13 – RECOMMENDER SYSTEMS

CS 1656

Introduction to Data Science

Alexandros Labrinidis – http://labrinidis.cs.pitt.edu
University of Pittsburgh

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

What are recommender systems?

- Recommender systems or recommendation systems are a subclass of *information filtering system* that seek to predict the 'rating' or 'preference' that user would give to an item.

[Source: http://en.wikipedia.org/wiki/Recommender_system]

- Most popular type of recommender systems: **collaborative filtering**

[Source: http://en.wikipedia.org/wiki/Collaborative_filtering]

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Examples of Recommender Systems

- Movies
- Music
- News
- Books
- Research articles
- Search queries
- Products in general
- Restaurants
- Social Networks:
  - friends / followers / likes
- Online dating
- App store

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

What is the filter bubble?

- A result of a personalized search in which a website algorithm selectively guesses what information a user would like to see
  - This happens based on information about the user (e.g., location, past click and search history)

- As a result, users become separated from information that disagrees with their viewpoints, effectively isolating them in their own cultural or ideological bubbles

  - [Source: http://en.wikipedia.org/wiki/Filter_bubble]

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

COLLABORATIVE FILTERING

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

STEP 1: COLLECT DATA

Show Me The Data

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Different Users Different Ratings – 1

- Alice 2 stars
- Bob 3 stars
- Christine 4 stars
- David 5 stars
- Elaine 5 stars
- Frank has not seen it

[images used for educational purposes only]

Different Users Different Ratings – 2

[images used for educational purposes only]

- Alice 5 stars
- Bob has not seen it
- Christine 5 stars
- David has not seen it
- Elaine 3 stars
- Frank 3 stars

Different Users Different Ratings – 3

[images used for educational purposes only]

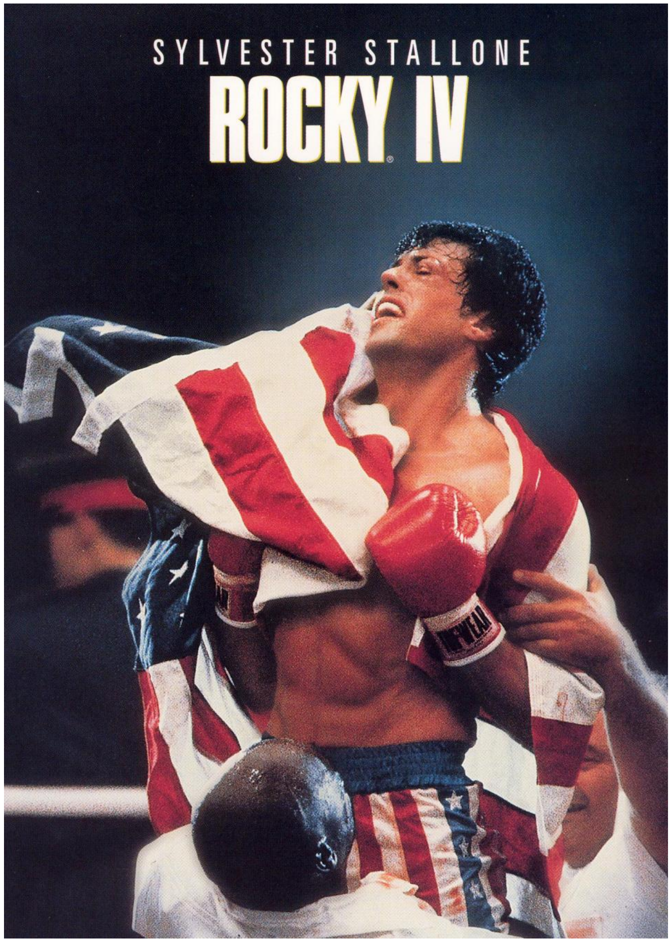
- Alice 2 stars
- Bob 1 star
- Christine 2 stars
- David 2 stars
- Elaine 1 star
- Frank 1 star

Different Users Different Ratings – 4

[images used for educational purposes only]

- Alice 4 stars
- Bob 4 stars
- Christine 5 stars
- David 2 stars
- Elaine has not seen it
- Frank 3 stars

Different Users Different Ratings – 5



[images used for educational purposes only]

• Alice	2 stars
• Bob	2 stars
• Christine	3 stars
• David	4 stars
• Elaine	3 stars
• Frank	has not seen it

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Movie Ratings

	The Matrix	Gone with the Wind	Jack and Jill	Planes	Rocky IV
Alice	2	5	2	4	2
Bob	3		1	4	2
Christine	4	5	2	5	3
David	5		2	2	4
Elaine	5	3	1		3
Frank		3	1	3	

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

STEP 2: USE THE DATA

Extremely Simple Approach

Source: http://xkcd.com/1098/

How can we compare different users?

- Treat ratings as sets and use Jaccard Similarity
  - i.e., ratio of intersection over union of datasets

$$J(A,B) = \frac{\|A \bigcap B|}{\|A \bigcup B\|}$$

[Source: http://en.wikipedia.org/wiki/Jaccard_index]

