral bands (see Tables 2, 3, and 4), we intend to the classify the satellite data given in Tables 2, 3, & 4 into three classes: water, undeveloped, and developed. Our objective here is to use the satellite spectral bands given in Tables 2, 3, & 4 to derive the ground truth data given in Table 1. All the three spectral bands in Tables 2, 3, and 4 contain hypothetical DN<sup>1</sup> values.

**Table 1:** Ground truth data (water=1, undeveloped cells=2, and developed cells=3)

1	1	1	1	2
1	1	1	2	2
3	3	1	2	2
3	3	2	2	2

**Table 2 Band 1**

1	0	4	2	26
8	10	9	27	20
42	40	7	26	24
47	43	22	29	30
46	45	50	23	25

**Table 3 Band 2**

78	73	72	74	103
75	70	80	104	101
180	190	76	106	108
186	182	100	109	107
188	184	183	105	110

**Table 4 Band 3**

30	36	34	37	66
33	38	31	67	63
90	93	39	68	62
97	96	60	65	61
92	98	99	66	64

To classify the satellite data given in Tables 2, 3, & 4, a training set has to be randomly selected. The training data (six pixels) consist of elements from the three classes (see Table 5).

**Table 5: Training data**

Water	Band 1 (1,2) = 0	Band 2 (1,2) = 73	Band 3 (1,2) = 36
Water	Band 1 (2,1) = 8	Band 2 (2,1) = 75	Band 3 (2,1) = 33
Undeveloped	Band 1 (1,5) = 26	Band 2 (1,5) = 103	Band 3 (1,5) = 66
Undeveloped	Band 1 (2,4) = 27	Band 2 (2,4) = 104	Band 3 (2,4) = 67
Developed	Band 1 (3,1) = 42	Band 2 (3,1) = 180	Band 3 (3,1) = 90
Developed	Band 1 (4,2) = 43	Band 2 (4,2) = 182	Band 3 (4,2) = 96

For modelling convenience let the remaining nineteen cells that were not used for training the classifier (see Table 6) represent the test set. Conventionally the size of the test set is usually

smaller than that of the training set in machine learning. But for the purpose of illustration, let the remaining nineteen cells serve as the test set.

Table 6: Test data

Band 1 (1,1) = 1	Band 2 (1,1) = 78	Band 3 (1,1) = 30
Band 1 (4,1) = 47	Band 2 (4,1) = 186	Band 3 (4,1) = 97
Band 1 (5,1) = 46	Band 2 (5,1) = 188	Band 3 (5,1) = 92
Band 1 (2,2) = 10	Band 2 (2,2) = 70	Band 3 (2,2) = 38
Band 1 (3,2) = 40	Band 2 (3,2) = 190	Band 3 (3,2) = 93
Band 1 (5,2) = 45	Band 2 (5,2) = 184	Band 3 (5,2) = 98
Band 1 (1,3) = 4	Band 2 (1,3) = 72	Band 3 (1,3) = 34
Band 1 (2,3) = 9	Band 2 (2,3) = 80	Band 3 (2,3) = 31
Band 1 (3,3) = 7	Band 2 (3,3) = 76	Band 3 (3,3) = 39
Band 1 (4,3) = 22	Band 2 (4,3) = 100	Band 3 (4,3) = 60
Band 1 (5,3) = 50	Band 2 (5,3) = 183	Band 3 (5,3) = 99
Band 1 (1,4) = 2	Band 2 (1,4) = 74	Band 3 (1,4) = 37
Band 1 (3,4) = 26	Band 2 (3,4) = 106	Band 3 (3,4) = 68
Band 1 (4,4) = 29	Band 2 (4,4) = 109	Band 3 (4,4) = 65
Band 1 (5,4) = 23	Band 2 (5,4) = 105	Band 3 (5,4) = 66
Band 1 (2,5) = 20	Band 2 (2,5) = 101	Band 3 (2,5) = 63
Band 1 (3,5) = 24	Band 2 (3,5) = 108	Band 3 (3,5) = 62
Band 1 (4,5) = 30	Band 2 (4,5) = 107	Band 3 (4,5) = 61
Band 1 (5,5) = 25	Band 2 (5,5) = 110	Band 3 (5,5) = 64

