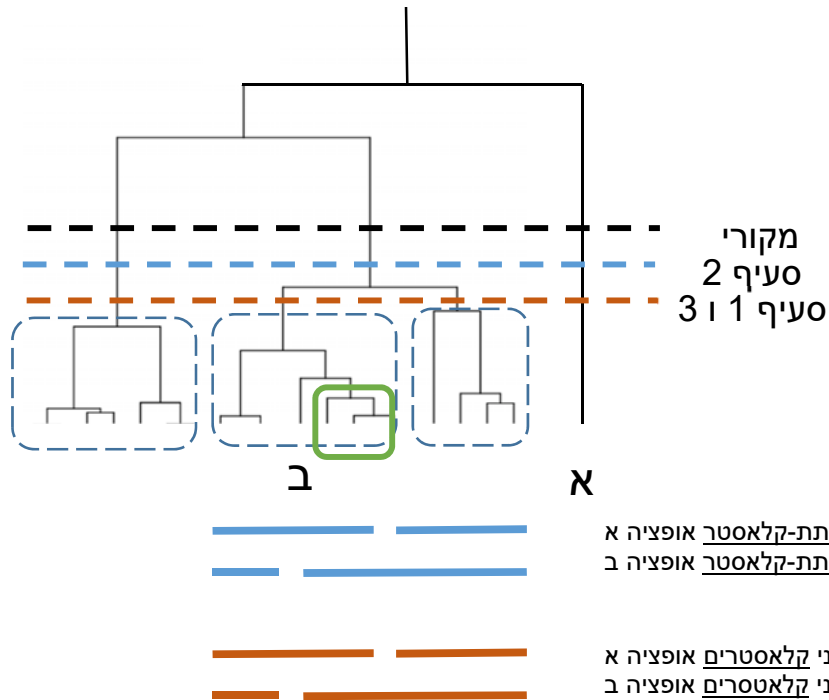


שאלה פתוחה 3:

נתונה דנדרוגרמה של תוצאות הפעלת Hierarchical clustering על נתונים עם centroid linkage ממנה הוגדרו שלושה קלאסטרים



1. בהינתן שנשתמש במרחק אוקלידי ו-centroid linkage אם נזרוק את דוגמא א' מהנתונים, ונשמור על שלושה קלאסטרים: – כיצד זה ישפיע על מבנה הדנדרוגרמה? כיצד זה ישפיע על החברות בקלאסטרים?
2. בהינתן שנשתמש במרחק אוקלידי ו-centroid linkage אם נזרוק את דוגמא ב' מהנתונים, ונשמור על שלושה קלאסטרים: – כיצד זה ישפיע על מבנה הדנדרוגרמה? כיצד זה ישפיע על החברות בקלאסטרים?
3. בהינתן שנשתמש במרחק אוקלידי ו-centroid linkage אם נזרוק את דוגמא א' ואת ב' מהנתונים ונשמור על שלושה קלאסטרים: – כיצד זה ישפיע על מבנה הדנדרוגרמה? כיצד זה ישפיע על החברות בקלאסטרים?

תשובות:

1.
 - א. מבנה הדנדרוגרמה ישאר זהה, רק ללא דוגמא א' הקיצונית.
 - ב. הקלאסטרים ישתנו ולכן חברות בקלאסטר תשתנה: הקלאסטר המרכזי יפוצל לשתי קבוצות (קו כחול מקווקו בשרטוט)
2.
 - א. עפ"י גובה החיבורים, שליפה של דוגמא ב' עלול לגרום לערבול מבנה הדנדרוגרמה בתוך הקלאסטר, או להשאירה זהה, כתלות מהשפעת דוגמא ב' על ה-centroid בתת-הקלאסטר (מוקף בירוק) ומרחקו מתת-הקבוצות הבאות. אם תת-הקלאסטר יזרק המבנה ישאר יציב.
 - ב. החברות בקלאסטרים תישאר יציבה, אך עלולה להשתנות בתת-הקלאסטר (שתי אופציות משורטטות בכחול)
3.
 - א. מבנה הדנדרוגרמה ישתנה בדיוק כמו בסעיף 2
 - ב. הקלאסטרים ישתנו ולכן חברות בקלאסטר תשתנה (שתי אופציות משורטטות בכתום)

```
function clust = kmeans(K,X);
% k-means spike sorting
% K: number of classes
% X: dataset matrix (num. of cases x dimensions)
% cluster = kmeans(K,X);
```

```
[n,d] = size(X);
clust = ones(n,1);
for t = 1:n;
    randomsample = randperm(K);
    clust(t) = randomsample(1);
end
clust2 = clust;
g = max(clust);
term = 1; ← 1 א', נאתחל ב-0
```