  - For ratings, it could be used to identify which ratings were the same

- Example:
  - J(Alice, Bob) = 2/5 = 0.4
  - J(David, Elaine) = 1/5 = 0.2

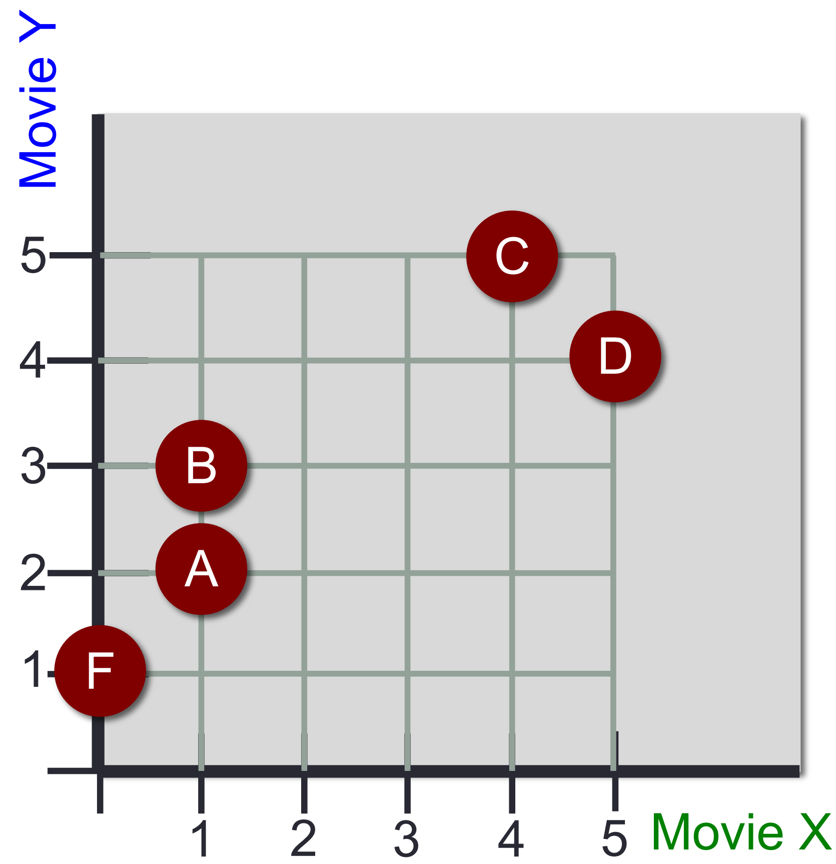
- **Q**: What is the problem with this method?

How can we compare different users?

- Create a plot
  - X axis is rating for one movie
  - Y axis is rating for another move
  - Points reflect ratings for the two movies from a particular user

= people in preference space

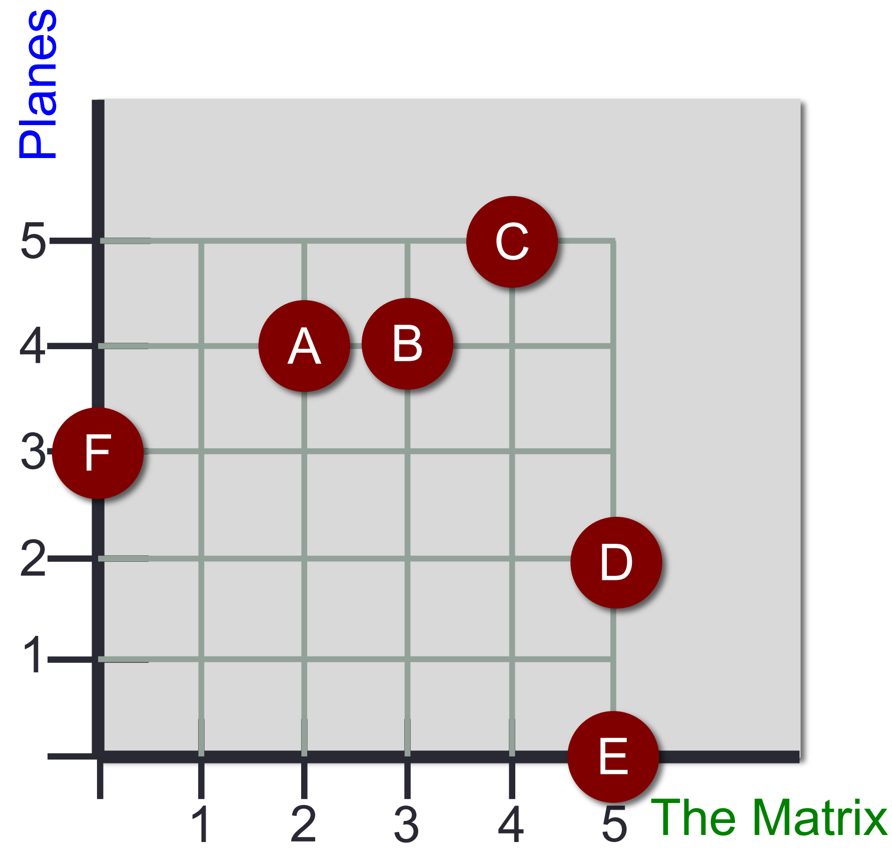
- A: 1 star for movie X and 2 stars for movie Y
- B: 1 star for movie X and 3 stars for movie Y
- F: has not seen movie X 1 star for movie Y



How can we compare different users?

