

# Μηχανική Μάθηση

## 3<sup>ο</sup> Σετ Ασκήσεων

ΤΜΗΜΑ: Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

ΟΝΟΜΑ: ΚΩΝΣΤΑΝΤΙΝΟΣ

ΕΠΩΝΥΜΟ: ΛΑΓΑΡΟΣ

ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ: 1053705

ΕΤΟΣ ΦΟΙΤΗΣΗΣ: 6ο

### Πρόβλημα 3.1

Το πρόβλημα αυτό αποτελεί ουσιαστικά υλοποίηση της μεθόδου που μάθαμε με την οποία, μέσω ενός Πυρήνα και υλοποιήσεων μιας τυχαίας μεταβλητής, μπορούμε να προσεγγίσουμε την συνάρτηση πυκνότητας πιθανότητας της.

Ο πυρήνας(Kerner) θα πρέπει να είναι :  $K(x, h) = g\left(\frac{x}{h}\right) \times \frac{1}{h}$  , όπου  $g(x)$  μια συνάρτηση ώστε  $\int g(x)dx = 1$  .

Για τα Kernel αυτά ισχύει  $\lim_{h \rightarrow 0} K(x, h) = \delta(x)$  , όπου  $\delta(x)$  η γνωστή συνάρτηση Δέλτα του Dirac.

Στο παρόν πρόβλημα έχουμε επιλέξει το Gaussian Kernel :  $K(x, h) = \frac{1}{\sqrt{2\pi}h} e^{-\frac{1}{2h^2}x^2}$  , που είναι η Γκαουσιανή πυκνότητα πιθανότητας με διασπορά  $h$  .

Από τις ιδιότητες της συνάρτησης Δέλτα, και από αναλυτικό ορισμό του Μέσου Όρου έχουμε :

$$f(Y) = \int f(X)\delta(Y - X)dX \approx \int f(X)K(Y - X, h)dX = E_x[K(Y - X, h)]$$

, όπου  $f(Y)$  σ.π.π της μεταβλητής  $Y$ .

Και από τον Νόμο των μεγάλων αριθμών έχουμε :

$$E_x[K(Y - X, h)] \approx \frac{1}{N} \{K(Y - X_1, h) + \dots + K(Y - X_N, h)\}$$

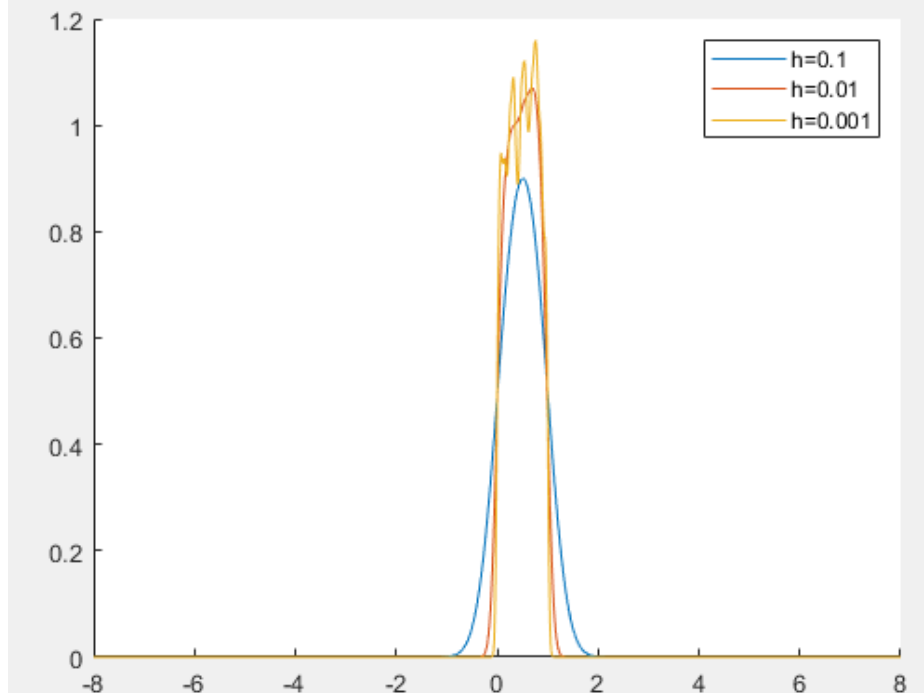
Άρα μπορούμε να προσεγγίσουμε μια σ.π.π μιας τυχαίας μεταβλητής μέσα από υλοποίησης της ως :

$$f(X) \approx \frac{1}{N} \{K(X - X_1, h) + \dots + K(X - X_N, h)\}$$

**Προφανώς όσο μεγαλύτερο το  $N$  , τόσο καλύτερη και προσέγγισή μας.**

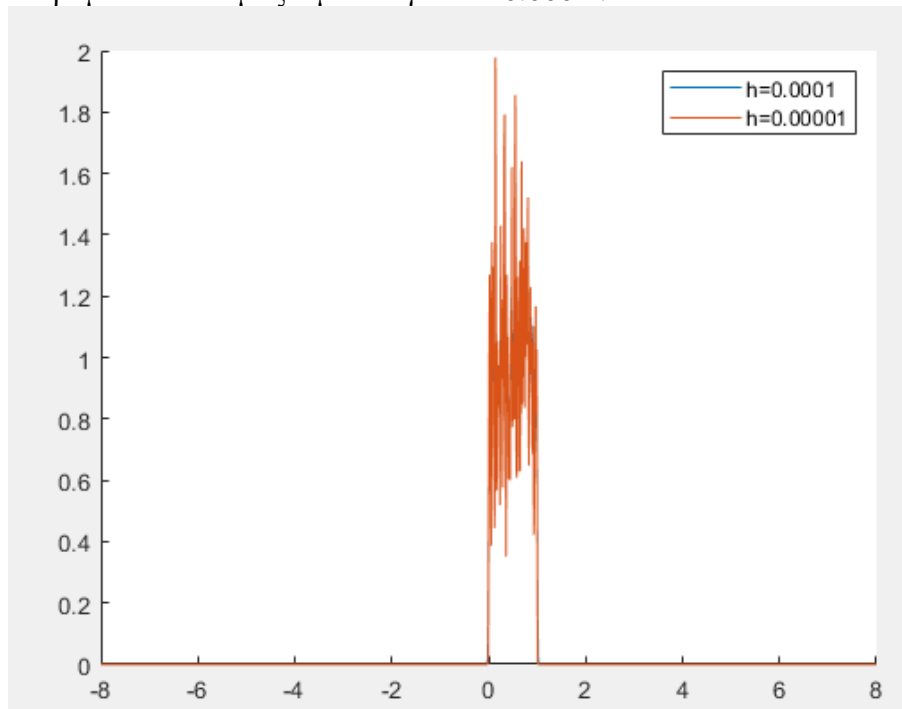
Στο συγκεκριμένο πρόβλημα καλούμαστε να προσεγγίσουμε την σ.π.π μιας ομοιόμορφης στο  $[0,1]$  τυχαίας μεταβλητής έχοντας στην διάθεσή μας 1000 υλοποίησης της.

Η προσέγγιση έγινε για διάφορες τιμές της μεταβλητής  $h$  του Kernel.



Παρατηρούμε ότι όσο για μεγάλες τιμές του  $h$  η προσέγγιση μας χάνει σημαντικά στο πεδίο ορισμού της  $f(X)$  που είναι το  $[0,1]$ . Όσο μικραίνει το  $h$  (πχ  $h = 0.01$ ) παρατηρούμε μια βελτίωση τόσο στο πεδίο ορισμού όσο και στο πεδίο τιμών. Για μεγαλύτερες τιμές όμως (πχ  $h = 0.001$ ) παρατηρούμε ότι αν και έχουμε κερδίσει στην προσέγγιση του πεδίου ορισμού, έχουμε χάσει στην προσέγγιση του πεδίου τιμών που είναι η μονάδα.

