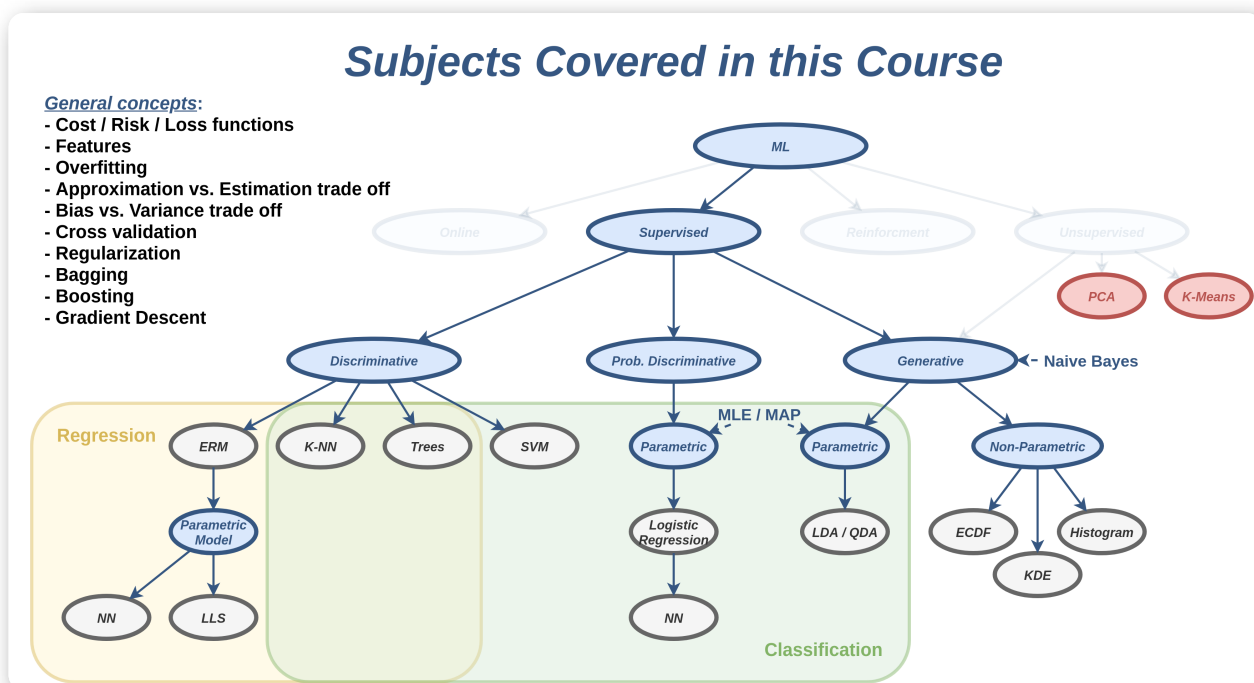


# הרצאה 12 - PCA and K-Means

Slides PDF Code

מה נלמד היום



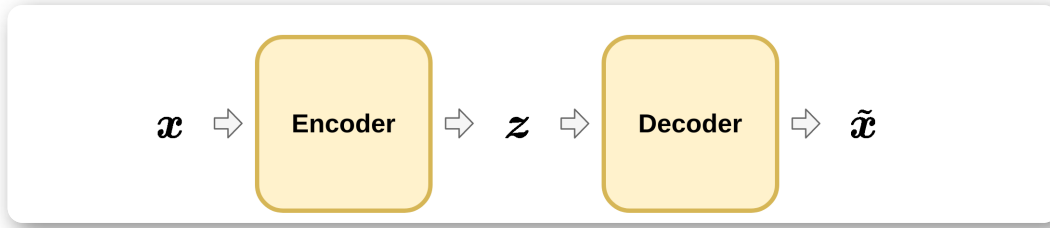
## למידה לא מודרכת (Unsupervised Learning)

Unsupervised learning הינה שם כולל למגוון של בעיות בהם אנו מנסים בהינתן מדגם, ללמוד את התכונות של הדגימות או של המדגם כולו. בניגוד ל supervised learning, ב unsupervised learning המדגם יכול רק אוסף של דגימות  $(\mathbf{x})$ , ללא תווית ( $y$ ). להלן דוגמאות לכמה בעיות ב unsupervised learning:

- אשכול (חלוקה לקבוצות).
- מציאת ייצוג "נוח" יותר של הדגימות.
- דחיסה.
- זיהוי אנומליות.
- למידת הפילוג של הדגימות.

כפי שציינו בעבר, בקורס זה לא נציג את הנושא של unsupervised learning באופן מקיף אלא רק נלמד על שני אלגוריתמים פופולריים מתחום זה, PCA ו K-Means.

## מערכת Encoder-Decoder



במערכת מסוג זה נרצה להשתמש בפונקציית ה encoder על מנת למפות את הוקטור  $\mathbf{x}$  לייצוג אלטרנטיבי  $\mathbf{z}$  אשר יהיה מתאים יותר לשימושים כל שהם. בכדי לנסות שחזר את  $\mathbf{x}$  נוכל להעביר את  $\mathbf{z}$  דרך ה decoder.

דוגמאות לשימושים במערכת encoder-decoder הינם:

- דחיסה: כאן נרצה ש  $\mathbf{z}$  יהיה קטן ככל האפשר (במובן של כמות הביט שנדרשים בכדי לייצג אותו).
- תקשורת: כאן נרצה ש  $\mathbf{z}$  יהיה כמה שפחות רגיש לרעשים של התווך.
- הצפנה: כאן נרצה שפעולת השחזור של  $\mathbf{x}$  תהיה כמה שיותר קשה ללא ה decoder המתאים.

הוקטור  $\tilde{\mathbf{x}}$  המתקבל מהפעלה של ה decoder על הוקטור  $\mathbf{z}$  נקרא השחזור של  $\mathbf{x}$ . בחלק מהמערכות ניתן להגיע לשיחזור מושלם,  $\tilde{\mathbf{x}} = \mathbf{x}$ , ובחלק מהמערכות לא.

## (Principle Component Analysis (PCA

ב PCA ננסה לבנות מערכת encoder-decoder שבה:

1. אנו מגבילים את האורך של הוקטור  $\mathbf{z}$ .
2. אנו דורשים שה encoder וה decoder יהיו פונקציות אפיניות (affine = linear + offset).
3. התוחלת של שגיאת השחזור הריבועית  $\mathbb{E} [\|\tilde{\mathbf{x}} - \mathbf{x}\|_2^2]$  היא מינימאלית.