- Real example
  - The Matrix vs Planes

  - A: 2 4
  - B: 3 4
  - C: 4 5
  - D: 5 2
  - E: 5 -
  - F: - 3



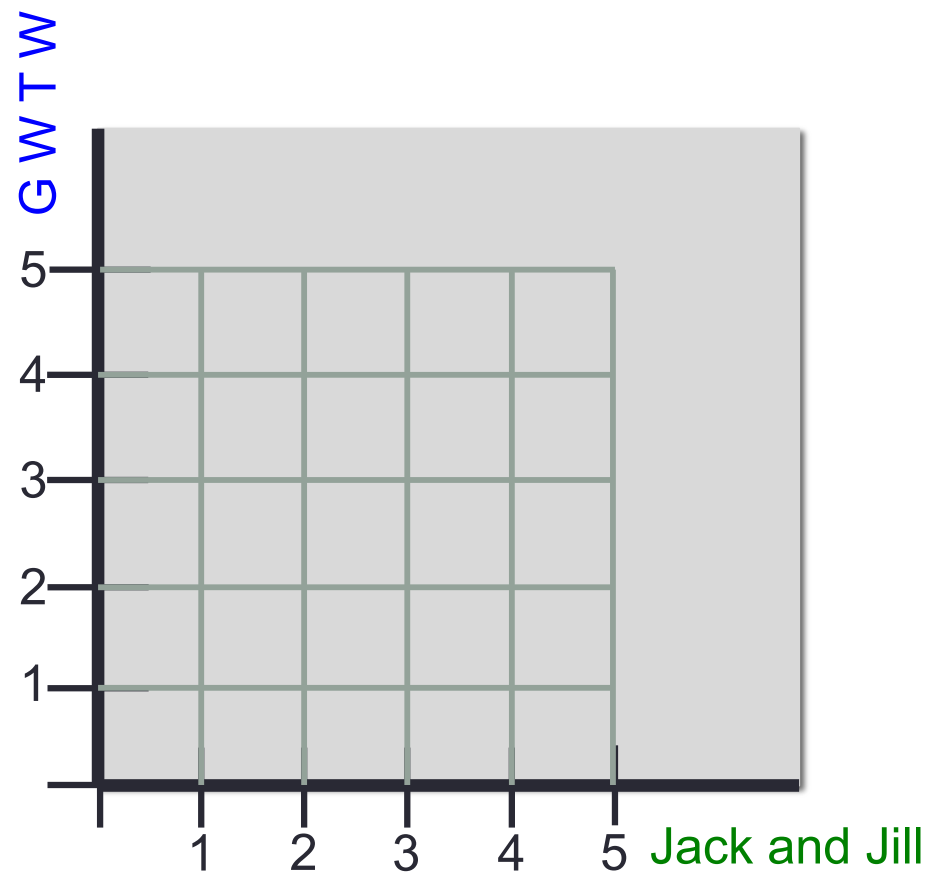
- The closer users are, the more similar they are

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Question #1: Draw similarity

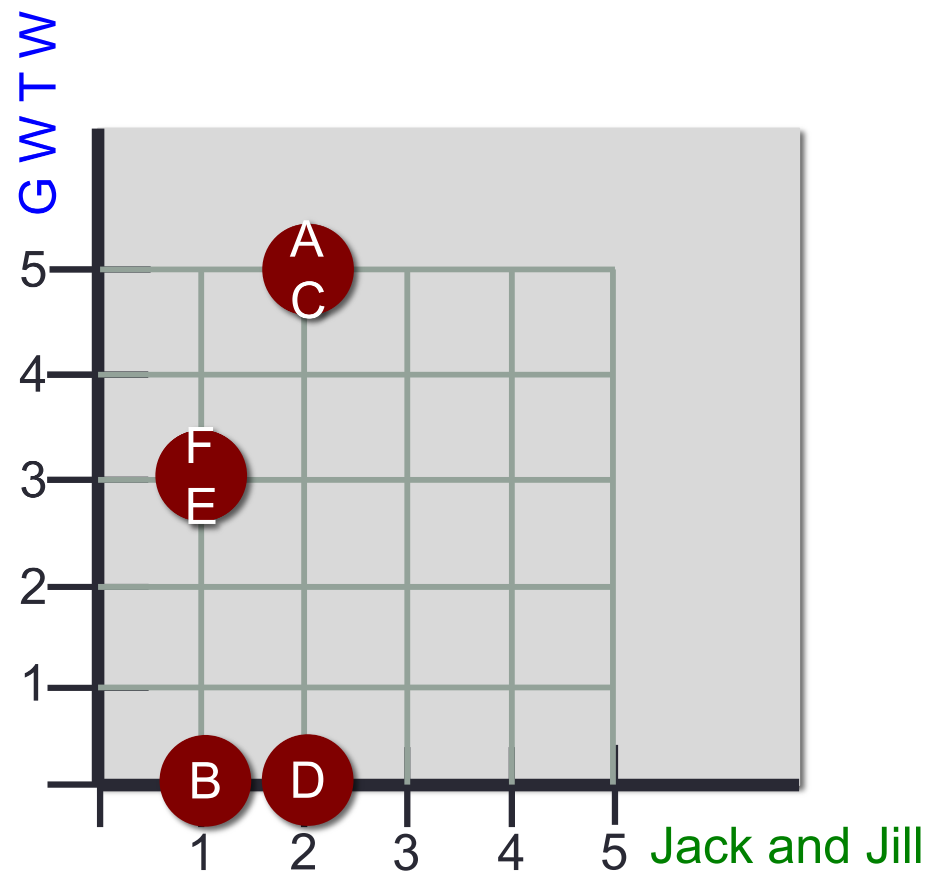
- Real example
  - Jack and Jill vs Gone with the Wind

A:	2	5
B:	1	-
C:	2	5
D:	2	-
E:	1	3
F:	1	3



Question #1: Draw similarity

- Real example
  - Jack and Jill vs Gone with the Wind
  - A: 2 5
  - B: 1 -
  - C: 2 5
  - D: 2 -
  - E: 1 3
  - F: 1 3
  - Answer = 2

