

# Information Retrieval

- Clustering
- Relevance Feedback (part2)
- Query Expansion

Development:  
Moshe Friedman

Credits:

Yoav Goldberg, Ido Dagan, Reut Tsarfaty , Moshe Koppel, Wei Song,  
David Bamman, Ed Grefenstette, Chris Manning, Tsvi Kuflik,  
Hinrich Schütze, Christina Lioma and more

# Information Retrieval - administration

Moshe Friedman

Email: [moshefr.teach@gmail.com](mailto:moshefr.teach@gmail.com)

Reception time: before/after lesson/zoom with coordination

# אשכול - מוטיבציה - חלוקת קבוצת לקוחות לקבוצות



- למנהל קשרי לקוחות יש חמישה עובדים ורוצה לחלק את הלקוחות ל-5 קבוצות כך שכל קבוצה תשוך 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?

# למידה מונחית מול למידה לא מונחית

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

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

# סוגי בעיות בלמידה לא מונחת

**Clustering (אשכול):** נייצג כל דוגמה על ידי "אב-טיפוס" (prototype), למשל k-means, GMM ואחרים.

**Dimensionality reduction (הורדת הממדיות):** נייצג כל דוגמה על ידי מספר קטן יותר של מאפיינים. למשל Principal Components Analysis, Factor Analysis ואחרים.

**Density estimation (הערכת צפיפות):** נעריך את ההתפלגות מעל מרחב ה-data

# "אישכול" Clustering

• Cluster Analysis היא הפעולה של חלוקת קבוצה לתתי קבוצות ("אשכולות"/Clusters) כך ש:

- אובייקטים באותו אשכול "דומים" זה לזה
- אובייקטים באשכולות שונים, אינם "דומים" זה לזה.

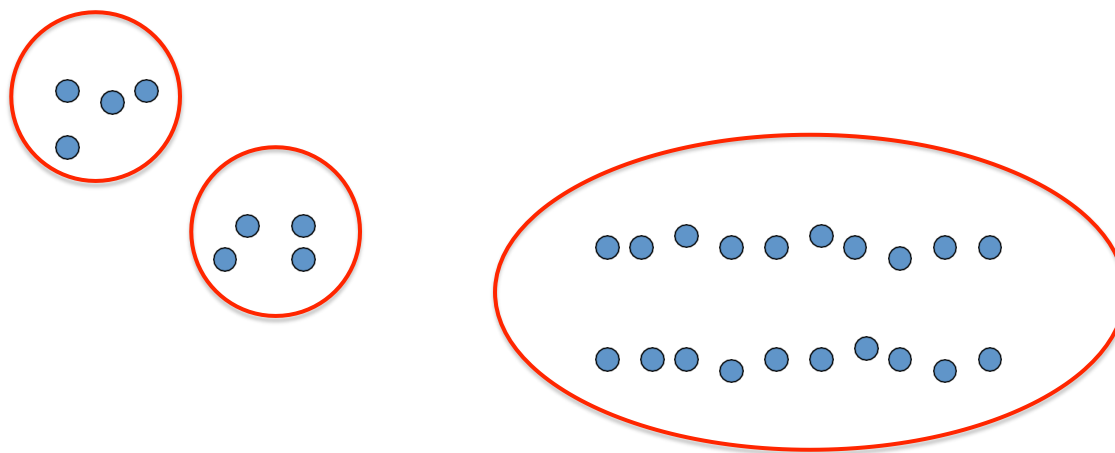
## Unsupervised•

# Clustering – שאלה 1: איזו חלוקה מהווה חלוקה "נכונה"?

## אפשרות א'

- ❖ רעיון בסיסי: לקבץ יחד דוגמאות "דומות"
- למשל רוצים לקבץ ביחד לקוחות "דומים" לקבוצות (למשל ע"מ שנוכל למכור ולתמוך בהם באופן דומה).

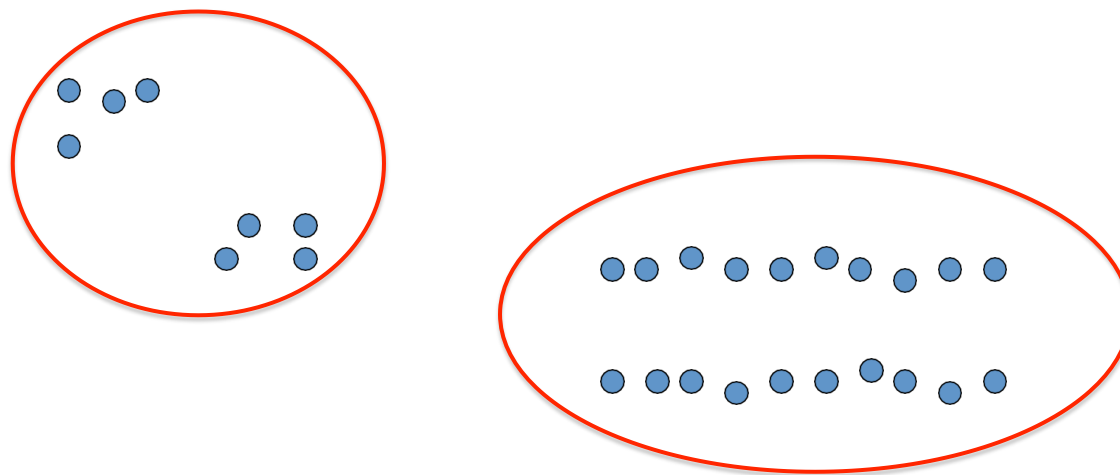
❖ דוגמא: תבניות של נקודות בדו-מימד



# Clustering – שאלה 1: איזו חלוקה מהווה חלוקה "נכונה"?

## אפשרות ב'

- רעיון בסיסי: לקבץ יחד דוגמאות "דומות"
- למשל רוצים לקבץ ביחד לקוחות "דומים" לקבוצות (למשל ע"מ שנוכל למכור ולתמוך בהם באופן דומה).
- דוגמא: תבניות של נקודות בדו-מימד

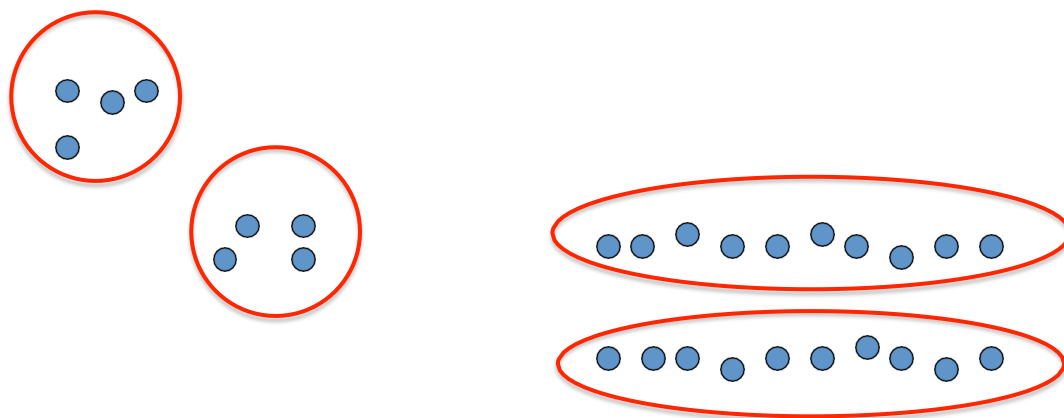




# Clustering – שאלה 1: איזו חלוקה מהווה חלוקה "נכונה"?

## אפשרות ג'

- רעיון בסיסי: לקבץ יחד דוגמאות "דומות"
- למשל רוצים לקבץ ביחד לקוחות "דומים" לקבוצות (למשל ע"מ שנוכל למכור ולתמוך בהם באופן דומה).
- דוגמא: תבניות של נקודות בדו-מימד



# Clustering - שאלה 2: כיצד נמדוד דמיון?



Similarity is  
hard to define,  
but...  
*"We know it  
when we see it"*

Credit: Eamonn  
Keogh

כל הזכויות שמורות למשה פרידמן וד"ר יהונתן שלר ©

# Clustering - שאלה 2: כיצד נמדוד דמיון?

❖ רעיון בסיסי: לקבץ יחד דוגמאות דומות

❖ דוגמאות:

- תבניות של נקודות בדו-מימד
- דוגמה נוספת – קיבוץ לקוחות דומים.

❖ כיצד נמדוד "דמיון"?

- אפשרות אחת: 2 נקודות (דוגמאות) יחשבו "דומות", אם יהיה ביניהן מרחק קטן.
- למשל מרחק אוקלידי קטן:  $\text{dist}(\vec{x}_1, \vec{x}_2) = \|\vec{x}_1 - \vec{x}_2\|_2$
- מסקנה 1: כמו שכבר מבינים: תוצאות האשכול תלויות במידה רבה בפונקציות המרחק אותן נבחר..

# Clustering - נניח שנמדוד דמיון על ידי מרחק (קטן) שאלה 3: בין מי למי מודדים מרחק?

מוטיבציה: רוצים לחלק את הלקוחות לקבוצות.

- נוכל להחליט על "דמיון" בין הלקוחות, על ידי מציאת לקוחות עם מרחק (קטן ביניהם), אך בין מי למי מודדים את המרחק?

- חלק מהאלגוריתמים דורשים **מרחק בין נקודה  $x_i$**  (דוגמה  $x_i$ , או לקוח מסוים, כמו במקרה שלנו) **לבין קבוצת נקודות A** (קבוצת דוגמאות A, או קבוצת לקוחות, כמו במקרה שלנו).

- במקרה זה נמדוד את המרחק  $d(x, A)$

- אלגוריתמים אחרים דורשים **מרחק בין קבוצת נקודות A** (קבוצת דוגמאות A, או קבוצת לקוחות, במקרה שלנו) **לבין קבוצת נקודות B** (קבוצת דוגמאות B, או קבוצת לקוחות אחרת, במקרה שלנו).

- במקרה זה נמדוד את המרחק  $d(A, B)$

# Clustering - מוטיבציה אפליקטיבית

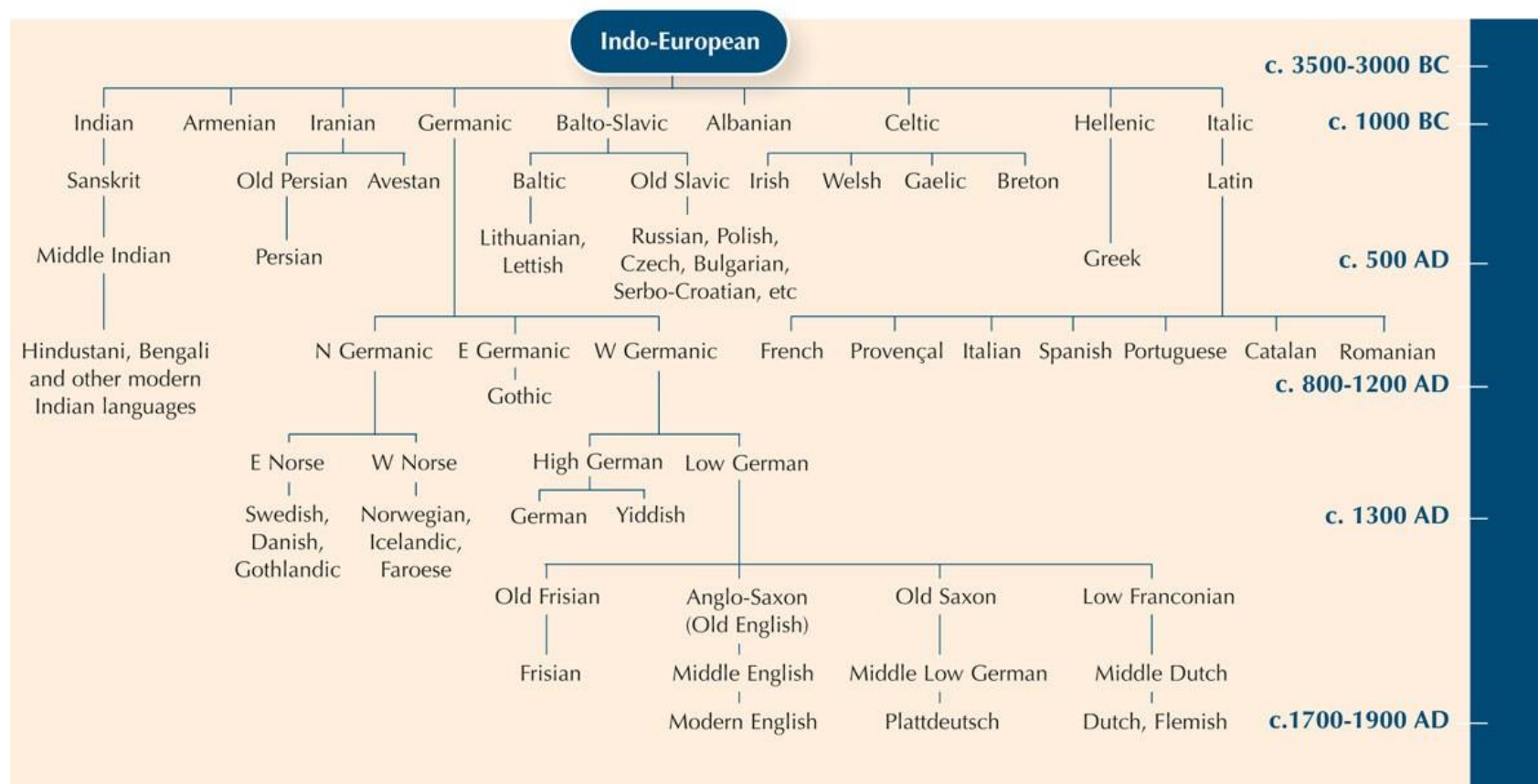
ניקח צעד אחד אחורה ...

