machine learning

K-Means

Exercise V

קרדיט - ד"ר יונתן רובין

פיתוח:
ד"ר יהונתן שלר
משה פרידמן

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?

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

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

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:

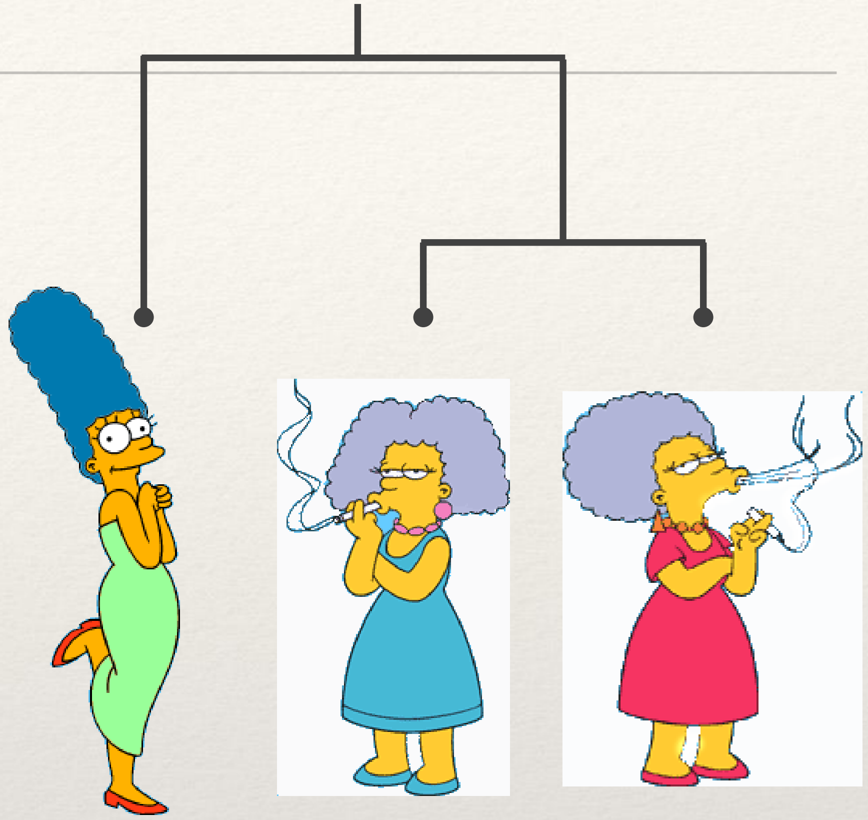
```
Change dress color,    1 point
Change earring shape,  1 point
Change hair part,      1 point
```

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"

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)

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 $D(O_1,O_2)$

Peter Piotr

0.23

3

342.7

Manhattan Distance example - Reminder

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} = \sum_{i=1}^{n} |x_i - y_i|$$

p = 1, Manhattan Distance

- Observation_1: [1, 7, 9]
- Observation_2: [11, 21, 4]
- |1–11| + |7–21| + |9–4| = 10 + 14 + 5 = 29
- The distance between the two observations is 29 now!

Euclidean Distance example - Reminder

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

p = 2, Euclidean Distance

- Observation_1: [1, 7, 9]
- Observation_2: [11, 21, 4]
- $(1–11)^2 + (7–21)^2 + (9–4)^2 = 100 + 196 + 25 = 321$
- The square root of 321 ~ 17.91
- The distance between these two observations is 17.91!

Chebyshev Distance example - Reminder

Chebyshev Distance = $\max_i(|x_i - y_i|)$

- Observation_1: [1, 7, 9]
- Observation_2: [11, 21, 4]
- max(|1–11|,|7–21|,|9–4|)= max(10, 14, 5) =14
- The distance between the two observations is 14 now!

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:

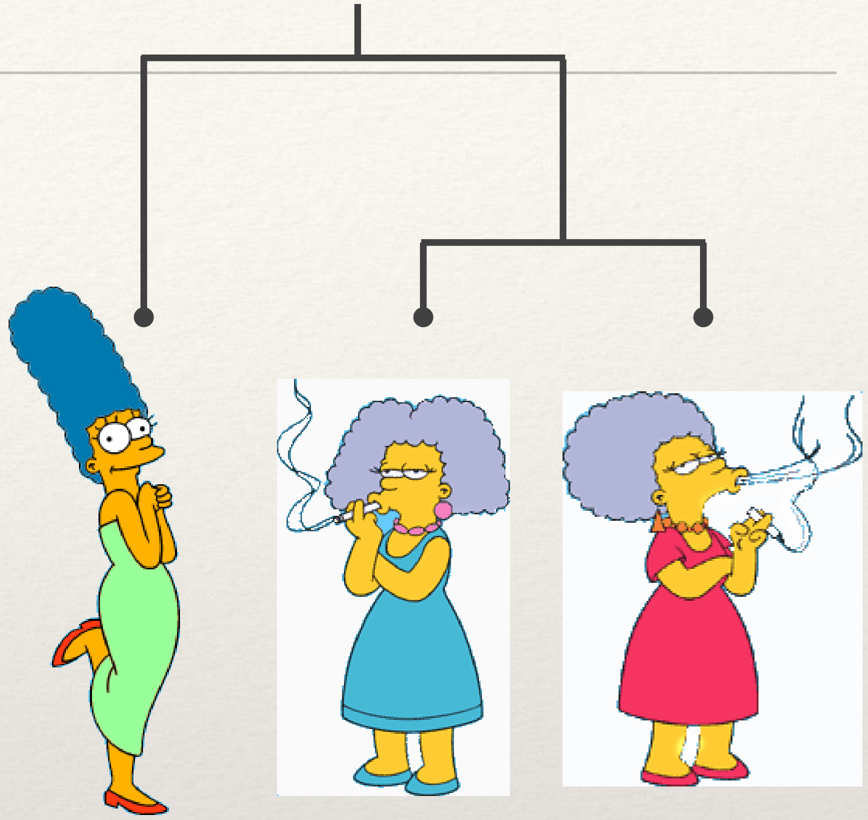
```
Change dress color,    1 point
Change earring shape,  1 point
Change hair part,      1 point
```

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”

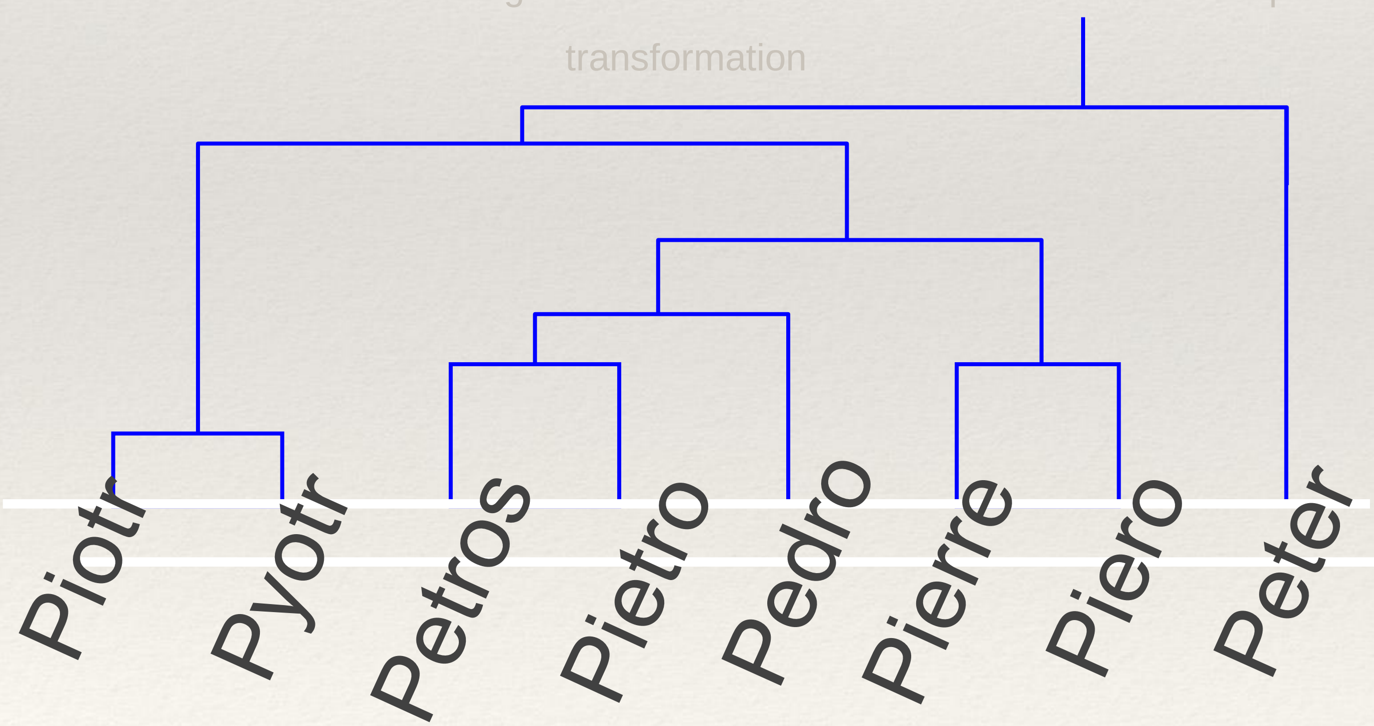
Slide based on one by Eamonn Keogh

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



Slide based on one by Eamonn Keogh

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

`Peter`

Substitution (i for e)

`Piter`

Insertion (o)

`Pioter`

Deletion (e)

`Piotr`

Cosine similarity measure

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

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

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

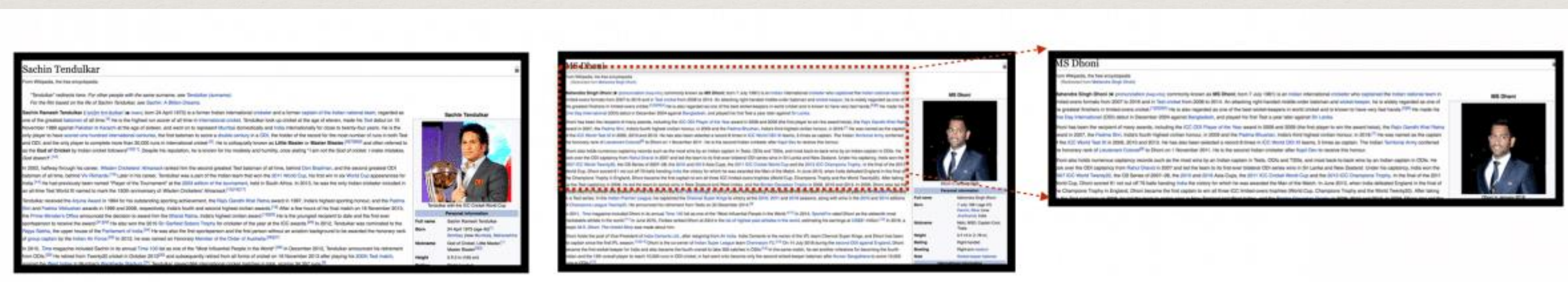
Cosine Similarity with L_2

Cosine Similarity Example

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

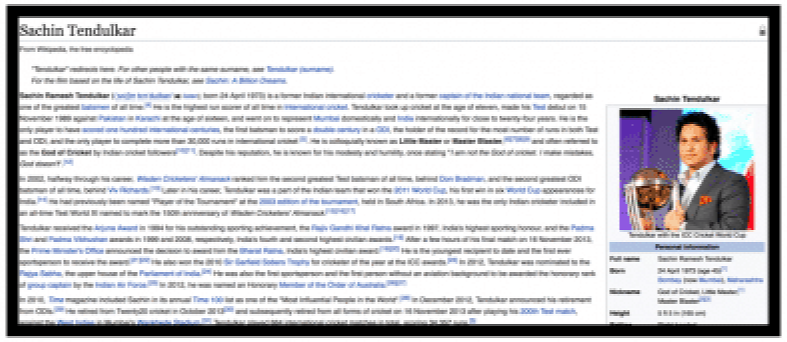
Let's suppose you have 3 documents based on a couple of star cricket players – **Sachin Tendulkar and Dhoni.**

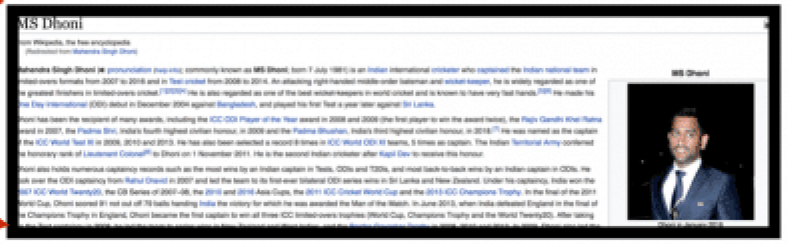
Two of the documents (A) and (B) are from the wikipedia pages on the respective players and the third document (C) is a smaller snippet from Dhoni's wikipedia page.



