

תרגול 12 - PCA and K-means

Slides

PDF

Code

תקציר התיאוריה - PCA

PCA הוא אלגוריתם מאוד נפוץ אשר משמש במקומות רבים על מנת למצוא יצוג נוח יותר לוקטורים על סמך מדגם נתון. אחד השימושים העיקריים של האלגוריתם הינו בכדי לבצע **הורדת מימד** של הוקטורים. (ליוצג וקטורים בעזרת וקטור ממימד נמוך יותר).

הגדרות

בעבור מדגם נתון $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ של N וקטורים באורך D נגדיר את הגדלים הבאים:

- הממוצע של המדגם: $\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}^{(i)}$

- מטריצת הדגימות:

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

- הקווריאנס האמפירי של המדגם: $P = X^\top X$

נתייחס לפירוק (ליכסון) הבא: $P = U\Lambda U^\top$ כאשר U היא מטריצה אורתונורמלית אשר העמודות שלה הם וקטורים עצמיים של P :

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

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

הטרנספורמציה אותה מבצע PCA

PCA מייצר מתוך מדגם נתון $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ טרנספורמציה אפינית (affine = linear + offset) אשר ממפה וקטור \mathbf{x} באורך D לוקטור \mathbf{z} באורך K . כאשר $K \leq D$. הוא קבוע אשר נבחר מראש. הטרנספורמציה הינה:

$$\mathbf{z} = T^T(\mathbf{x} - \bar{\mathbf{x}})$$

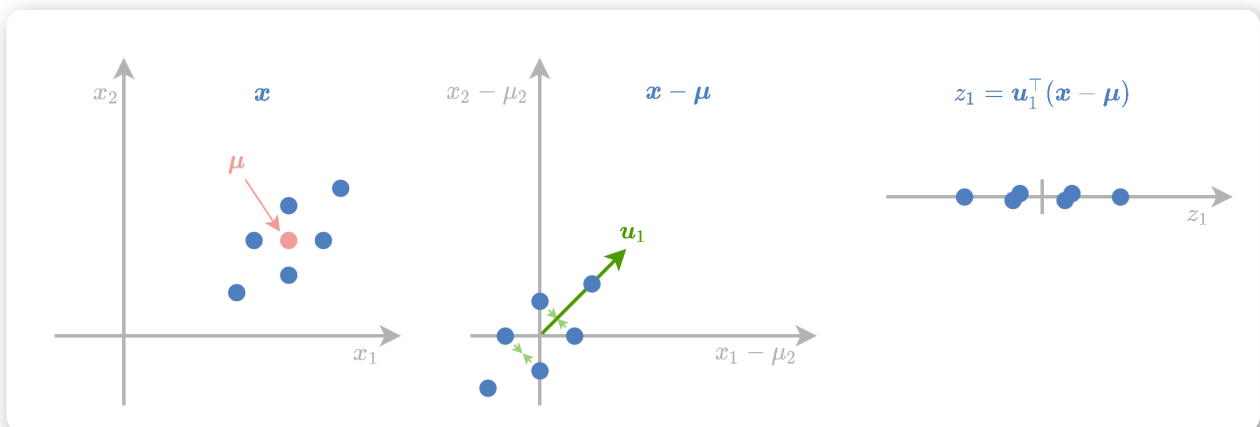
כאשר T הינה מטריצה המכילה את K העמודות הראשונות של U (זאת אומרת הוקטורים העצמיים המתאימים ל K הערכים העצמיים הגדולים ביותר).

האיברים של \mathbf{z} נקראים הרכיבים הראשיים (principal components) של \mathbf{x} .

פרשנות גיאומטרית

הפעולה שאותה מבצעת הטרנספורמציה הינה:

1. להזיז את הנקודות של המדגם כך שהמרכז שלהם יהיה בראשית.
2. הטלה של הנקודות המוזזות על תת-המרחב שמוגדרת על ידי הוקטורים $\{\mathbf{u}_j\}$.



מוטיבציה ראשונה: מקסימום שונות

תחת האילוץ ש T הינה מטריצה בגודל $D \times K$ בעלת עמודות אורתו-נורמאליות, הבחירה הנוכחית של T ממקסמת את הגודל:

$$\frac{1}{N} \sum_{i=1}^N \|\mathbf{z}^{(i)}\|_2^2$$

אשר מכונה לרוב השונות של אוסף הוקטורים $\{\mathbf{z}^{(i)}\}_{i=1}^N$ (בפועל זה שווה ל $\text{trace}(Z^T Z)$, כאשר Z מוגדרת בדומה ל X רק עם הוקטורים $\{\mathbf{z}^{(i)}\}$).

מוטיבציה שניה: מזעור שגיאת השיחזור הריבועית

נסתכל על זוג טרנספורמציות אפיניות כלליות מ \mathbf{x} ל \mathbf{z} באורך K , ומ \mathbf{z} ל $\tilde{\mathbf{x}}$:

$$\mathbf{z} = A\mathbf{x} + \mathbf{b}$$

$$\tilde{\mathbf{x}} = C\mathbf{z} + \mathbf{d}$$

נסמן את שגיאת השיחזור הריבועית באופן הבא: $\sum_{i=1}^N (\tilde{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)})^2$.

