machine learning

Unsupervised learning: K-Means and PCA

Exercise VII

פיתוח:
ד"ר יהונתן שלר
משה פרידמן

קרדיט - ד"ר יונתן רובין

למידה מונחית מול למידה לא מונחית

- למידה מונחית – לאלגוריתם יש מטרה ברורה: לחזות פלט רצוי, בהנתן קלט מסוים. בשלב האימון נתונים דגימות של זוגות $\{(X^{(i)}, y^{i})\}$ ועל פיהם נבנה מודל החיזוי

- למידה לא מונחית – מטרת האלגוריתם ברורה פחות (אין פידבק ברור האם הפלט הנוצר הינו נכון). בשלב האימון נתונים דגימות של $\{x^{(i)}\}$ (האם ללא ה y שלהם)

סוגי בעיות בלמידה לא מונחית

Clustering: represent each input case using a prototype example (we will review k-means)

Dimensionality reduction: represent each input case using a small number of variables (we will review PCA - principal components analysis)

Density estimation: estimating the probability distribution over the data space

מוטיבציה –
חלוקת סטודנטים לקבוצות למידה בזמן הקורונה

- מכללת מדבר סהרה החליטה לחלק את הסטודנטים למתמטיקה ל-7 קבוצות למידה. הקבוצות צריכות להיות יחסית הומוגניות.
  - האתגר שלנו – למצוא 7 קבוצות
  - הבעיה: אין לנו את ה- class label של כל קבוצה
- נמדוד הומוגניות ע"י דמיון בין הסטודנטים.
- אבל איך נמדוד המוגניות? לפי גיל? לפי צבע בגדים? לפי תחומי עניין? לפי רמת לימודים?

Clustering "אישכול"

- Cluster Analysis היא הפעולה של חלוקת קבוצה לתתי קבוצות ("אשכולות"/Clusters) כך ש:
  - אובייקטים באותו אשכול "דומים" זה לזה
  - אובייקטים באשכולות שונים, אינם "דומים" זה לזה.

- **Unsupervised**

חלק א' – תרגול והסבר k-means

- אלגוריתם k-means
- Scaling
- Distance and proximity
- אתחול ה-centroids ב-k-means – Forgy method
- כלל עצירה ו/או התכנסות
- שלבי הביניים

K-means - שיטה המבוססות חלוקה
יצוג ה-cluster ע"י אב-טיפוס (protoype)

<u>יצוג ע"י אב-טיפוס (prototype)</u> – לכל cluster יש אב-טיפוס שמייצג את הוקטורים ששייכים לאותו cluster.

- אינטואיציה גאומטרית: ה"נקודות" (וקטורים) ב-cluster, קרובים ל"אב-טיפוס" (prototype) מרכזי.
- ובשאיפה כל "נקודה" רחוקה משאר ה- prototypes.

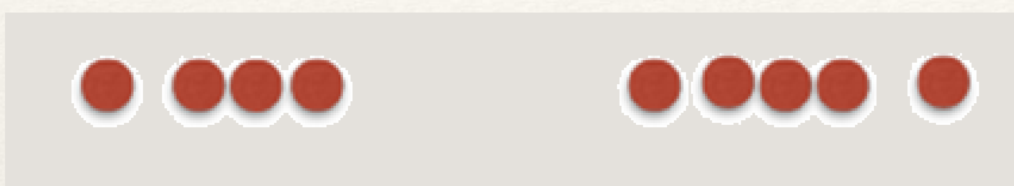
מטרה: מצא אוסף של אבות-טיפוס (prototypes)

- Cluster מס j – יכיל את הנקודות שהכי קרובות ל-"אב-טיפוס" j.

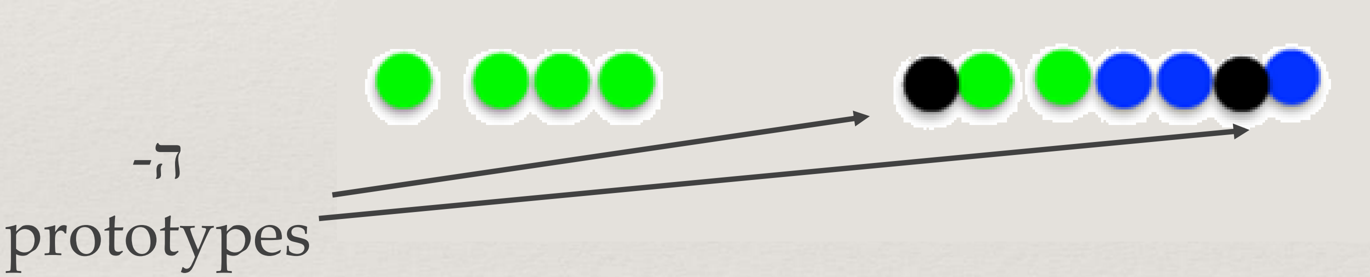
K-Means –
מציאת prototypes טובים ושיוך נכון של הנקודות

<u>המטרה</u>: לייצר
prototypes טובים,
כך שה"נקודות"
(וקטורים) ב-cluster,
קרובים ל"אב-טיפוס"
μ_j ככל האפשר

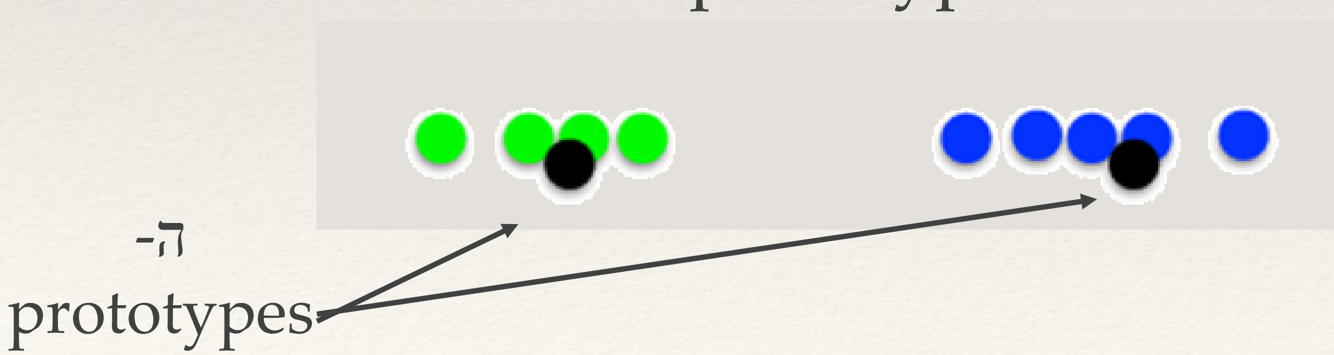
ה-data



דוגמה ל- prototypes <u>שאינם</u> איכותיים



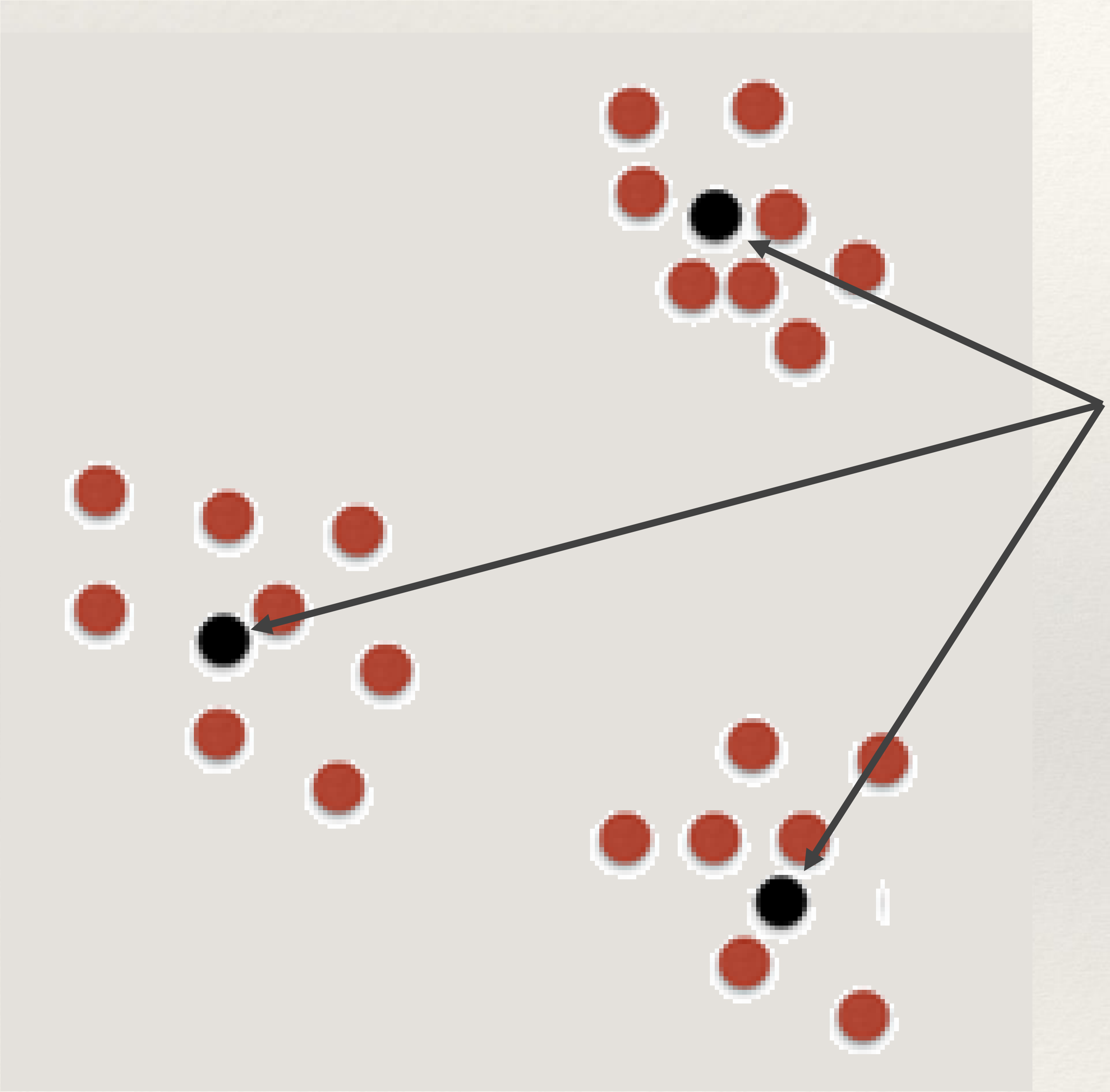
דוגמה ל- prototypes איכותיים



– K-Means
מציאת prototypes טובים ושיוך נכון של הנקודות - הרעיון

- נניח שמספר ה-clusters הוא k
- הנחה של prototype אחד עבור כל cluster
- נסמן את ה-prototypes ע"י $\mu_1, \dots, \mu_k$ (כזכור μ מייצג תוחלת).
- לעיתים מסמנים את ה-prototype כ-m (המסמן mean – ממוצע), או ע"י c (המסמן center – מרכז).

<u>המטרה</u>: לייצר prototypes טובים, כך שה"נקודות" (וקטורים) ב-cluster, קרובים ל"אב-טיפוס" μ_j ככל האפשר



K-means –
מה הם ה-prototypes
המשמשים במרכזים?

K-means –
מה הם ה-prototypes המשמשים במרכזים?

