

COLLABORATIVE FITLERING

Rubi's Motivation for CF

- Find a *PhD* problem
- Find “real life” *PhD* problem
- Find an interesting *PhD* problem
- Make Money!



Recommender Systems

Basic implementations:

- Most popular / cheap / etc.
- New items
- Can they go shopping together?



Live Demonstrations

- ◎ Amazon

- ◎ Netflix

XBOX360 usage:

<http://www.youtube.com/watch?v=litD0hdOCvA>

Netflix Example

The screenshot shows the Netflix interface for the movie 'Shadow of a Doubt' (1947). The page is divided into several sections: a top navigation bar, a main header with the movie title and year, a synopsis, a list of recommended movies, and a section for user reviews. Red circles and lines highlight specific elements: the movie poster, the 'Add' button, the 'UR' (Unrated) rating, the 'DETAILS' section, the 'Screen Formats' section, the 'Subtitles' section, the 'Language and Sound' section, the 'Original Release Date' section, the 'Average of people who rate like you' section, the 'Average of 58,546 ratings' section, the 'At A Glance' section, the 'Friends' section, the 'Member Reviews' section, the 'Critic's Picks' section, and the 'More Like This' section.

Shadow of a Doubt (1947)

Master of suspense Alfred Hitchcock directed this tale about Charlie (Teresa Wright), a small-town girl consumed with finding out whether her unhinged Uncle (Joseph Cotton) is a serial killer. The arrival of detectives and a murder-infatuated neighbor (Hume Cronyn) only increase Charlie's paranoia. Tension builds as she draws closer to the truth, and in classic Hitchcock style, the film culminates in a nail-biting scene aboard a speeding train.

WHY IS THIS RECOMMENDED

- [Rear Window](#)
- [North by Northwest](#)
- [Psycho](#)
- [Chinatown](#)
- [On the Waterfront](#)
- [A Streetcar Named Desire](#)
- [Casablanca](#)
- [The Third Man](#)

Screen Formats:
Full Screen 1.33:1

Subtitles:
French

Language and Sound:
English: Dolby Digital Mono
Spanish: Dolby Digital Mono

Original Release Date:
1943

Average of people who rate like you: 4.0 stars
Average of 58,546 ratings: 3.9 stars

Reviews:

ST from IL 4
(See my other reviews...)
26 out of 30 people found this review helpful.
They say this was Hitchcock's favorite of all his films and there are cinematic elements he used here that he would come back to in other films. Like a junkball pitcher at the top of his game, Hitch gives you everything he's got in this one—humor, real life on the streets in 40's California, sideplots, close calls, mcguffins, and, of course, suspense. He is the man—but let's not forget how good Joseph Cotton and Teresa Wright are in this film.

Grizbuh
(See my other reviews...)
22 out of 25 people found this review helpful.
Shadow of a Doubt is a subtle exploration of some classical Hitchcock themes in which the sinister and murderous faces of human nature quite often exist in the

Netflix Example

The screenshot shows the Netflix interface with a red header. The main section is titled "Other Movies You Might Enjoy". It features a grid of movie cards. The first row includes "Strangers on a Train" and "The Man Who Knew Too Much". The second row includes "Rear Window", "Dial M for Murder", "Notorious", and "North by Northwest". Below the grid, there are sections for "Also directed by Alfred Hitchcock:" and "Also in Classics:". A red circle highlights the "Strangers on a Train" and "The Man Who Knew Too Much" cards. Another red circle highlights a notification box on the right that says "Rope has been added to your Queue at position 115." and a button "Move To Top Of My Queue". A third red circle highlights the "Also directed by Alfred Hitchcock:" section, which includes cards for "The 39 Steps" and "Lifeboat".

NETFLIX

Other Movies You Might Enjoy

Strangers on a Train
Special Edition

The Man Who Knew Too Much

Rope has been added to your Queue at position 115.
This movie is available now.
[Move To Top Of My Queue](#)

< [Continue Browsing](#) [Visit your Queue >](#)

Rear Window

Dial M for Murder

Notorious

North by Northwest

Also directed by Alfred Hitchcock:

The 39 Steps

Lifeboat

The Lady Vanishes

Saboteur

Also in Classics:

Netflix Prize

- Goal: Improve the accuracy of predictions about how much someone is going to love a movie by 10%
- Started at 2006 (Max until 2011)
- Prize: \$1,000,000
- September 2009 - 10.06%!! by Bellkor

Recommender Systems

- ⦿ Personalized Recommendations!!!
- ⦿ Predicts user rating →
Provide Recommendations
- ⦿ Attempt to profile user preferences
- ⦿ Model interaction between users and product