הטרנספורמציות שימזערו את שגיאת השיחזור הריבועית הינם:

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

$$\tilde{\mathbf{x}} = U^T \mathbf{z} + \boldsymbol{\mu}$$

תקציר התיאוריה - K-Means

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 מסויימת 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:

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

הבעיה השקולה

נגדיר את מרכז המסה של כל אשכול כממוצע של כל הזקטורים 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:

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

האלגוריתם

K-mean הוא אלגוריתם חמדי אשר בכל פעם משייך מחדש את הדגימות ומעדכן את המרכזים.

האלגוריתם מאתחל בצעד $t = 0$ על ידי בחירה אקראית של K מרכזי מסה: $\{\boldsymbol{\mu}_k\}_{k=1}^K$.

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 כל דגימה \mathbf{x} לפי:

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

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

2. עדכון של מרכזי המסה על פי:

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

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

תנאי העצירה של האלגוריתם הינו כשהאשכולות מפסיקות להשתנות.

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

- 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.

תרגיל 12.1 - PCA

עבוד מדגם נתון של וקטורים ב \mathbb{R}^2 חושבו וקטור הממוצע ומטריצת הקוואריאנס הבאים:

$$\bar{x} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$P = \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}$$

(1) איזה מהוקטורים הבאים מייצג את הכיוון הראשון u_1 במטריצת ההטלה של PCA?

$$\frac{1}{\sqrt{5}} \begin{pmatrix} -2 \\ 1 \end{pmatrix}, \quad \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \frac{1}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \end{pmatrix},$$

(2) חשבו את שני ה principal components של $x = (1, 0)^T$.

פתרון 12.1

(1)

נשתמש בעובדה ש u_1 צריך להיות וקטור עצמי של P ולכן מקיים $Pu_1 = \lambda_1 u_1$. נבדוק מי מהוקטורים הבאים מקיים זאת:

$$Pu_1 = \frac{1}{\sqrt{5}} \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix} \begin{pmatrix} -2 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{5}} \begin{pmatrix} -4 \\ 2 \end{pmatrix} = 2u_1$$

$$Pu_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 5 \\ 8 \end{pmatrix} \neq \alpha u_1$$

$$Pu_1 = \frac{1}{\sqrt{5}} \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \frac{1}{\sqrt{5}} \begin{pmatrix} 7 \\ 14 \end{pmatrix} = 7u_1$$

מכאן שגם הוקטור הראשון וגם השלישי הם וקטורים עצמיים. הוקטור הראשון בהטלה של PCA יהיה השלישי שכן הוא מתאים לערך עצמי גדול יותר:

$$u_1 = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \end{pmatrix},$$

(2)

הרכיב העיקרי (principal component) הראשון יהיה נתון על ידי:

$$z_1 = u_1^T (x - \mu) = \frac{1}{\sqrt{5}} (1 \ 2) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{5}}$$

והרכיב השני יהיה:

$$z_2 = u_2^T (x - \mu) = \frac{1}{\sqrt{5}} (-2 \ 1) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{-2}{\sqrt{5}}$$

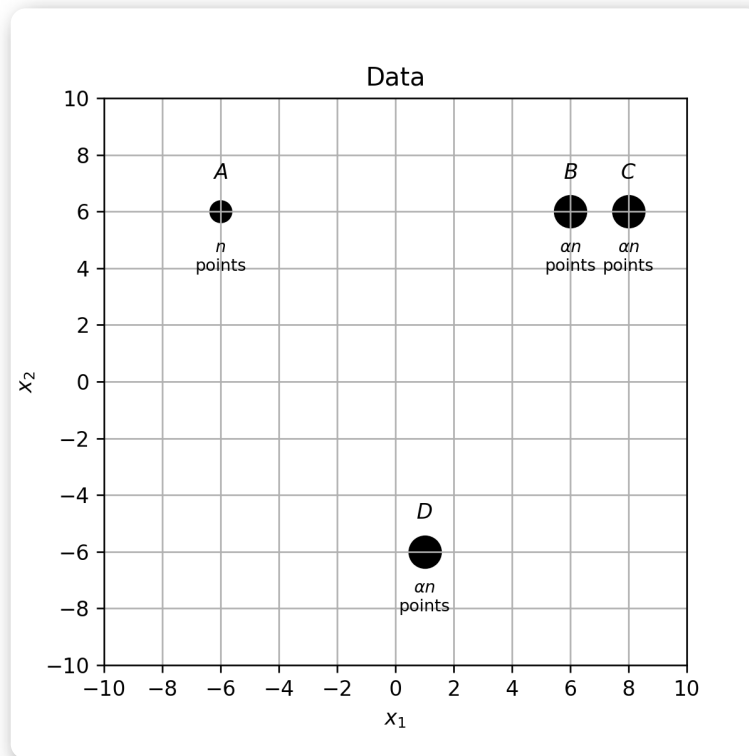
בעבור PCA עם $k = 2$ נקבל:

$$z = \frac{1}{\sqrt{5}}(1, -2)^T$$

תרגיל 12.2

נתונות n נקודות $(1 + 3\alpha)$ שונות:

- n נקודות בקואורדינטות $A = (-6, 6)$
- αn נקודות בכל אחת מהקואורדינטות $B = (6, 6), C = (8, 6), D = (1, -6)$



(הנקודות יושבות אחת על השניה בכל קואורדינטה, ומצויירות כעיגולים רק לצורך השרטוט). רוצים לבצע אשכול של הנקודות ל-3 אשכולות בעזרת K-Means.