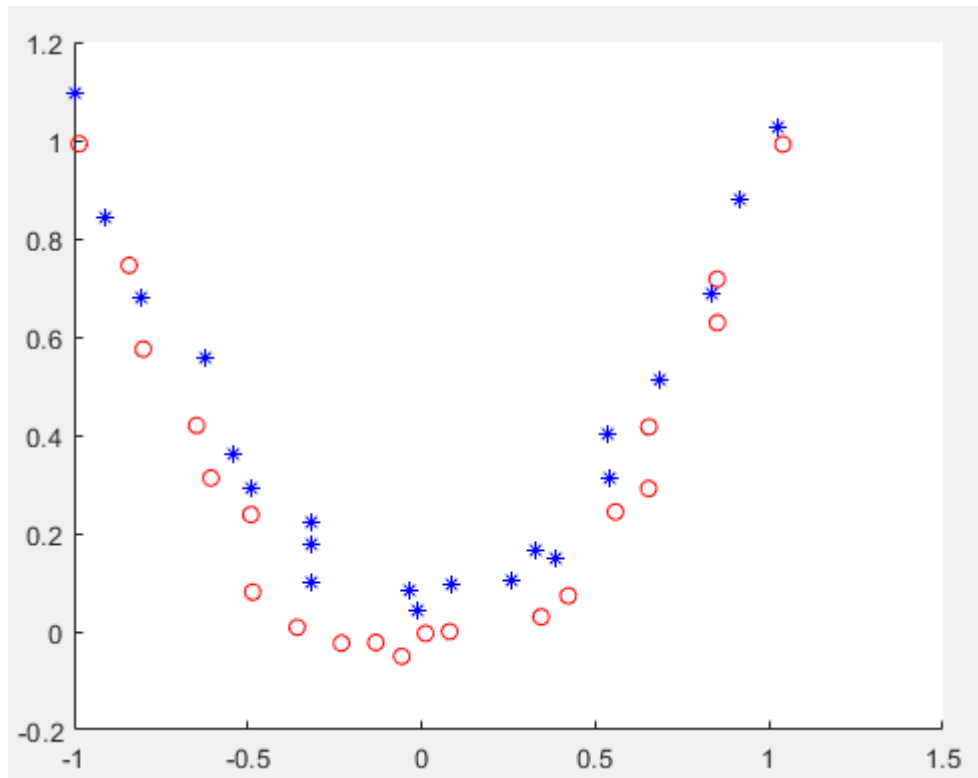
Πειραματικά δοκιμάζουμε και για  $h = 0.0001$  :



Από εδώ καταλαβαίνουμε πως για  $h \leq 0.0001$  αρχίζουμε και πιάνουμε ακριβώς το πεδίο ορισμού αλλά θα χάνουμε όλο και περισσότερο στην προσέγγιση του πεδίου τιμών όσο μικραίνουμε το  $h$ .

### Πρόβλημα 3.2

Σε αυτό το πρόβλημα έχουμε δεδομένα 2δισδιάστατα από 2 κλάσεις (stars, circles).



Και χρησιμοποιώντας την μέθοδο (Support Vector Machines) με τους πυρήνες Mercer προσπαθούμε να βρούμε ένα διαχωριστικό σύνορο μεταξύ τους.

Την εν λόγω αντιστοίχιση των δεδομένων μας στις αντίστοιχες κλάσεις τους (stars='1', circles='-1') θα πραγματοποιεί μια συνάρτηση  $\Phi(X)$ ,  $X = [x_1 x_2]^T$

Η συνάρτηση  $\Phi(x)$  ζει σε έναν διανυσματικό χώρο  $V$  που δημιουργεί ο πυρήνας Mercer,  $K(X, Y) = e^{-\frac{1}{h}\|X-Y\|^2} = e^{-\frac{1}{h}\{(x_1-y_1)^2+(x_2-y_2)^2\}}$

Και σε αυτόν τον διανυσματικό χώρο ορίζουμε εσωτερικό γινόμενο :

$$\langle \Phi(X), \Psi(X) \rangle = \sum_{i=0}^m \sum_{j=1}^{m'} \alpha_i \beta_j K(X_i, X_j)$$

Πιο συγκεκριμένα η  $\Phi(X) = \sum_{i=1}^m \alpha_i K(X, X_i)$  , για **οποιοδήποτε** πεπερασμένο πλήθος διανυσμάτων  $X_1, X_2, \dots, X_m \in R^2$ , οποιουσδήποτε πραγματικούς αριθμούς  $\alpha_1, \alpha_2, \dots, \alpha_m$ .

Προκειμένου να βρούμε την  $\Phi(X)$  επιλύουμε το παρακάτω πρόβλημα βελτιστοποίησης :

$$\min_{\Phi \in V} \left\{ \sum_{X_i \in \text{stars}} (1 - \Phi(X_i))^2 + \sum_{X_j \in \text{circles}} (1 + \Phi(X_j))^2 + \lambda \|\Phi(X)\|^2 \right\} =$$

$$\min_{\Phi \in V} \left\{ \sum_{X_i} (y_i - \Phi(X_i))^2 + \lambda \|\Phi(X)\|^2 \right\}$$

, όπου  $X_i$  όλα μας τα δεδομένα

και  $y_i = 1$  όταν  $X_i \in \text{stars}$  και  $y_i = -1$  όταν  $X_j \in \text{circles}$

α)

Στο πρόβλημα αυτό καλούμαστε να λύσουμε και να βρούμε τις βέλτιστες παραμέτρους  $m, \alpha_1, \alpha_2, \dots, \alpha_m, X_1, X_2, \dots, X_m$  το οποίο είναι αρκετά δύσκολο καθώς οι παράμετροι  $X_1, X_2, \dots, X_m$  μπαίνουν στο πρόβλημα μη γραμμικά.

Το πρόβλημα αυτό μπορεί να γίνει γραμμικό αν κάνουμε χρήση του Representer Theorem.

Δημιουργούμε έναν γραμμικό υπόχωρο του  $V$ , ο οποίος θα είναι η γραμμική θήκη  $\Omega = \{c_1 K(X, X_1) + c_2 K(X, X_2) + \dots + c_N K(X, X_N)\}$ , όπου  $X_1, X_2, \dots, X_N$  τα δεδομένα μας. Σε αυτόν τον υπόχωρο  $\Omega$  ζούμε συναρτήσεις  $\hat{\Phi}(X) = \sum_{i=1}^N a_i K(X, X_i)$ .

τότε από αρχή της ορθογωνιότητας έχουμε :

$$\langle \Phi(X) - \hat{\Phi}(X), K(X, X_i) \rangle = 0$$

$$\langle \Phi(X), K(X, X_i) \rangle = \langle \hat{\Phi}(X), K(X, X_i) \rangle$$

(από ορισμό εσωτερικού γινομένου)

$$\Phi(X_i) = \hat{\Phi}(X_i)$$

που σημαίνει ότι η συνάρτηση  $\Phi(X)$  την οποία ψάχνουμε έχει την ίδια τιμή με την κάθετη προβολή της  $\hat{\Phi}(X)$  στον υπόχωρο  $\Omega$  στα  $X_1, X_2, \dots, X_N$  (τα δεδομένα μας).

Οπότε μπορούμε να λύσουμε το ισοδύναμο πρόβλημα βελτιστοποίησης :

$$\min \left\{ \sum_{X_i} (y_i - \hat{\Phi}(X_i))^2 + \lambda \|\Phi(X)\|^2 \right\}$$

β)

Επίσης έχουμε ότι :

$$\|\Phi(X)\|^2 = \|\hat{\Phi}(X) + \Phi(X) - \hat{\Phi}(X)\|^2$$

$$= \| \hat{\Phi}(X) \|^2 + \| \Phi(X) - \hat{\Phi}(X) \|^2 + 2\langle \hat{\Phi}(X), \Phi(X) - \hat{\Phi}(X) \rangle$$

$$\text{Όμως } \langle \hat{\Phi}(X), \Phi(X) - \hat{\Phi}(X) \rangle = 0$$

, γιατί όπως είπαμε η  $\hat{\Phi}(X)$  είναι η κάθετη προβολή της  $\Phi(X)$  στον υπόχωρο  $\Omega$ , οπότε το διάνυσμα  $\Phi(X) - \hat{\Phi}(X)$  θα είναι κάθετο σε κάθε διάνυσμα του  $\Omega$ , και το  $\hat{\Phi}(X) \in \Omega$ , οπότε το εσωτερικό γινόμενο θα είναι ίσο με 0.

Άρα τελικά έχω :

$$\| \Phi(X) \|^2 = \| \hat{\Phi}(X) \|^2 + \| \Phi(X) - \hat{\Phi}(X) \|^2$$

$$\text{που σημαίνει ότι : } \| \hat{\Phi}(X) \|^2 \leq \| \Phi(X) \|^2$$

Επομένως μπορώ να αντικαταστήσω το  $\| \Phi(X) \|^2$  με το  $\| \hat{\Phi}(X) \|^2$  αφού θα βρω κάτι μικρότερο στην βελτιστοποιώντας το.