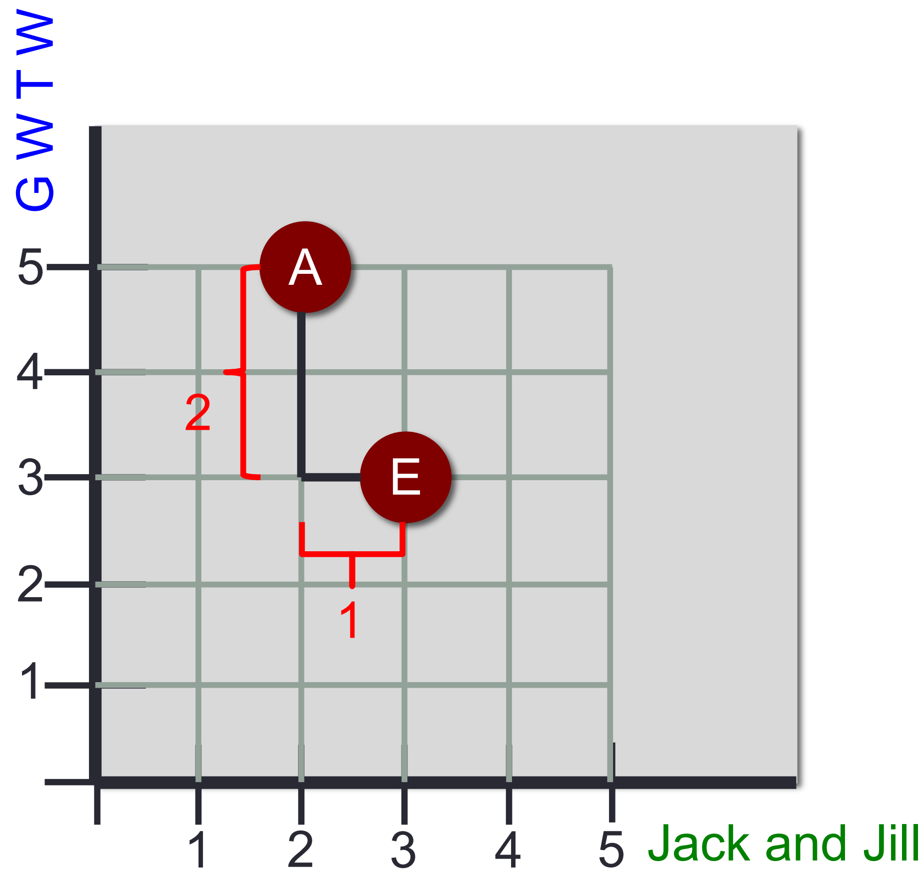


How to compute similarity? (I)

- Start by computing the **distance** between two users!

- (I) Manhattan Distance
  - Distance between points
    - A (x1, y1), and
    - B (x2, y2) is:

$$\|x_1 - x_2\| + \|y_1 - y_2\|$$

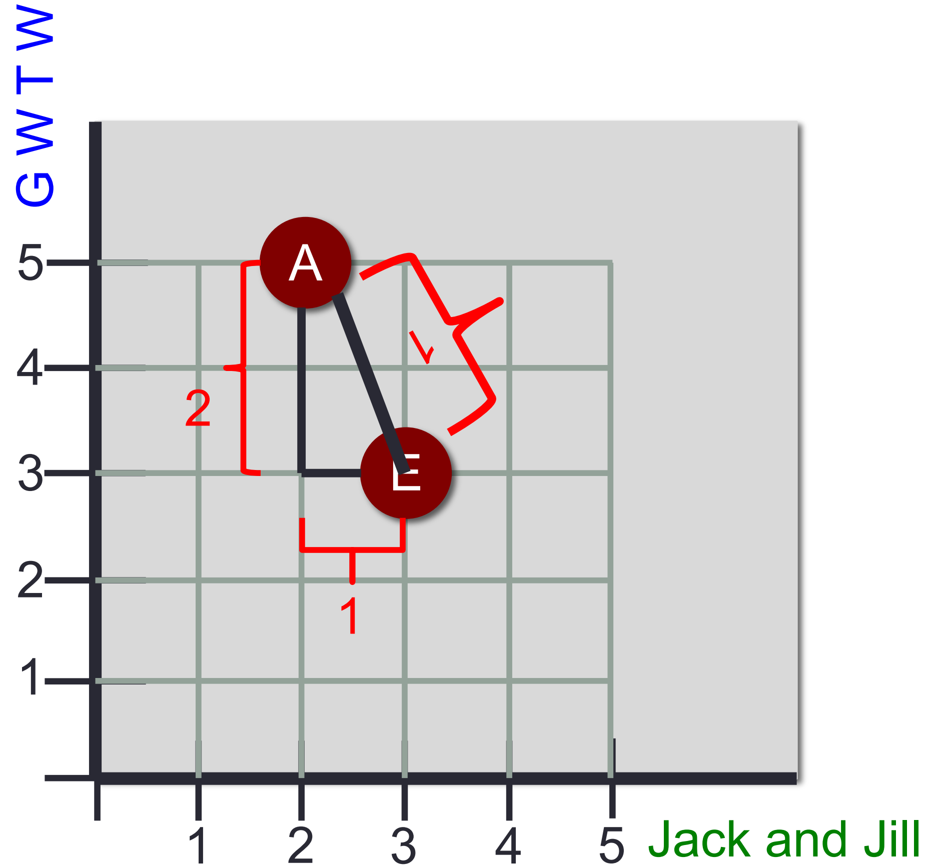


How to compute similarity? (II)

- Start by computing the **distance** between two users!

- (II) Euclidean Distance
  - Distance between points
    - A (x_1, y_1), and
    - B (x_2, y_2) is:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$



How to compute similarity? (III)

- Given **distance** metric (e.g., Manhattan or Euclidean)
- We need to compute **similarity** between two people i, j

- Formula:

$$sim(i,j) = \frac{1}{1 + distance(i,j)}$$

- Limits:
  - Similarity score is close to 0, if distance is close to ∞
  - Similarity score is 1, if distance is 0.

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Computing Distances on n Dimensions

- Distance formulas are trivially generalizable for multiple dimensions
  - I.e., to compute distance of people over multiple movies

- Example:
  - Assume we want to compute the distance of **Frank** to all other users. For simplicity, let's assume Manhattan Distance.