(1) מאתחלים את המרכזים על ידי בחירה אקראית של 3 מתוך ארבעת הנקודות A, B, C, D. לאילו חלוקות יתכנס האלגוריתם בעבור כל אחת מארבעת האתחולים האפשריים.

(2) מהו האשכול האופטימאלי (הממזער את פונקציית המטרה)? רשמו את הפתרון כתלות בפרמטר α . (ניתן להניח כי בפתרון האופטימאלי כל הנקודות שנמצאות באותו המקום משויכות לאותו האשכול)

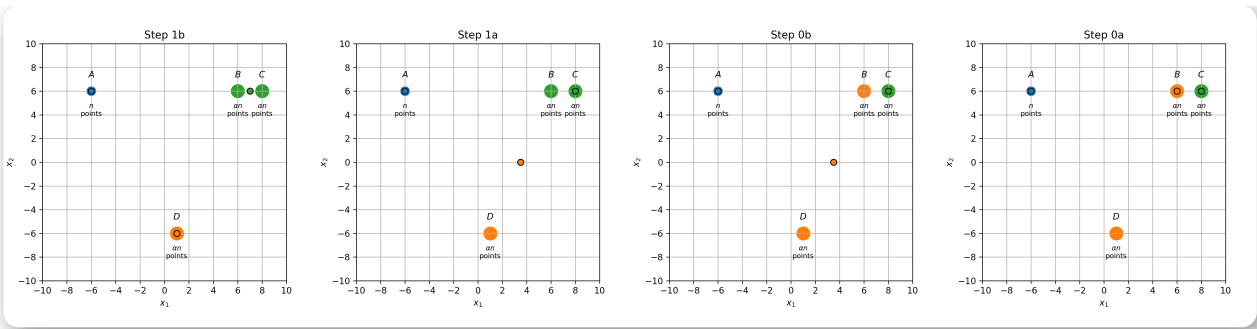
(3) האם קיים אתחול אשר בעבורו האלגוריתם לא יתכנס לפתרון האופטימאלי שמצאתם בסעיף הקודם? הדגמו.

פתרון 12.2

(1)

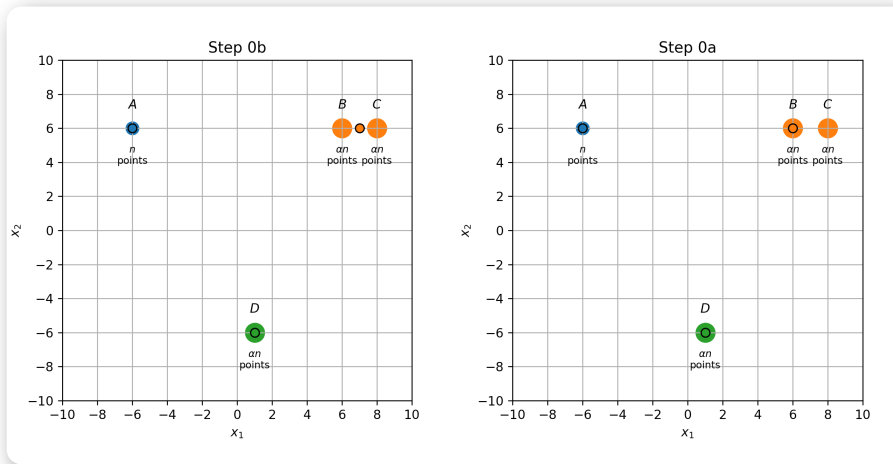
נחשב את תוצאת האלגוריתם בעבור כל אחת מארבעת האתחולים:

מרכזים ב A, B ו C:



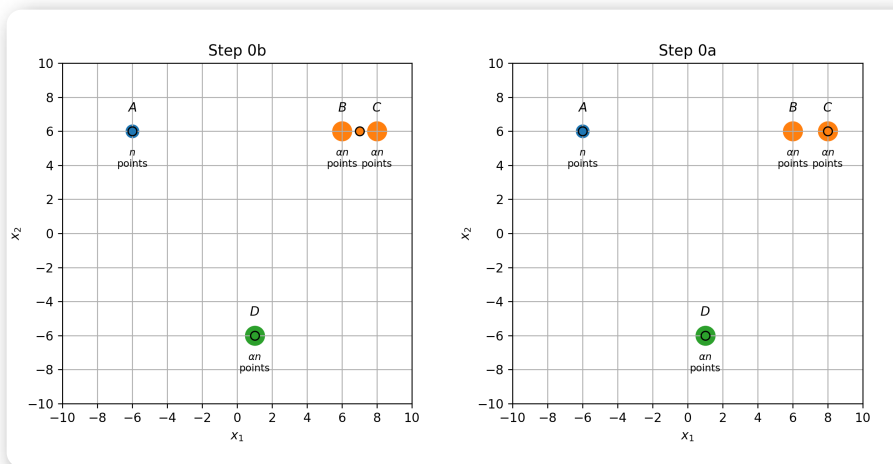
- שיוך התחלתי (0a): נקודות B, A ו C ישוויכו למרכז אשר הנמצא עליהם, והנקודות B ו D ישוויכו למרכז שב. D.
- עדכון מרכזים (0b): המרכז שב B יזוז לאמצע הדרך שבין הנקודות B ו D.
- עדכון אשכולות (1a): הנקודות שב B ישוויכו כעת למרכז שב.
- עדכון מרכזים (1b): המרכז שבין B ל D יזוז ל D, והמרכז שב B למחצית הדרך שבין B ל C.