Cosine Similarity Example

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





Considering only the 3 words from the above documents: ‘sachin’, ‘dhoni’, ‘cricket’

Doc Sachin: Wiki page on Sachin Tendulkar	Doc Dhoni: Wiki page on Dhoni	Doc Dhoni_Small: Subsection of wiki on Dhoni
Dhoni - 10	Dhoni - 400	Dhoni - 10
Cricket - 50	Cricket - 100	Cricket - 5
Sachin - 200	Sachin - 20	Sachin - 1

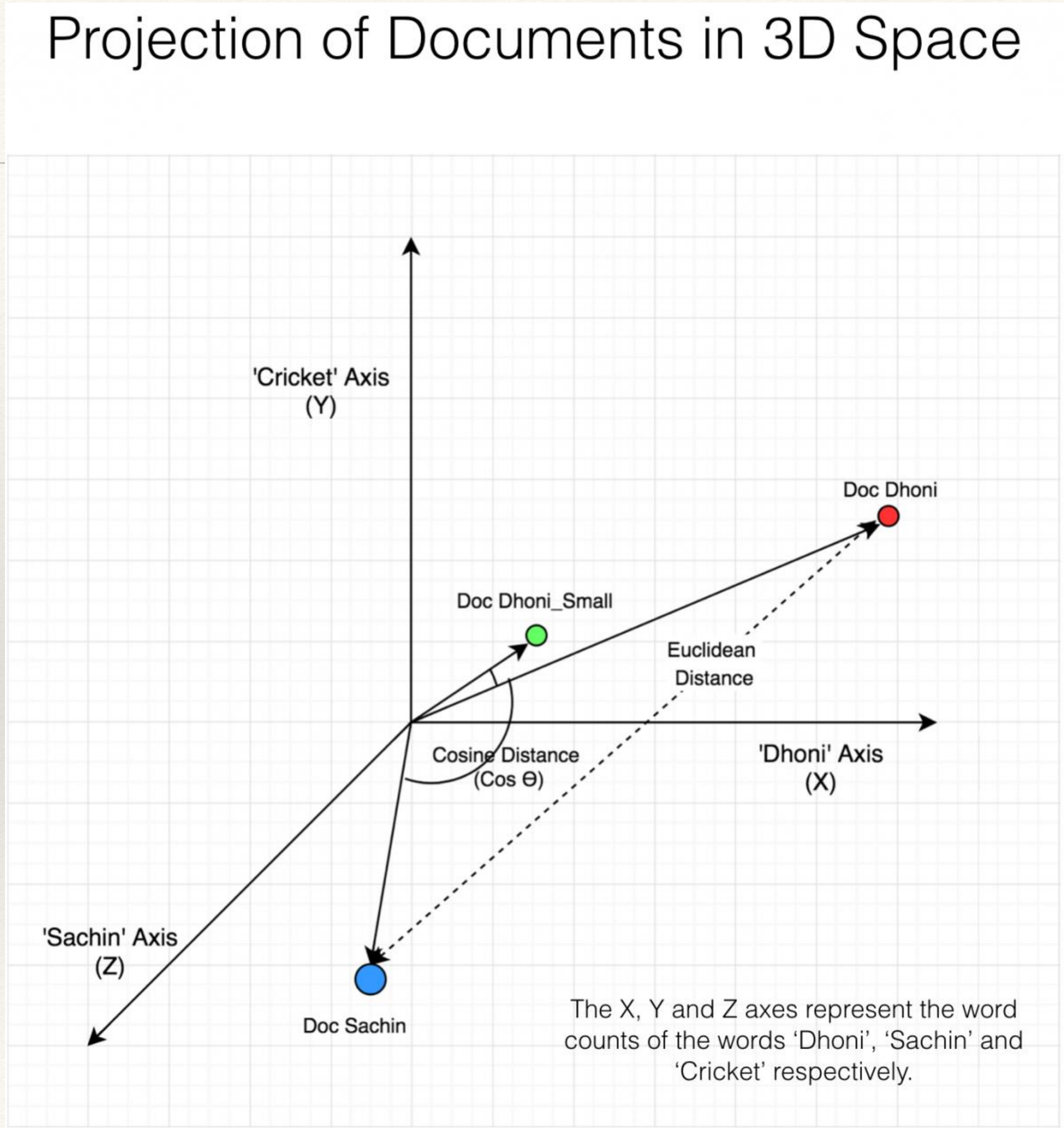
Document - Term Matrix (Word Counts)

Word Counts	*“Dhoni”*	*“Cricket”*	*“Sachin”*
Doc Sachin	10	50	200
Doc Dhoni	400	100	20
Doc Dhoni_Small	10	5	1

Similarity Metrics

Similarity or Distance Metrics	*Total Common Words*	*Euclidean distance*	*Cosine Similarity*
Doc Sachin & *Doc Dhoni*	10 + 50 + 10 = 70	432.4	0.15
Doc Dhoni & *Doc Dhoni_Small*	20 + 10 + 7 = 37	204.0	0.23
Doc Sachin & *Doc Dhoni_Small*	10 + 10 + 7 = 27	401.85	0.77

Cosine Similarity Example



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).
- אלגוריתם ++kmeans (Arthur & Vassilvitskii, 2007)

עבור כל השיטות הנ"ל

- ניתן לחזור על k-means כמה פעמים ולבחור את התוצר הטוב ביותר (שינתן תוצאות שונות, מכיוון שבכל פעם מגרילים מחדש את המרכזים)

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),

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

distance between a vector to its centroid

Cluster j

Centroid of x_i

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

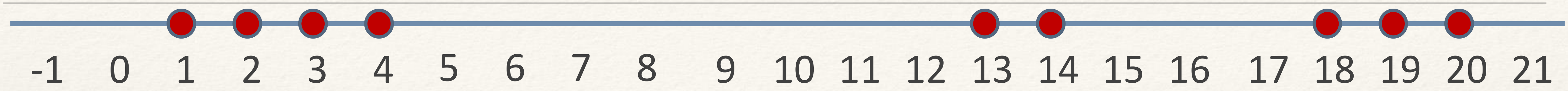
$$\boxed{r_{i,j} = \begin{cases} 1 & \hat{y}_i = j \\ 0 & \hat{y}_i \neq j \end{cases}}$$

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

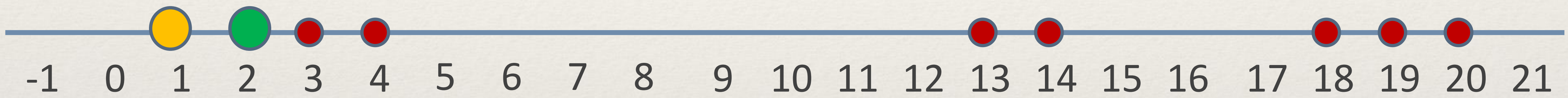
K-means – תרגיל 6
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

- 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 6 - <u>פתרון</u>
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית



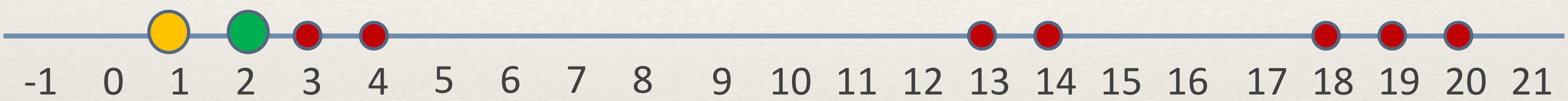
K-means שלב 1 (Forgy method) - ננחש 2 "מרכזים" ראשונים 1,2



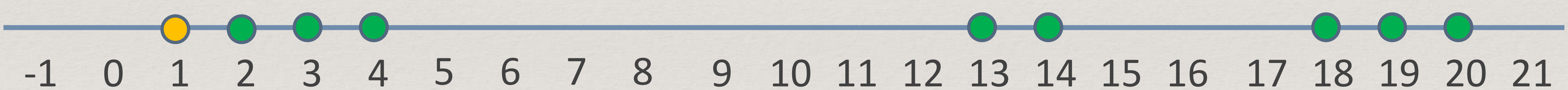
בחירה מאוד לא מושכלת

K-means – תרגיל 6 - <u>פתרון</u>
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

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



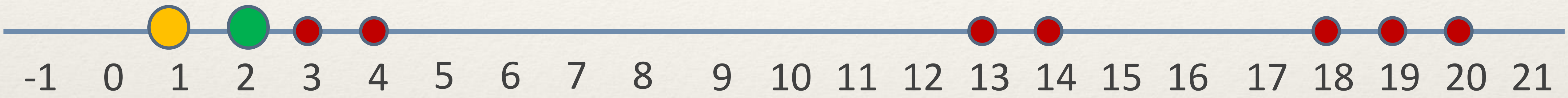
<u>איטרציה 1</u>: **K-means שלב 2** - נשייך כעת את הנקודות ל"מרכזים"(עפ"י מרחק אוקלידי)



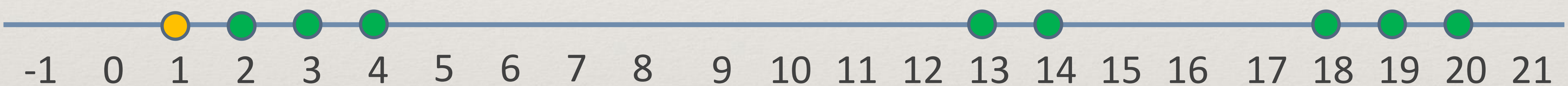
- 4 קרוב יותר (אוקלידית) ל-2 מאשר ל-1
- ...
- 14 יותר קרוב (אוקלידית) ל-2 מאשר ל-1

K-means – תרגיל 6 - <u>פתרון</u>
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

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

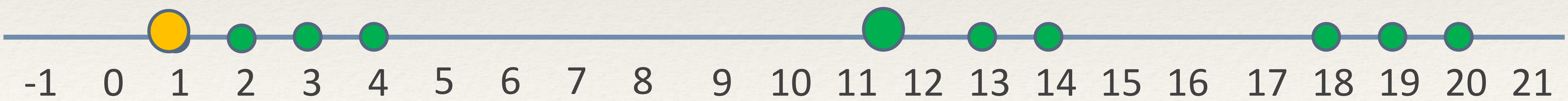


<u>איטרציה 1</u>: **K-means שלב 2** - נשייך כעת את הנקודות ל"מרכזים"(עפ"י מרחק אוקלידי)



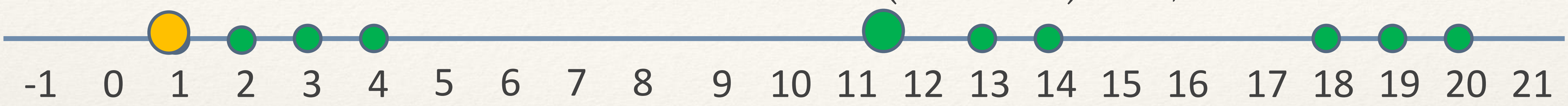
<u>איטרציה 1</u>: **K-means שלב 3** - נעדכן "מרכזים":

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

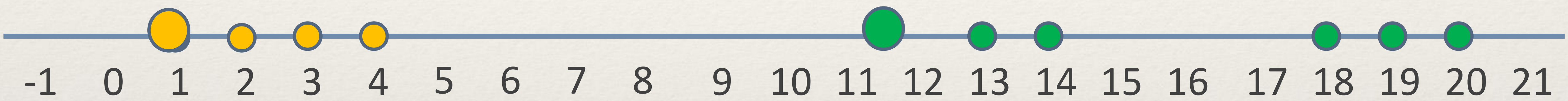


K-means – תרגיל 6 - פתרון
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

התוצאה מהאיטרציה הקודמת (איטרציה 1):

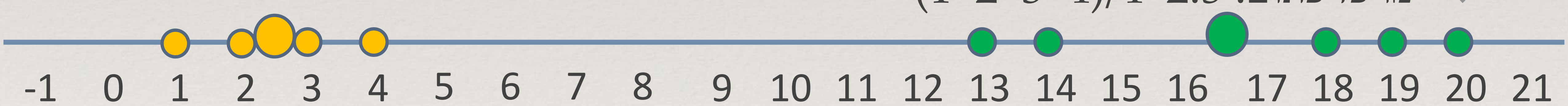


איטרציה 2: **K-means שלב 2** - נשייך כעת את הנקודות ל"מרכזים"



איטרציה 2: **K-means שלב 3** - נעדכן "מרכזים":

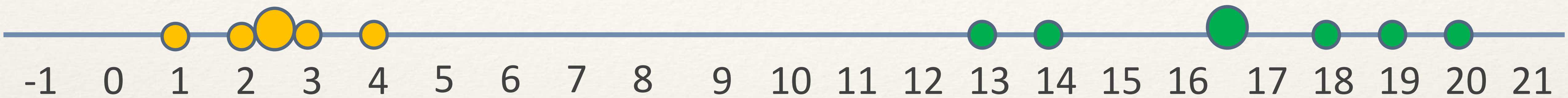
- מרכז ירוק: (13+14+18+19+20)/5=16.8
- מרכז כתום: (1+2+3+4)/4=2.5



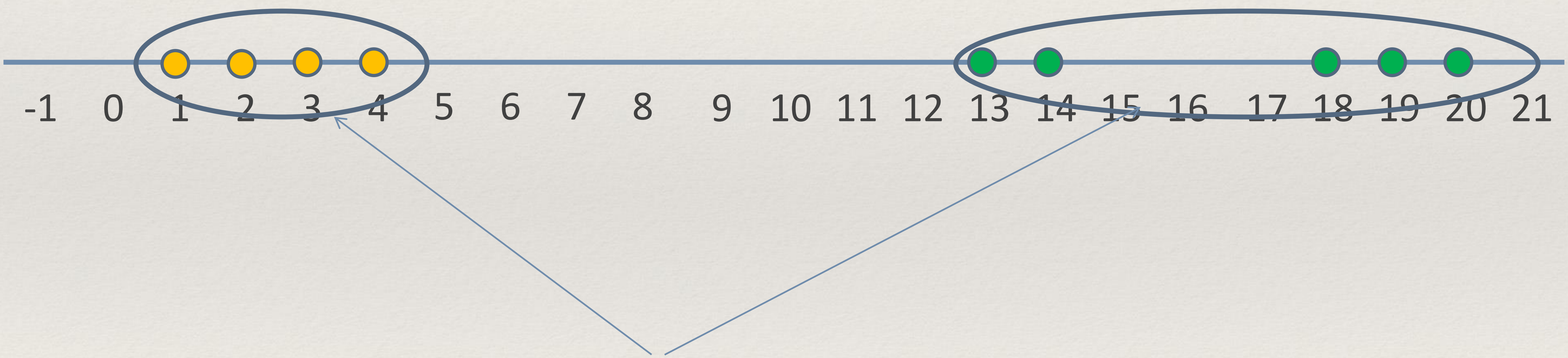
איטרציה 3: **K-means שלב 2** - נשייך כעת את הנקודות ל"מרכזים"

K-means – תרגיל 6 - <u>פתרון</u>
דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