  - distance (Alice, Frank) = |5-3| + |2-1| + |4-3| = 4
    - Note: we do not compute distance if there is no rating for one of the users.
  - distance (Bob, Frank) = |1-1| + |4-3| = 1

Q2. Understanding Question

- **Question**:
  - Compute *distance(Christine, Frank)*, *distance(David, Frank)*, and *distance(Elaine, Frank)*. Who has the highest distance from Frank?

- **Possible Answers**:
  - Alice
  - Bob
  - Christine
  - David
  - Elaine

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Q2. Understanding Question (Answer)

- **Question**:
  - Compute *distance(Christine, Frank)*, *distance(David, Frank)*, and *distance(Elaine, Frank)*. Who has the highest distance from Frank?

- **Answer**:
  - Alice d(Alice, Frank) = 4
  - Bob d(Bob, Frank) = 1
  - **Christine** d(Christine,Frank) = |5-3| + |2-1| + |5-3| = 5
  - David d(David,Frank) = |2-1| + |2-3| = 2
  - Elaine d(Elaine, Frank) = |3-3| + |1-1| = 0

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Distance → Similarity

- distance(Alice, Frank) = 4
  - **similarity** (Alice, Frank) = 1 / (1+4) = 1/5 = 0.2
- distance(Bob, Frank) = 1
  - **similarity** (Bob, Frank) = 1 / (1+1) = 1/2 = 0.5
- distance(Christine,Frank) = 5
  - **similarity** (Christine, Frank) = 1 / (1+5) = 1/6 = 0.1667
- distance(David,Frank) = 2
  - **similarity** (David, Frank) = 1 / (1+2) = 1/3 = 0.3333
- distance(Elaine, Frank) = 0
  - **similarity** (Elaine, Frank) = 1 / (1+0) = 1

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

MAKING PREDICTIONS

Movie Ratings → Prediction

	The Matrix	Gone with the Wind	Jack and Jill	Planes	Rocky IV
Alice	2	5	2	4	2
Bob	3		1	4	2
Christine	4	5	2	5	3
David	5		2	2	4
Elaine	5	3	1		3
Frank		3	1	3	

Q: What if we just used the **average rating** to predict the missing ratings?

A: Although this may work in some cases (e.g., Jack and Jill), it will not work for the general case!

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Solution

- We utilize the **similarity metric**, to give more weight to ratings from users who are similar to user in question

- **Compute weighted average rating:**

$$\text{predicted rating} = \frac{\sum (w_i * r_i)}{\sum w_i}$$

- Note: only include non-zero ratings

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Movie Ratings + Similarity

	Similarity to Frank	The Matrix	Gone with the Wind	Jack and Jill	Planes	Rocky IV
Alice	0.2	**2**	5	2	4	2
Bob	0.5	**3**		1	4	2
Christine	0.1667	**4**	5	2	5	3
David	0.3333	**5**		2	2	4
Elaine	1.0	**5**	3	1		3
Frank			3	1	3	

prediction: **rating(Frank, The Matrix)** =
(0.2 * 2 + 0.5 * 3 + 0.1667 * 4 + 0.3333 * 5 + 1.0 * 5) /
(0.2 + 0.5 + 0.1667 + 0.3333 + 1) = 9.2333 / 2.2 = **4.1969**

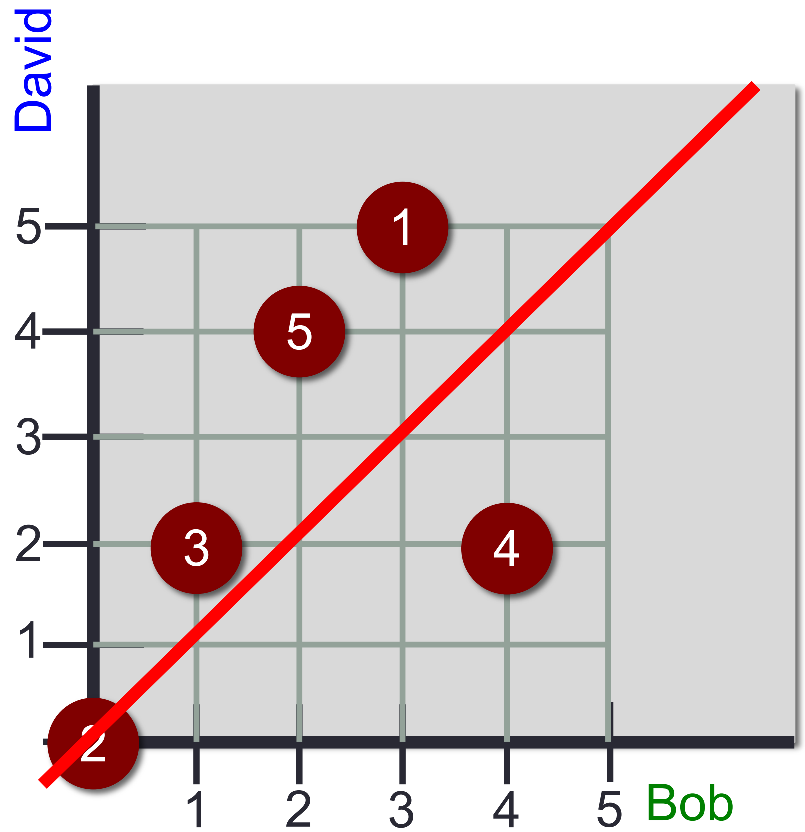
VS (2+3+4+5+5) / 5 = 3.8

DISTANCE SENSITIVITY

What if visually compared two users?