מרכזים B, A ו D:



- שיוך התחלתי (0a): נקודות B, A ו D ישוויכו למרכז אשר נמצא עליהם, והנקודות B ו C ישוויכו למרכז שב.
- עדכון מרכזים (0b): המרכז שב B יזוז לאמצע הדרך שבין הנקודות B ו C.

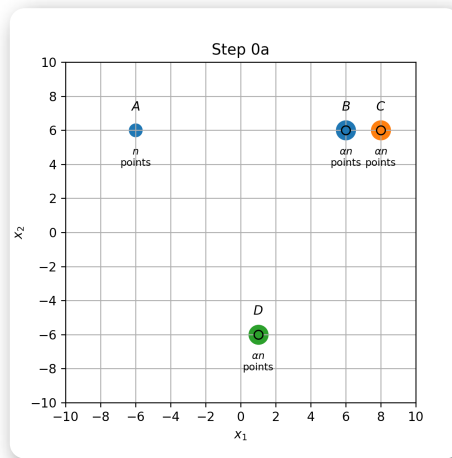
מרכזים B, A, C ו D:



- שיוך התחלתי (0a): נקודות B, A ו D ישוויכו למרכז אשר נמצא עליהם, והנקודות B ישוויכו למרכז שב.

- עדכון מרכזים (0b): המרכז שב C יזוז לאמצע הדרך שבין הנקודות B ו C.

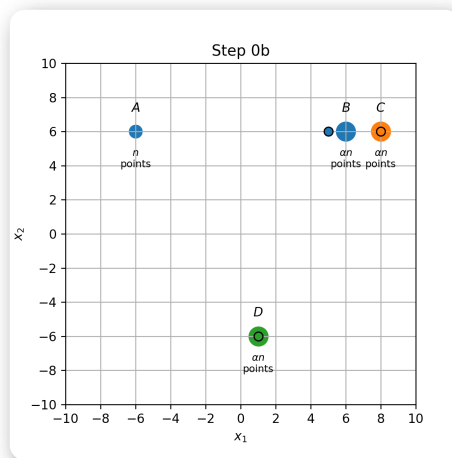
מרכזים ב B, C ו D:



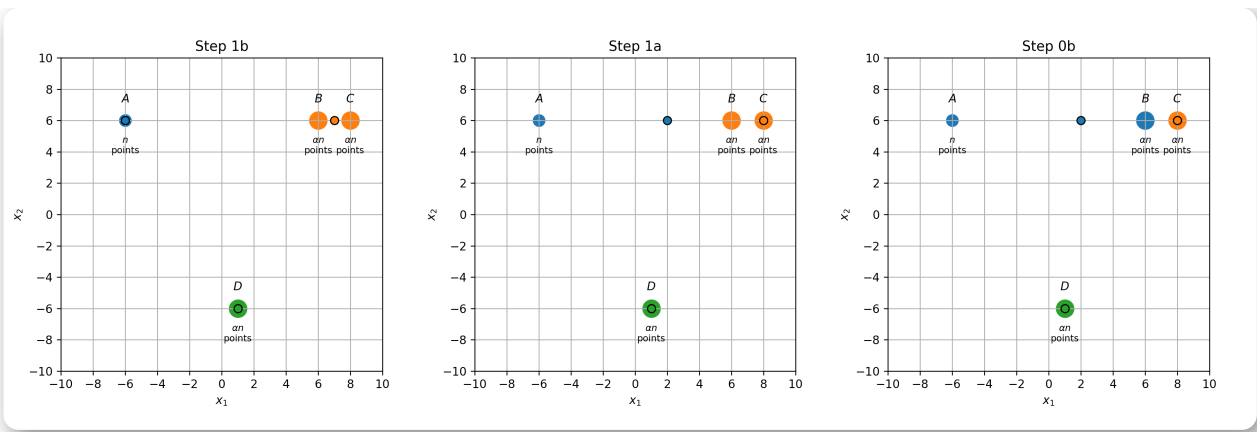
- שיוך התחלתי (0a): נקודות B, C ו D ישוייכו למרכז אשר נמצא עליהם, והנקודות A ישוייכו למרכז שב. B.
- עדכון מרכזים (0b): המרכז שב B יזוז לנקודה שהיא המרכז של הנקודות A ו B. (משום שכמות הנקודות בשתי הקבוצות שונה, נקודה זו היא לא אמצע הדרך ביניהם).

השלב הבא של עידכון האשכולות תלוי במיקום של המרכז החדש.

מקרה 1: הנקודות ב-B קרובות יותר למרכז החדש מאשר למרכז שב-C ולכן האלגוריתם מסתיים.



