

```

1 function clust = kmeans(K,X);
2 % K: number of classes
3 % X: dataset matrix (num. of cases x dimensions)
4
5 [n,d] = size(X);
6 clust = ones(n,1);
7 for t = 1:n;
8     randomsample = randperm(K);
9     clust(t) = randomsample(1);
10 end
11 clust2 = clust;
12 g = max(clust);
13 term = 1;
14
15 while term ~= 0;
16     % Find centroids
17     centroids = [];
18     for c = 1:g;
19         index = find(clust == c);
20         if isempty(index) ~= 1;
21             centroids = [centroids; mean(X(index,:),1)];
22         end
23     end
24
25     [g,dim] = size(centroids);
26     dist = ones(n,1);
27     for s = 1:n;
28         x = X(s,:);
29         x = ones(g,1)*x;
30         d = (centroids - x).^2;
31         d = sqrt(sum(d'));
32         [m,index] = min(d);
33         clust2(s) = index;
34         dist(s) = m;
35     end
36
37     term = sum(clust ~= clust2);
38     clust = clust2;
39 end

```

שאלה פתוחה.

א. תארו במשפט או שניים את תנאי העצירה בקוד

ב. אם נרצה לשנות את תנאי העצירה לכך שהמרחק הממוצע השתנה בפחות מ-5%, באילו שורות נבצע שינויים?

a. בסעיף זה לא צריך לרשום את הקוד המלא שמיישם את השינוי. צריך להבהיר אם זקוקים למשתנה חדש, ואם כן אז להסביר **(לא יותר מ-4 שורות!)**:

i. מה משתנה זה מייצג?

ii. ציינו באופן ברור על גבי הקוד איפה מעדכנים אותו

iii. ציינו באופן ברור על גבי הקוד איפה משתמשים בערך שלו

ג. מה יקרה אם נשנה את שורה מספר 18 ל $c=1:K$ **(לא יותר מ-4 שורות!)?**

פתרון

א. האלגוריתם עוצר כאשר אין יותר שינויים בשיוך הנקודות לקלאסטרים.

ב.

(הקוד שעליו מסומן הפתרון מופיע למטה).
בשורה 37 נחליף את התנאי לכך שיבדוק אם המרחק לא השתנה ביותר מ-5%
לצורך כך, צריך לשמור את המרחק הקודם במשתנה, למשל `old_dist`
נעדכן את `old_dist` בתחילת הלולאה

ג

g זה מספר הקלאסטרים בשלב מסוים של האלגוריתם, K זה מספר הקלאסטרים המכסימלי.

אם הלולאה תמשיך עד K , אז נחפש גם קלאסטרים ריקים.

מכיוון שיש בדיקה לכך בשורה 20, זה לא ישפיע על התוצאה הסופית.

ההשפעה היחידה היא שהאלגוריתם יקח יותר זמן בגלל בדיקות מיותרות.

```

1 function clust = kmeans(K,X);
2 % K: number of classes
3 % X: dataset matrix (num. of cases x dimensions)
4
5 [n,d] = size(X);
6 clust = ones(n,1);
7 for t = 1:n;
8     randomsample = randperm(K);
9     clust(t) = randomsample(1);
10 end
11 clust2 = clust;
12 g = max(clust);
13 term = 1;
14
15 while term ~= 0;
16     % Find centroids
17     centroids = [];
18     for c = 1:g;
19         index = find(clust == c);
20         if isempty(index) ~= 1;
21             centroids = [centroids; mean(X(index,:),1)];
22         end
23     end
24
25     [g,dim] = size(centroids);
26     dist = ones(n,1);
27     for s = 1:n;
28         x = X(s,:);
29         x = ones(g,1)*x;
30         d = (centroids - x).^2;
31         d = sqrt(sum(d'));
32         [m,index] = min(d);
33         clust2(s) = index;
34         dist(s) = m;
35     end
36
37     term = sum(clust ~= clust2);
38     clust = clust2;
39 end

```

Handwritten notes for line 16: $1 \times K \rightarrow$
 $1 \rightarrow 2 \rightarrow 3 \rightarrow N$

Handwritten notes for line 37: $1 \times K \rightarrow$
 $1 \rightarrow 2 \rightarrow 3 \rightarrow N$