Άρα **τελικά** το πρόβλημα βελτιστοποίησης που λύνω είναι το :

$$\min_{\hat{\Phi} \in \Omega} \left\{ \sum_{i=0}^N (y_i - \hat{\Phi}(X_i))^2 + \lambda \| \hat{\Phi}(X) \|^2 \right\}$$

Το μεγάλο κέρδος εδώ είναι πως πλέον οι βέλτιστοι παράμετροι  $m=N$  και  $X_1, X_2, \dots, X_m = X_1, X_2, \dots, X_N$  είναι γνωστοί και οι μόνοι παράμετροι που ψάχνω είναι οι  $c_1, c_2, \dots, c_m$ , οι οποίοι όμως μπαίνουν με γραμμικό τρόπο.

γ)

Προτού αρχίσουμε να παραγωγίζουμε ως προς τις παραμέτρους  $c$  ας φέρουμε την συνάρτηση κόστους που θέλουμε να ελαχιστοποιήσουμε σε μια καλύτερη μορφή :

$$\hat{\Phi}(X_i) = \sum_{i=0}^N c_i K(X, X_i) = [K(X, X_1) \quad K(X, X_2) \quad \dots \quad K(X, X_N)] * \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}$$

Άρα :

$$\sum_{i=0}^N (y_i - \hat{\Phi}(X_i))^2 = (y_1 - \hat{\Phi}(X_1))^2 + (y_2 - \hat{\Phi}(X_2))^2 + \dots + (y_n - \hat{\Phi}(X_n))^2 =$$

$$= \left( y_1 - [K(X_1, X_1) \quad K(X_1, X_2) \quad \dots \quad K(X_1, X_N)] * \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} \right)^2 +$$

$$\left( y_2 - [K(X_2, X_1) \quad K(X_2, X_2) \quad \dots \quad K(X_2, X_N)] * \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} \right)^2 + \dots +$$

$$\left( y_N - [K(X_N, X_1) \quad K(X_N, X_2) \quad \dots \quad K(X_N, X_N)] * \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} \right)^2$$

Άρα

$$\sum_{i=0}^N \left( y_i - \hat{\Phi}(X_i) \right)^2 = \| Y - K \times C \|^2$$

,όπου Y είναι ένα διάνυσμα μήκους N (=42 ,όσα και τα δεδομένα μας) με τα πρώτα 21 στοιχεία ίσα με 1 και τα υπόλοιπα 21 ίσα με -1

,C είναι το διάνυσμα των αγνώστων παραμέτρων που ψάχνουμε

,και K είναι ο πίνακας :

$$K = \begin{bmatrix} K(X_1, X_1) & \dots & K(X_1, X_N) \\ \vdots & \ddots & \vdots \\ K(X_N, X_1) & \dots & K(X_N, X_N) \end{bmatrix} \quad , \text{οποίος είναι συμμετρικός.}$$

$$\text{Επίσης } \|\hat{\Phi}(X)\|^2 = \langle \hat{\Phi}(X), \hat{\Phi}(X) \rangle = C^T K C$$

Οπότε έχω :

$$\min_{\min C} \{ \| Y - K \times C \|^2 + \lambda C^T K C \}$$

Και παραγωγίζω ως προς C εξισώνοντας με 0 :

$$\nabla_C \{ \| Y - K \times C \|^2 + \lambda C^T K C \} = 0$$

$$\nabla_C \{ C^T (K^T K + \lambda K) C - 2 C^T K Y \} = 0$$

$$2(K^T K + \lambda K)C - 2KY = 0$$

$$(K^T + \lambda I)C = Y$$

Οπότε αν υποθέσουμε ότι υπάρχουν οι αντίστροφοι πίνακες οι βέλτιστοι συντελεστές είναι :

$$C = (K^T + \lambda I)^{-1} Y$$

δ)

Έχοντας προσδιορίσει το βέλτιστο  $\hat{\Phi}(X)$  μπορούμε να το χρησιμοποιήσουμε για να κατηγοριοποιήσουμε πλέον ένα νέο σημείο  $X_{new}$ .

Ο τρόπος που θα το κάνουμε είναι το πρόσημο της  $\hat{\Phi}(X_{new})$ .

Αν  $\hat{\Phi}(X_{new}) > 0$ , τότε το  $X_{new}$  θα κατηγοριοποιείται ως star, ενώ

Αν  $\hat{\Phi}(X_{new}) < 0$ , τότε το  $X_{new}$  θα κατηγοριοποιείται ως circle.

δ)

Το σύνορό μας δηλαδή είναι το  $\hat{\Phi}(X) = 0$ , όπου  $X = [x_2]^T$ .

Για να το σχεδιάσουμε χρειάζεται να βρούμε για ποια  $X$  ικανοποιείται η παραπάνω σχέση.

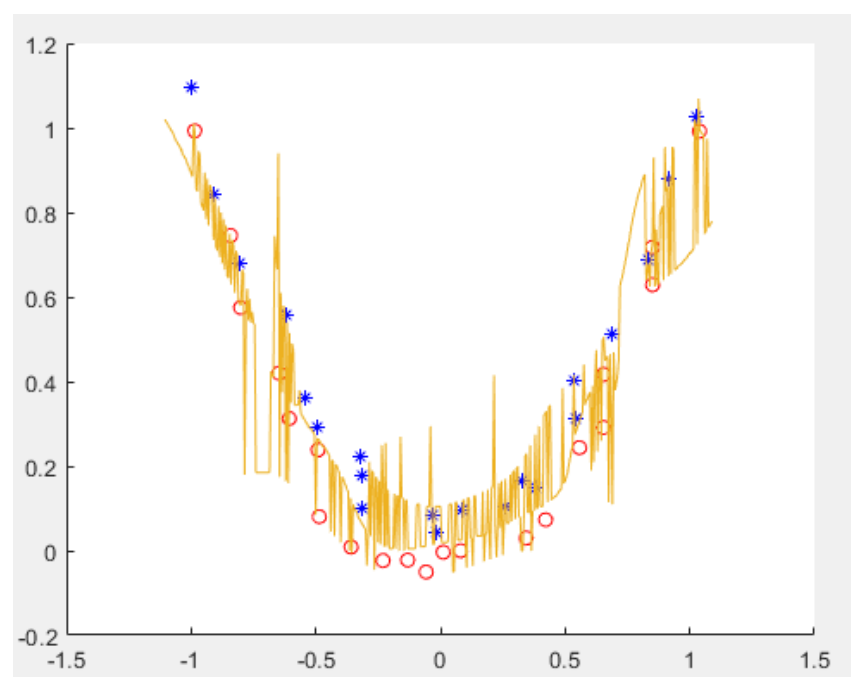
Ο τρόπος βρίσκουμε είναι αριθμητικός δηλαδή δειγματοληπτούμε το  $x_1$  και κάθε τιμή του λύνουμε την εξίσωση  $\hat{\Phi}([x_1 x_2]) = 0$

Την εξίσωση αυτή τώρα μπορούμε να την λύσουμε επίσης αριθμητικά δειγματοληπτώντας το  $x_2$ .

Οπότε για κάθε τιμή του  $x_1$  πάμε υπολογίζουμε το  $\hat{\Phi}([x_1 x_2])$  και κρατάμε την το ζευγάρι  $(x_1, x_2)$  για το οποίο η συνάρτηση  $\hat{\Phi}([x_1 x_2])$  μας έδωσε την μικρότερη απόλυτη τιμή (δηλαδή όσο πιο κοντά στο 0).