מקרה 2, המרכז החדש **רחוק** יותר לנקודה B מאשר הנקודה C, אזי הנקודות ב B יהיו מושייכות כעת למרכז בנקודה C, והמשך האלגוריתם יהיה:



נמצא את התנאי על α שבעבורו מתרחש מקרה 2. נסמן ב μ_1 את המרכז שבין A ל B לאחר עדכון המרכזים הראשון. המיקום של μ_1 נתון על ידי הממוצע המשוכלל של הקואורדינטות של A ו B:

$$\mu_1 = \frac{n\vec{A} + \alpha n\vec{B}}{(1 + \alpha)n} = \frac{(-6\hat{x}_1 + 6\hat{x}_2) + \alpha(6\hat{x}_1 + 6\hat{x}_2)}{1 + \alpha} = \frac{\alpha - 1}{\alpha + 1}6\hat{x}_1 + 6\hat{x}_2$$

על מנת שיתרחש עידכון, על המרחק בין המרכז החדש לנקודה B צריך להיות גדול מ 2:

$$\begin{aligned} \left\| (6\hat{x}_1 + 6\hat{x}_2) - \left(\frac{\alpha - 1}{\alpha + 1}6\hat{x}_1 + 6\hat{x}_2 \right) \right\| &> 2 \\ \Leftrightarrow 6 - \frac{\alpha - 1}{\alpha + 1}6 &> 2 \\ \Leftrightarrow \frac{\alpha - 1}{\alpha + 1}6 &< 4 \\ \Leftrightarrow \alpha &< 5 \end{aligned}$$

(2)

ב) אנו מועניינים למצוא את האשכול אשר מביא למינימום את הפונקציית המטרה הבאה:

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

נוכל לפסול פתרונות בהן ישנו אשכול ריק, משום שבמקרה זה נוכל לשייך אליו נקודות כלשהן על מנת להקטין את פונקציית המטרה. לכן הפתרון האופטימאלי חייב להיות אחד מששת האישכולים הבאים:

- (A,B), (C), (D) •
- (A,C), (B), (D) •
- (A,D), (B), (C) •
- (B,C), (A), (D) •
- (B,D), (A), (C) •
- (C,D), (A), (B) •

התרומה של האשכולות שמכילים קבוצה בודדת לפונקציית המטרה הינה 0, ולכן יש לחשב רק את התרומה של האשכול שמכיל שתי קבוצות של נקודות. למשל, עבור האשכול (A,B), (C), (D) נקבל:

$$\arg \min_{\{\mathcal{I}_j\}_{k=1}^K} \frac{1}{N} \sum_{k=1}^K \sum_{i \in \mathcal{I}_k} \|\mathbf{x}^{(i)} - \mu_k\|_2^2 = n \left(-6 - 6 \frac{\alpha - 1}{\alpha + 1} \right)^2 + \alpha n \left(6 - 6 \frac{\alpha - 1}{\alpha + 1} \right)^2 = n \cdot \frac{36}{(\alpha + 1)^2} (4\alpha^2 + 4\alpha)$$

ועבור האשכול (B,C), (A), (D) נקבל:

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

נחשב את הערך של פונקציית המטרה בעבור כל אחד מששת האשכולים:

Objective	Clusters
$144 \frac{\alpha n}{\alpha+1}$	(A,B), (C), (D)
$193 \frac{\alpha n}{\alpha+1}$	(A,C), (B), (D)
$196 \frac{\alpha n}{\alpha+1}$	(A,D), (B), (C)
$2\alpha n$	(B,C), (A), (D)
$30.5\alpha n$	(B,D), (A), (C)
$42.5\alpha n$	(C,D), (A), (B)

נשים לב כי הפתרון האופטימאלי יהיה חייב להיות (A,B),(C),(D) או (B,C),(A),(D) (משום שכל השאר בהכרח גדולים מהם). נבדוק בעבור אלו ערכים של α האשכול הראשון הינו האופטימאלי:

$$144 \frac{\alpha n}{\alpha+1} < 2\alpha n$$

$$\Leftrightarrow \alpha > 71$$

אם כן, בעבור $\alpha > 71$ הפתרון האופטימאלי הינו (A,B),(C),(D) ובעבור $\alpha < 71$ הפתרון האופטימאלי הינו (B,C),(A),(D).

נסכם כי עבור אתחול המרכזים בנקודות B,C ו-D נקבל:

- עבור $\alpha < 5$ האלגוריתם ישדך את B ו-C וזהו הפתרון האופטימאלי גלובלי.
- עבור $\alpha > 71$ האלגוריתם ישדך את A ו-B וזה הפתרון האופטימאלי גלובלי.
- עבור $5 < \alpha < 71$ האלגוריתם ישדך את A ו-B אולם זהו אינו הפתרון הגלובלי.

נבדוק בעבור אתחולים מהסעיף הקודם, מהם המקרים שבהם האלגוריתם אינו מתכנס לפתרון האופטימאלי:

- בעבור $\alpha > 71$ הפתרון האופטימאלי הינו (A,B),(C),(D), אך עבור 3 מתוך 4 האיחולים שבדקנו האלגוריתם התכנס לפתרון של (B,C),(A),(D).
- בעבור $\alpha < 71$ הפתרון האופטימאלי הינו (B,C),(A),(D), אך במקרה של $\alpha > 5$ ואתחול של מרכזים B, C ו-D מתקבל הפתרון של (A,B),(C),(D).

ג) כל המקרים שצויינו בסעיף הקודם. בנוסף, ניתן לדוגמא לאתחל שניים מתוך שלושת המרכזים בנקודות מאד רחוקות, ואת כל הנקודות ישוייכו למרכז השלישי.

