machine learning

K-Means

Lecture VII

פיתוח:
ד"ר יהונתן שלר
משה פרידמן

מוטיבציה – חלוקת קבוצת לקוחות לקבוצות

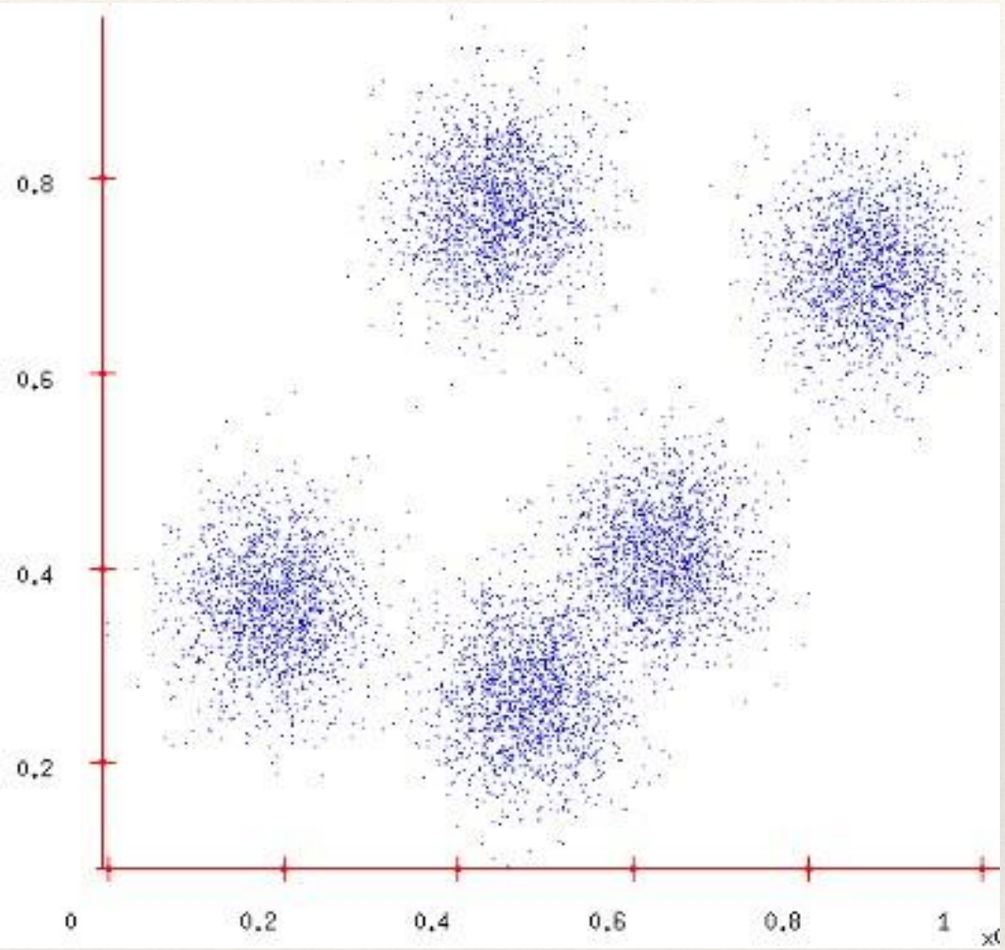
- למנהל קשרי לקוחות יש חמישה עובדים ורוצה לחלק את הלקוחות ל-5 קבוצות כך שכל קבוצה תשויך למנהל אחר.
  - האתגר שלנו – ל"חשוף" 5 קבוצות "מעניינות"
  - הבעיה: אין לנו את ה- class label של כל קבוצה
  - איך נעשה זאת?
    - לפי גיל? לפי צבע בגדים? לפי גובה?

שיטות מבוססות חלוקה
ואלגוריתם k-means

גישה 1: שיטות מבוססות חלוקה

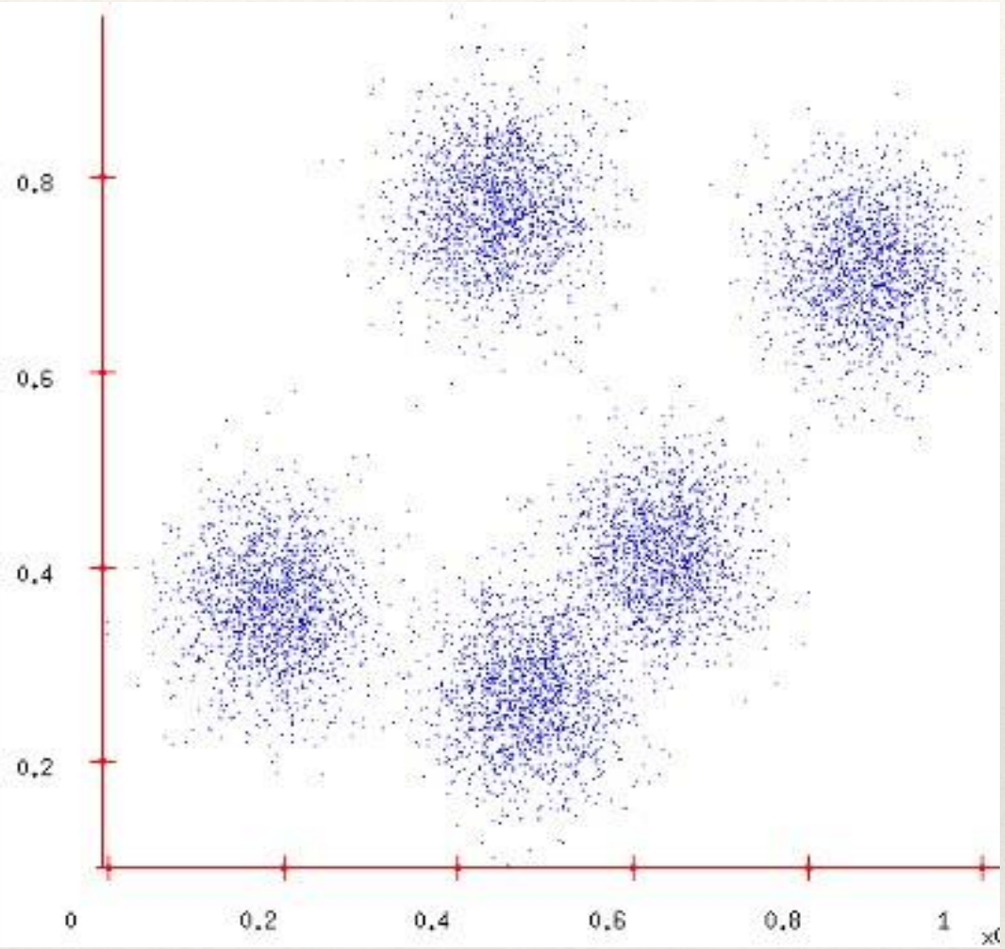
- המטרה - בהינתן K (כקלט לאלגוריתם) - מצא חלוקה ל-K אשכולות שמביאה אותנו לאופטימיזציה
- השאיפה - אופטימום גלובלי – עבור על כל החלוקות האפשריות ובחר את הטובה ביותר.
- היצוג - כל אשכול מיוצג ע"י אב-טיפוס (prototype) - מרכז האשכול