נתון מדגם של n דוגמאות:

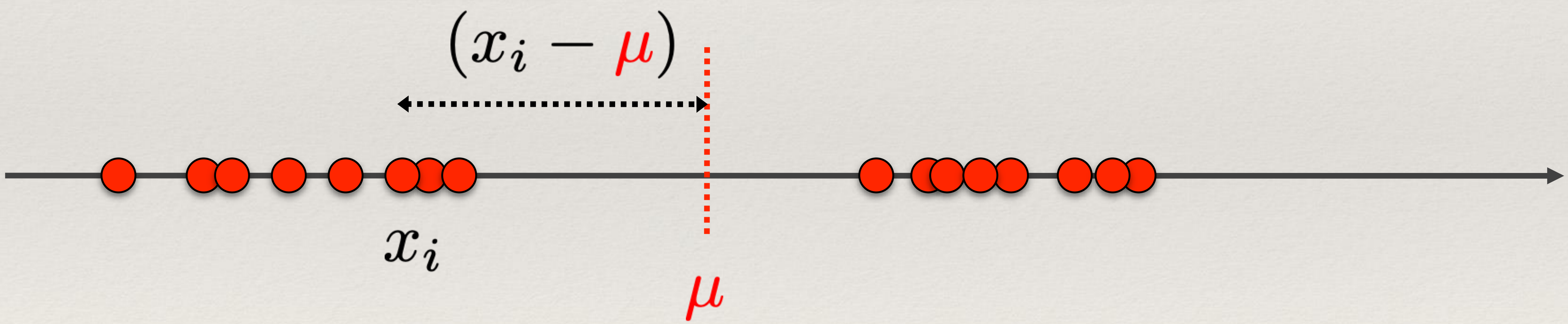
$$x_1, x_2, , \ldots, x_n \quad ; \quad x_i \in \mathbb{R}$$

ממוצע המדגם

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

שונות
(מדד לפיזור)

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2$$



K-means –
מה הם ה-prototypes המשמשים במרכזים?

מה קורה אם ניקח 2 clusters C_1, C_2?

אינטואיציה גאומטרית – 2 clusters נראים יותר מתאימים מ-cluster אחד.

$$\mu_1 = \frac{1}{n_1} \sum_{i \in C_1} x_i$$

ה-prototypes

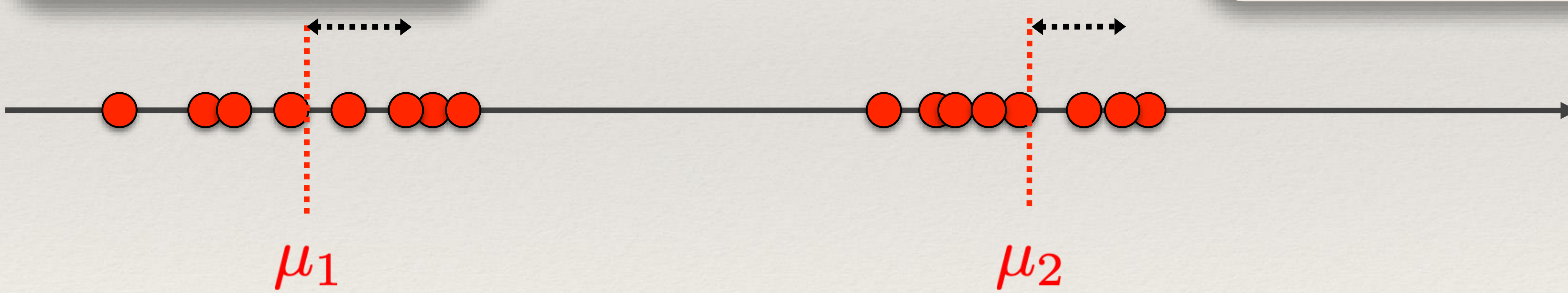
$$\mu_2 = \frac{1}{n_2} \sum_{i \in C_2} x_i$$

$$\sigma_1^2 = \frac{1}{n_1} \sum_{i=1}^{n_1} (x_i - \mu_1)^2$$

הפיזור ב-clusters

המטרה: פיזור מינימלי

$$\sigma_2^2 = \frac{1}{n_2} \sum_{i=1}^{n_1} (x_i - \mu_2)^2$$



שאלות ביניים – שאלה 1

1. איך נחשב את ה-prototype לכל cluster ב-kmeans ואיך נדע שה-cluster אייכותי ביחס לווקטרים השייכים אליו?

<u>תשובות אפשרויות</u>:

א. מחשבים prototype ע"י שונות, ונדע שה-cluster אייכותי ע"י ממוצע וקטורי ושאיפה לממוצע מינימלי

ב. מחשבים prototype ע"י ממוצע וקטורי, ונדע שה-cluster אייכותי ע"י חישוב שונות ושאיפה לשונות מינימלית

<u>תשובה</u> – ב.

איך מגדירים דמיון?

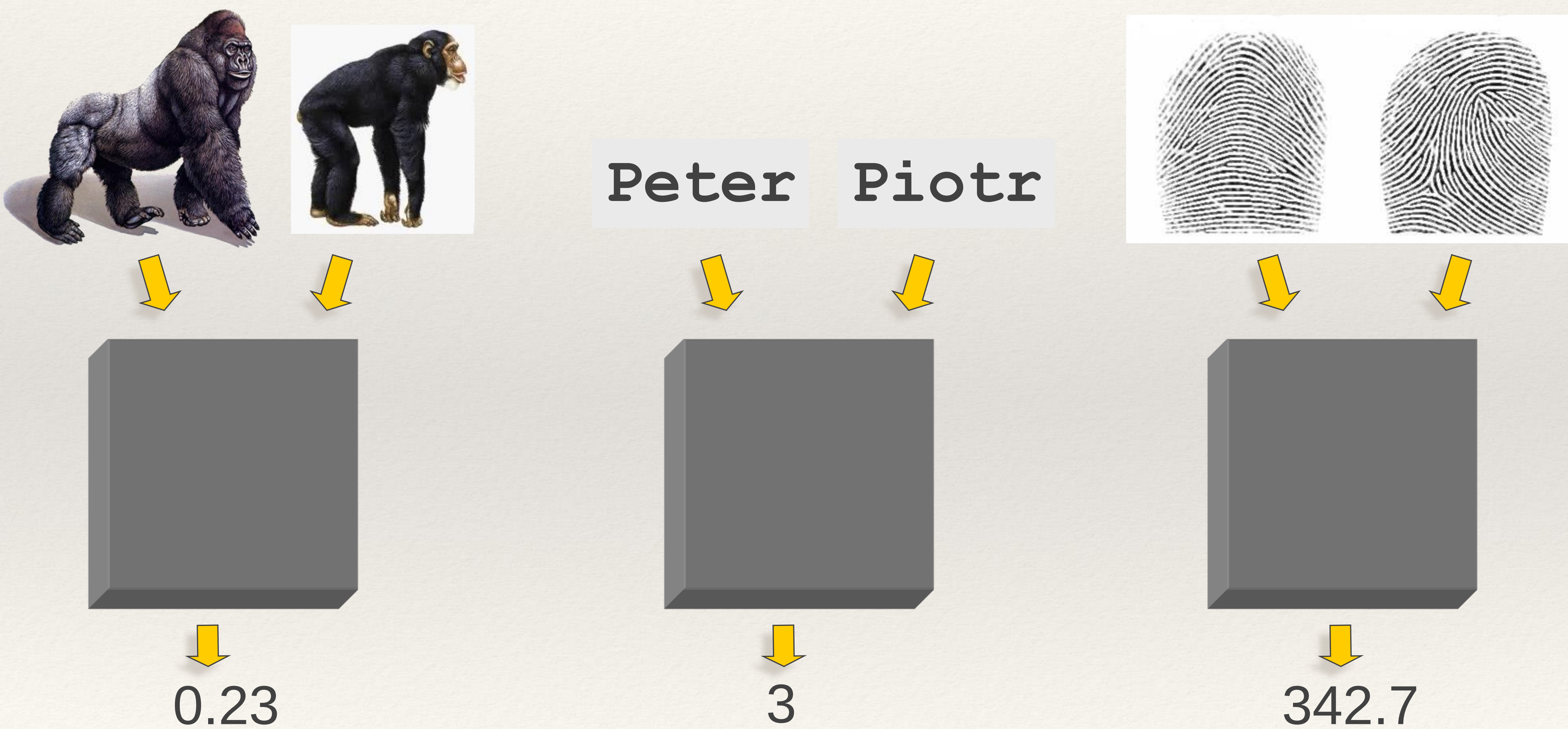
Similarity is hard to define, but… “*We know it when we see it*”

Credit: Eamonn Keogh

Defining Distance Measures

Slide from Eamonn Keogh

Definition: Let O_1 and O_2 be two objects from the universe of possible objects. The distance (dissimilarity) between O_1 and O_2 is a real number denoted by $D(O_1,O_2)$



A generic technique for measuring similarity

To measure the similarity between two objects, transform one into the other, and measure how much effort it took. The measure of effort becomes the distance measure.

The distance between Patty and Selma:

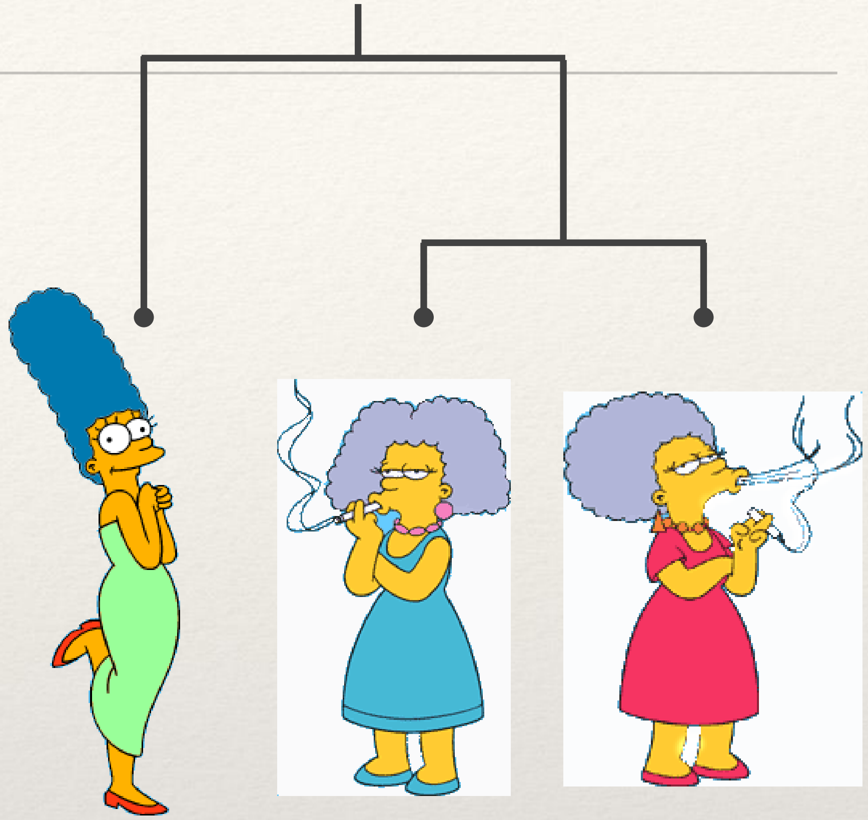
Change dress color, 1 point
Change earring shape, 1 point
Change hair part, 1 point

D(Patty,Selma) = **3**

The distance between Marge and Selma:

Change dress color, 1 point
Add earrings, 1 point
Decrease height, 1 point
Take up smoking, 1 point
Lose weight, 1 point