חלק מעשי - מיקום חניונים בניו יורק

Code

תזכורת: מדגם נסיעות המונית ב New York

נחזור למדגם של נסיעות מונית בניו-יורק בו השתמשנו בתרגולים הראשונים לחיזוי זמן הנסיעה. נציג את 10 הדגימות הראשונות במדגם (סה"כ במדגם זה 100,000 נסיעות).

id	duration	dropoff northing	dropoff easting	pickup northing	pickup easting	tip amount	fare amount	payment type	trip distance	passenger count
3	11.5167	4515.18	588.155	4512.98	586.997	0	9.5	2	2.76806	2

ay of ek	duration	dropoff northing	dropoff easting	pickup northing	pickup easting	tip amount	fare amount	payment type	trip distance	passenger count	
6	12.6667	4512.63	584.85	4512.92	587.152	0	10	2	3.21868	1	1
0	5.51667	4513.17	585.434	4513.36	587.005	2.49	7	1	2.57494	1	2
1	9.88333	4512.55	586.672	4511.73	586.649	1.65	7.5	1	0.965604	1	3
2	8.68333	4511.76	585.262	4511.89	586.967	1.66	7.5	1	2.46229	1	4
3	9.43333	4511.54	585.169	4512.88	585.926	2.2	7.5	1	1.56106	5	5
5	7.95	4514.21	588.71	4515.08	586.731	1	8	1	2.57494	1	6
5	4.95	4509.55	585.844	4509.71	585.345	0	5	2	0.80467	1	7
5	11.0667	4507.74	583.671	4509.48	585.422	1.1	10	1	3.6532	1	8
3	4.21667	4513.71	587.701	4514.93	587.875	1.36	5.5	1	1.62543	6	9

הבעיה: מציאת חניונים

חברת מוניות רוצה לשכור K מגרשי חניה ברחבי העיר NYC בהם יוכלו לחכות המוניות שלה בין הנסיעות.

לשם כך היא מעוניינת לבחור באופן אופטימאלי את המיקומים של מגרשי החניות האלו כך שהמרחק הממוצע מנקודת הורדת הנוסע למרגש החניה הקרוב יהיה מינימאלי.

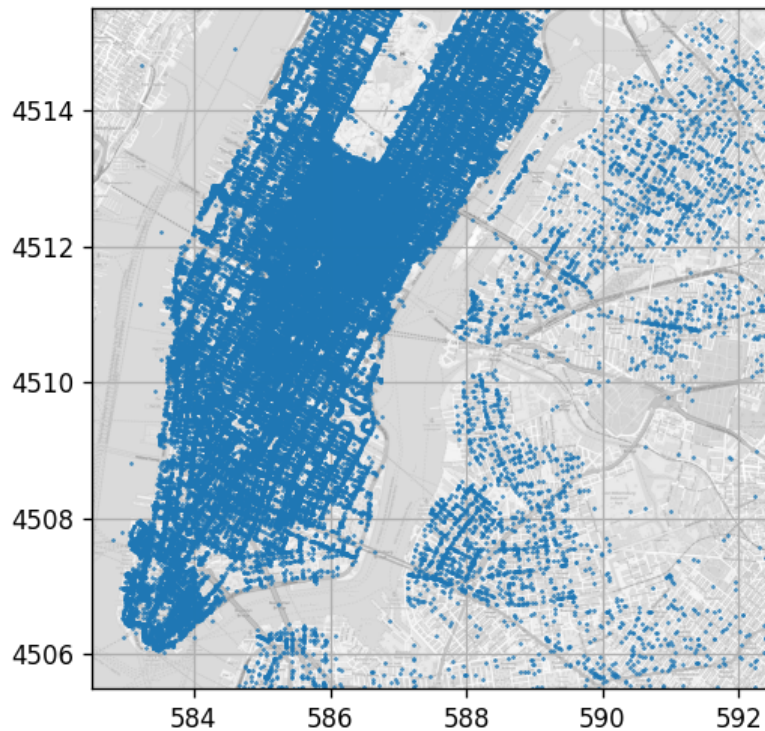
שדות רלוונטיים

הפעם נתמקד בשני השדות הבאים מהמדגם:

- **dropoff_easting** - הקואורדינאטה האורכית (מזרח-מערב) של סיום הנסיעה
- **dropoff_northing** - הקואורדינאטה הרוחבית (צפון-דרום) של סיום הנסיעה

(למתעניינים: הקואורדינאטות נתונות ב-UTM-WGS84, היחידות הן בקירוב קילומטר).

ויזואליזציה של נקודות ההורדה



הגדרה פורמאלית של הבעיה

נשתמש בסימונים הבאים:

- \mathbf{x} הוקטור האקראי של מיקום סיום הנסיעה
- N : מספר הנסיעות במדגם.
- $\mathbf{x}^{(i)}$ הוקטור של מיקום סיום הנסיעה ה- i .
- \mathbf{c}_k : המיקום של מגרש החניה ה- k .