מוטיבציה אפליקטיביות:

- עבור איזה סוגי מידע נרצה לבצע clustering?
- דוגמאות עבור אפליקציות ל-clustering

# Clustering - מוטיבציה אפליקטיבית

## דוגמאות אפליקטיביות

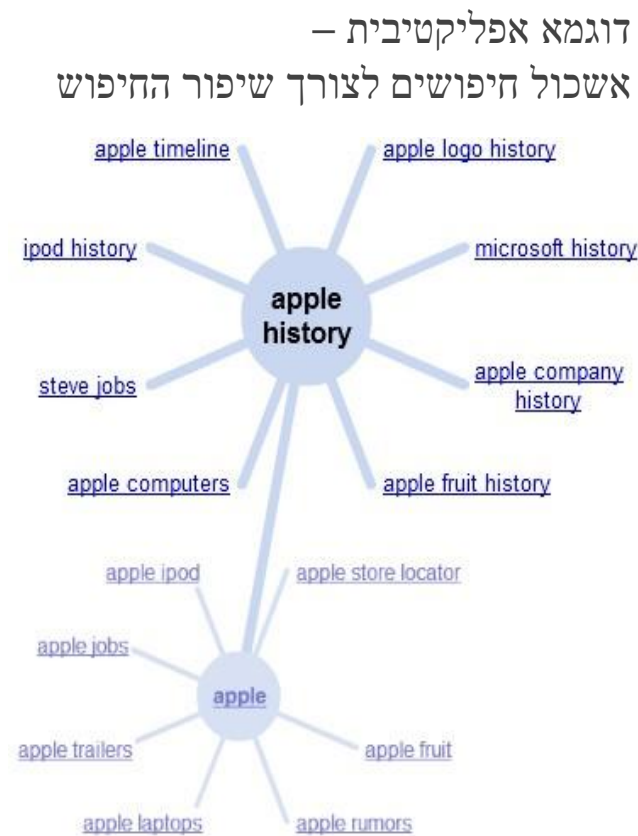


[Image from scienceinschool.org]

# Clustering - מוטיבציה אפליקטיבית

## סוגי data שונים עבור clustering

- Numerical data
- Categorical data: e.g. demographic, many times binary (has some category or not)
- Text data (popular in social media, web, social nets):
  - Features: high dimensional, sparse, values corresponding to word frequencies
  - Methods: combinations of: k-means, agglomerative (hierarchical); topic modeling; co-clustering





# Clustering - מוטיבציה אפליקטיבית

## דוגמאות אפליקטיביות – אשכול מאמרי עיתונות

Cluster news  
articles

The screenshot shows the Google News homepage. On the left is a sidebar with categories: Top Stories, Recommended, U.S., World, Sci/Tech, Business, More Top Stories, Health, Spotlight, Elections, Entertainment, Sports, Technology, and Science. The main content area displays several news stories under the 'Top Stories' heading. Each story includes a headline, source, time, and a brief summary. To the right of each story is a small thumbnail image. The stories shown are: 1. 'Teen suspect saw movie moments after allegedly killing beloved Massachusetts ...' from Fox News, featuring a photo of a young woman. 2. 'Obamacare contractors tell their stories at congressional hearing' from CNN, featuring a photo of a woman. 3. 'EU leaders meet amid concern about US spying claims' from CNN, featuring a photo of Barack Obama. 4. 'US jobless claims miss forecasts, trade deficit widens slightly' from Reuters, featuring a photo of a man at a desk. 5. 'Kennedy cousin gets new trial in 1975 killing of neighbor; victim's mother ...' (partially visible at the bottom).



# Clustering - מוטיבציה אפליקטיבית

## סוגי data שונים עבור clustering

- Multimedia data [image, audio, video] (e.g., flicker, YouTube):
  - Multi-model (often combine with text data)
  - Contextual: containing both behavioral and contextual attributes
  - Images: position of pixel represents its context, value represents its behavior
  - Video & music: temporal ordering of records represent its meaning

דוגמא אפליקטיבית –  
– Image segmentation  
אשכול מקטעים בתמונה

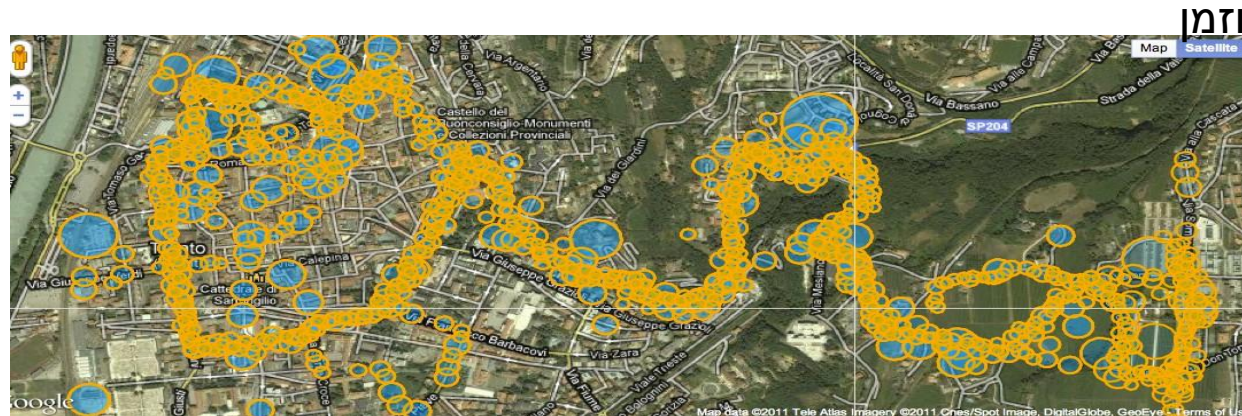


# Clustering - מוטיבציה אפליקטיבית

## סוגי data שונים עבור clustering

- Time-series data: sensor data, stock market, temporal tracking, forecasting and so on data is temporal dependent
  - time: context, data: behavioral
  - correlation based online analysis (e.g., online clustering of stocks to find stock trickers)
  - shape-based offline analysis (e.g., cluster ECG based on overall shapes)

דוגמא אפליקטיבית –  
אשכול אנשים לפי מקום



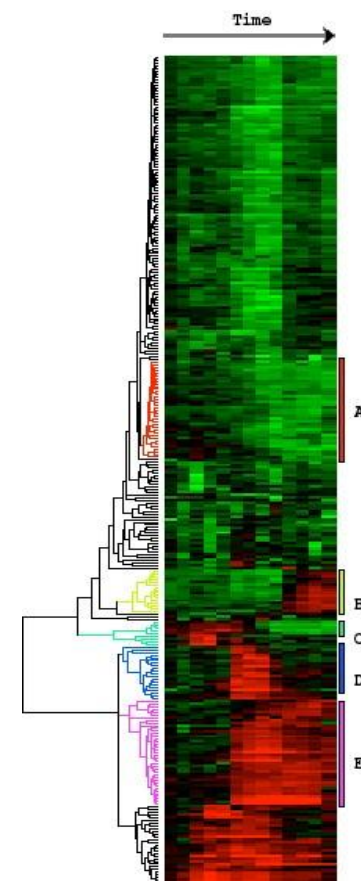
[Image from Pilho Kim]

# Clustering - מוטיבציה אפליקטיבית

## סוגי data שונים עבור clustering

- sequence data: weblogs, biological sequences, system command sequences
  - contextual attributes: Placement (rather than time)
  - Similarity attributes : hamming distance, edit distance, longest common sequence
  - sequence clustering: suffix trees, generative model (e.g. HMM - hidden markov model)

דוגמא אפליקטיבית –  
אשכול micro-arrays



Eisen et al, PNAS 1998

# Clustering – סיכום ביניים

מה הבנו עד כה?

- כמה שאלות בסיסיות, כמו:
  - איך ניצור את ה-clusters (לא ענינו על השאלה הזו)
  - איך נמדוד דמיון
  - בין מה למה נמדוד דמיון
- בנוסף, הבנו את המוטיבציה האפליקטיבית לשימוש ב-clustering

הנושאים (והשאלות) הבאים בהם נדון:

- התכונות הרצויות של אלגוריתם clustering
- הגישות המרכזיות לביצוע clustering

## – Clustering

גישות מרכזיות כיצד לבצע Clustering (אלגוריתמית)

1. שיטות מבוססות חלוקה (Partitioning)

2. שיטות היררכיות

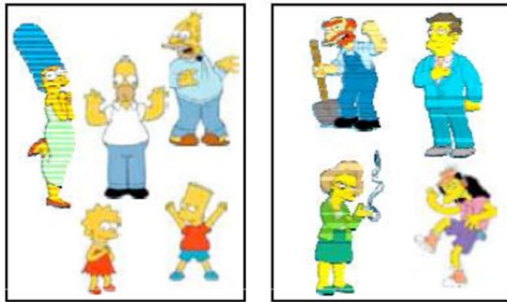
3. שיטות מבוססות צפיפות (Density Based)

4. הסתברותי וגנרטיבי

# גישות מרכזיות ב-Clustering

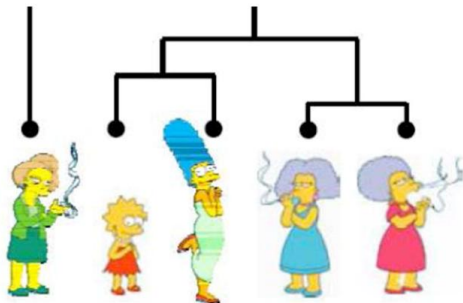
## 1. שיטות מבוססות חלוקה (Partitioning) –

- בהינתן קבוצה של  $n$  אובייקטים, חלק ל- $k$  תתי קבוצות ( $k \leq n$ ). כל תת קבוצה צריכה להכיל אובייקט אחד לפחות וכל אובייקט משויך לקבוצה אחת בלבד.



## 2. שיטות היררכיות –

- בונים מבנה היררכי של תתי הקבוצות גישות:



• Agglomerative (bottom-up)

• Divisive (top-down)

# גישות מרכזיות ב-Clustering

3. מבוססות צפיפות (Density Based) – לא נסתכל רק על המרחק בין הנקודות על גם האם יש "מסלול" ביניהן

4. הסתברותי וגנרטיבי:

- מניחים תצורה מסוימת של מודל גנרטיבי (mixture of Gaussian)
- שערך הפרמטרים בעזרת אלגוריתם expectation maximization (EM) ומשתמשים ב-dataset, כדי לשערך maximum likelihood
- שערך ההסתברות הגנרטיבית של נקודת נתונות.
- יש גמישות לכל נקודה להיות שייכת לכמה clusters

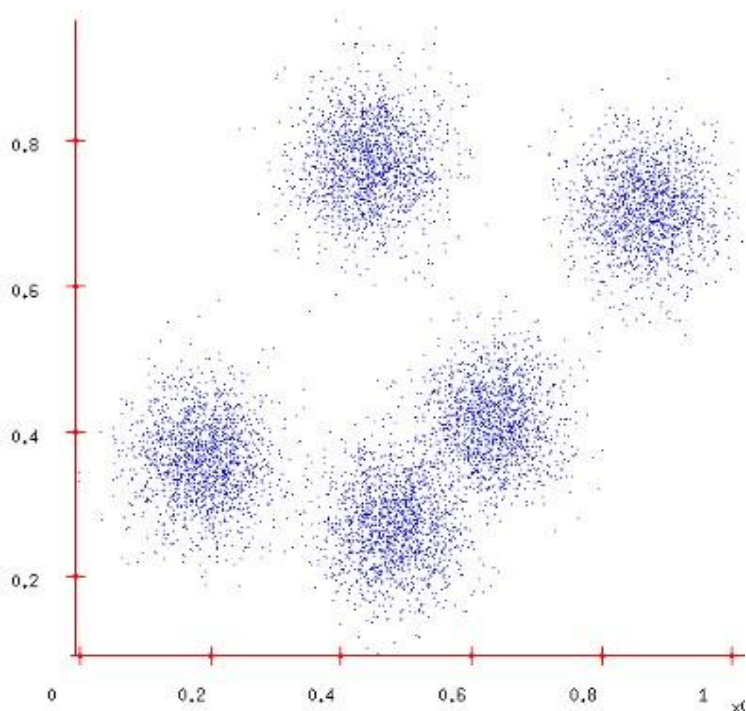
# שיטות מבוססות חלוקה ואלגוריתם k-means



# גישה 1: שיטות מבוססות חלוקה

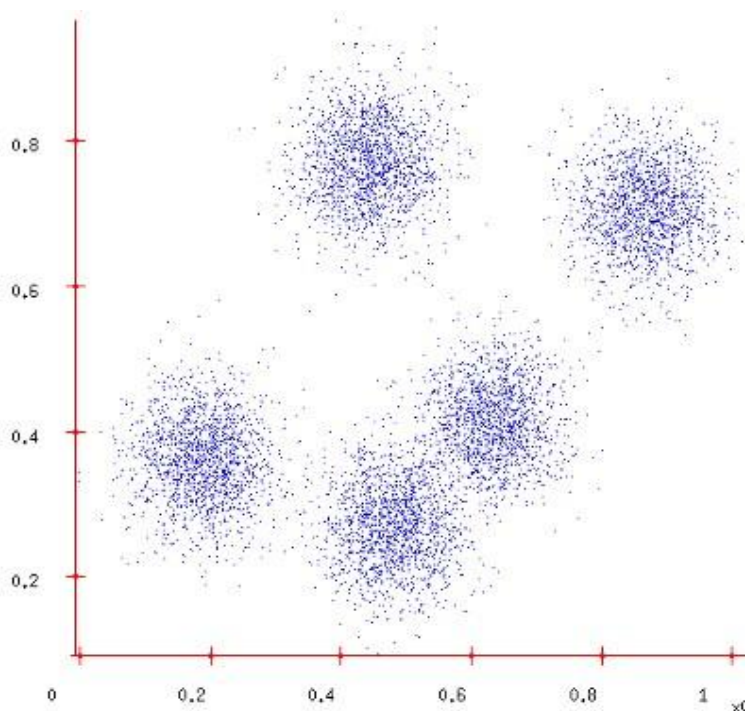
- המטרה - בהינתן K (כקלט לאלגוריתם) - מצא חלוקה ל-K אשכולות שמביאה אותנו לאופטימיזציה
- השאיפה - אופטימום גלובלי – עבור על כל החלוקות האפשריות ובחר את הטובה ביותר.
- היצוג - כל אשכול מיוצג ע"י אב-טיפוס (prototype) - מרכז האשכול

# 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 ייצוג ה-cluster ע"י אב-טיפוס (prototype)

ייצוג ע"י אב-טיפוס (prototype) – לכל cluster יש אב-טיפוס שמייצג את הווקטורים 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 ידי 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