מכיוון שהפילוג של  $\mathbf{x}$  לרוב לא יהיה ידוע נשתמש במדגם ונחליף את התוחלת בתוחלת אמפירית על המדגם.

נסמן את האורך של הוקטור  $\mathbf{z}$  שאותו אנו מעוניינים לייצר ב  $K$  ואת האורך של  $\mathbf{x}$  ב  $D$  ונגדיר את הבעיה באופן יותר פורמאלי. אנו מעוניינים למצוא encoder מהצורה:

$$\mathbf{z} = T_1 \mathbf{x} + \mathbf{b}_1$$

ו decoder מהצורה של:

$$\tilde{\mathbf{x}} = T_2 \mathbf{z} + \mathbf{b}_2$$

כאשר:

- $T_1$  הינה מטריצה בגודל  $K \times D$ .
- $T_2$  הינה מטריצה בגודל  $D \times K$ .
- $\mathbf{b}_1$  הינה וקטור באורך  $K$ .
- $\mathbf{b}_2$  הינה וקטור באורך  $D$ .

אשר ממזערים את התוחלת האמפירית של שגיאת השחזור הריבועית:

$$\arg \min_{T_1, T_2, \mathbf{b}_1, \mathbf{b}_2} \frac{1}{N} \sum_{i=1}^N \|\tilde{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}\|_2^2$$

## שימושים

ישנם מקרים רבים בהם נרצה למצוא לוקטורים יצוג ממימד נמוך. פעולה זו מכונה **הורדת מימד (dimensionality reduction)** ודוגמאות למקומות שבהם נרצה להשתמש בפעולה זו הינם:

1. בחירת מאפיינים לבעיות supervised learning - בהם נרצה להשתמש בוקטורים ממימד נמוך יותר על מנת להקטין את ה overfitting.
2. ויזואליזציה - בהם נרצה להפוך וקטורים ממימד גבוה למימד 2 או 3 שאותם אנו יודעים לשרטט.
3. דחיסה.

## הפתרון לבעיית האופטימיזציה

נתחיל בלהציג את הפתרון לבעיה.

### הפשטת הבעיה תוך ביטול היתירות

מסתבר שלבעיה זו יש מספר רב של פתרונות. בתהליך פיתוח הפתרון ניתן להראות שניתן לבחור את הפרמטרים כך שיקיימו את האילוצים הבאים מבלי לפגוע באופטימאליות של הפתרון:

$$\begin{aligned} \mathbf{b}_1 &= -T_1 \boldsymbol{\mu} \\ \mathbf{b}_2 &= \boldsymbol{\mu} \\ T_1 &= T_2^\top = T^\top \\ T^\top T &= I \end{aligned}$$

לדוגמה, עבור המיפוי הבא

$$\mathbf{z} = T_1 \mathbf{x} + \mathbf{b}_1 \quad \tilde{\mathbf{x}} = T_2 \mathbf{z} + \mathbf{b}_2 \quad T_1 \in \mathbb{R}^{K \times D}, T_2 \in \mathbb{R}^{D \times K}$$

איברי ההטיה  $\mathbf{b}_1$  ו- $\mathbf{b}_2$  יכולים להיקבע ע"י הדרישות

$$E[\mathbf{z}] = 0 \quad \Rightarrow \quad \mathbf{b}_1 = -T_1 \boldsymbol{\mu}$$

-1

$$E[\tilde{\mathbf{x}}] = E[\mathbf{x}] \quad \Rightarrow \quad \mathbf{b}_2 = E[\mathbf{x}] = \boldsymbol{\mu}$$

כאשר  $\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}^{(i)}$ . הטרנספורמציות במקרה זה הופכות להיות:

$$\begin{aligned} \mathbf{z} &= T^\top (\mathbf{x} - \boldsymbol{\mu}) \\ \tilde{\mathbf{x}} &= T \mathbf{z} + \boldsymbol{\mu} \end{aligned}$$

ובעיית האופטימיזציה הינה:

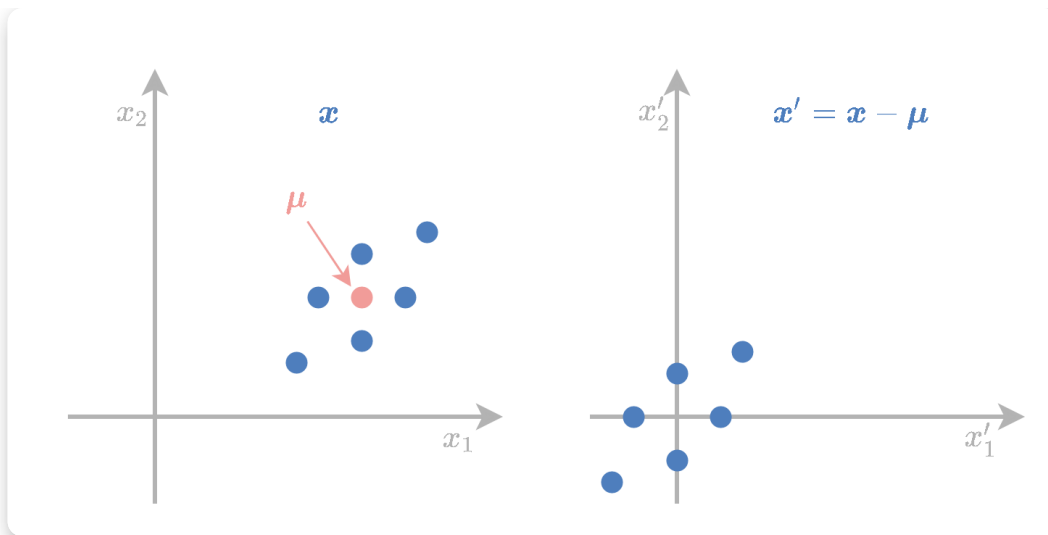
$$\begin{aligned} T^* &= \arg \min_T \frac{1}{N} \sum_{i=1}^N \|\tilde{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}\|_2^2 \\ \text{s.t.} \quad & T^\top T = I \\ T^* &= \arg \min_T \frac{1}{N} \sum_{i=1}^N \|(TT^\top - I)(\mathbf{x}^{(i)} - \boldsymbol{\mu})\|_2^2 \\ \text{s.t.} \quad & T^\top T = I \end{aligned}$$

**שימו לב:**  $T \in \mathbb{R}^{D \times K}$  כך שמתקיים כי  $T^\top T \in \mathbb{R}^{K \times K} = I_K$  היא מטריצת היחידה. בנוסף, מתקיים  $TT^\top \in \mathbb{R}^{D \times D}$  והיא לא שווה בהכרח ל- $I_D$ .

