DATA MINING
SIMILARITY & DISTANCE

Similarity and Distance

Similarity and Distance

- For many different problems we need to quantify how close two objects are.
- Examples:
  - Find similar movies/books/songs
  - Group together the customers of a site so that similar customers are shown the same ad.
  - Group together web documents so that you can separate the ones that talk about entertainment and the ones that talk about sports.
  - Find credit card transactions that are very different from previous transactions.
  - Find similar moving patterns
- To solve these problems we need a definition of similarity, or distance.
  - The definition depends on the type of data that we have

Similarity

- Numerical measure of how alike two data objects are.
  - A function that maps pairs of objects to real values
  - Higher when objects are more alike.
- Often falls in the range [0,1], sometimes in [-1,1]

- Desirable properties for similarity (but not must have)
  1. s(p, q) = 1 (or maximum similarity) only if p = q. (Identity 同一性)
  2. s(p, q) = s(q, p) for all p and q. (Symmetry 对称性)

Similarity between sets

- Consider the following documents

apple releases new iphone

apple releases new ipad

I will cook apple pie

- Which ones are more similar?

- How would you quantify their similarity?

Similarity: Intersection

- Number of words in common

apple releases new iphone

apple releases new ipad

I will cook apple pie

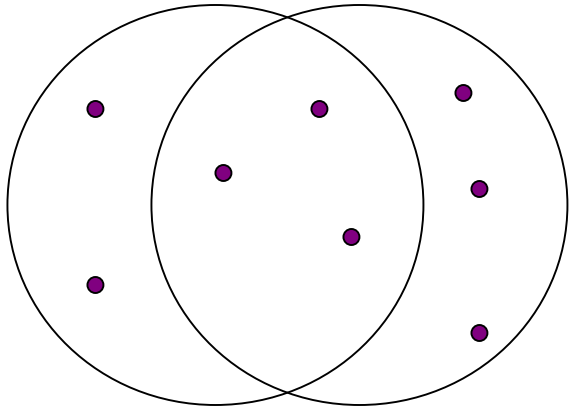
- Sim(D1,D2) = 3, Sim(D1,D3) = Sim(D2,D3) =1
- What about this document?

I hear that she **releases** a **new** album
I will buy the album and an **apple** pie tomorrow

- Sim(D1,D2) = Sim(D1,D4) = Sim(D2,D4) = 3; D1 and D2 should be more similar, D4 has 3 overlapped words by chance

Jaccard Similarity

- The Jaccard similarity (Jaccard coefficient) of two sets S_1, S_2 is the size of their intersection divided by the size of their union.
  - JSim (S_1, S_2) = $|S_1 \cap S_2|$ / $|S_1 \cup S_2|$.



3 in intersection.
8 in union.
Jaccard similarity = 3/8

- Extreme behavior:
  - JSim(X,Y) = 1, iff X = Y
  - JSim(X,Y) = 0 iff X,Y have no elements in common
- JSim is symmetric

Jaccard Similarity between sets

- The distance for the documents

apple releases new ipod

apple releases new ipad

I hear that she **releases** a **new** album
I will buy the album and an **apple** pie tomorrow

- JSim(D1,D2) = 3/5
- JSim(D1,D4) = JSim(D2,D4) = 3/18

Similarity between vectors

Documents (and sets in general) can also be represented as vectors

document	Apple	Microsoft	Biden	Election
D1	10	20	0	0
D2	30	60	0	0
D3	60	30	0	0
D4	0	0	10	20

How do we measure the similarity of two vectors?

- We could view them as sets of words. Jaccard Similarity will show that D4 is different from the rest
- But all pairs of the other three documents are equally similar (JSim = 1)