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



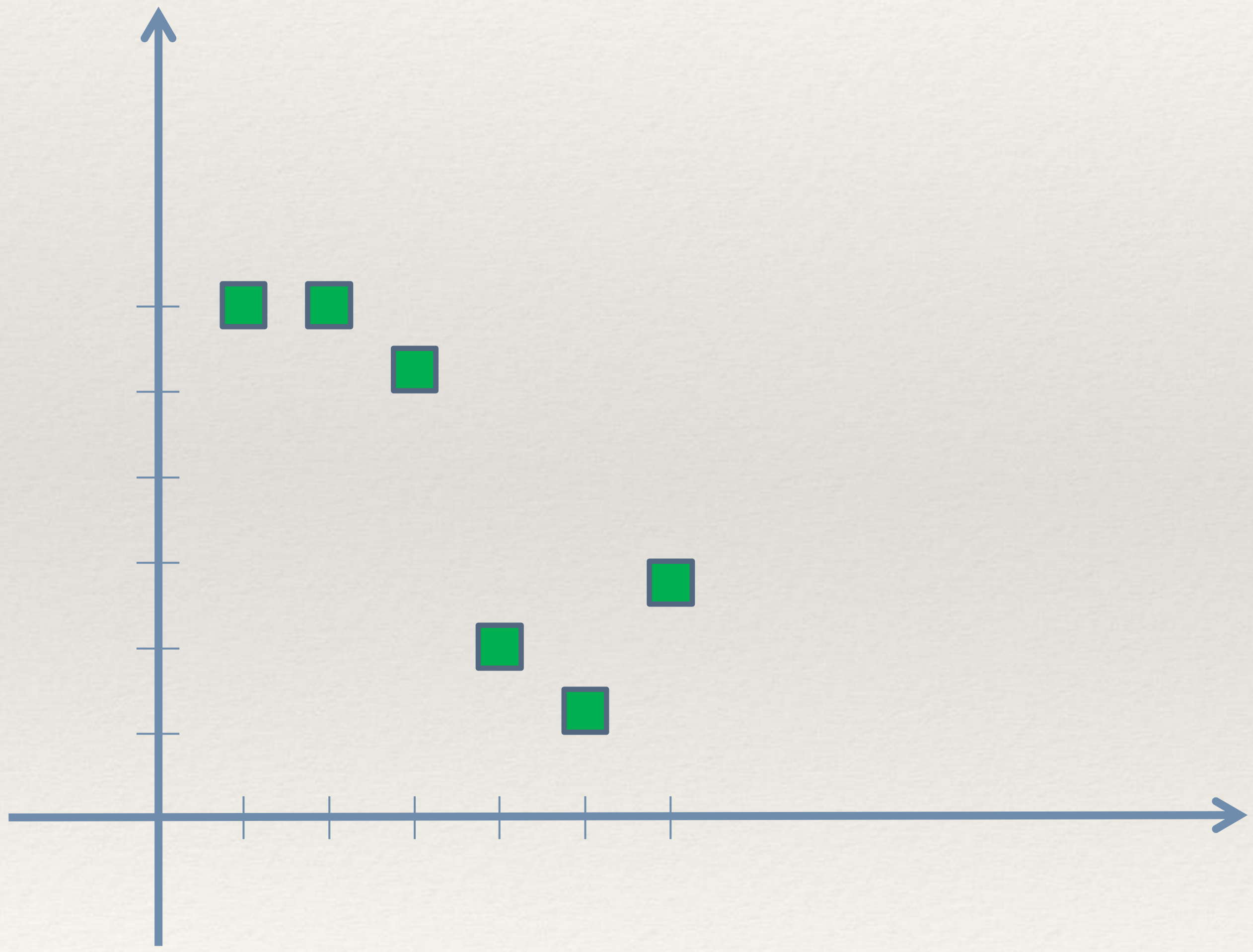
<u>איטרציה 3</u>: **K-means שלב 4** – אין עדכונים ולכן האלגוריתם עוצר



אלו 2 ה"אשכולות" שנוצרו

K-means – תרגיל 8
דוגמא עם 2 מאפיינים (2D), K=2

- נתונים הווקטורים הבאים:



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

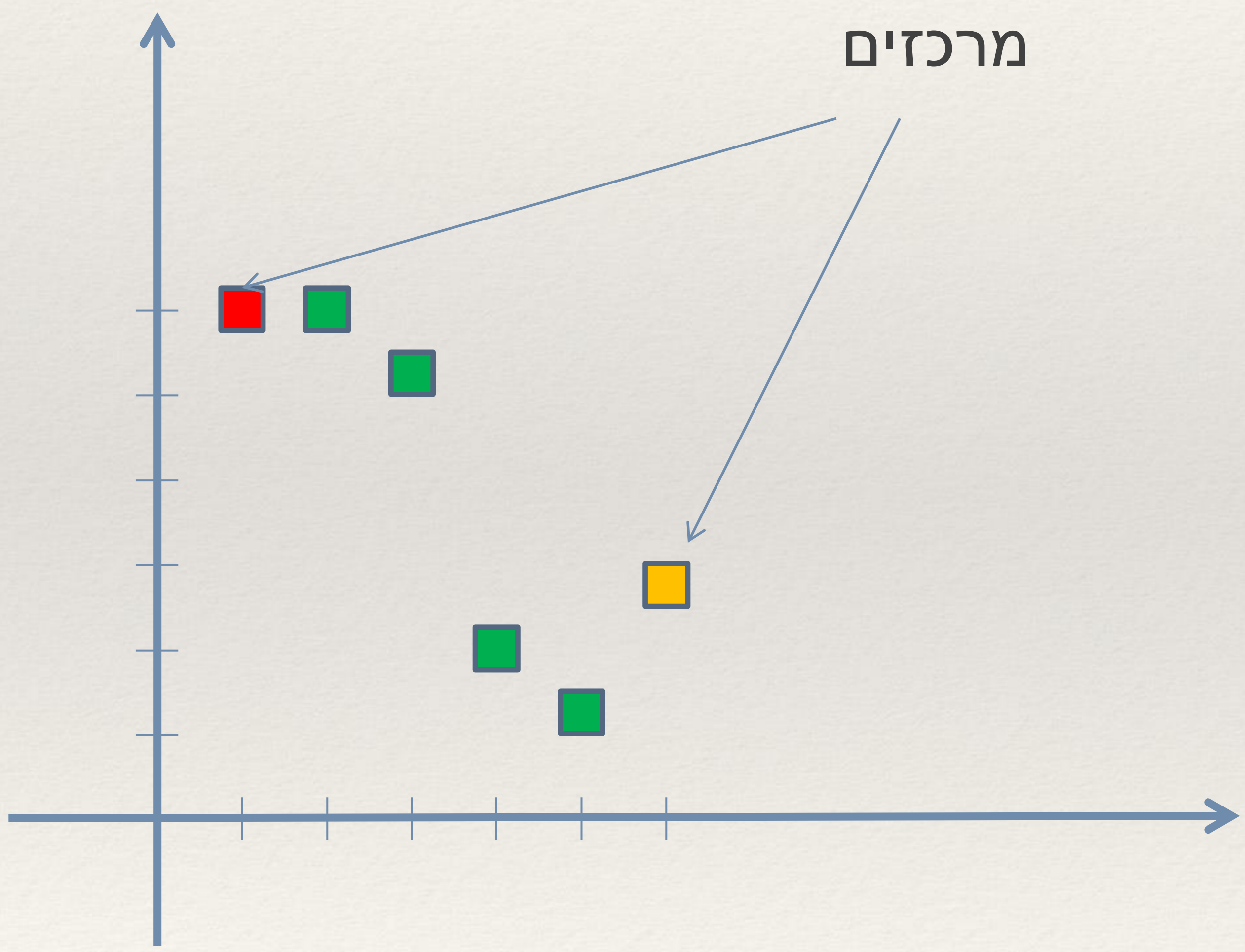
Recalculating centroids

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

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

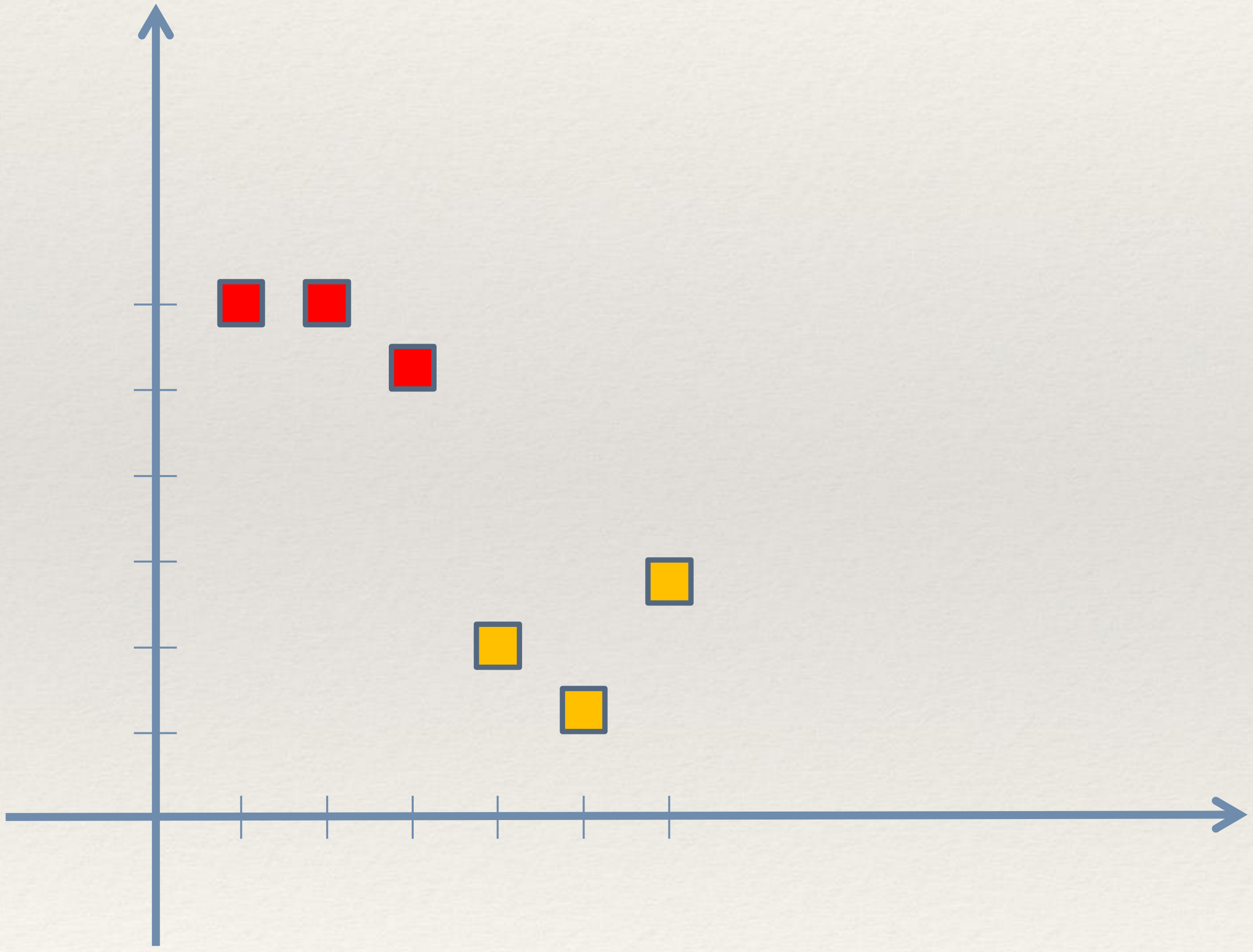
K-means – תרגיל 8 - <u>פתרון</u>
דוגמא עם 2 מאפיינים (2D), K=2

