Introducing the Recommender

Calculating Weighted Averages

Beyond Average Ratings

- Average ratings may be flawed

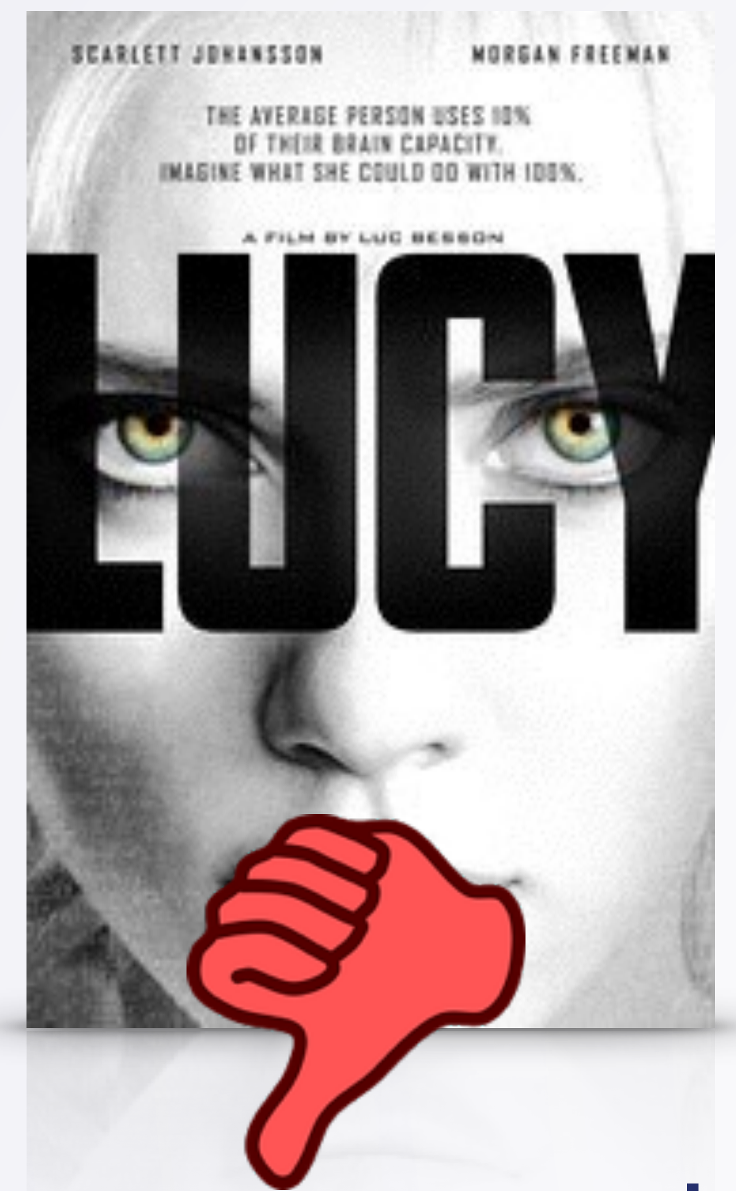
	Mission Impossible:	The Martian	Pitch Perfect 2	Star Wars: The Force Awakens
Chris	7	7	7	7
Sam	4	8	8	4
Morgan	9	8	7	7
Jessie	6	3	4	8
total	26	26	26	26

Beyond Average Ratings

- Average ratings may be flawed
  - Treat each rater equally, but I may be more like Morgan than Jessie in what I like to watch
- Collaborative Filtering Averages
  - Weight raters differently