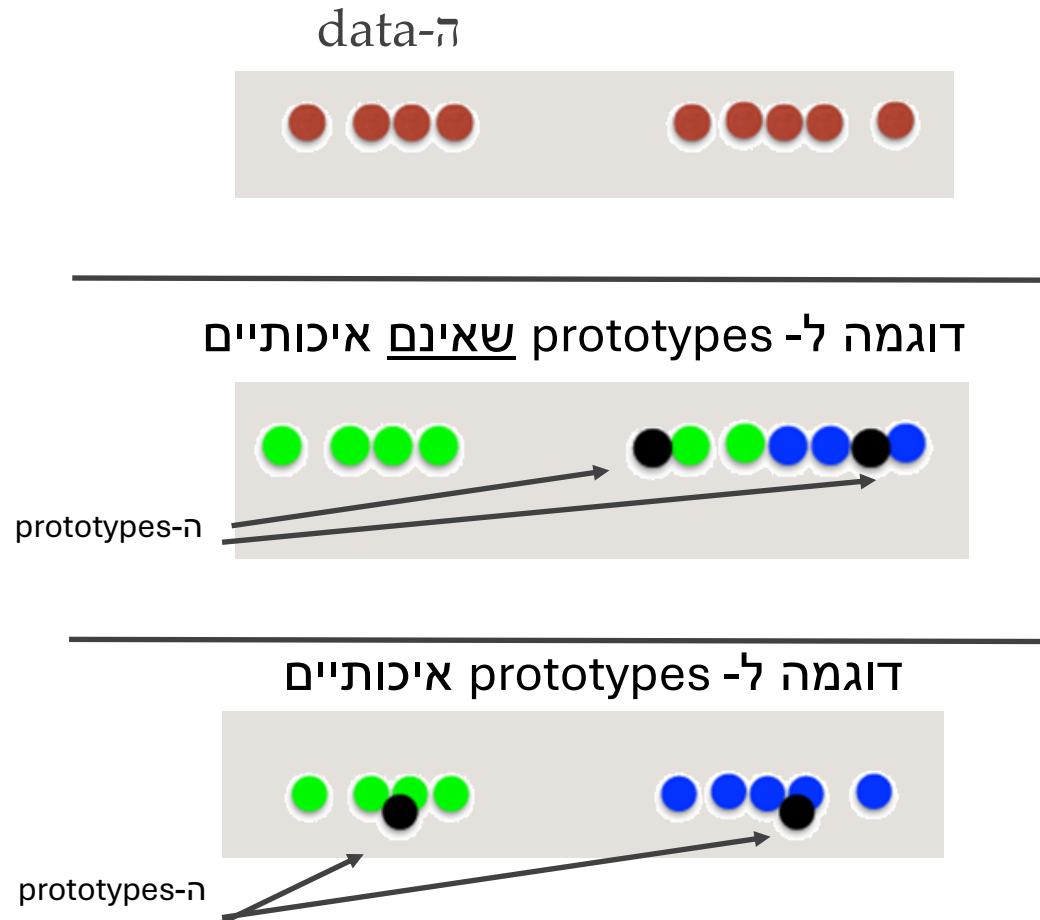
- נניח שמספר ה-clusters הוא  $k$
- הנחה של prototype אחד עבור כל cluster
- נסמן את ה-prototypes ע"י  $\mu_1, \dots, \mu_k$  (כזכור  $\mu$  מייצג תוחלת).
- לעיתים מסמנים את ה-prototype כ- $m$  (המסמן mean – ממוצע), או ע"י  $c$  (המסמן center – מרכז).

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



# ג'ישה 1: שיטות מבוססות חלוקה

## K-Means - המטרה



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

# גישה 1: שיטות מבוססות חלוקה

## K-Means – המטרה – מינימיזציה של המרחק

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

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

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

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

- הפונקציה אינה פונקציה קמורה (פונקציית 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 פונקציית המטרה)





# פונקציות מרחק שניתן להשתמש לצורך clustering - תזכורת

$$d(\vec{x}_j, \vec{x}_i) = \sum_{m=1}^d \sqrt{(x_{j_m} - x_{i_m})^2} : \text{ב K-means משתמשים בעיקר במרחק אוקלידי}$$

פונקציות מיניקובסקי:

$$d(\vec{x}_j, \vec{x}_i) = \left( \sum_{m=1}^d |x_{j_m} - x_{i_m}|^p \right)^{\frac{1}{p}} : \text{מרחק מיניקובסקי}$$

$$d(\vec{x}_j, \vec{x}_i) = \sum_{m=1}^d |x_{j_m} - x_{i_m}| : \text{מרחק מנהטן}$$

$$d(\vec{x}_j, \vec{x}_i) = \max_{1 \leq m \leq d} |x_{j_m} - x_{i_m}| : \text{מרחק צ'בישב}$$

אפשרויות נוספות:

$$d(\vec{x}_j, \vec{x}_i) = \frac{\vec{x}_j^T \cdot \vec{x}_i}{\|\vec{x}_j\| \cdot \|\vec{x}_i\|} : \text{Cosine similarity}$$

Edit distance

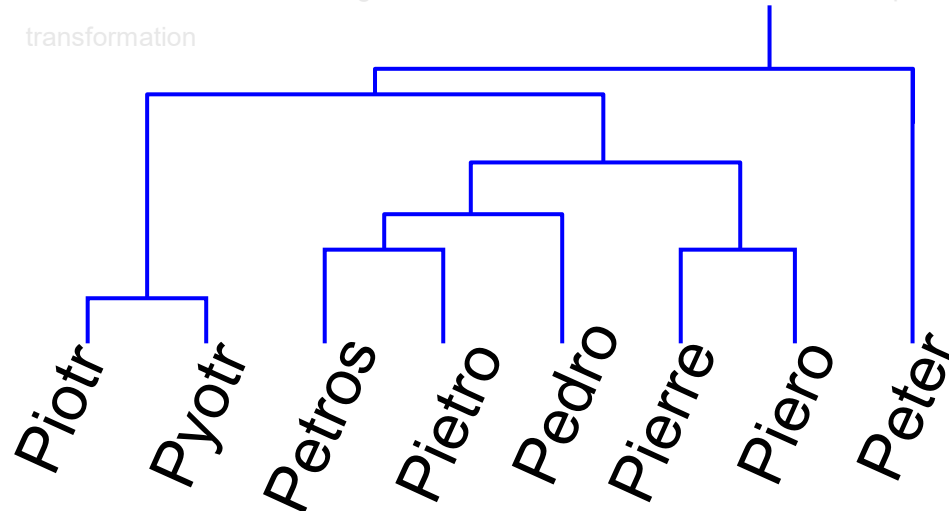
# Edit Distance Example

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

<i>Substitution</i>	1 Unit
<i>Insertion</i>	1 Unit
<i>Deletion</i>	1 Unit

$D(\text{Peter}, \text{Piotr})$  is 3

**Peter**



**Piter**



**Pioter**



**Piotr**

כל הזכויות שמורות למשה פרידמן וד"ר יהונתן שלר ©

# Cosine similarity measure - Reminder

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  $L_2$

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

# Cosine Similarity Example – Reminder

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

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

# Cosine Similarity Example – Solution - Reminder

- Denote the first two term-frequency vectors as  $\vec{x}, \vec{y}$

- $\vec{x} = (5,0,3,0,2,0,0,2,0,0)$

- $\vec{y} = (3,0,2,0,1,1,0,1,0,1)$

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

Exercise: Calculate the cosine similarity.

- Assume normalization with  $L_2$

- $\|\vec{x}\| = \sqrt{5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2} = 6.48$

- $\|\vec{y}\| = \sqrt{3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2} = 4.12$

- $\vec{x}^T \cdot \vec{y} = 5 \cdot 3 + 0 \cdot 0 + 3 \cdot 2 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot 1 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot 0 + 0 \cdot 1 = 25$

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

# אלגוריתם K-means

- נתון: אוסף ווקטורים (feature vectors) והפרמטרים:  $K$  ופונקציית המרחק
- מצא חלוקה אופטימאלית שמחלקת ל- $K$  אשכולות



- אלגוריתם:

1. אתחול - "נחש"  $K$  מרכזים
2. שיטת Lloyd - כל ה"נקודות" במרחב המדגם הם מועמדים פוטנציאליים.
3. שייך כל ווקטור ל"מרכז" הקרוב אליו
4. חשב מרכזים מחדש ע"י מציאת מרכז האשכול
4. עצירה - חזור על צעדים 2-3 עד שאין יותר עדכונים

# אלגוריתם K-means - דוגמה

- נתון: אוסף ווקטורים והפרמטרים:  $K$  ופונקציית המרחק

- מצא חלוקה אופטימאלית שמחלקת ל- $K$  אשכולות

- נתון  $k=2$  (2 clusters צריך למצוא 2 prototypes)

- אלגוריתם:

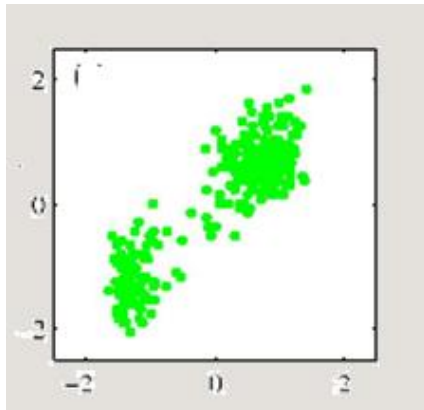
1. אתחול - "נחש"  $K$  מרכזים

2. שייך כל ווקטור ל"מרכז" הקרוב אליו

3. חשב מרכזים מחדש ע"י מציאת מרכז האשכול

4. עצירה - חזור על צעדים 2-3 עד שאין יותר

עדכונים



# אלגוריתם K-means - דוגמה

- נתון: אוסף ווקטורים והפרמטרים:  $K$  ופונקציית המרחק

- מצא חלוקה אופטימאלית שמחלקת ל- $K$  אשכולות

- נתון  $k=2$  (2 clusters צריך למצוא 2 prototypes)

- אלגוריתם:

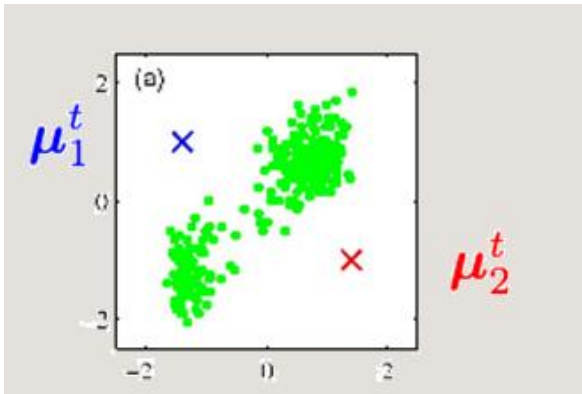
1. אתחול - "נחש"  $K$  מרכזים

2. שייך כל ווקטור ל"מרכז" הקרוב אליו

3. חשב מרכזים מחדש ע"י מציאת מרכז האשכול

4. עצירה - חזור על צעדים 2-3 עד שאין יותר

עדכונים





# אלגוריתם K-means - דוגמה

- נתון: אוסף ווקטורים והפרמטרים:  $K$  ופונקציית המרחק

- מצא חלוקה אופטימאלית שמחלקת ל- $K$  אשכולות

- נתון  $k=2$  (2 clusters צריך למצוא 2 prototypes)

- אלגוריתם:

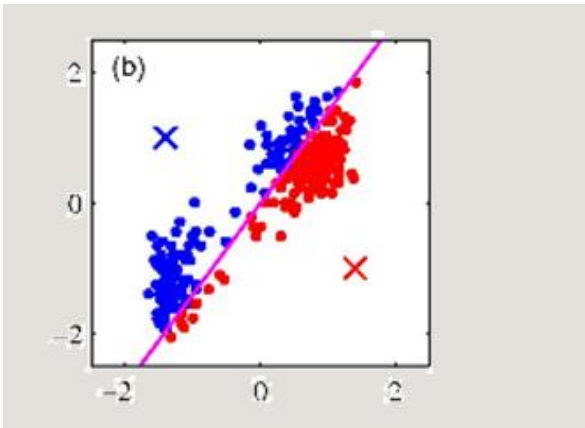
1. אתחול - "נחש"  $K$  מרכזים

2. שייך כל ווקטור ל"מרכז" הקרוב אליו

3. חשב מרכזים מחדש ע"י מציאת מרכז האשכול

4. עצירה - חזור על צעדים 2-3 עד שאין יותר

עדכונים



# אלגוריתם K-means - דוגמה

- נתון: אוסף ווקטורים והפרמטרים:  $K$  ופונקציית המרחק

- מצא חלוקה אופטימאלית שמחלקת ל- $K$  אשכולות

- נתון  $k=2$  (2 clusters צריך למצוא 2 prototypes)

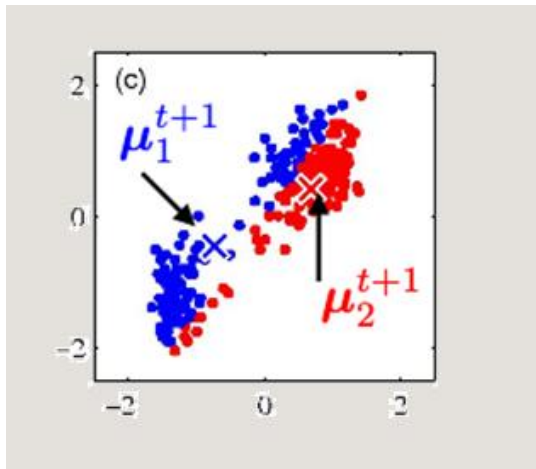
- אלגוריתם:

1. אתחול - "נחש"  $K$  מרכזים

2. שייך כל ווקטור ל"מרכז" הקרוב אליו

3. חשב מרכזים מחדש ע"י מציאת מרכז האשכול

4. עזירה - חזור על צעדים 2-3 עד שאין יותר עדכונים



# אלגוריתם K-means - דוגמה

- נתון: אוסף ווקטורים והפרמטרים:  $K$  ופונקציית המרחק

- מצא חלוקה אופטימאלית שמחלקת ל- $K$  אשכולות

- נתון  $k=2$  (2 clusters צריך למצוא 2 prototypes)

