Clustering Algorithms: Course Project

Vasilis Margonis
ALMA - A0005

January 20, 2019

Abstract

Every algorithm is implemented from scratch which efficiency always in mind (vectorization when possible, avoid computation of inverse matrices, etc). The following files are contained along with the project materials:

`k_means.m`	implementation of k-means.
`fuzzy.m`	implementation of fuzzy.
`possibilistic_1.m`	implementation of possibilistic (GPAS1).
`possibilistic_2.m`	implementation of possibilistic (GPAS2).
`exp_max.m`	implementation of Expectation-Maximization.
`accuracy.m`	computation of total and per-cluster accuracy.
`k_means_salinas.m`	script for k-means.
`fuzzy_salinas.m`	script for fuzzy.
`possibilistic_salinas.m`	script for possibilistic.
`probabilistic_salinas.m`	script for EM.
`plots.m`	several plots of the data set.

1 Introduction

Data preprocess: First of all, we discuss how we preprocess the dataset X, before feeding it to the algorithms:

- We apply dimension reduction to the matrix X with PCA via the "`pca_fun.m`" function, using only *the first 7 principal components*. We do so for two reasons:

  1. The first 7 principal components explain more than 99.99% of the total variance in the data, so we lose almost no information. For example, we tried k-means with and without PCA (with 7 PC's), and the difference in accuracy was ≈ 0.0001.
  2. The algorithms will run faster.

- After the application of PCA, we get a matrix Y of dimension $7 \times N$, where N is the number of points. Then, we apply a *uniform scale* to Y:

$$Y \leftarrow \frac{Y}{\text{mean}(|Y|)}$$

  In this way, we reduce the range of values of Y while preserving the relative distances of the points. We want the range reduced so that the distances do not become very large.

$$\begin{aligned} \text{Range of } Y \text{ before scaling:} &\quad [-2.466 * 10^4, 3.2407 * 10^4] \\ \text{Range of } Y \text{ after scaling:} &\quad [-1.4195, 1.8653] \end{aligned}$$

Cluster Initialization: For all algorithms, we initialize the representatives with the max-min method via the "`most_dist_repre.m`" function. The initialization is fixed for all algorithms, for robust comparison, and is depicted in figure 1.

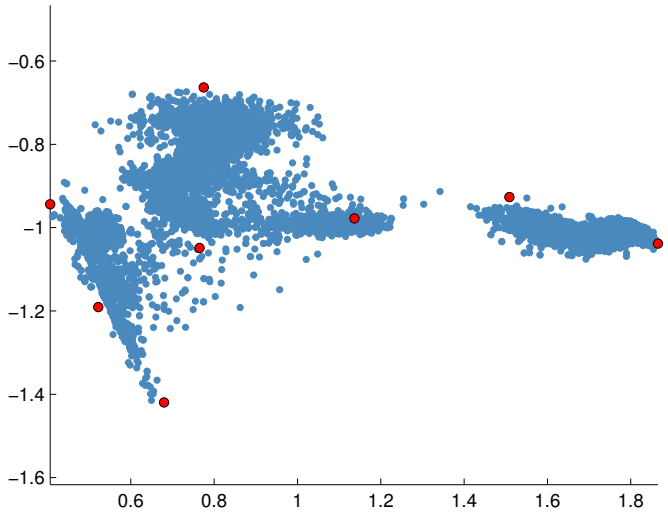
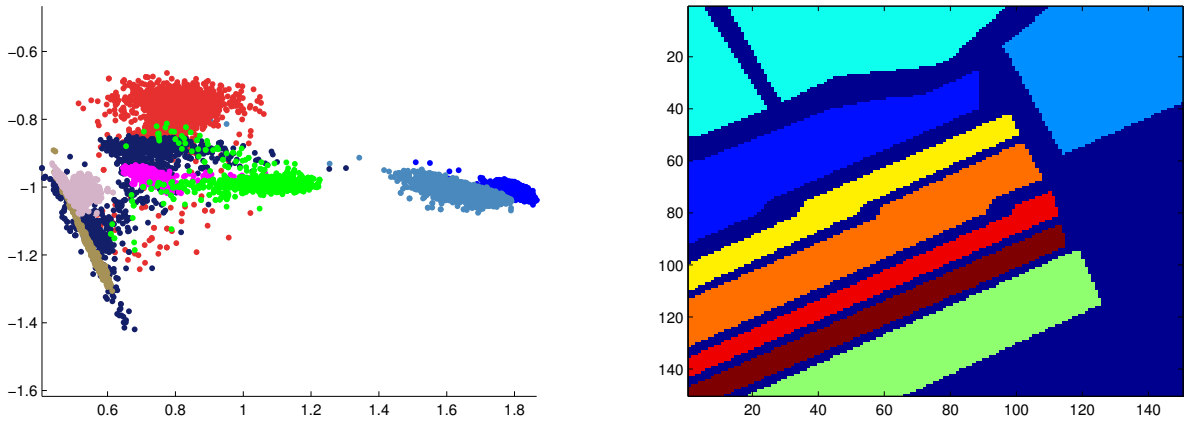