D(Marge,Selma) = **5**



This is called the “edit distance” or the “transformation distance”

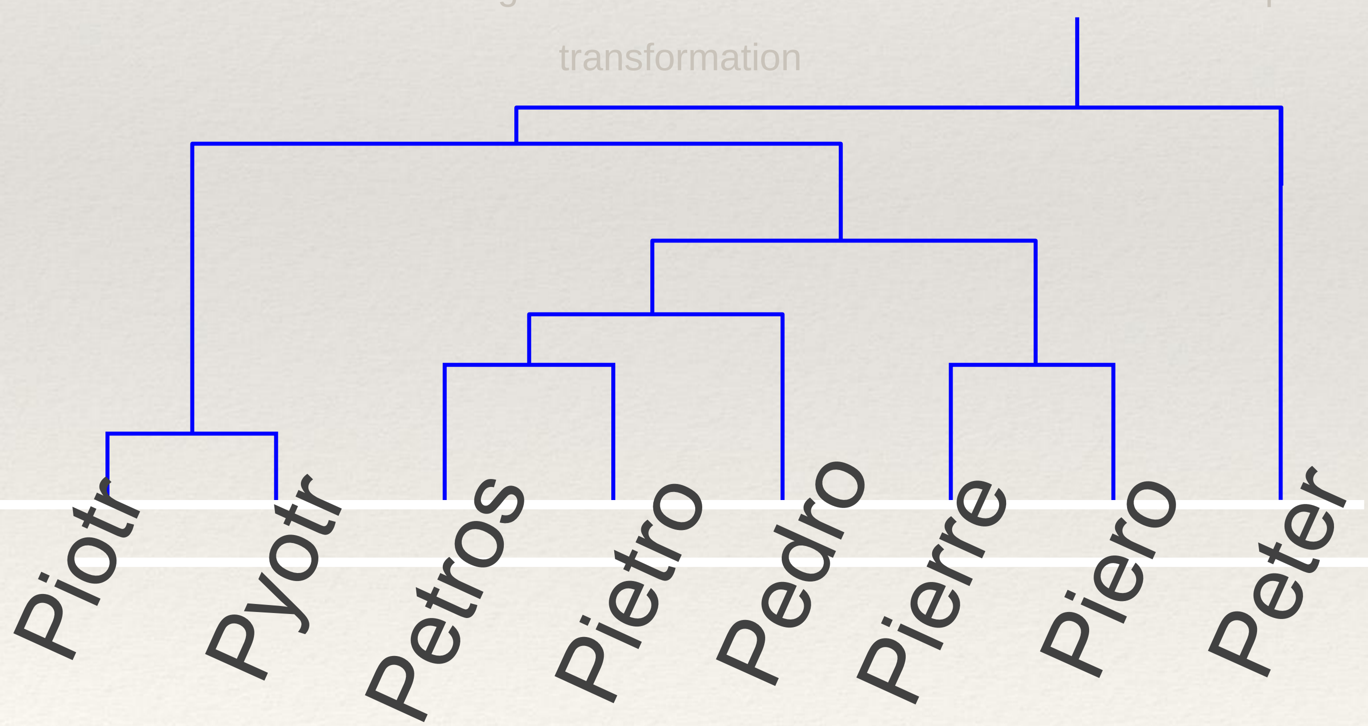
Slide based on one by Eamonn Keogh

Edit Distance Example

It is possible to transform any string *Q* into string *C*, using only *Substitution*, *Insertion* and *Deletion*.
Assume that each of these operators has a cost associated with it.

The similarity between two strings can be defined as the cost of the cheapest transformation from *Q* to *C*.

Note that for now we have ignored the issue of how we can find this cheapest transformation

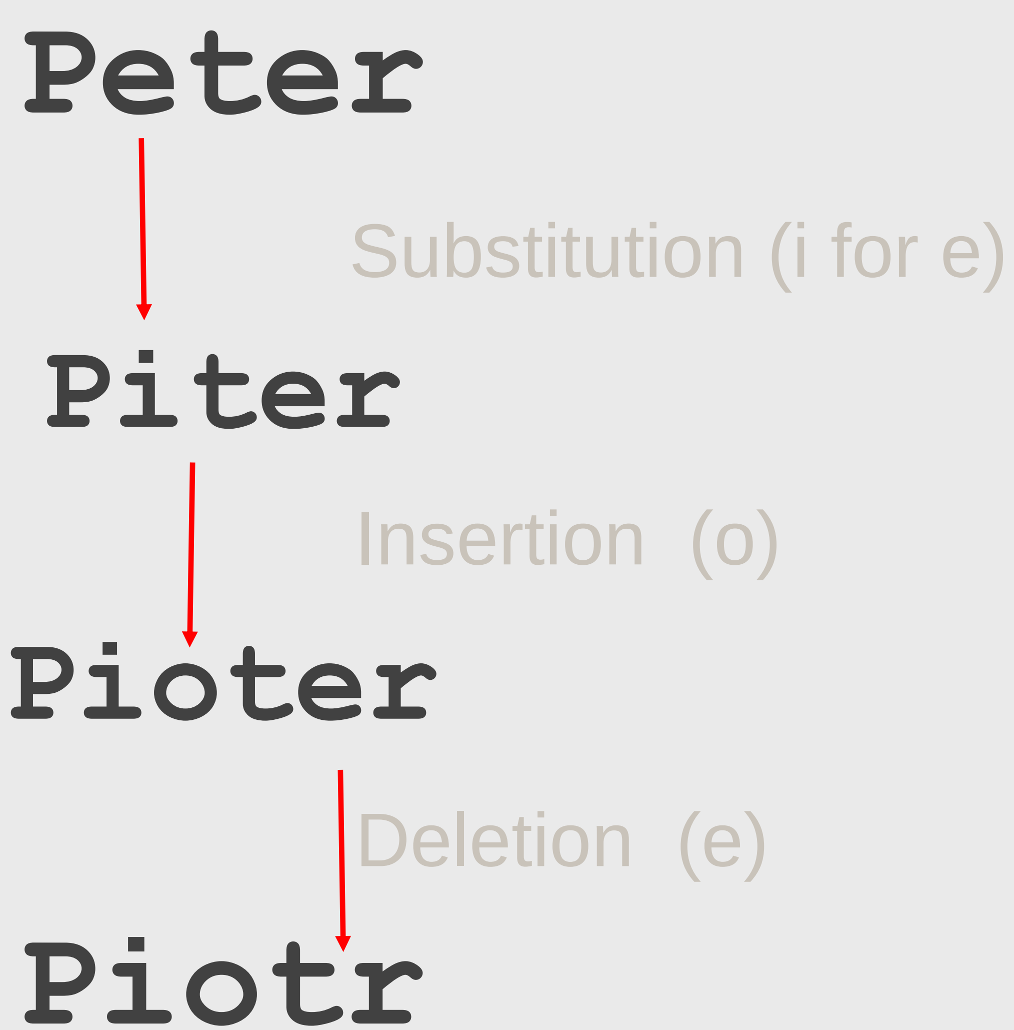


How similar are the names "Peter" and "Piotr"?

Assume the following cost function

Substitution	1 Unit
Insertion	1 Unit
Deletion	1 Unit

D(**Peter**, **Piotr**) is 3



Slide based on one by Eamonn Keogh

Cosine similarity measure

Cosine of the angle between two vectors (instances) gives a similarity function:

$$s(x, x') = \frac{x^t x'}{\|x\| \|x'\|}$$

When features are binary this becomes the number of attributes shared by x and x' divided by the geometric mean of the number of attributes in x and the number in x'.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

Cosine Similarity with L_2

תרגיל 1 – Cosine Similarity

Document Term Frequency – for each term we count the number of occurrences of the term in the document

Document	team	coach	hockey	baseball	soccer	penalty	score	win	loss	season
Doc1	5	0	3	0	2	0	0	2	0	0
Doc2	3	0	2	0	1	1	0	1	0	1
Doc3	0	7	0	2	1	0	0	3	0	0
Doc4	0	1	0	0	1	2	2	0	3	0

<u>פתרון</u> - תרגיל 1 – Cosine Similarity

- Denote the first two term-frequency vectors as $\vec{x}$, $\vec{y}$
- $\vec{x} = (5,0,3,0,2,0,0,2,0,0)$
- $\vec{y} = (3,0,2,0,1,1,0,1,0,1)$

$$\mathrm{Sim}(\vec{x}, \vec{y}) = \frac{\vec{x}^T \cdot \vec{y}}{\|\vec{x}\| \cdot \|\vec{y}\|}$$

<u>Exercise</u>: Calculate the cosine similarity.

- Assume normalization with L_2
- $\|\vec{x}\| = \sqrt{5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2} = 6.48$
- $\|\vec{y}\| = \sqrt{3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2} = 4.12$
- $\vec{x}^T \cdot \vec{y} = 5 \cdot 3 + 0 \cdot 0 + 3 \cdot 2 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot 1 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot 0 + 0 \cdot 1 = 25$

$$\mathrm{Sim}(\vec{x}, \vec{y}) = \frac{\vec{x}^T \cdot \vec{y}}{\|\vec{x}\| \cdot \|\vec{y}\|} = \frac{25}{6.48 \cdot 4.12} = 0.94$$

פונקציות מרחק נוספות - Minkowski Distance - תזכורת

מרחק מיניקובסקי: $\boldsymbol{d(\overrightarrow{x_j},\overrightarrow{x_i})}=\left(\sum_{m=1}^{n}|x_{j_m}-x_{i_m}|^p\right)^{\frac{1}{p}}$

מרחק מנהטן: $\boldsymbol{d\left(\overrightarrow{x_j},\overrightarrow{x_i}\right)}=\sum_{m=1}^{n}\left|x_{j_m}-x_{i_m}\right|=$

$$=\left|x_{j_m}-x_{i_m}\right|+\left|x_{j_m}-x_{i_m}\right|+\cdots+\left|x_{j_m}-x_{i_m}\right|$$

מרחק אוקלידי: $\boldsymbol{d(\overrightarrow{x_j},\overrightarrow{x_i})}=\sqrt{\sum_{m=1}^{n}(x_{j_m}-x_{i_m})^2}$

$$=\sqrt{(x_{j_1}-x_{i_1})^2+(x_{j_2}-x_{i_2})^2\ldots+(x_{j_n}-x_{i_n})^2}$$

<u>מרחק צ'בישב</u>: $d\left(\overrightarrow{x_j},\overrightarrow{x_i}\right)=\max_{1\le m\le d}\left|x_{j_m}-x_{i_m}\right|$

אלגוריתם K-means

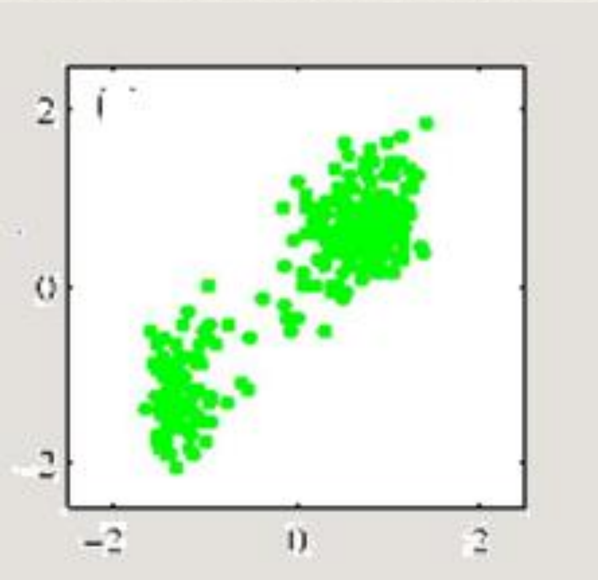
- נתון: אוסף ווקטורים ופרמטר K
- מצא חלוקה אופטימאלית שמחלקת ל-K אשכולות