### פרשנות גיאומטרית

ראשית נשים לב שה encoder מתחיל בלחסר את הממוצע של  $\mathbf{x}$  וה decoder מסיים בלהוסיף אותו בחזרה. ניח מעתה שהנתונים ממורכזים סביב האפס.

נדגים זאת עבור המקרה של  $D = 2$  ו- $K = 1$ :



הטרנספורמציות המתקבלות הינן:

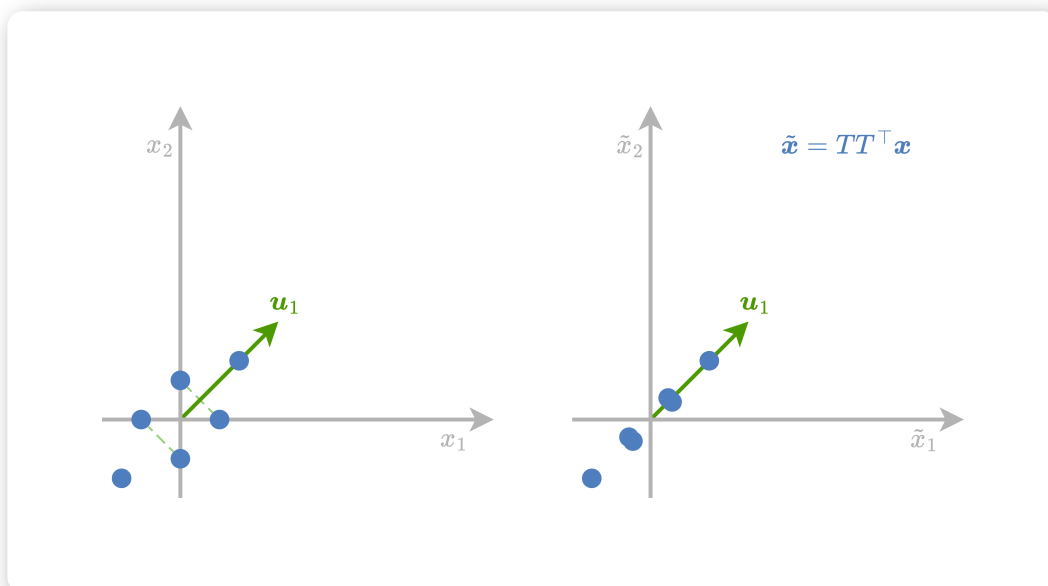
$$z = T^T x$$

$$\tilde{x} = Tz = TT^T x$$

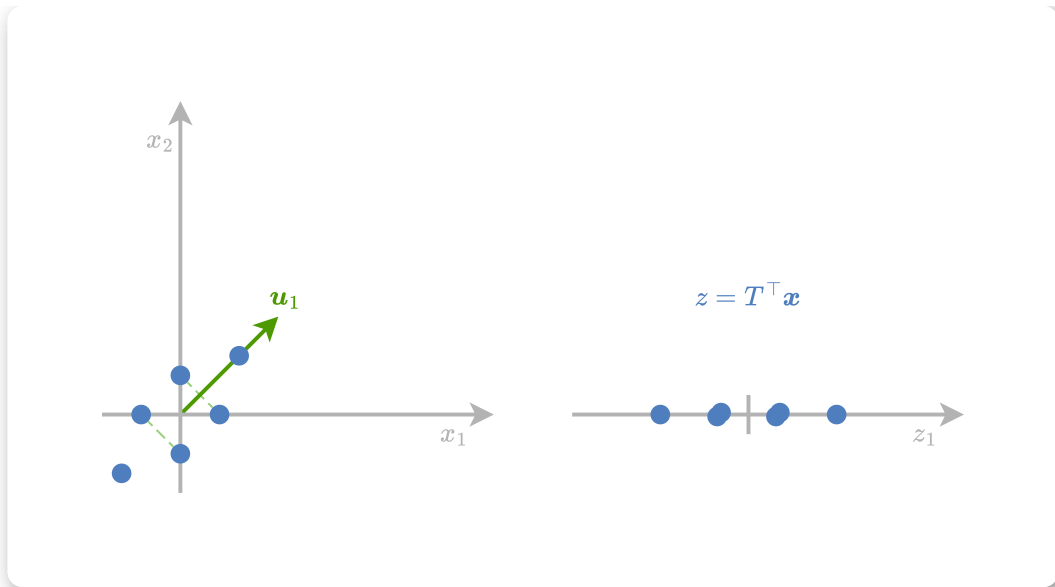
נתייחס כעת לאילוף של  $T^T T = I$ . נציין רק שאילוף זה הוא לא הכרחי בשביל שהפתרון יהיה אופטימאלי אך הוא לא מפשט מאד את הבעיה ומקיים ולא פוגע באופטימאליות של הפתרון. אילוף זה אומר שהעמודות של  $T$  צריכות להיות אורתונורמאליות (אורתוגונאליות ומנורמלות). נסמן את העמודות של  $T$  ב  $u_j$ :

$$T = \begin{pmatrix} | & | & \dots & | \\ u_1 & u_2 & \dots & u_K \\ | & | & \dots & | \end{pmatrix}$$

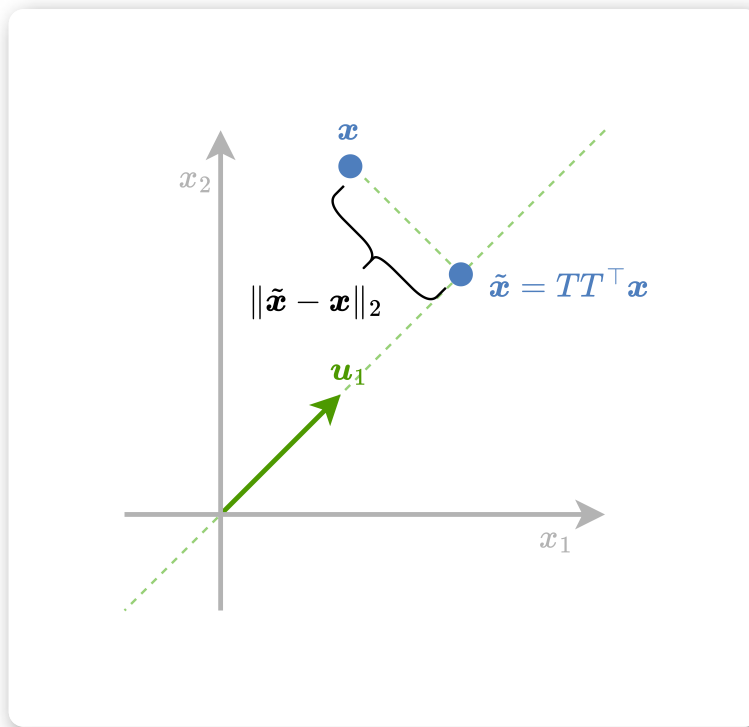
הפעולה של  $\tilde{x} = TT^T x'$  מטילה את הוקטור  $x$  על תת-המרחב הלינארי הנפרס על ידי הוקטורים  $u_j$ . נדגים זאת על המקרה הקודם:



הפעולה של  $z = T^T x$  למעשה גם כן מטילה את  $x$  על אותו תת-מרחב, היא רק משאירה אותו במערכת הצירים אשר מגדרת על ידי הוקטורים  $u_j$ :



נסתכל כעת על המשמעות הגיאומטרית של שגיאת השחזור  $\|\tilde{\mathbf{x}} - \mathbf{x}\|_2^2$ :



הוקטור  $\tilde{\mathbf{x}} - \mathbf{x}$  הוא וקטור המחבר את  $\mathbf{x}$  ל  $\tilde{\mathbf{x}}$ . שגיאת השחזור הריבועית הינה האורך של וקטור זה בריבוע. בעיית האופטימיזציה היא אם כן הבעיה של מציאת תת-המרחב ממימד  $K$  אשר ההטלה של נקודות המדגם עליו הם הקרובות ביותר לנקודות המקוריות.

## הבעיה השקולה

מתוך העובדה ש  $T^T T = I$  ניתן להראות ש:

$$\|\tilde{\mathbf{x}} - \mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2 - \|\tilde{\mathbf{x}}\|_2^2 = \|\mathbf{x}\|_2^2 - \|\mathbf{z}\|_2^2$$

שכן, עבור  $T^T T = I$  מתקיים כי  $(I - TT^T)^2 = (I - TT^T)$  לכן,

$$\begin{aligned}\|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2 &= \|\mathbf{x} - TT^\top \mathbf{x}\|_2^2 \\ &= \|(I - TT^\top) \mathbf{x}\|_2^2 \\ &= \mathbf{x}^\top (I - TT^\top) \mathbf{x} \\ &= \|\mathbf{x}\|_2^2 - \|\mathbf{z}\|_2^2\end{aligned}$$

ובנוסף

$$\|\tilde{\mathbf{x}}\|_2^2 = \|T\mathbf{z}\|_2^2 = \mathbf{z}^\top T^\top T \mathbf{z} = \|\mathbf{z}\|_2^2$$

כעת נוכל לרשום את בעיית האופטימיזציה באופן הבא:

$$\begin{aligned}T^* &= \arg \min_T \frac{1}{N} \sum_{i=1}^N \left( \|\mathbf{x}^{(i)}\|_2^2 - \|\mathbf{z}^{(i)}\|_2^2 \right) \\ \text{s.t. } & T^\top T = I\end{aligned}$$

נזכור ש  $\|\mathbf{x}\|_2^2$  והוא תכונה של הוקטורים במדגם; הם אינם תלויים ב  $T$  ולכן:

$$\begin{aligned}T^* &= \arg \min_T - \frac{1}{N} \sum_{i=1}^N \|\mathbf{z}^{(i)}\|_2^2 \\ \text{s.t. } & T^\top T = I\end{aligned}$$

לכן הבעיה של מזעור שגיאת השחזור הריבועית שקולה לבעיה של מקסום הגודל  $\sum_{i=1}^N \|\mathbf{z}^{(i)}\|_2^2$  אשר מכונה לרוב ה variance של אוסף הוקטורים  $\{\mathbf{z}^{(i)}\}_{i=1}^N$  (בפועל זה ה trace של מטריצת ה covariance האמפירית של  $\mathbf{z}$ )

## הפתרון

בכדי לתאר את הפתרון של בעיית האופטימיזציות האלה (מזעור שגיאת השחזור או מקסום ה variance של  $\mathbf{z}$ ) נגדיר את המטריצות הבאות:

מטריצת המדידות  $X$ :

$$X = \begin{pmatrix} - & \mathbf{x}'^{(1)} & - \\ - & \mathbf{x}'^{(2)} & - \\ & \vdots & \\ - & \mathbf{x}'^{(N)} & - \end{pmatrix} = \begin{pmatrix} - & (\mathbf{x}^{(1)} - \boldsymbol{\mu})^\top & - \\ - & (\mathbf{x}^{(2)} - \boldsymbol{\mu})^\top & - \\ & \vdots & \\ - & (\mathbf{x}^{(N)} - \boldsymbol{\mu})^\top & - \end{pmatrix}$$

מטריצת ה covariance האמפירית של  $\mathbf{x}$  תהיה:  $P = X^\top X$ .

מכיוון ש המטריצה  $P$  היא ממשית וסימטרית מובטח כי ניתן לפרק אותה באופן הבא (ליכסון של המטריצה)  $P = U\Lambda U^\top$  כאשר  $U$  היא מטריצה אורתונורמלית אשר העמודות שלה הם וקטורים עצמיים של  $P$ :

$$U = \begin{pmatrix} | & | & \dots & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_D \\ | & | & & | \end{pmatrix}$$

$\Lambda$  היא מטריצה אלכסונית אשר מכילה את הערכים העצמיים של  $P$ :

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_D \end{pmatrix}$$

כך שהערך העצמי  $\lambda_j$  מתאים לוקטור העצמי  $\mathbf{u}_j$  והערכים העצמיים מסודרים מהגדול לקטן:  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_D$ .

בעזרת מטריצות אלו ניתן כעת לרשום את הפתרון למטריצה  $T$  האופטימאלית. מטריצה זו תהיה מטריצה אשר העמודות שלה הם  $K$  העמודות הראשונות במטריצה  $U$ :

$$T = \begin{pmatrix} | & | & \dots & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_K \\ | & | & \dots & | \end{pmatrix}$$

הכיוונים  $\mathbf{u}^{(j)}$  מכונים **הכיוונים העיקריים** והרכיבים של הוקטור  $\mathbf{z}$  מכונים **הרכיבים העיקריים (principal components)**.