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



K-means – תרגיל 8 - פתרון
דוגמא עם 2 מאפיינים (2D), K=2

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



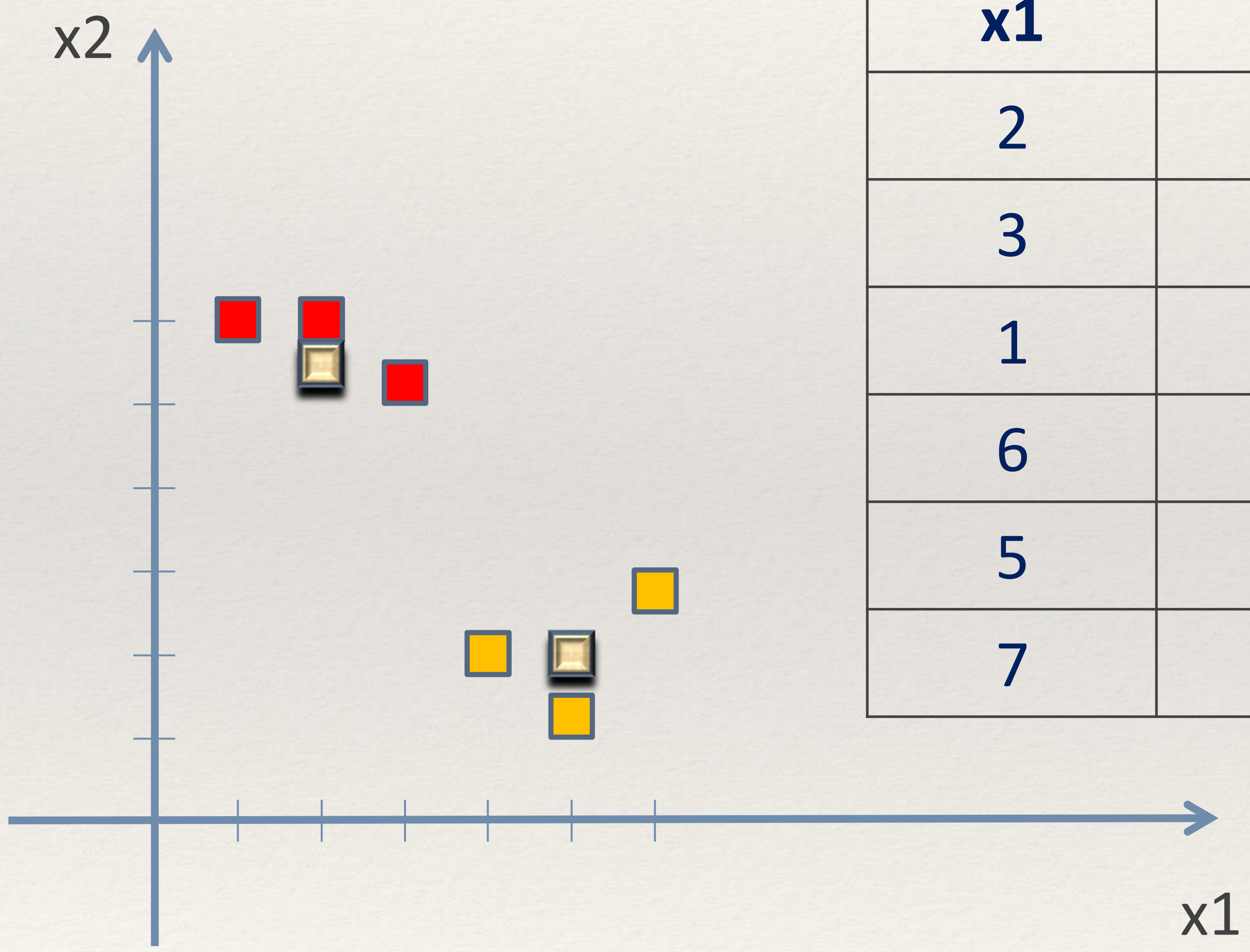
x1	x2
2	7
3	6
1	7
6	1
5	2
7	3

K-means – תרגיל 8 - פתרון
דוגמא עם 2 מאפיינים (2D), K=2

איטרציה 1: **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

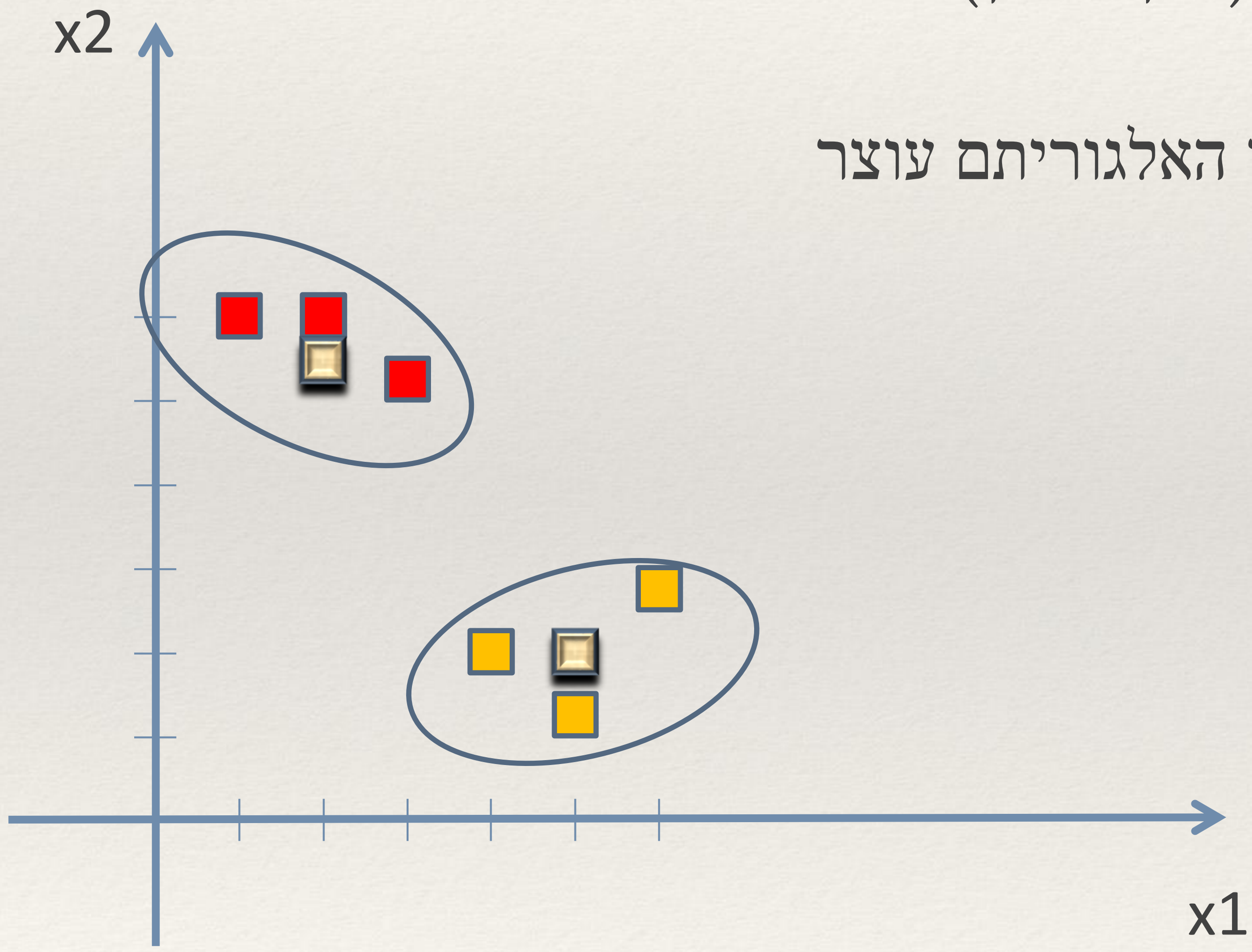


K-means – תרגיל 8 - <u>פתרון</u>
דוגמא עם 2 מאפיינים (2D), K=2

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

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

<u>איטרציה 2</u>: **K-means שלב 4** – אין עדכונים ולכן האלגוריתם עוצר



Cluster evaluation (a hard problem)

1. Good clusters - Good Clusters produce high quality:

<u>Intra-cluster cohesion (compactness)</u>:

$$WSS = \sum_{i} \sum_{x \in C_i} (x - \mu_i)^2$$

- Cohesion measures how near the data points in a cluster are to the cluster centroid.
- Sum of squared error (SSE) is a commonly used measure.

<u>Inter-cluster separation (isolation)</u>:

$$BSS = \sum_i |C_i| (\mu - \mu_i)^2$$

- We measure this by the between cluster sum of squares

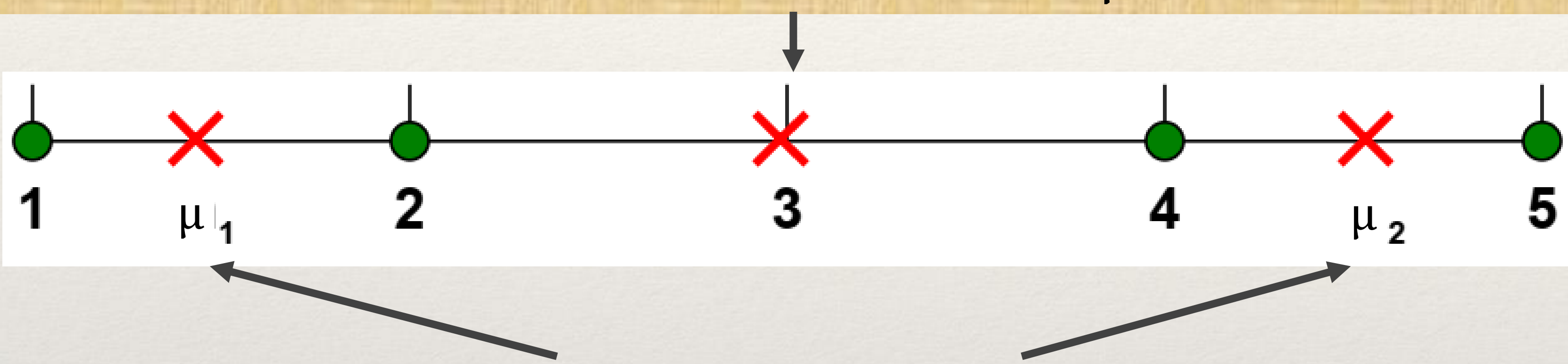
μ=center of all the dataset
μ_i=centroid of cluster i

Cluster evaluation - Intra-cluster, Inter-cluster

1. Good clusters - Good Clusters produce high quality:

Centroid, k=1 , point between clusters

μ



WSS+BSS= constant

K=1 cluster:

$$WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$BSS = 4 \times (3-3)^2 = 0$$

$$Total = 10 + 0 = 10$$

K=2 clusters:

$$WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$

$$Total = 1 + 9 = 10$$

$$\boldsymbol{WSS} = \sum_{i} \sum_{x \in C_i} (\boldsymbol{x} - \boldsymbol{m_i})^2$$

$$BSS = \sum |C_i| (m - m_i)^2$$

Cluster evaluation – Silhouette score – Evaluation both Intra-cluster & Inter-cluster

a(i) = עבור וקטור i, השייך לcluster מסויים C_i, יתן את ממוצע המרחקים מוקטור i לכל שאר הוקטורים ב-cluster זה

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

b(i) = א. עבור וקטור i, השייך לcluster מסויים C_i, עבור cluster מסויים C_k, כך ש $C_k \neq C_i$, ניקח את כל הוקטורים השייכים ל-C_k ונחשב את ממוצע המרחקים מוקטור i לכל שאר הוקטורים ב-cluster זה (כלומר, כביכול נוסיף את וקטור i ל-C_k ואז נחשב את a(i)).

ב. נעשה את החישוב הנ"ל עבור כל C_k, כך ש $C_k \neq C_i$, וניקח את ה-minimum (כלומר את ה- a(i) המינימלי אם הוא היה ב-cluster אחר C_k, כך ש $C_k \neq C_i$)

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

s(i) = עבור וקטור i, השייך ל-C_i, יתן את ההפרש בין b(i) ל-a(i) , חלקי המקסימום ביניהם.

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

- $-1<s(i)<1$
- כאשר השיוך של וקטור i מתאים יותר ל C_i, a(i)<b(i) ו-s(i) יהיה חיובי.
- כאשר s(i) קרוב ל-1 ניתן לומר כי הוקטור i מתאים C_i.
  - ערך שכזה מתקבל כאשר ערך הלכידות קטנה בצורה משמעותית מערך ההפרדה.
- כאשר s(i) קרוב ל-0 ניתן לומר כי הנתון נמצא קרוב מאוד לגבול בין שני אשכולות שכנים.
- כאשר s(i) קרוב ל- (-1) ניתן לומר כי הנתון נמצא באשכול שלא מתאים לו.

silhouette score יתן את הערך הממוצע של s(i), עבור כל הוקטורים ב-dataset.

- ככל שה-*silhouette score* קרוב ל-1, אומר שיש השמה נכונה, עבור רוב הוקטורים ב-dataset, ככל שה-*silhouette score* קרוב ל- -1 אומר שיש השמה לא נכונה, עבור רוב הוקטורים ב-dataset

silhouette score – $\tilde{s}(k)$– represents the mean s(i)

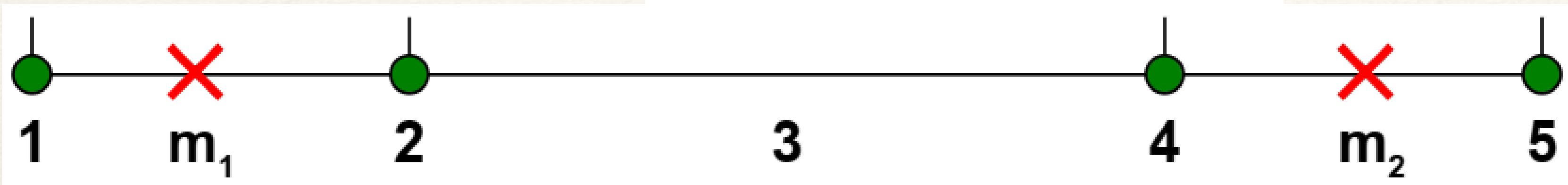
Cluster evaluation – Silhouette score –
Evaluation both Intra-cluster & Inter-cluster– example

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j)$$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

silhouette score – $\tilde{s}(k)$– represents the mean s(i)



<u>תרגיל</u>: נתון תוצר ה-clustering
הבא, בו k=2
<u>חשבו את ה- silhouette score</u>

<u>פתרון</u>:

Given $k = 2$; C_1 info: $|C_1|$=2, points: 1,2, centroid: m_1=1.5; C_2 info: $|C_2|$=2 points:4,5, centroid: m_2=4.5;

<u>1. Compute a(i)</u> – intra-cluster proximity (similarity)

a(1)= $\sqrt{(1-2)^2}$=**1**=$\sqrt{(2-1)^2}$=**a(2)**= $\sqrt{(4-5)^2}$=**a(4)**=$\sqrt{(5-4)^2}$=**a(5)**

<u>2. Compute b(i)</u> – proximity to other clusters:

b(1) $=\frac{1}{2}(\sqrt{(1-4)^2} + \sqrt{(1-5)^2}) = 3.5 = \frac{1}{2}(\sqrt{(5-2)^2} + \sqrt{(5-1)^2})$=b(5)

b(2)$=\frac{1}{2}(\sqrt{(2-4)^2} + \sqrt{(2-5)^2}) = 2.5 = \frac{1}{2}(\sqrt{(4-2)^2} + \sqrt{(4-1)^2})$=b(4)

<u>3. Compute s(i)</u> – the silhouette value

$s(1) = \frac{3.5-1}{3.5} \approx 0.71 = s(5)$

$s(2) = \frac{2.5-1}{2.5} \approx 0.6 = s(4)$

<u>4. Compute $\tilde{s}$</u> – the mean silhouette value

$\tilde{s}(k = 2) \approx 1/4(2 \cdot 0.71 + 2 \cdot 0.6) \approx 0.66$

Cluster evaluation *labeled data* based
The Purity metric – example 2

$Purity\ (\omega_i) = \frac{1}{n_i} \max_j(n_{ij})\ j \in C$

ω_i - cluster i;

n_i - members in cluster ω_i;

π_i - dominant class in the cluster ω_i;

C - gold standard classes

$\mathbf{Purity}_{total} = \sum_{i=1}^{k} \frac{n_i}{n} \cdot Purity\ (\omega_i)$

We have a dataset of 900 documents,
Divided equally to 3 topics, we performed clustering,
and got the following results:

Cluster	Science	Sports	Politics
1	250	20	10
2	20	180	80
3	30	100	210
Total	300	300	300

Best cluster

Calculate per cluster purity

Cluster 1 – majority class: Science $n_{1\text{Science}}=250$ $Purity\ (\omega_1) = \frac{1}{280} \cdot 250 \approx 0.89$