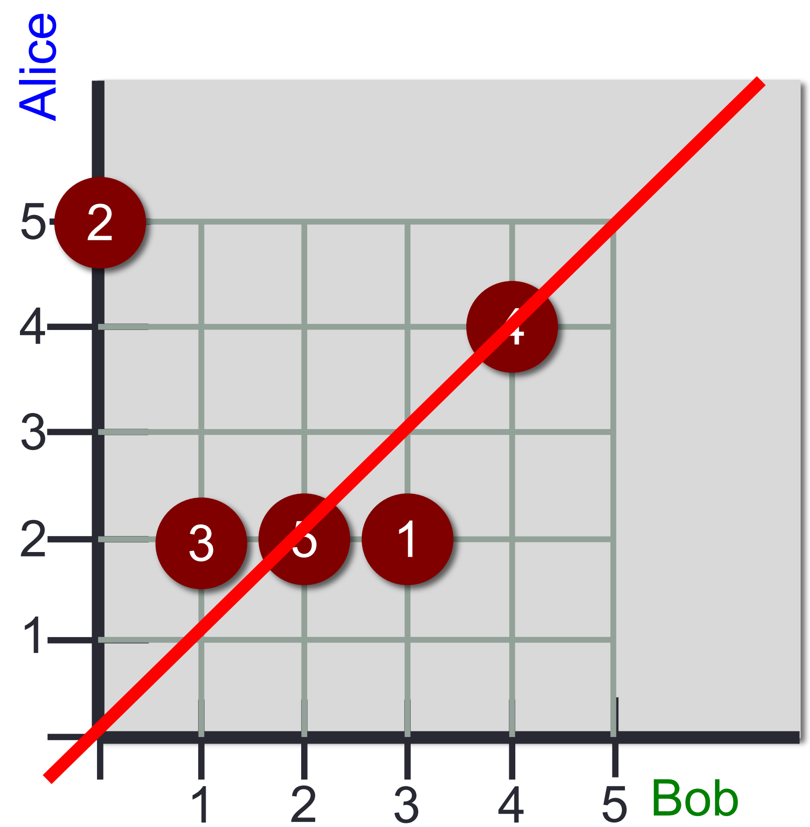
- Two users as axes
- Movies are points

  - 1: The Matrix
  - 2: Gone with the Wind
  - 3: Jack and Jill
  - 4: Planes
  - 5: Rocky IV



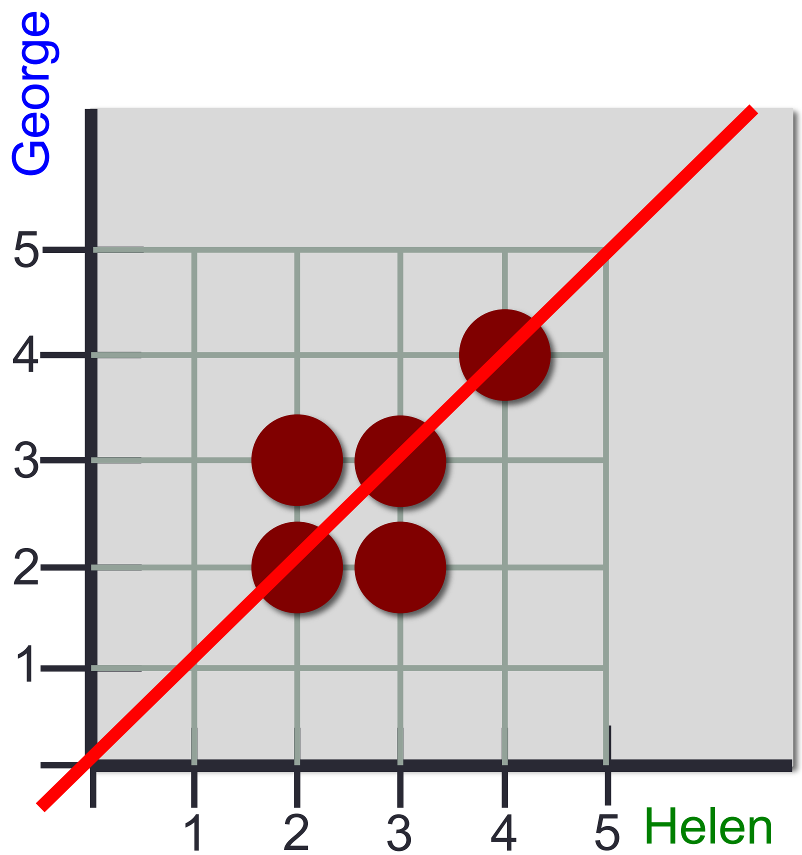
What if visually compared two users?

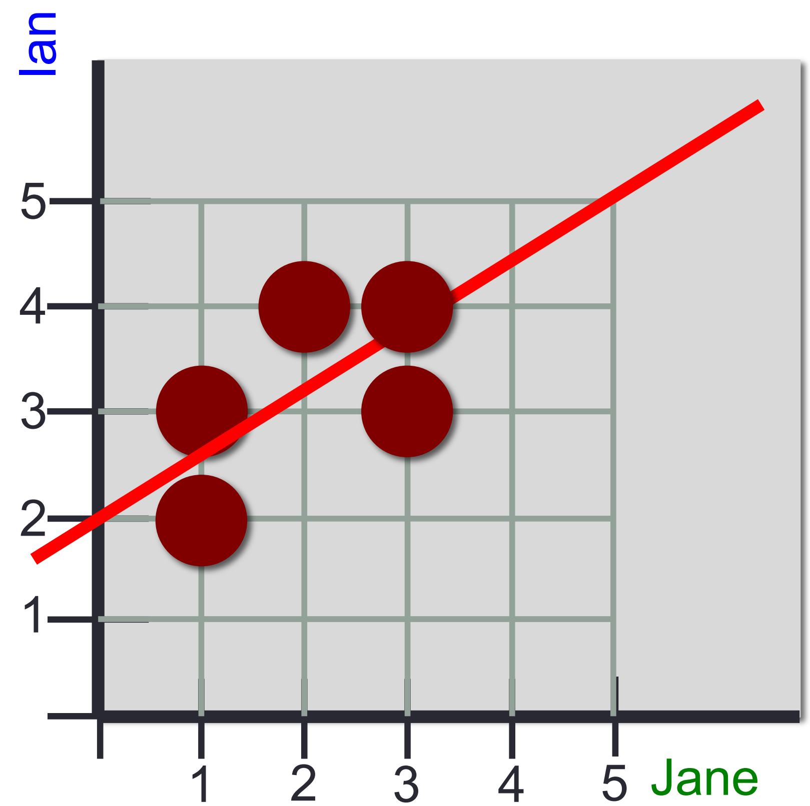
- Two users as axes
- Movies are points
  - 1: The Matrix
  - 2: Gone with the Wind
  - 3: Jack and Jill
  - 4: Planes
  - 5: Rocky IV



