Στατιστική Μοντελοποίηση και Αναγνώριση Προτύπων (ΤΗΛ311)

Αναφορά 3ης Σειράς Ασκήσεων

Ανδρεαδάκης Αντώνης 2013030059

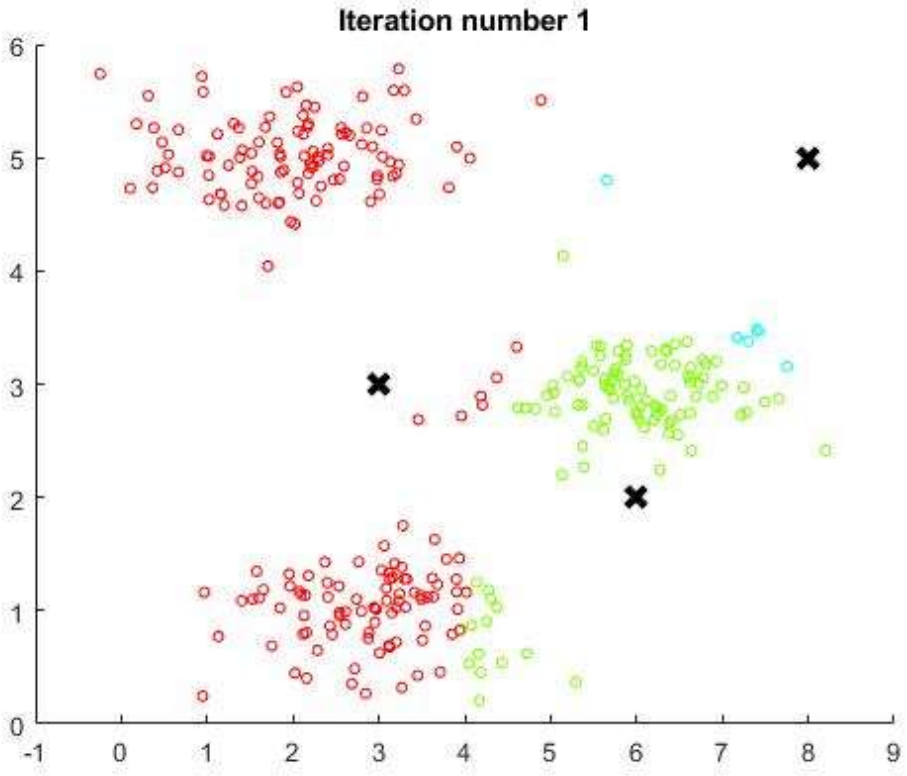
1. Στην άσκηση αυτή, υλοποιήθηκε ο αλγόριθμος K-NN για κατηγοριοποίηση κειμένου. Μια διαδικασία κατάταξης κειμένων φυσικής γλώσσας σε προκαθορισμένο αριθμό κατηγοριών. Η βάση δεδομένων που χρησιμοποιήθηκε είναι ένα υποσύνολο της WebKB και περιέχει ιστοσελίδες από 4 πανεπιστήμια της Αμερικής. Με δεδομένο ότι, αρχικά είχε γίνει μια επεξεργασία και ταξινόμηση των δεδομένων που μας δόθηκαν, υλοποιήσαμε μερικές βοηθητικές συναρτήσεις. Μια από αυτές είναι ο υπολογισμός της εντροπίας, ώστε να περιορίσουμε τις λέξεις αξιοποιώντας μόνο όσες είναι πιο σημαντικές. Επίσης, υλοποιήσαμε τη συνάρτηση που επιστρέφει το νέο λεξικό. Έπειτα από την εύρεση της εντροπίας, με τη μικρότερη ή μέγιστη τιμή (αυτό επηρεάζει μόνο το accuracy), με βάση αυτή την τιμή επιλέγουμε τις λέξεις που είναι πιο σημαντικές. Τέλος, ο αλγόριθμος k-nn με τον οποίο υπολογίζουμε την απόσταση στα test-train samples είτε με την ευκλείδεια είτε με την cosine similarity συνάρτηση και ταξινομούμε σε ένα πίνακα τις αποστάσεις αυτές. Επομένως, για την επίδοση του συστήματος «χωρίζουμε» τα δείγματα σε train-test, καλούμε την knn_classify η οποία μας επιστρέφει ένα πίνακα που περιέχει τιμές 0 έως περίπου 1 (ανάλογα την επιλογή για εύρεση απόστασης-Ευκλείδεια ή Cosine Similarity) και για τον υπολογισμό της επίδοσης σε κάθε φάκελο, ελέγχουμε αν τα labels της train_y αντιστοιχούν στην test_z, αθροίζουμε όλες τις αληθείς τιμές αυτές και διαιρούμε με το μέγεθος του πίνακα test_z (ή train_y μιας κι έχουν το ίδιο μέγεθος). Στο τέλος, για τον υπολογισμό της συνολικής επίδοσης, αθροίζουμε τις επιδόσεις από κάθε φάκελο και το άθροισμα το διαιρούμε με το πλήθος των φακέλων.
   Παρακάτω βλέπουμε τις επιδόσεις, για την ευκλείδεια και cosine similarity αντίστοιχα:

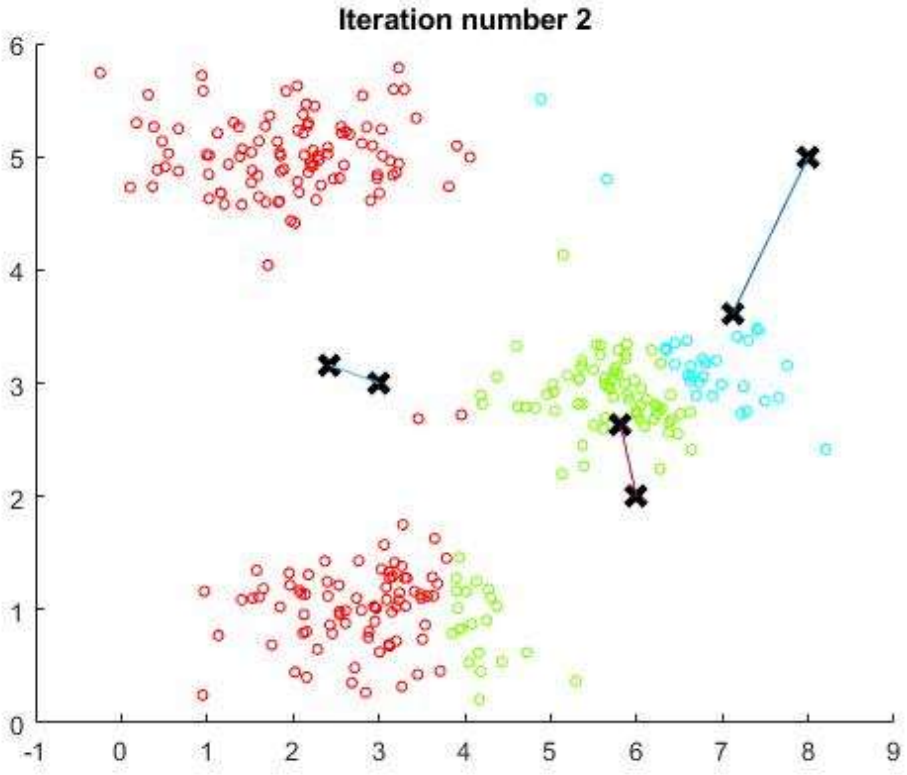
```
Number of words selected: 300
Distance Metric: norm2
Number of K nearest neighbors: 1
Fold: 1, Accuracy: 61.53%
Fold: 2, Accuracy: 63.27%
Fold: 3, Accuracy: 66.15%
Fold: 4, Accuracy: 53.04%
Fold: 5, Accuracy: 63.71%
K=1 -- Total Accuracy: 61.54%
Number of K nearest neighbors: 3
Fold: 1, Accuracy: 61.19%
Fold: 2, Accuracy: 56.67%
Fold: 3, Accuracy: 61.30%
Fold: 4, Accuracy: 55.40%
Fold: 5, Accuracy: 61.43%
K=3 -- Total Accuracy: 59.20%
Number of K nearest neighbors: 5
Fold: 1, Accuracy: 63.84%
Fold: 2, Accuracy: 61.20%
Fold: 3, Accuracy: 61.27%
Fold: 4, Accuracy: 63.26%
Fold: 5, Accuracy: 52.92%
K=5 -- Total Accuracy: 60.50%
Number of K nearest neighbors: 10
Fold: 1, Accuracy: 66.29%
Fold: 2, Accuracy: 63.12%
Fold: 3, Accuracy: 63.58%
Fold: 4, Accuracy: 61.49%
Fold: 5, Accuracy: 61.33%
K=10 -- Total Accuracy: 63.16%
```

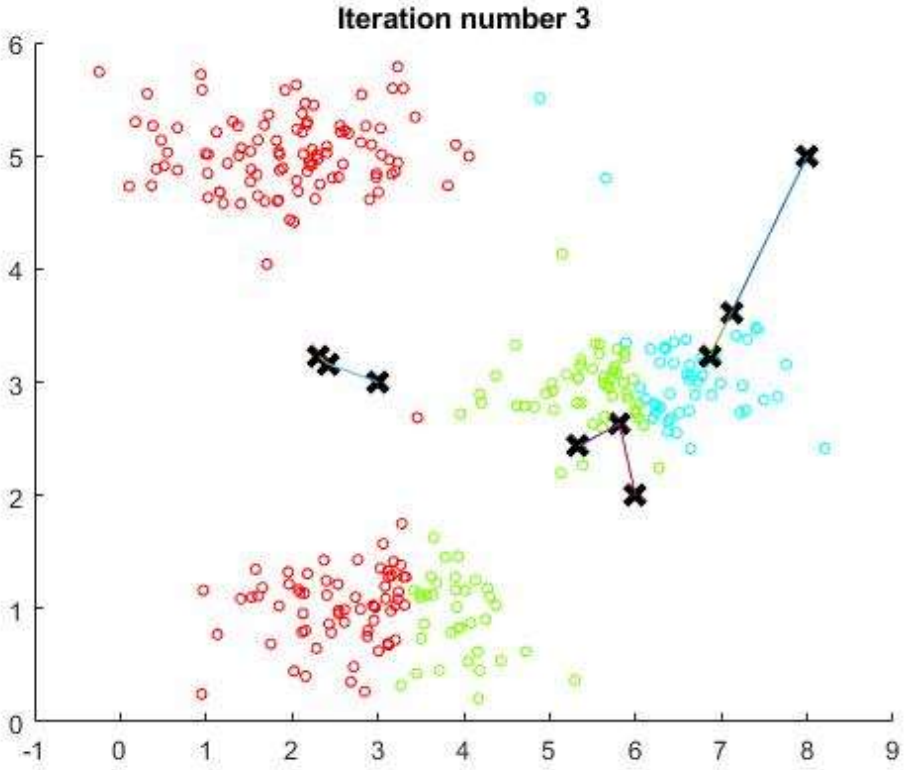
```
Number of words selected: 300
Distance Metric: Cosine Similarity
Number of K nearest neighbors: 1
Fold: 1, Accuracy: 42.87%
Fold: 2, Accuracy: 42.93%
Fold: 3, Accuracy: 43.42%
Fold: 4, Accuracy: 42.56%
Fold: 5, Accuracy: 43.36%
K=1 -- Total Accuracy: 43.03%
Number of K nearest neighbors: 3
Fold: 1, Accuracy: 42.95%
Fold: 2, Accuracy: 42.81%
Fold: 3, Accuracy: 43.32%
Fold: 4, Accuracy: 43.11%
Fold: 5, Accuracy: 43.48%
K=3 -- Total Accuracy: 43.13%
Number of K nearest neighbors: 5
Fold: 1, Accuracy: 42.59%
Fold: 2, Accuracy: 42.97%
Fold: 3, Accuracy: 43.70%
Fold: 4, Accuracy: 42.76%
Fold: 5, Accuracy: 43.36%
K=5 -- Total Accuracy: 43.08%
Number of K nearest neighbors: 10
Fold: 1, Accuracy: 43.19%
Fold: 2, Accuracy: 43.17%
Fold: 3, Accuracy: 43.44%
Fold: 4, Accuracy: 43.29%
Fold: 5, Accuracy: 42.81%
K=10 -- Total Accuracy: 43.18%
```