Recommender Systems

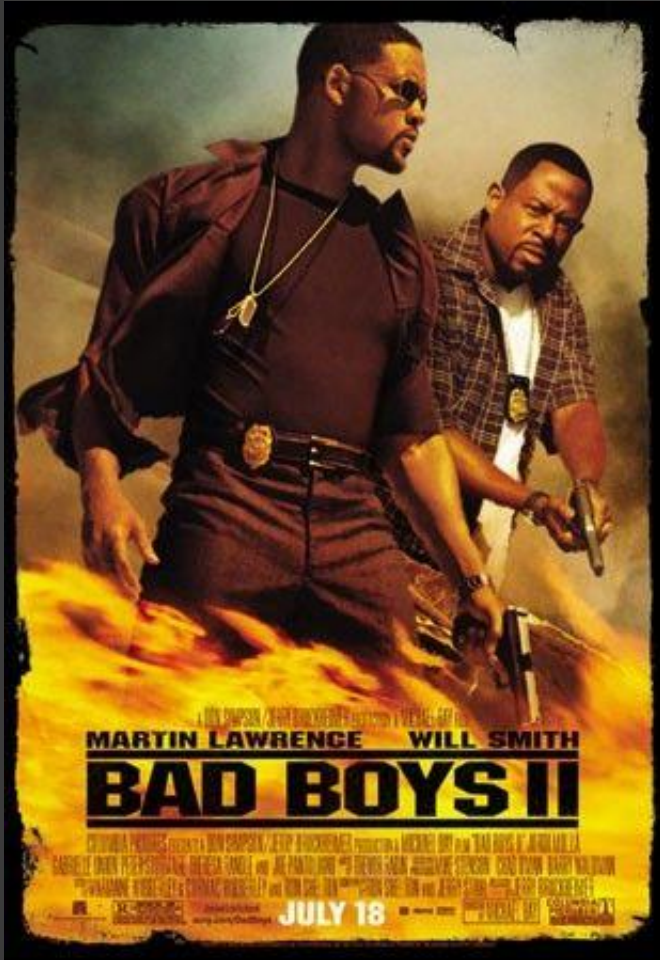
Requirements:

- ⦿ Provide good recommendations (daaaa)
- ⦿ Justify the recommendation
- ⦿ Feasible in Run-Time

Strategies

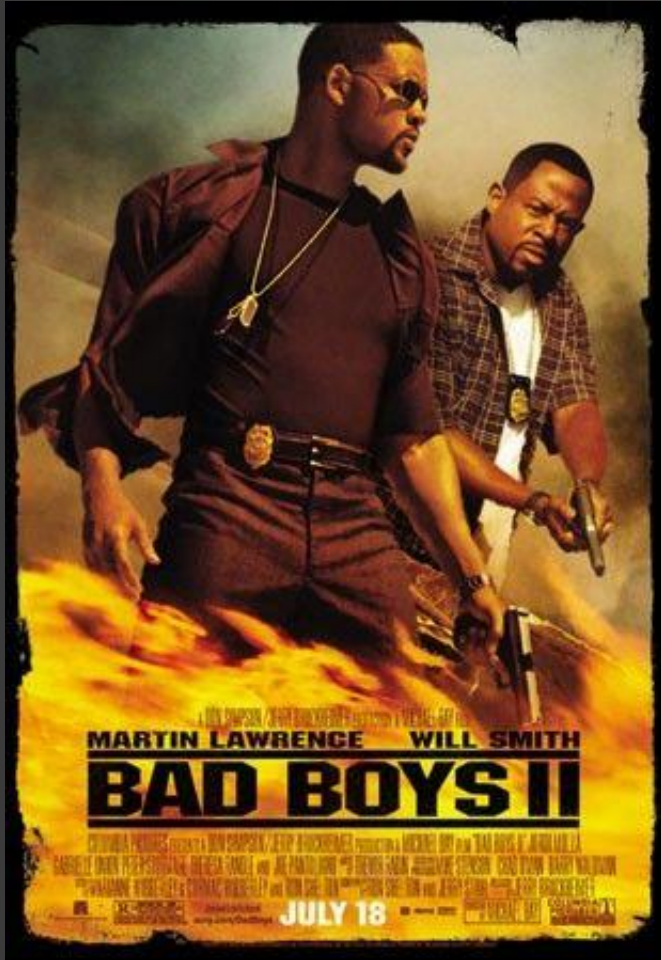
- ⦿ Content-Based
- ⦿ Collaborative Filtering (CF)

Content-Based



- **Actors:**
Will Smith, Martin...
- **Genre:**
Action / Comedy
- **Director:**
Michael Bay