המטרה: למצוא את מיקומי החניונים האופטימאליים אשר ממזערים את הגודל הבא

$$\{\mathbf{c}_k\}_{k=1}^{K^*} = \arg \min_{\{\mathbf{c}_k\}_{k=1}^K} \mathbb{E} \left[\min_k \|\mathbf{x} - \mathbf{c}_k\|_2 \right]$$

מכיוון שהפילוג האמיתי של \mathbf{x} לא ידוע ננסה למזער את התוחלת האמפירית:

$$\{\mathbf{c}_k\}_{k=1}^{K^*} = \arg \min_{\{\mathbf{c}_k\}_{k=1}^K} \frac{1}{N} \sum_i \min_k \|\mathbf{x}^{(i)} - \mathbf{c}_k\|_2$$

נרשום את הבעיה על ידי חלוקת המדגם לאשכולות. נגדיר את האשכול \mathcal{I}_k כאוסף של כל הנסיעות שהחניון ה- k הוא הקרוב ביותר לנקודת הסיום שלהן. באופן זה נוכל לרשום את בעיית האופטימיזציה באופן הבא:

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

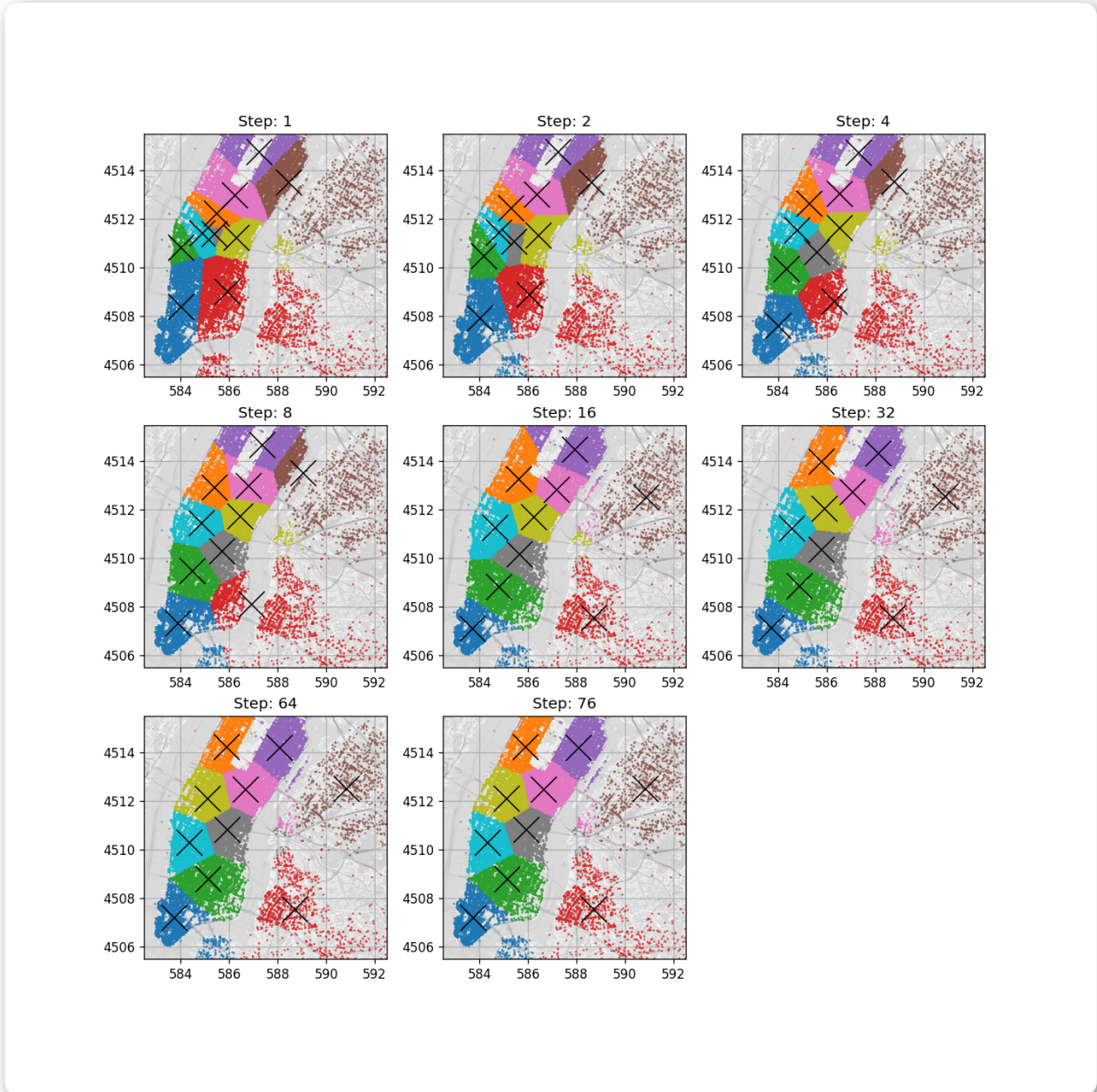
פתרון באמצעות K-Means

נשים לב כי הבעיה שקיבלנו דומה מאד לבעיה אותה K-Means מנסה לפתור, עם הבדל משמעותי אחד. K-Means ממזער את המרחק הריבועי הממוצע בעוד שאנו מחפשים למזער את המרחק האוקלידי. ישנם אלגוריתמים מורכבים יותר אשר

פותרים את הבעיה שלנו, אך לבינתיים נשאר עם K-Means.

נציין שזהו מצב נפוץ שבו איננו מסוגלים לפתור בעיה מסויימת באופן ישיר אז אנו פותרים בעיה דומה לה בתקווה לקבל תוצאות מספקות, אך לא בהכרח אופטימאליות.

נשתמש באלגוריתם K-means על מנת לבחור את המיקום של 10 מגרשי חניה.