The computer program used to model the GMM algorithm was implemented in MATLAB. To train the GMM classifier the means of each class was computed using the training data given in Table 5. The *variance* of each class was also computed. An *a priori* probability for each class was chosen such that the *a priori* probability fulfils:  $0 < a \text{ priori} < 1$ ; 0.5 was chosen for each

class. Then *a posteriori* probabilities were computed for each test point (see Table 8). For each pixel, based on the maximum likelihood principle, the class with the highest *a posteriori* probability was assigned to that pixel. The model also computes the mixing proportion for each class (see Table 7). The training and test results are display in Tables 7 and 8 respectively.

**Table 7: Training result**


---

Gaussian mixture distribution with 3 components in 3 dimensions

Component 1: Water

Mixing proportion: 0.333333

Mean: 4.0000 74.0000 34.5000

Component 2: Undeveloped

Mixing proportion: 0.333333

Mean: 27.5000 103.5000 66.5000

Component 3: Developed

Mixing proportion: 0.333333

Mean: 42.5000 181.0000 93.0000

Variances of components 1, 2, & 3 respectively

1.0e+003 \* [0.3079 2.4443 0.6911]

---

**Table 8: Test result (water =1, undeveloped =2, and developed =3)**

Cell	Posterior probability			Class	Remark
	Water	Undeveloped	Developed		
1	0.9004	0.0993	0.0004	1	Correct
2	0.0002	0.0670	0.9328	3	„
3	0.0003	0.0790	0.9207	3	„
4	0.7752	0.2234	0.0013	1	„
5	0.0006	0.0939	0.9054	3	„
6	0.0003	0.0751	0.9246	3	„
7	0.8652	0.1343	0.0006	1	„
8	0.8202	0.1787	0.0011	1	„
9	0.7938	0.2049	0.0013	1	„
10	0.2468	0.7071	0.0461	2	„
11	0.0001	0.0595	0.9404	3	„
12	0.8639	0.1355	0.0006	1	„
13	0.1275	0.772	0.1005	2	„
14	0.1125	0.7732	0.1142	2	„
15	0.1732	0.751	0.0758	2	„
16	0.2465	0.7052	0.0483	2	„
17	0.1839	0.7419	0.0742	2	„
18	0.1288	0.7745	0.0967	2	„
19	0.1542	0.7546	0.0911	2	„

**Table 9: Confusion matrix for GMM classification**

	Reference data			
	Water	Undeveloped	Developed	Unclassified
Predicted data				
Water	8	0	0	0
Undeveloped	0	10	0	0
Developed	0	0	7	0
Unclassified	0	0	0	0

It can be discerned from Table 8 that all the test points were correctly classified by the GMM algorithm. Using the confusion matrix visualisation (Table 9), the overall accuracy can be computed as the sum of the diagonal elements in the matrix divided by all the elements in the matrix. Therefore, overall accuracy =  $25/25 = 100\%$ .

### Empirical modelling

A multispectral Landsat 7 ETM image of Porirua, New Zealand, acquired in 2006 was used for the experiment (see Figure 3). The Landsat image consists of seven spectral bands, and has a cell size of 25m x 25m. The original satellite data were first reviewed in GIS (ArcGIS software); and all seven bands were extracted using the *layer properties tool* and visualised in MATLAB (see Figure 3). Before importing the data into MATLAB, they were first converted from raster to ASCII data using the *ArcGIS conversion tool*. MATLAB cannot read raster files; hence the data must be in ASCII format for onward processing in MATLAB. In MATLAB the final study area was extracted from the original satellite image. Some regions of the satellite image were affected by cloud, which was why the final study area did not include those regions affected by cloud. All the seven bands were used for the classification experiment. The stratified random sampling was used to select the training data. The classified image was visualised in the GIS.