- אלגוריתם:

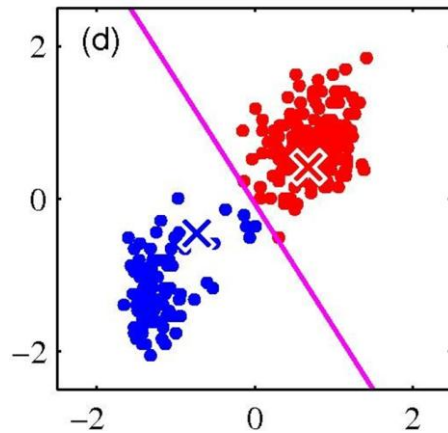
1. אתחול - "נחש"  $K$  מרכזים

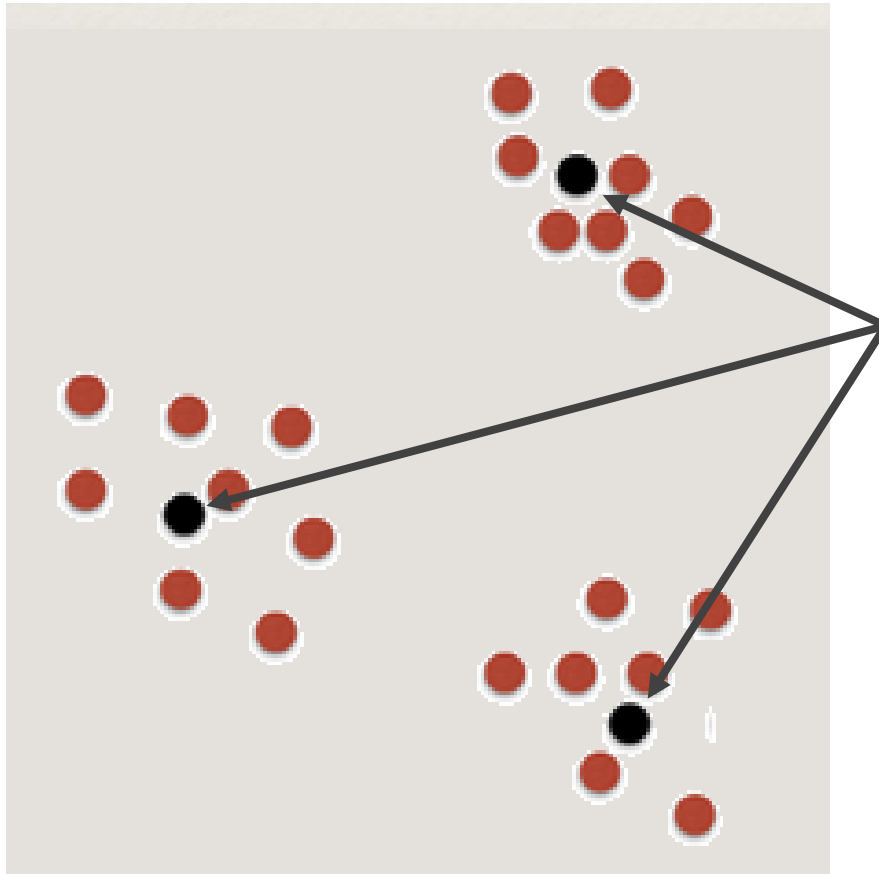
2. שייך כל ווקטור ל"מרכז" הקרוב אליו

3. חשב מרכזים מחדש ע"י מציאת מרכז האשכול

4. עצירה - חזור על צעדים 2-3 עד שאין יותר

עדכונים





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

# מה הם prototypes-ה המשמשים כמרכזים?

## – K-means

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

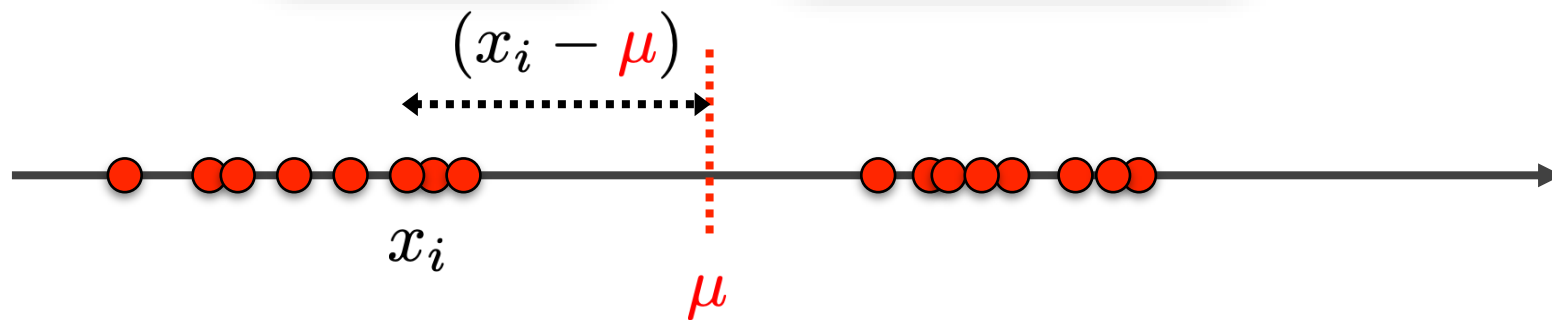
נתון מדגם של  $n$  דוגמאות:

ממוצע המדגם

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

שונות (מדד לפיזור)

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

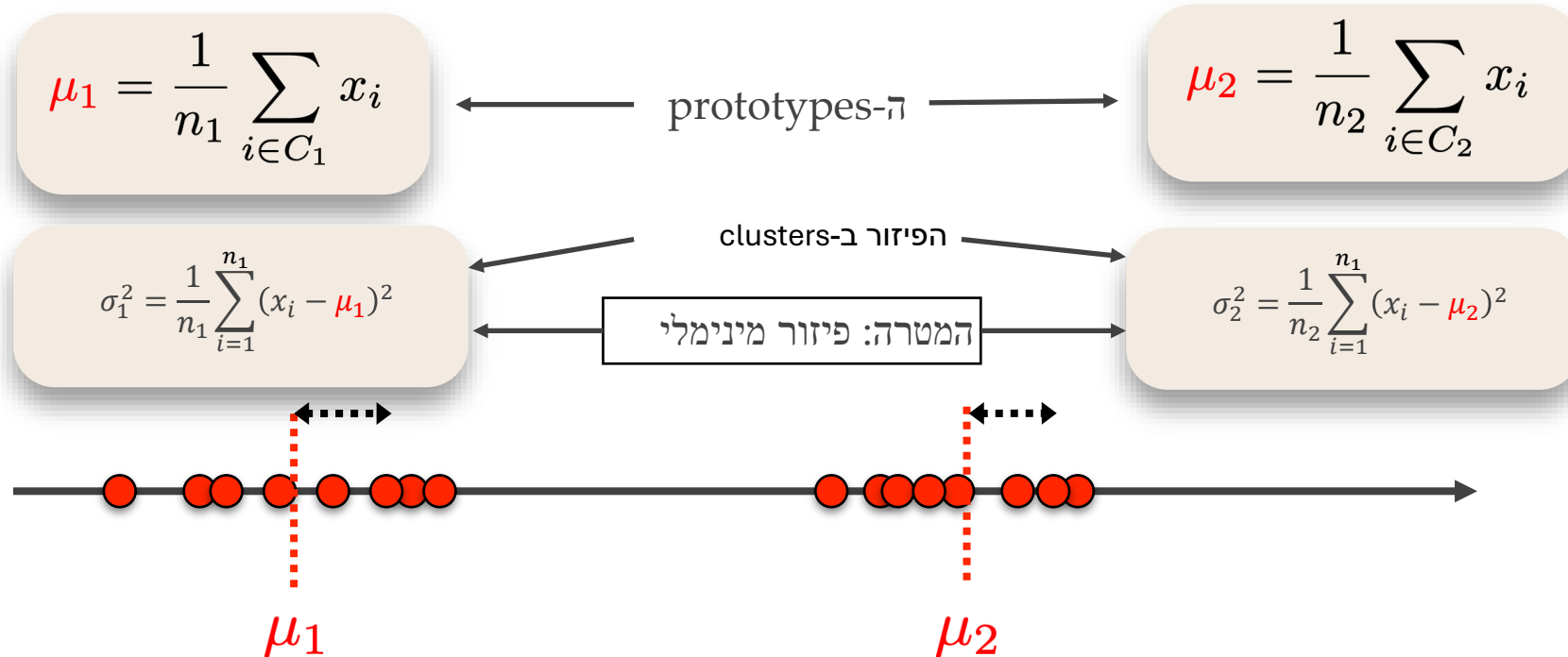


# - K-means

## מה הם ה-prototypes כמרכזים?

מה קורה אם ניקח 2 clusters  $C_1, C_2$ ?

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



# שאלת סקר

1. איך נחשב את ה-prototype לכל cluster ב-k-means ואיך נדע שה-cluster איכותי ביחס לווקטורים השייכים אליו?

## תשובות אפשריות:

- א. מחשבים prototype ע"י שונות, ונדע שה-cluster איכותי ע"י ממוצע ווקטורי ושאיפה לממוצע מינימלי
- ב. מחשבים prototype ע"י ממוצע ווקטורי, ונדע שה-cluster איכותי ע"י חישוב שונות ושאיפה לשונות מינימלית

# שאלת סקר

1. איך נחשב את ה-prototype לכל cluster ב-k-means ואיך נדע שה-cluster איכותי ביחס לווקטורים השייכים אליו?

תשובות אפשריות:

- א. מחשבים prototype ע"י שונות, ונדע שה-cluster איכותי ע"י ממוצע ווקטורי ושאיפה לממוצע מינימלי
- ב. מחשבים prototype ע"י ממוצע ווקטורי, ונדע שה-cluster איכותי ע"י חישוב שונות ושאיפה לשונות מינימלית



## K-Means – פונקציית המטרה (objective function)



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

עבור כל cluster, קיים וקטור מייצג - prototype :  
 $\vec{\mu}_1, \dots, \vec{\mu}_k$

$$J = \sum_{j=1}^k \sum_{\hat{y}_i=j} \sigma_j^2 : \text{פונקציית המחיר – הפיזור המשותף:}$$
$$= \sum_{j=1}^k \sum_{\hat{y}_i=j} ||x_i - \mu_j||^2$$

• בעיית אופטימיזציה:  $\min_{\{\hat{y}_i, \mu_j\}} [\sum_{j=1}^k \sum_{\hat{y}_i=j} ||x_i - \mu_j||^2]$

$$\min[J]=$$

• בעיה NP-hard ולכן נפתור בשלבים:

- מינימיזציה של  $\{\hat{y}_i\}$  - 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

$$= \sum_{j=1}^k \sum_{\hat{y}_i=j} ||x_i - \mu_j||^2$$

$r_{i,j}$  – נגדיר פונקציית עזר

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

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

כעת נגדיר את פונקציית 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

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

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

בעיית מינימיזציה א':

בהינתן המרכזים  $\{\mu_j\}$

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

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

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

## K-Means – פונקציית המטרה (objective function) – חלק 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  $\{r_{i,j}\}$

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

## 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** (WSSE),

$$\begin{aligned} \text{WSSE} &= \sum_{j=1}^k \sum_{\hat{y}_i=j} d(x_i, \mu_j)^2 \\ &= \sum_{j=1}^k \sum_{i=1}^n r_{i,j} \cdot \|x_i - \mu_j\|^2 \end{aligned}$$

distance between a  
vector to its centroid

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

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

## K-means – שלב 1 - אתחול ה-centroids – שיפור 1

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 פוטנציאליים.

- **Forgy method (Hamerly & Elkan, 2002) - בחירה אקראית של נקודות מתוך ה-dataset (ולא מתוך כל ערך אפשרי).**

## K-means – תרגיל 7 – אותם נתונים עם $K=3$ דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

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



# K-means – תרגיל 7 – אותם נתונים עם $K=3$ - פתרון דוגמא עם מאפיין 1 (1D) – שימוש בפונ' מרחק אוקלידית



K-means שלב 1 (Forgy method) - נבחר 2 "מרכזים" ראשוניים 1,2,3

• שוב בחירה לא מוצלחת



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



איטרציה 1: K-means שלב 3 - נעדכן "מרכזים":

❖ מרכז סגול:  $(3+4+13+14+18+19+20)/7=13$

❖ מרכז כתום:  $1/1=1$

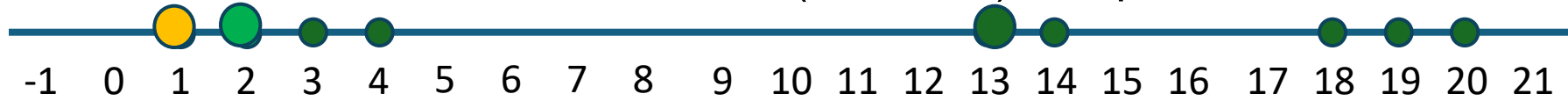
❖ מרכז ירוק:  $2/1=2$





# K-means – תרגיל 7 – אותם נתונים עם $K=3$ 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 סגול:  $(13+14+18+19+20)/5=16.8$