## קווים כלליים לפתרון

נציג את הרעיון הכללי לפתרון הבעיה בלי הפיתוחים המתמטיים מלאים.

### חישוב ה offsets

ראשית ניתן למצוא תנאי על  $\mathbf{b}_1$  ו  $\mathbf{b}_2$  על ידי גזירה והשוואה ל-0. תנאי זה הינו:

$$T_2 \mathbf{b}_1 + \mathbf{b}_2 = -T_2 T_1 \boldsymbol{\mu} + \boldsymbol{\mu}$$

כאשר  $\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}^{(i)}$ . כאשר כל בחירה של  $\mathbf{b}_1$  ו  $\mathbf{b}_2$  שמקיימת את התנאי תהיה אופטימאלית. בפרט נוכל לבחור:

$$\mathbf{b}_1 = -T_1 \boldsymbol{\mu}, \quad \mathbf{b}_2 = \boldsymbol{\mu}$$

ומכאן אנו מקבלים את הטרנספורמציות של:

$$\mathbf{z} = T_1 (\mathbf{x} - \boldsymbol{\mu})$$

$$\tilde{\mathbf{x}} = T_2 \mathbf{z} + \boldsymbol{\mu}$$

### הקשר בין $T_1$ ו $T_2$

על ידי קיבוע  $T_2$  וחיפוש ה  $\mathbf{z}$  אשר ממזער את בעיית האופטימיזציה מקבלים ש:

$$T_1 = (T_2^\top T_2)^{-1} T_2^\top$$

בפרט ניתן להראות שניתן לבחור את  $T_2$  כך ש  $T_2^\top T_2 = I$ . נסמן את  $T_2 = T$  ונקבל ש:

$$\mathbf{z} = T^\top \mathbf{x}'$$

$$\tilde{\mathbf{x}}' = T \mathbf{z} = T T^\top \mathbf{x}'$$

### הפירוק של שגיאת החיזוי

את שגיאת החיזוי הריבועית ניתן לפרק באופן הבא:

$$\begin{aligned} \|\tilde{\mathbf{x}} - \mathbf{x}\|_2^2 &= \|T T^\top \mathbf{x} - \mathbf{x}\|_2^2 \\ &= \|(T T^\top - I) \mathbf{x}\|_2^2 \\ &= \mathbf{x}^\top (T T^\top - I)^\top (T T^\top - I) \mathbf{x} \\ &= \mathbf{x}^\top \underbrace{T T^\top T T^\top}_{=I} \mathbf{x} - 2 \mathbf{x}^\top T T^\top \mathbf{x} + \mathbf{x}^\top \mathbf{x} \\ &= \mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top T T^\top \mathbf{x} \\ &= \mathbf{x}^\top \mathbf{x} - \mathbf{z}^\top \mathbf{z} \\ &= \|\mathbf{x}\|_2^2 - \|\mathbf{z}\|_2^2 \end{aligned}$$

### מציאת ה $T$ האופטימאלי

נסתכל על בעיית האופטימיזציה השקולה:

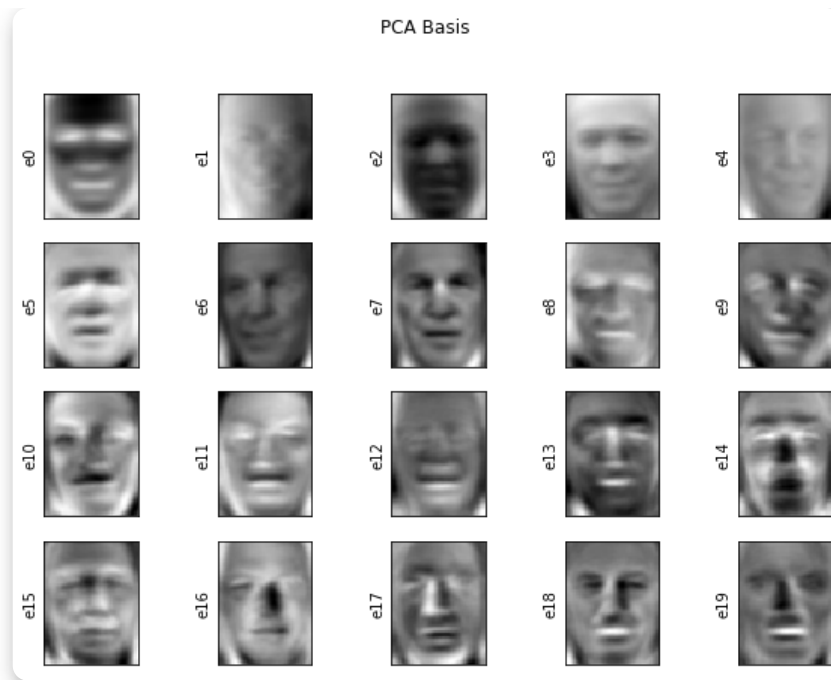
$$\begin{aligned}
 T^* &= \arg \min_T -\frac{1}{N} \sum_{i=1}^N \|\mathbf{z}^{(i)}\|_2^2 \\
 &\text{s.t. } T^\top T = I \\
 &= \arg \min_T -\frac{1}{N} \sum_{i=1}^N \|T^\top \mathbf{x}'^{(i)}\|_2^2 \\
 &\text{s.t. } T^\top T = I \\
 &= \arg \min_T -\frac{1}{N} \sum_{i=1}^N \mathbf{x}'^{(i)\top} T T^\top \mathbf{x}'^{(i)} \\
 &\text{s.t. } T^\top T = I \\
 &= \arg \min_T -\frac{1}{N} \sum_{i=1}^N \text{tr} \left( \mathbf{x}'^{(i)\top} T T^\top \mathbf{x}'^{(i)} \right) \\
 &\text{s.t. } T^\top T = I \\
 &= \arg \min_T -\text{tr} \left( \left( \frac{1}{N} \sum_{i=1}^N \mathbf{x}'^{(i)} \mathbf{x}'^{(i)\top} \right) T T^\top \right) \\
 &\text{s.t. } T^\top T = I \\
 &= \arg \min_T -\text{tr} (X^\top X T T^\top) \\
 &\text{s.t. } T^\top T = I \\
 &= \arg \min_T -\text{tr} (T^\top P T) \\
 &\text{s.t. } T^\top T = I
 \end{aligned}$$

ניתן להראות שהפתרון לבעיה זו הינה המטרצה  $T$  שתוארה בפתרון על ידי שימוש באינדוקציה, כאשר מתחילים מ  $K = 1$  ומגדילים אותו כל פעם ב 1.