Figure 1: Cluster initialization with the max-min method (first two PC's).

2 Applying the Algorithms - Qualitative verification

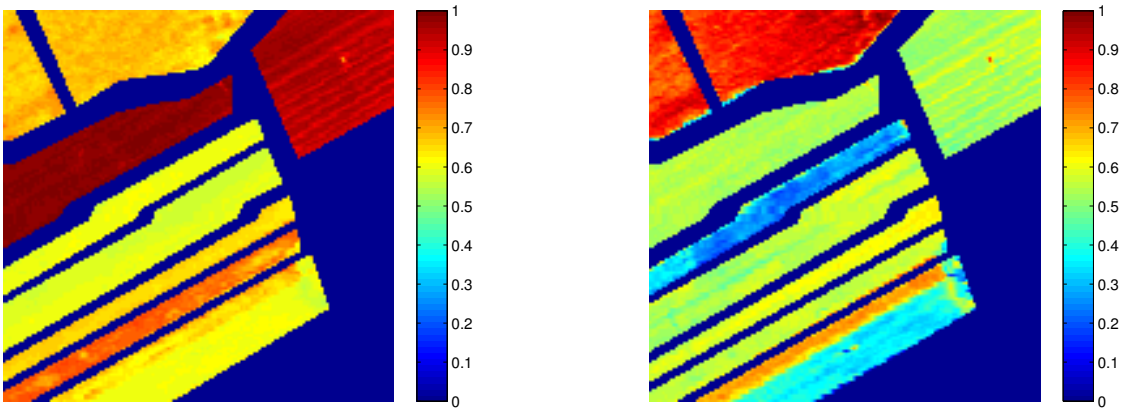
Below (figure 2a), we see the physical clusters projected to the first two principal components. This is a very difficult data set for any point-representative clustering algorithm, as the physical clusters vary in compactness, shape and volume.



(a) Plot of physical clusters on the first two PC's

(b) Labeling map.

Figure 2: The data set.



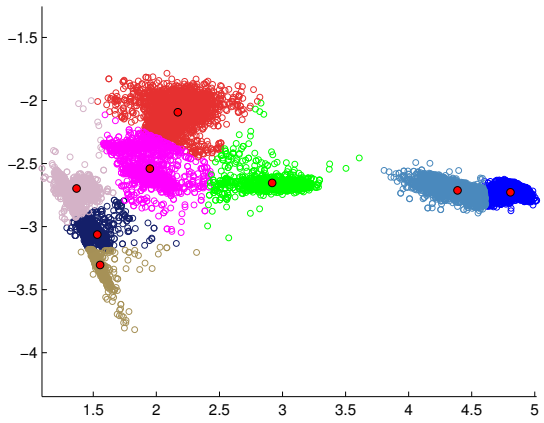
(a) 1st PC

(b) 2nd PC

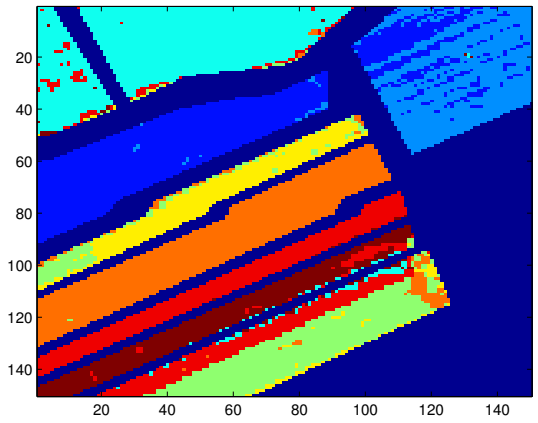
Figure 3: First 2 PC's of the bands

2.1 k-means

We start with the k-means algorithm. The parameter in this case is only the number of clusters, which we fix a priori to $m = 8$, since we know how many clusters we are looking for. We initialized the representatives as in figure 1. We tried random initialization of the representatives that resulted in different clusterings, which where inferior (both qualitatively and quantitatively) to the clustering obtained with max-mix initialization, so we omit those results. We faced no problems regarding the execution of the algorithm.