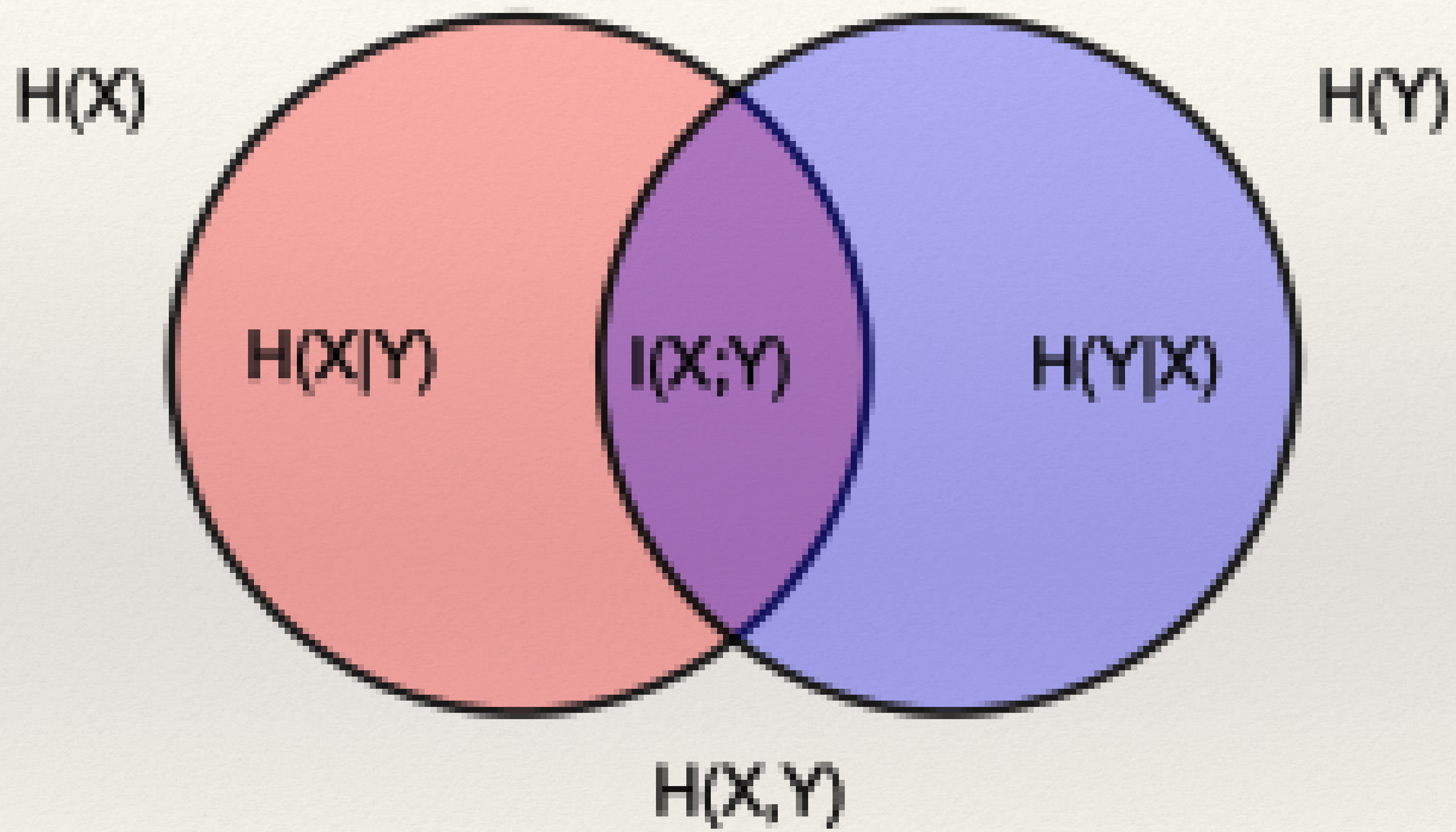
Cluster 2 – majority class: Sports $n_{2Sports}=180$ $Purity\ (\omega_2) = \frac{1}{280} \cdot 180 \approx 0.64$

Cluster 3 – majority class: Politics $n_{3\text{Politics}}=210$ $Purity\ (\omega_3) = \frac{1}{340} \cdot 210 \approx 0.62$

$\mathbf{Purity}_{total} = \sum_{i=1}^{3} \frac{n_i}{n} \cdot Purity\ (\omega_i) = \frac{280}{900} \cdot \frac{1}{280} \cdot 250 + \frac{280}{900} \cdot \frac{1}{280} \cdot 180 + \frac{340}{900} \cdot \frac{1}{340} \cdot 210 = \frac{640}{900} \approx 0.711$

Calculate total purity

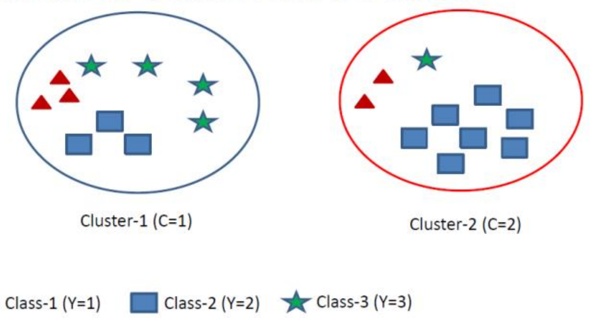
Cluster evaluation - *labeled data* based



Cluster evaluation – evaluating label data with NMI - example

Calculating NMI for Clustering

- Assume m=3 classes and k=2 clusters



$$\text{NMI} = \frac{I(C;\ \Omega)}{|H(\Omega) + H(C)|/2}$$

C - gold standard classes
Ω - clusters found
ω_i - cluster i;
c_j - class j;
n_i - members in cluster ω_i;
n_j - members in class c_j;
n_{ij} - members in cluster ω_i and in class c_j
n – number of instance in the dataset
π_i - dominant class in the cluster ω_i;

$$\boxed{H(C) = \sum_{j \in C} -pr(c = c_j) \cdot \log pr(c = c_j)}$$

H(Y) = Entropy of Class Labels

- P(Y=1) = 5/20 = ¼
- P(Y=2) = 5/20 = ¼
- P(Y=3) = 10/20 = ½
- H(Y) = $-\frac{1}{4}\log\left(\frac{1}{4}\right) - \frac{1}{4}\log\left(\frac{1}{4}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = 1.5$

This is calculated for the entire dataset and can be calculated prior to clustering, as it will not change depending on the clustering output.

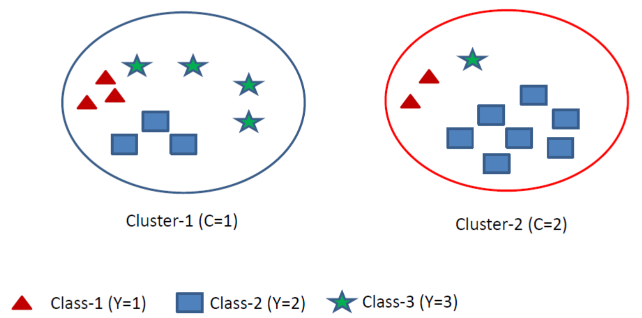
Calculate NMI:

1. Calculate class entropy - H(C)

Cluster evaluation – evaluating label data with NMI - example

Calculating NMI for Clustering

- Assume m=3 classes and k=2 clusters



$$\mathrm{NMI} = \frac{I(C;\ \Omega)}{|H(\Omega) + H(C)|/2}$$

$\mathrm{H}(\Omega) = \sum_{i \in \Omega} -pr(\Omega = \omega_i) \cdot \log pr(\Omega = \omega_i)$

C - gold standard classes
$\boldsymbol{\Omega}$ - clusters found
ω_i - cluster i;
c_j - class j;
n_i - members in cluster ω_i;
n_j - members in class c_j;
n_{ij} - members in cluster ω_i and in class c_j
n – number of instance in the dataset
π_i - dominant class in the cluster ω_i;

$\mathrm{H}(\Omega)$ = Entropy of Cluster Labels

$\mathrm{P}(\Omega = 1)$ = 10/20 = 1/2

$\mathrm{P}(\Omega = 2)$ = 10/20 = ½

- $\mathrm{H}(\Omega) = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = 1$

This will be calculated every time the clustering changes. You can see from the figure that the clusters are balanced (have equal number of instances).

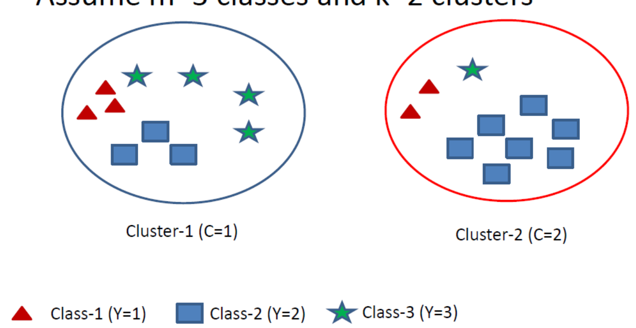
Calculate NMI:

1. Calculate class entropy - H(C)
2. Calculate cluster entropy - H(Ω)

Cluster evaluation – evaluating label data with NMI - example

Calculating NMI for Clustering

- Assume m=3 classes and k=2 clusters



$$\mathrm{NMI} = \frac{I(C;\ \Omega)}{|H(\Omega) + H(C)|/2}$$

C - gold standard classes
Ω - clusters found
ω_i - cluster i;
c_j - class j;
n_i - members in cluster ω_i;
n_j - members in class c_j;
n_{ij} - members in cluster ω_i and in class c_j
n – number of instance in the dataset
π_i - dominant class in the cluster ω_i;

$H(C|\Omega) = \sum_{i \in \Omega} pr(\Omega = \omega_i) \cdot H(C|\Omega = \omega_i)$

- Start with ω_1

$H(C|\Omega)$: conditional entropy of class labels for clustering C

- Consider Cluster-1:
  - $P(C = 1|\Omega = 1)$ =3/10 (three triangles in cluster-1)
  - $P(C = 2|\Omega = 1)$ =3/10 (three rectangles in cluster-1)
  - $P(C = 3|\Omega = 1)$ =4/10 (four stars in cluster-1)
  - Calculate conditional entropy as:

$pr(\Omega = \omega_1)$

$$H(C|\Omega = 1) = -P(C = 1) \sum_{y \in \{1,2,3\}} P(C = y|\Omega = 1) \cdot \log(P(C = y|\Omega = 1))$$

$$= -\frac{1}{2} \times \left[\frac{3}{10}\log\left(\frac{3}{10}\right) + \frac{3}{10}\log\left(\frac{3}{10}\right) + \frac{4}{10}\log\left(\frac{4}{10}\right)\right] = 0.7855$$

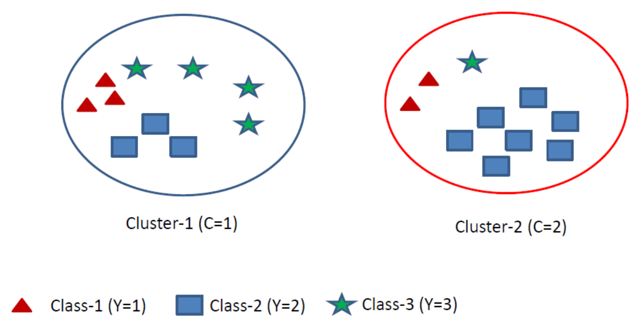
Calculate NMI:

1. Calculate class entropy - H(C)
2. Calculate cluster entropy - H(Ω)
3. Calculate conditional entropy - H(C|Ω)

Cluster evaluation – evaluating label data with NMI - example

Calculating NMI for Clustering

- Assume m=3 classes and k=2 clusters



$$\mathrm{NMI} = \frac{I(C;\ \Omega)}{|H(\Omega) + H(C)|/2}$$

C - gold standard classes
Ω - clusters found
ω_i - cluster i;
c_j - class j;
n_i - members in cluster ω_i;
n_j - members in class c_j;
n_{ij} - members in cluster ω_i and in class c_j
n – number of instance in the dataset
π_i - dominant class in the cluster ω_i;

$H(C|\Omega) = \sum_{i\in\Omega} pr(\Omega = \omega_i) \cdot H(C|\Omega = \omega_i)$

- Now consider ω_2

$H(C|\Omega)$ conditional entropy of class labels for clustering C

- Now, consider Cluster-2:
  - $P(C = 1|\Omega = 2)$ =2/10 (two triangles in cluster-1)
  - $P(C = 2|\Omega = 2)$ =7/10 (seven rectangles in cluster-1)
  - $P(C = 3|\Omega = 2)$ =1/10 (one star in cluster-1)
  - Calculate conditional entropy as:

$\boxed{pr(\Omega = \omega_2)}$

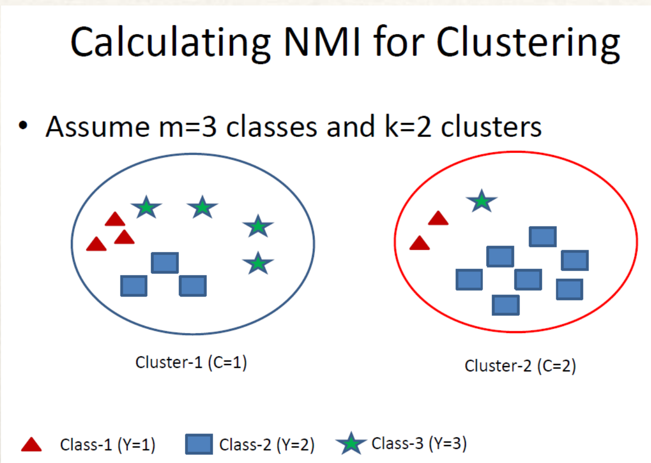
$$H(C|\Omega = 2) = -P(\Omega = 2) \sum_{y\in\{1,2,3\}} P(C = y|\Omega = 2)\cdot\log(P(C = y|\Omega = 2))$$

$$= -\frac{1}{2} \times \left[\frac{2}{10}\log\left(\frac{2}{10}\right) + \frac{7}{10}\log\left(\frac{7}{10}\right) + \frac{1}{10}\log\left(\frac{1}{10}\right)\right] = 0.5784$$

Calculate NMI:

1. Calculate class entropy - H(C)
2. Calculate cluster entropy - $H(\Omega)$
3. Calculate conditional entropy - $H(C|\Omega)$

Cluster evaluation – evaluating label data with NMI - example



$$\mathrm{NMI} = \frac{I(C;\ \Omega)}{|H(\Omega) + H(C)|/2}$$

$$\mathrm{H}(\mathrm{C}|\Omega) = \sum_{i\in\Omega} pr(\Omega = \omega_i) \cdot H(C|\Omega = \omega_i) =$$