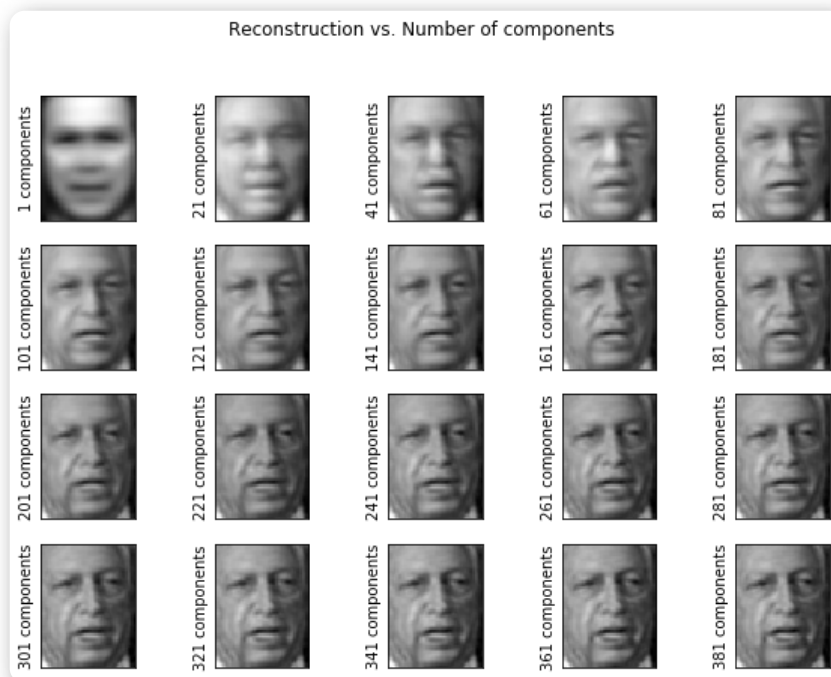
## דוגמא

נציג דוגמא לפירוק PCA של תמונות. נתייחס לתמונות בעל וקטור ארוך של פיקסלים. בדומא הבאה נסתכל על תמונות של 381 פיקסלים. 20 הכיוונים העיקריים (הוקטורים העצמיים המתאימים לערכים העצמיים הכי גדולים) הינם:



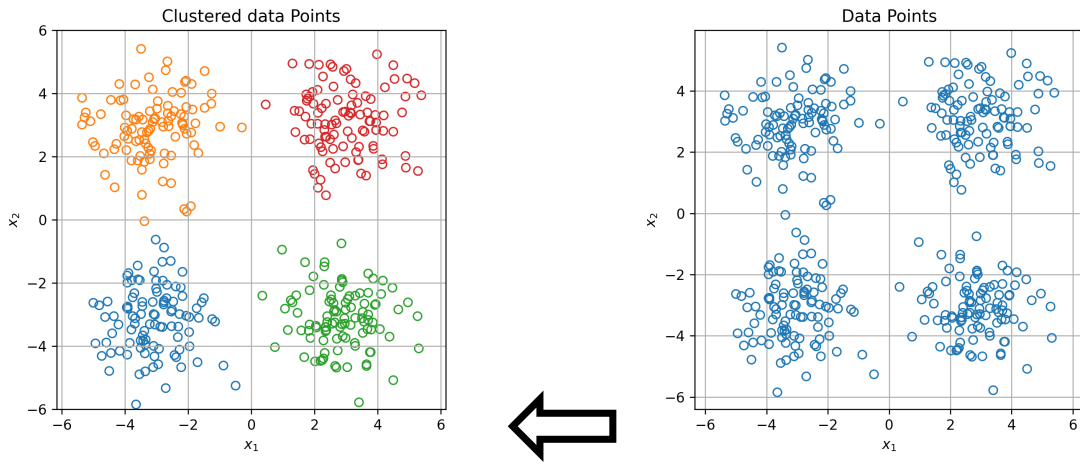


נציג כעת את התמונה המשוחזרת בעבור ערכים שונים של  $K$ :



## אשכול

באלגוריתמי אשכול ננסה לחלק אוסף של פרטים לקבוצות המכונים אשכולות (clusters), כאשר לכל קבוצה איזשהן תכונות דומות.