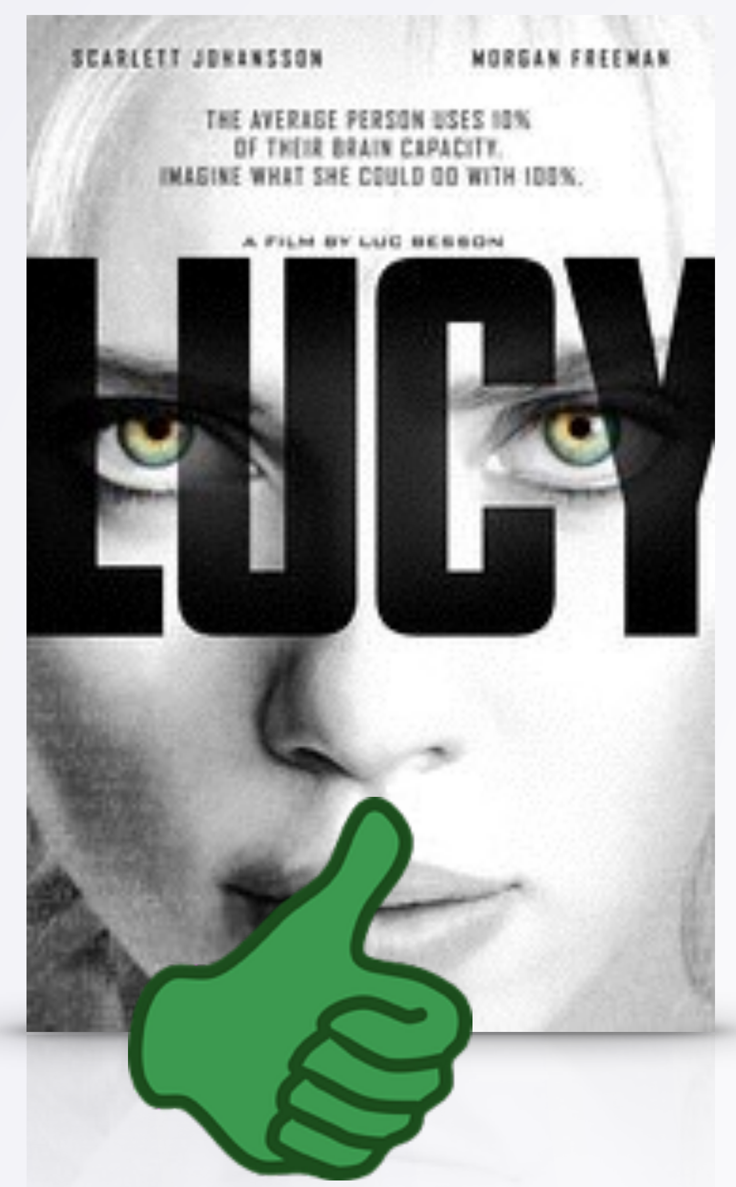
	Mission Impossible:	*The Martian*	*Pitch Perfect 2*	*Star Wars: The Force Awakens*
Chris	*7*	*7*	*7*	*7*
Sam	*4*	*8*	*8*	*4*
Morgan	*9*	*8*	*7*	*7*
Jessie	*6*	*3*	*4*	*8*
total	*26*	*26*	*26*	*26*

Collaborative Recommendations


SCARLETT JOHANSSON
MORGAN FREEMAN
THE AVERAGE PERSON USES 10% OF THEIR BRAIN CAPACITY. IMAGINE WHAT SHE COULD DO WITH 100%.
A FILM BY LUC BESSON
LUCY

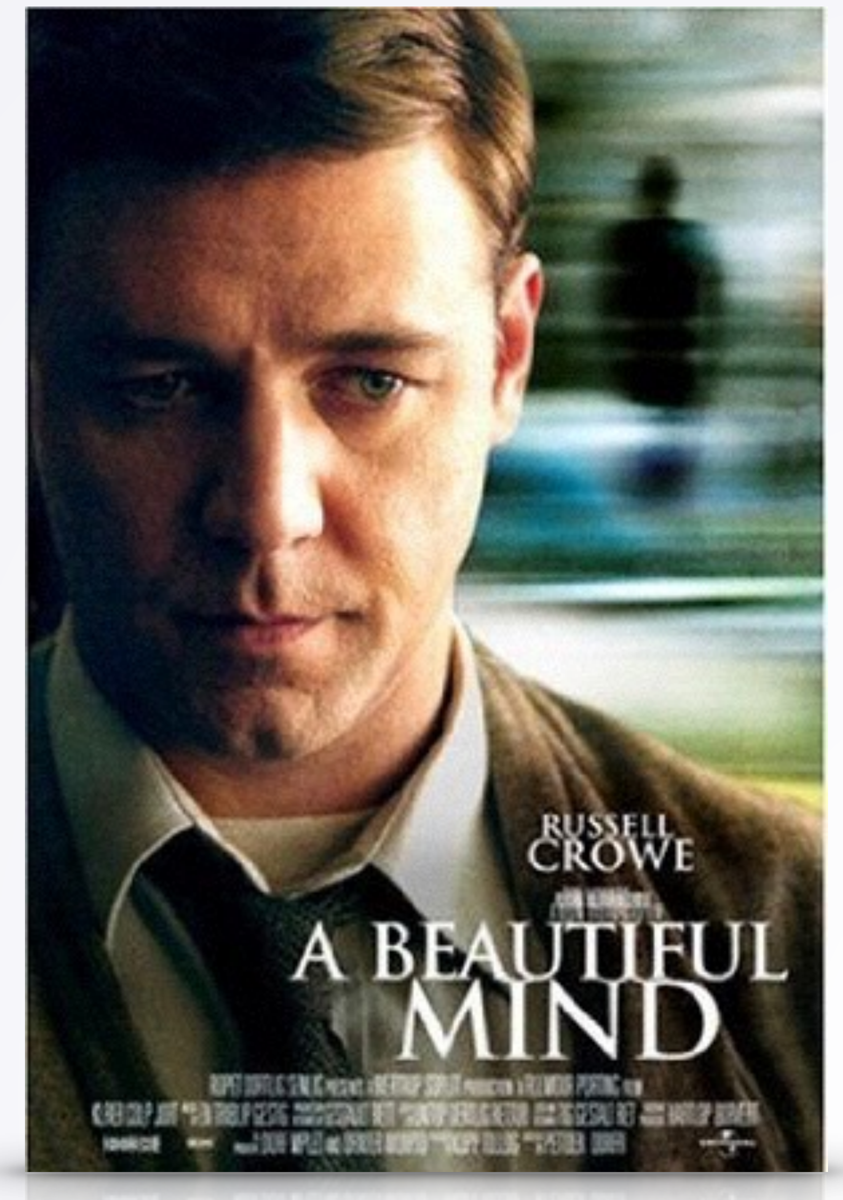
- Find raters more like me, use their ratings
    - Weight recommendations from Chris if Chris likes movies I like, dislikes those I don't.