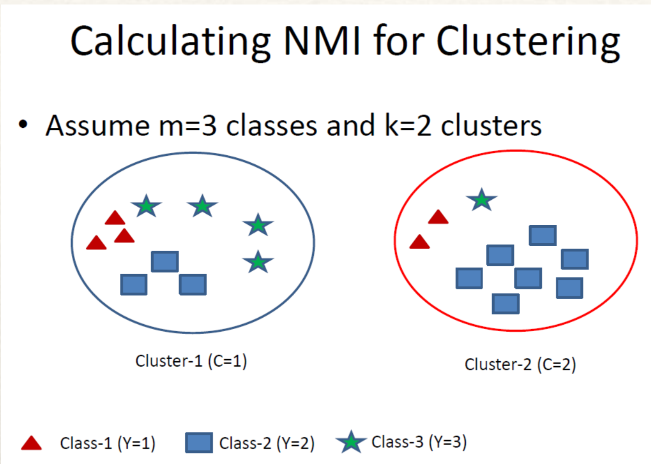
$$= pr(\Omega = \omega_1) \cdot H(C|\Omega = \omega_1) + pr(\Omega = \omega_2) \cdot H\left(C|\Omega = \omega_2\right) =$$

$=0.7855 + 0.5784 =$ <u>1.3639</u>

C - gold standard classes
$\boldsymbol{\Omega}$ - clusters found
ω_i - cluster i;
c_j - class j;
$\boldsymbol{n_i}$ - members in cluster ω_i;
$\boldsymbol{n_j}$ - members in class c_j;
$\boldsymbol{n_{ij}}$ - members in cluster ω_i and in class c_j
$\boldsymbol{n}$ – number of instance in the dataset
$\boldsymbol{\pi_i}$ - dominant class in the cluster ω_i;

Calculate NMI:
1. Calculate class entropy - H(C)
2. Calculate cluster entropy - H(Ω)
3. Calculate conditional entropy - H(C|Ω)

Cluster evaluation – evaluating label data with NMI - example



$$\text{NMI} = \frac{I(C;\ \Omega)}{|H(\Omega) + H(C)|/2}$$

C - gold standard classes
Ω - clusters found
ω_i - cluster i;
c_j - class j;
n_i - members in cluster ω_i;
n_j - members in class c_j;
n_{ij} - members in cluster ω_i and in class c_j
n – number of instance in the dataset
π_i - dominant class in the cluster ω_i;

I(C; Ω) = H(C) – H(C | Ω) =
≈ 1.5-1.364≈0.136

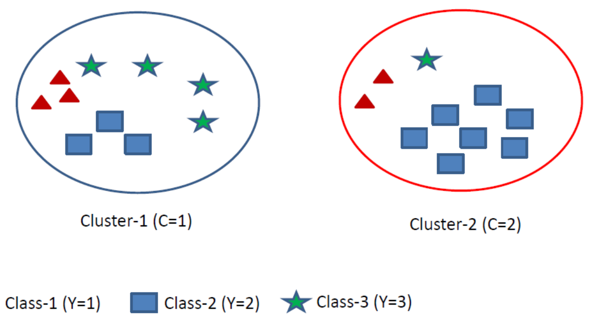
Mutual Information tells us the reduction in the entropy of class labels that we get if we know the cluster labels. (Similar to Information gain in deicison trees)

Calculate NMI:
1. Calculate class entropy - H(C)
2. Calculate cluster entropy - H(Ω)
3. Calculate conditional entropy - H(C|Ω)
4. Calculate mutual information – I(C; Ω)

Cluster evaluation – evaluating label data with NMI - example

Calculating NMI for Clustering

- Assume m=3 classes and k=2 clusters



$$\text{NMI} = \frac{I(C;\ \Omega)}{|H(\Omega) + H(C)|/2}$$

C - gold standard classes
Ω - clusters found
ω_i - cluster i;
c_j - class j;
n_i - members in cluster ω_i;
n_j - members in class c_j;
n_{ij} - members in cluster ω_i and in class c_j
n – number of instance in the dataset
π_i - dominant class in the cluster ω_i;

$\text{NMI} = (I(C;\Omega))/(|H(\Omega)+H(C)|/2) =$
$= 0.136/(|1+1.5|/2) \approx 0.109$

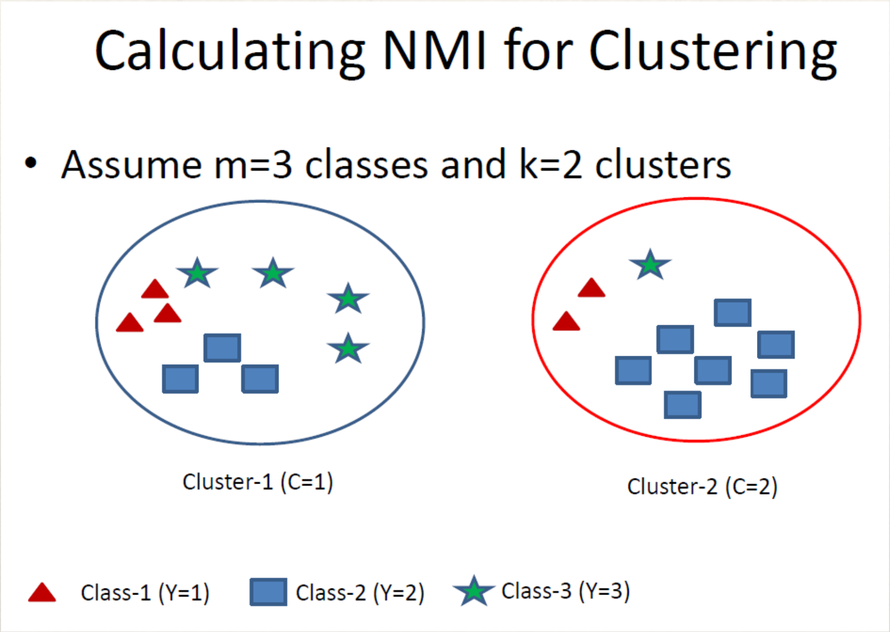
NMI

- NMI is a good measure for determining the quality of clustering.
- It is an external measure because we need the class labels of the instances to determine the NMI.
- Since it's normalized we can measure and compare the NMI between different clusterings having different number of clusters.

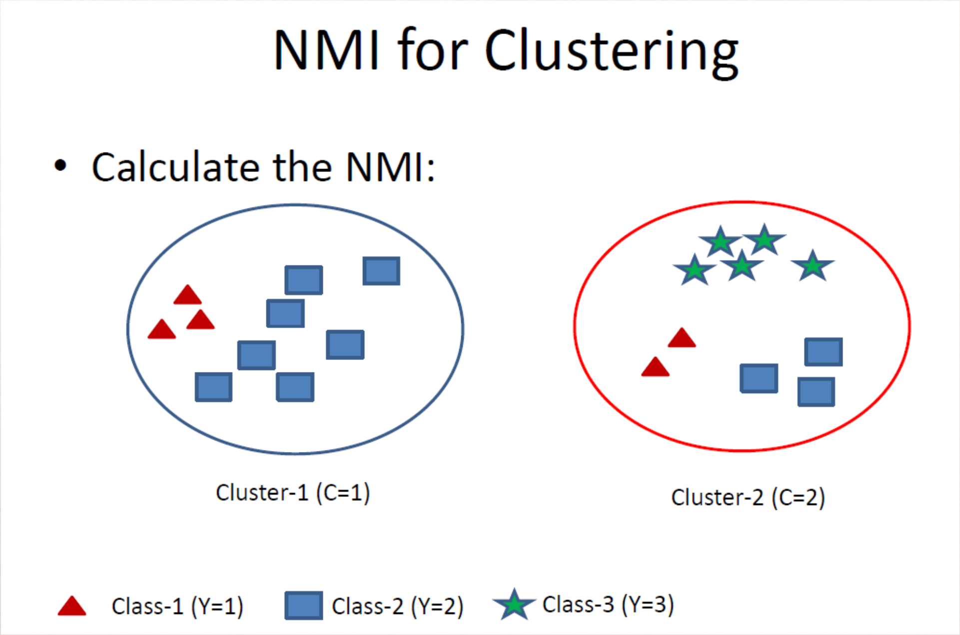
Calculate NMI:

1. Calculate class entropy - H(C)
2. Calculate cluster entropy - H(Ω)
3. Calculate conditional entropy - H(C|Ω)
4. Calculate mutual information – I(C; Ω)
5. Calculate NMI - $(I(C;\Omega))/(|H(\Omega)+H(C)|/2)$

Cluster evaluation – evaluating label data with NMI - example



VS.



Cluster evaluation – evaluating label data with NMI - example

$$I(C;\ \Omega)$$

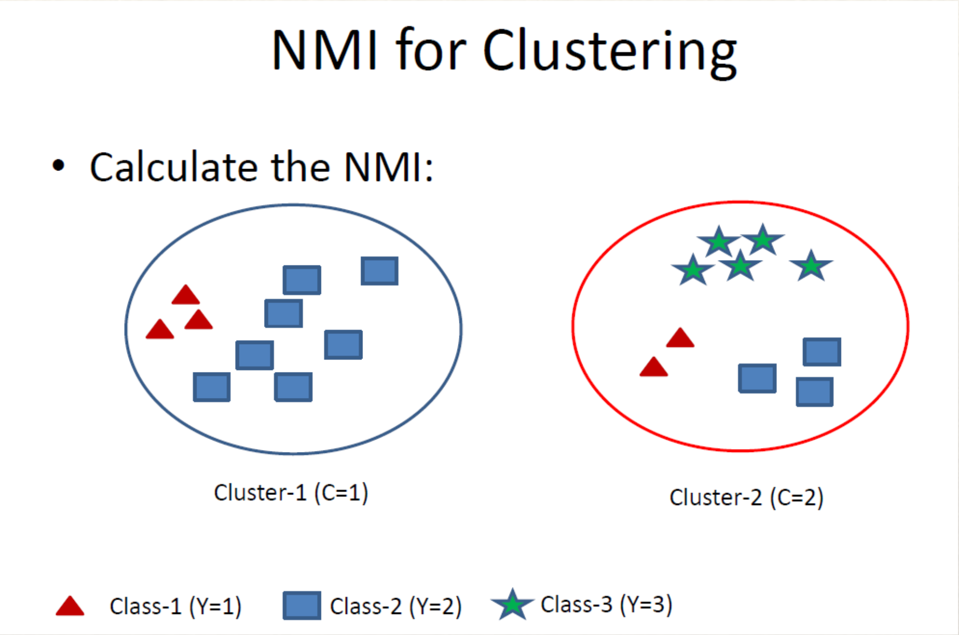
- Finally the mutual information is:

$$\begin{aligned} I(C;\ \Omega) &= H(C) - H(C|\Omega) \\ &= 1.5 - [0.4406 + 0.7427] \\ &= 0.3167 \end{aligned}$$

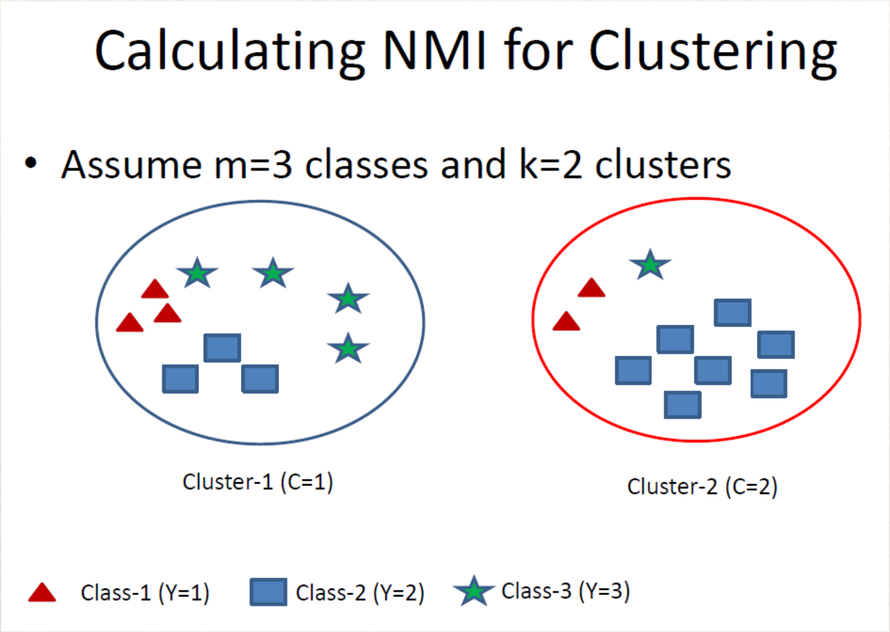
The NMI is therefore,

$$NMI(C, \Omega) = \frac{I(C;\ \Omega)}{[H(C) + H(\Omega)]/2}$$

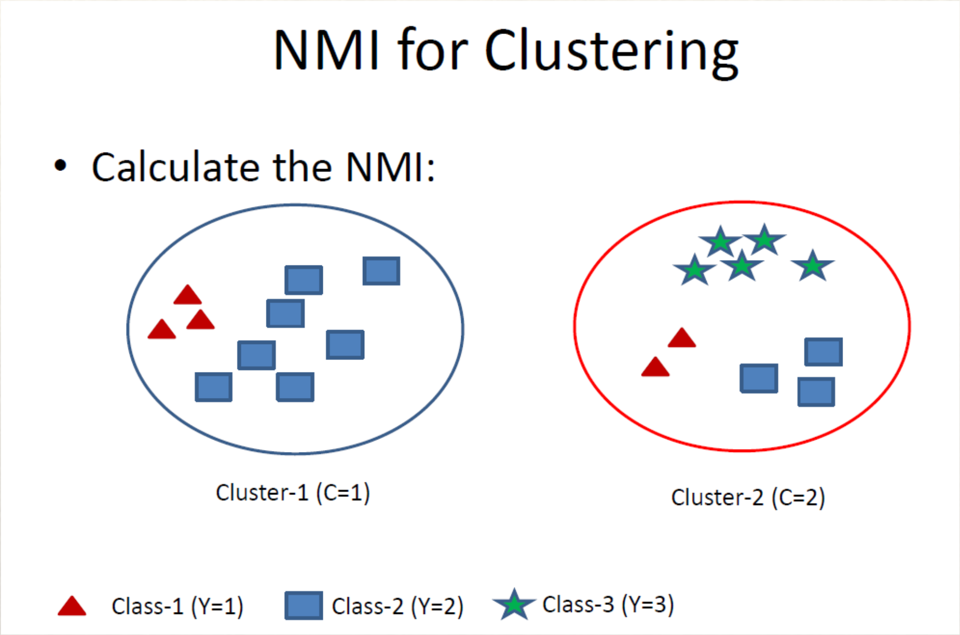
$$NMI(C, \Omega) = \frac{0.3167}{[1.5 + 1]/2} = 0.2533$$



Cluster evaluation – evaluating label data with NMI - example



VS.