We want to capture how well the two vectors are aligned

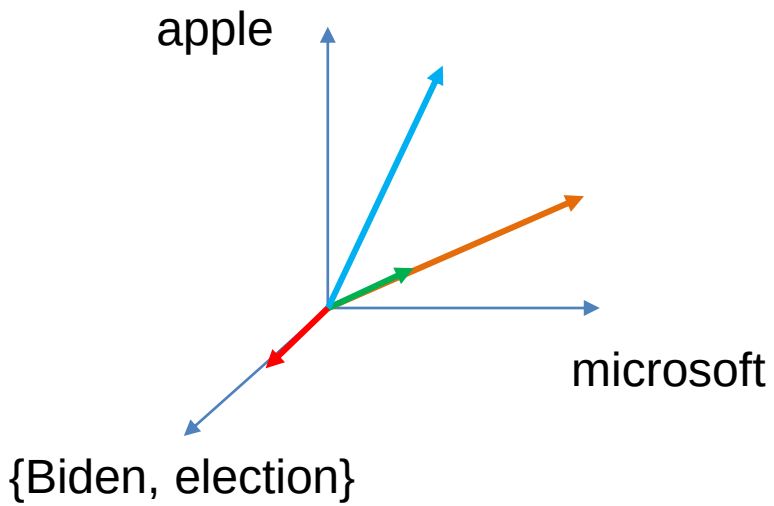
Example

document	Apple	Microsoft	Biden	Election
D1	10	20	0	0
D2	30	60	0	0
D3	60	30	0	0
D4	0	0	10	20

Documents D1, D2 are in the “same direction”

Document D3 is on the same plane as D1, D2

Document D4 is orthogonal to the rest



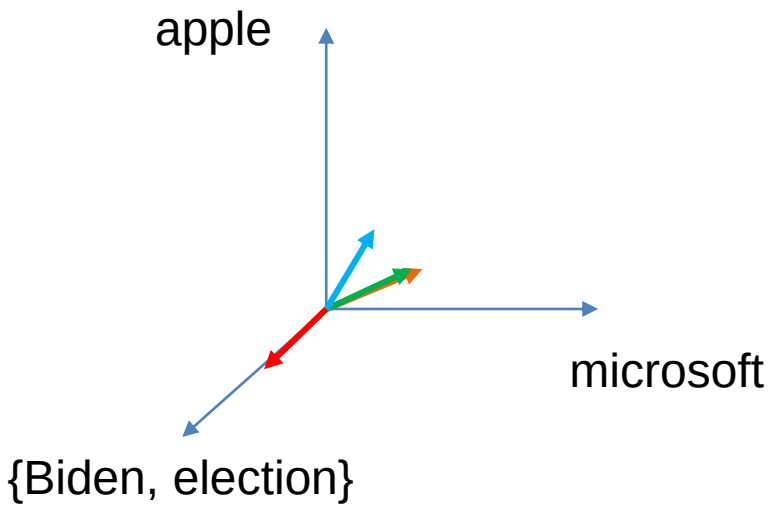
Example

document	Apple	Microsoft	Biden	Election
D1	10	20	0	0
D2	30	60	0	0
D3	60	30	0	0
D4	0	0	10	20

Documents D1, D2 are in the “same direction”

Document D3 is on the same plane as D1, D2

Document D4 is orthogonal to the rest



Cosine Similarity

- Sim(X,Y) = cos(X,Y)
  - The cosine of the angle between X and Y

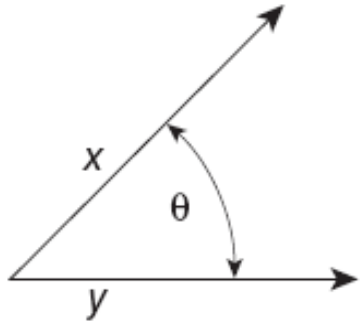


Figure 2.16. Geometric illustration of the cosine measure.

- If the vectors are aligned (correlated) angle is zero degrees and cos(X,Y)=1
- If the vectors are orthogonal (no common coordinates) angle is 90 degrees and cos(X,Y) = 0

- Cosine is commonly used for comparing documents, sometimes the vectors are normalized by the document length, or words are weighted by tf-idf.

Cosine Similarity - math

- If d_1 and d_2 are two vectors, then

  $$\cos(d_1, d_2) = (d_1 \bullet d_2) / \|d_1\| \|d_2\| ,$$

  where • indicates vector dot product and || *d* || is the length of vector *d*.

- Example:

  d_1 = **3 2 0 5 0 0 0 2 0 0**
  d_2 = **1 0 0 0 0 0 0 1 0 2**

  $d_1 \bullet d_2 = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$

  $\|d_1\| = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$

  $\|d_2\| = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{0.5} = (6)^{0.5} = 2.449$