Τα αποτελέσματα για διάφορες τιμές του  $h$  και  $\lambda$ :

```
h = 1; % kernel h parameter  
l = 0; %lambda in equation
```



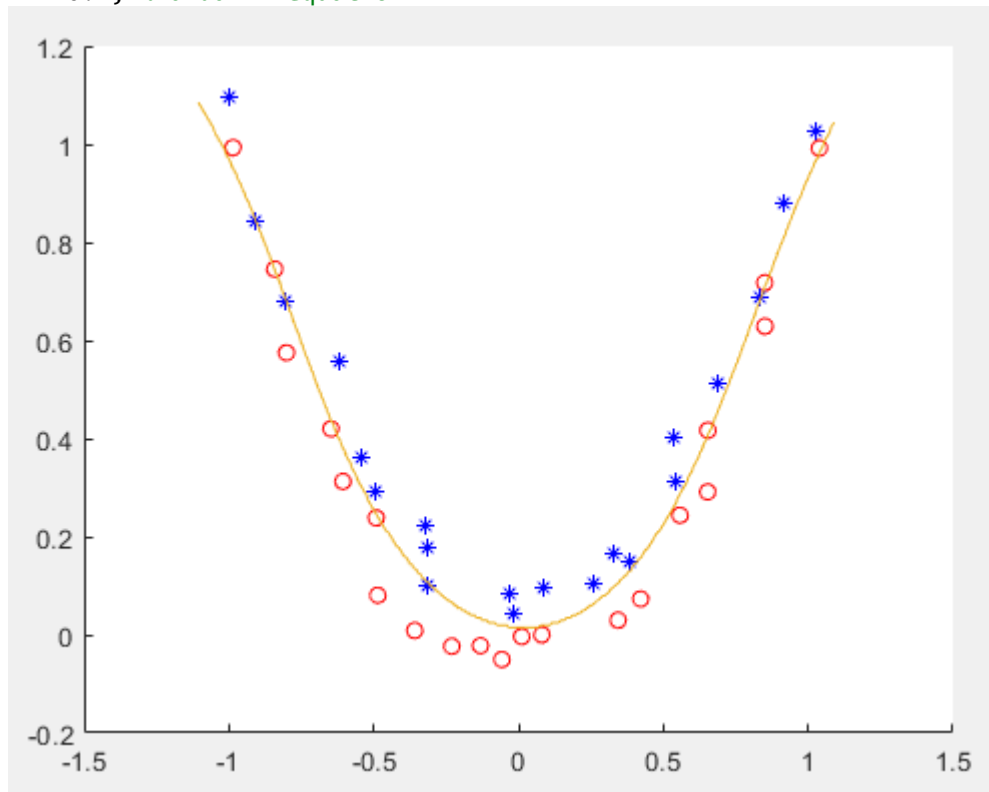


Για  $\lambda=0$  έχουμε την περίπτωση του **overfitting**.

(οπότε είναι λογικό που το σύνολο μας βγήκε ‘κατσαρό’)

Σε αυτή την περίπτωση έχουμε φτιάξει έναν κανόνα κατηγοριοποίησης ο οποίος κάνει 0 σφάλμα για τα ήδη γνωστά δεδομένα αλλά περισσότερα σφάλματα για νέα δεδομένα.

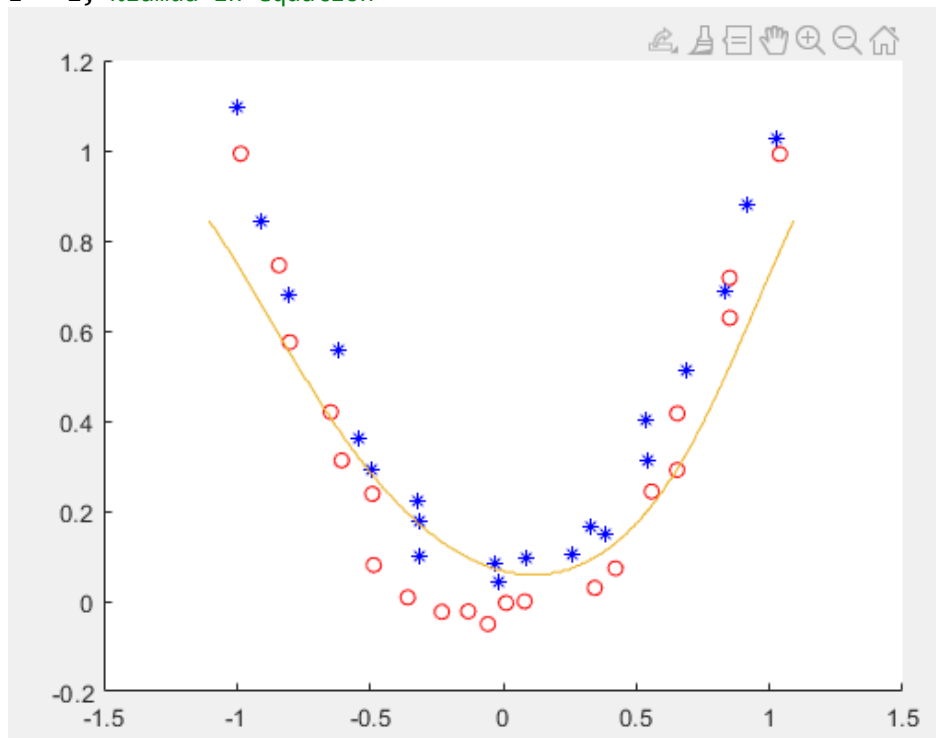
```
h = 1; % kernel h parameter  
l = 0.1; %lambda in equation
```



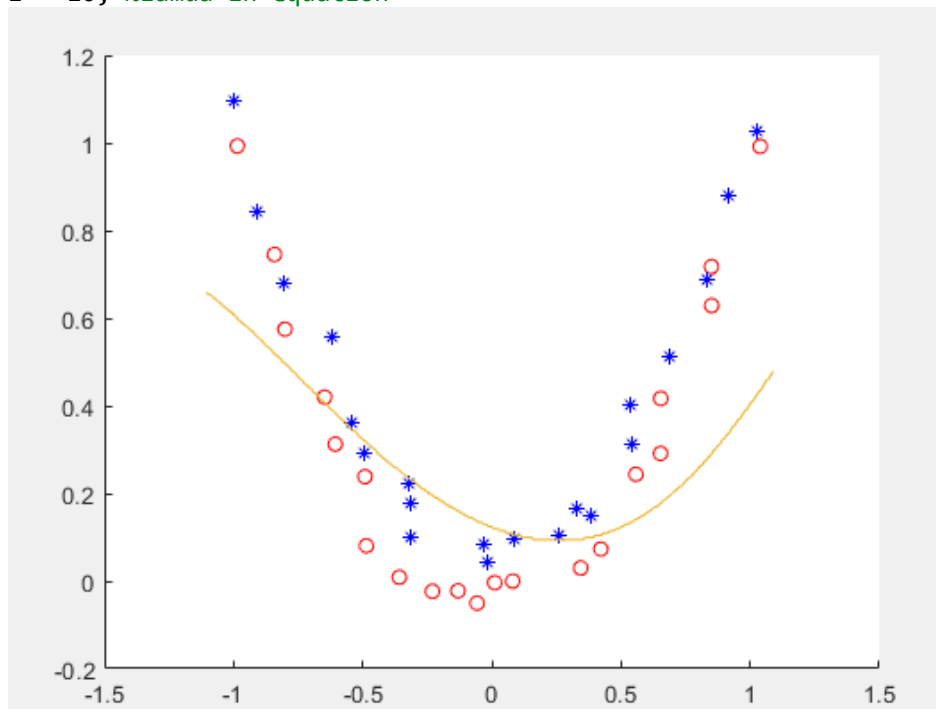
Για αυτό το ζευγάρι  $h$  και  $\lambda$  φαίνεται να δημιουργεί ένα αρκετά ικανοποιητικό όριο.

Ας δούμε ενδεικτικά και άλλα ζευγάρια  $h$  και  $\lambda$  όχι τόσο αποδοτικά :

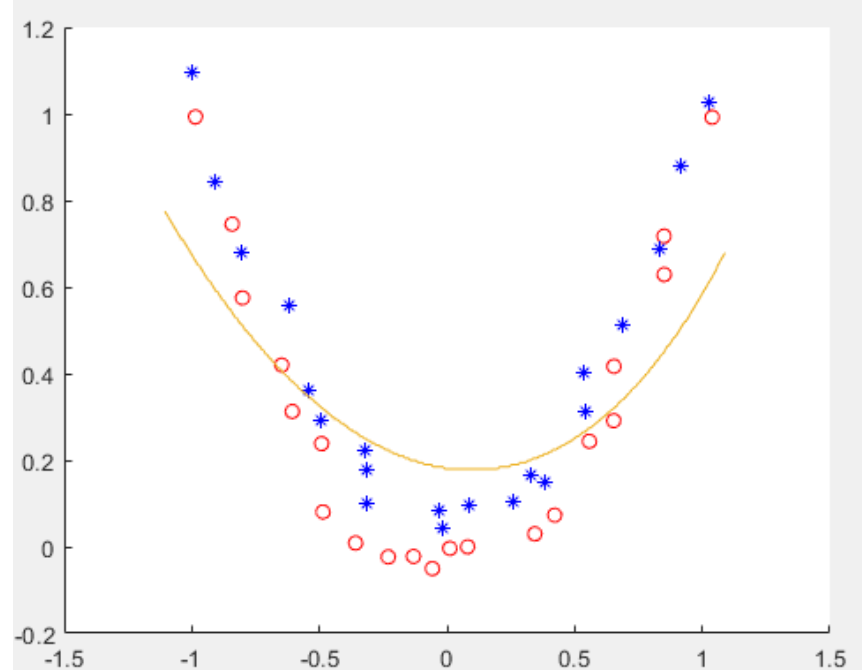
```
h = 1; % kernel h parameter  
l = 2; %lamda in equation
```



```
h = 1; % kernel h parameter  
l = 10; %lamda in equation
```



```
h = 10; % kernel h parameter
l = 0.1; %lamda in equation
```



## Πρόβλημα 3.2

Έχουμε 200 διανύσματα μήκους 2 τα οποία ανήκουν σε 2 ομάδες και χρησιμοποιώντας τον αλγόριθμο K-means προσπαθούμε να τα ομαδοποιήσουμε.