- אלגוריתם:

1. "נחש" K מרכזים
2. שייך כל ווקטור ל"מרכז" הקרוב אליו
3. חשב מרכזים מחדש ע"י מציאת מרכז האשכול
4. חזור על צעדים 2-3 עד שאין יותר עדכונים (עד התכנסות או קיום תנאי עצירה)

אלגוריתם K-means - דוגמה

- נתון: אוסף ווקטורים ופרמטר K
- מצא חלוקה אופטימאלית שמחלקת ל-K אשכולות
    - נתון k=2 (2 clusters צריך למצוא 2 prototypes)

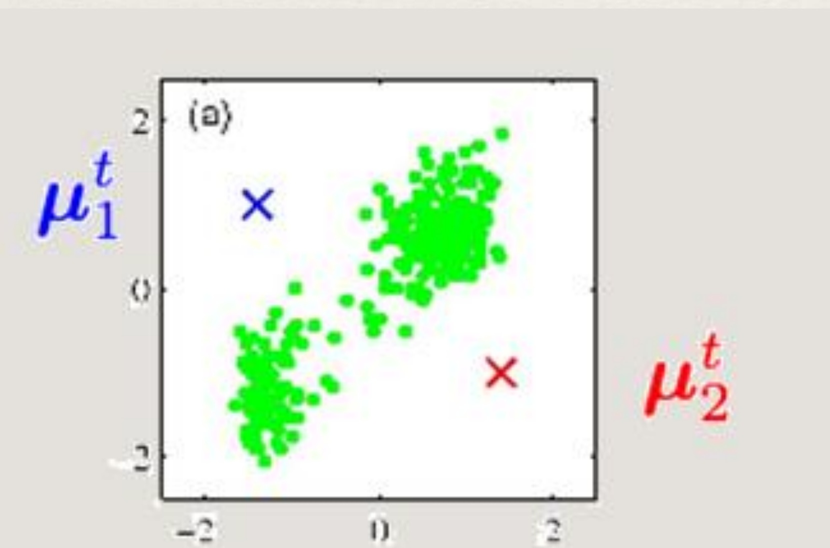


- אלגוריתם:

1. "נחש" K מרכזים
2. שייך כל ווקטור ל"מרכז" הקרוב אליו
3.
4. חזור על צעדים 2-3 עד שאין יותר עדכונים

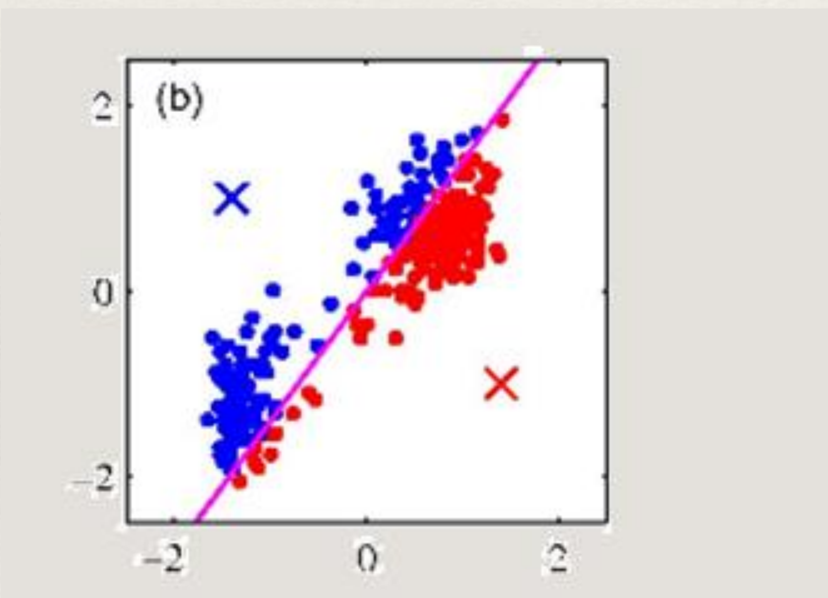
אלגוריתם K-means - דוגמה

- נתון: אוסף ווקטורים ופרמטר K
- מצא חלוקה אופטימאלית שמחלקת ל-K אשכולות
    - נתון k=2 (2 clusters צריך למצוא 2 prototypes)
- <u>אלגוריתם</u>:
    1. "נחש" K מרכזים
    2. שייך כל ווקטור ל"מרכז" הקרוב אליו
    3.
    4. חזור על צעדים 2-3 עד שאין יותר עדכונים



אלגוריתם K-means - דוגמה

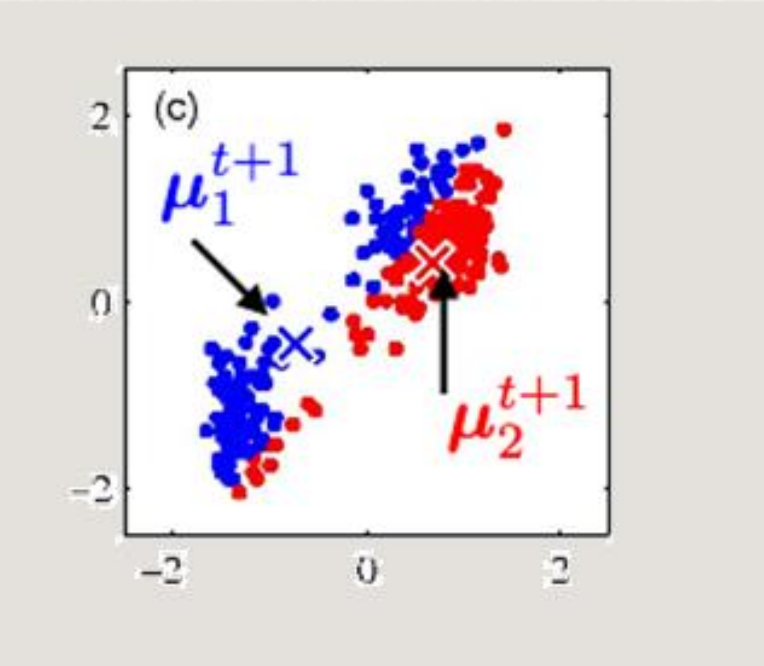
- נתון: אוסף ווקטורים ופרמטר K
- מצא חלוקה אופטימאלית שמחלקת ל-K אשכולות
    - נתון k=2 (2 clusters צריך למצוא 2 prototypes)
- <u>אלגוריתם</u>:



1. "נחש" K מרכזים
2. שייך כל ווקטור ל"מרכז" הקרוב אליו
3.
4. חזור על צעדים 2-3 עד שאין יותר עדכונים

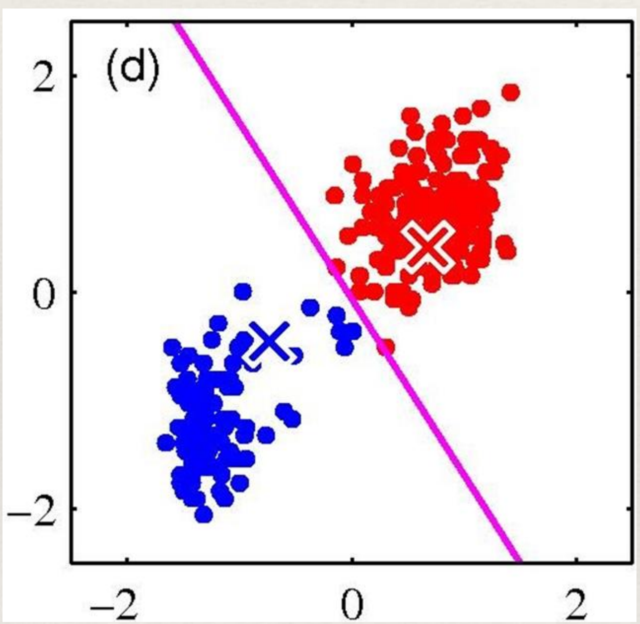
אלגוריתם K-means - דוגמה

- נתון: אוסף ווקטורים ופרמטר K
- מצא חלוקה אופטימאלית שמחלקת ל-K אשכולות
  - נתון k=2 (2 clusters צריך למצוא 2 prototypes)
- <u>אלגוריתם</u>:
  1. "נחש" K מרכזים
  2. שייך כל ווקטור ל"מרכז" הקרוב אליו
  3. חשב מרכזים מחדש ע"י מציאת מרכז האשכול
  4. חזור על צעדים 2-3 עד שאין יותר עדכונים



אלגוריתם K-means - דוגמה

- נתון: אוסף ווקטורים ופרמטר K
- מצא חלוקה אופטימאלית שמחלקת ל-K אשכולות
    - נתון k=2 (2 clusters צריך למצוא 2 prototypes)
- <u>אלגוריתם:</u>
    1. "נחש" K מרכזים
    2. שייך כל ווקטור ל"מרכז" הקרוב אליו
    3. חשב מרכזים מחדש ע"י מציאת מרכז האשכול
    4. חזור על צעדים 2-3 עד שאין יותר עדכונים (עד התכנסות או קיום תנאי עצירה)



K-means – שלב 1 - אתחול ה-centroids

עבור האתחול הבסיסי של אלגוריתם K-means (שלב 1 באלגוריתם) יש להגריל את המרכזים (ה-centroids) בצורה אקראית בהתפלגות אחידה

- באלגוריתם המקורי (Lloyd, 1957), כל נקודות בתחום ההגדרה (לפי המימדיות) הם מועמדים פוטנציאלים.
- Forgy method (Hamerly & Elkan, 2002) - בחירה אקראית של נקודות מתוך ה-dataset (ולא מתוך כל ערך אפשרי).
- אפשרות נוספת – לקחת למשל kmeans ולבצע את שלב 1 לפי שיטת Forgy, אך להריץ כך את k-means, כמה פעמים ולבחור את תוצר ה-clustering הטוב ביותר.
- בהמשך נלמד - kmeans++

K-means – שלב 4 – כלל עצירה ו/או התכנסות

- No (or minimum) re-assignments of data points to different clusters, *or*
- No (or minimum) change of centroids, *or*
- minimum decrease in the **sum of squared error** (SSE),

$$\text{WSS} = \sum_{j=1}^{k} \sum_{\widehat{y_i}=j} d(x_i, \mu_j)^2 = \sum_{j=1}^{k} \sum_{i=1}^{n} r_{i,j} \cdot ||x_i - \mu_j||^2$$

distance between a vector to its centroid

Cluster j

Centroid of x_i

$$\boxed{r_{i,j} = \begin{cases} 1 & \hat{y}_i = j \\ 0 & \hat{y}_i \neq j \end{cases}}$$

- To deal with complex cases, we usually also add a maximum number of iterations

שאלות ביניים – שאלה 2

2. באיזו שיטה מהשיטות נשתמש ע"מ לוודא עצירת k-means?