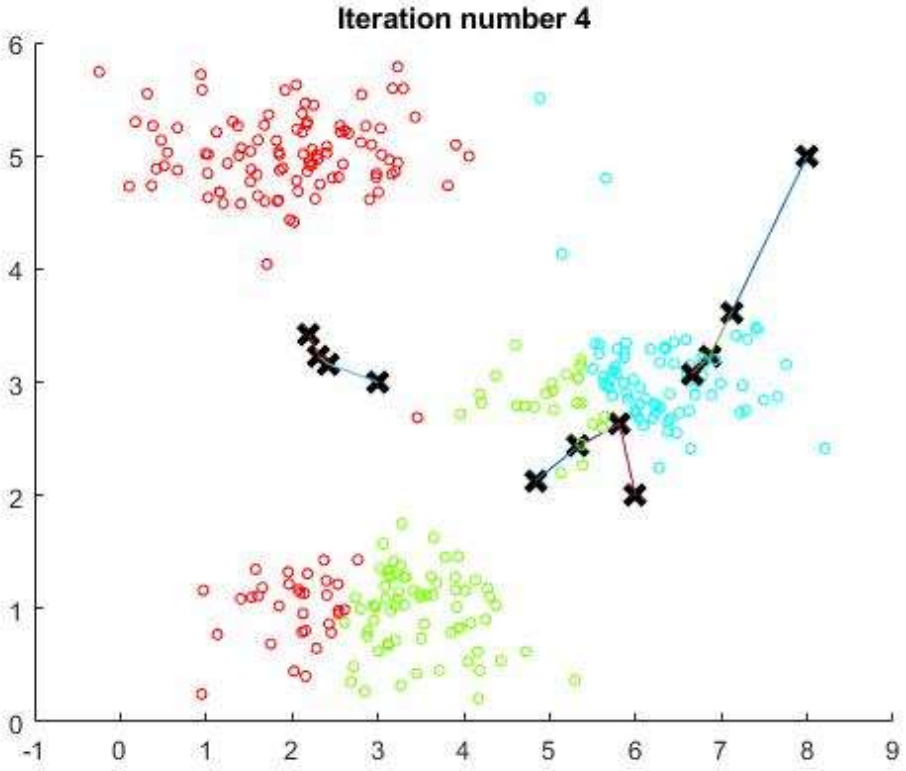
Παρατηρούμε ότι με την ευκλείδεια απόσταση, έχουμε μεγαλύτερη συνολική επίδοση (ενώ η επίδοση κάθε φακέλου διαφοροποιείται). Αντιθέτως, με την cosine similarity έχουμε μικρότερη συνολική επίδοση, αλλά σταθερή για κάθε φάκελο (πάρα πολυ μικρή απόκλιση). Η επίδοση δεν συμπίπτει με τα αποτελέσματα που δόθηκαν κατά τη διάρκεια των φροντιστηρίων, όμως αυτό είναι κάτι που δε θα πρέπει να μας ανησυχεί τόσο όσο το γεγονός ότι ο αλγόριθμος μας έχει υλοποιηθεί σωστά.

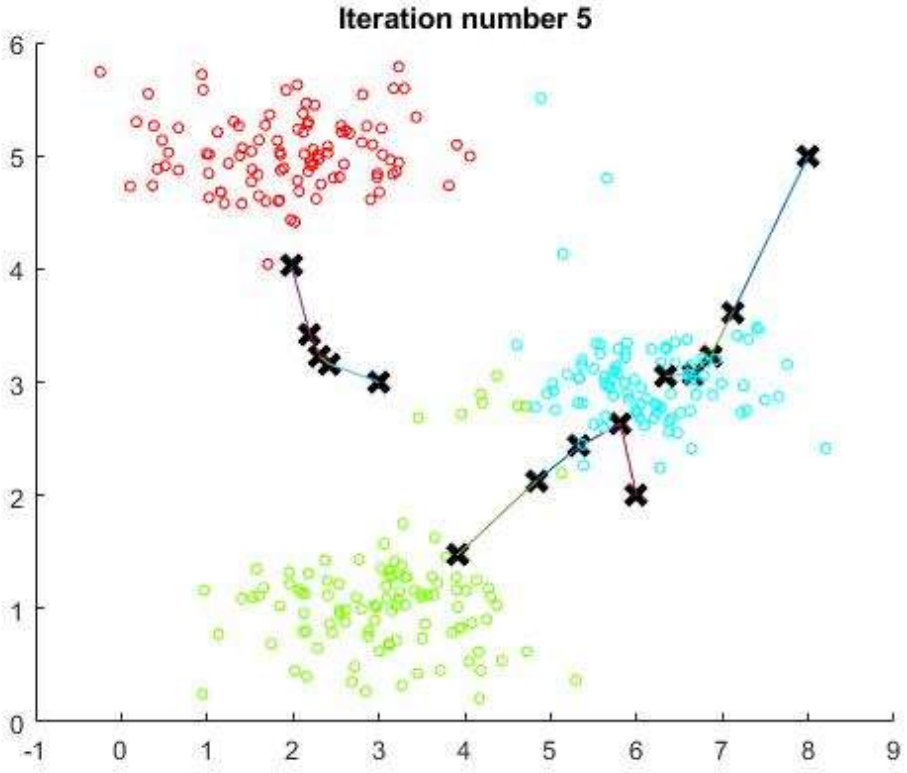
2. Εδώ υλοποιήσαμε τον αλγόριθμο k-means, ώστε να συμπιέσουμε μια δισδιάστατη εικόνα. Παρακάτω βλέπουμε γραφικά, πώς δουλεύει ο αλγόριθμος βήμα – βήμα και στη συνέχεια τις συμπιεσμένες εικόνες για κάθε διαφορετικό K.