Clustering



- החלוקה של n איברים לתוך K קבוצות – נחשבת לאחת מאבות הטיפוס של למידה לא מונחית.
- **הנחת היסוד** היא, שהנתונים נוצרו ממספר מחלקות שונות
- **המטרה** היא לקבץ נתונים שנוצרו ממחלקות זהות לתוך אותו קלסטר.

Clustering



- החלוקה של n איברים לתוך K קבוצות – נחשבת לאחת מאבות הטיפוס של למידה לא מונחית.
- **הנחת היסוד** היא, שהנתונים נוצרו ממספר מחלקות שונות
- **המטרה** היא לקבץ נתונים שנוצרו ממחלקות זהות לתוך אותו קלסטר.

שאלות שנובעות מהנחות אלו:

- כמה מחלקות יש?
- למה בעצם שלא נשייך כל נתון לתוך מחלקה בפני עצמה?
- מהי פונקציית המטרה שאנחנו רוצים למקסם ב-clustering?

גישה 1: שיטות מבוססות 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.

גישה 1: שיטות מבוססות חלוקה
K-Means

K-Means – אחד האלגוריתמים הפשוטים והנפוצים עבור clustering

- הומצא על יד Lloyd, 1957
- הרעיון – למצוא אוסף של prototypes, המייצגים את ה-clusters.
- הדרך בה ה-prototype מייצג את מרכז ה-cluster רמוזה על ידי שם האלגוריתם K-Means (כפי שנראה בהמשך).

גישה 1: שיטות מבוססות חלוקה
K-Means - המטרה

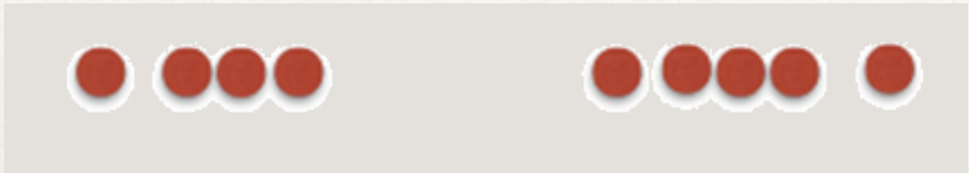
- נניח שמספר ה-clusters הוא k
  - הנחה של prototype אחד עבור כל cluster
  - נסמן את ה-prototypes ע"י $\mu_1, \ldots, \mu_k$ (כזכור μ מייצג תוחלת).
  - לעיתים מסמנים את ה-prototype כ-m (המסמן mean – ממוצע), או ע"י c (המסמן center – מרכז).

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

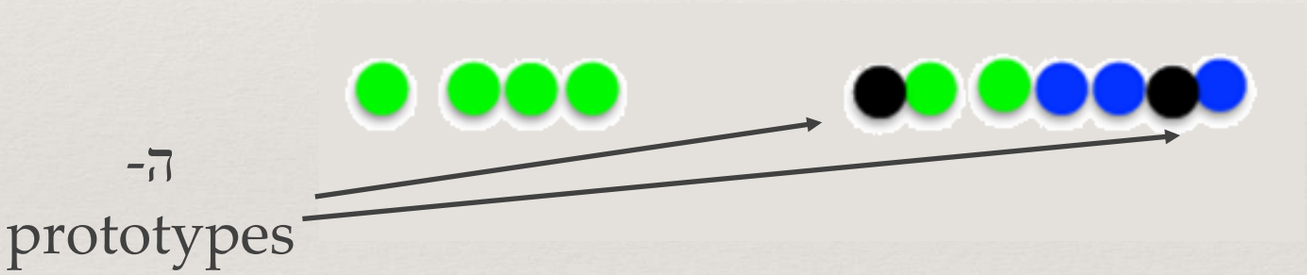
גישה 1: שיטות מבוססות חלוקה
K-Means - המטרה

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

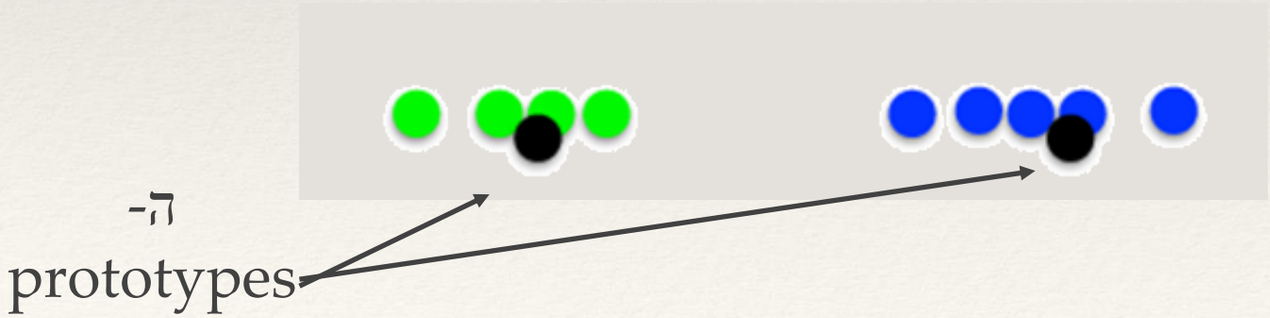
ה-data



דוגמה ל- prototypes שאינם איכותיים



דוגמה ל- prototypes איכותיים



גישה 1: שיטות מבוססות חלוקה
K-Means – המטרה – מינימיזציה של המרחק

- עבור נקודה x_i, נגדיר את המרחק ל-prototype הקרוב ביותר:

$$d(x_i, \mu) = \min_j ||x_i - \mu_j||^2$$

- פונקצית המטרה – למצוא $f(\mu)$, הממזער את:

$$f(\mu) = \sum_i d(x_i, \mu)$$

- הפונקציה אינה פוקציה קמורה (פונקצית convex).
  - כלומר, המינימום המקומי אינו בהכרח המינימום הגלובלי
  - אין פונקציה פשוטה לפתור את ה-optimum (כמו gradient descent ברגרסיה לינארית).
- אלגוריתם K-means משתפר בכל צעד (מבחינת פונקצית המטרה)

פונקציות מרחק שניתן להשתמש לצורך clustering

<u>ב K-means משתמשים בעיקר במרחק אוקלידי:</u> $d(\overrightarrow{x_j},\overrightarrow{x_i})=\sum_{m=1}^{d}\sqrt{(x_{j_m}-x_{i_m})^2}$

פונקציות מיניקובסקי:

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

אפשרויות נוספות:

- <u>Cosine similarity</u>: $d(\overrightarrow{x_j},\overrightarrow{x_i})=\frac{\overrightarrow{x_j}^T\cdot\overrightarrow{x_i}}{\|\overrightarrow{x_j}\|\cdot\|\overrightarrow{x_i}\|}$
- Edit distance

K-means – רגע לפני ...

- לפני הרצת אלגוריתם k-means, בד"כ נרצה להריץ scaling, מדוע? ...

K-means - scaling

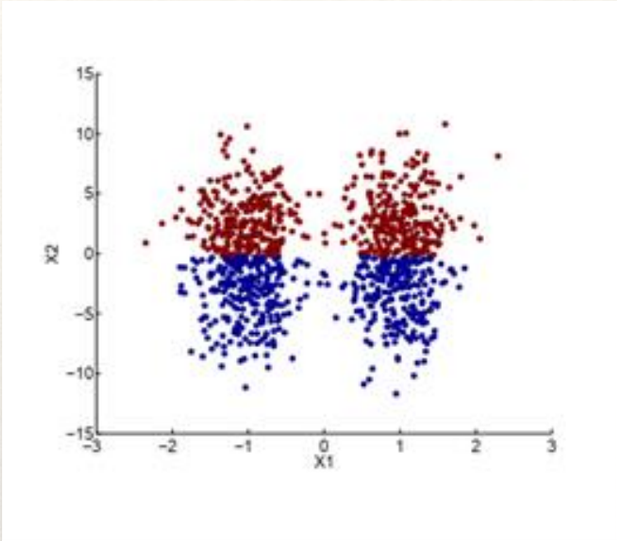
<u>Importance</u>:

- No scaling let's attributes with high variability / range to dominate the metric
- K-means performed well on Normal distributed data

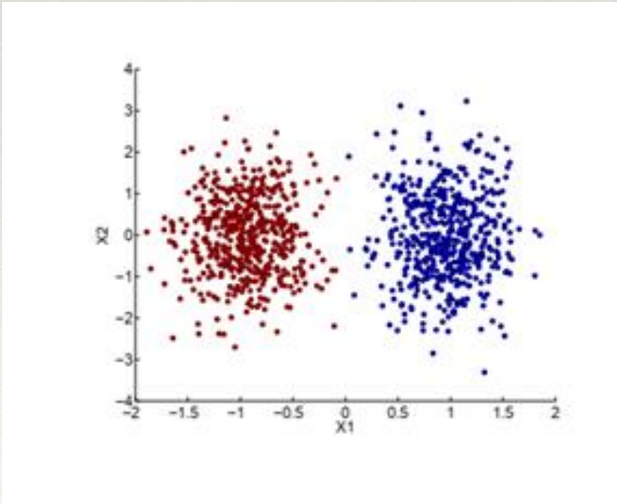
Standardization w/t-scale:

$$t_{attr\_val} = \frac{attr\_val - mean(attr)}{std_{sample}(attr)}$$

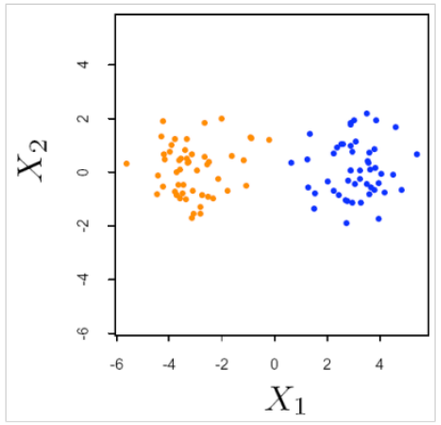
Without scaling



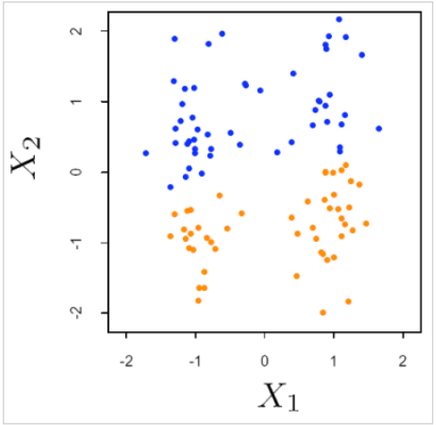
With scaling



K-means - scaling



Without standardization



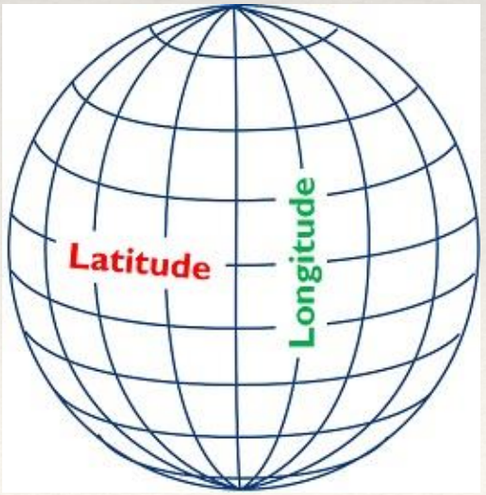
With standardization

<u>Risk</u>: Standardization discards information.