3 מלא- נוסף לולאה חיצונית שתריץ את האלגוריתם מספר פעמים ותשמור את ההרצה בה סכום המרחקים בתוך הקלאסטרינג הוא מינימלי (הסכימה במקרה זה תהיה מה-centroid ולא בין כל זוג). מחוץ לולאה נאתחל בערך גדול (למשל סך מרחקים) את minDist וניצור משתנה bestClust

1.ב. נאתחל משתנה חדש פה

```
while term ~= 0; ← 1 א', נשנה תנאי עצירה
    % Find centroids
    centroids = [];
    for c = 1:g;
        index = find(clust == c);
        if isempty(index) ~= 1;
            centroids = [centroids; mean(X(index,:))];
        end
    end
end
```

1.ב. נוסף תנאי עצירה שבודק עם המשתנה הגיע ל-10

```
[g,dim] = size(centroids);
dist = ones(n,1);
for s = 1:n;
    x = X(s,:);
    x = ones(g,1)*x;
    d = (centroids - x).^2;
    d = sqrt(sum(d'));
    [m,index] = min(d);
    clust2(s) = index;
    dist(s) = m;
end
```

3. נוסף לולאה על מספר ה-קלאסטרים, שסוכמת את מרחק השמור ב-dist בתוך כל הקלאסטרים, ושומרת אותו במשתנה חדש minDist

1 א', פה נעדכן את TERM

```
term = sum(clust ~= clust2);
clust = clust2;
end
```

1.ב. נעדכן את המשתנה שסיימו עוד איטרציה

3. נבדוק אם minDist קטן מה-minDist הקודם, אם כן נשמור את ה-clustering שבצענו עכשיו בתוך משתנה חדש bestClust

1. אם נרצה לשנות את תנאי העצירה כך שנוגבל ל-10 איטרציות לפני עצירה, באילו שורות נבצע שינויים
2. מה יהיו ההשלכות אם בשורה 42, נרשום בטעות `clust=clust;`
3. נרצה להימנע מ-מינימום לוקאלי ולמצוא את הקלאסטרינג הטוב ביותר. איזה משתנה נצטרך להוסיף לקוד (כלומר איזה מידע צריך לשמור), ובאילו שורות תבצע שינויים על מנת לשמרו

תשובות:

1. קיימות שתי אופציות:
 - א. נחליף את תנאי העצירה של האלגוריתם כך ש-term סופר את מספר האיטרציות – נדרש לשנות את שורת האתחול, בדיקת התנאי והעדכון, כמסומן בקוד (תחת הכותרת אופציה 1.א) אופציה זו פחות מוצלחת כיוון שהאלגוריתם ירוץ לחינם במידה ויתכנס קודם.
 - ב. נוסיף משתנה נוסף אותו נאתחל מחוץ ללולאה הראשית, הסופר את מספר האיטרציות ותנאי עצירה הבודק את ערכו לאחר כל איטרציה. ראה סימון בקוד 1.ב.
2. השינוי יגרום לכך שנשאר עם חברות הקלאסטרים הראשונית והאלגוריתם לא יתכנס.
3. נדרש ללולאה חיצונית נוספת המבצעת הגרלת נתוני איתחול חדשים לכל הרצה, ושומרת את ההרצה סך המרחק בתוך קלאסטרים הוא הקטן ביותר. לשם כך נצטרך משתנה שישמור את המרחק המינימלי והן משתנה שישמור את סידור הקלאסטרינג הנבחר.