Αρχικά ας δούμε ποιο είναι το πρόβλημα που θέλουμε να λύσουμε :

Έχουμε  $X_1, X_2, \dots, X_N$  δεδομένα που ανήκουν στον χώρο  $R^d$ .  
(οι αλγόριθμοι σαν τον K-Means προϋποθέτουν το d να μην είναι πολύ μεγάλο γιατί από ένα σημείο και μετά τα δεδομένα αρχίζουν και φαίνονται να απέχουν πάρα πολύ από το άλλο και αλγόριθμοι αυτοί φαίνεται να μην αποδίδουν).

Έχουμε τις K ομάδες

$$C_k : L_1^k, L_2^k, \dots, L_{n_k}^k, Z_k$$

, όπου k δηλώνει τον αριθμό της ομάδα

$L_j^k$  δηλώνει τον δείκτη του στοιχείου της k ομάδας (πλήθος δεικτών =  $n_k$ )

και  $Z_k$  ο αντιπρόσωπος(κεντρικό στοιχείο) κάθε ομάδας

Ορίζουμε ως συνάρτηση κόστους της συνολική διασπορά :

$$\sum_{l=1}^k \sum_{j=1}^{nl} \| X_{L_j^l} - Z_l \|^2$$

, όπου το εσωτερικό άθροισμα αποτελεί την διασπορά της l ομάδας .

Το κόστος λοιπόν είναι μια συνάρτηση  $C(C_1, C_2, \dots, C_k, Z_1, Z_2, \dots, Z_k)$

Και να προσπαθήσουμε να το ελαχιστοποιήσουμε θα καταλήξουμε σε ένα πολύ μη γραμμικό πρόβλημα βελτιστοποίησης.

Ο K-Means ουσιαστικά αυτό που κάνει είναι ότι φιζάροντας τα  $Z_k$  πάει και υπολογίζει βέλτιστα τα  $C_k$  λύνοντας ένα πλέον πανεύκολο πρόβλημα βελτιστοποίησης, και φιζάροντας τις ομάδες  $C_k$  υπολογίζει τα βέλτιστα νέα κέντρα  $Z_k$ .

Φυσικά ο τρόπος αυτός που σπάμε το πολύ δύσκολο μη γραμμικό πρόβλημα βελτιστοποίησης σε 2 εύκολα γραμμικά προβλήματα δεν έρχεται δίχως τίμημα, καθώς ο K-Means δεν είναι σε θέση να εγγυάται την εύρεση Ολικού Ελάχιστου της συνάρτησης κόστους όταν πάει και συγκλίνει. Αντιθέτως πάει και κολλάει σε κάποιο τοπικό ελάχιστο.

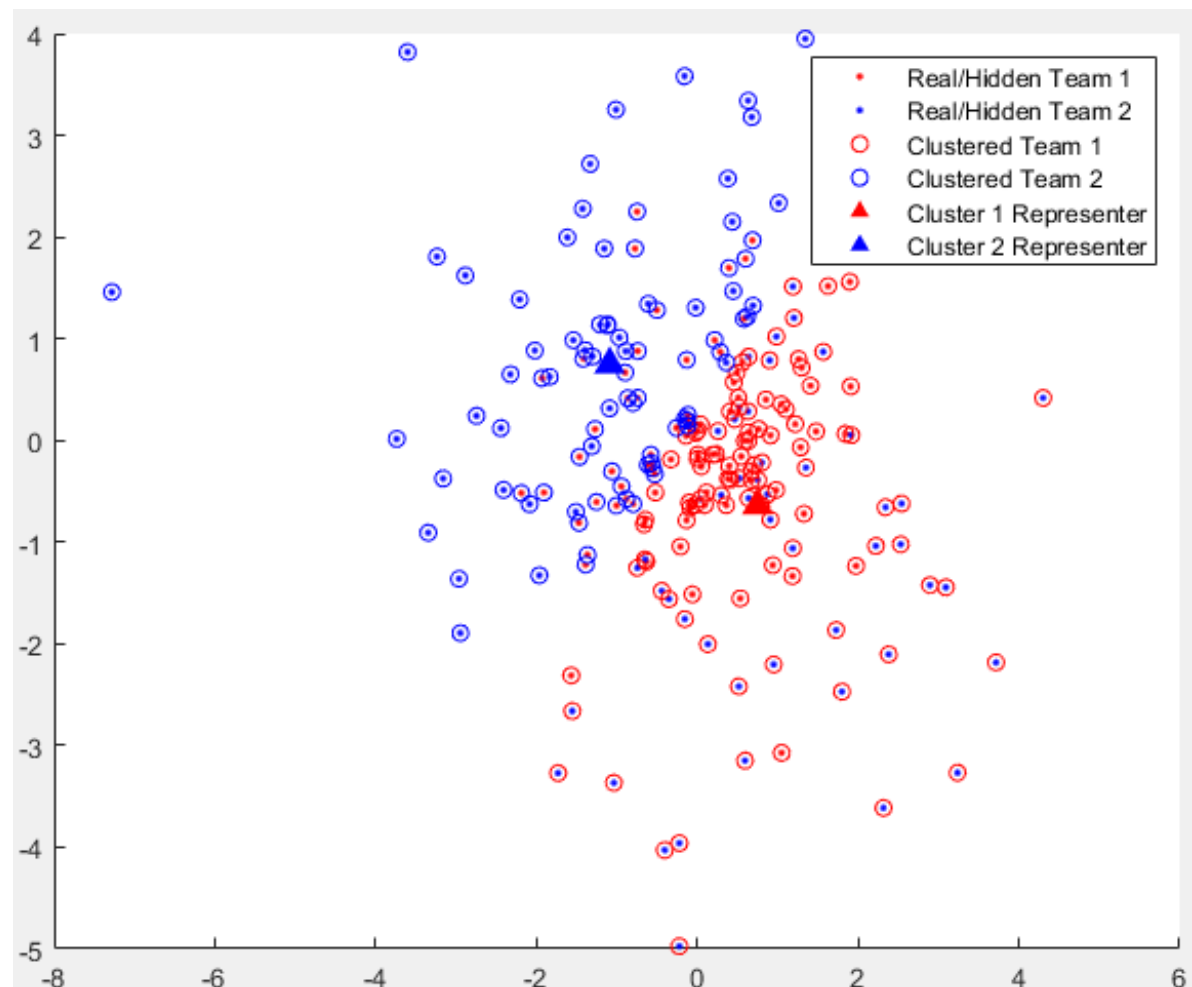
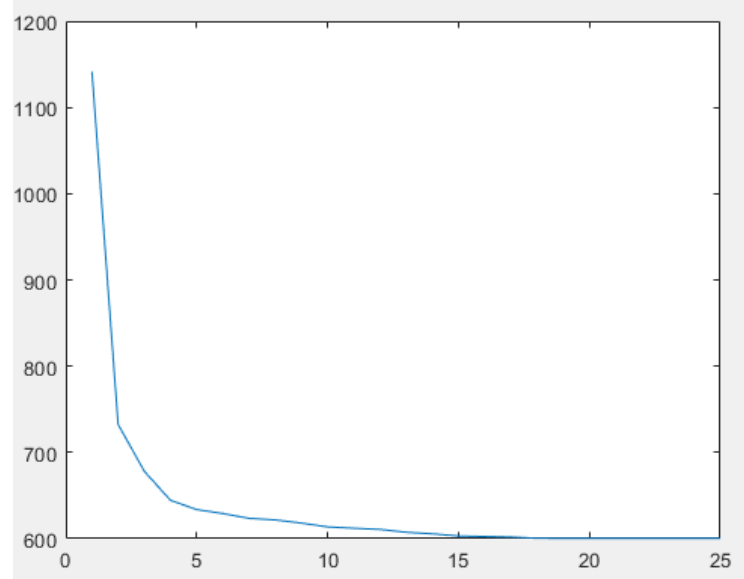
Επομένως ο K-Means , στην περίπτωση της άσκησης που έχουμε 2 ομάδες (K=2) , υλοποιείται με τον εξής απλό τρόπο:

- α) επιλέγουμε τυχαία 2 αντιπροσώπους κάθε ομάδας
- β) ομαδοποιούμε τα στοιχεία έχοντας ως κριτήριο να μουν στην ομάδα που απέχουν λιγότερο από τον αντιπρόσωπό της (τετράγωνο της ευκλείδειας απόστασης).
- γ) έχοντας δημιουργήσει τις 2 ομάδες δημιουργώ τον νέο αντιπρόσωπο για την κάθε ομάδα υπολογίζοντας τον μέσο όρο των δεδομένων της ομάδας.
- δ) πηγαίνουμε στο βήμα β.