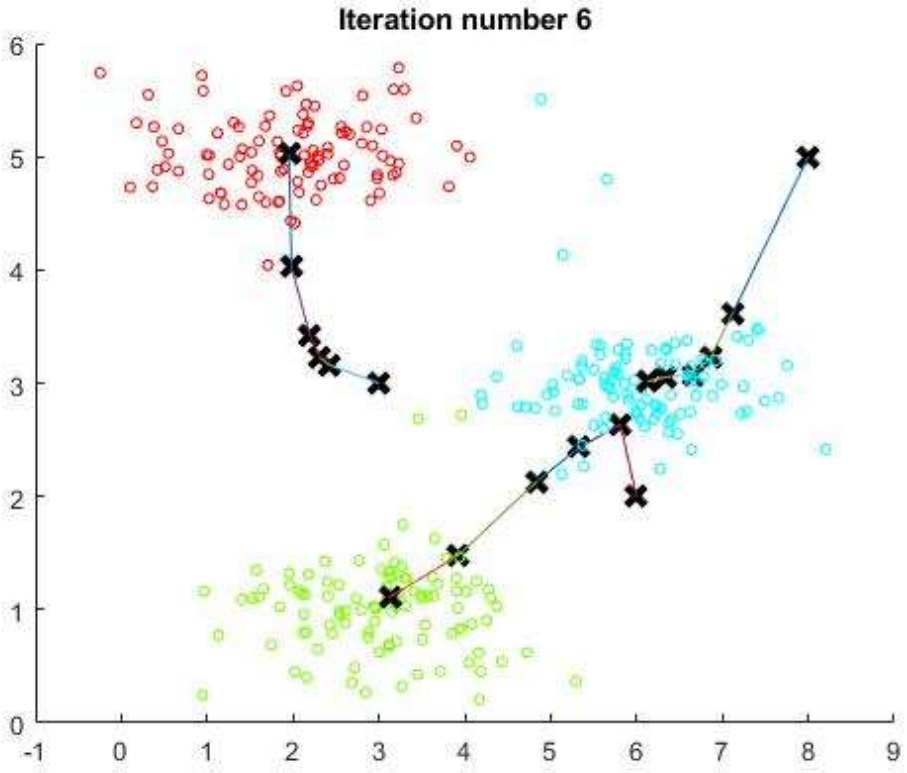


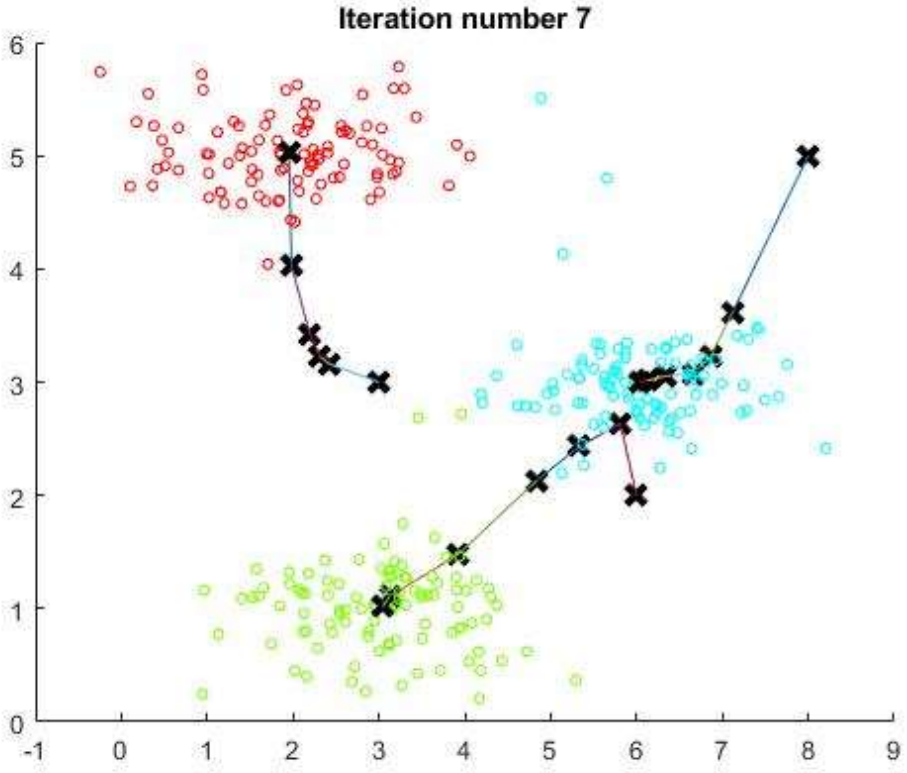


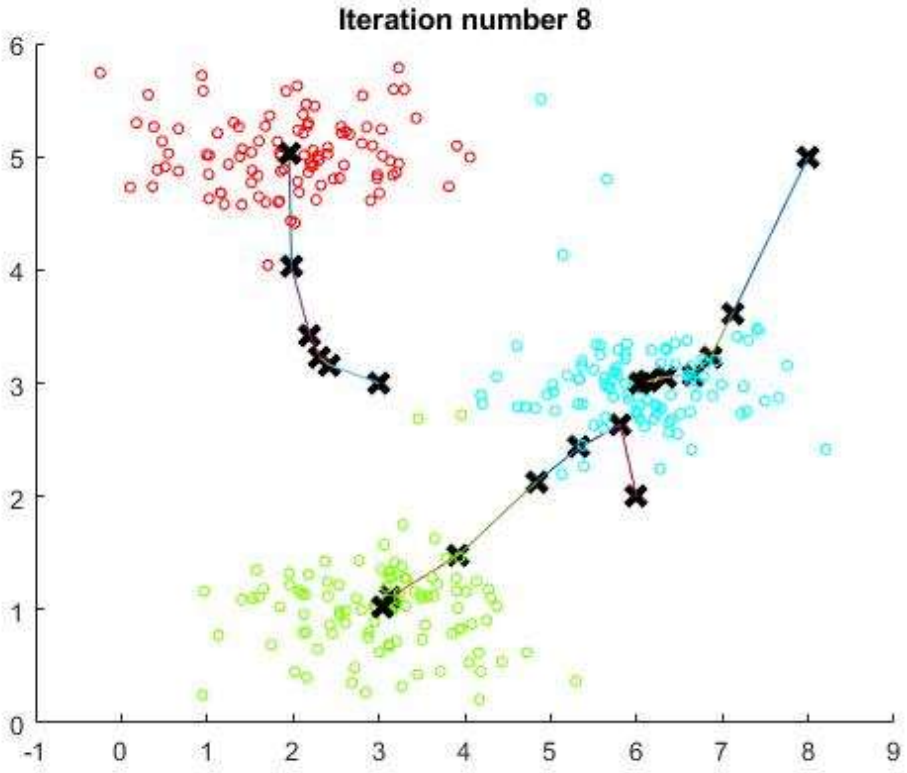

Iteration number 3
6
5
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9

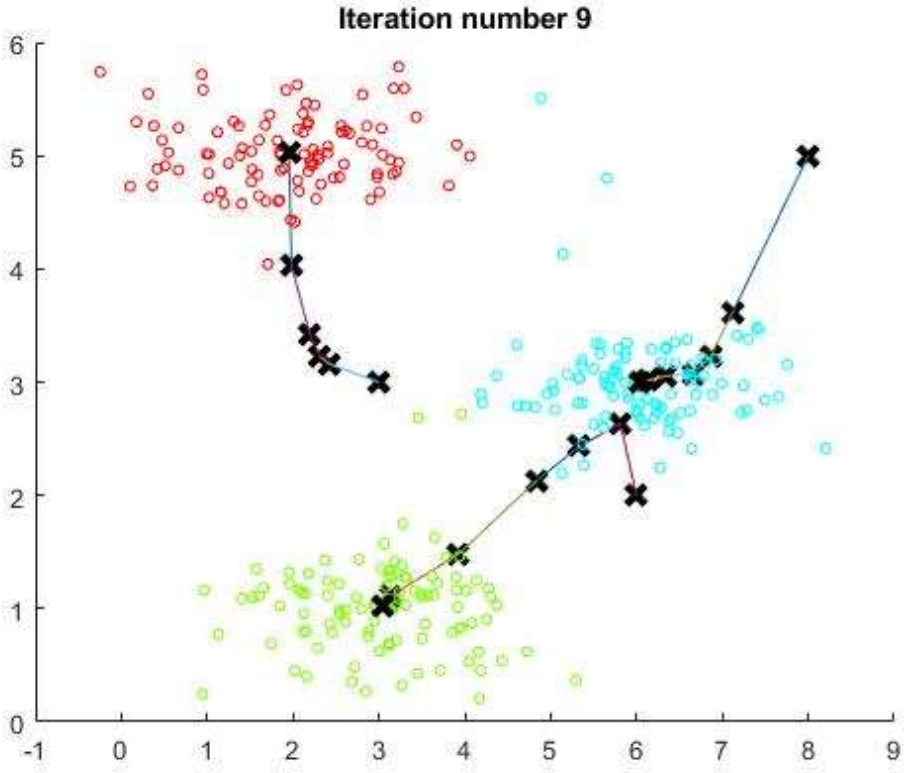

Iteration number 4
6
5
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9

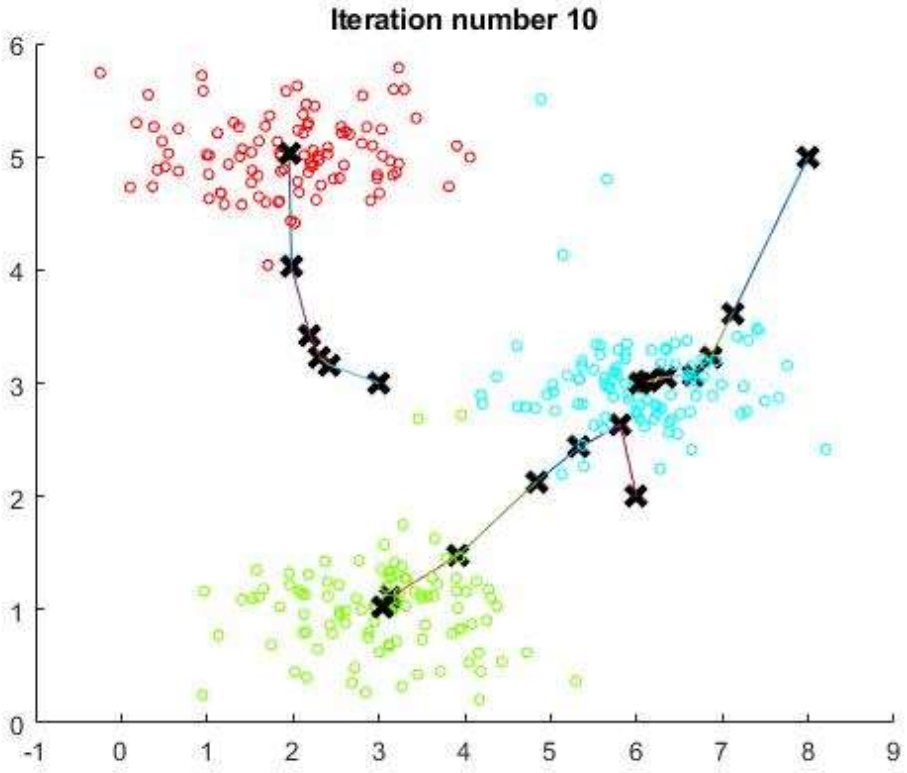

Iteration number 5
6
5
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9

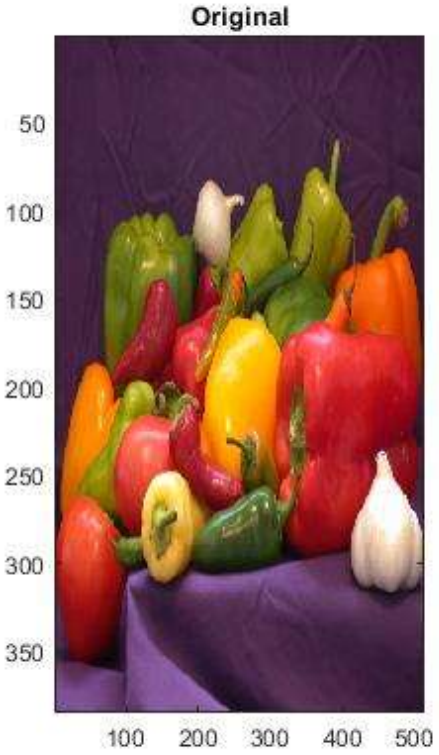

Iteration number 6
6
5
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9