Collaborative Recommendations


SCARLETT JOHANSSON
MORGAN FREEMAN
THE AVERAGE PERSON USES 10% OF THEIR BRAIN CAPACITY. IMAGINE WHAT SHE COULD DO WITH 100%.
A FILM BY LUC BESSON
LUCY

- Find raters more like me, use their ratings
  - Weight recommendations from Chris if Chris likes movies I like, dislikes those I don't.
  - Don't weight Sam's recommendations if we share little in common

Collaborative Recommendations



- Value Chris' ratings more than Sam's!
    - Chris likes A Beautiful Mind, I haven't seen it!!
    - Time to think about seeing the movie

Weighted Averages

	The Fly	*Spider-Man*	*Butterfly Effect*	*Beetlejuice*
20, Chris	*8*	*5*	*7*	
10, Sam		*7*	*8*	*9*
5, Morgan	*6*	*6*	*6*	*6*
Average	*14/2 = 7*	*18/3 = 6*	*21/3 = 7*	*15/2 = 7.5*

- Conceptually two changes to averages
  - Only count N raters "close" to me
  - Weight each rater's ratings by closeness to me
- Which movie has the highest average?

Weighted Averages

	The Fly	*Spider-Man*	*Butterfly Effect*	*Beetlejuice*
20, Chris	*8*	*5*	*7*	
10, Sam		*7*	*8*	*9*
5, Morgan	*6*	*6*	*6*	*6*
Average	*14/2 = 7*	*18/3 = 6*	*21/3 = 7*	*15/2 = 7.5*

- Conceptually two changes to averages
    - Only count N raters "close" to me
    - Weight each rater's ratings by closeness to me
- Which movie has the highest average?

Weighted Averages

	The Fly	*Spider-Man*	*Butterfly Effect*	*Beetlejuice*
20, Chris	*8*	*5*	*7*	
10, Sam		*7*	*8*	*9*
5, Morgan	*6*	*6*	*6*	*6*
Average	*14/2 = 7*	*18/3 = 6*	*21/3 = 7*	*15/2 = 7.5*

- Conceptually two changes to averages
  - Only count N raters "close" to me
  - Weight each rater's ratings by closeness to me
- Which movie has the highest average?

Weighted Averages

	The Fly	*Spider-Man*	*Butterfly Effect*	*Beetlejuice*
20, Chris	*8*	*5*	*7*	
10, Sam		*7*	*8*	*9*
5, Morgan	*6*	*6*	*6*	*6*
Average	*14/2 = 7*	*18/3 = 6*	*21/3 = 7*	*15/2 = 7.5*

- Conceptually two changes to averages
  - Only count N raters "close" to me
  - Weight each rater's ratings by closeness to me
- Which movie has the highest average?