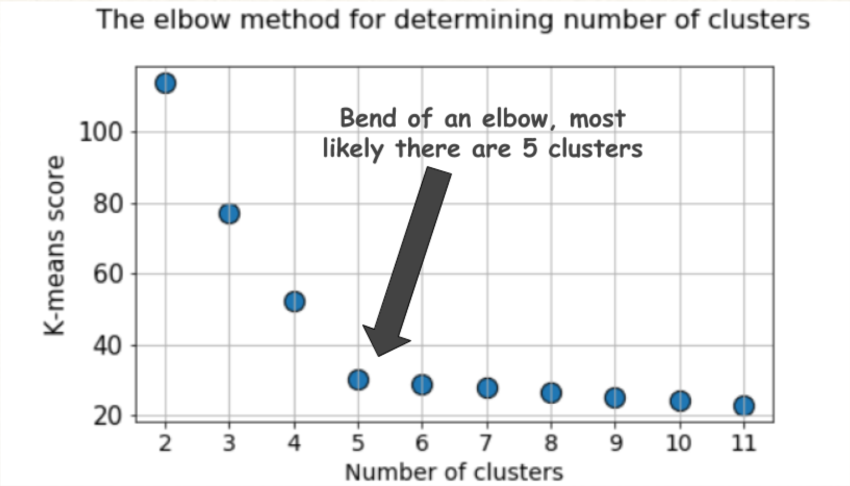
NMI = 0.1089

NMI = 0.2533

Better clustering

K-means – Choosing K w/the elbow method

The basic idea is the notion that we could see a drop in the error for higher ks



<u>The elbow method</u>:

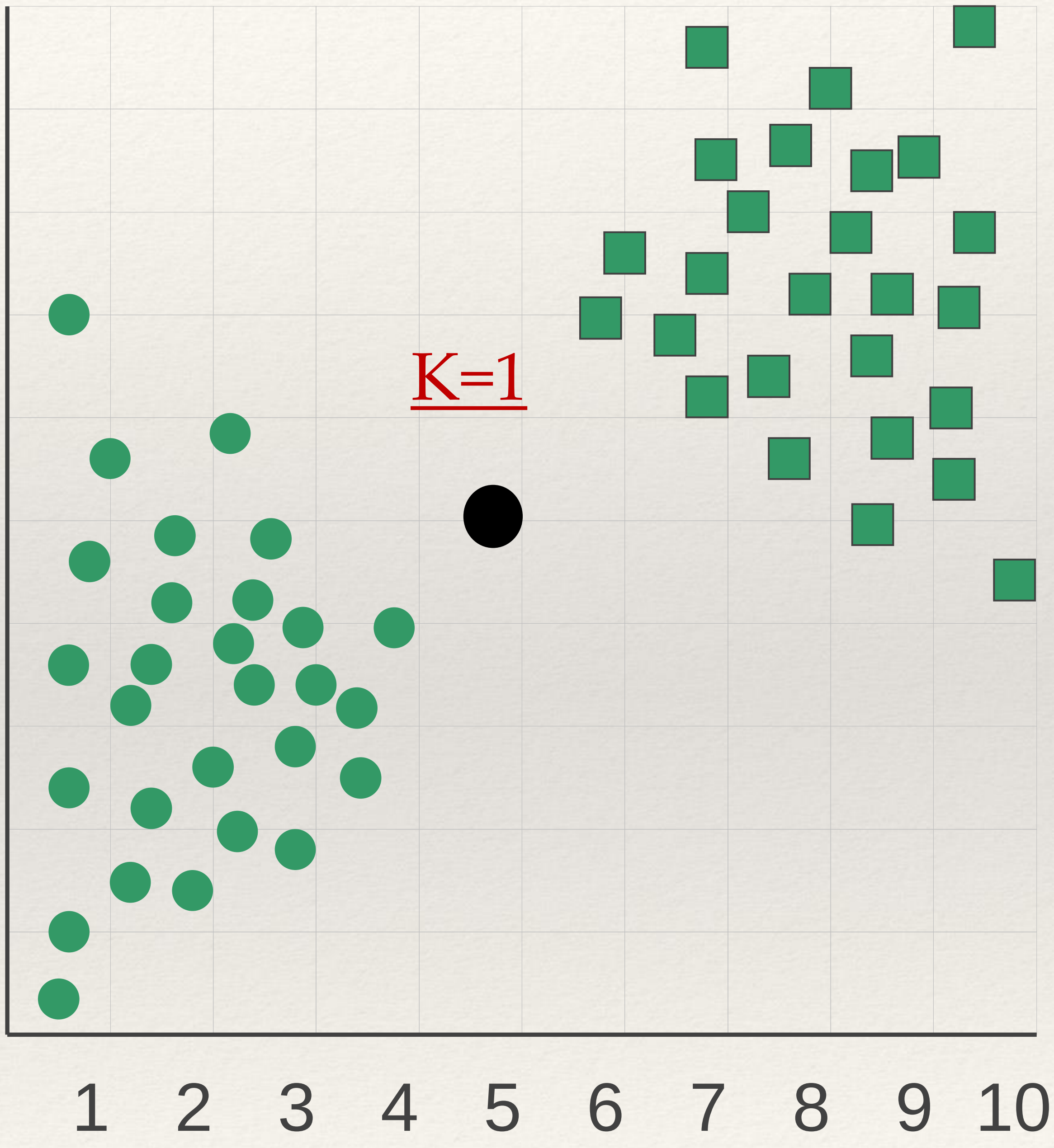
- Loop over the k-means algorithm
- Every iteration – increase k
- Plot the scores and check for the bend

<u>**Question**</u>: How will we decide about the quality of the clustering result?

<u>**Answer**</u>: We could use the WSS (within cluster SSE) as a measurement

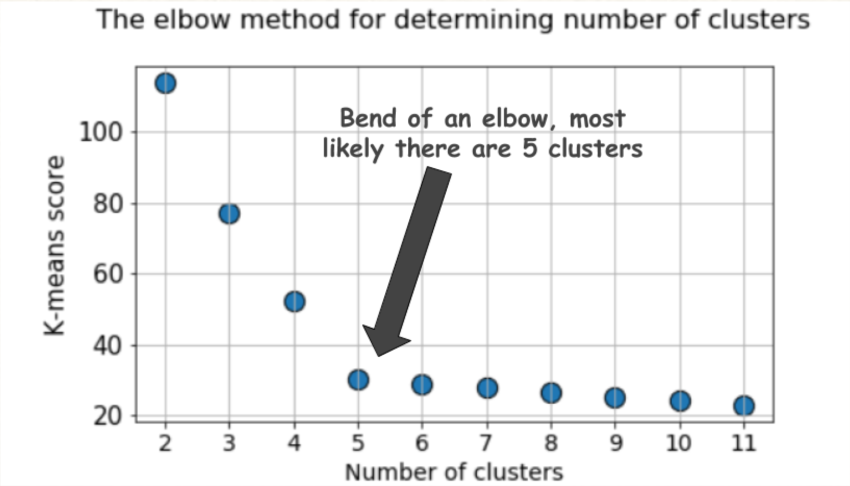
<u>When k = 1</u>

WSS=873.0



K-means – Choosing K w/the elbow method

The basic idea is the notion that we could see a drop in the error for higher ks



<u>The elbow method</u>:

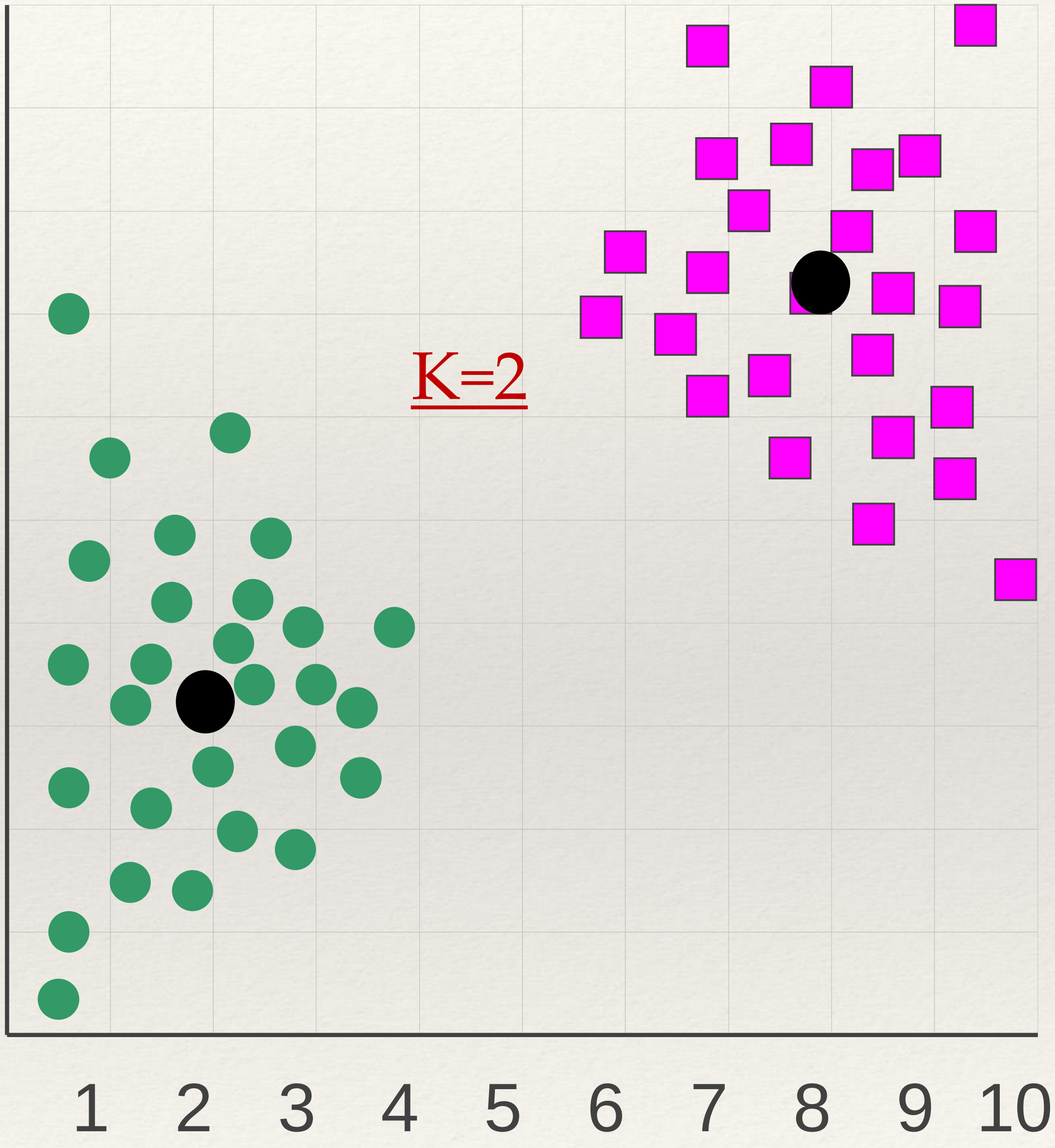
- Loop over the k-means algorithm
- Every iteration – increase k
- Plot the scores and check for the bend

<u>**Question**</u>: How will we decide about the quality of the clustering result?

<u>**Answer**</u>: We could use the WSS (within cluster SSE) as a measurement

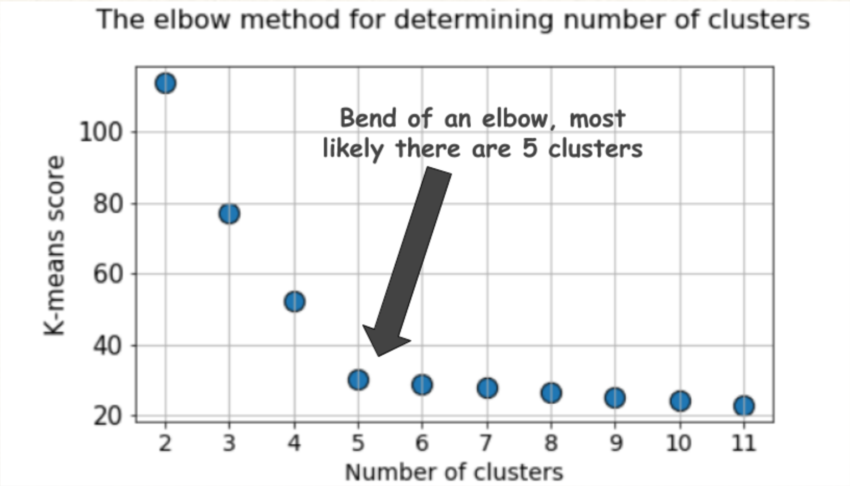
<u>When k = 2</u>

WSS=173.1



K-means – Choosing K w/the elbow method

The basic idea is the notion that we could see a drop in the error for higher ks



The elbow method:

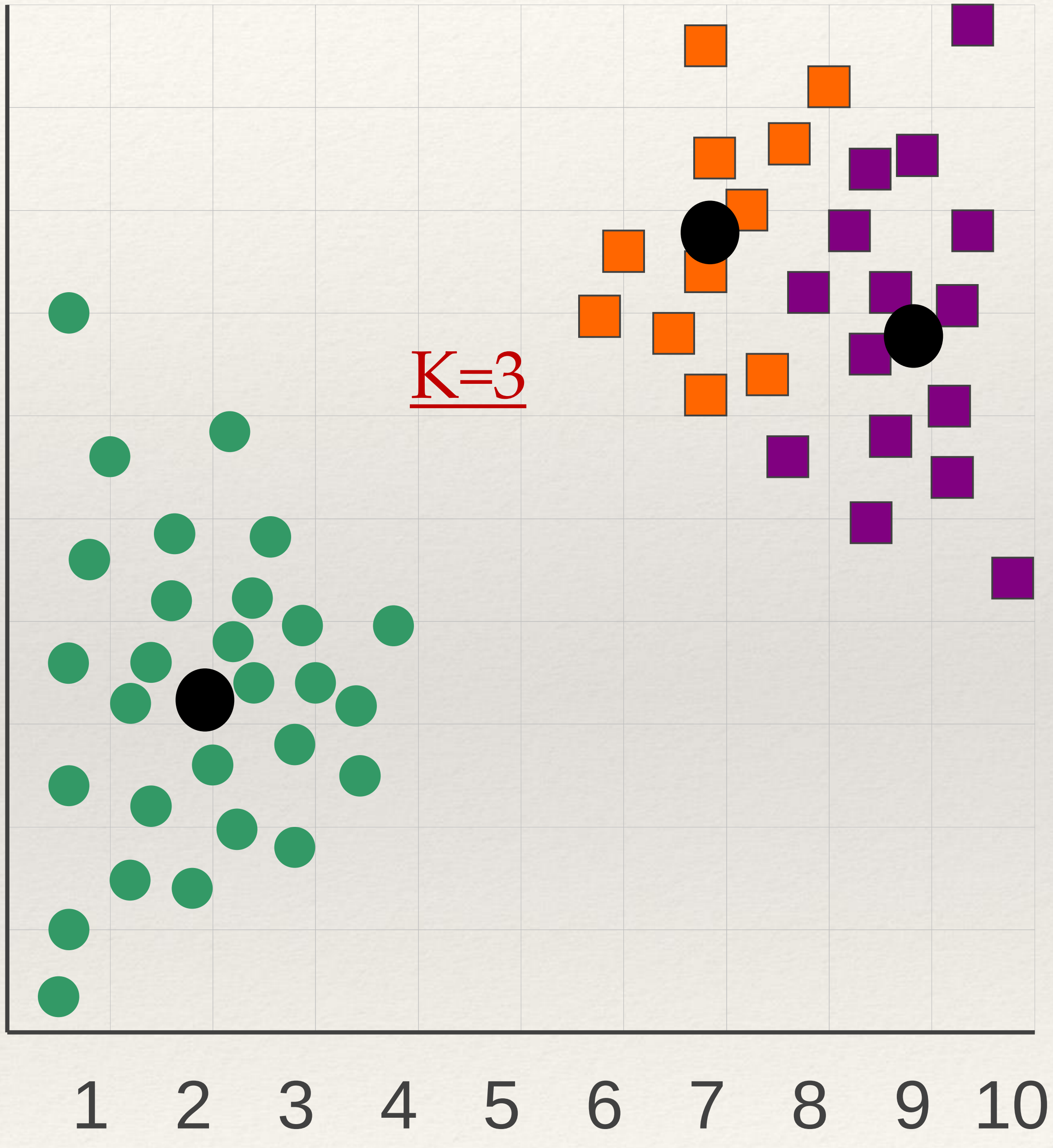
- Loop over the k-means algorithm
- Every iteration – increase k
- Plot the scores and check for the bend

Question: How will we decide about the quality of the clustering result?

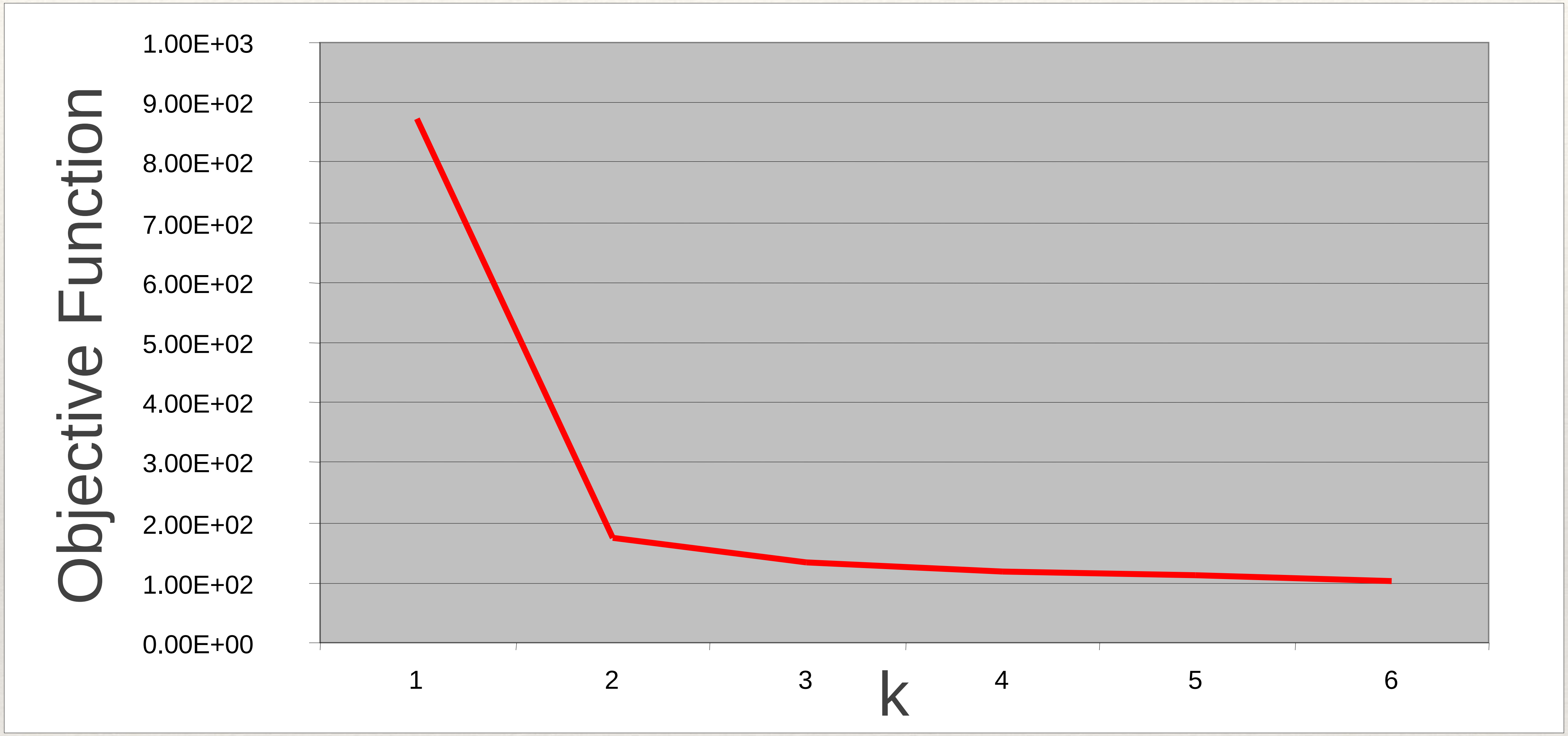
Answer: We could use the WSS (within cluster SSE) as a measurement

When k = 3

WSS=133.6



K-means – Choosing K w/the elbow method



If we choose k=1..6
We can plot the error function values

The abrupt change at k = 2,
→ We choose k=2

<u>WSS for k=1..6:</u>

$WSS_{k=1}$=873.0
$WSS_{k=2}$=173.1
$WSS_{k=3}$=133.6…

Choosing K w/ the Silhouette coefficient – K-means

a(i) = עבור וקטור i, השייך לcluster מסויים C_i, יתן את ממוצע המרחקים מוקטור i לכל שאר הוקטורים ב-cluster זה

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j)$$

b(i) = א. עבור וקטור i, השייך לcluster מסויים C_i, עבור cluster מסויים C_k, כך ש $C_k \neq C_i$, ניקח את כל הוקטורים השייכים ל-C_k ונחשב את ממוצע המרחקים מוקטור i לכל שאר הוקטורים ב-cluster זה (כלומר, כביכול נוסיף את וקטור i ל-C_k ואז נחשב את a(i)).

ב. נעשה את החישוב הנ"ל עבור כל C_k, כך ש $C_k \neq C_i$, וניקח את ה-minimum (כלומר את ה- a(i) המינימילי אם הוא היה ב-cluster אחר C_k, כך ש $C_k \neq C_i$)

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

s(i) = עבור וקטור i, השייך ל-C_i, יתן את ההפרש בין b(i) ל-a(i) , חלקי המקסימום ביניהם.

- 1-<s(i)<1
- כאשר השיוך של וקטור i מתאים יותר ל C_i, a(i)<b(i) ו-s(i) יהיה חיובי.
- כאשר s(i)קרוב ל-1 ניתן לומר כי הוקטור i מתאים C_i.
  - ערך שכזה מתקבל כאשר ערך הלכידות קטנה בצורה משמעותית מערך ההפרדה.
- כאשר s(i)קרוב ל-0 ניתן לומר כי הנתון נמצא קרוב מאוד לגבול בין שני אשכולות שכנים.
- כאשר s(i)קרוב ל- (-1) ניתן לומר כי הנתון נמצא באשכול שלא מתאים לו.

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

$\tilde{s}(k)$ = ה- *silhouette score* עבור כמות clusters K=k, יתן את הערך הממוצע של s(i), עבור כל הוקטורים ב-dataset.

- ככל ש-$\tilde{s}(k)$ קרוב ל-1, אומר שיש השמה נכונה, עבור רוב הוקטורים ב-dataset, ככל ש-$\tilde{s}(k)$ קרוב ל-1- אומר שיש השמה <u>לא נכונה</u>, עבור רוב הוקטורים ב-dataset

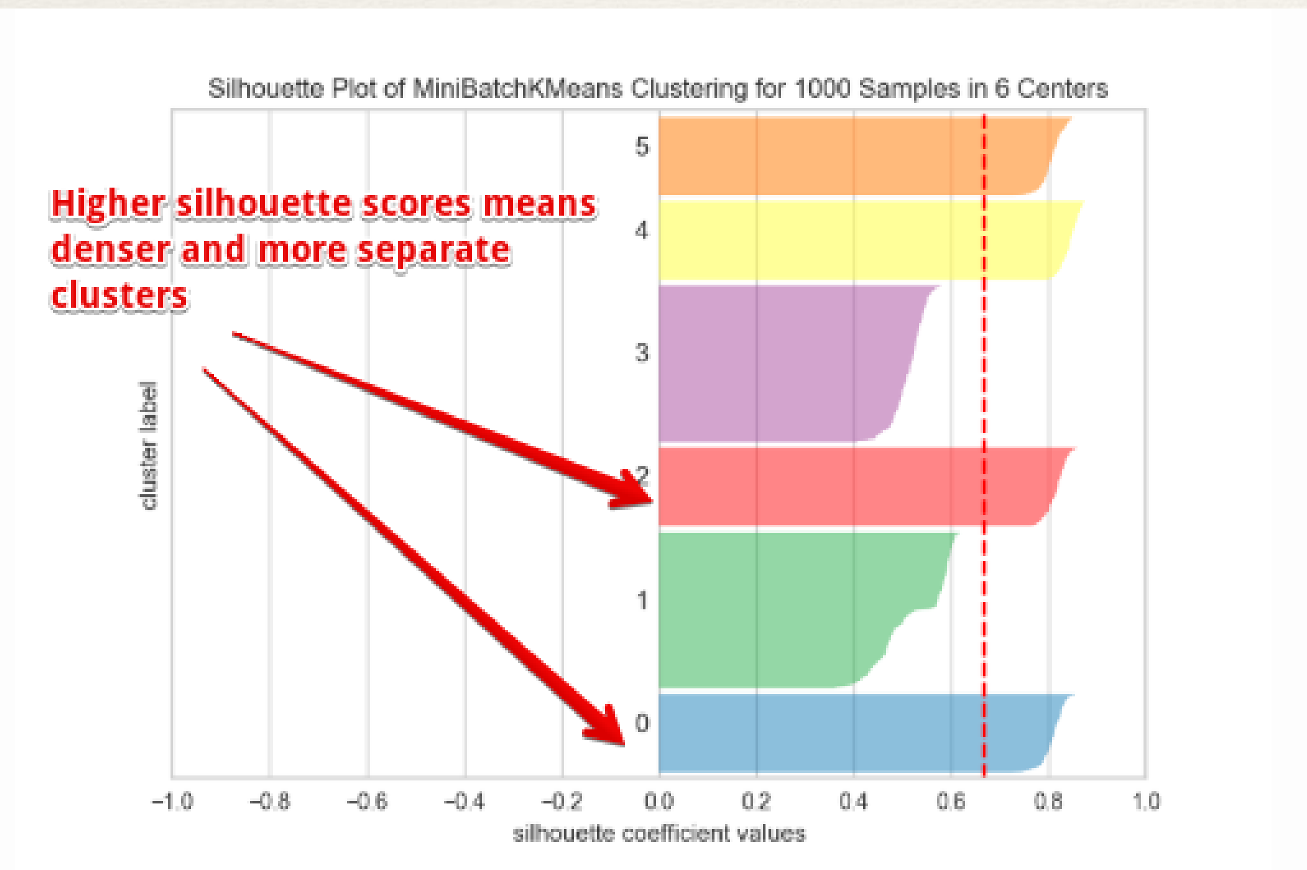
$\tilde{s}\,(k)$ represents the mean $s(i)$

SC= עבור ערכי k שונים, $\tilde{s}(k)$ המקסימלי.

$$SC = \max_k \tilde{s}\,(k)$$

Choosing K w/ the Silhouette coefficient – K-means

ערכי s(i) בממוצע עבור כל cluster, כאשר k=6 (ישנם 6 clusters)



עבור וקטור i, השייך לcluster מסויים C_i

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j)$$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

עבור כמות K=k של clusters, יתן את הערך הממוצע של s(i)

$\tilde{s}\ (k)$ represents the mean $s(i)$

$\tilde{s}(k)$ המקסימלי, עבור ערכי k שונים

$$SC = \max_k \tilde{s}\ (k)$$

K-means – Choosing K w/ the Silhouette coefficient – how should we use it?

1. choose range for k= (let's say we chose) 2..7
2. Run the k-means algorithm and calculate the Silhouette coefficient:

$\tilde{s}(k=2)$: 0.028

$\tilde{s}(k=3)$: 0.024

$\tilde{s}(k=4)$: 0.013

$\tilde{s}(k=5)$: 0.014

$\tilde{s}(k=6)$: 0.016

$\tilde{s}(k=7)$:-0.003

3. Select k=2, since it is maximizing $SC = \max_k \tilde{s}(k)$