❖ מרכז ירוק:  $(2+3+4)/3=3$

❖ מרכז כתום:  $1/1=1$



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

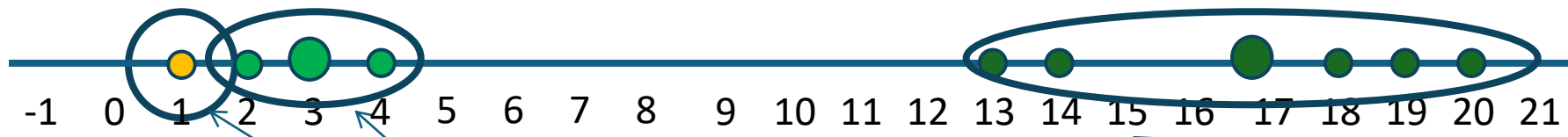


# K-means – תרגיל 7 – אותם נתונים עם $K=3$ דוגמא עם מאפיין 1 (1D) – שימוש בפו' מרחק אוקלידית

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



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



חשבו את ה- WSSE שנוצר

**במצגת k-means - תרגול נוסף:**

- תרגול K-means, עבור  $K=2$

- תרגול K-means, עבור 2

מאפיינים

## K-means – תרגיל 7 – אותם נתונים עם K=3 נחשב את הסטיה שנוצרה (ה-WSSE)

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

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

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

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

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

## K-means – תרגיל 7 – אותם נתונים עם K=3 נחשב את הסטיה שנוצרה (ה-WSSE)

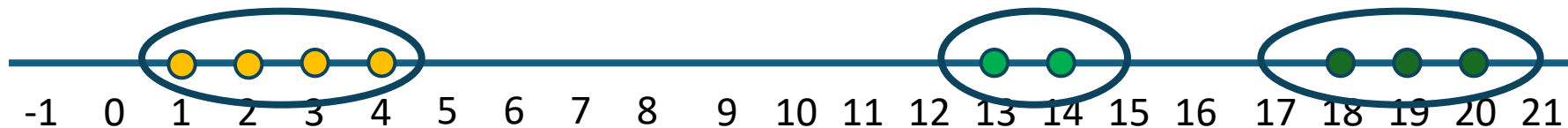
$$cluster1: (1-1)^2 = 0$$

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

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

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

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



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

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

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

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

# From Relevance Feedback to Query Expansion - Overview

- ① Motivation
- ② Relevance feedback: Basics
- ③ Relevance feedback: Details
- ④ Query expansion

# How can we improve recall in search?

- Main topic today: two ways of improving recall: relevance feedback and query expansion
- As an example consider query  $q$ : [aircraft] . . .
- . . . and document  $d$  containing “plane”, but not containing “aircraft”
- A simple IR system will not return  $d$  for  $q$ .
- Even if  $d$  is the most relevant document for  $q$ !
- We want to change this:
- Return relevant documents even if there is no term match with the (original) query

# Recall

- Loose definition of recall in this lecture: “increasing the number of relevant documents returned to user”

# Options for improving recall

- Local: Do a “local”, on-demand analysis for a user query
  - Main local method: **relevance feedback**
  - Part 1
- Global: Do a global analysis once (e.g., of collection) to produce **thesaurus**
  - Use thesaurus for **query expansion**
  - Part 2



# From Relevance Feedback to Query Expansion - Overview

- ① Motivation
- ② Relevance feedback: Basics
- ③ Relevance feedback: Details
- ④ Query expansion

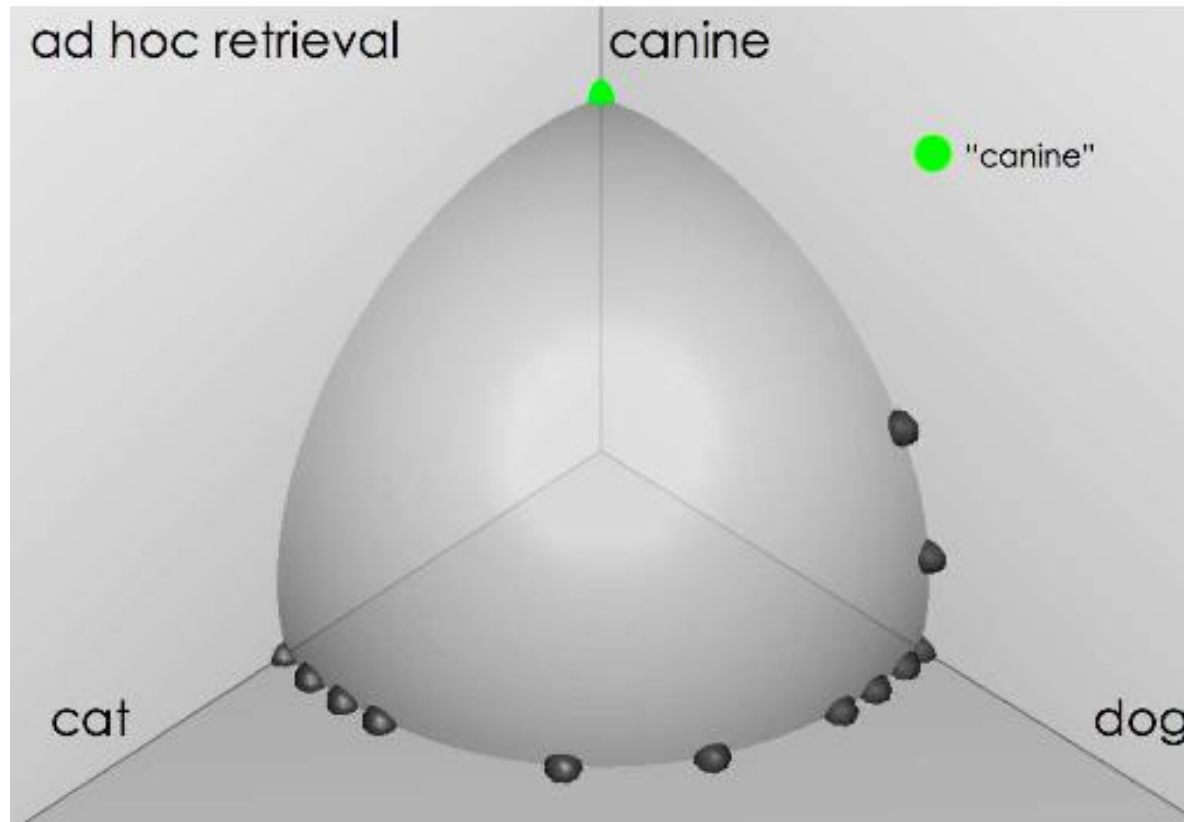
# Relevance feedback: Basic idea

- Relevance feedback: user feedback on relevance of docs in initial set of results
  - The user issues a (short, simple) query.
  - The search engine returns a set of documents.
  - User marks some docs as relevant, some as nonrelevant.
  - **Search engine computes a new representation of the information need. Hope: better than the initial query.**
  - Search engine runs new query and returns new results.
  - New results have (hopefully) better recall.
- Idea: it may be difficult to formulate a good query when you don't know the collection well, so iterate

# Relevance feedback

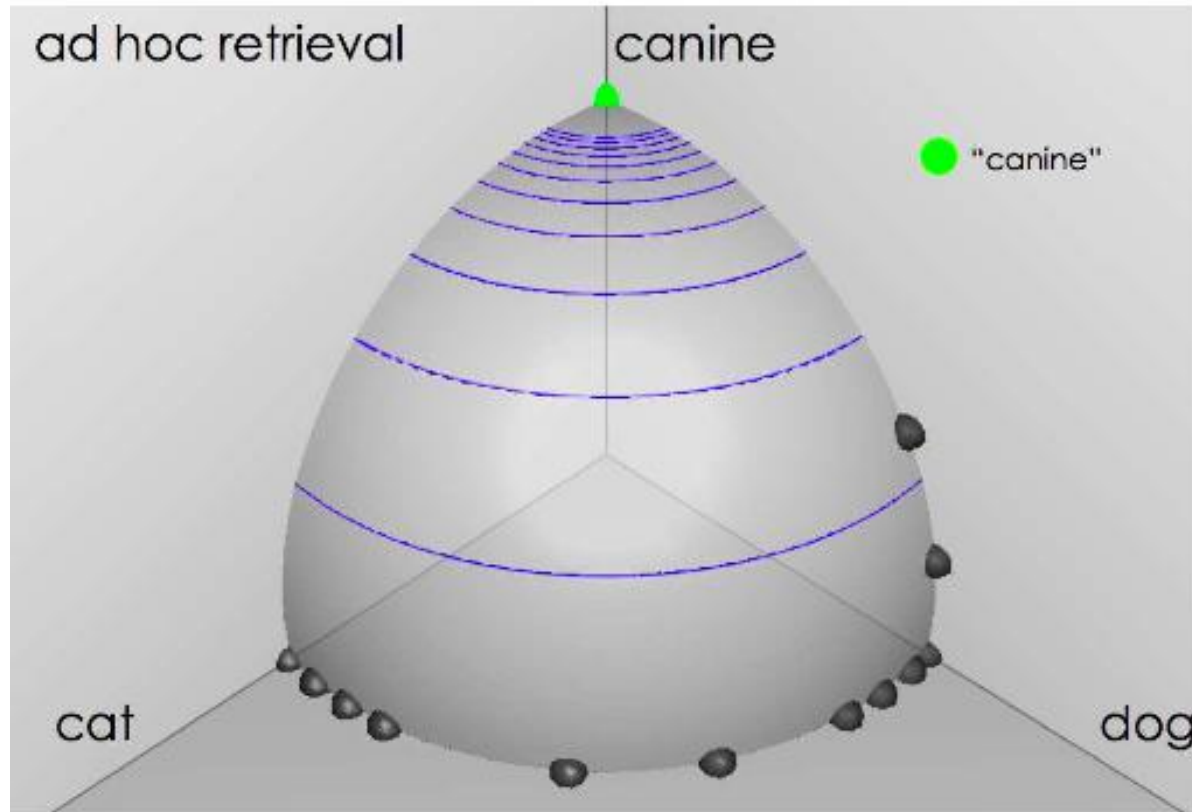
- We can iterate this: several rounds of relevance feedback.
- We will use the term **ad hoc retrieval** to refer to regular retrieval without relevance feedback.
- We will now look at three different examples of relevance feedback that highlight different aspects of the process.

# Vector space example: query “canine” (1)



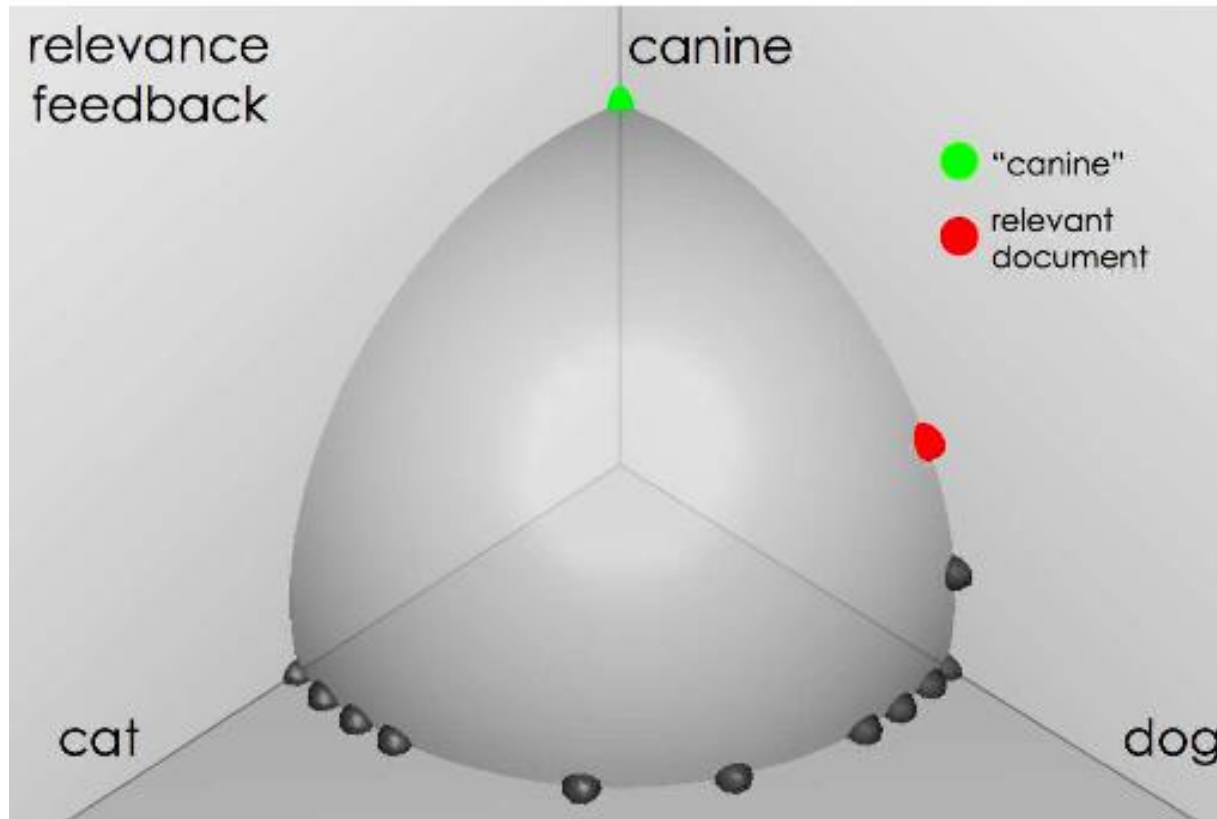
Source:  
Fernando Díaz

# Similarity of docs to query “canine”



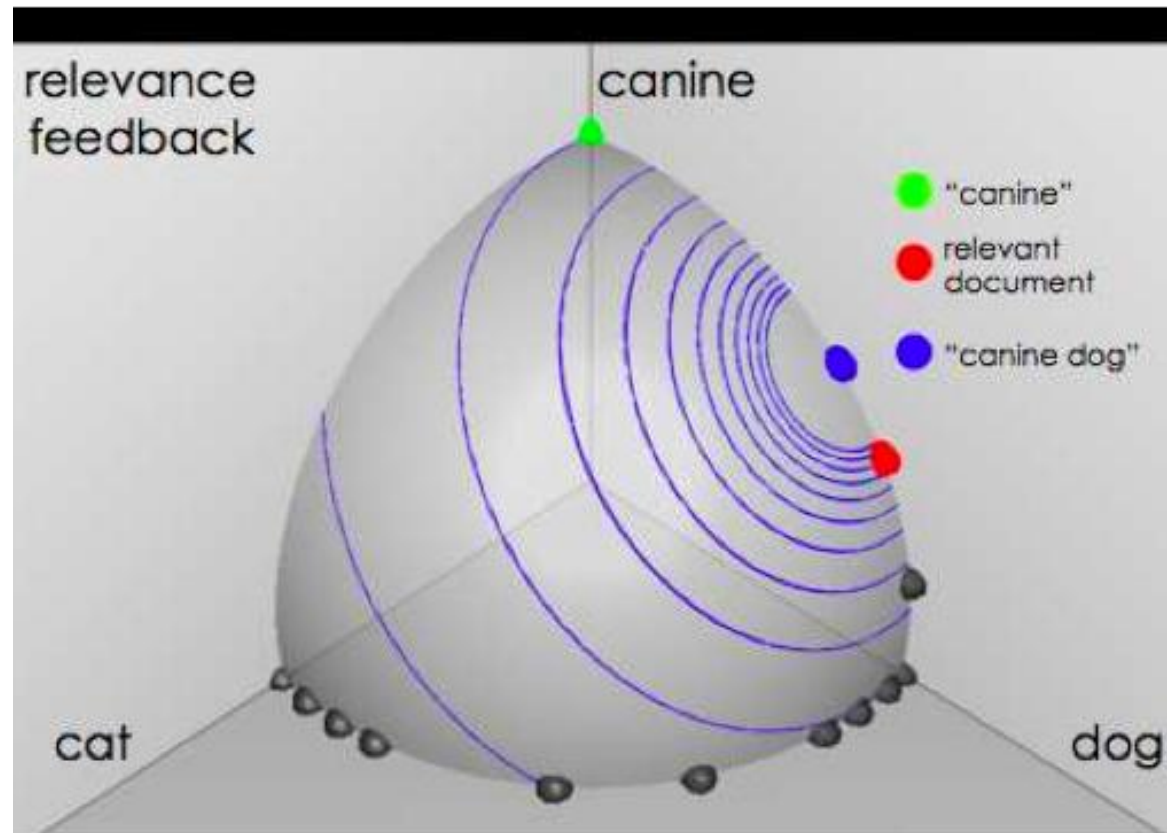
Source:  
Fernando Díaz

# User feedback: Select relevant documents



Source:  
Fernando Díaz

# Results after relevance feedback



Source:

Fernando Díaz

# Example 3: A real (non-image) example

Initial query:

[new space satellite applications] Results for initial query: ( $r$  = rank)

	$r$		
Plan Launches of Feat: Staying Satellites for Satellites Telesat	+	1	0.539 NASA Hasn't Scrapped Imaging Spectrometer
	+	2	0.533 NASA Scratches Environment Gear From Satellite
		3	0.528 Science Panel Backs NASA Satellite Plan, But Urges Smaller Probes
		4	0.526 A NASA Satellite Project Accomplishes Incredible Within Budget
		5	0.525 Scientist Who Exposed Global Warming Proposes Climate Research
		6	0.524 Report Provides Support for the Critics Of Using Big to Study Climate
		7	0.516 Arianespace Receives Satellite Launch Pact From



# Expanded query after relevance feedback

2.074	new	15.106	space
30.816	satellite	5.660	application
5.991	nasa	5.196	eos
4.196	launch	3.972	aster
3.516	instrument	3.446	arianespace
3.004	bundespost	2.806	ss
2.790	rocket	2.053	scientist
2.003	broadcast	1.172	earth
0.836	oil	0.646	measure

Compare to original query: [new space satellite applications]

# Results for expanded query

	<i>r</i>		
*	1	0.513	NASA Scratches Environment Gear From Satellite Plan
*	2	0.500	NASA Hasn't Scrapped Imaging Spectrometer
	3	0.493	When the Pentagon Launches a Secret Satellite, Space Sleuths Do Some Spy Work of Their Own
	4	0.493	NASA Uses 'Warm' Superconductors For Fast Circuit
*	5	0.492	Telecommunications Tale of Two Companies
	6	0.491	Soviets May Adapt Parts of SS-20 Missile For Commercial Use
	7	0.490	Gaping Gap: Pentagon Lags in Race To Match the Soviets In Rocket Launchers
	8	0.490	Rescue of Satellite By Space Agency To Cost \$90 Million

# From Relevance Feedback to Query Expansion - Overview

- 1 Motivation
- 2 Relevance feedback: Basics
- 3 Relevance feedback: Details
- 4 Query expansion

# Key concept for relevance feedback: Centroid

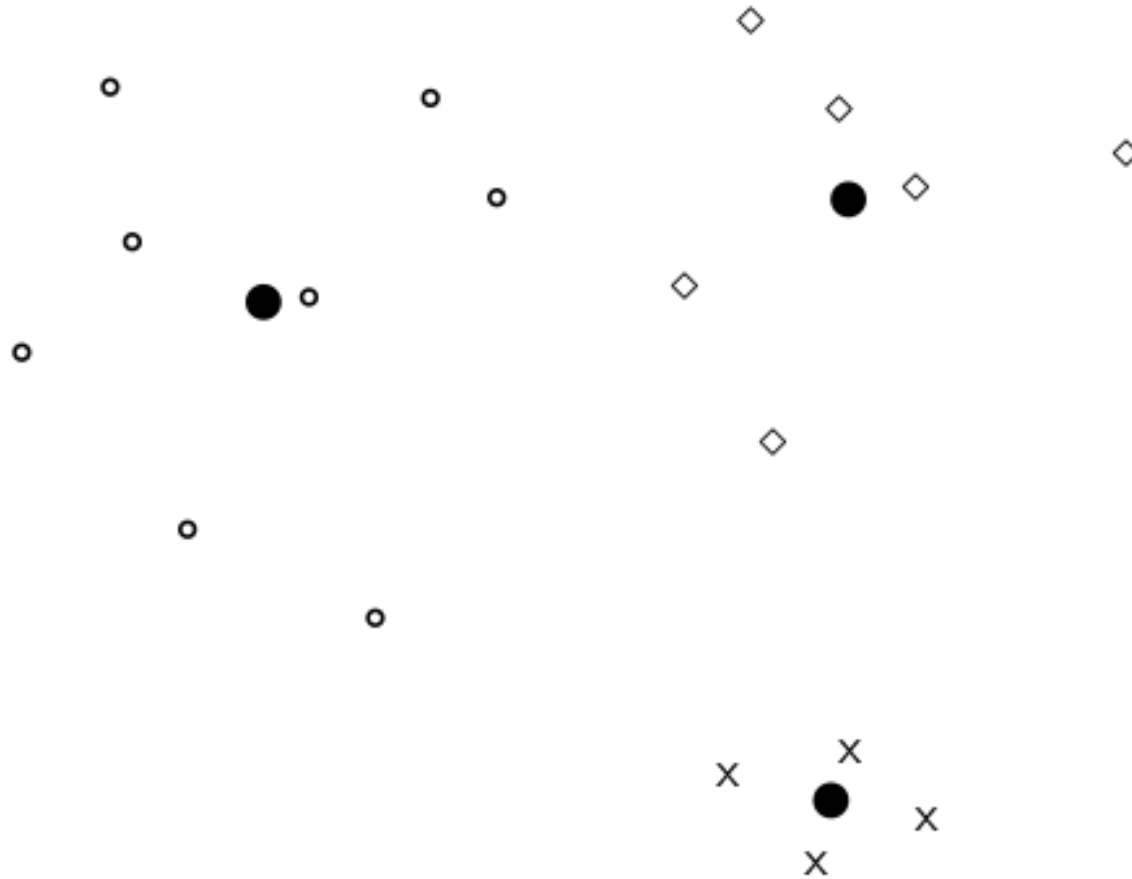
- The centroid is the center of mass of a set of points.
- Recall that we represent documents as points in a high-dimensional space.
- Thus: we can compute centroids of documents.

■ Definition:

$$\vec{\mu}(D) = \frac{1}{|D|} \sum_{d \in D} \vec{v}(d)$$

where  $D$  is a set of documents and  $\vec{v}(d) = \vec{d}$  is the vector we use to represent document  $d$ .

# Centroid: Example



# Rocchio' algorithm

- The Rocchio' algorithm implements relevance feedback in the vector space model.

- Rocchio' chooses the query  $\vec{q}_{opt}$  that maximizes

$$\vec{q}_{opt} = \arg \max_{\vec{q}} [\text{sim}(\vec{q}, \mu(D_r)) - \text{sim}(\vec{q}, \mu(D_{nr}))]$$

$D_r$  : set of relevant docs;  $D_{nr}$  : set of nonrelevant docs

- Intent:  $\vec{q}_{opt}$  is the vector that separates relevant and nonrelevant docs maximally.
- Making some additional assumptions, we can rewrite  $\vec{q}_{opt}$ :

$$\vec{q}_{opt} = \mu(D_r) + [\mu(D_r) - \mu(D_{nr})]$$

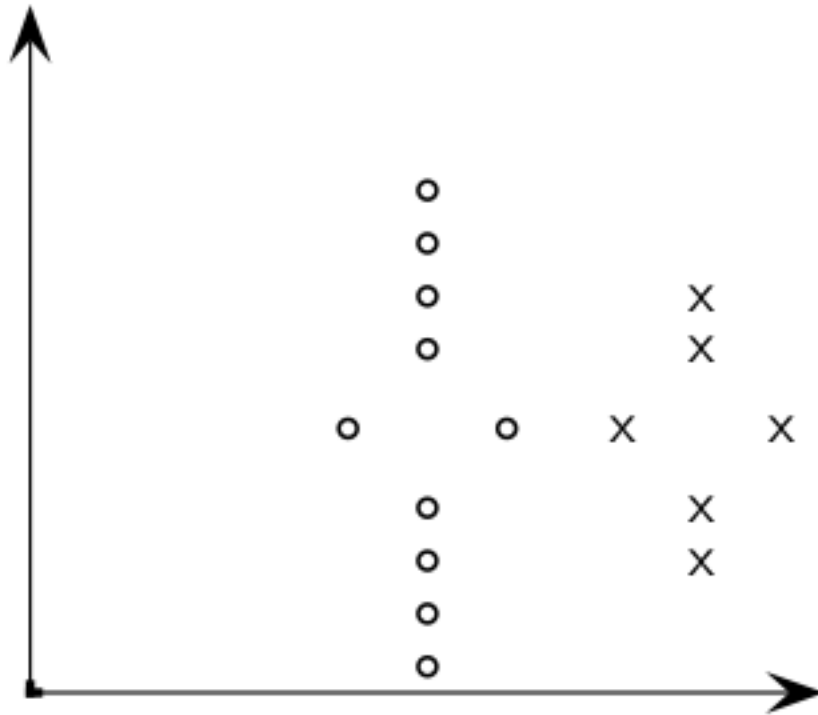
# Rocchio' algorithm

- The optimal query vector is:

$$\begin{aligned}\vec{q}_{opt} &= \mu(D_r) + [\mu(D_r) - \mu(D_{nr})] \\ &= \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j + \left[ \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j \right]\end{aligned}$$

- We move the centroid of the relevant documents by the difference between the two centroids.