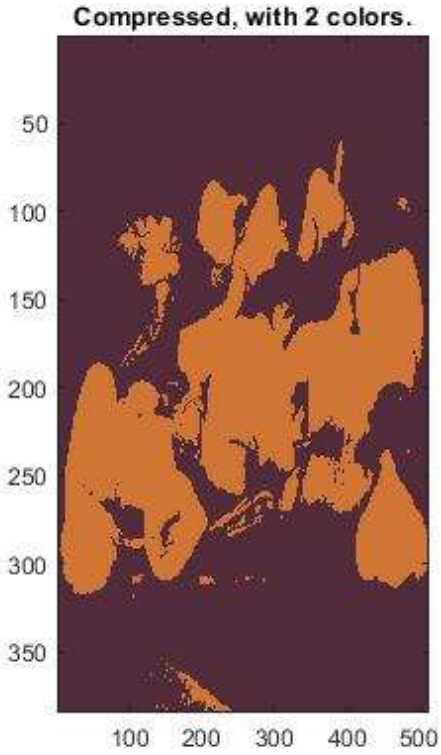

Iteration number 7
6
5
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9


Iteration number 8
6
5
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9

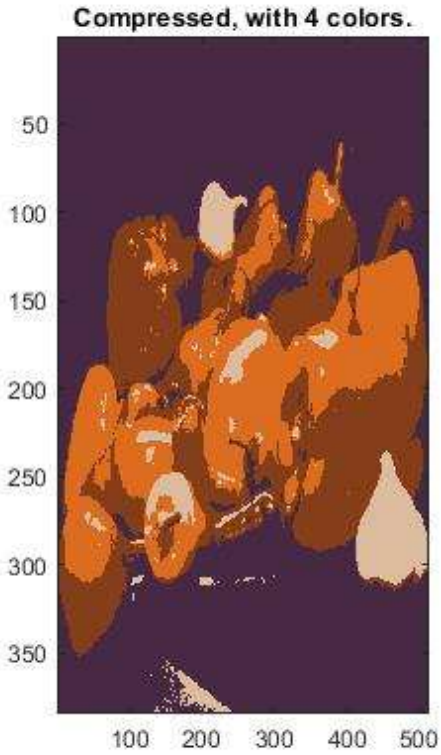

Iteration number 9
6
5
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9

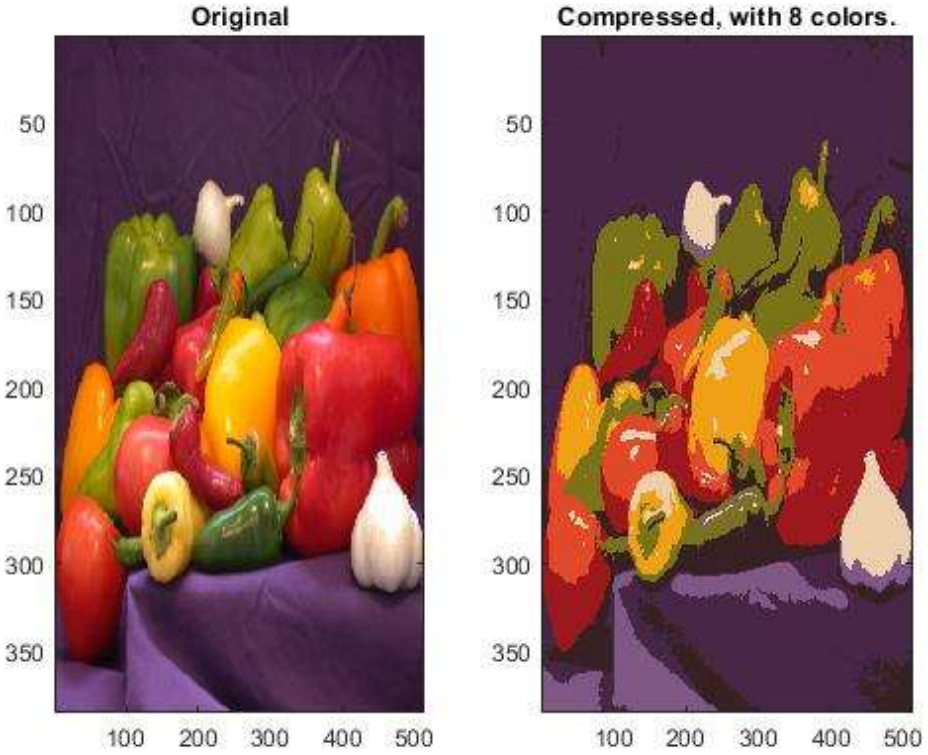

Iteration number 10
6
5
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9

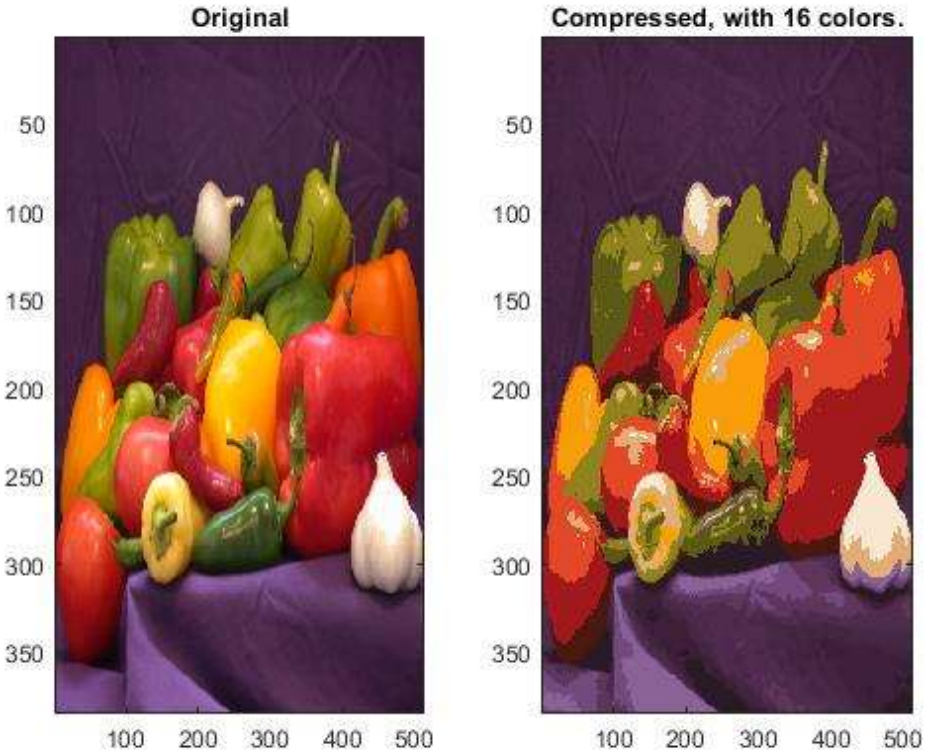

Original
50
100
150
200
250
300
350
100
200
300
400
500


Compressed, with 2 colors.
50
100
150
200
250
300
350
100
200
300