Επαναλαμβάνουμε αυτά τα βήματα ώσπου να σταματήσουν να αλλάζουν πλέον οι αντιπρόσωποι.

- α) Αποτελέσματα για την περίπτωση που τα δεδομένα μας έχουν 2 διαστάσεις :

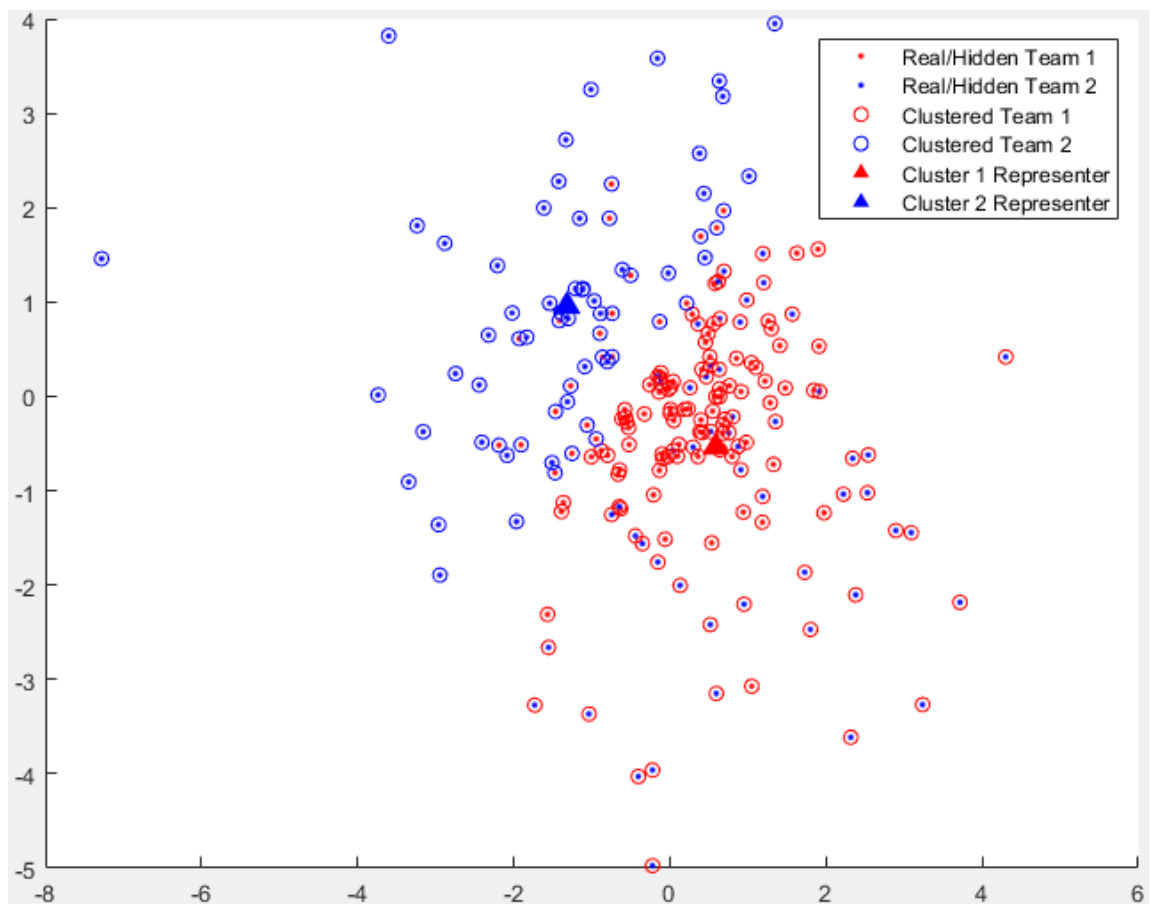
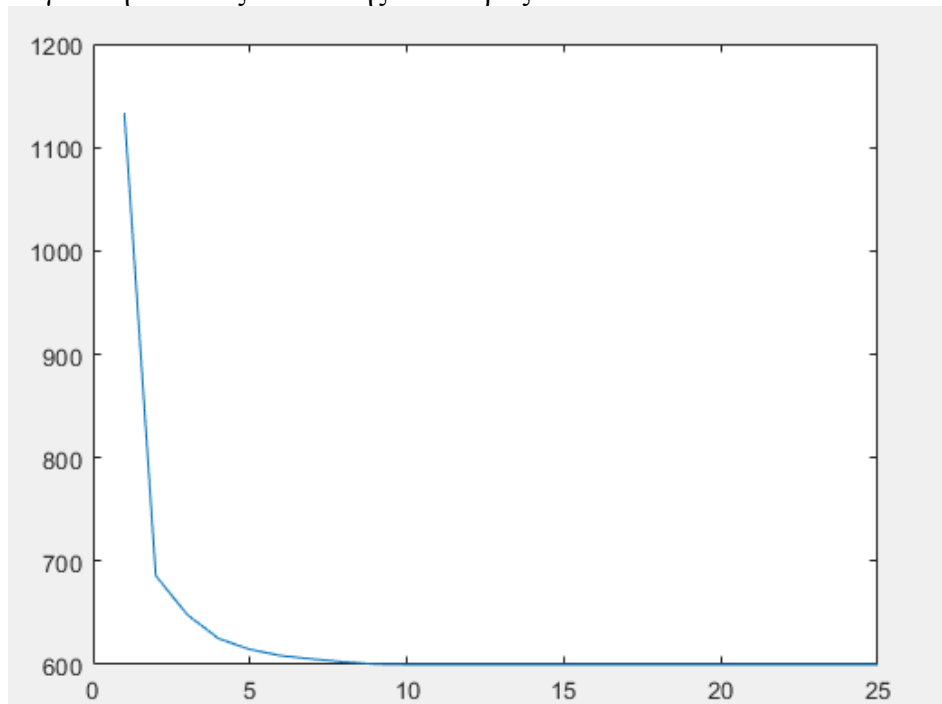
Σύγκλιση κόστους συνολικής διασποράς :



Με ποσοστό σφαλμάτων :

**Cluster\_Error** 0.4350

Για ένα διαφορετικό τρέξιμο του αλγορίθμου :  
Σύγκλιση κόστους συνολικής διασποράς :



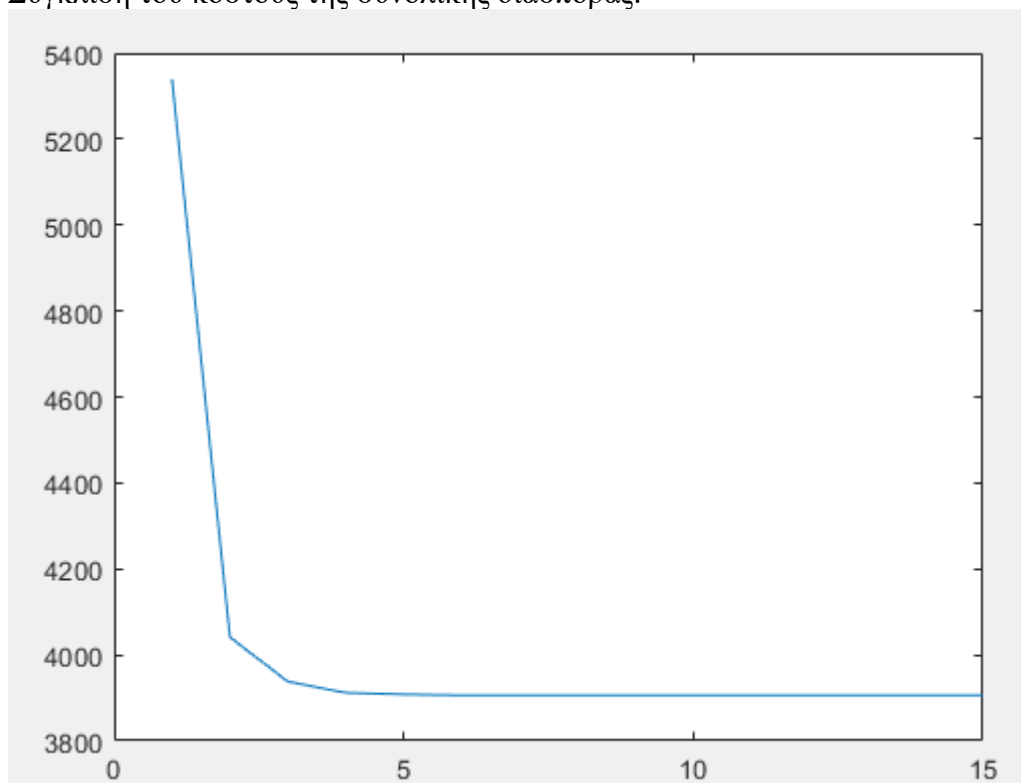
Με ποσοστό σφαλμάτων :