  $$\cos(d_1, d_2) = 0.3150$$

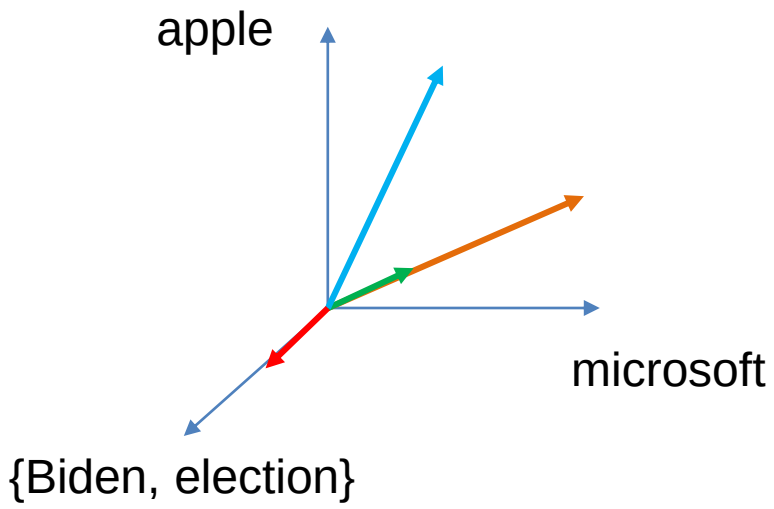
Example

document	Apple	Microsoft	Biden	Election
D1	10	20	0	0
D2	30	60	0	0
D3	60	30	0	0
D4	0	0	10	20

Cos(D1,D2) = 1

Cos (D3,D1) = Cos(D3,D2) = 4/5

Cos(D4,D1) = Cos(D4,D2) = Cos(D4,D3) = 0



Correlation Coefficient

- The correlation coefficient measures correlation between two random variables.
- If we have observations (vectors) $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_n)$ is defined as

$$CorrCoeff(X,Y) = \frac{\sum_i (x_i - \mu_X)(y_i - \mu_Y)}{\sqrt{\sum_i (x_i - \mu_X)^2}\sqrt{\sum_i (y_i - \mu_Y)^2}}$$

- (This is essentially the cosine similarity between the normalized vectors where from each entry we remove the mean value of the vector.)
- The correlation coefficient takes values in [-1,1]
  - -1 negative correlation, +1 positive correlation, 0 no correlation.
- Most statistical packages also compute a p-value that measures the statistical importance of the correlation
  - Lower value – higher statistical importance

Correlation Coefficient

document	Apple	Microsoft	Biden	Election
D1	10	20	0	0
D2	30	60	0	0
D3	60	30	0	0
D4	0	0	10	20

$$CorrCoeff(X,Y) = \frac{\sum_i (x_i - \mu_X)(y_i - \mu_Y)}{\sqrt{\sum_i (x_i - \mu_X)^2}\sqrt{\sum_i (y_i - \mu_Y)^2}}$$

CorrCoeff(D1,D2) = 1

CorrCoeff(D1,D3) = CorrCoeff(D2,D3) = 0.63

CorrCoeff(D1,D4) = CorrCoeff(D2,D4) = CorrCoeff(D3,D4) = -0.81

Distance

- Numerical measure of how different two data objects are
  - A function that maps pairs of objects to real values
  - Lower when objects are more alike
  - Higher when two objects are different
- Minimum distance is 0, when comparing an object with itself.
- Upper limit varies

Distance Metric

- A distance function d is a metric if it is a function from pairs of objects to real numbers such that:
  1. $d(x, y) \geq 0$. (non-negativity 非负性)
  2. $d(x, y) = d(y, x)$. (symmetry 对称性)
  3. $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality 三角不等式).

Distances for real vectors

L_p norms are known to be distance metrics

- Vectors $x = (x_1, \dots, x_d)$ and $y = (y_1, \dots, y_d)$

- L_2-norm L_2-范数: Euclidean distance 欧几里得距离:

$$L_2(x, y) = \sqrt{|x_1 - y_1|^2 + \cdots + |x_d - y_d|^2}$$

- L_1-norm: Manhattan distance 曼哈顿距离:

$$L_1(x, y) = |x_1 - y_1| + \cdots + |x_d - y_d|$$

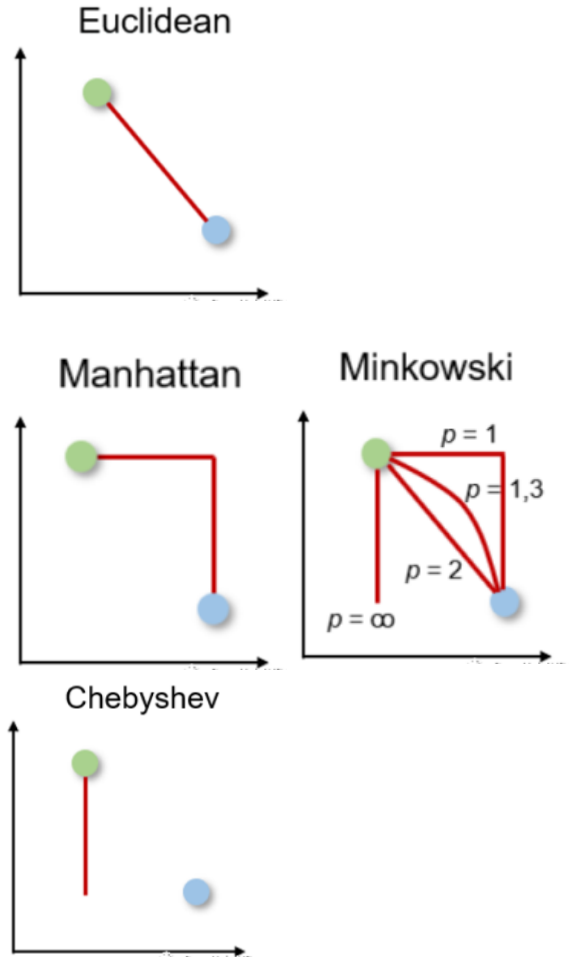
- L_p-norms or Minkowski distance 闵可夫斯基距离:

$$L_p(x, y) = [|x_1 - y_1|^p + \cdots + |x_d - y_d|^p]^{1/p}$$

- L_∞-norm $L_{\max}$-范数;切比雪夫距离 Chebyshev distance; 上确界距离:

$$L_\infty(x, y) = \max\{|x_1 - y_1|, \dots, |x_d - y_d|\}$$

  - The limit of L_p as p goes to infinity.

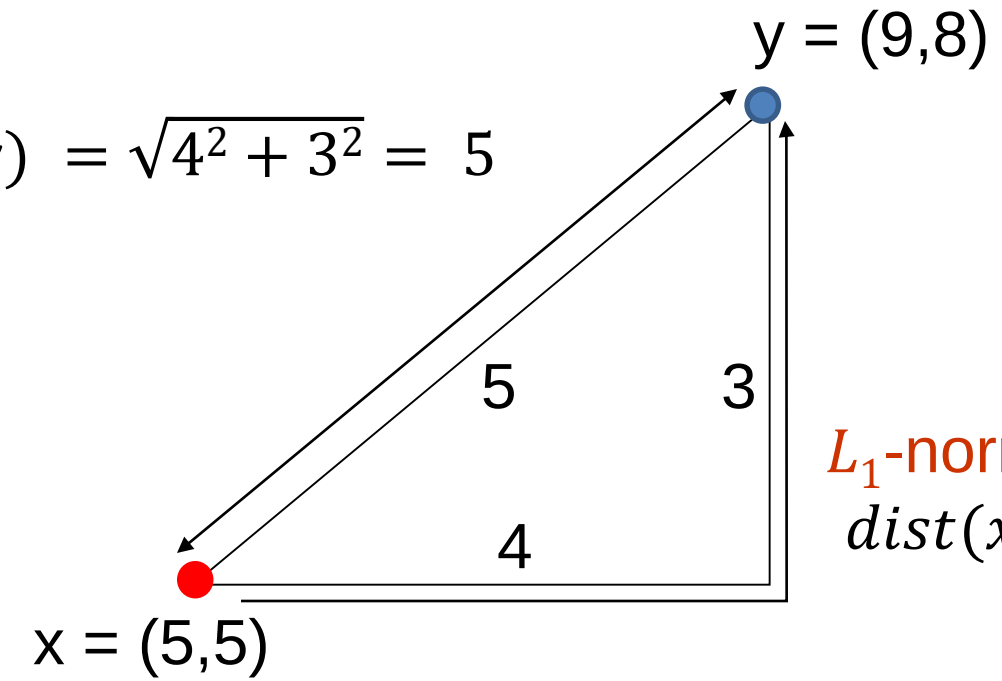


《数据挖掘导论》p75

Example of Distances

L_2-norm:

$$dist(x, y) = \sqrt{4^2 + 3^2} = 5$$



L_1-norm:

$$dist(x, y) = 4 + 3 = 7$$

L_∞-norm:

$$dist(x, y) = \max\{3,4\} = 4$$

L_p distances for sets

- We can apply all the L_p distances to the cases of sets of attributes, with or without counts, if we represent the sets as vectors
  - E.g., a transaction is a 0/1 vector
  - E.g., a document is a vector of counts.