400
500


Original
50
100
150
200
250
300
350
100
200
300
400
500

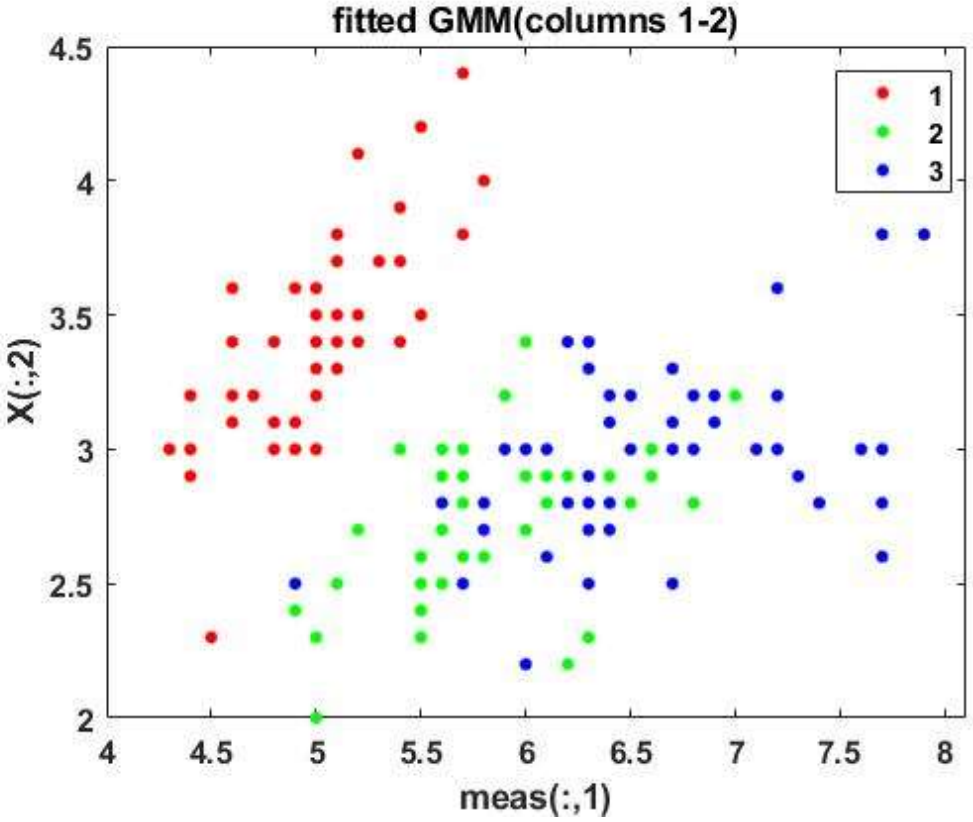

Compressed, with 4 colors.
50
100
150
200
250
300
350
100
200
300
400
500

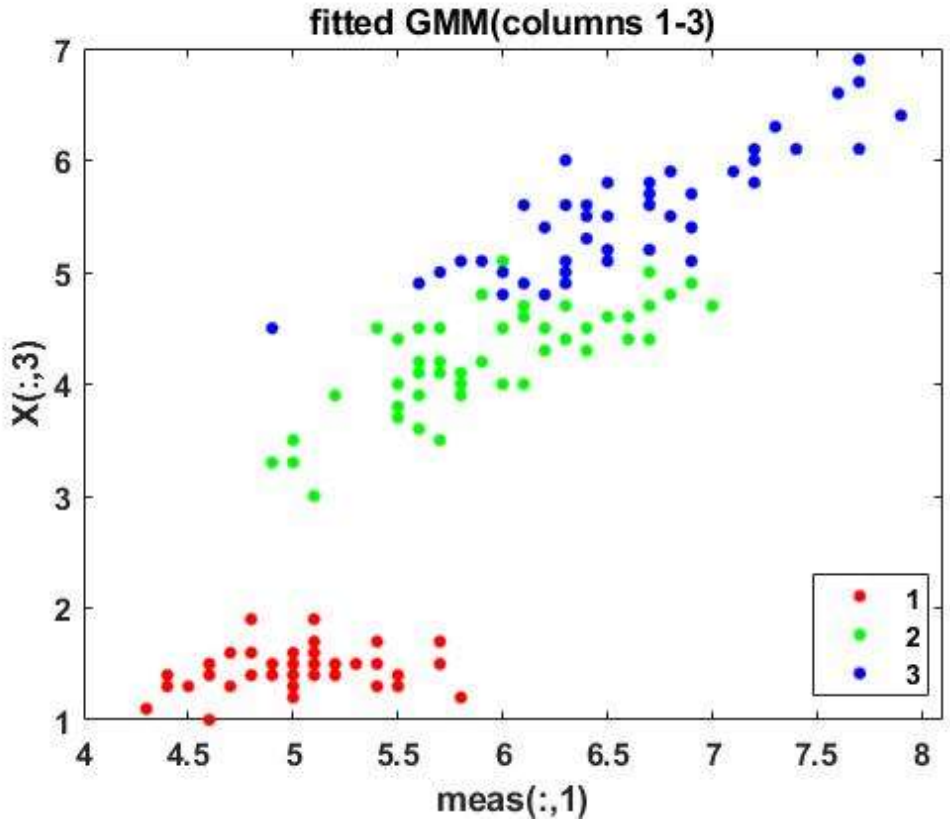


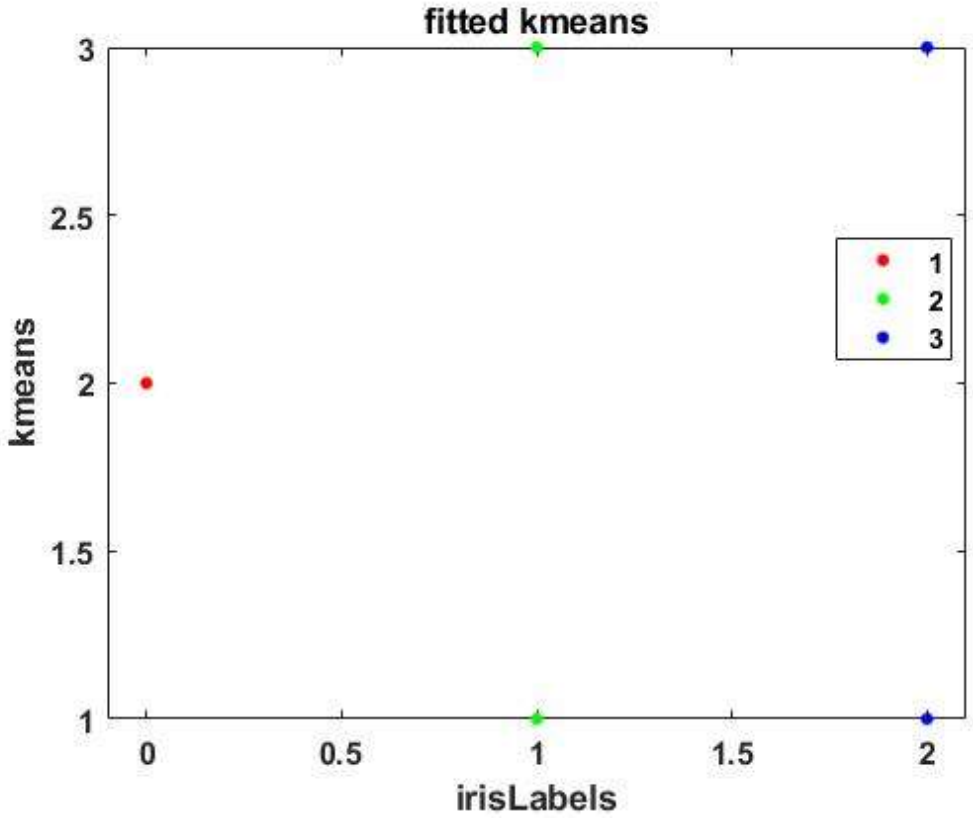


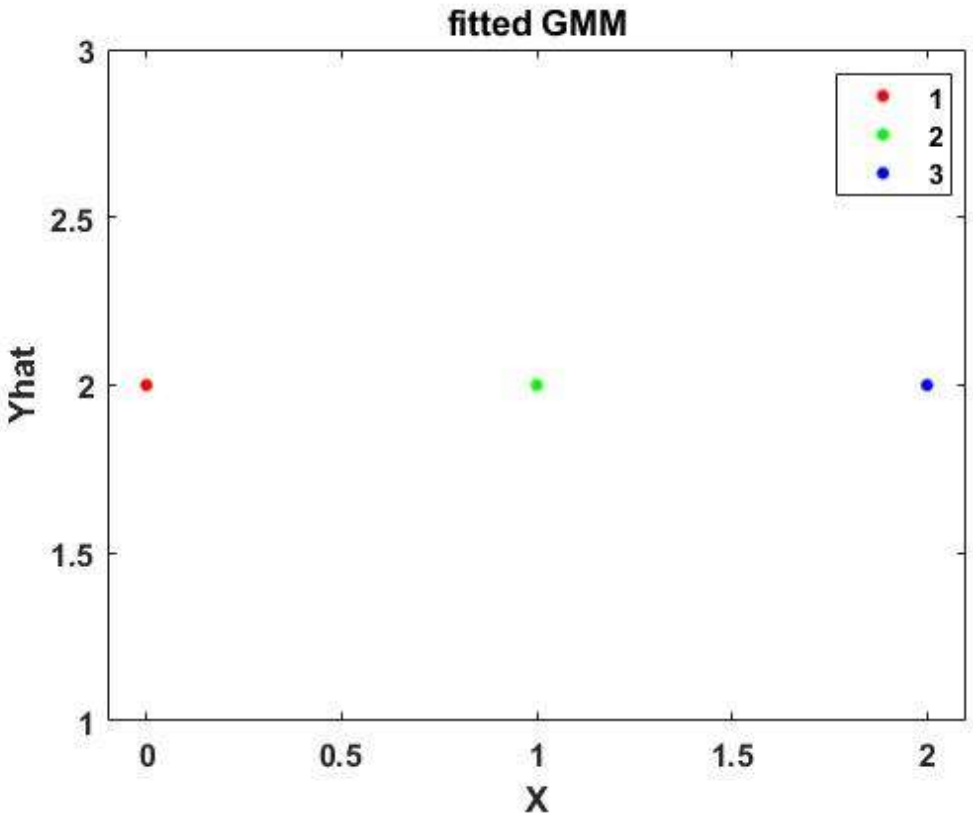
Παρατηρούμε λοιπόν ότι όσο μεγαλύτερη είναι η τιμή του K, τόσο πιο ευδιάκριτο είναι το αποτέλεσμα.

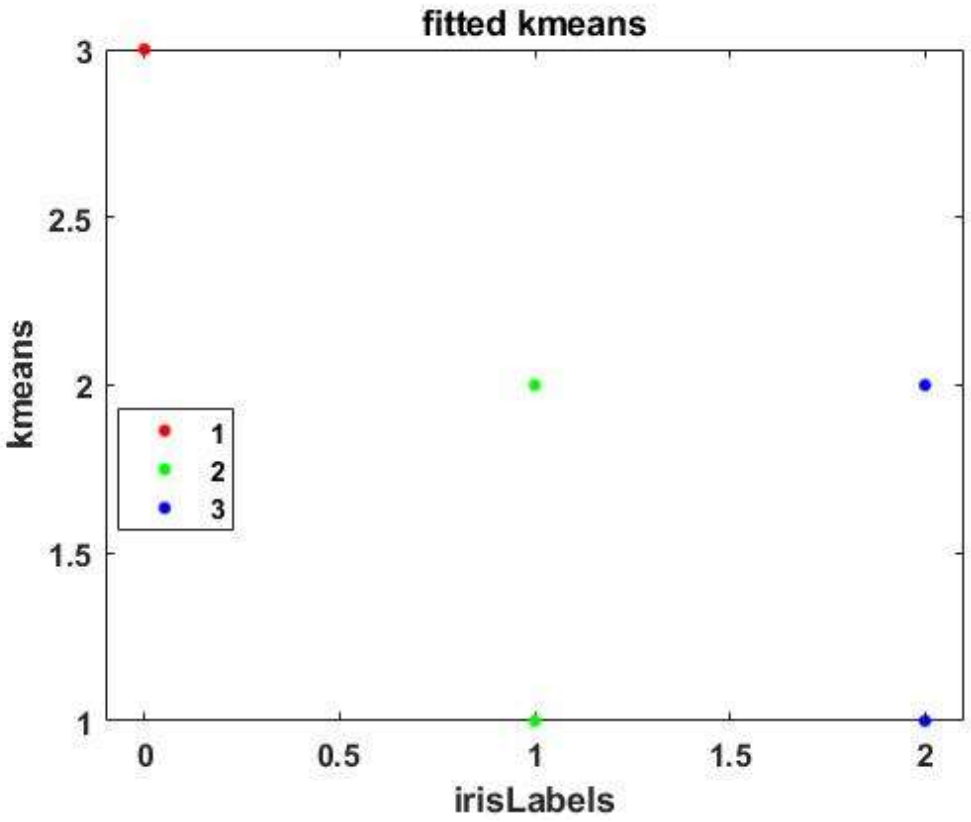
3. Η άσκηση αυτή είχε σκοπό να υλοποιήσουμε τον αλγόριθμο Expectation Maximization, ώστε να εκπαιδεύσουμε τις παραμέτρους ενός μοντέλου