(a) first two PC's

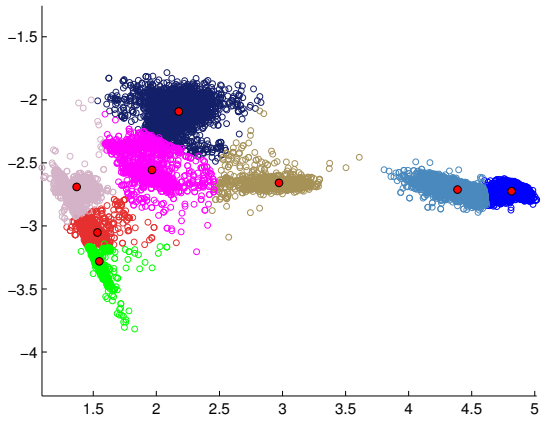


(b) Clustering map

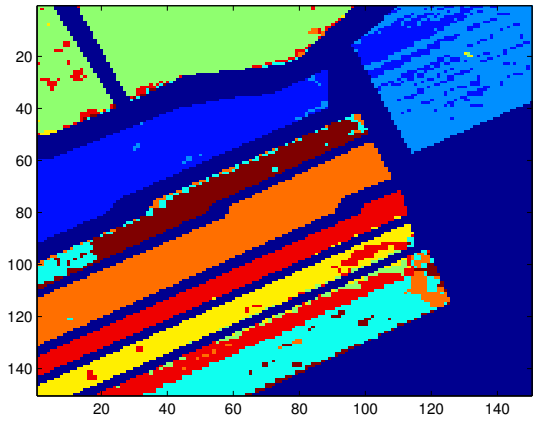
Figure 4: **k-means**. Comparing the labeling map with the clustering map of k-means, we see that the algorithm did a pretty good job. The light blue, yellow, and green clusters are split though, but this was expected if we consider the structure of the 2nd PC of the bands (fig 3b).

2.2 Fuzzy c-means

Now we apply the fuzzy-c-means algorithm. Again, we fix $m = 8$, so the only free parameter is q. We tried for several values of q in the range $[2, 10]$, and we got the most interesting results for $q = 2, 4$, which are displayed in figures 5 and 6. We initialize the representatives with max-min. Again, random initialization of the representatives resulted in inferior clusterings and we omit them. We faced no problems regarding the execution of the algorithm.

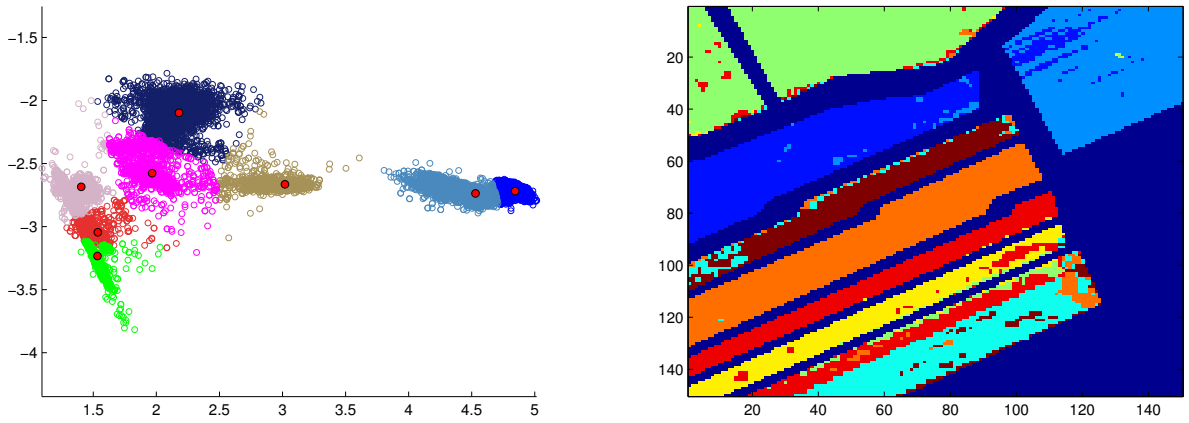


(a) first two PC's



(b) Clustering map

Figure 5: **Fuzzy-c-means (q=2)**. In qualitative terms, the clustering obtained by fuzzy for $q = 2$ is the same with that of k-means, and this can be seen by both their respective clustering maps and 2D plots. This was expected, as for small values of q the fuzzy-c-means and k-means behave similarly. However, the clustering of fuzzy achieves a slightly better accuracy as we will see later.