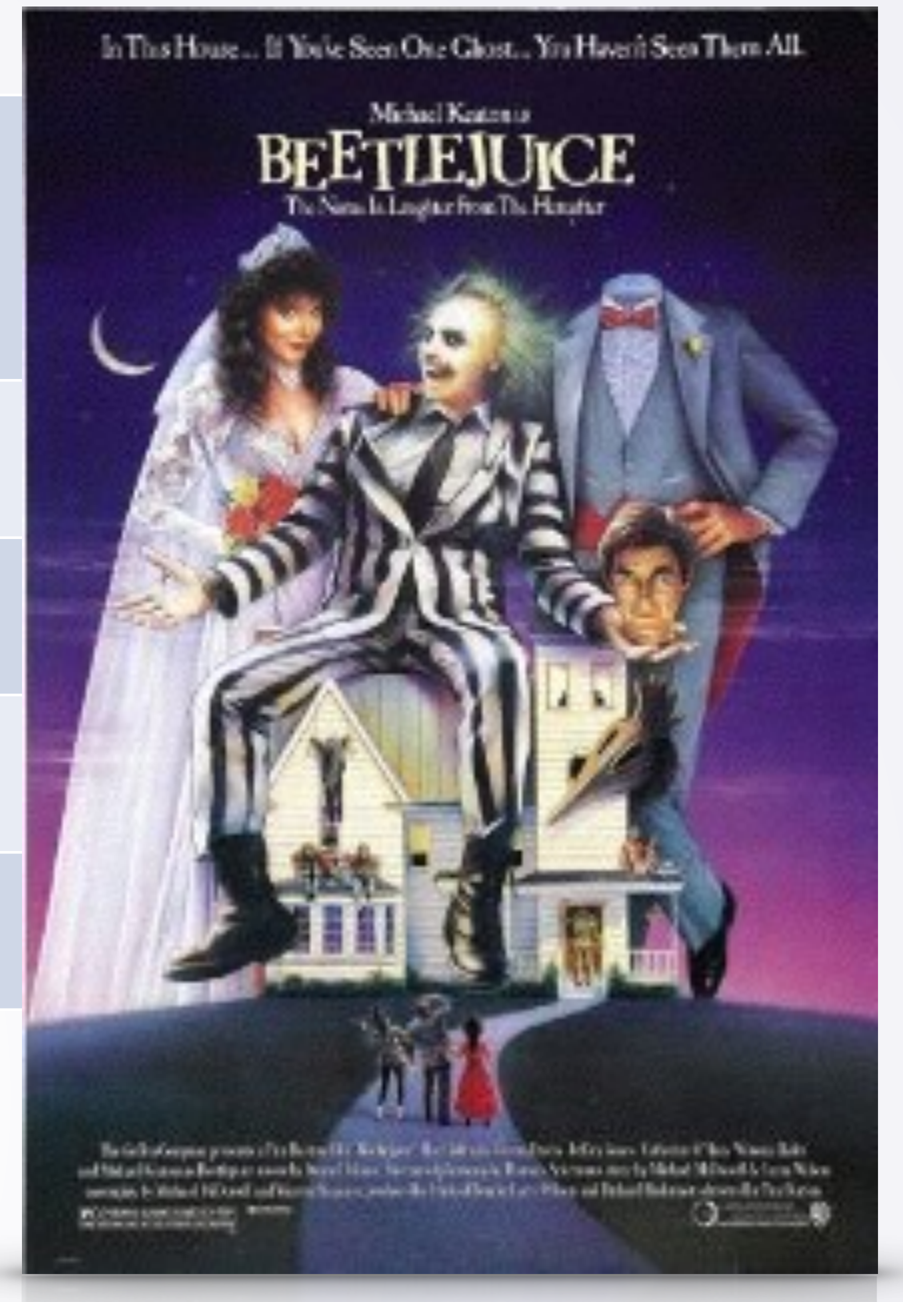
Weighted Averages

	The Fly	*Spider-Man*	*Butterfly Effect*	*Beetlejuice*
20, Chris	*8*	*5*	*7*	
10, Sam		*7*	*8*	*9*
5, Morgan	*6*	*6*	*6*	*6*
Average	*14/2 = 7*	*18/3 = 6*	*21/3 = 7*	*15/2 = 7.5*

- Conceptually two changes to averages
  - Only count N raters "close" to me
  - Weight each rater's ratings by closeness to me
- Which movie has the highest average?

Weighted Averages

	The Fly	*Spider-Man*	*Butterfly Effect*
20, Chris	*8*	*5*	*7*
10, Sam		*7*	*8*
5, Morgan	*6*	*6*	*6*
Average	*14/2 = 7*	*18/3 = 6*	*21/3 = 7*


In This House... If You've Seen One Ghost... You Haven't Seen Them All.
Michael Keaton is
BEETLEJUICE
The Name In Laughter From The Hereafter

- Conceptually two changes to averages
  - Only count N raters "close" to me
  - Weight each rater's ratings by closeness to me
- Which movie has the highest average?
- Chris is more like me than Morgan, value more!

Weighted Averages in Detail

	The Fly	*Spider-Man*	*Butterfly Effect*	*Beetlejuice*
20, Chris	*8*	*5*	*7*	
10, Sam		*7*	*8*	*9*
5, Morgan	*6*	*6*	*6*	*6*
Average	*14/2 = 7*	*18/3 = 6*	*21/3 = 7*	*15/2 = 7.5*
Weighted	*160+30* *95.0*	*100+70+30* *66.67*	*140+80+30* *83.3*	*90+30* *60.0*

- Use "closeness" weight for each rater
  - Multiply ratings by the corresponding weight
  - Movies not rated by everyone, what to watch?

Weighted Averages in Detail

	The Fly	*Spider-Man*	*Butterfly Effect*	*Beetlejuice*
20, Chris	*8*	*5*	*7*	
10, Sam		*7*	*8*	*9*
5, Morgan	*6*	*6*	*6*	*6*
Average	*14/2 = 7*	*18/3 = 6*	*21/3 = 7*	*15/2 = 7.5*
Weighted	*160+30* *95.0* ←	*100+70+30* *66.67*	*140+80+30* *83.3*	*90+30* *60.0*

- Use "closeness" weight for each rater
  - Multiply ratings by the corresponding weight
  - Movies not rated by everyone, what to watch?