- If that information is irrelevant, then standardizing cases can be quite helpful.
- If that information is important, then standardizing cases can be disastrous.

For instance:

- If you have attributes with a well-defined meaning. Say, latitude and longitude, then you should not scale your data, because this will cause distortion..



אלגוריתם K-means

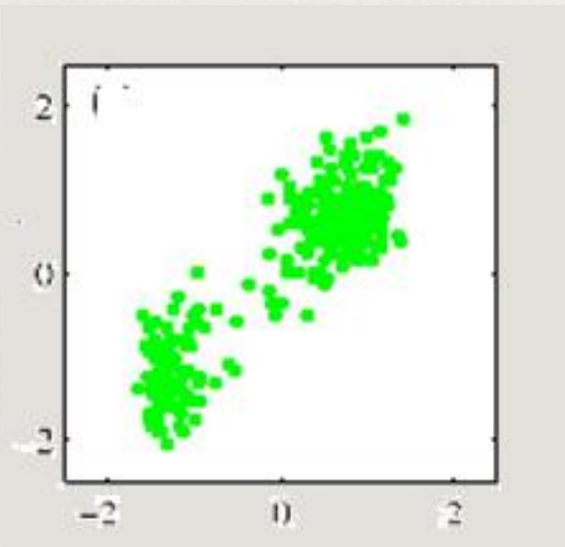
- נתון: אוסף ווקטורים ופרמטר K
- מצא חלוקה אופטימאלית שמחלקת ל-K אשכולות

- 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)

אלגוריתם K-means - דוגמה

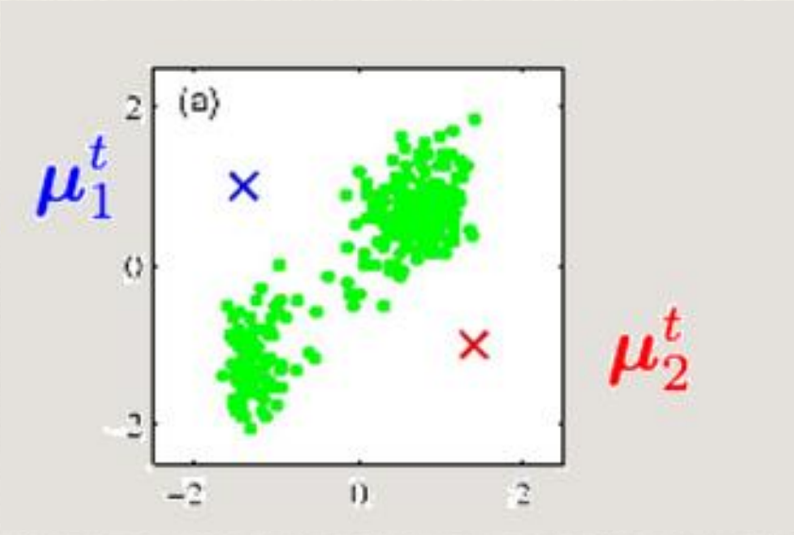
- נתון: אוסף ווקטורים ופרמטר K
- מצא חלוקה אופטימאלית שמחלקת ל-K אשכולות
  - נתון k=2 (2 clusters צריך למצוא 2 prototypes)



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

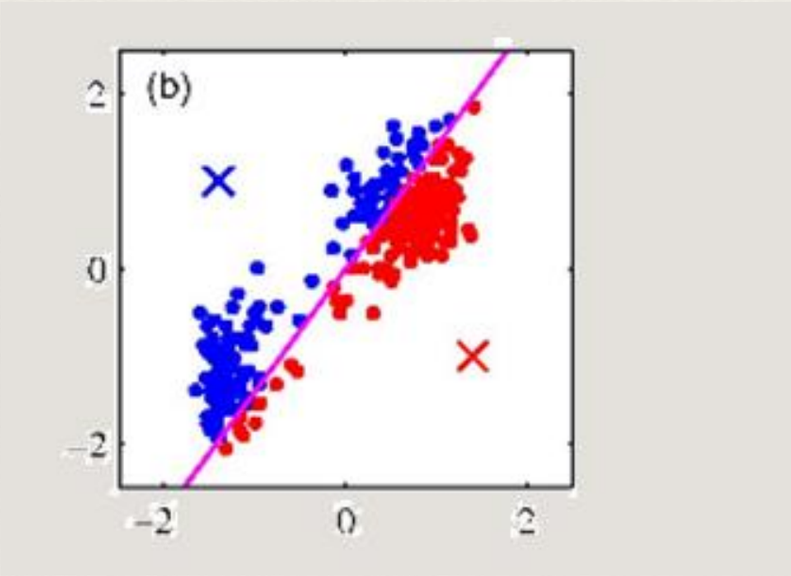
אלגוריתם K-means - דוגמה

- 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>:

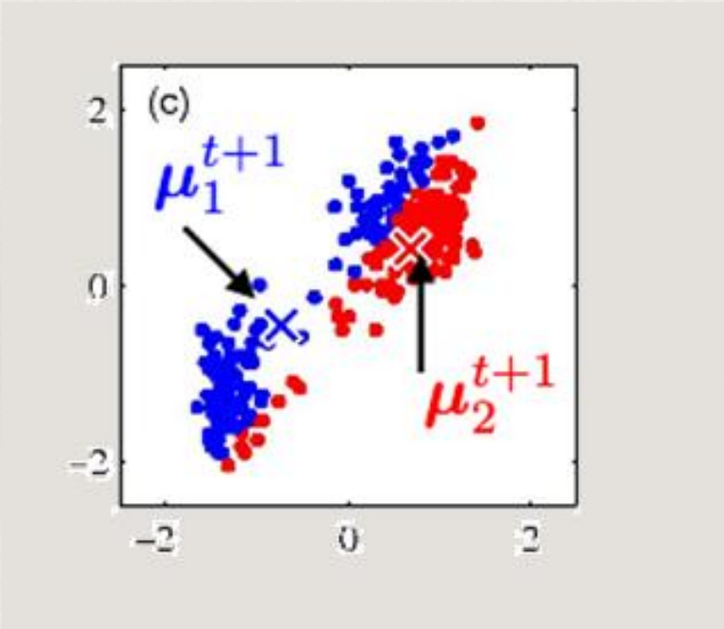


1. "נחש" K מרכזים
2. שייך כל ווקטור ל"מרכז" הקרוב אליו
3.
4. חזור על צעדים 2-3 עד שאין יותר עדכונים