(a) first two PC's

(b) Clustering map

Figure 6: **Fuzzy-c-means (q=4)**. For $q = 4$, we get the same clustering as well, with the only difference being that the algorithm managed to capture more of the light blue cluster without compromising another cluster. The rest of the clustering still follows the same pattern with the 2nd PC of the bands.

2.3 Possibilistic c-means

For the Possibilistic algorithm, we used the scheme which minimizes:

$$J_q(U, \Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} \left(u_{ij} \cdot \|x_i - \theta_j\|_2^2 \right) + \sum_{j=1}^{m} \eta_j \sum_{i=1}^{N} (1 - u_{ij})^q. \qquad \text{(GPAS1)}$$

We initialized θ's and η_j's with the following procedure:

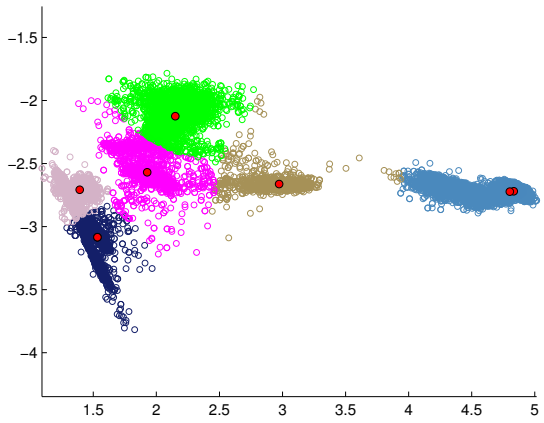
1. Initialize the representatives with max-min.
2. for some $q > 1$, run fuzzy-c-means with the θ_j's from step 1.
3. Let U be the matrix returned by fuzzy-c-means. Compute η_j's as

$$\eta_j = \frac{\sum_{i=1}^{N} \left(u_{ij}^q \cdot \|x_i - \theta_j\|_2^2 \right)}{\sum_{i=1}^{N} u_{ij}^q}$$

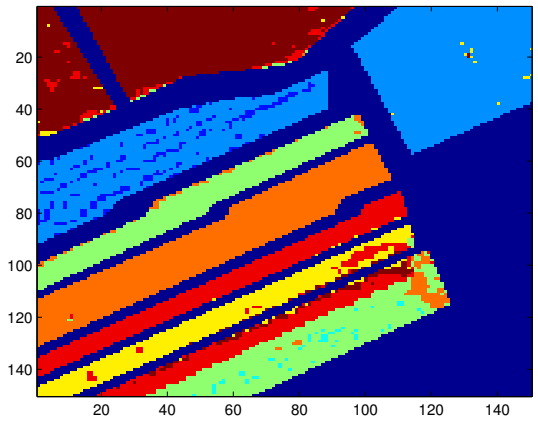
4. Run Possibilistic algorithm with η's from step 3.

Parameters: Now, the parameters are m and q. The possibilistic scheme is known for converging to physical clusters independently of the choice of m. However, larger values of m (e.g. $9, 10, 15$) did not produce better results, as the extra representatives converged on the initial 8, so we stick with $m = 8$. For q, we tried for many values and we got the best results for $q = 2$ and $q = 3$ (figures 7, 8).

Problems: The main problem with the possibilistic algorithm is that some representatives tended to converge to the same point, even though we initialized them via the max-min method, and this happened independently of the choice of q. This can be seen in figures 7a and 8a, where the 8 representatives converged to only 6 and 5 points, respectively.