<u>תשובות אפשרויות</u>:

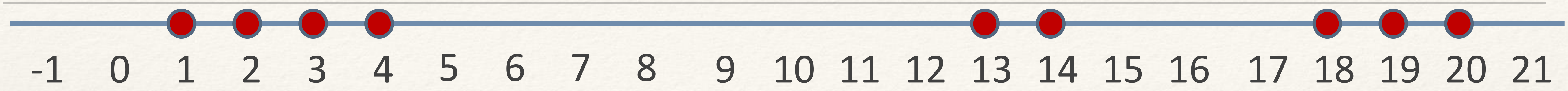
א. שינוי מזערי או אין שינוי בשיוך נקודות למרכזים
ב. שינוי מזערי ב-SSE בתוך ה- cluster (כלומר WSS)
ג. כמות מקסימלית של איטרציות
ד. כל התשובות נכונות

<u>תשובה</u> – ד.
הערה – ניתן לבדוק חוסר שינוי במרכזים, שקול לתשובה א., מדוע?

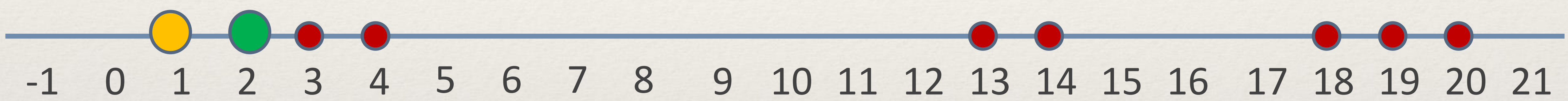
K-means – תרגיל 2
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

- נתונות הנקודות הבאות:
  - 1,2,3,4,13,14,18,19,20
- הרץ את אלגוריתם k-means על נקודות אלו.
  - הנח ש-k=2

K-means – תרגיל 2 - פתרון
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית



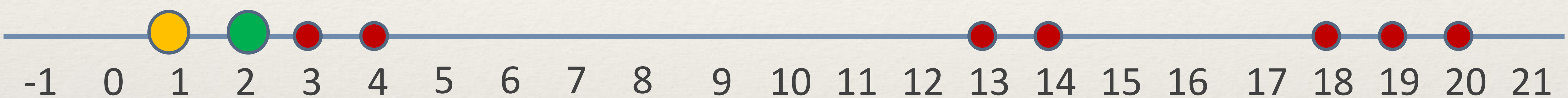
K-means שלב 1 (Forgy method) - ננחש 2 "מרכזים" ראשונים 1,2



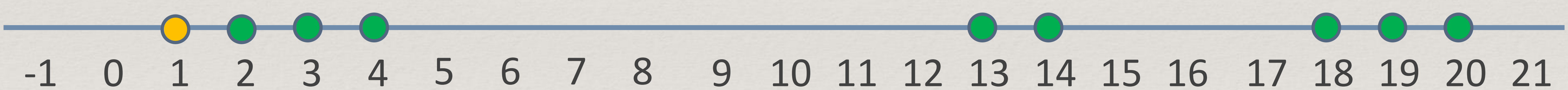
בחירה מאוד לא מושכלת

K-means – תרגיל 2 - <u>פתרון</u>
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

K-means שלב 1 (Forgy method) - ננחש 2 "מרכזים" ראשונים 1,2

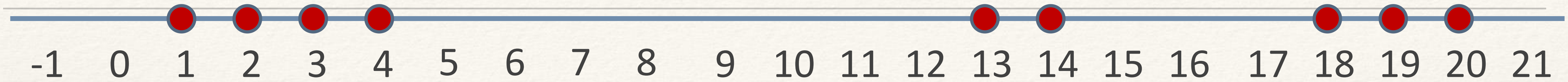


<u>איטרציה 1</u>: **K-means שלב 2** - נשייך כעת את הנקודות ל"מרכזים"(עפ"י מרחק אוקלידי)

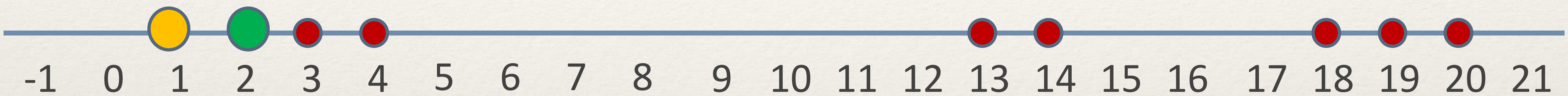


- 4 קרוב יותר (אוקלידית) ל-2 מאשר ל-1
- ...
- 14 יותר קרוב (אוקלידית) ל-2 מאשר ל-1

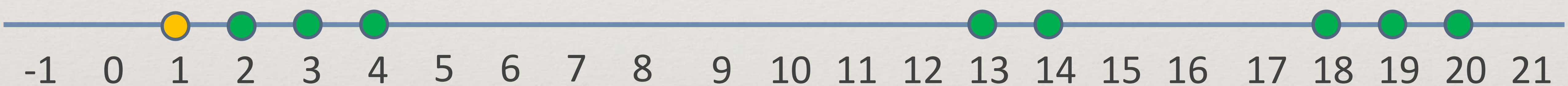
K-means – תרגיל 2 - <u>פתרון</u>
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית



K-means שלב 1 (Forgy method) - ננחש 2 "מרכזים" ראשונים 1,2

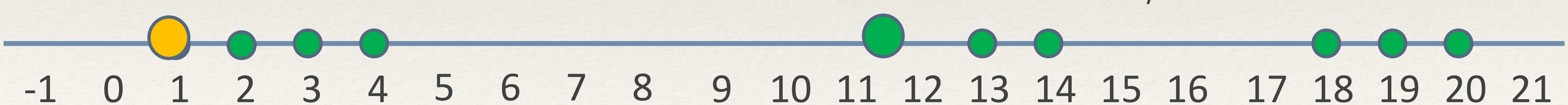


<u>איטרציה 1</u>: **K-means שלב 2** - נשייך כעת את הנקודות ל"מרכזים"(עפ"י מרחק אוקלידי)



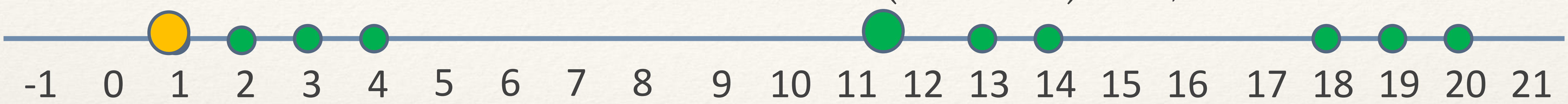
<u>איטרציה 1</u>: **K-means שלב 3** - נעדכן "מרכזים":

- מרכז ירוק: (2+3+4+13+14+18+19+20)/8=11.6
- מרכז כתום: 1/1=1

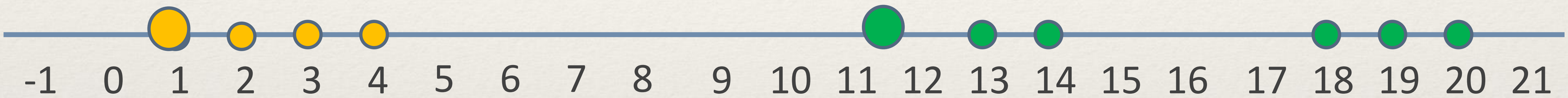


K-means – תרגיל 2 - <u>פתרון</u>
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

התוצאה מהאיטרציה הקודמת (איטרציה 1):

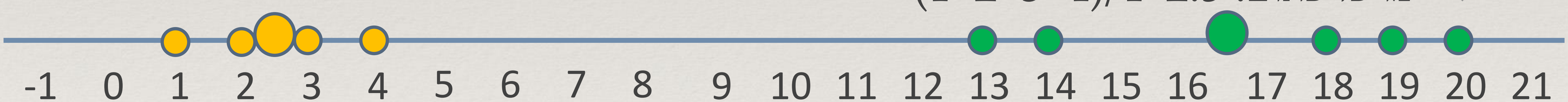


<u>איטרציה 2</u>: **K-means שלב 2** - נשייך כעת את הנקודות ל"מרכזים"



<u>איטרציה 2</u>: **K-means שלב 3** - נעדכן "מרכזים":

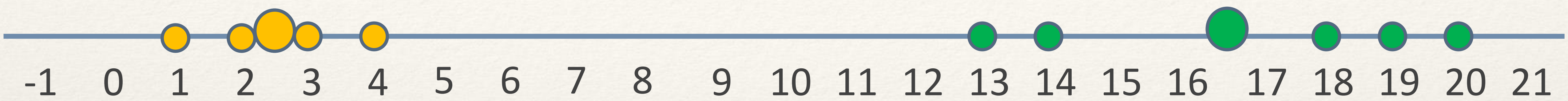
- מרכז ירוק: $(13+14+18+19+20)/5=16.8$
- מרכז כתום: $(1+2+3+4)/4=2.5$



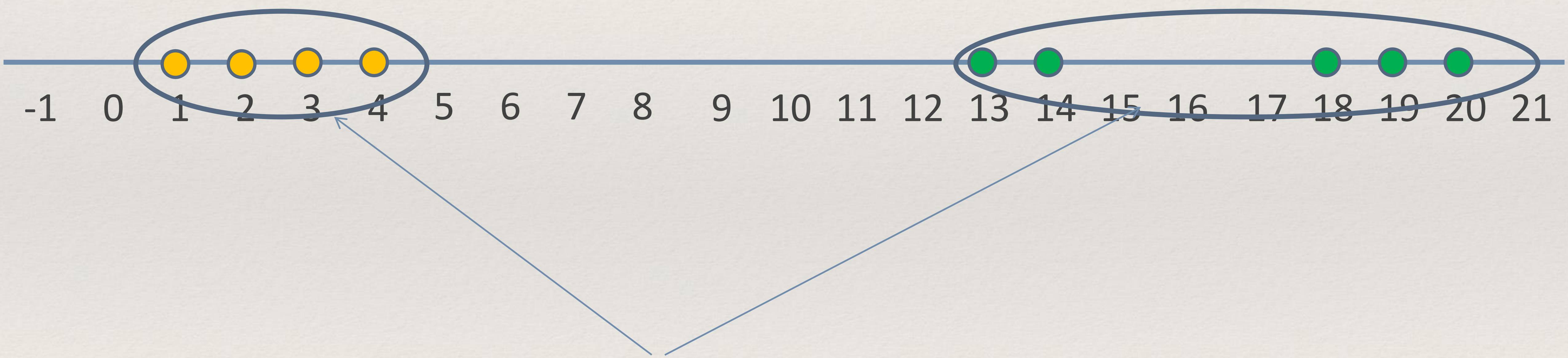
<u>איטרציה 3</u>: **K-means שלב 2** - נשייך כעת את הנקודות ל"מרכזים"

K-means – תרגיל 2 - <u>פתרון</u>
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

<u>איטרציה 3</u>: **K-means שלב 3** - נעדכן "מרכזים"



<u>איטרציה 3</u>: **K-means שלב 4** – אין עדכונים ולכן האלגוריתם עוצר



אלו 2 ה"אשכולות" שנוצרו

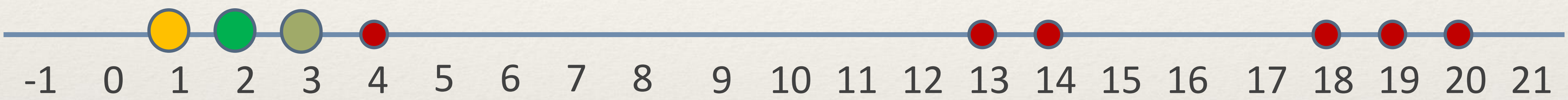
K-means – תרגיל 3 – אותם נתונים עם K=3
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