Gaussian Mixture (GMM). Δοκιμάσαμε τον αλγόριθμο στα λουλούδια fisher iris και πήραμε τις παρακάτω γραφικές:






fitted kmeans
kmeans
irisLabels
1
2
3
0
0.5
1
1.5
2


fitted GMM
Yhat
X
1
2
3
0
0.5
1
1.5
2



Από την κατανομή των δειγμάτων σε κάθε γραφική, παρατηρούμε ότι ταξινομούνται λάθος τα λουλούδια στην $2^{η}$ και $3^{η}$ ομάδα κυρίως, από τον GMM αλγόριθμο. Στον kmeans είναι λίγο διαφορετικά. Μπορεί να μην έχουμε λάθος (σε οποιαδήποτε ομάδα), άρα έχουμε μεγάλο accuracy, όμως μπορεί να έχουμε πάρα πολλά κι επομένως να έχουμε μηδαμινό accuracy. Στη συγκεκριμένη υλοποίηση, το μέτρο σύγκρισης για τους 2 αλγορίθμους ήταν το Accuracy που υπολογίζω για τον κάθε αλγόριθμο και παρατήρησα ότι στον kmeans ήταν πολύ μεγάλη η διαφοροποίηση από του GMM. Για αρκετές υλοποιήσεις (εκτελώντας αρκετές φορές τον κώδικα) παρέμενε χαμηλά το accuracy (<10% ή 30%) για τον kmeans, με κάποιες ξαφνικές εκτινάξεις στο 56% περίπου. Δηλαδή μπορεί να ήταν από 1.33% έως 56%, ενώ ο GMM ήταν σταθερός στο 33.33% το οποίο είναι λογικό αφού ο GMM είναι greedy αλγόριθμος. Ενδεικτικά:

1) kmeans accuracy = 1.33%.
   GMM accuracy = 33.33%.
2) kmeans accuracy = 56.00%.
   GMM accuracy = 33.33%.