המרחק נסיעה הממוצע המתקבל הינו:

$$\frac{1}{N} \sum_{k=1}^K \sum_{i \in \mathcal{I}_k} \|\mathbf{x}^{(i)} - \mathbf{c}_k\|_2 = 700m$$

חשוב לציין שהפתרון הזה הוא לא בהכרח הפתרון האופטימאלי משתי סיבות:

1. K-Mean לא מבטיח התכנסות למינימום הגלובלי. דרך אחת לשפר את תוצאות האלגוריתם הינה להריץ אותו מספר פעמים עם איתחולים שונים.
2. כפי שצינו קודם K-Mean ממזער את השגיאה הריבועית הממוצעת. ניתן אם כן לשפר קלות את התוצאות על ידי שמירה על האשכולות אך תיקון המרכז לנקודה אשר ממזערת את המרחק עצמו.

הערה הנקודה אשר ממזערת את המרחק האוקלידי (בלי הריבוע) בינה לבין כל הנקודות באשכול נקראת החציון הגיאומטרי ([The Geometric Median](#) wiki). ניתן למצוא נקודה זו על ידי שימוש באלגוריתם המוכונה *Weiszfeld's algorithm*.

מציאת מספר החניונים האופטימאלי

עד כה השתמשנו ב-10 חניונים, נרצה כעת לבחור גם מספר זה בצורה מיטבית. באופן כללי ככל שנגדיל את מספר החניונים מרחק הנסיעה לחניונים יקטן, אך מנגד התחזוקה של כל חניון עולה כסף.

נניח כי:

1. עלות האחזקה של חניון הינה \$10k לחודש.
2. בכל חודש יהיו בדיוק 100k נסיעות.
3. עלות הנסיעה של מונית בדרך לחניון הינה \$3 לקילומטר.

נרשום תחת הנחות אלו את העלות החודשית של אחזקת החניונים והנסיעה אליהם:

$$10 \cdot K + 100 \cdot 3 \cdot \mathbb{E} \left[\min_k \| \mathbf{x} - \mathbf{c}_k \|_2 \right]$$

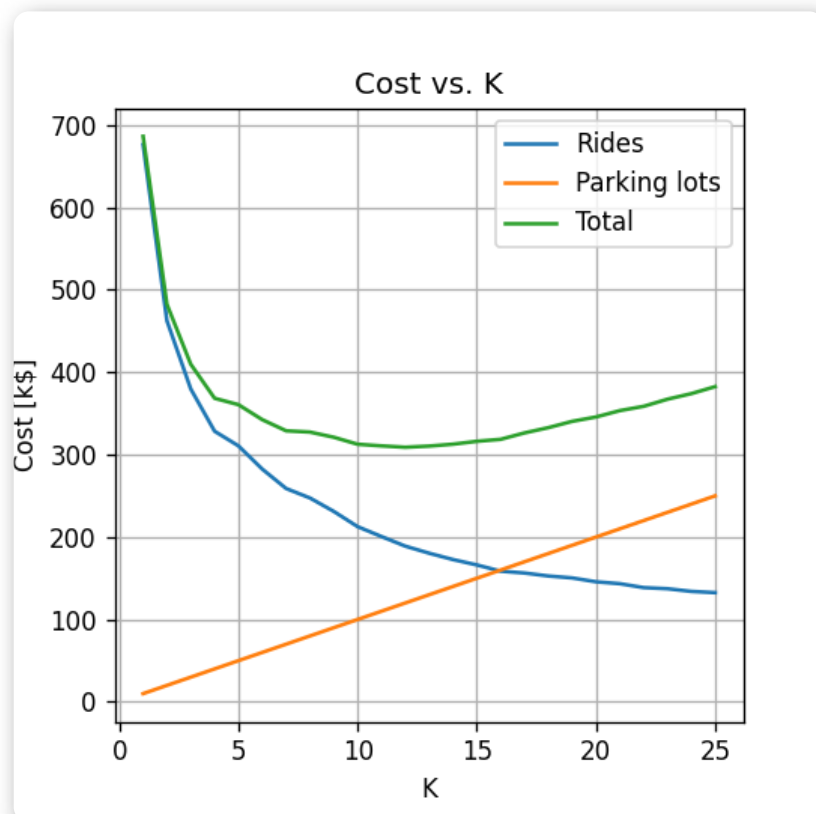
והמקבילה האמפירית:

$$10 \cdot K + 100 \cdot 3 \cdot \frac{1}{N} \sum_{k=1}^K \sum_{i \in \mathcal{I}_k} \| \mathbf{x}^{(i)} - \mathbf{c}_k \|_2$$

מספר החניונים כ Hyper parameter

כעת עלינו לבצע אופטימיזציה גם על מספר החניונים וגם המיקום שלהם. ראינו כיצד ניתן למצוא פתרון בעבור K נתון, אך אין לנו דרך פשוטה להכליל את זה ל K כלשהו. נוכל לעבור על כל ערכי K הרלוונטיים, לפתור את הבעיה עבורם ולבסוף לקחת את הפתרון הטוב ביותר. K הוא למעשה hyper-parameter של הבעיה.

נריץ את אלגוריתם ה K-Means בעבור כל ערך של $K \in [1, 25]$, נשרטט את עלות הנסיעה, עלות אחזקת החניונים והעלות הכוללת:



נקבל כי:

- מספר החניונים האופטימאלי הינו: 12.
- מרחק הנסיעה הממוצע יהיה 630 מ'.
- העלות הכוללת תהיה \$308.12k לחודש.