Weighted Averages in Detail

	The Fly	*Spider-Man*	*Butterfly Effect*	*Beetlejuice*
20, Chris	*8*	*5*	*7*	
10, Sam		*7*	*8*	*9*
5, Morgan	*6*	*6*	*6*	*6*
Average	*14/2 = 7*	*18/3 = 6*	*21/3 = 7*	*15/2 = 7.5*
Weighted	*160+30* *95.0*	*100+70+30* *66.67* ←	*140+80+30* *83.3*	*90+30* *60.0*

- Use "closeness" weight for each rater
  - Multiply ratings by the corresponding weight
  - Movies not rated by everyone, what to watch?

Weighted Averages in Detail

	The Fly	*Spider-Man*	*Butterfly Effect*	*Beetlejuice*
20, Chris	*8*	*5*	*7*	
10, Sam		*7*	*8*	*9*
5, Morgan	*6*	*6*	*6*	*6*
Average	*14/2 = 7*	*18/3 = 6*	*21/3 = 7*	*15/2 = 7.5*
Weighted	*160+30* *95.0*	*100+70+30* *66.67*	*140+80+30* *83.3* ←	*90+30* *60.0*

- Use "closeness" weight for each rater
  - Multiply ratings by the corresponding weight
  - Movies not rated by everyone, what to watch?

Duke
UNIVERSITY

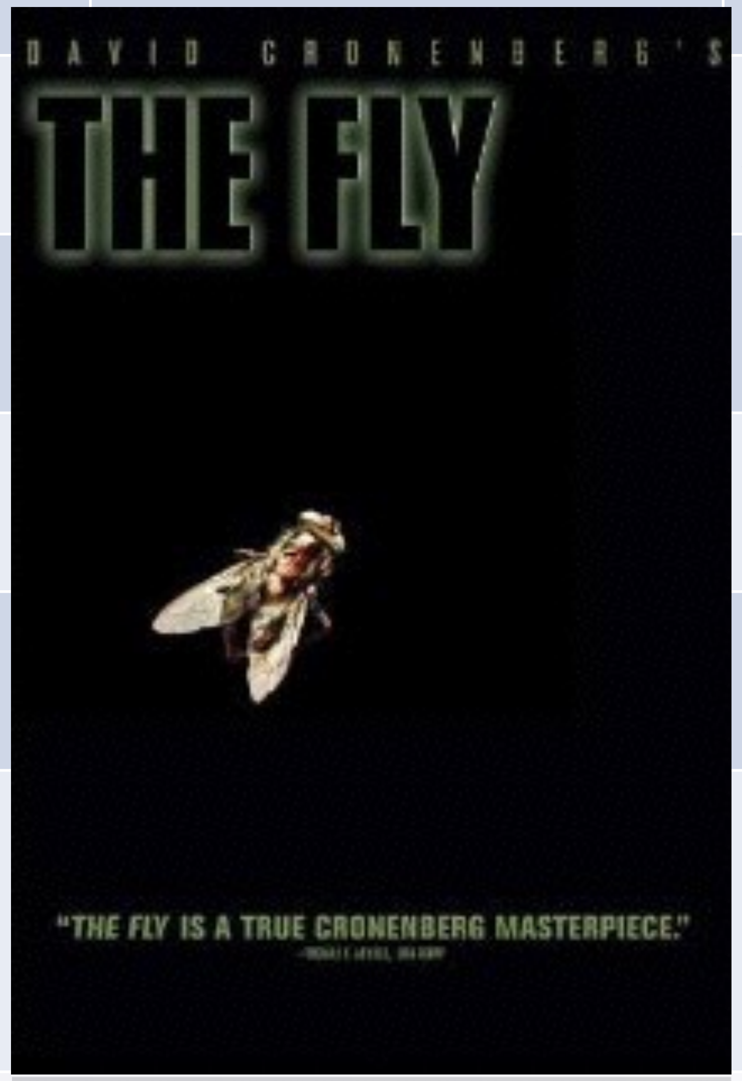
Weighted Averages in Detail

	The Fly	Spider-Man	Butterfly Effect	Beetlejuice
20, Chris	8	5	7	
10, Sam		7	8	9
5, Morgan	6	6	6	6
Average	14/2 = 7	18/3 = 6	21/3 = 7	15/2 = 7.5
Weighted	160+30 95.0	100+70+30 66.67	140+80+30 83.3	90+30 60.0 ←

- Use "closeness" weight for each rater
  - Multiply ratings by the corresponding weight
  - Movies not rated by everyone, what to watch?

Duke UNIVERSITY

Weighted Averages in Detail



	The Fly	Spider-Man	Butterfly Effect	Beetlejuice
20, Chris		5	7	
10, Sam		7	8	9
5, Morgan		6	6	6
Average		18/3 = 6	21/3 = 7	15/2 = 7.5
Weighted		100+70+30 66.67	140+80+30 83.3	90+30 60.0 ←

- Use "closeness" weight for each rater
  - Multiply ratings by the corresponding weight
  - Movies not rated by everyone, what to watch?