Similarities into distances

- Jaccard distance:

$$JDist(X, Y) = 1 - JSim(X, Y)$$

- Jaccard Distance is a metric

- Cosine distance:

$$Dist(X, Y) = 1 - \cos(X, Y)$$

- Cosine distance is a metric

Hamming Distance

- Hamming distance is the number of positions in which bit-vectors differ.
  - Example:
    - p_1 = 10101
    - p_2 = 10011.
    - $d(p_1, p_2) = 2$ because the bit-vectors differ in the 3rd and 4th positions.
    - The L_1 norm for the binary vectors

- Hamming distance between two vectors of categorical attributes is the number of positions in which they differ.
  - Example:
    - x = (married, low income, credit card cheat)
    - y = (single, low income, not credit card cheat)
    - $d(x, y) = 2$

Distance between strings

- How do we define similarity between strings?

weird	wierd
intelligent	unintelligent
Athena	Athina

- Important for recognizing and correcting typing errors.

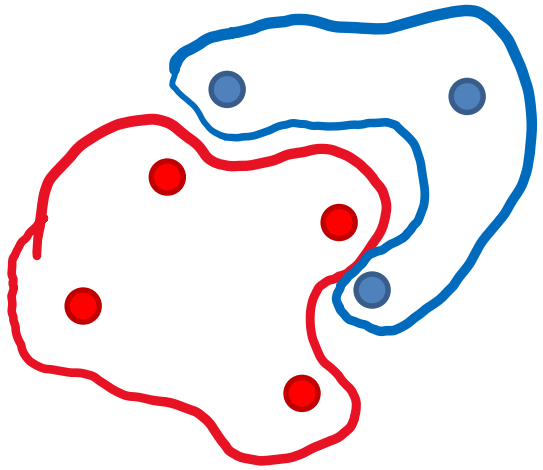
Edit Distance for strings

- The edit distance of two strings is the number of inserts and deletes of characters needed to turn one into the other.
- Example: x = abcde ; y = bcduve.
  - Turn *x* into *y* by deleting a, then inserting u and v after d.
  - Edit distance = 3.
- Minimum number of operations can be computed using dynamic programming
- Common distance measure for comparing DNA sequences

Variant Edit Distances

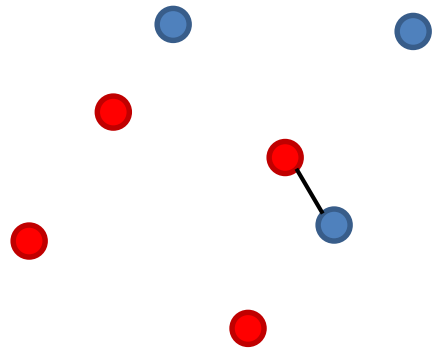
- Allow insert, delete, and mutate (換位).
  - Change one character into another.
- Minimum number of inserts, deletes, and mutates also forms a distance measure.

Distance between sets of points



How do we measure the distance between the two sets?

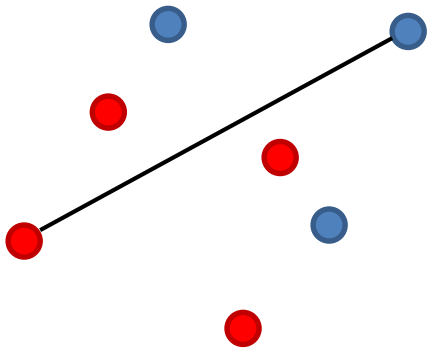
Distance between sets of points



How do we measure the distance between the two sets?

Minimum distance over all pairs

Distance between sets of points

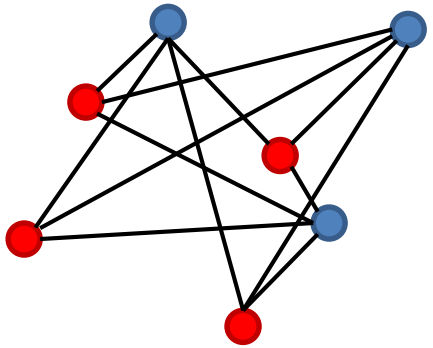


How do we measure the distance between the two sets?

Minimum distance over all pairs

Maximum distance over all pairs

Distance between sets of points



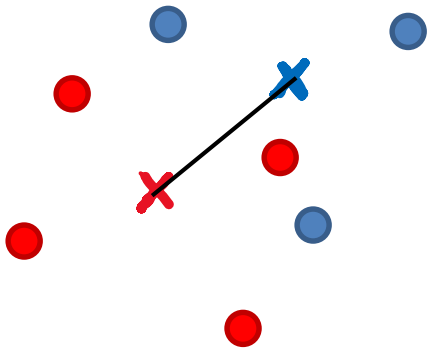
How do we measure the distance between the two sets?