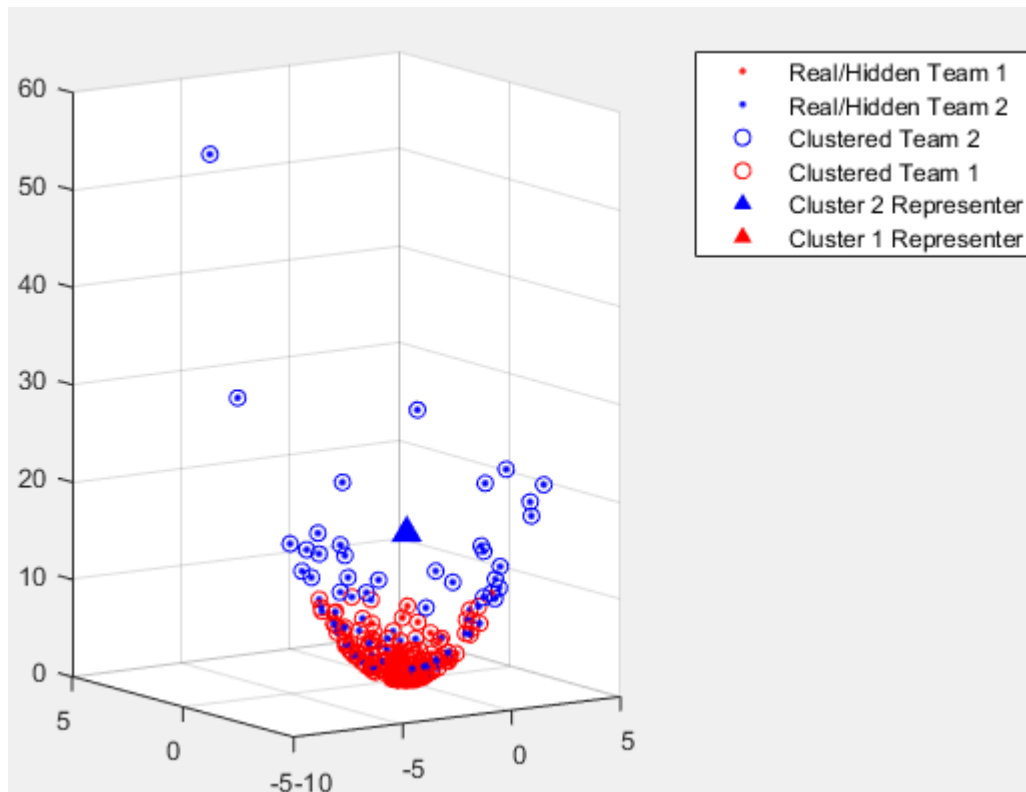
**Cluster\_Error**                      **0.3850**

Βλέπουμε πως ο αλγόριθμος είναι ευαίσθητος στην αρχικοποίηση των κέντρων  $Z$ .

Β) εδώ κάνουμε μικρή χρήση της μυστικής πληροφορίας και ορίζουμε ως τρίτη διάσταση στα δεδομένα μας την απόσταση τους από την αρχή των αξόνων.

Σύγκλιση του κόστους της συνολικής διασποράς:





Με ποσοστό σφαλμάτων :

**Cluster\_Error**                      **0.3500**

Η αποτελεσματικότητα του K-Means, όπως και άλλων τέτοιων παρόμοιων αλγορίθμων συσταδοποίησης, βασίζεται πάρα πολύ στην **εγγύτητα** των δεδομένων κάθε ομάδας. Όσο λιγότερο έντονο είναι το στοιχείο της εγγύτητας των δεδομένων μιας ομάδας τόσο λιγότερο αποτελεσματικοί είναι αυτοί οι αλγόριθμοι.

Για την διεκπεραίωση των προβλημάτων του 3<sup>ου</sup> σετ ασκήσεων χρησιμοποιήθηκε ο παρακάτω κώδικας σε Matlab :



## Πρόβλημα 1

```
function y = Kernel(x,h)
    y = 1/sqrt(2*pi*h) * exp((-1/(2*h))*x^2);
end

function y = pdf_estim(x,Y,N,h)
    y = zeros(1,length(x));
    for i = 1:length(x)
        for j = 1:N
            y(i) = y(i) + Kernel(x(i)-Y(j),h);
        end
        y(i) = y(i)/N;
    end
end
```

```
close all;
clear;
clc;
```

```
N = 1000;
Y = rand(N,1);
```

```
x = -8:0.01:8;
```

```
figure;
hold on;
for h = [0.1 0.01 0.001]
    plot(x,pdf_estim(x,Y,N,h));
end
hold off;
legend('h=0.1','h=0.01','h=0.001');
```

```
figure;
hold on;
for h = [0.0001 0.00001]
    plot(x,pdf_estim(x,Y,N,h));
end
hold off;
legend('h=0.0001','h=0.00001');
```

## Πρόβλημα 2

```
function y = Mercer(x,y,h)
    y = exp(-1/h * ( (x(1)-y(1))^2 + (x(2)-y(2))^2 ) );
end
```

```
close all;
clear;
clc;
load("data32.mat");
```

```

X = [stars;circles];

N = length(X);
N1 = length(stars);
N2 = length(circles);

h = 10; % kernel h parameter
l = 0.1; %lamda in equation

K = zeros(N);

for i = 1:N
    for j = 1:N
        K(i,j) = Mercer(X(i,:),X(j,:),h);
    end
end

I = eye(42);
y1 = ones(N1,1);
y2 = -ones(N2,1);
Y = [y1;y2];

A = (K + I*l);
B = Y;
C = mldivide(A,B);

                                % Plotting of border
Xmin = min(X(:,1));
Xmax = max(X(:,1));
Ymin = min(X(:,2));
Ymax = max(X(:,2));

x1 = -1.105: 0.005 : 1.09;
y1tst = Ymin : 0.005 : Ymax;

% finding of y1 in y1tst that Phi(x) = 0 (border)
y1 = zeros(1,length(x1));

for i = 1:length(x1)
    smallestKm=inf;
    for j = 1:length(y1tst)
        Km = zeros(1,N);
        for k = 1:N
            Km(k) = Mercer([x1(i) y1tst(j)],X(k,:),h);
        end
        if smallestKm > abs(Km*C)
            smallestKm = abs(Km*C);
            y1(i) = y1tst(j);
        end
    end
end

figure;
scatter(circles(:,1),circles(:,2),'r','o');
hold on;
scatter(stars(:,1),stars(:,2),'b','*');
plot(x1,y1);
hold off;