Calculating Closeness

- Each rater represented by vector of ratings
  - Sam **`[0,5,2,7,0,8,1]`**
  - Chris **`[6,7,5,0,0,0,9]`**
  - Me **`[2,6,0,4,5,4,6]`**
- Sam and me: **`5*6+7*4+8*4+1*6 = 96`**

Calculating Closeness

- Each rater represented by vector of ratings
  - Sam `[0,5,2,7,0,8,1]`
  - Chris `[6,7,5,0,0,0,9]`
  - Me `[2,6,0,4,5,4,6]`
- Sam and me: **`5*6+7*4+8*4+1*6 = 96`**

Calculating Closeness

- Each rater represented by vector of ratings
  - Sam `[0,5,2,7,0,8,1]`
  - Chris `[6,7,5,0,0,0,9]`
  - Me `[2,6,0,4,5,4,6]`
- Sam and me: **`5*6+7*4+8*4+1*6 = 96`**

Calculating Closeness

- Each rater represented by vector of ratings
  - Sam `[0,5,2,7,0,8,1]`
  - Chris `[6,7,5,0,0,0,9]`
  - Me `[2,6,0,4,5,4,6]`
- Sam and me: **`5*6+7*4+8*4+1*6 = 96`**

Calculating Closeness

- Each rater represented by vector of ratings
  - Sam `[0,5,2,7,0,8,1]`
  - Chris `[6,7,5,0,0,0,9]`
  - Me `[2,6,0,4,5,4,6]`
- Sam and me: **`5*6+7*4+8*4+1*6 = 96`**

Calculating Closeness

- Each rater represented by vector of ratings
  - Sam `[0,5,2,7,0,8,1]`
  - Chris `[6,7,5,0,0,0,9]`
  - Me `[2,6,0,4,5,4,6]`
- Sam and me: **`5*6+7*4+8*4+1*6 = 96`**
- Chris and me: **`6*2+7*6+9*6 = 108`**
- This is dot product, measure of closeness in mathematical vector space
  - Sum of rating products for each movie rated

Duke UNIVERSITY

What Does "Close" Mean?

- To represent 1-10 scale, we adjust ratings
  - Ratings of 1 and 2 compared to 8 and 9
    - (1-5)*(2-5) = 12, and (8-5)*(9-5) = 12
  - Sam `[ *,0,-3, 2,*, 3,-4]` `[0,5,2,7,0,8,1]`
  - Chris `[ 1,2, 0, *,*, *, 4]` `[6,7,5,0,0,0,9]`
  - Me `[-3,1, *,-1,0,-1, 1]` `[2,6,0,4,5,4,6]`
- Sam and me: `0*1 + 2*-1 + 3*-1 + -4*1 = -9`

What Does "Close" Mean?

- To represent 1-10 scale, we adjust ratings
  - Ratings of 1 and 2 compared to 8 and 9
    - (1-5)*(2-5) = 12, and (8-5)*(9-5) = 12
  - Sam `[ *,0,-3, 2,*, 3,-4]` `[0,5,2,7,0,8,1]`
  - Chris `[ 1,2, 0, *,*, *, 4]` `[6,7,5,0,0,0,9]`
  - Me `[-3,1, *,-1,0,-1, 1]` `[2,6,0,4,5,4,6]`
- Sam and me: `0*1 + 2*-1 + 3*-1 + -4*1 = -9`

What Does "Close" Mean?

- To represent 1-10 scale, we adjust ratings
  - Ratings of 1 and 2 compared to 8 and 9
    - (1-5)*(2-5) = 12, and (8-5)*(9-5) = 12
  - Sam `[ *,0,-3, 2,*, 3,-4]` `[0,5,2,7,0,8,1]`
  - Chris `[ 1,2, 0, *,*, *, 4]` `[6,7,5,0,0,0,9]`
  - Me `[-3,1, *,-1,0,-1, 1]` `[2,6,0,4,5,4,6]`
- Sam and me: `0*1 + 2*-1 + 3*-1 + -4*1 = -9`

What Does "Close" Mean?

- To represent 1-10 scale, we adjust ratings
  - Ratings of 1 and 2 compared to 8 and 9
    - (1-5)*(2-5) = 12, and (8-5)*(9-5) = 12
  - Sam `[ *,0,-3, 2,*, 3,-4]` `[0,5,2,7,0,8,1]`
  - Chris `[ 1,2, 0, *,*, *, 4]` `[6,7,5,0,0,0,9]`
  - Me `[-3,1, *,-1,0,-1, 1]` `[2,6,0,4,5,4,6]`
- Sam and me: `0*1 + 2*-1 + 3*-1 + -4*1 = -9`

What Does "Close" Mean?

- To represent 1-10 scale, we adjust ratings
  - Ratings of 1 and 2 compared to 8 and 9
    - (1-5)*(2-5) = 12, and (8-5)*(9-5) = 12
  - Sam `[ *,0,-3, 2,*, 3,-4]` `[0,5,2,7,0,8,1]`
  - Chris `[ 1,2, 0, *,*, *, 4]` `[6,7,5,0,0,0,9]`
  - Me `[-3,1, *,-1,0,-1, 1]` `[2,6,0,4,5,4,6]`
- Sam and me: `0*1 + 2*-1 + 3*-1 + -4*1 = -9`

What Does "Close" Mean?