Minimum distance over all pairs

Maximum distance over all pairs

Average distance over all pairs

Distance between sets of points



How do we measure the distance between the two sets?

Minimum distance over all pairs

Maximum distance over all pairs

Average distance over all pairs

Distance between averages

Distances between distributions

- Some times data can be represented as a distribution (e.g., a document is a distribution over the words)

document	Apple	Microsoft	Biden	Election
D1	0.35	0.5	0.1	0.05
D2	0.4	0.4	0.1	0.1
D3	0.05	0.05	0.6	0.3

- How do we measure distance between distributions?

Variational distance

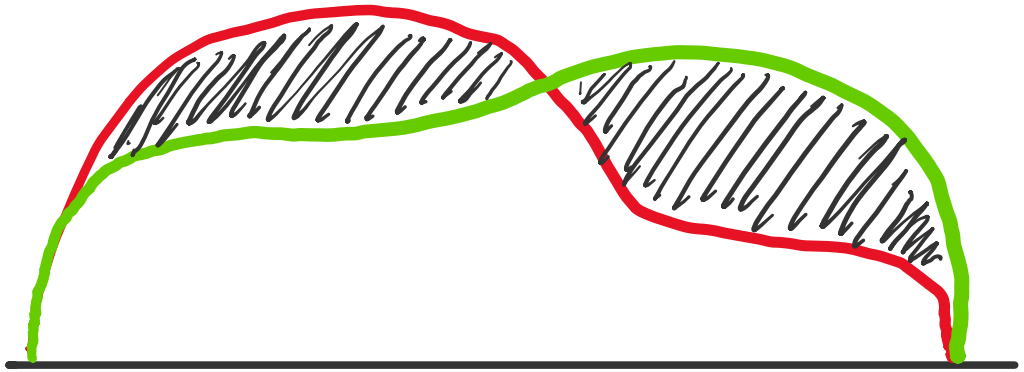
- Variational distance: The L_1 distance between the distribution vectors

document	Apple	Microsoft	Biden	Election
D1	0.35	0.5	0.1	0.05
D2	0.4	0.4	0.1	0.1
D3	0.05	0.05	0.6	0.3

Dist(D1,D2) = 0.05+0.1+0.05 = 0.2

Dist(D2,D3) = 0.35+0.35+0.5+ 0.2 = 1.4

Dist(D1,D3) = 0.3+0.45+0.5+ 0.25 = 1.5



Information theoretic distances

document	Apple	Microsoft	Biden	Election
D1	0.35	0.5	0.1	0.05
D2	0.4	0.4	0.1	0.1
D3	0.05	0.05	0.6	0.3

- KL-divergence (Kullback-Leibler) for distributions P,Q

$$D_{KL}(P\|Q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

- From Information Theory: amount of additional information needed to represent distribution P, given distribution Q.
- KL-divergence is asymmetric. We can make it symmetric by taking the average of both sides

$$\frac{1}{2}\left(D_{KL}(P\|Q) + D_{KL}(Q\|P)\right)$$

假设你有一枚骰子，它是均匀的，每个面的概率都是1/6。这是一个已知的真实分布，我们将其表示为分布P。
你的朋友拿来另一枚骰子，但你不确定它每个面的概率。这是一个未知的分布，我们将其表示为分布Q。
你想知道这两个骰子之间的差异有多大，这时你可以使用KL散度来衡量它们之间的距离。计算公式如下：
$D_{KL}(P||Q) = \Sigma P(x) * \log(P(x) / Q(x))$
对于我们的例子，真实分布P是均匀分布，每个面的概率都是1/6。现在假设朋友的骰子的概率分布Q如下：
Q(1) = 1/12
Q(2) = 1/12
Q(3) = 1/6
Q(4) = 1/6
Q(5) = 1/3
Q(6) = 1/6
我们可以将这些值代入KL散度的计算公式，得到：
$D_{KL}(P||Q) = (1/6) * \log((1/6) / (1/12)) + (1/6) * \log((1/6) / (1/12)) + (1/6) * \log((1/6) / (1/6)) + (1/6) * \log((1/6) / (1/6)) + (1/6) * \log((1/6) / (1/3)) + (1/6) * \log((1/6) / (1/6))$
通过计算可以得到KL散度的值为：
$D_{KL}(P||Q) \approx 0.231$

Ranking distances

- The input in this case is two rankings/orderings of the same n items. For example:

$$R_1 = \langle x, y, z, w \rangle$$
$$R_2 = \langle y, w, z, x \rangle$$

  - How do we define distance in this case?
- Kendall's tau: All possible 6 pairs without orders:

$(x, y), (x, z), (x, w), (y, z), (y, w), (z, w)$

- For each pair, examine whether it has the same order in R_1 and R_2
  - 对于(x, y)， R_1中，x在前，y在后，而在R_2中，y在前，x在后
- Number of pairs of items that are in different order:

$$|\{(x, y), (x, z), (x, w), (z, w)\}| = 4$$

  - Maximum: $\frac{n(n-1)}{2}$ when rankings are reversed.
- Spearman rank distance: L_1 distance between the ranks
  - $SR(R_1, R_2) = |1 - 4| + |2 - 1| + |3 - 3| + |4 - 2| = 6$

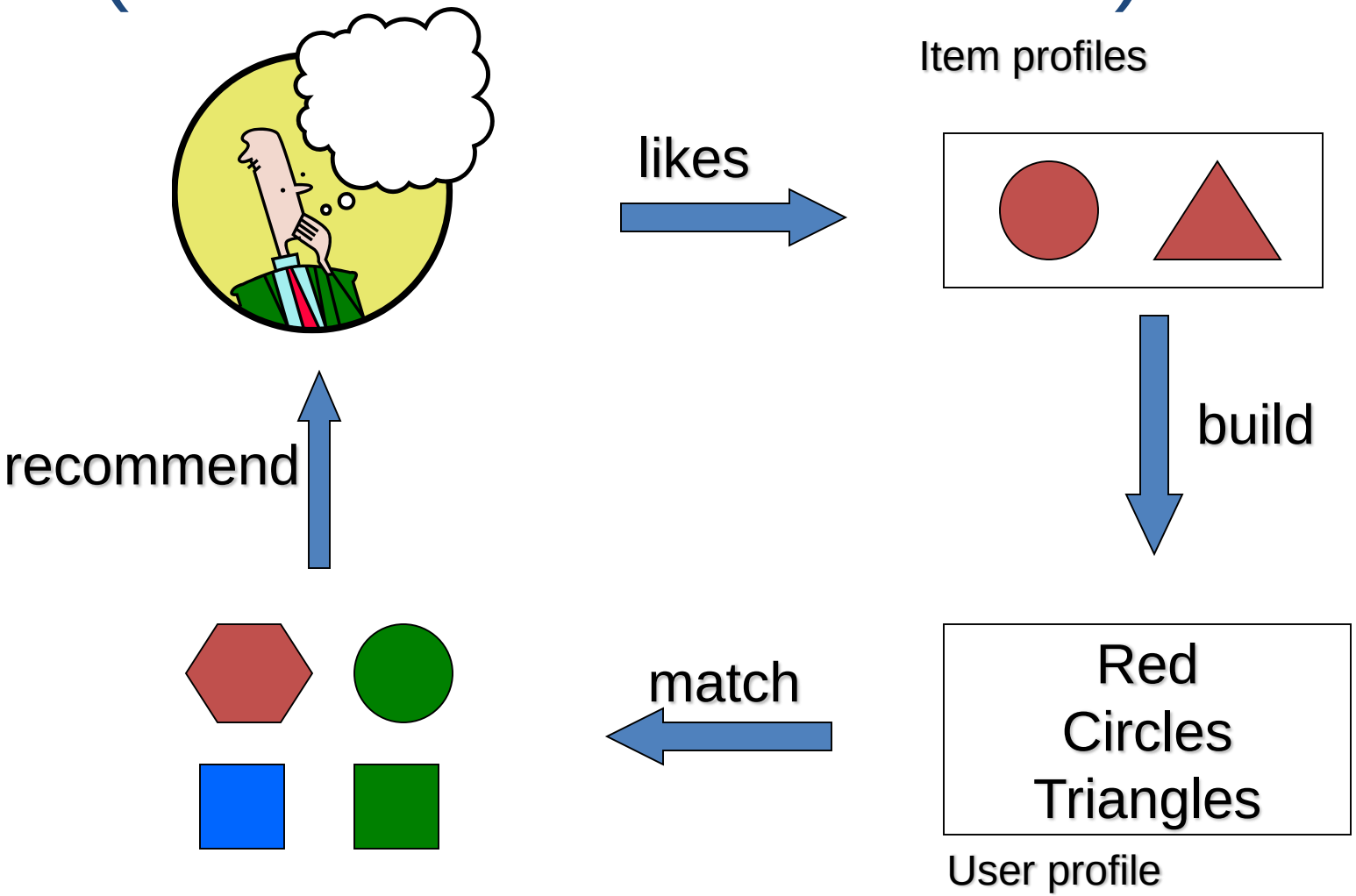
	x	y	z	w
R_1	1	2	3	4
R_2	4	1	3	2

APPLICATIONS OF SIMILARITY: RECOMMENDATION SYSTEMS

An important problem

- Recommendation systems
  - When a user buys an item (initially books) we want to recommend other items that the user may like
  - When a user rates a movie, we want to recommend movies that the user may like
  - When a user likes a song, we want to recommend other songs that they may like