Points on the diagonal ➔ complete agreement

What if visually compared two users?





On diagonal ➔ complete agreement

Different angle line ➔ agreement with skew (e.g., grade inflation)

Another Similarity Metric

- **Pearson Correlation Coefficient**
  - Measures how well two data sets fit on a straight line
  - Ranges from -1 to 1, inclusive
  - -1 ➔ perfect disagreement
  - +1 ➔ perfect agreeement

- Negative:
  - More complicated formula

- Positive:
  - Gives better results when ratings not normalized

[Source: http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient]

Pearson Correlation Coefficient

- Original Formula:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

- (Simpler to compute) Approximation:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y)^2)}}$$

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

General Case

1. Collect ratings from users

2. Assign a weight to all users with respect to similarity with the active user
   - multiple metrics for similarity

3. Select k users that have the highest similarity with the active user
   - Commonly called the *neighborhood*

4. Compute a prediction score from a weighted combination of the selected neighborhood ratings

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Extensions

- Plethora of similarity functions
  - E.g., cosine similarity (will cover next time)
- Item-based similarity (instead of user-based similarity)
  - Better for sparse datasets
- Further reading:
  - *Item-Based Collaborative Filtering Recommendation Algorithms* http://files.grouplens.org/papers/www10_sarwar.pdf

ITEM-BASED COLLABORATIVE FILTERING

Movie Ratings

	The Matrix	Gone with the Wind	Jack and Jill	Planes	Rocky IV
Alice	2	5	2	4	2
Bob	3		1	4	2
Christine	4	5	2	5	3
David	5		2	2	4
Elaine	5	3	1		3
Frank		3	1	3	

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

User-Based Collaborative Filtering

1. Collect ratings from users

2. Assign a weight to all users with respect to similarity with the active user
   - multiple metrics for similarity

3. Select k users that have the highest similarity with the active user
   - Commonly called the *neighborhood*

4. Compute a prediction score from a weighted combination of the selected neighborhood ratings

Issues with user-based collaborative filtering

- **Challenge #1**: finding enough data for each user
- Movie reviews submitted by students of CS1655 class (Fall 2014 term)
  - 16 students submitted reviews
  - 159 movie reviews
  - 140 unique movies!
  - Movies with 3 reviews:
    - Despicable Me, Inception, Pulp Fiction, The Avengers, The Dark Knight
  - Movies with 2 reviews:
    - Aliens, Blood Diamond, Fight Club, Gladiator, Predator, The Exorcist, The Matrix, The Room, The Shining
- **Challenge #2**: need to recompute often, as users add new rankings

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Movie Ratings (Item-centric view)

	Alice	Bob	Christine	David	Elaine	Frank
The Matrix	2	3	4	5	5	
Gone with the Wind	5		5		3	3
Jack and Jill	2	1	2	2	1	1
Planes	4	4	5	2		3
Rocky IV	2	2	2	4	3	

Movie Ratings (Item-centric view)

The Matrix	2	3	4	5	5	
Gone with the Wind	5		5		3	3
Jack and Jill	2	1	2	2	1	1
Planes	4	4	5	2		3
Rocky IV	2	2	2	4	3	

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Compute Similarity

The Matrix	2	3	4	5	5	
Gone with the Wind	5		5		3	3

- (1) Identify users who reviewed both movies

And perform one of two options:

- (2a) Compute distance and then similarity
  - E.g., Manhattan, Euclidean
- (2b) Compute similarity directly
  - E.g., Pearson, **Cosine Similarity**

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Cosine Similarity

- Treat movie ratings for each movie as **vectors** (A and B) in m-dimensional space (m=number of users)
  - Measure similarity by computing the cosine of the angle between the two vectors:

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

[Source: http://en.wikipedia.org/wiki/Cosine_similarity]

- The resulting similarity ranges from **−1** meaning exactly opposite, to **1** meaning exactly the same, with **0** usually indicating independence, and in-between values indicating intermediate similarity or dissimilarity

- Online Cosine Similarity Calculator:
  - http://calculator.vhex.net/calculator/distance/cosine-distance

Cosine Similarity Example

The Matrix	2	3	4	5	5	
Gone with the Wind	5		5		3	3

- <u>Input</u>:
  - Vector #1: [2, 4, 5]
  - Vector #2: [5, 5, 3]

$$\text{Cosine Similarity} = \frac{2*5+4*5+5*3}{\sqrt{2^2+4^2+5^2}\times\sqrt{5^2+5^2+3^2}}$$

  - Cosine similarity (Matrix, GWTW) = 0.8733

Movie Ratings Example 2

The Matrix	2	3	4	5	5	
Jack and Jill	2	1	2	2	1	1

- Vector #1: [2, 3, 4, 5, 5]
- Vector #2: [2, 1, 2, 2, 1]
- Cosine Similarity (Matrix, Jack and Jill) = 0.09021