Content-Based - VSM



- Domain of Features
- Describing Vector

0

1

Will Smith

0

Pamela Anderson

0

1

Michael Bay

0

1

Action

1

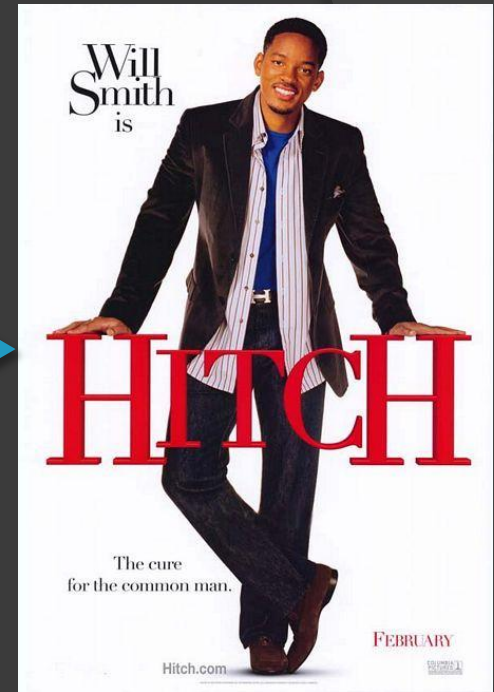
Comedy

Comparing Two Vectors

- ⦿ Calculate the angle between the vectors
- ⦿ Easier to calculate the cosine

$$\cos \Theta = \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\| \|\mathbf{v}_2\|}$$

VSM – “near” vectors



- Michael Bay
- Action

- Will Smith
- Comedy

Content-Based - Disadvantages

- ⦿ Static
- ⦿ Can't find “special” correlations
- ⦿ Requires gathering external information

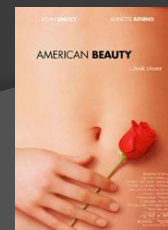
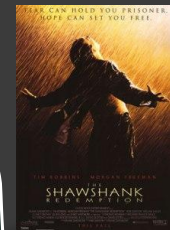
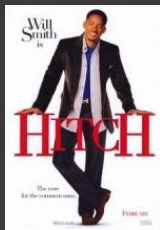
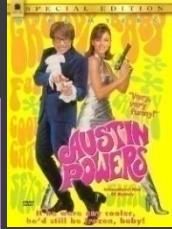
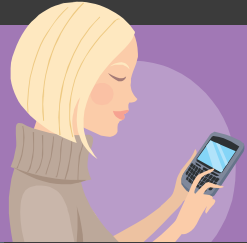
Collaborative Filtering

- ⦿ Relies just on users behavior
- ⦿ No profiles are required
- ⦿ Analyzes the relationships between users and items

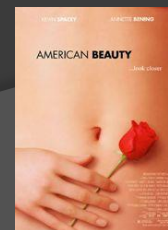
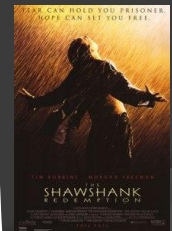
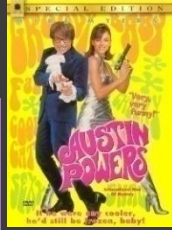
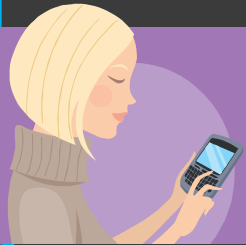
CF - Levels

- ⦿ Neighborhood Based
(local area)
- ⦿ Factorization Based
(regional area)