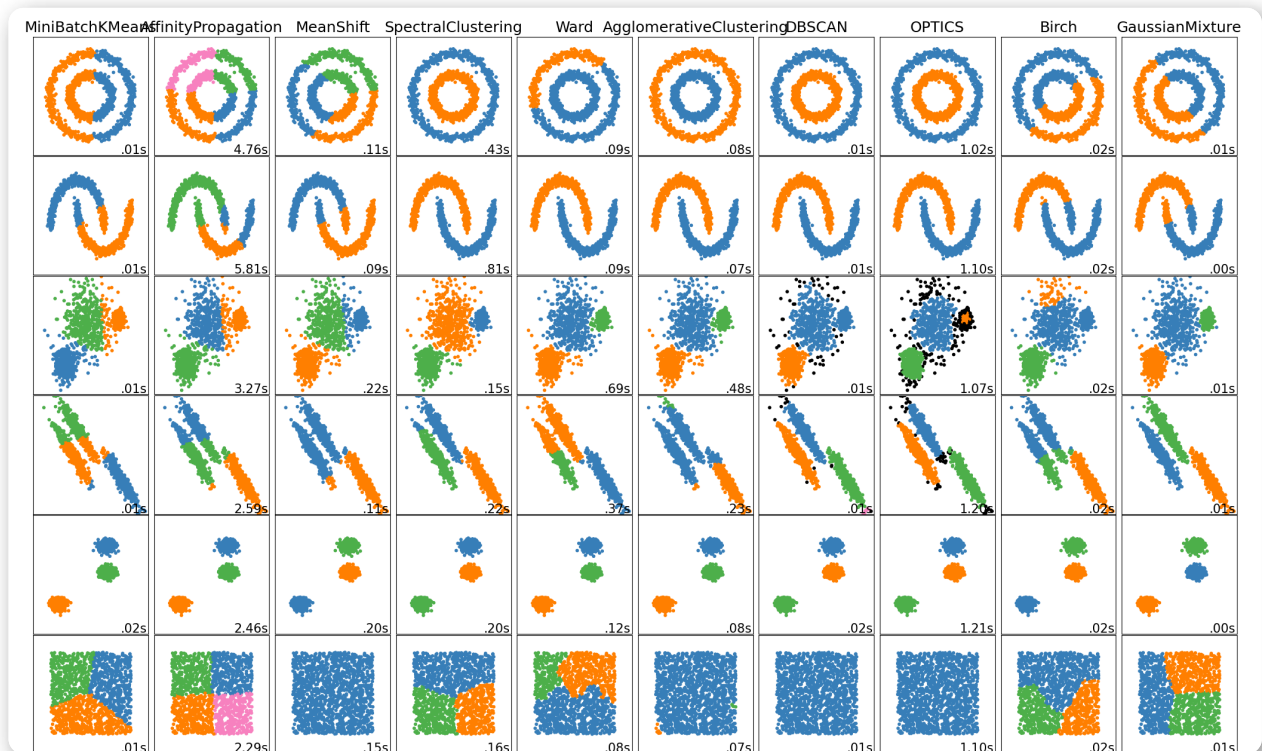


2 דוגמאות למקרים שבהם נרצה לאשכל אוסף נתונים:

1. על מנת לבצע הנחות על אחד מהפרטים באשכול על סמך פרטים אחרים באשכול. לדוגמא: להציע ללקוח מסוים בחנות אינטרנט מוצרים על סמך מוצרים שקנו לקוחות אחרים באשכול שלו.
2. לתת טיפול שונה לכל אשכול. לדוגמא משרד ממשלתי שרוצה להפנות קבוצות שונות באוכלוסיה לערוצי מתן שירות שונים: אפליקציה, אתר אינטרנט, נציג טלפוני או הפניה פיסית למוקד שירות.

## אלגוריתמי אשכול שונים

קיימות דרכים רבות לבצע אישכול לאוסף של נתונים. בהתאם לכך קיימים גם מספר רב של אלגוריתמים לעשות כן. בתיעוד של החבילה הפייתונית [scikit-learn](#), בה נעשה שימוש רב בתרגילים הרטובים בקורס, ישנה השוואה בין האשכולות המתקבלים מאלגוריתמים האישכול השונים בחבילה, בעבור שישה toy models ID מימדיים:



נציין כי לרוב נעבוד עם נתונים ממימד גבוה, שם לא נוכל, כמו כאן, לצייר את האשכולות על מנת להבין את אופי החלוקה. בקורס זה נלמד על האלגוריתם K-means (העמודה השמאלית ביותר).

K-Means הוא אלגוריתם אשכול אשר מנסה לחלק את הדגימות במדגם ל  $K$  קבוצות על סמך המרחק בין הדגימות.

## סימונים

- $K$  - מספר האשכולות (גודל אשר נקבע מראש).
- $\mathcal{I}_k$  - אוסף האינדקסים של האשכול ה- $k$ . לדוגמא:  $\mathcal{I}_5 = \{3, 6, 9, 13\}$
- $|\mathcal{I}_k|$  - גודל האשכול ה- $k$  (מספר הפרטים בקבוצה)
- $\{\mathcal{I}_k\}_{k=1}^K$  - חלוקה מסוימת לאשכולות

## בעיית האופטימיזציה

בהינתן מדגם K-Means  $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ , מנסה למצוא את החלוקה לאשכולות אשר תמצער את המרחק הריבועי הממוצע בין כל דגימה לכל שאר הדגימות שאיתו באותו האשכול. זאת אומרת, K-means מנסה לפתור את בעיית האופטימיזציה הבאה:

$$\arg \min_{\{\mathcal{I}_j\}_{j=1}^K} \frac{1}{N} \sum_{k=1}^K \frac{1}{2|\mathcal{I}_k|} \sum_{i,j \in \mathcal{I}_k} \|\mathbf{x}^{(j)} - \mathbf{x}^{(i)}\|_2^2$$

## הבעיה השקולה

נגדיר את מרכז המסה של כל אשכול כממוצע של כל הוקטורים באשכול:

$$\boldsymbol{\mu}_k = \frac{1}{|\mathcal{I}_k|} \sum_{i \in \mathcal{I}_k} \mathbf{x}^{(i)}$$

ניתן להראות כי בעיית האופטימיזציה המקורית, שקולה לבעיה של מיזעור המרחק הממוצע של הדגימות ממרכז המסה של האשכול:

$$\arg \min_{\{\mathcal{I}_j\}_{j=1}^K} \frac{1}{N} \sum_{k=1}^K \sum_{i \in \mathcal{I}_k} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2$$

כלומר,

$$\begin{aligned} \sum_{i,j \in \mathcal{I}_k} \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_2^2 &= \sum_{i,j \in \mathcal{I}_k} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k + \boldsymbol{\mu}_k - \mathbf{x}^{(j)}\|_2^2 \\ &= \sum_{i,j \in \mathcal{I}_k} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2 + \sum_{i,j \in \mathcal{I}_k} \|\mathbf{x}^{(j)} - \boldsymbol{\mu}_k\|_2^2 - 2 \sum_{i,j \in \mathcal{I}_k} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_k)^\top (\mathbf{x}^{(j)} - \boldsymbol{\mu}_k) \\ &= 2|\mathcal{I}_k| \sum_{i \in \mathcal{I}_k} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2 - 2 \sum_{i \in \mathcal{I}_k} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_k)^\top \sum_{j \in \mathcal{I}_k} (\mathbf{x}^{(j)} - \boldsymbol{\mu}_k) \\ &= 2|\mathcal{I}_k| \sum_{i \in \mathcal{I}_k} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2 \end{aligned}$$

שכן:

$$\sum_{i \in \mathcal{I}_k} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_k) = |\mathcal{I}_k| \cdot \frac{1}{|\mathcal{I}_k|} \sum_{i \in \mathcal{I}_k} \mathbf{x}^{(i)} - |\mathcal{I}_k| \boldsymbol{\mu}_k = 0$$

## האלגוריתם

K-mean הוא אלגוריתם חמדן אשר בכל פעם משיך מחדש את הדגימות ומעדכן את המרכזים.

האלגוריתם מאותחל בצעד  $t = 0$  על ידי בחירה אקראית של  $K$  מרכזי מסה:  $\{\mu_k\}_{k=1}^K$ .