Intuition (recommend similar items)



Recommendation Systems

- Content-based:
  - Represent the items into a feature space and recommend items to customer C similar to previous items rated highly by C
    - Movie recommendations: recommend movies with same actor(s), director, genre, …
    - Websites, news: recommend other sites with “similar” content

Limitations of content-based approach

- Finding the appropriate features
  - e.g., images, movies, music
  - Embeddings and deep learning can help
- Overspecialization
  - Never recommends items outside user's content profile
  - People might have multiple interests
- Recommendations for new users
  - How to build a profile?

An alternative: Collaborative filtering

Similar users like similar movies.

Two users are similar if they rate the same items in a similar way

Recommend to user C, the items liked by many of the most similar users.

	Harry Potter 1	Harry Potter 2	Harry Potter 3	Twilight	Star Wars 1	Star Wars 2	Star Wars 3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Rows: Users

Columns: Movies (in general Items)

Values: The rating of the user for the movie

User Similarity

	Harry Potter 1	Harry Potter 2	Harry Potter 3	Twilight	Star Wars 1	Star Wars 2	Star Wars 3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Which pair of users do you consider as the most similar?

What is the right definition of similarity?

User Similarity

	Harry Potter 1	Harry Potter 2	Harry Potter 3	Twilight	Star Wars 1	Star Wars 2	Star Wars 3
A	1			1	1		
B	1	1	1				
C				1	1	1	
D		1					1

Jaccard Similarity: users are sets of movies

Disregards the ratings.

Jsim(A,B) = 1/5

Jsim(A,C) = 1/2

Jsim(B,D) = 1/4

User Similarity

	Harry Potter 1	Harry Potter 2	Harry Potter 3	Twilight	Star Wars 1	Star Wars 2	Star Wars 3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Cosine Similarity:

Cos(A,B) = 0.38
Cos(A,C) = 0.32
Zero entries, 0 or negative value?

User Similarity

	Harry Potter 1	Harry Potter 2	Harry Potter 3	Twilight	Star Wars 1	Star Wars 2	Star Wars 3
A	2/3			5/3	-7/3		
B	1/3	1/3	-2/3				
C				-5/3	1/3	4/3	
D		0					0

Normalized Cosine Similarity:

- Subtract the mean rating per user (without the zeros) and then compute Cosine (correlation coefficient)

Corr(A,B) = 0.092
Corr(A,C) = -0.559

User-User Collaborative Filtering

- For a user u, find the set $TopK(u)$ of the K users whose ratings are most "similar" to u's ratings
- Estimate u's ratings based on ratings of users in $TopK$ using some aggregation function. For item i:

$$\widehat{r_{ui}} = \frac{1}{Z} \sum_{v \in TopK(u)} \text{sim}(u,v) r_{vi}$$

$$Z = \sum_{v \in TopK(u)} \text{sim}(u,v)$$

  - Modeling deviations:

$$\widehat{r_{ui}} = \overline{r_u} + \frac{1}{Z} \sum_{v \in TopK(u)} \text{sim}(u,v)(r_{vi} - \overline{r_v})$$

Mean rating of u

Deviation from mean for v

Mean deviation of similar users

Evaluation

- Split the data into train and test set
  - Keep a fraction of the ratings to test the accuracy of the predictions

- Metrics:
  - Root Mean Square Error (RMSE) for measuring the quality of predicted ratings:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i,j}\left(\widehat{r_{ij}} - r_{ij}\right)^2}$$

  - Precision 准确率/Recall 召回率 for measuring the quality of binary (action/no action) predictions:
    - Precision = fraction of predicted actions that were correct
    - Recall = fraction of actions that were predicted correctly
    - We may also consider the top-k recommendations for the evaluation

  - Kendall's tau for measuring the quality of predicting the ranking of items:
    - The fraction of pairs of items that are ordered correctly (or incorrectly)

The Netflix Challenge

- 1M prize to improve the prediction accuracy by 10%