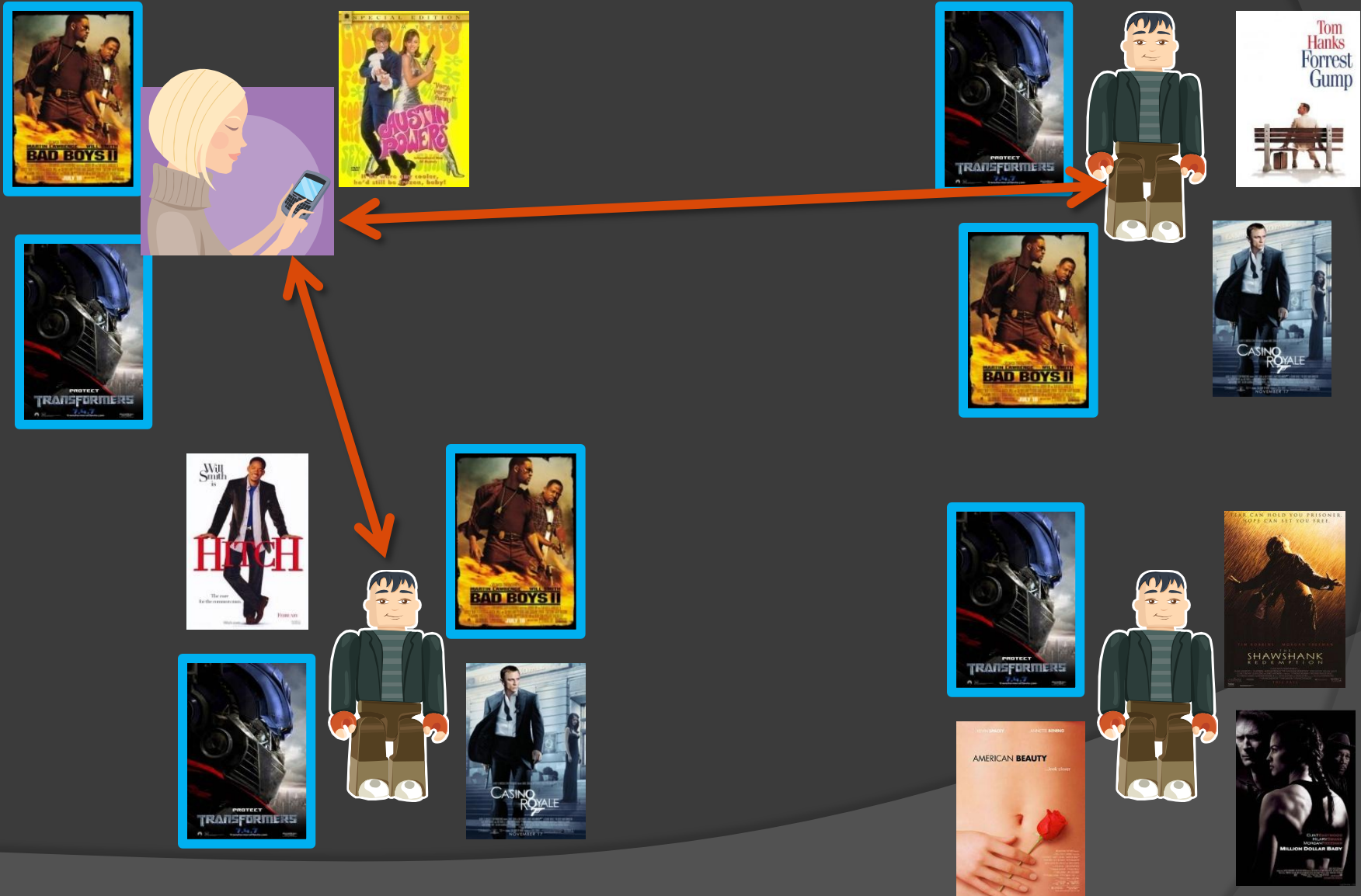
CF – Neighborhood Based



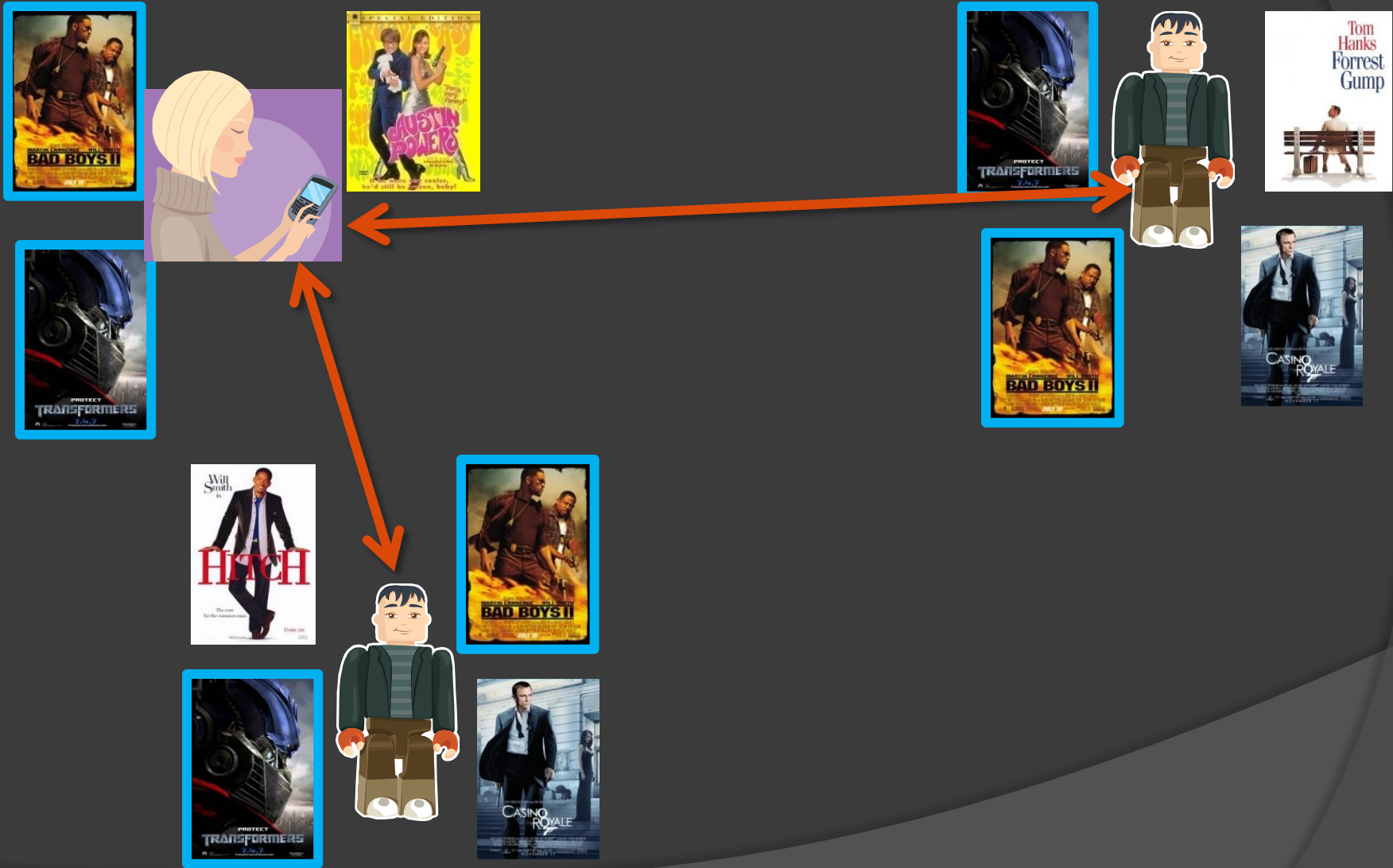
CF – Neighborhood Based



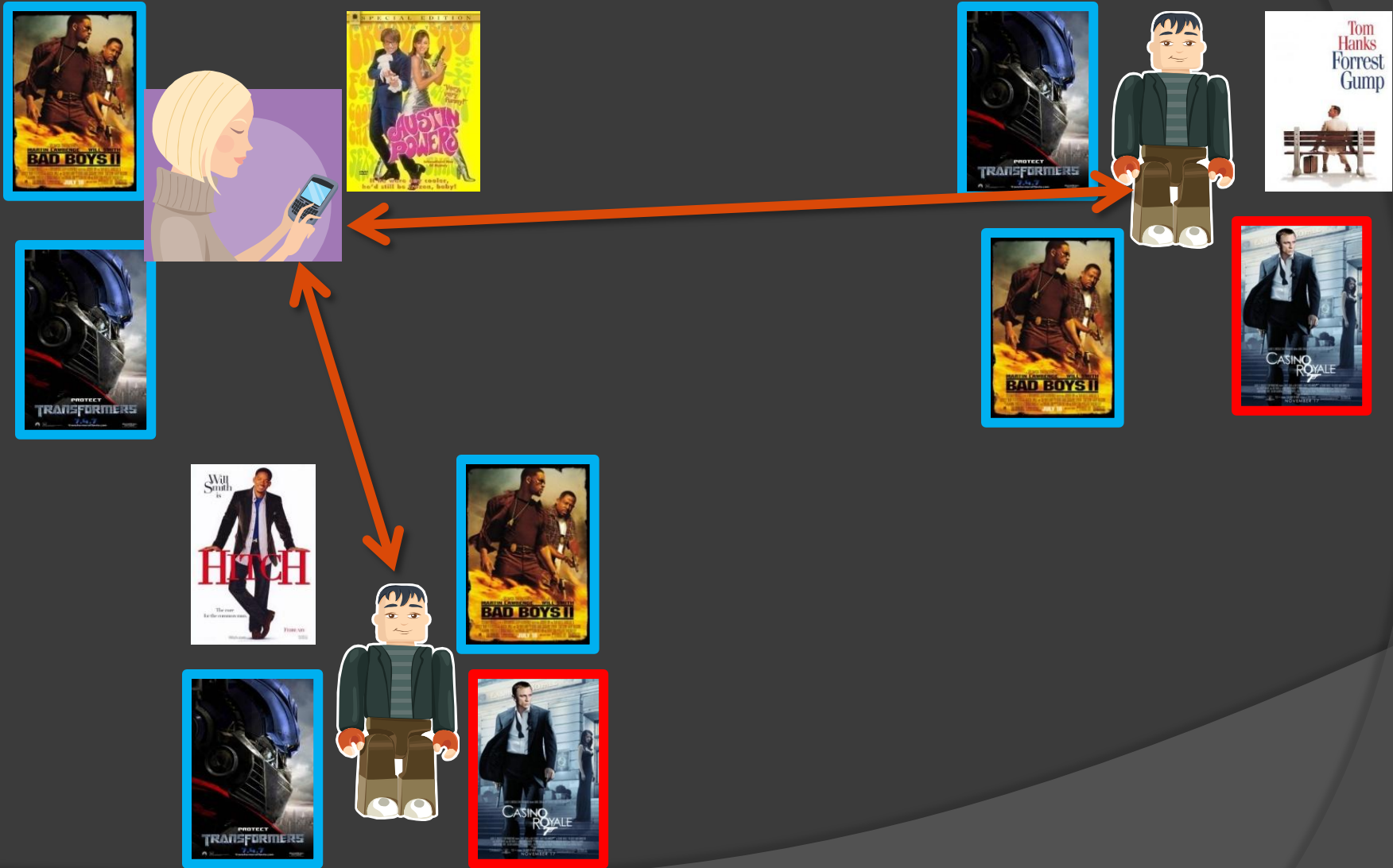
CF – Neighborhood Based



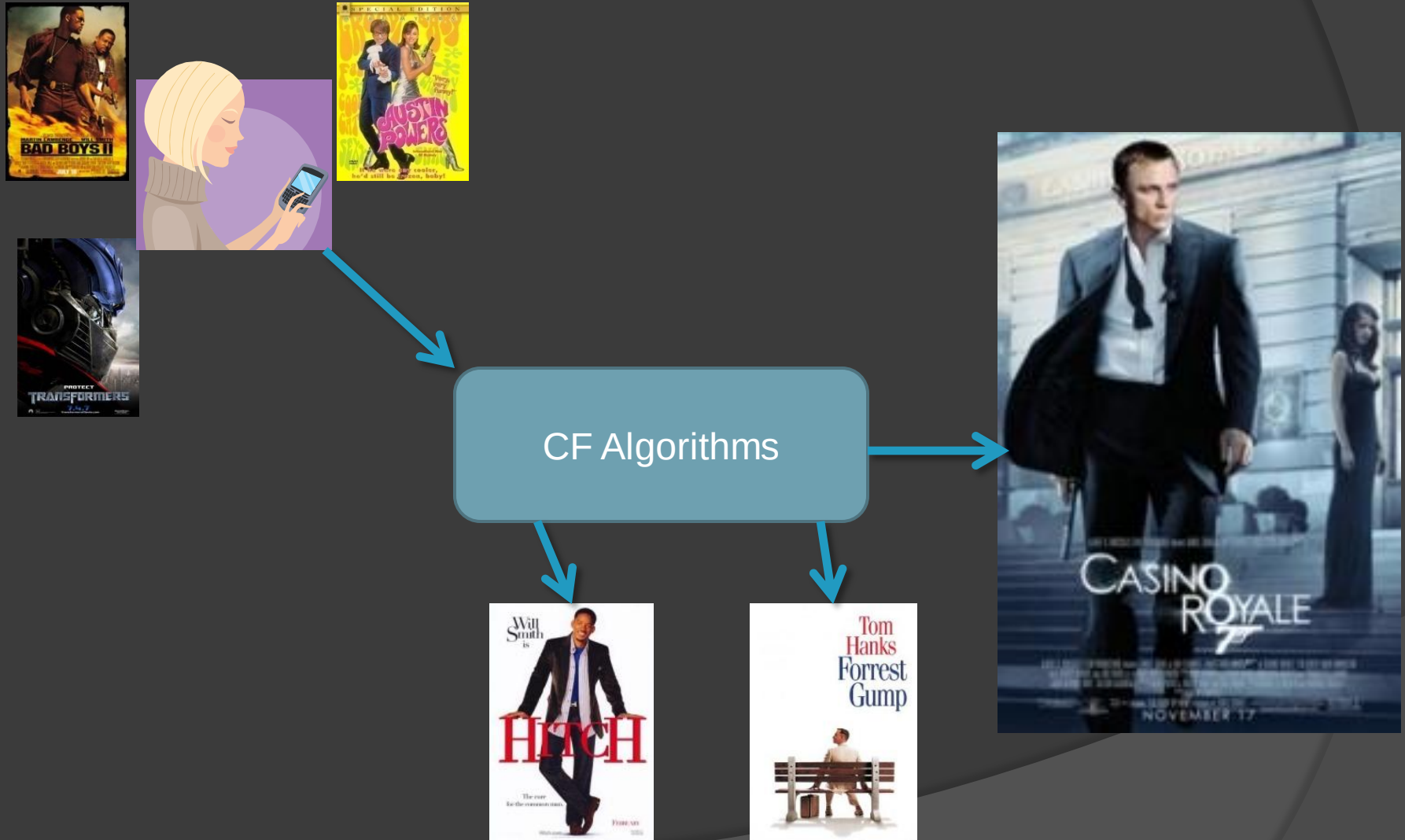
CF – Neighborhood Based



CF – Neighborhood Based



CF – Neighborhood Based



Little more formally

- ⦿ Missing value estimation
- ⦿ User-Item matrix of scores
- ⦿ Predict unknown scores within the matrix

Scores??

According to:

- ⦿ Purchases
- ⦿ Rating
- ⦿ Browsing history
- ⦿ ...

Formally..

- ⦿ M ($|M|=m$) users
- ⦿ N ($|N|=n$) items
- ⦿ R $m \times n$ matrix
- ⦿ $r_{u,i}$ the rating of user u of item i

More Problems

- Massive amount of Data
- 99% of the matrix R is unknown (sparse matrix)
- Data is NOT uniform across users & items

Netflix Real-Life Data

- ⦿ 17,700 Movies
- ⦿ 480,000 Users
- ⦿ (rating in a scale of 1-5)
- ⦿ Over 100,000,000 Ratings!!

Netflix – How to Win??

- Quality is measured by RMSE
(more emphasis on large errors)
- Predict unknown 1,400,000 rating and compare them to real rating
- Improve Netflix's system (Cinematch) by 10%

Netflix – How to Win??