- To represent 1-10 scale, we adjust ratings
    - Ratings of 1 and 2 compared to 8 and 9
        - (1-5)*(2-5) = 12, and (8-5)*(9-5) = 12
    - Sam `[ *,0,-3, 2,*, 3,-4]` `[0,5,2,7,0,8,1]`
    - Chris `[ 1,2, 0, *,*, *, 4]` `[6,7,5,0,0,0,9]`
    - Me `[-3,1, *,-1,0,-1, 1]` `[2,6,0,4,5,4,6]`
- Sam and me: `0*1 + 2*-1 + 3*-1 + -4*1 = -9`
- Chris and me: `1*-3 + 2*1 + 4*1 = 3`

What Does "Close" Mean?

- To represent 1-10 scale, we adjust ratings
  - Ratings of 1 and 2 compared to 8 and 9
    - (1-5)*(2-5) = 12, and (8-5)*(9-5) = 12
  - Sam `[ *,0,-3, 2,*, 3,-4]` `[0,5,2,7,0,8,1]`
  - Chris `[ 1,2, 0, *,*, *, 4]` `[6,7,5,0,0,0,9]`
  - Me `[-3,1, *,-1,0,-1, 1]` `[2,6,0,4,5,4,6]`
- Sam and me: `0*1 + 2*-1 + 3*-1 + -4*1 = -9`
- Chris and me: `1*-3 + 2*1 + 4*1 = 3`
- Standard: center by subtracting middle rating

Closeness with getSimilarities

- Find raters near me, call **`getSimilarities`**

```
private ArrayList<Rating> getSimilarities(String id) {
     ArrayList<Rating> list = new ArrayList<Rating>();
     Rater me = RaterDatabase.getRater(id);
     for(Rater r : RaterDatabase.getRaters()) {
       // add dot_product(r,me) to list if r != me
     }
     Collections.sort(list, Collections.reverseOrder());
     return list;
    }
```

Duke UNIVERSITY

Closeness with getSimilarities

- Find raters near me, call **`getSimilarities`**
  - RaterDatabase: access to Raters

```
private ArrayList<Rating> getSimilarities(String id) {
     ArrayList<Rating> list = new ArrayList<Rating>();
     Rater me = RaterDatabase.getRater(id);
     for(Rater r : RaterDatabase.getRaters()) {
       // add dot_product(r,me) to list if r != me
     }
     Collections.sort(list, Collections.reverseOrder());
     return list;
    }
```

Closeness with getSimilarities

- Find raters near me, call **`getSimilarities`**
    - RaterDatabase: access to Raters

```
private ArrayList<Rating> getSimilarities(String id) {
     ArrayList<Rating> list = new ArrayList<Rating>();
     Rater me = RaterDatabase.getRater(id);
     for(Rater r : RaterDatabase.getRaters()) {
       // add dot_product(r,me) to list if r != me
     }
     Collections.sort(list, Collections.reverseOrder());
     return list;
    }
```

Closeness with getSimilarities

- Find raters near me, call **`getSimilarities`**
  - RaterDatabase: access to Raters
  - Calculate dot product between me and r

```
private ArrayList<Rating> getSimilarities(String id) {
     ArrayList<Rating> list = new ArrayList<Rating>();
     Rater me = RaterDatabase.getRater(id);
     for(Rater r : RaterDatabase.getRaters()) {
      // add dot_product(r,me) to list if r != me
     }
     Collections.sort(list, Collections.reverseOrder());
     return list;
    }
```

Closeness with getSimilarities

- Find raters near me, call **`getSimilarities`**
  - RaterDatabase: access to Raters
  - Calculate dot product between me and r
  - List of raters sorted in reverse order

```
private ArrayList<Rating> getSimilarities(String id) {
     ArrayList<Rating> list = new ArrayList<Rating>();
     Rater me = RaterDatabase.getRater(id);
     for(Rater r : RaterDatabase.getRaters()) {
       // add dot_product(r,me) to list if r != me
     }
     Collections.sort(list, Collections.reverseOrder());
     return list;
    }
```

Calculating Weighted Averages

- Similar to **`getAverages`**, specific to a Rater

```
public ArrayList<Rating>
getRecommendations(String id, int numRaters) {
    ArrayList<Rating> list = getSimilarities(id);
    ArrayList<Rating> ret = new ArrayList<Rating>();
    for(String movieID : // get movies) {
        for(int k=0; k < …; k++) {
            Rating r = list.get(k);
            // use Rater id and weight in r
            // to update running totals
        }
        // add Rating for movieID to ret
    }
    return ret; // sort first
}
```