```

### Πρόβλημα 3.α

```
close all;
clear;
clc;
load("data33.mat");

% X = [X(2,:);X(1,:)];

% figure;
% title("mystery information");
% hold on;
% scatter(X(1,1:100),X(2,1:100),'r','.');
% scatter(X(1,101:200),X(2,101:200),'b','.');
% hold off;
%
% figure;
% title("unclusterd");
%
% scatter(X(1,:),X(2:,:), 'r','.');

Xmax = max(abs(X(1,:)));
Ymax = max(abs(X(2,:)));

Z1 = [-Xmax + 2*Xmax*rand; -Ymax + 2*Ymax*rand];
Z2 = [-Xmax + 2*Xmax*rand; -Ymax + 2*Ymax*rand];

N=25; % number of times that we run K-Means
Cost = zeros(1,N);

for k = 1:N
    C1=[];
    C2=[];

    for i = 1:200

        dist1 = (X(1,i)-Z1(1,1))^2 + (X(2,i)-Z1(2,1))^2;
        dist2 = (X(1,i)-Z2(1,1))^2 + (X(2,i)-Z2(2,1))^2;

        if dist1 < dist2
            C1(:,end+1) = X(:,i);
        elseif dist1 > dist2
            C2(:,end+1) = X(:,i);
        elseif rand>0
            C1(:,end+1) = X(:,i);
        else
            C2(:,end+1) = X(:,i);
        end
    end
end
```

```

N1 = length(C1);
N2 = length(C2);

for j = 1:N1
    Cost(k) = Cost(k) + (C1(1,j)-Z1(1,1))^2 + (C1(2,j)-Z1(2,1))^2;
end

    for j = 1:N2
        Cost(k) = Cost(k) + (C2(1,j)-Z2(1,1))^2 + (C2(2,j)-Z2(2,1))^2;
    end

Z1 = zeros(2,1);
Z2 = zeros(2,1);

for i = 1:N1
    Z1 = Z1 + C1(:,i);
end

for i = 1:N2
    Z2 = Z2 + C2(:,i);
end

Z1 = Z1./N1;
Z2 = Z2./N2;

end
figure;
plot(Cost);

count1=0;
count2=0;
for i = 1:N1
    for j = 1:100
        if C1(:,i)==X(:,j)
            count1 = count1 +1;
        end
    end
end
for i = 1:N2
    for j = 101:200
        if C2(:,i)==X(:,j)
            count1 = count1 +1;
        end
    end
end

for i = 1:N2
    for j = 1:100
        if C2(:,i)==X(:,j)
            count2 = count2 +1;
        end
    end
end

for i = 1:N1
    for j = 101:200

```

```

        if C1(:,i)==X(:,j)
            count2 = count2 +1;
        end
    end
end

Cluster_Error = min([1-count1/200 1-count2/200]);

% PLOTTING
figure;
hold on;
% mystic information
scatter(X(1,1:100),X(2,1:100),'r','.', 'DisplayName', 'Real/Hidden Team 1');
scatter(X(1,101:200),X(2,101:200),'b','.', 'DisplayName', 'Real/Hidden Team 2');

% Kmeans clustering
if count2>count1
    scatter(C2(1,:),C2(2:,:), 'r', 'o', 'DisplayName', 'Clustered Team 1');
    scatter(C1(1,:),C1(2:,:), 'b', 'o', 'DisplayName', 'Clustered Team 2');

    scatter(Z2(1,1),Z2(2,1),100, 'r', '^', 'filled', 'DisplayName', 'Cluster 1 Representative');
    scatter(Z1(1,1),Z1(2,1),100, 'b', '^', 'filled', 'DisplayName', 'Cluster 2 Representative');
else
    scatter(C2(1,:),C2(2:,:), 'b', 'o', 'DisplayName', 'Clustered Team 2');
    scatter(C1(1,:),C1(2:,:), 'r', 'o', 'DisplayName', 'Clustered Team 1');

    scatter(Z2(1,1),Z2(2,1),100, 'b', '^', 'filled', 'DisplayName', 'Cluster 2 Representative');
    scatter(Z1(1,1),Z1(2,1),100, 'r', '^', 'filled', 'DisplayName', 'Cluster 1 Representative');
end
hold off;

legend;

```

### **Πρόβλημα 3.β**

```

close all;
clear;
clc;
load("data33.mat");

X= [X(1,:);X(2,:);X(1,:).^2 + X(2,:).^2];

% figure;
% title("mystery information");
%
% scatter3(X(1,1:100),X(2,1:100),X(3,1:100),'r','.');

```

```

% hold on;
% scatter3(X(1,101:200),X(2,101:200),X(3,101:200),'b','.');
% hold off;
%
% figure;
% title("unclusterd");
%
% scatter3(X(1,:),X(2,:),X(3,:),'r','.');

Xmax = max(abs(X(1,:)));
Ymax = max(abs(X(2,:)));
Zmax = max(X(3,:));
% Zmax = 10;

Z1 = [-Xmax + 2*Xmax*rand;-Ymax + 2*Ymax*rand;Zmax*rand];
Z2 = [-Xmax + 2*Xmax*rand;-Ymax + 2*Ymax*rand;Zmax*rand];

N=15; % number of times that we run K-Means
Cost = zeros(1,N);

for k = 1:N
    C1=[];
    C2=[];

    for i = 1:200

        dist1 = (X(1,i)-Z1(1,1))^2 + (X(2,i)-Z1(2,1))^2 + (X(3,i)-
Z1(3,1))^2;
        dist2 = (X(1,i)-Z2(1,1))^2 + (X(2,i)-Z2(2,1))^2 + (X(3,i)-
Z2(3,1))^2;

        if dist1 < dist2
            C1(:,end+1) = X(:,i);
        elseif dist1 > dist2
            C2(:,end+1) = X(:,i);
        elseif rand>0
            C1(:,end+1) = X(:,i);
        else
            C2(:,end+1) = X(:,i);
        end

    end

    N1 = length(C1(1,:));
    N2 = length(C2(1,:));

    for j = 1:N1
        Cost(k) = Cost(k) + (C1(1,j)-Z1(1,1))^2 + (C1(2,j)-Z1(2,1))^2 +
(C1(3,j)-Z1(3,1))^2 ;
    end

    for j = 1:N2
        Cost(k) = Cost(k) + (C2(1,j)-Z2(1,1))^2 + (C2(2,j)-Z2(2,1))^2 +
(C2(3,j)-Z2(3,1))^2;
    end

    Z1 = zeros(3,1);

```

```

Z2 = zeros(3,1);

for i = 1:N1
    Z1 = Z1 + C1(:,i);
end

for i = 1:N2
    Z2 = Z2 + C2(:,i);
end

Z1 = Z1./N1;
Z2 = Z2./N2;

end

figure;
plot(Cost);

count1=0;
count2=0;
for i = 1:N1
    for j = 1:100
        if C1(:,i)==X(:,j)
            count1 = count1 +1;
        end
    end
end
for i = 1:N2
    for j = 101:200
        if C2(:,i)==X(:,j)
            count1 = count1 +1;
        end
    end
end

for i = 1:N2
    for j = 1:100
        if C2(:,i)==X(:,j)
            count2 = count2 +1;
        end
    end
end

for i = 1:N1
    for j = 101:200
        if C1(:,i)==X(:,j)
            count2 = count2 +1;
        end
    end
end

Cluster_Error = min([1-count1/200 1-count2/200]);

% PLOTTING
figure;
% mystic information
scatter3(X(1,1:100),X(2,1:100),X(3,1:100),'r','.', 'DisplayName', 'Real/Hidden Team 1');
hold on;

```

```

scatter3(X(1,101:200),X(2,101:200),X(3,101:200),'b','.', 'DisplayName', 'Real
/Hidden Team 2');
hold off;
% Kmeans clustering
if count2>count1
    hold on;
    scatter3(C2(1,:),C2(2,:),C2(3:),'r','o', 'DisplayName', 'Clustered Team
1');
    hold off;
    hold on;
    scatter(C1(1,:),C1(2,:),C1(3:),'b','o', 'DisplayName', 'Clustered Team
2');
    hold off;
    hold on;

scatter3(Z2(1,1),Z2(2,1),Z2(3,1),100,'r','^', 'filled', 'DisplayName', 'Cluste
r 1 Representer');
    hold off;
    hold on;

scatter3(Z1(1,1),Z1(2,1),Z1(3,1),100,'b','^', 'filled', 'DisplayName', 'Cluste
r 2 Representer');
    hold off;

else
    hold on;
    scatter3(C2(1,:),C2(2,:),C2(3:),'b','o', 'DisplayName', 'Clustered Team
2');
    hold off;
    hold on;
    scatter3(C1(1,:),C1(2,:),C1(3:),'r','o', 'DisplayName', 'Clustered Team
1');
    hold off;
    hold on;

scatter3(Z2(1,1),Z2(2,1),Z2(3,1),100,'b','^', 'filled', 'DisplayName', 'Cluste
r 2 Representer');
    hold off;
    hold on;

scatter3(Z1(1,1),Z1(2,1),Z1(3,1),100,'r','^', 'filled', 'DisplayName', 'Cluste
r 1 Representer');
    hold off;
end
legend;

```