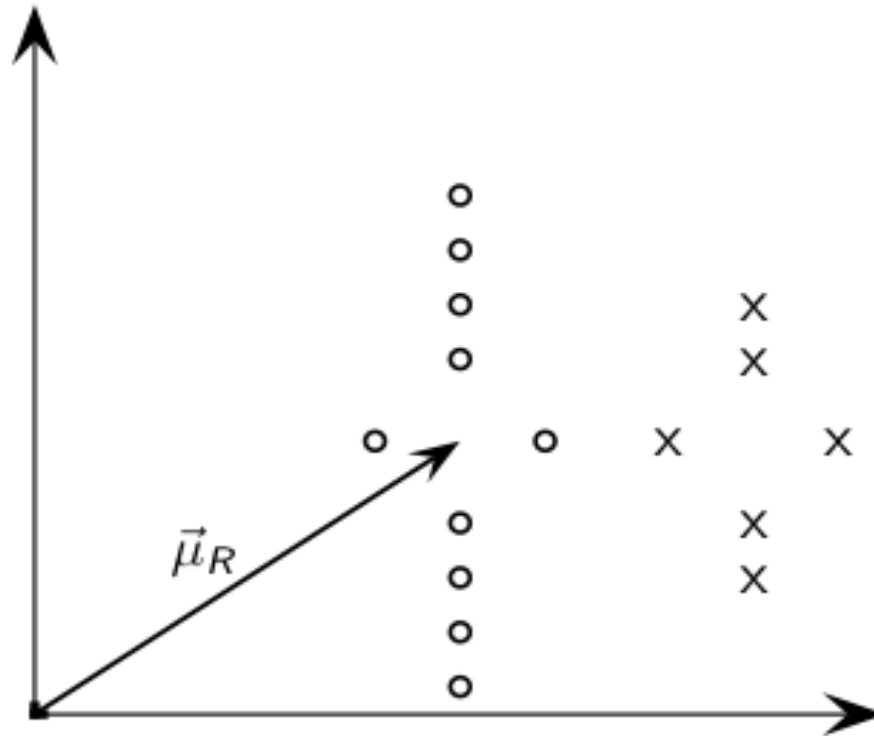
# Exercise: Compute Rocchio' vector



circles: relevant documents, Xs: nonrelevant documents

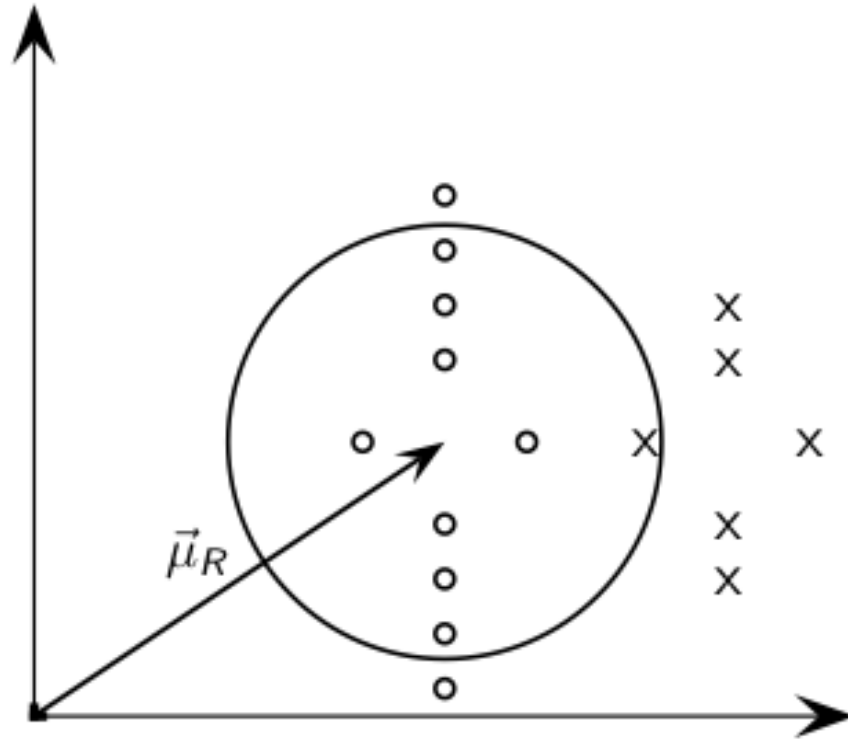


# Rocchio' illustrated



$\vec{\mu}_R$  : centroid of relevant documents

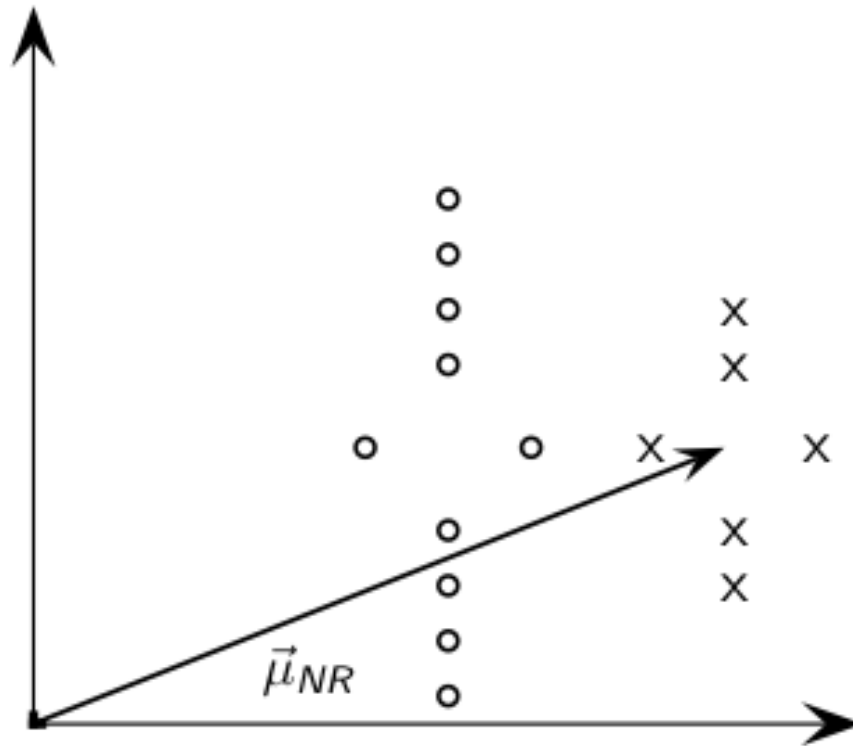
# Rocchio' illustrated



$\vec{\mu}_R$

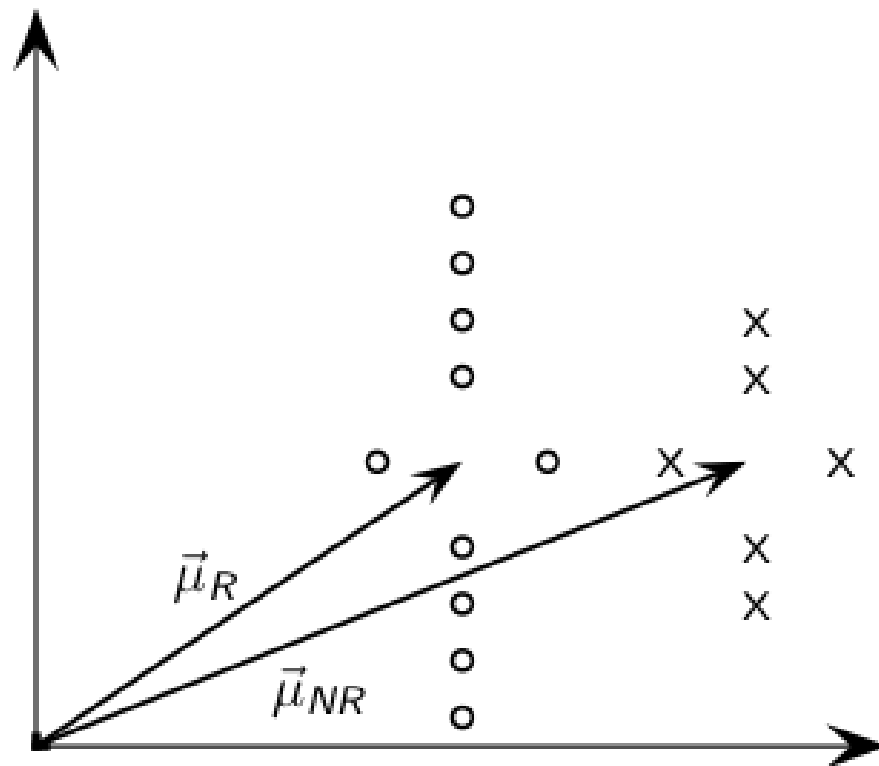
does not separate relevant / nonrelevant.

# Rocchio' illustrated

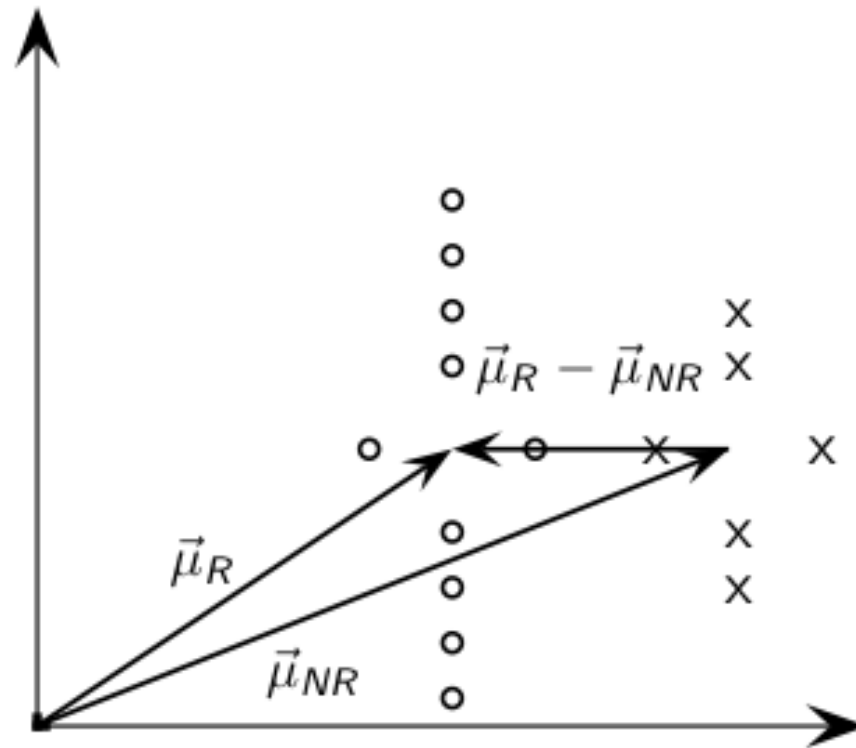


$\vec{\mu}_{NR}$ : centroid of nonrelevant documents.

# Rocchio' illustrated

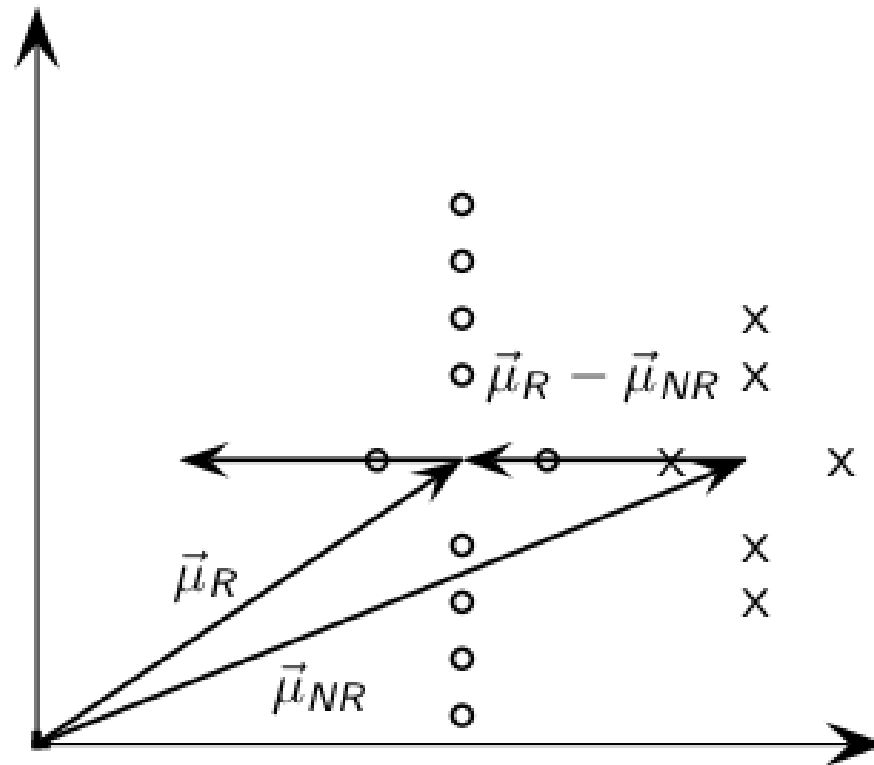


# Rocchio' illustrated



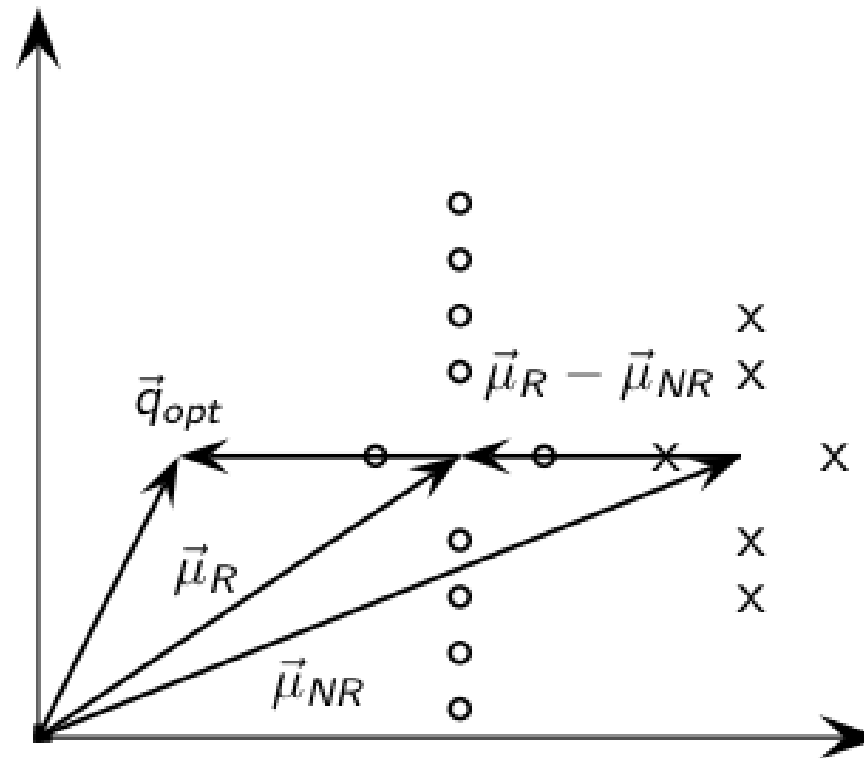
$\vec{\mu}_R - \vec{\mu}_{NR}$ : difference vector