אלגוריתם K-means - דוגמה

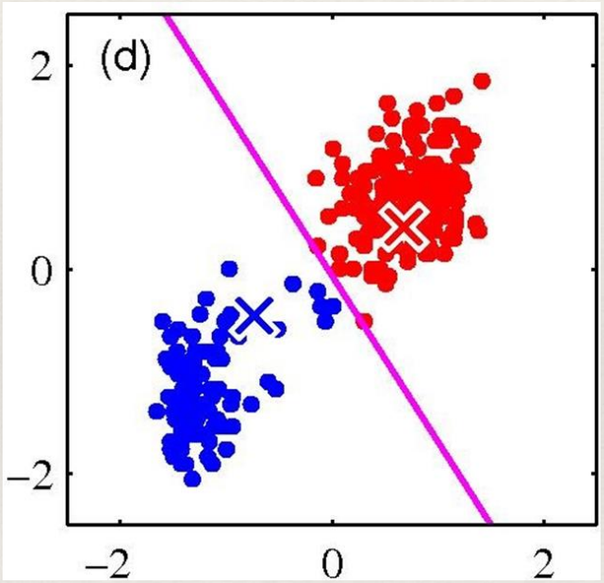
- נתון: אוסף ווקטורים ופרמטר K
- מצא חלוקה אופטימאלית שמחלקת ל-K אשכולות
  - נתון k=2 (2 clusters צריך למצוא 2 prototypes)
- <u>אלגוריתם</u>:

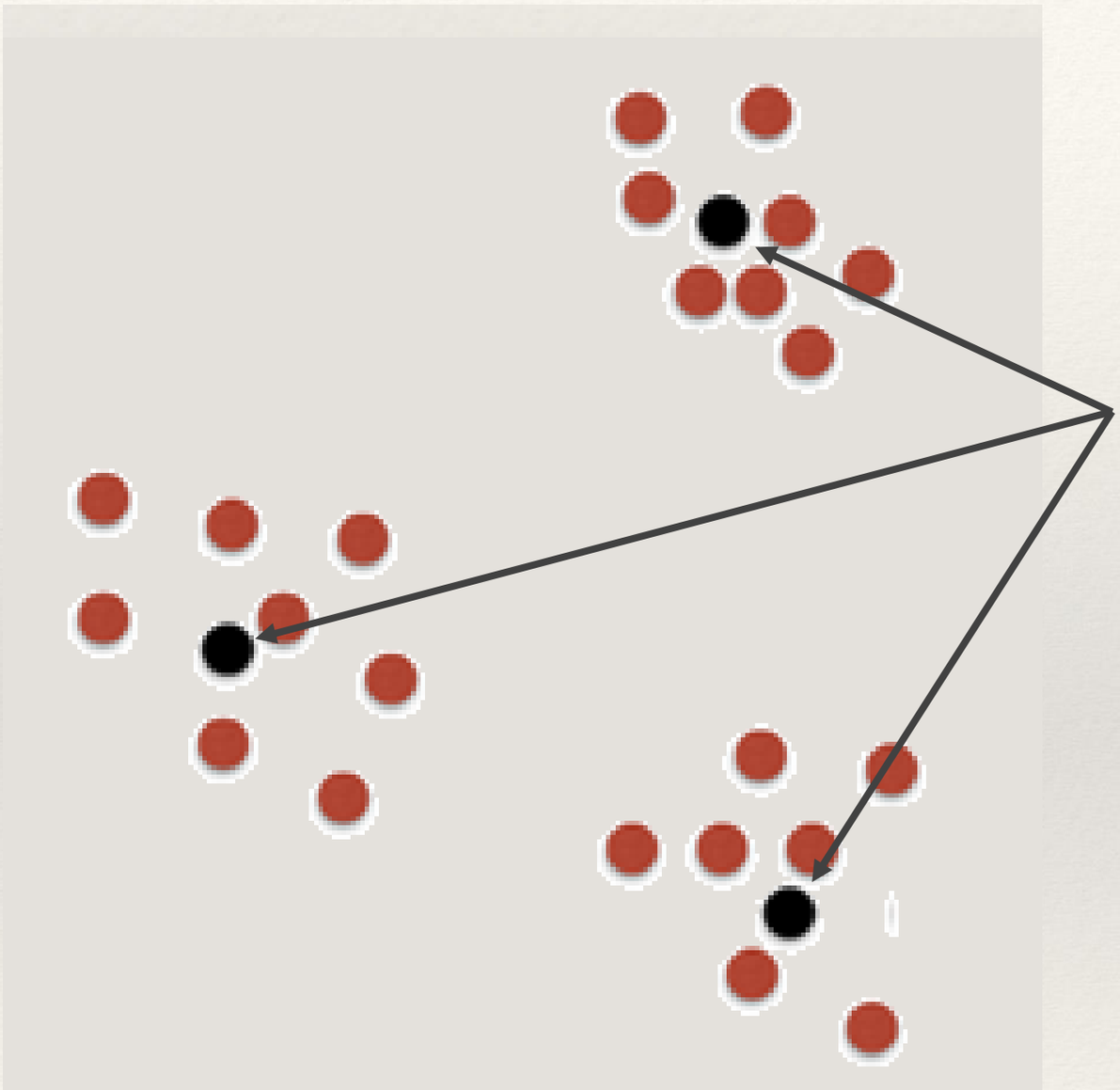
1. "נחש" K מרכזים
2. שייך כל ווקטור ל"מרכז" הקרוב אליו
3. חשב מרכזים מחדש ע"י מציאת מרכז האשכול
4. חזור על צעדים 2-3 עד שאין יותר עדכונים