- נתונות הנקודות הבאות:
  - 1,2,3,4,13,14,18,19,20
- הרץ את אלגוריתם k-means על נקודות אלו.
  - הנח ש-k=3

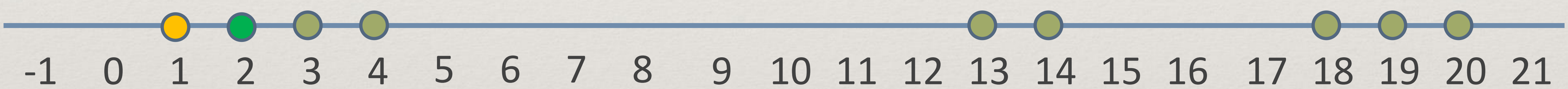
K-means – תרגיל 3 – אותם נתונים עם K=3 - <u>פתרון</u>
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

K-means שלב 1 (Forgy method) - ננחש 2 "מרכזים" ראשונים 1,2,3

- שוב בחירה לא מוצלחת



<u>איטרציה 1</u>: **K-means שלב 2** - נשייך כעת את הנקודות ל"מרכזים"(עפ"י מרחק אוקלידי)

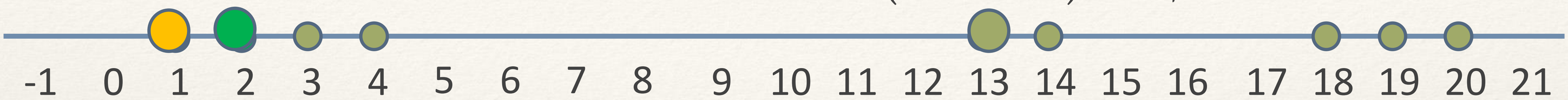


<u>איטרציה 1</u>: **K-means שלב 3** - נעדכן "מרכזים":

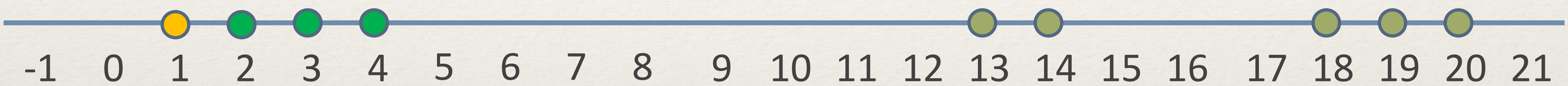
- מרכז סגול: (3+4+13+14+18+19+20)/7=13
- מרכז כתום: 1/1=1
- מרכז ירוק: 2/1=2

K-means – תרגיל 3 – אותם נתונים עם K=3
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

התוצאה מהאיטרציה הקודמת (איטרציה 1):



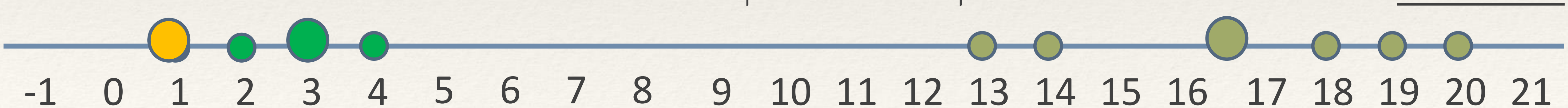
<u>איטרציה 2</u>: **K-means שלב 2** - נשייך כעת את הנקודות ל"מרכזים"



<u>איטרציה 2</u>: **K-means שלב 3** - נעדכן "מרכזים":

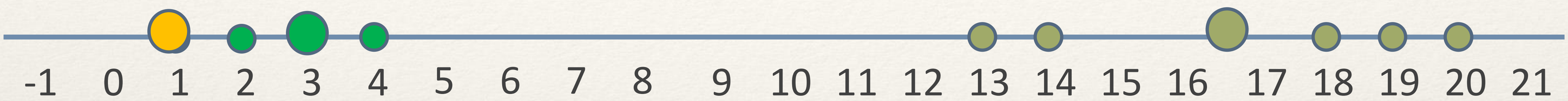
- מרכז סגול: (13+14+18+19+20)/5=16.8
- מרכז ירוק : (2+3+4)/3=3
- מרכז כתום: 1/1=1

<u>איטרציה 3</u>: **K-means שלב 2** - נשייך כעת את הנקודות ל"מרכזים"

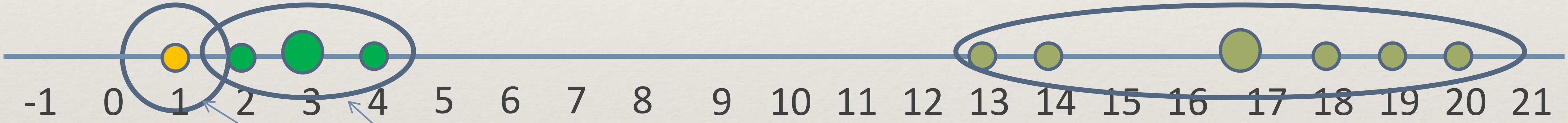


K-means – תרגיל 3 – אותם נתונים עם K=3
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

איטרציה 3: **K-means שלב 3** - נעדכן "מרכזים"



איטרציה 3: **K-means שלב 4** – אין עדכונים ולכן האלגוריתם עוצר



- אין עדכונים ולכן האלגוריתם עוצר

אלו 3 ה"אשכולות" שנוצרו

K-means – תרגיל 3 – אותם נתונים עם K=3
נחשב את הסטיה שנוצרה

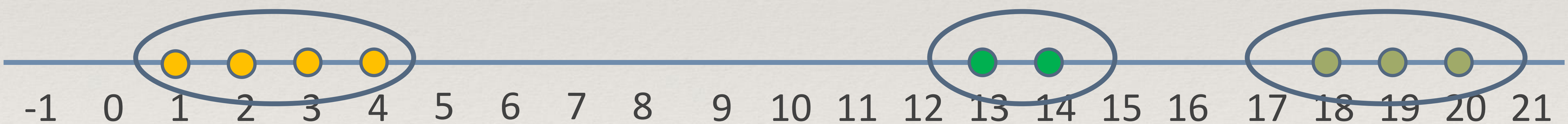
$$cluster1:(1-1)^2=0$$

$$cluster2:(2-3)^2+(3-3)^2+(4-3)^2=2$$

$$cluster3:(13-16.8)^2+(14-16.8)^2+(18-16.8)^2+(19-16.8)^2+(20-16.8)^2=38.8$$

$$\boxed{Total:0+2+38.8=40.8}$$ =WSS

האם יכולנו למצוא סטיה קטנה יותר? – נבחן את האופציה הבאה



$$cluster1:(1-2.5)^2+(2-2.5)^2+(3-2.5)^2+(4-2.5)^2=5$$

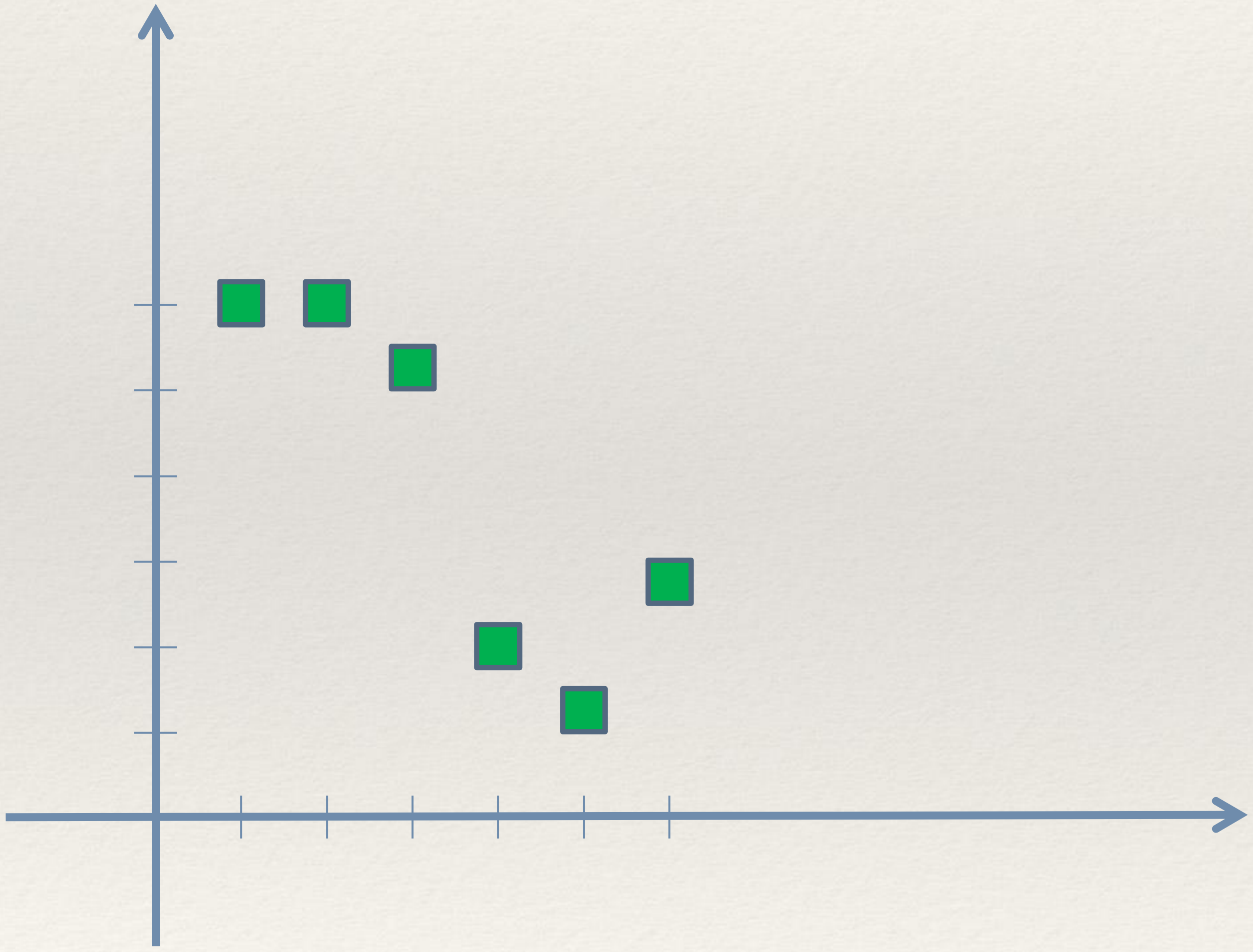
$$cluster2:(13-13.5)^2+(14-13.5)^2=0.5$$

$$cluster3:(18-19)^2+(19-19)^2+(20-19)^2=2$$

$$Total:5+0.5+2=7.5$$ =WSS

תרגיל 4 – K-means
דוגמא עם 2 מאפיינים (2D), K=2

- נתונים הווקטורים הבאים:



x1	x2
2	7
3	6
1	7
5	1
4	2
6	3

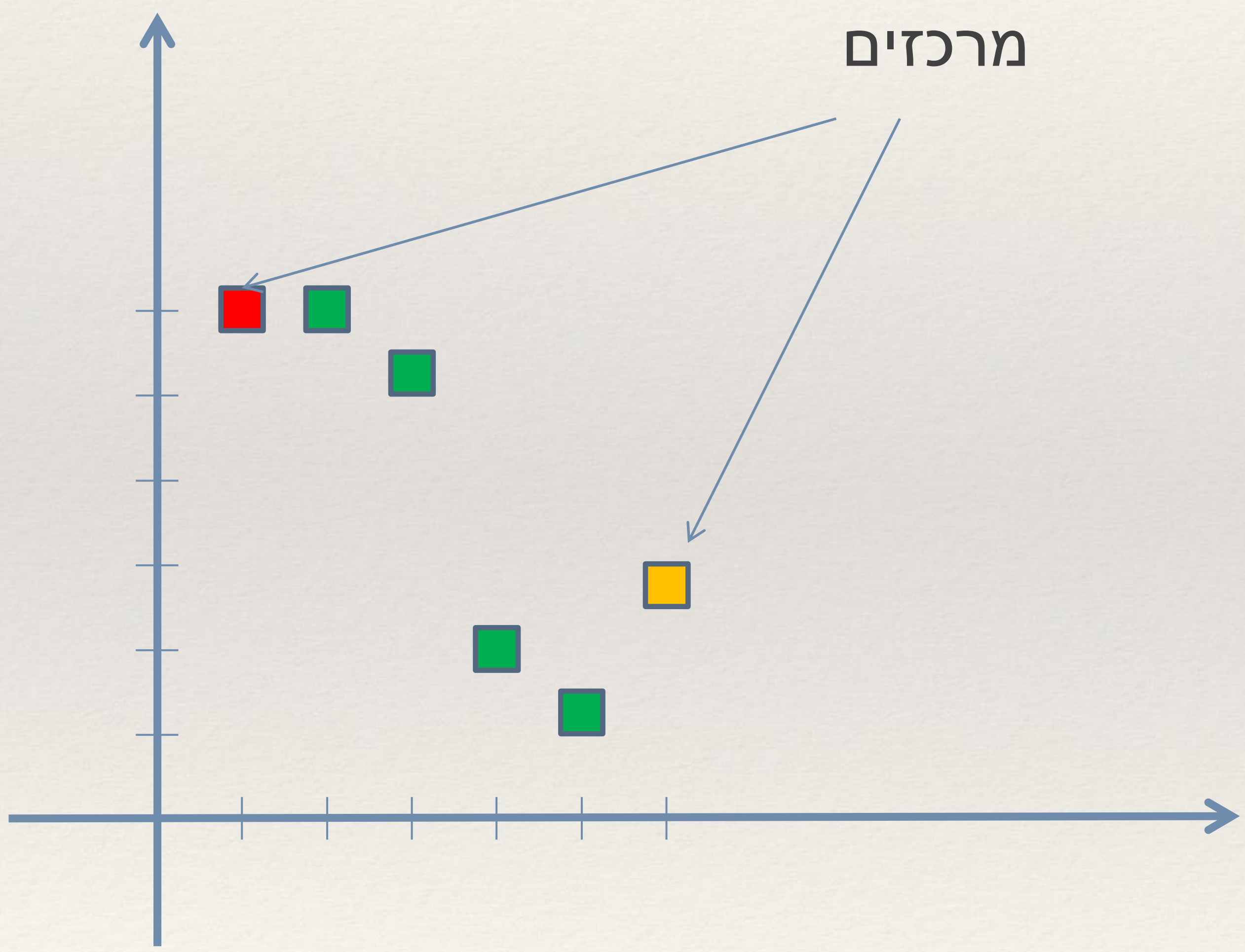
Recalculating centroids

- Clusters based on *centroids* (aka the *center of gravity* or mean) of points in a cluster, *c*:

$$\vec{\mu}(\mathrm{c}) = \frac{1}{|c|} \sum_{\vec{x} \in c} \vec{x}$$

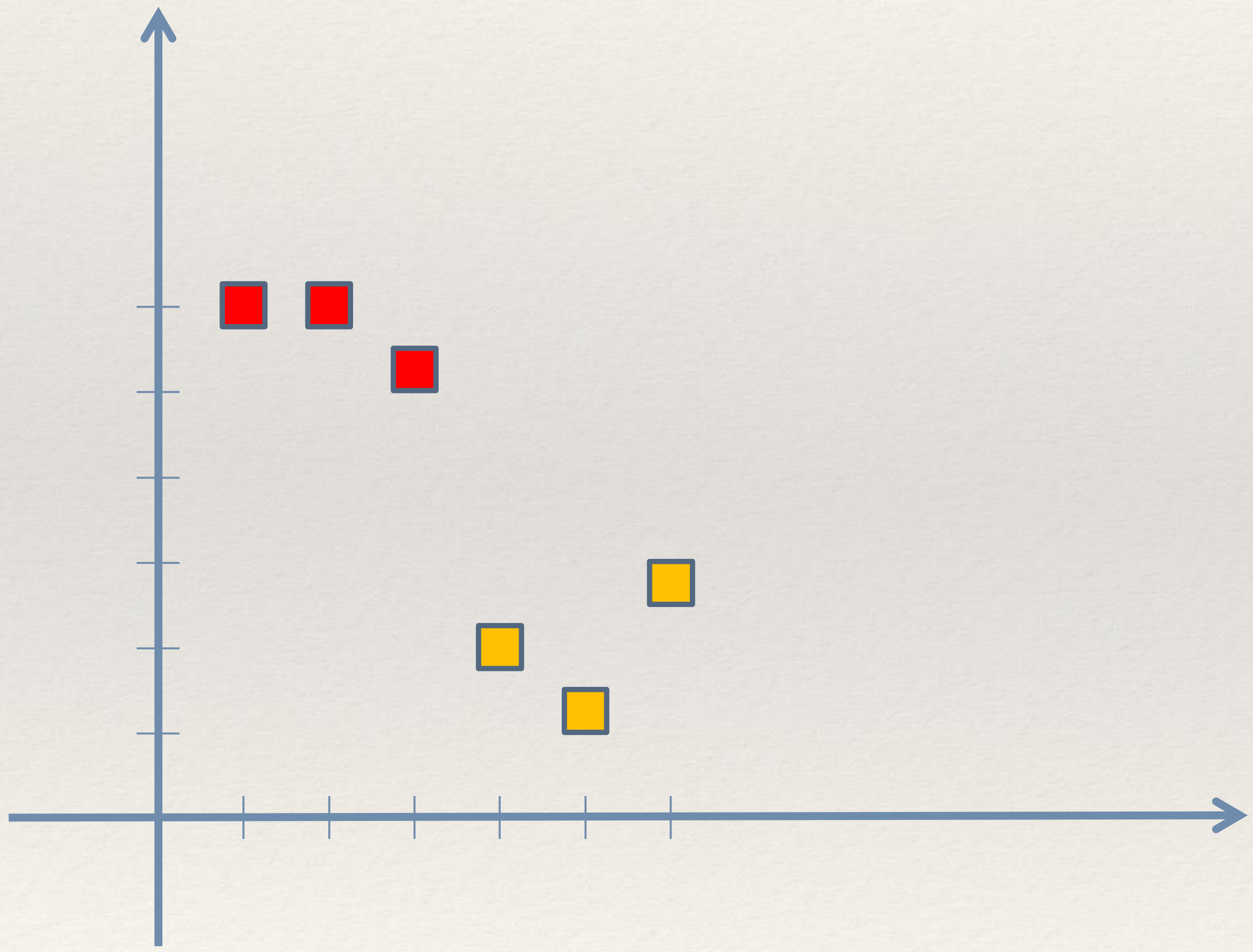
K-means – תרגיל 4 - פתרון
דוגמא עם 2 מאפיינים (2D), K=2

K-means שלב 1 (Forgy method) - ננחש 2 "מרכזים" ראשונים



K-means – תרגיל 4 - <u>פתרון</u>
דוגמא עם 2 מאפיינים (2D), K=2

<u>איטרציה 1</u>: **K-means שלב 2** - נשייך כעת את הנקודות ל"מרכזים"(עפ"י מרחק אוקלידי)



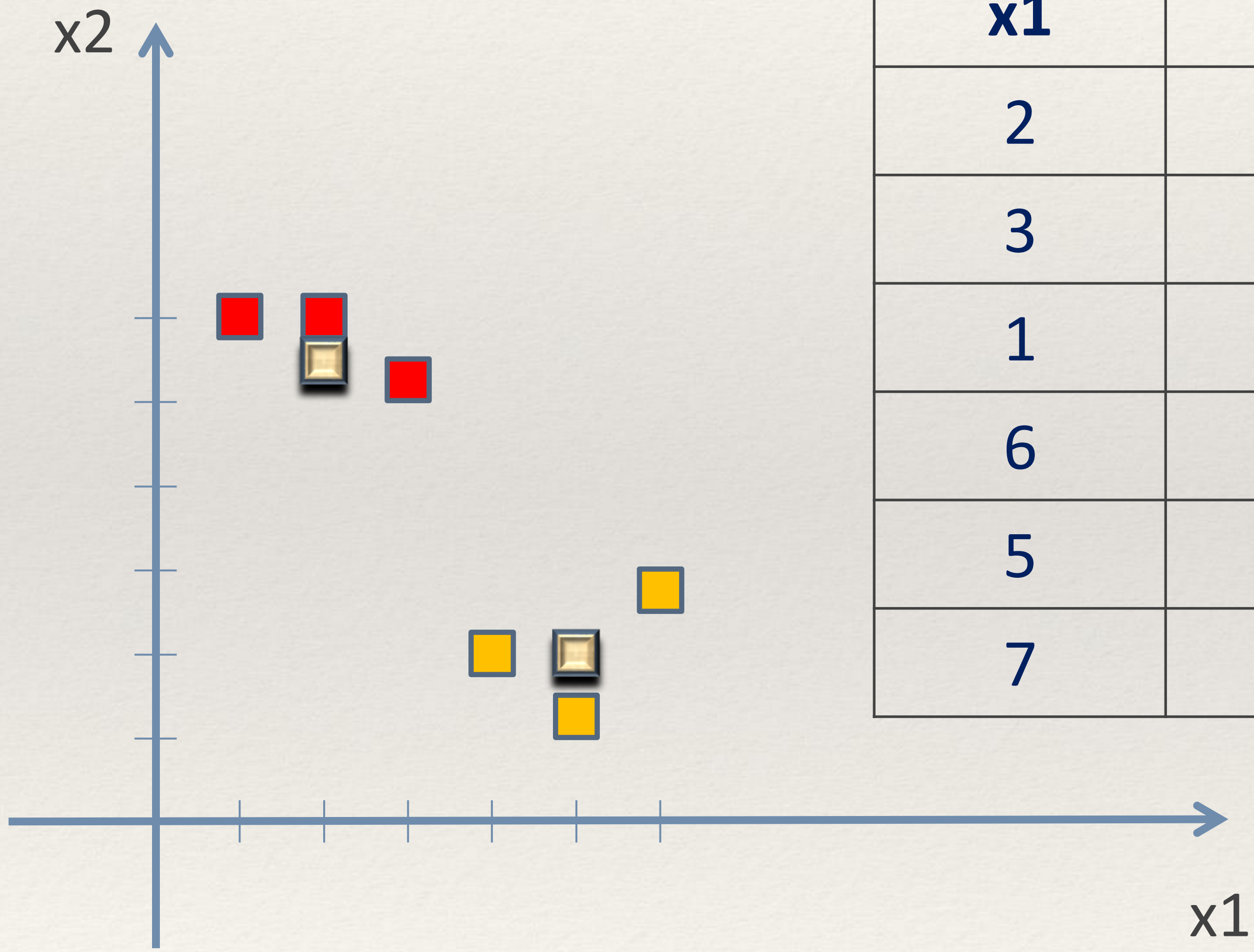
x1	x2
2	7
3	6
1	7
5	1
4	2
6	3

K-means – תרגיל 4 - <u>פתרון</u>
דוגמא עם 2 מאפיינים (2D), K=2

<u>איטרציה 1</u>: **K-means שלב 3** - נעדכן "מרכזים":