The results of the GMM experiment are given in Figure 1 and Table 10. The confusion matrix given in Table 10 was computed by comparing the result of the GMM classification and the reference data given in Figure 2. Using the confusion matrix given in Table 10, the computed overall accuracy was 85.35%.

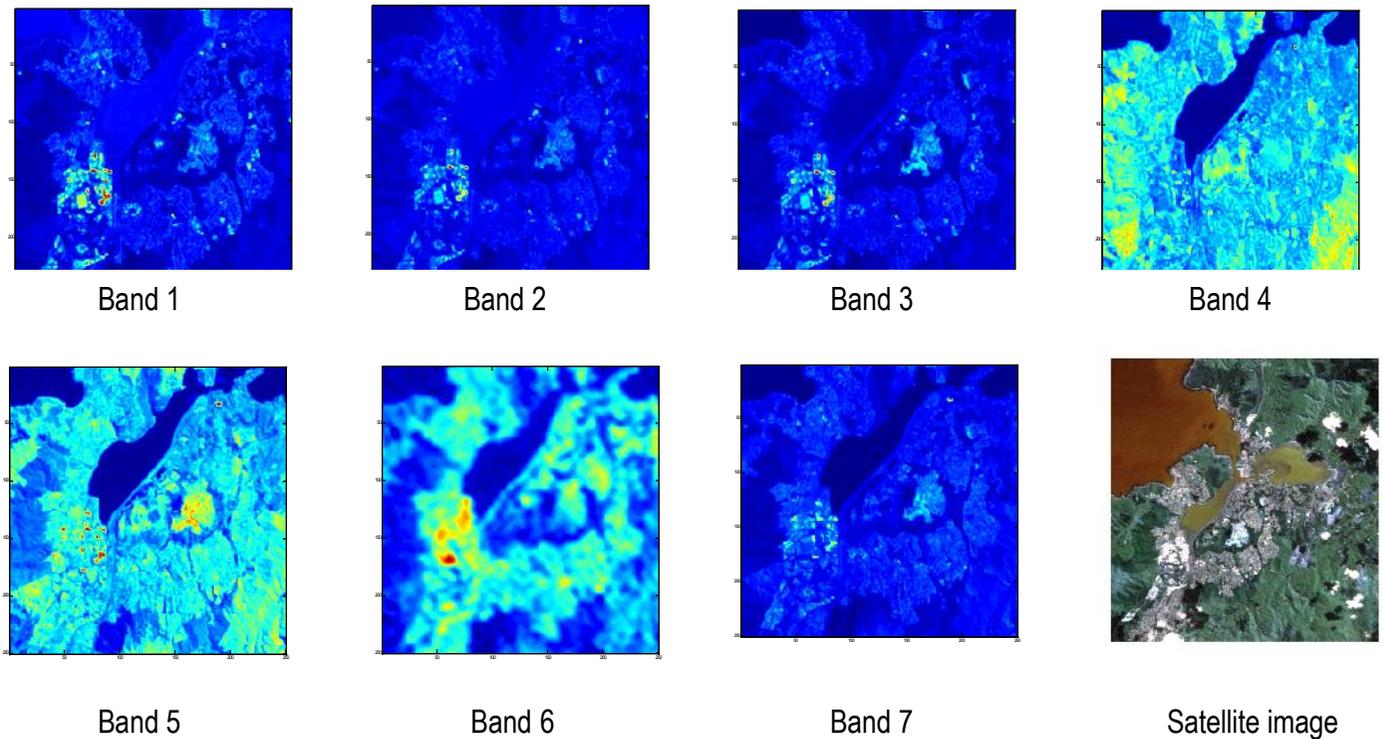


Figure 1 Extracted bands 1 - 7 of Landsat image of Porirua and original Landsat image of Porirua, New Zealand