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)





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

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

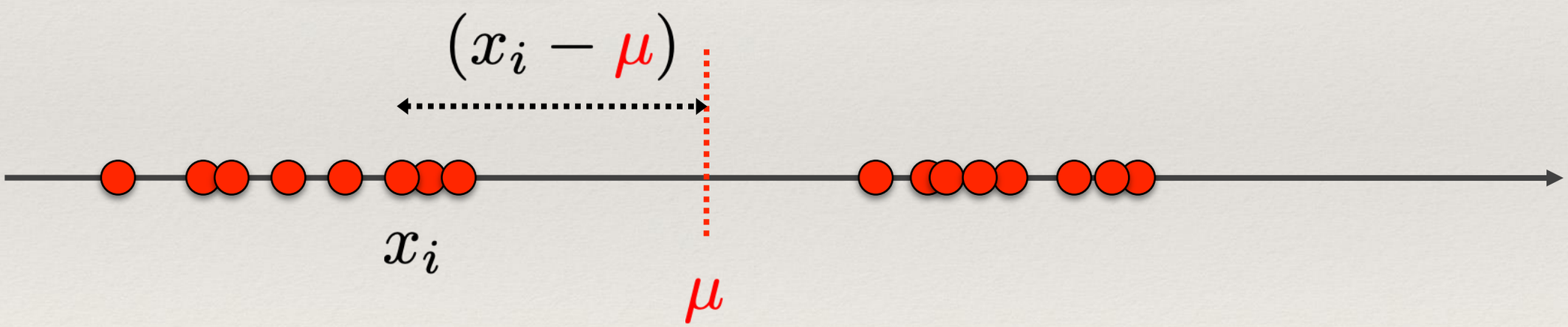
נתון מדגם של n דוגמאות: $x_1, x_2, , \ldots, x_n \quad ; \quad x_i \in \mathbb{R}$

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

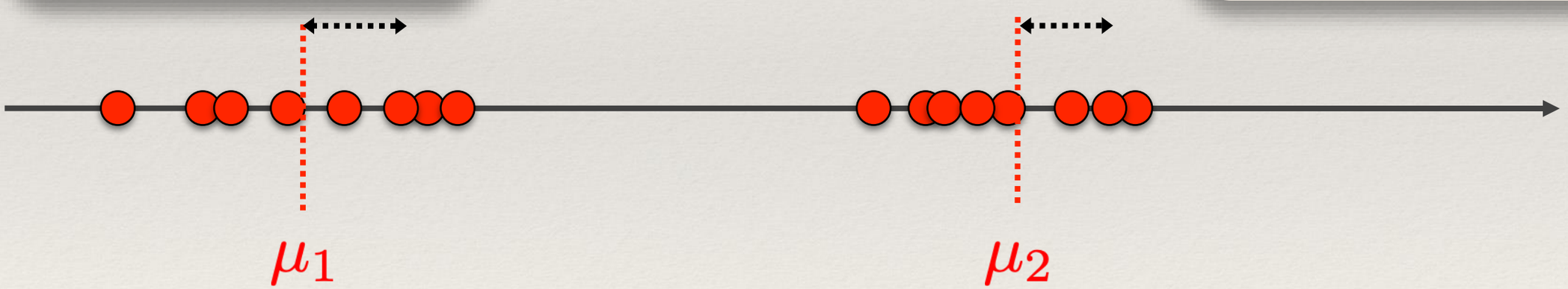
$$\mu_2 = \frac{1}{n_2}\sum_{i\in C_2} x_i$$

הפיזור ב-clusters

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

המטרה: פיזור מינימלי

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



שאלת סקר

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.

K-Means – פונקציית המטרה (objective function)

$\widehat{y_i}$ - נסמן את ה-cluster שנשייך אליו את דוגמה x_i כ- $\widehat{y_i}$, כאשר $\widehat{y_i}\in\{1,\dots,k\}$

עבור כל cluster, קיים וקטור מייצג - prototype : $\overrightarrow{\mu_1},\dots,\overrightarrow{\mu_k}$

פונקצית המחיר – הפיזור המשותף: $J=\sum_{j=1}^{k}\sum_{\widehat{y_i}=j}\sigma_j{}^2$

$$=\sum_{j=1}^{k}\sum_{\widehat{y_i}=j}||x_i-\mu_j||^2$$

- בעית אופטימיזציה: $\min[J]=min_{\{\widehat{y_i},\mu_j\}}\left[\sum_{j=1}^{k}\sum_{\widehat{y_i}=j}||x_i-\mu_j||^2\right]$
- בעיה NP-hard ולכן נפתור בשלבים:
  - מינימיזציה של $\{\widehat{y_i}\}$ - שיוך למרכזים המייצגים (שלב 2 ב-k-means)
  - מינימיזציה של $\{\mu_j\}$ - מציאת המרכזים המייצגים (שלב 3 ב-k-means)

K-Means – פונקציית המטרה (objective function) – בעיית המינימיזציה

פונקצית המחיר – הפיזור המשותף: $J = \sum_{j=1}^{k} \sum_{\widehat{y_i}=j} ||x_i - \mu_j||^2$

$r_{i,j}$ - נגדיר פונקצית עזר

$$\boxed{r_{i,j} = \begin{cases} 1 & \widehat{y_i} = j \\ 0 & \widehat{y_i} \neq j \end{cases}}$$

המטרה של פונקצית r – בחירת השיוך הטוב ביותר עבור כל וקטור i.