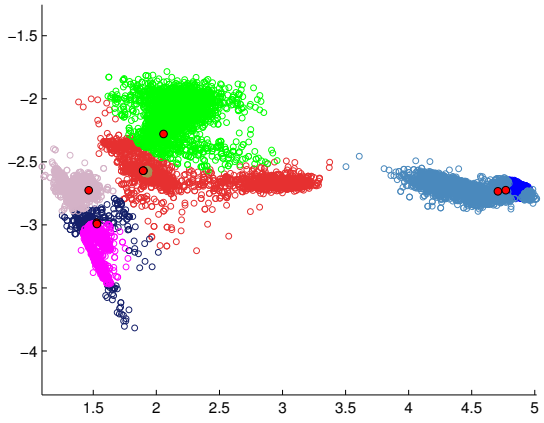


(a) first two PC's

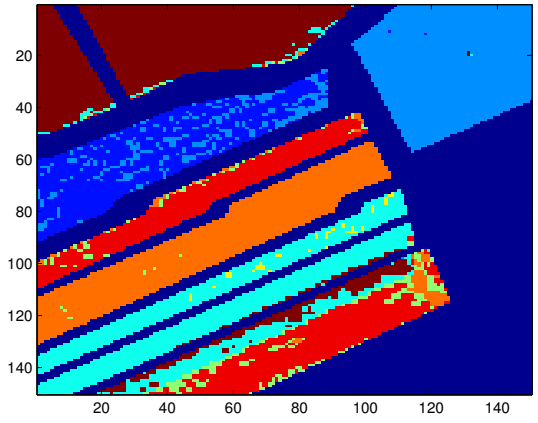


(b) Clustering map

Figure 7: **Possibilistic (q=2)**: For $q = 2$ we observe a clustering very similar to the pattern of the 1st PC of the bands. However, if we consider the labeling and the clustering maps (figs 7b, 3b), the clustering is very poor: Two physical clusters were absorbed by others.



(a) first two PC's



(b) Clustering map

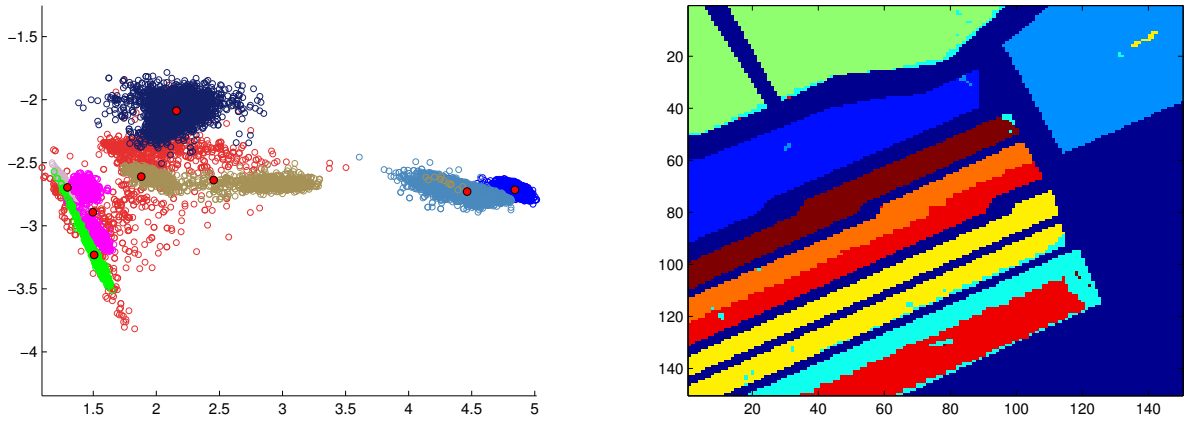
Figure 8: **Possibilistic (q=3)**: The results now are even weaker than before. The clustering do not seem to follow any pattern regarding the first two PC's of the bands, and its also poor in terms of the final clustering map.

2.4 Expectation Maximization

For the probabilistic algorithm with Gaussian mixtures, we fix $m = 8$ and initialize the values as:

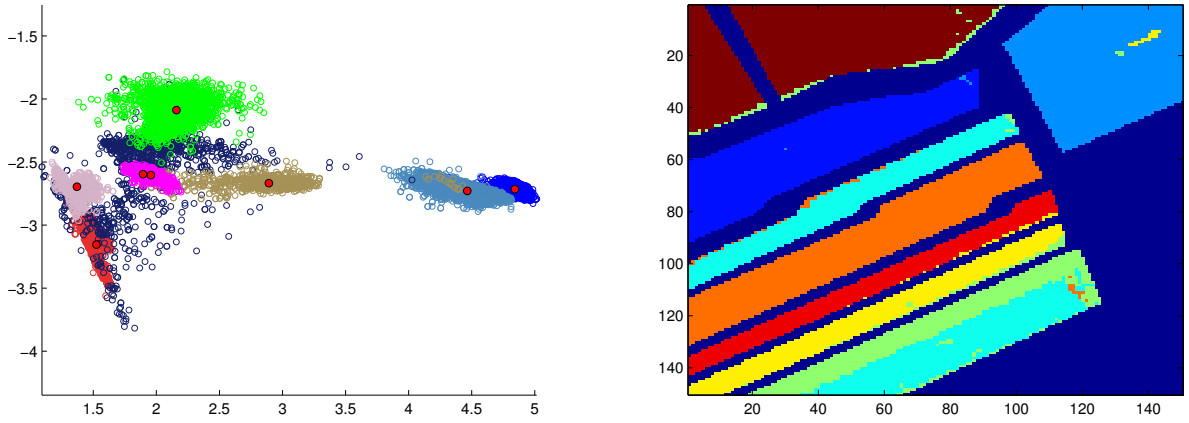
- θ_j (means), with max-min method.
- S_j (covariance matrices), with $\sigma^2 I$ for some σ user-defined, for every j.
- P_j (prior probabilities), equiprobable: $P_j = 1/m$, for every j.

Parameters: The only parameter left is σ. We tried for several values and got the better results for $\sigma = 1, 2$ (figures 9, 10).



(a) first two PC's (b) Clustering map

Figure 9: **Probabilistic, σ=1**: The EM algorithm produced the "cleaner" results, as can be seen from both clustering maps. For $\sigma = 1$, two clusters are split in the exact same way as the are split in the 2nd PC of the bands (orange-red and cyan-red). However, two other clusters are merged into one (yellow). The same two clusters were also merged by the possibilistic algorithm for $q = 3$.



(a) first two PC's (b) Clustering map

Figure 10: **Probabilistic, σ=2**: For $\sigma = 2$ we get outstanding results. All the clusters are almost perfectly identified except for the green cluster which is once again split as in figure 3b. This clustering is also the one closest to the 2D plot of the physical clusters (compare figures 10a and 2a).

3 Quantitative Verification

In this section, we analyze the clustering quantitatively, with the help of the function "`accuracy.m`". This function takes the true labels and the clustering labels of the points and computes the confusion matrix, as well as the accuracy of the clustering.

However, the clustering labels do not necessarily match the true labels. For example, an algorithm may have correctly identified a cluster for which it gave label "bel=4", when the true label for that cluster is "1". For that reason, we first compute a dummy confusion matrix, and then we try all possible permutations of its columns and keep the one that results in the best total accuracy.

In the table below, we gather all the accuracy results for all four algorithms.

Algorithm	Param.	Accuracy per cluster								Accuracy
		1	2	3	4	5	6	7	8	(total)
k-means	-	0.986	0.732	0.932	0.569	0.739	0.999	0.989	0.883	0.8468
Fuzzy	$q = 2$	0.985	0.741	0.938	0.561	0.755	0.999	0.990	0.871	0.8483
	$q = 4$	0.970	0.907	0.947	0.541	0.821	0.999	0.990	0.861	0.8760
Possibilistic	$q = 2$	0.141	0.993	0.940	0.570	0	0.997	0.990	0.871	0.7111
	$q = 3$	0.785	0.997	0.964	0.121	0.894	0.995	0.051	0.941	0.7523
EM	$\sigma = 1$	0.995	0.992	0.959	0.523	0.990	0.502	1	0.084	0.7798
	$\sigma = 2$	0.997	0.989	0.959	0.455	0.952	1	0.961	0.929	0.8957

As we can see from the table, the cluster 4 (that is, the lower most, green cluster of fig 2b) could not be fully identified by any algorithm, and in most cases was cut in half. This is reasonable if we consider the 1st and 2nd principal components of the bands (figures 3a, 3b).

4 Performance comparison

The qualitative and quantitative analysis of the results completely agree. The superior algorithm for this data set proved to be EM, with an accuracy of 0.8957 (for $\sigma = 2$), although it was the slowest. The second best clusterings were produced by fuzzy and k-means with some trade off: The fuzzy algorithm for $q = 4$ achieved higher accuracy with a slightly better-quality clustering but it was slower than k-means. Finally, the possibilistic algorithm had the worst performance in quality and accuracy.

Algorithm	Param.	Running time (sec.)
k-means	-	0.18
Fuzzy-c-means	$q = 2$	0.21
	$q = 4$	6.21
Possibilistic	$q = 2$	0.60
(+ computing of η's)	$q = 3$	1.56
EM	$\sigma = 1$	15.39
	$\sigma = 2$	8.75