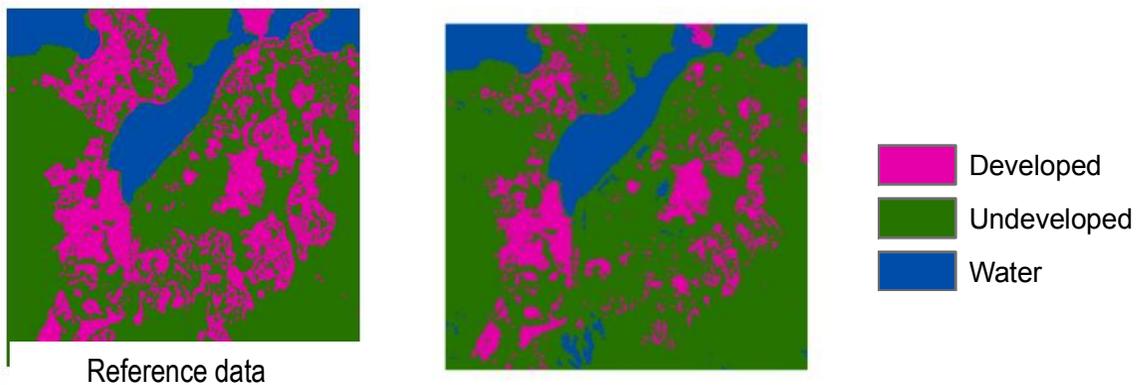


Table 10: Confusion matrix for GMM

	Reference data			
	Developed	Undeveloped	Water	Unclassified
Predicted data				
Developed	9411	46	0	0
Undeveloped	7686	37117	0	0
Water	190	1233	6817	0
Unclassified	0	0	0	0

## CONCLUSION

This study illustrated basically how the GMM algorithm can be applied to the classification of satellite remote sensing data. The essence of first illustrating the experiment using simulated data was to help explain how the empirical experiment was implemented. In the simulated modelling, from Table 9, no water, undeveloped, and developed cells was wrongly predicted. In the empirical modelling, from Table 10, 46 undeveloped cells were wrongly predicted as developed; 7686 developed cells were wrongly predicted as undeveloped; 190 and 1233 developed and undeveloped cells were respectively wrongly predicted as water; and no cell was left unclassified. . From Tables 9 and 10, note that, the diagonal elements are the correctly classified pixels. The high computed overall accuracy espouses previous findings that adjudge the Gaussian mixture model as one of the most robust parametric classifiers.

## REFERENCES

- Archambeau, C., Lee, J. A., and Verleysen, M., 2003. On convergence problems of the EM algorithm for finite gaussian mixtures. In Verleysen, M., (Ed.), *Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2003)* (99-106), Belgium: D-side.
- Bishop, C. M., 1995. *Neural Networks for Pattern Recognition*. Oxford University Press, UK.
- Foody, M. G. and Mather, A., 2004. Toward Intelligent Training of Supervised Image
- Li, T. S., Chen, C. Y., and Su, C. T., 2003. Comparison of neural and statistical algorithms for supervised classification of multi-dimensional data. *Int. J. Indus. Eng.—Theory Appl. Pract.*, 10, 73–81.
- Mahesh P. and Mather, P. M., 2003. An assessment of the effectiveness of decision tree methods for land cover classification. *Remote Sens. Environ.*, vol. 86, 554–565.
- Mather, P., 1987. *Computer Processing of Remotely Sensed Images – An Introduction*. New York: John Wiley and sons.
- Peddle D.R., Foody, G. M., Zhang, A., Franklin, S. E., and LeDrew, E. F., 1994. Multisource image classification II: An empirical comparison of evidential reasoning and neural network approaches. *Canadian Journal of Remote Sensing*, 20, 396 – 407.
- Rogan, J., Franklin, J., and Roberts, D. A., 2002. A comparison of methods for monitoring multitemporal vegetation change using Thematic Mapper imagery. *Remote Sens. Environ.*, 80, 143–156.