Calculating Weighted Averages

- Similar to **getAverages**, specific to a Rater

```
public ArrayList<Rating>
getRecommendations(String id, int numRaters) {
    ArrayList<Rating> list = getSimilarities(id);
    ArrayList<Rating> ret = new ArrayList<Rating>();
    for(String movieID : // get movies) {
        for(int k=0; k < …; k++) {
            Rating r = list.get(k);
            // use Rater id and weight in r
            // to update running totals
        }
        // add Rating for movieID to ret
    }
    return ret; // sort first
}
```

Calculating Weighted Averages

- Similar to **`getAverages`**, specific to a Rater

```
public ArrayList<Rating>
getRecommendations(String id, int numRaters) {
    ArrayList<Rating> list = getSimilarities(id);
    ArrayList<Rating> ret = new ArrayList<Rating>();
    for(String movieID : // get movies) {
        for(int k=0; k < …; k++) {
            Rating r = list.get(k);
            // use Rater id and weight in r
            // to update running totals
        }
        // add Rating for movieID to ret
    }
    return ret; // sort first
}
```

Calculating Weighted Averages

- Similar to **getAverages**, specific to a Rater

```
public ArrayList<Rating>
getRecommendations(String id, int numRaters) {
    ArrayList<Rating> list = getSimilarities(id);
    ArrayList<Rating> ret = new ArrayList<Rating>();
    for(String movieID : // get movies) {
        for(int k=0; k < …; k++) {
            Rating r = list.get(k);
            // use Rater id and weight in r
            // to update running totals
        }
        // add Rating for movieID to ret
    }
    return ret; // sort first
}
```

Calculating Weighted Averages

- Similar to **`getAverages`**, specific to a Rater

```
public ArrayList<Rating>
getRecommendations(String id, int numRaters) {
    ArrayList<Rating> list = getSimilarities(id);
    ArrayList<Rating> ret = new ArrayList<Rating>();
    for(String movieID : // get movies) {
        for(int k=0; k < …; k++) {
            Rating r = list.get(k);
            // use Rater id and weight in r
            // to update running totals
        }
        // add Rating for movieID to ret
    }
    return ret; // sort first
}
```

Presenting Recommendations

- Display movies already seen?

```
0  332.43   Rush
1  327.67   Interstellar
2  324.14   Whiplash
3  323.29   Gone Girl
4  321.17   The Fault in Our Stars
5  320.00   The Judge
6  318.29   Gravity
7  312.83   Dallas Buyers Club
8  306.17   The Secret Life of Walter Mitty
9  305.50   The Grand Budapest Hotel
10 304.14  *** Mission: Impossible - Rogue Nation
11 301.83   The Imitation Game
12 299.86   Birdman or (The Unexpected Virtue of Ignorance)
13 291.22  *** American Hustle
14 290.25  *** Kingsman: The Secret Service
```

Presenting Recommendations

- Display movies already seen?
  - Could help calibrate recommendations

```
0  332.43   Rush
1  327.67   Interstellar
2  324.14   Whiplash
3  323.29   Gone Girl
4  321.17   The Fault in Our Stars
5  320.00   The Judge
6  318.29   Gravity
7  312.83   Dallas Buyers Club
8  306.17   The Secret Life of Walter Mitty
9  305.50   The Grand Budapest Hotel
10 304.14  *** Mission: Impossible - Rogue Nation
11 301.83   The Imitation Game
12 299.86   Birdman or (The Unexpected Virtue of Ignorance)
13 291.22  *** American Hustle
14 290.25  *** Kingsman: The Secret Service
```

Presenting Recommendations

- Display movies already seen?
  - Could help calibrate recommendations
- Print top 15? Print weighted average?

```
0  332.43   Rush
1  327.67   Interstellar
2  324.14   Whiplash
3  323.29   Gone Girl
4  321.17   The Fault in Our Stars
5  320.00   The Judge
6  318.29   Gravity
7  312.83   Dallas Buyers Club
8  306.17   The Secret Life of Walter Mitty
9  305.50   The Grand Budapest Hotel
10 304.14  *** Mission: Impossible - Rogue Nation
11 301.83   The Imitation Game
12 299.86   Birdman or (The Unexpected Virtue of Ignorance)
13 291.22  *** American Hustle
14 290.25  *** Kingsman: The Secret Service
```

Duke
UNIVERSITY

Presenting Recommendations

- Display movies already seen?
    - Could help calibrate recommendations
- Print top 15? Print weighted average?
    - Movie information? HTML?

```
0  332.43   Rush
1  327.67   Interstellar
2  324.14   Whiplash
3  323.29   Gone Girl
4  321.17   The Fault in Our Stars
5  320.00   The Judge
6  318.29   Gravity
7  312.83   Dallas Buyers Club
8  306.17   The Secret Life of Walter Mitty
9  305.50   The Grand Budapest Hotel
10 304.14  *** Mission: Impossible - Rogue Nation
11 301.83   The Imitation Game
12 299.86   Birdman or (The Unexpected Virtue of Ignorance)
13 291.22  *** American Hustle
14 290.25  *** Kingsman: The Secret Service
```