- מרכז אדום:
  - $x1 = (1+2+3)/3 = 2$
  - $x2 = (7+7+6)/3 = 6.6$
- מרכז צהוב:
  - $x1 = (4+5+6)/3 = 5$
  - $x2 = (1+2+3)/3 = 2$

x1	x2
2	7
3	6
1	7
6	1
5	2
7	3

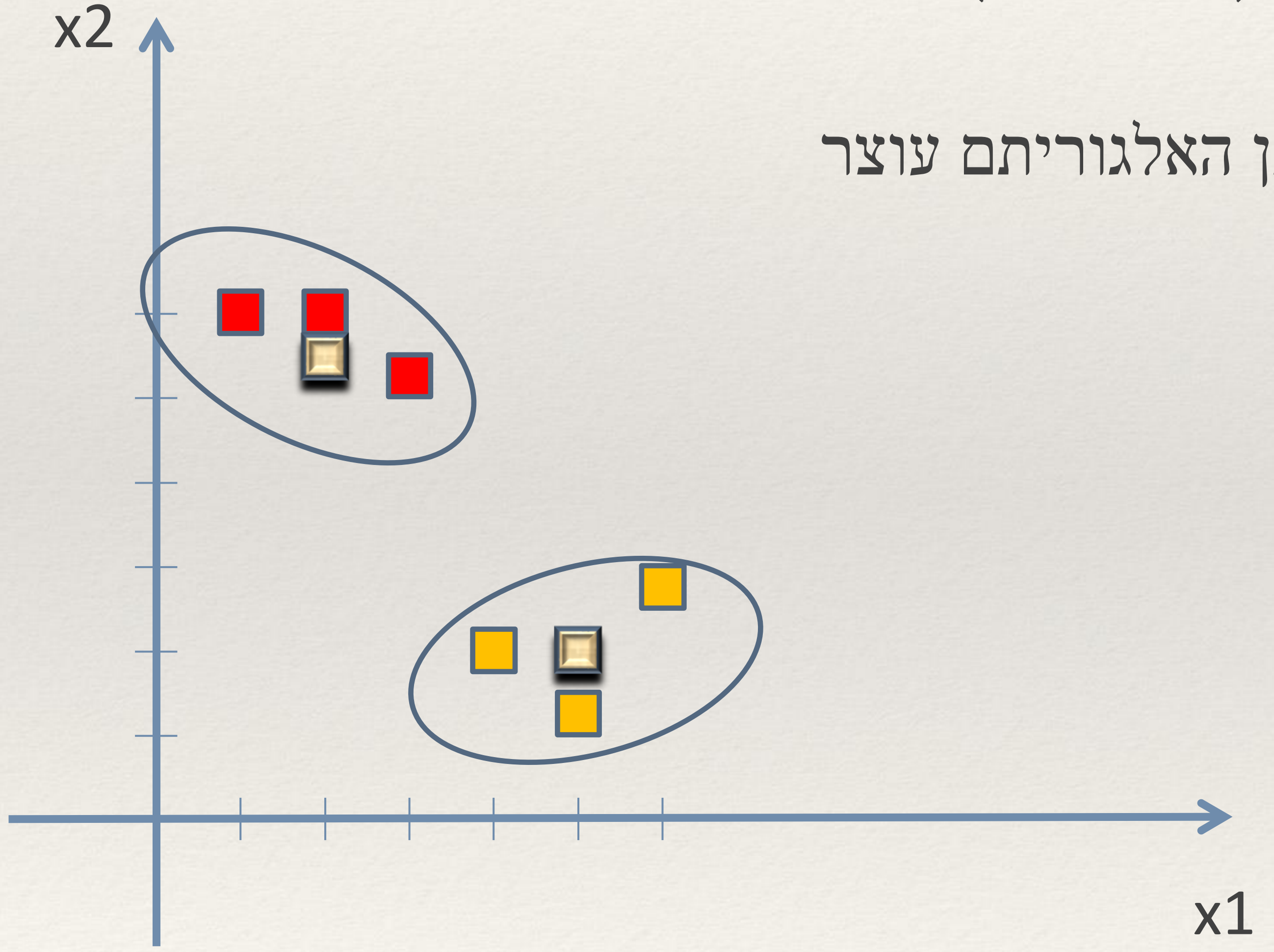


K-means – תרגיל 4 - פתרון
דוגמא עם 2 מאפיינים (2D), K=2

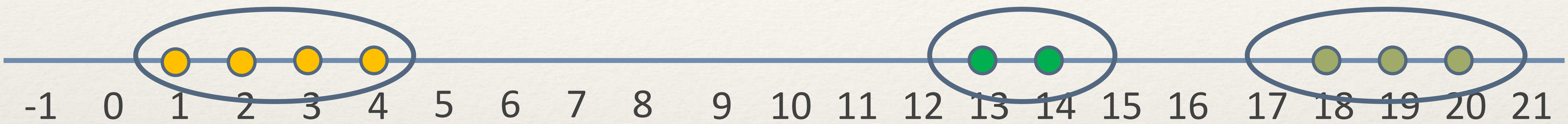
איטרציה 2: **K-means שלב 2** - נשייך כעת את הנקודות ל"מרכזים"

איטרציה 2: **K-means שלב 3** - נעדכן "מרכזים" (אין עדכון)

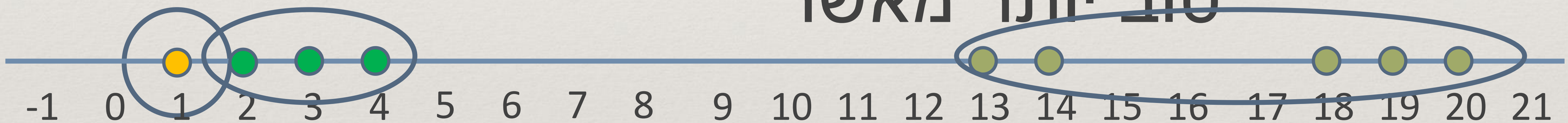
איטרציה 2: **K-means שלב 4** – אין עדכונים ולכן האלגוריתם עוצר



בעיית האתחול - K-means



טוב יותר מאשר



בחירה לא טובה של נקודות ההתחלה הביאה אותנו למינימום מקומי

סוגי בעיות בלמידה לא מונחית

Clustering: represent each input case using a prototype example (we will review k-means)

Dimensionality reduction: represent each input case using a small number of variables (we will review PCA -principal components analysis)

Density estimation: estimating the probability distribution over the data space

הורדת מימדים (dimensionality reduction) – הגדרה

הגדרת הורדת המימדים:

- נתונות לנו n דוגמאות במימד d
- נרצה למצוא יצוג לכל הדוגמאות במימד d נמוך יותר ($k<d$)

Feature selection vs. dimensionality reduction

- ב-Feature Selection בוחרים רק חלק מהמאפיינים, וחלק מסננים.
- ב-dimensionality reduction מייצגים את המאפיינים בפחות מימדים (עם איבוד מידע מנמלי).

איך עושים זאת? הטלה ממימד d למימד k

- PCA – דו' בה ההיטל מורכב מקומבינציות לינאריות של המאפיינים
- tSNE – דו' בה ההיטל מורכב קומבינציות לא לינאריות של המאפיינים

PCA – פעולות מרכזיות

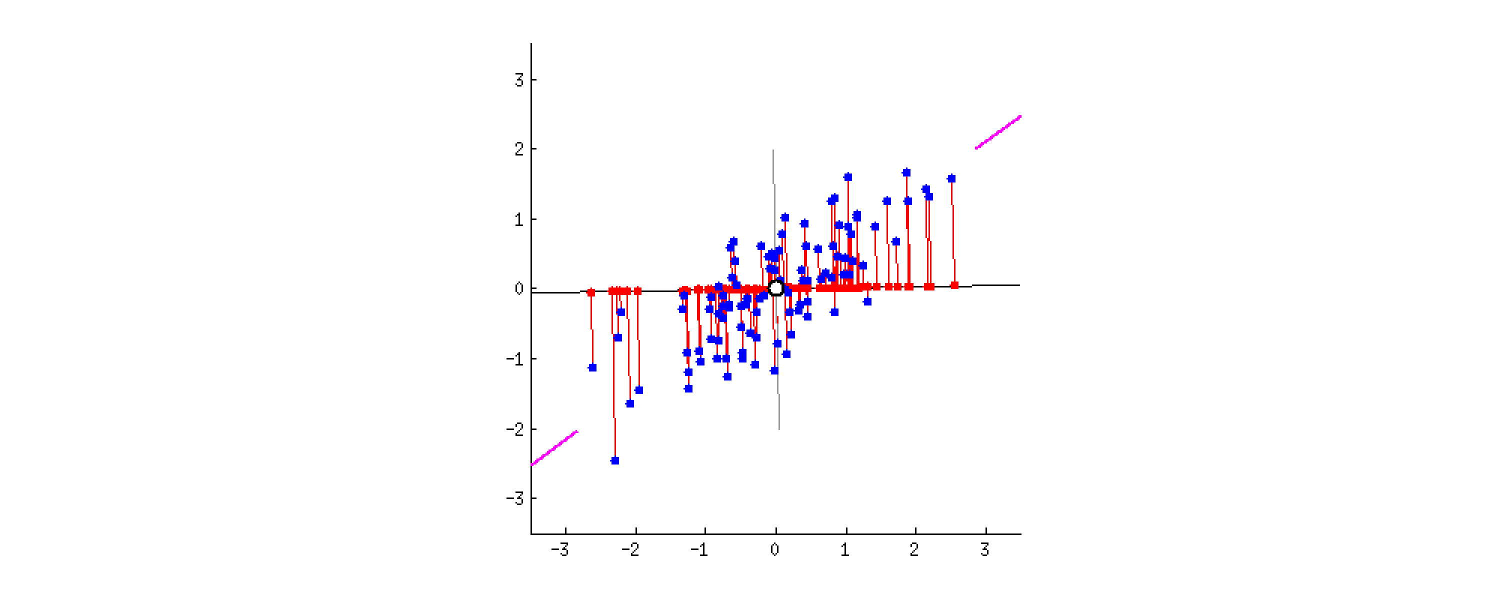
PCA does the following:

- finds orthonormal basis for data
- Sorts dimensions in order of "importance"
- Discard low significance dimensions

Explanations:

- Principal components – the W_i vectors
- Singular values – the coefficients of the principal components
  - higher coefficients mean more important principal components
- λ_i - eigenvalues – square of singular values

Using PCA



Notations

- Reduced dataset – Z
- W – principal components
- $X^{scaled} = standartized\ original\ dataset$

PCA Flow

- Find principal components
- Sort principal components, by the singular values/eigen values
- Select the most significant principal components

Transfer dataset in the following way:

- $Z = W^{T} * {X^{scaled}}^{T}$

PCA - How to choose k ?

Principal components – the W_i vectors

Singular values – the coefficients of the principal components

- λ_i - eigenvalues – square of singular values

How do we choose k?

Use the following proportion: $$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

when λ_i are sorted in descending order

- Typically, stop when proportion>0.9
- K could be also predefined