כעת נגדיר את פונקצית המחיר כך:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} r_{i,j} \cdot ||x_i - \mu_j||^2$$

K-Means – פונקציית המטרה (objective function) – חלק א של המינימיזציה –שיוכיים למרכזים המייצגים

$$r_{i,j} = \begin{cases} 1 & \widehat{y_i} = j \\ 0 & \widehat{y_i} \neq j \end{cases}$$

$$\mathrm{J} = \sum_{j=1}^{k} \sum_{i=1}^{n} r_{i,j} \cdot ||x_i - \mu_j||^2$$

בעיית מינימיזציה א':

בהינתן המרכזים $\{\mu_j\}$

$$min_{\{\widehat{y_i}\}} \left[\sum_{j=1}^{k} \sum_{i=1}^{n} r_{i,j} \cdot ||x_i - \mu_j||^2\right]$$

לחלופין ניתן להגדיר את מינימיזציה א' כך (בהינתן המרכזים $\{\mu_j\}$):

$$min_{\{r_{i,j}\}} \left[\sum_{j=1}^{k} \sum_{i=1}^{n} r_{i,j} \cdot ||x_i - \mu_j||^2\right]$$

K-Means – פונקציית המטרה (objective function) – חלק ב של המינימיזציה – מציאת המרכזים המייצגים

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

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} r_{i,j} \cdot ||x_i - \mu_j||^2$$

בעיית מינימיזציה ב':

בהינתן השיוכיים $\{r_{i,j}\}$

$$min_{\{\mu_j\}} \left[\sum_{j=1}^{k} \sum_{i=1}^{n} r_{i,j} \cdot ||x_i - \mu_j||^2\right]$$

שאלת סקר

לגבי k means איזו מהטענות הבאות נכונה:

א. עבור דאטה סט מסוים, ככל שמגדילים את ה k, הסטיה מהסנטרואידים קטנה

ב. עבור דאטה סט גדול מספיק k-means יתכנס לאופטימום הגלובלי ללא קשר לאיתחול שלו

ג. כל התשובות נכונות

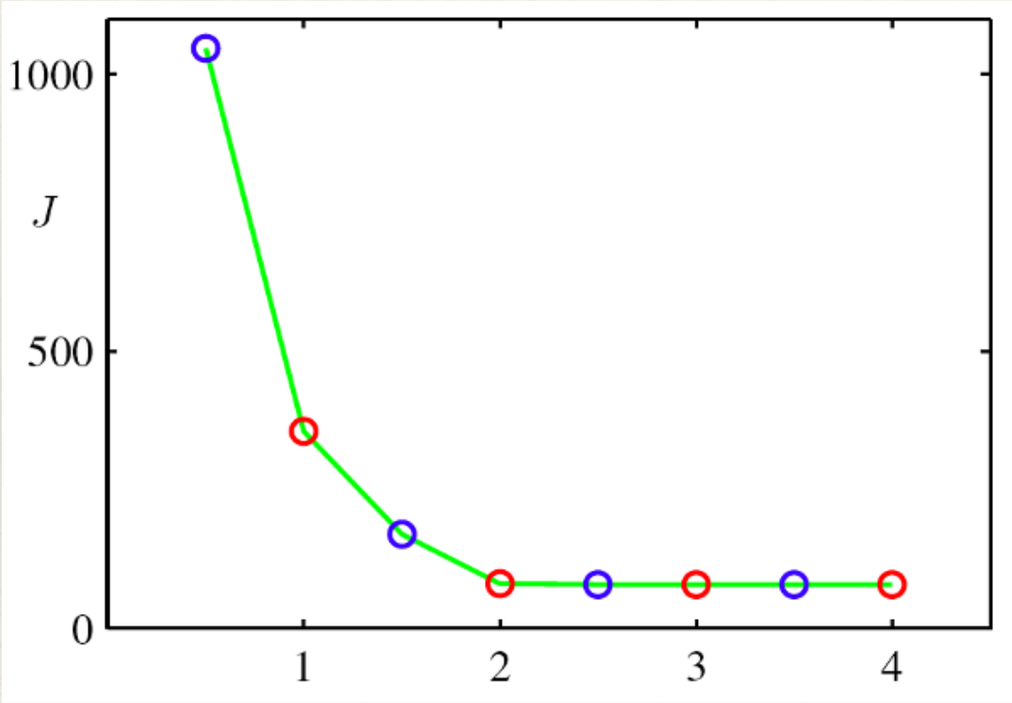
- <u>תשובה</u> – א.

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).
- בשיעור הבא נלמד שיטה אחרת - kmeans++

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



מדוע מובטח לנו שהתהליך יתכנס?

- יש p(n,k) אפשרויות לחלק n ווקטורים ל-K קבוצות
  - כלומר, יש מס' (גדול, אך) סופי של מרכזים אפשריים
- אם השתנתה הקונפיגורציה (שינוי מרכזים/שיוך נקודות מחדש למרכזים) סימן שיש לנו "סטייה" קטנה יותר מאשר הייתה לפני כן.
- כלומר: בוודאי שבכל עדכון אנו מגיעים לקונפיגורציה שעדיין לא היינו בה לפני כן.
- ולכן מס' האיטרציות האפשריות הינו סופי.