בכל צעד  $t$  מבצעים את שתי הפעולות הבאות:

1. עדכון מחדש של החלוקה לאשכולות  $\{\mathcal{I}_k\}_{k=1}^K$  כך שכל דגימה משוייכת למרכז המסה הקרוב עליה ביותר. כלומר אנו נשייך את כל דגימה  $\mathbf{x}$  לפי:

$$k = \arg \min_{k \in [1, K]} \|\mathbf{x} - \mu_k\|_2^2$$

(במקרה של שני מרכזים במרחק זהה נבחר בזה בעל האינדקס הנמוך יותר).

2. עדכון של מרכזי המסה המסה על פי:

$$\mu_k = \frac{1}{|\mathcal{I}_k|} \sum_{i \in \mathcal{I}_k} \mathbf{x}^{(i)}$$

(אם  $|\mathcal{I}_k| = 0$  אז משאירים אותו ללא שינוי)

תנאי העצירה של האלגוריתם הינו כשהאשכולות מפסיקות להשתנות.

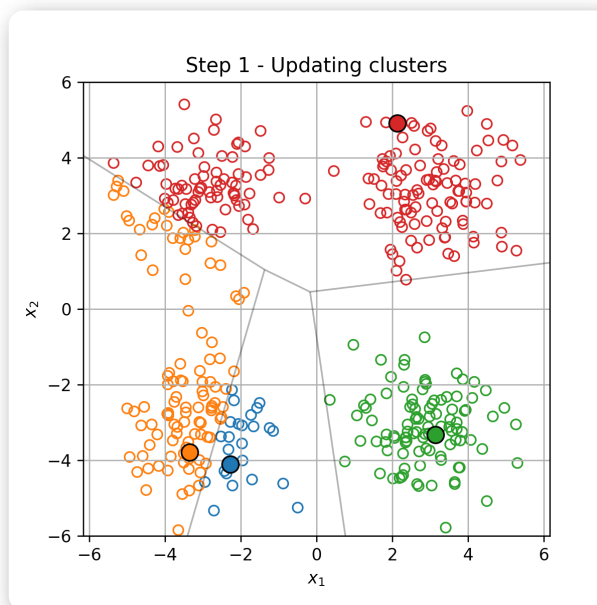
אחת הדרכים הנפוצות לאיתחול של  $\{\mu_k\}_{k=1}^K$  היא לבחור  $k$  נקודות מתוך המדגם.

## תכונות

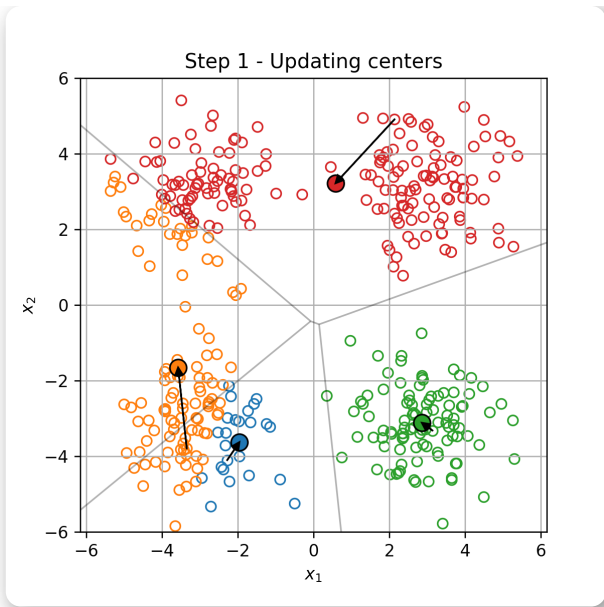
- מובטח כי פונקציית המטרה (סכום המרחקים מהממוצעים) תקטן בכל צעד.
- מובטח כי האלגוריתם יעצר לאחר מספר סופי של צעדים.
- **לא** מובטח כי האלגוריתם יתכנס לפתרון האופטימאלי, אם כי בפועל במרבית המקרים האלגוריתם מתכנס לפתרון אשר קרוב מאד לאופטימאלי.
- אתחולים שונים יכולים להוביל לתוצאות שונות.

## דוגמא

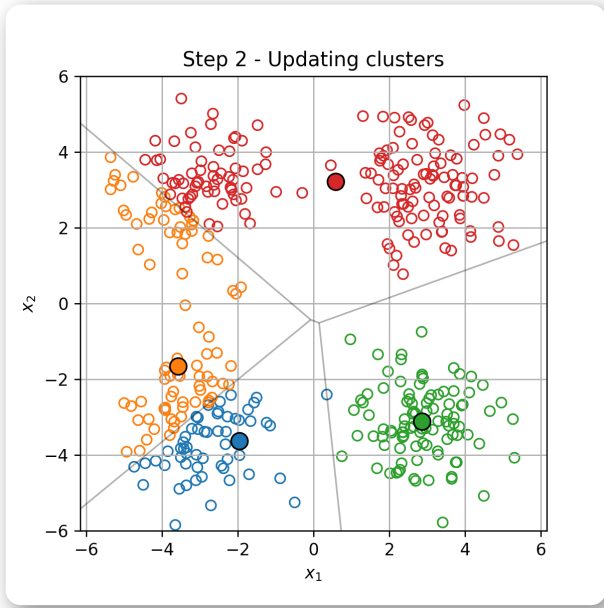
אתחול (וחלוקה ראשונית לאשכולות):



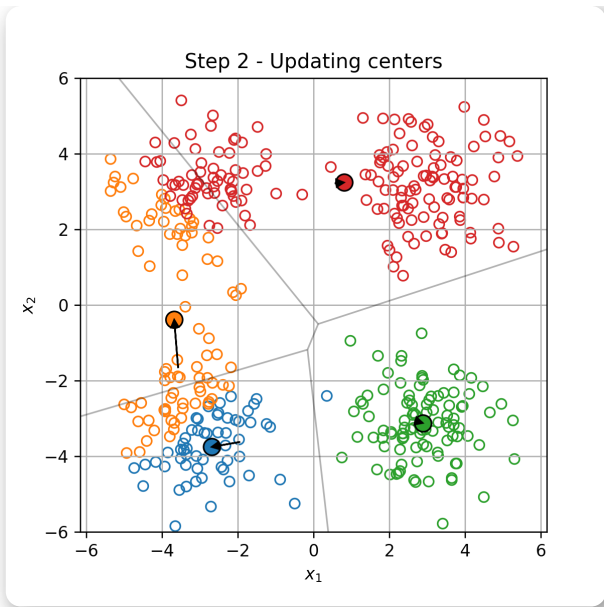
עדכון המרכזים:



עדכון האשכולות:



עדכון המרכזים:



והזר חלילה (הסדר הוא מימין לשמאל):