- RMSE

$$RMSE = \sqrt{\frac{\sum_{(u,i) \in TestSet} (r_{u,i} - \hat{r}_{u,i})^2}{|TestSet|}}$$

Netflix – Leaderboard

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
<u>Grand Prize</u> - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
<u>Progress Prize 2008</u> - RMSE = 0.8627 - Winning Team: BellKor in BigChaos				
<u>Progress Prize 2007</u> - RMSE = 0.8723 - Winning Team: KorBell				
<u>Cinematch score</u> - RMSE = 0.9525				

Netflix – Statistics

- ⦿ 51,051 contestants, 41,305 teams
- ⦿ 186 countries
- ⦿ 44,014 valid submissions from 5169 different teams

OK, so what's the plan?

- Find a “good” neighborhood

<http://www.youtube.com/watch?v=XOw-ak2aJS8>

(p.s. what about YouTube's related videos?)

- Take a weighted average on the neighbors rate

More Specifically

User-Based:

- $N(u;i)$ – set of users who rate similarly to u and actually rated i

$$r_{u,i} = \frac{\sum_{v \in N(u;i)} s_{u,v} r_{v,i}}{\sum_{v \in N(u;i)} s_{u,v}}$$

$$S_{u,v}$$

Key role! Used for:

- ⦿ Selecting $N(u;i)$
- ⦿ Weighting

Most popular implementations:

- ⦿ Pearson correlation coefficient
- ⦿ Cosine similarity

Pearson correlation coefficient

- $I(u,v)$ – Set of all items rated by both u and v

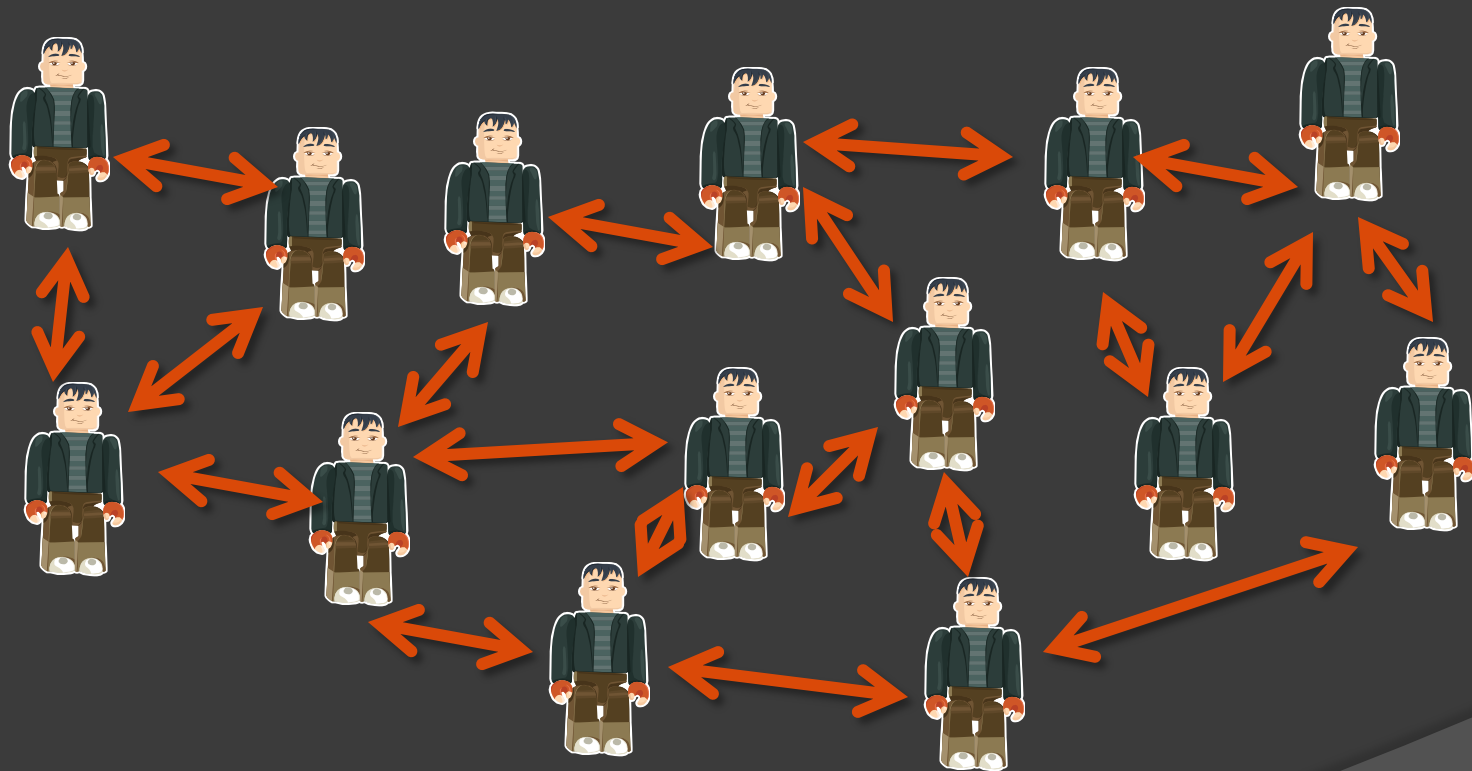
$$S_{u,v} = \frac{\sum_{k \in I(u,v)} (r_{u,k} - \bar{r}_u)(r_{v,k} - \bar{r}_v)}{\sqrt{\sum_{k \in I(u,v)} (r_{u,k} - \bar{r}_u)^2 \sum_{k \in I(u,v)} (r_{v,k} - \bar{r}_v)^2}}$$