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),

$$\text{SSE} = \sum_{j=1}^{k} \sum_{\widehat{y_i}=j} d(x_i, \mu_j)^2$$

distance between a vector to its centroid

Cluster j

Centroid of x_i

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

שאלת סקר

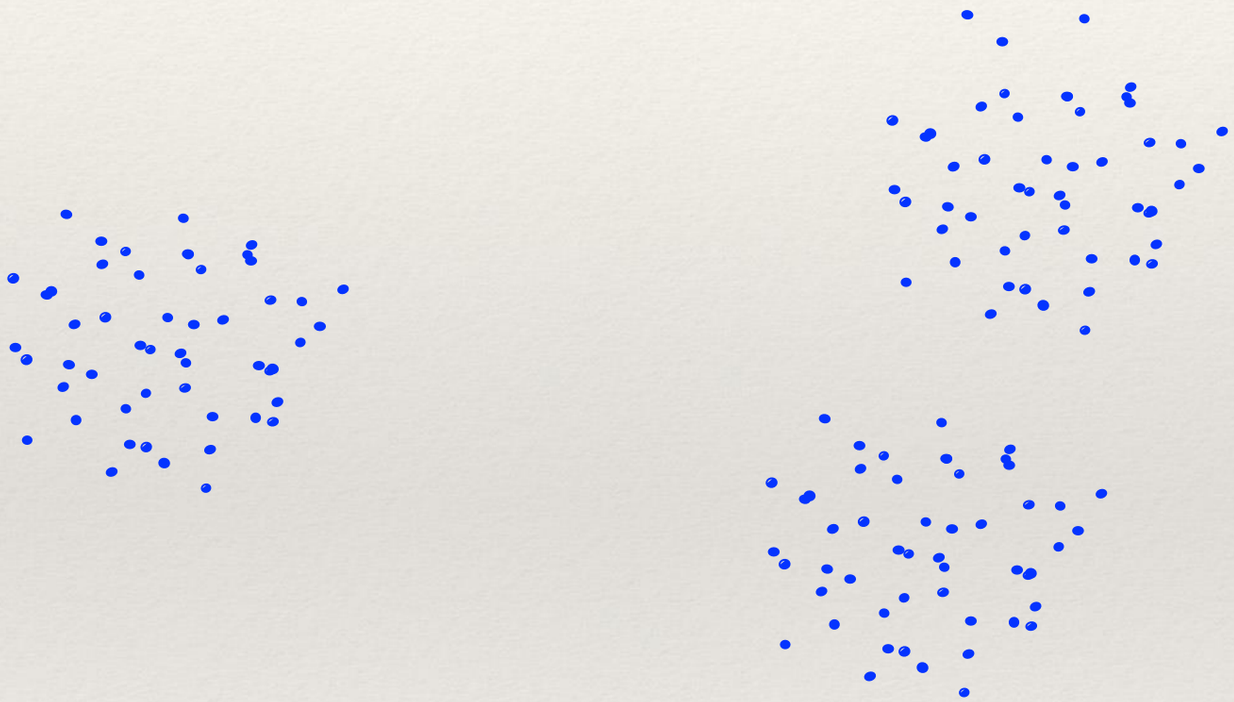
1. באיזו שיטה מהשיטות נשתמש ע"מ לוודא עצירת k-means?

<u>תשובות אפשרויות</u>:

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

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

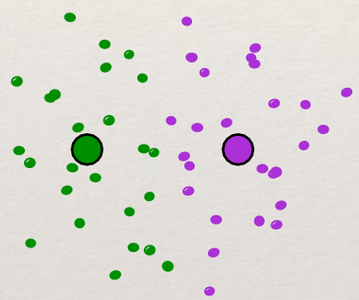
האם מובטח לנו למצוא את הקונפיגורציה האופטימאלית?



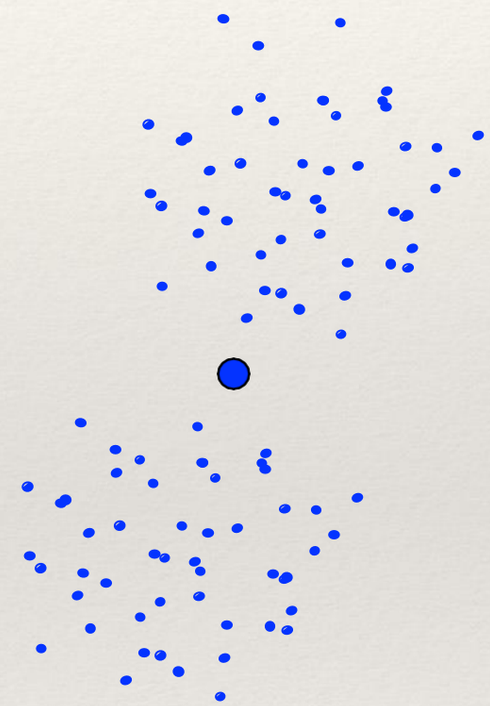
http://www.autonlab.org/tutorials/kmeans.html

האם מובטח לנו למצוא את הקונפיגורציה האופטימאלית?

תלוי בK ומציאת מינימום לוקאלי



Would be better to have one cluster here



… and two clusters here

http://www.autonlab.org/tutorials/kmeans.html

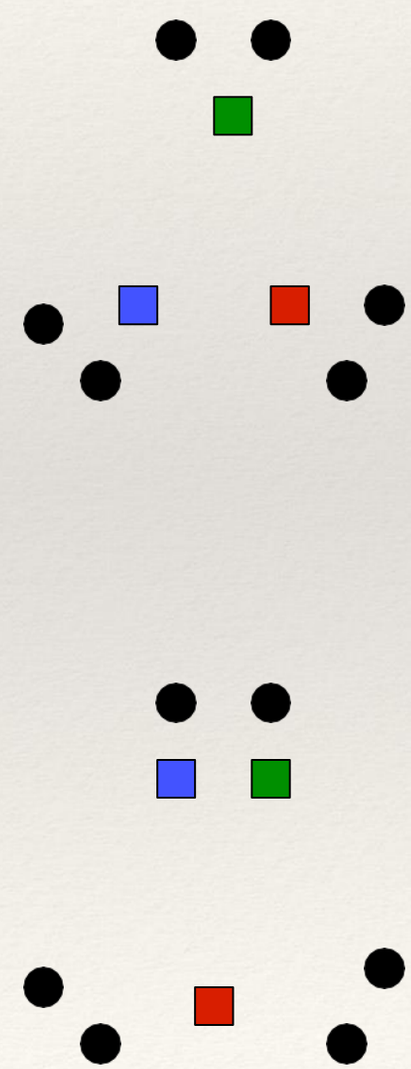
האם מובטח לנו למצוא את הקונפיגורציה האופטימאלית?

אלגוריתם k means הוא יוריסטי

- מצריך דוגמאות התחלתיות
- משנה מה בוחרים..
- מה יכול להשתבש?

מספר דרכים כדי למנוע בעיות מהסוג הזה (נלמד בהמשך) : איתחולים שונים, טכניקות שונות לחיתוך ואיחוד קבוצות

תלוי באיתחול



בשלב זה אתם צרכים לדעת (ביחס ל- k means)

מהו הדבר שאותו אנחנו מאפטמים?

האם אנחנו בטוחים שהאלגוריתם יסתיים?

האם אנחנו בטוחים שנמצא קלסטור אופטימלי?

מדוע כדאי לעשות סילום?

איך צריך לאתחל את המערכת?

DEMO

- http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html

סיכום K-means

- יתרונות:
  - קל למימוש
  - יעיל חישובית
- חסרונות:
  - עלול לעצור במינימום מקומי
  - מוגדר היטב כאשר ניתן לחשב ממוצע של מאפיין. מה נעשה עבור ערכים קטגוריים???
  - צריך להגדיר את K
  - רגיש ל-outliers