# Rocchio' illustrated



Add difference vector to ...  $\vec{\mu}_R$

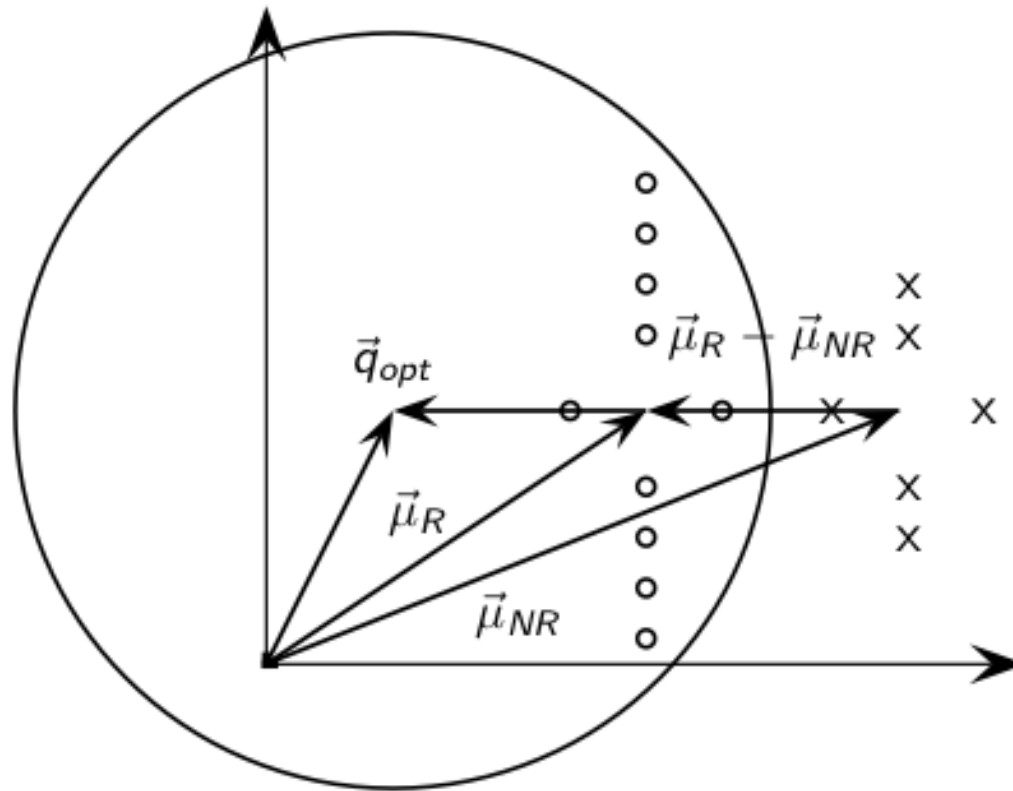
# Rocchio' illustrated



... to get

$\vec{q}_{opt}$

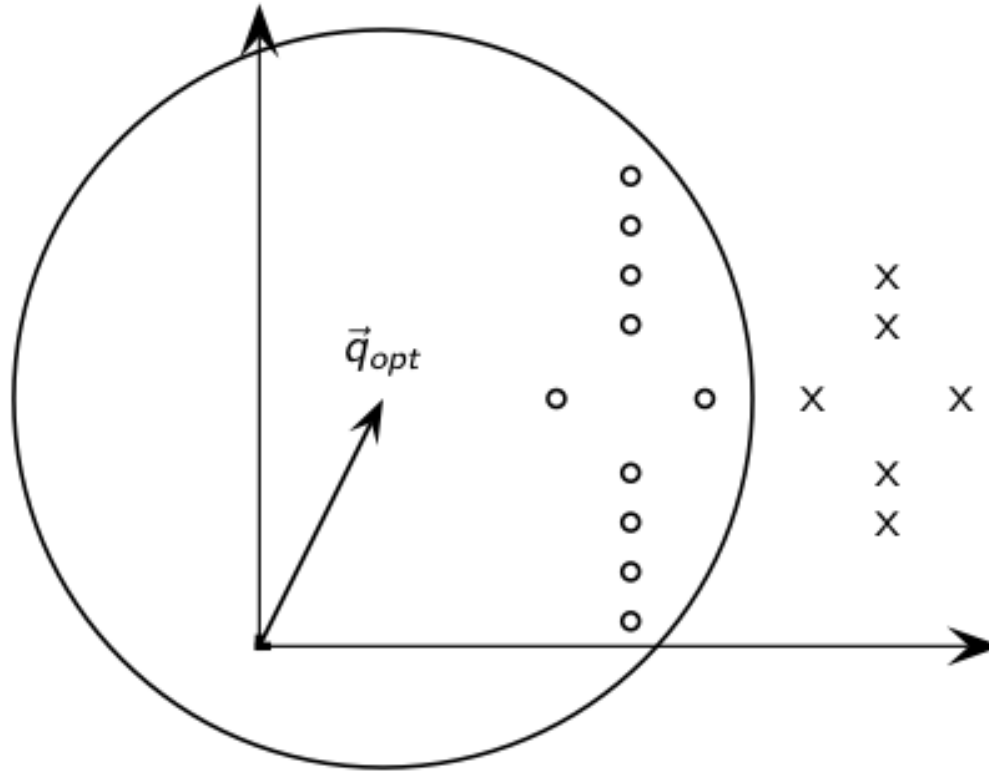
# Rocchio' illustrated



$\vec{q}_{opt}$  separates relevant / nonrelevant perfectly.



# Rocchio' illustrated



$\vec{q}_{opt}$  separates relevant / nonrelevant perfectly.

# Terminology

- We use the name Rocchio' for the theoretically better motivated original version of Rocchio.
- The implementation that is actually used in most cases is the SMART implementation – we use the name Rocchio (without prime) for that.

# Rocchio 1971 algorithm (SMART)

Used in practice:

$$\begin{aligned}\vec{q}_m &= \alpha \vec{q}_0 + \beta \mu(D_r) - \gamma \mu(D_{nr}) \\ &= \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j\end{aligned}$$

$q_m$ : modified query vector;  $q_0$ : original query vector;  $D_r$  and  $D_{nr}$ : sets of known relevant and nonrelevant documents respectively;  $\alpha$ ,  $\beta$ , and  $\gamma$ : weights

- New query moves towards relevant documents and away from nonrelevant documents.
- Tradeoff  $\alpha$  vs.  $\beta/\gamma$ : If we have a lot of judged documents, we want a higher  $\beta/\gamma$ .
- Set negative term weights to 0.
- “Negative weight” for a term doesn’t make sense in the vector space model.

# Positive vs. negative relevance feedback

- Positive feedback is more valuable than negative feedback.
- For example, set  $\beta = 0.75$ ,  $\gamma = 0.25$  to give higher weight to positive feedback.
- Many systems only allow positive feedback.

# Relevance feedback: Assumptions

- When can relevance feedback enhance recall?
- Assumption A1: The user knows the terms in the collection well enough for an initial query.
- Assumption A2: Relevant documents contain similar terms (so I can “hop” from one relevant document to a different one when giving relevance feedback).

# Violation of A1

- Assumption A1: The user knows the terms in the collection well enough for an initial query.
- Violation: Mismatch of searcher's vocabulary and collection vocabulary
- Example: cosmonaut / astronaut

# Violation of A2

- Assumption A2: Relevant documents are similar.
- Example for violation: [contradictory government policies]
- Several unrelated “prototypes”
  - Subsidies for tobacco farmers vs. anti-smoking campaigns
  - Aid for developing countries vs. high tariffs on imports from developing countries
- Relevance feedback on tobacco docs will not help with finding docs on developing countries.

# Relevance feedback: Evaluation

- Pick one of the evaluation measures from last lecture, e.g., precision in top 10:  $P@10$
- Compute  $P@10$  for original query  $q_0$
- Compute  $P@10$  for modified relevance feedback query  $q_1$
- In most cases:  $q_1$  is spectacularly better than  $q_0$ !
- Is this a fair evaluation?



# Relevance feedback: Problems

- Relevance feedback is expensive.
  - Relevance feedback creates long modified queries.
  - Long queries are expensive to process.
- Users are reluctant to provide explicit feedback.
- It's often hard to understand why a particular document was retrieved after applying relevance feedback.
- The search engine Excite had full relevance feedback at one point, but abandoned it later.

# Pseudo-relevance feedback

- Pseudo-relevance feedback automates the “manual” part of true relevance feedback.
- Pseudo-relevance algorithm:
  - Retrieve a ranked list of hits for the user’s query
  - Assume that the top  $k$  documents are relevant.
  - Do relevance feedback (e.g., Rocchio)
- Works very well on average
- But can go horribly wrong for some queries.
- Several iterations can cause *query drift*.

# From Relevance Feedback to Query Expansion - Overview

- 1 Motivation
- 2 Relevance feedback: Basics
- 3 Relevance feedback: Details
- 4 Query expansion

# Query expansion

- Query expansion is another method for **increasing recall**.
- We use “global query expansion” to refer to “global methods for query reformulation”.
- In global query expansion, the query is modified based on some global resource, i.e. a resource that is not query-dependent.
- Main information we use: (near-)synonymy
- A publication or database that collects (near-)synonyms is called a **thesaurus**.
- We will look at two types of thesauri: manually created and automatically created.

# Query expansion: Example

The screenshot shows a Yahoo! Search results page for the query "palm". At the top, the "YAHOO! SEARCH" logo is visible, followed by navigation links for Web, Images, Video, Audio, Directory, Local, News, Shopping, and More. The search bar contains the text "palm" and a "Search" button. Below the search bar, there are links for Answers, My Web, Search Services, Advanced Search, and Preferences. The "Search Results" section indicates that 1 - 10 of about 160,000,000 results were found for "palm" in 0.07 seconds. A "Also try:" section suggests related queries: palm springs, palm pilot, palm trees, palm reading, and More... The main results are divided into two columns. The left column features two sponsored results: "Official Palm Store" from store.palm.com, offering free shipping on handhelds, and "Palms Hotel - Best Rate Guarantee" from www.vegas.com, advertising the best rate guarantee at the Vegas travel site. Below these are links for "Palm Pilots" and "Palm Downloads" with a Yahoo! Shortcut and About link. The first organic result is "Palm, Inc.", described as the maker of handheld PDA devices, with category "B2B > Personal Digital Assistants (PDAs)", website www.palm.com, and 20k pages. The right column is titled "SPONSOR RESULTS" and contains three entries: "Palm Memory" from www.memorygiant.com, "The Palms, Turks and Caicos Islands" from www.worldwidereservationsystems.c, and "The Palms Casino Resort, Las Vegas" from lasvegas.hotelscorp.com, which offers a low price guarantee.

**YAHOO! SEARCH**

Web | Images | Video | Audio | Directory | Local | News | Shopping | More »

palm

Answers | My Web | Search Services | Advanced Search | Preferences

**Search Results** 1 - 10 of about 160,000,000 for palm - 0.07 sec. (About this page)

Also try: [palm springs](#), [palm pilot](#), [palm trees](#), [palm reading](#) [More...](#)

**SPONSOR RESULTS**

- [Official Palm Store](#)  
[store.palm.com](#) Free shipping on all handhelds and more at the official **Palm** store.
- [Palms Hotel - Best Rate Guarantee](#)  
[www.vegas.com](#) Book the **Palms** Hotel Casino with our best rate guarantee at VEGAS.com, the official Vegas travel site.

**Y** [Palm Pilots](#) - [Palm Downloads](#)  
[Yahoo! Shortcut](#) - [About](#)

1. [Palm, Inc.](#)   
Maker of handheld PDA devices that allow mobile users to manage schedules, contacts, and other personal and business information.  
Category: [B2B > Personal Digital Assistants \(PDAs\)](#)  
[www.palm.com](#) - 20k - [Cached](#) - [More from this site](#) - [Save](#)

**SPONSOR RESULTS**

[Palm Memory](#)  
Memory Giant is fast and easy.  
Guaranteed compatible memory.  
Great...  
[www.memorygiant.com](#)

[The Palms, Turks and Caicos Islands](#)  
Resort/Condo photos, rates, availability and reservations....  
[www.worldwidereservationsystems.c](#)

[The Palms Casino Resort, Las Vegas](#)  
Low price guarantee at the **Palms** Casino resort in Las Vegas. Book...  
[lasvegas.hotelscorp.com](#)

# Types of user feedback

- User gives feedback on documents.
  - More common in relevance feedback
- User gives feedback on words or phrases.
  - More common in query expansion

# Types of query expansion

- Manual thesaurus (maintained by editors, e.g., PubMed)
- Automatically derived thesaurus (e.g., based on co-occurrence statistics)
- Query-equivalence based on query log mining (common on the web as in the “palm” example)

# Thesaurus-based query expansion

- For each term  $t$  in the query, expand the query with words the thesaurus lists as semantically related with  $t$ .
- Example from earlier: HOSPITAL → MEDICAL
- Generally increases recall
- May significantly decrease precision, particularly with ambiguous terms
  - INTEREST RATE → INTEREST RATE FASCINATE
- Widely used in specialized search engines for science and engineering
- It's very expensive to create a manual thesaurus and to maintain it over time.
- A manual thesaurus has an effect roughly equivalent to annotation with a [controlled vocabulary](#).



# Example for manual thesaurus: PubMed

The screenshot displays the PubMed website interface. At the top, the NCBI logo is on the left, the PubMed logo is in the center, and the National Library of Medicine (NLM) logo is on the right. Below the logos, a navigation bar contains links to PubMed, Nucleotide, Protein, Genome, Structure, PopSet, and Taxonomy. The main search area features a search bar with the text 'cancer' and a dropdown menu set to 'PubMed'. To the right of the search bar are 'Go' and 'Clear' buttons. Below the search bar, a row of links includes 'Limits', 'Preview/Index', 'History', 'Clipboard', and 'Details'. On the left side, a vertical menu lists various links: 'About Entrez', 'Text Version', 'Entrez PubMed Overview', 'Help | FAQ', 'Tutorial', 'New/Noteworthy', 'E-Utilities', 'PubMed Services', 'Journals Database', 'MeSH Browser', 'Single Citation', and 'Matched'. The main content area shows the 'PubMed Query:' section with the query: `("neoplasms"[MeSH Terms] OR cancer[Text Word])`. At the bottom of the query area, there are 'Search' and 'URL' buttons.

NCBI

PubMed

National Library of Medicine NLM

PubMed Nucleotide Protein Genome Structure PopSet Taxonomy

Search PubMed for cancer Go Clear

Limits Preview/Index History Clipboard Details

About Entrez

Text Version

Entrez PubMed Overview Help | FAQ Tutorial New/Noteworthy E-Utilities

PubMed Services Journals Database MeSH Browser Single Citation Matched

PubMed Query:

`("neoplasms"[MeSH Terms] OR cancer[Text Word])`

Search URL

# Automatic thesaurus generation

- Attempt to generate a thesaurus automatically by analyzing the distribution of words in documents
- Fundamental notion: similarity between two words
- Definition 1: Two words are **similar if they co-occur with similar words**.
  - “car”  $\approx$  “motorcycle” because both occur with “road”, “gas” and “license”, so they must be similar.
- Definition 2: Two words are **similar if they occur in a given grammatical relation with the same words**.
  - You can harvest, peel, eat, prepare, etc. apples and pears, so apples and pears must be similar.
- Co-occurrence is more robust, grammatical relations are more accurate.

# Co-occurrence-based thesaurus: Examples

Word	Nearest neighbors
absolutely	absurd whatsoever totally exactly nothing
bottomed	dip copper drops topped slide trimmed
captivating	shimmer stunningly superbly plucky witty
doghouse	dog porch crawling beside downstairs
makeup	repellent lotion glossy sunscreen skin gel
mediating	reconciliation negotiate case conciliation
keeping	hoping bring wiping could some would
lithographs	drawings Picasso Dali sculptures Gauguin
pathogens	toxins bacteria organisms bacterial parasite
senses	grasp psyche truly clumsy naive innate

WordSpace demo on web

# Query expansion at search engines

- Main source of query expansion at search engines: query logs
- Example 1: After issuing the query [herbs], users frequently search for [herbal remedies].
  - → “herbal remedies” is potential expansion of “herb”.
- Example 2: Users searching for [flower pix] frequently click on the URL [photobucket.com/flower](http://photobucket.com/flower). Users searching for [flower clipart] frequently click on the [same URL](#).
  - → “flower clipart” and “flower pix” are potential expansions of each other.

# Take-away today

- **Interactive relevance feedback:** improve initial retrieval results by telling the IR system which docs are relevant / nonrelevant
- Best known relevance feedback method: Rocchio feedback
- **Query expansion:** improve retrieval results by adding synonyms / related terms to the query
  - **Sources for related terms:** Manual thesauri, automatic thesauri, query logs

# Resources

- Chapter 9 of IIR
- Resources at <http://ifnlp.org/ir>
  - Salton and Buckley 1990 (original relevance feedback paper)
  - Spink, Jansen, Ozmultu 2000: Relevance feedback at Excite
  - Schütze 1998: Automatic word sense discrimination (describes a simple method for automatic thesuarus generation)

Until the next time 😊