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Movie Ratings Example 3

The Matrix	2	3	4	5	5	
Planes	4	4	5	2		3

- Vector #1: [2, 3, 4, 5]   Vector #2: [4, 4, 5, 2]
- Cosine similarity (Matrix, Planes) = 0.8711

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Movie Ratings Example 4

The Matrix		2	3	4	5	5
Rocky IV		2	2	2	4	3

- Vector #1: [2, 3, 4, 5, 5] Vector #2: [2, 2, 2, 4, 3]
- Cosine similarity(Matrix, Rocky IV) = 0.9803

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

Understanding Question/Q3

- **Question**:
  - Assuming you want to compute the **cosine similarity** between the Planes and Rocky IV movies, what is the sum of all the elements of the **second** vector (i.e., for Rocky IV) you will use?

- **Answers**:
  - 2, 4, 10, 13, 18

Planes	4	4	5	2		3

Rocky IV	2	2	2	4	3	

Understanding Question/Q3 Answer

- **Question**:
  - Assuming you want to compute the **cosine similarity** between the Planes and Rocky IV movies, what is the sum of all the elements of the **second** vector (i.e., for Rocky IV) you will use?
- **Answers**:
  - 2, 4, 10, 13, 18

Planes	4	4	5	2		3
Rocky IV	2	2	2	4	3	

Rocky IV = [2, 2, 2, 4] Answer = 10

Adjusted Cosine Similarity

- Drawback of cosine similarity:
  - Difference in rating scale between different users is not taken into account

- Propose new metric:

$$\text{Adjusted Cosine Similarity} = \frac{\sum_{u \in U} (A_{u,i} - R_u) \times (B_{u,i} - R_u)}{\sqrt{\sum_{u \in U} (A_{u,i} - R_u)^2} \times \sqrt{\sum_{u \in U} (B_{u,i} - R_u)^2}}$$

  - where R_u is the average of the user u's ratings

We have the similarities, now what?

- **Option #1**:
  Can proceed with weighted average method to predict missing values for ratings

- **Option #2**:
  Can sort items based on similarity and present first few of them as recommended items, ordered from highest similarity to lowest.
  - This is what Amazon.com is doing

MIND THE SIGN

Considering the sign of the similarity

- Similarity score in [0, 1]
  ➔ use simple weighted average

- Similarity score in [-1, 1]
  ➔ must use absolute value of weight in denominator
  ➔ must normalize ranking

- Normalize rankings:
  - Convert from **[1, 5]** range to **[-1, 1]** range

- De-normalize rankings
  - Convert from **[-1, 1]** to **[1, 5]** range

Normalize / de-normalize formulas

- MinRating (*MinR*) = 1
- MaxRating (*MaxR*) = 5

- Normalize:

$$NR_{u,i} = \frac{2(R_{u,i} - MinR) - (MaxR - MinR)}{(MaxR - MinR)}$$

- De-normalize:

$$R_{u,i} = \frac{1}{2}((NR_{u,i} + 1) \times (MaxR - MinR)) + MinR$$

EVALUATING QUALITY

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh

How good are the predictions?

Standard evaluation technique:

- Given a dataset
- Allocate x% to "**training**" (i.e., input to the algorithm)
  - Could be as high as 80%
- Remaining percent is "**test**" data set
- Predict values for the test data
  - Without looking at the test data
- Compare predicted value with test data value
- Aggregate differences over entire test data set

Mean Absolute Error (MAE)

- Rating p_i, prediction q_i
- Compute absolute error between p_i and q_i
- Aggregate over all predictions
- Compute the average

$$\mathrm{MAE} = \frac{1}{N} \times \sum_{i=1}^{N} | \, p_i - q_i \, |$$

Mean Absolute Error Example

	Similarity to Frank	The Matrix	Gone with the Wind	Jack and Jill	Planes	Rocky IV
Alice	0.2	**2**	5	2	4	2
Bob	0.5	**3**		1	4	2
Christine	0.1667	**4**	5	2	5	3
David	0.3333	**5**		2	2	4
Elaine	1.0	**5**	3	1		3
Frank			**3**	**1**	**3**	
Average		3.8	4	1.5	3.6	2.8
DIFF			1	0.5	0.6	
Prediction		4.196	3.536	1.318	3.583	2.833
DIFF			0.536	0.318	0.583	

Mean Absolute Error Example

Average		3.8	4	1.5	3.6	2.8
DIFF			1	0.5	0.6	
Prediction		4.196	3.536	1.318	3.583	2.833
DIFF			0.536	0.318	0.583	

$$\text{MAE (average)} = \frac{1 + 0.5 + 0.6}{3} = 0.7$$

$$\text{MAE (prediction)} = \frac{0.536 + 0.318 + 0.583}{3} = 0.479$$

Understanding Question/Q4

- **Question**:
  - Given the table of the handout, what is the Mean Absolute Error for the predicted ratings for David (compared to his actual ratings)?

- **Possible Answers**:
  - Fill-in

Understanding Question/Q4 Answer

- **Question**:
  - Given the table of the handout, what is the Mean Absolute Error for the predicted ratings for David (compared to his actual ratings)?

- **Answer**:
  - (Actual, Predicted): (5, 4.5) and (2, 2) and (2, 3) and (4, 3.5).
    So we have 1/4 * (0.5 + 0 + 1 + 0.5) = **0.5**

Root Mean Squared Error (RMSE)

- Rating p_i, prediction q_i
- Compute error between p_i and q_i and square value
- Aggregate over all predictions and divide by N
- Compute square root

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (p_i - q_i)^2}{N}}$$

© 2016 – 2020 Alexandros Labrinidis, University of Pittsburgh