$N(u;i)$

Most popular / easiest ways:

- ⦿ Correlation Threshold
- ⦿ Best – n – neighbors
- ⦿ What about external data?

Social Networks!



Social Networks, Hot Topics

- ⦿ Facebook
- ⦿ MySpace
- ⦿ Delicious
- ⦿ Flickr

Quick Summary

Two main parameters:

- ⦿ How to choose the neighbors
- ⦿ How to choose the weights

What about performance?

Netflix Data:

- $N = 17,700$
- $M = 480,000$
- Calculating $N(u;i)$ is expensive
- $M \gg N$

Item-Based

- ⦿ Instead of “users” neighbors,
→ “items” neighbors
- ⦿ Estimate using known rating made by
the user on similar items

More Specifically

Item-Based:

- $N(i;u)$ – set of items who other users rate similar to i . Similarly, all items needs to be rated by u as well

$$r_{u,i} = \frac{\sum_{j \in N(i;u)} s_{i,j} r_{u,j}}{\sum_{j \in N(i;u)} s_{i,j}}$$

Reminder..

User-Based:

- $N(u;i)$ – set of users who rate similarly to u and actually rated i

$$r_{u,i} = \frac{\sum_{v \in N(u;i)} s_{u,v} r_{v,i}}{\sum_{v \in N(u;i)} s_{u,v}}$$

Why is it better?

- ⦿ Similarities is between Items (not Users)
→ Pre-compute all $S_{i,j}$
- ⦿ Provide better recommendations?
- ⦿ Easier Justification
- ⦿ Most industry systems use it (Amazon)

Checkpoint

- ⦿ We know the basics
- ⦿ Can we “Tweak” the basic algorithm?

“Tweaks” - Normalized Data

- ⦿ Some rate 3 and some 5 for movies they liked
- ⦿ Old solution: normalize the dataset
- ⦿ New solution: predict the change from the average rating instead of the rating

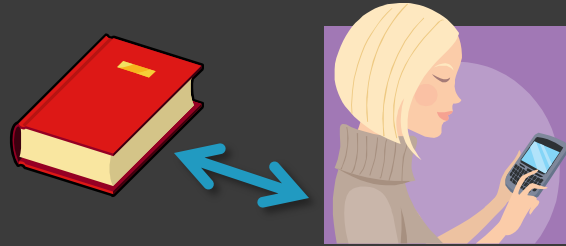
“Tweaks” - Remove Global Effects

- ⦿ A user rates 5 all the times
- ⦿ A user rated 10,000 movies
- ⦿ Remove old rating?
- ⦿ Using the Time variable is not “Tweak”..

TAU's Current Research

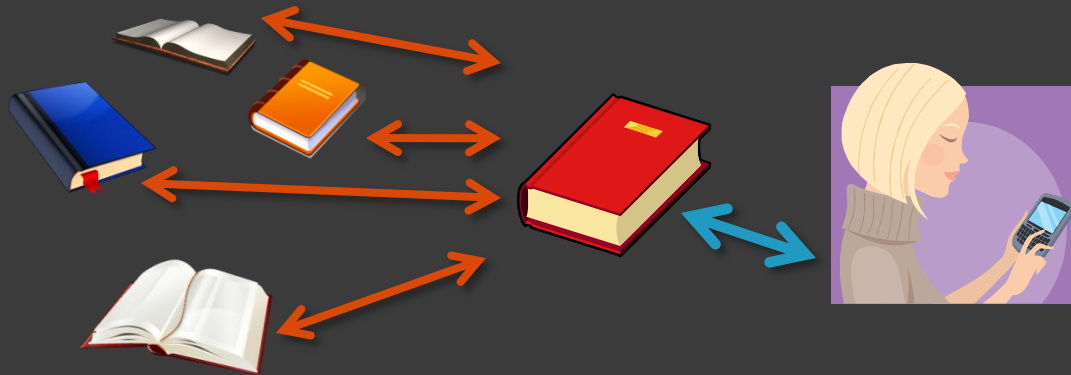
- ⦿ Distributed CF!!!
- ⦿ “Server” level

Distributed CF



Distributed CF

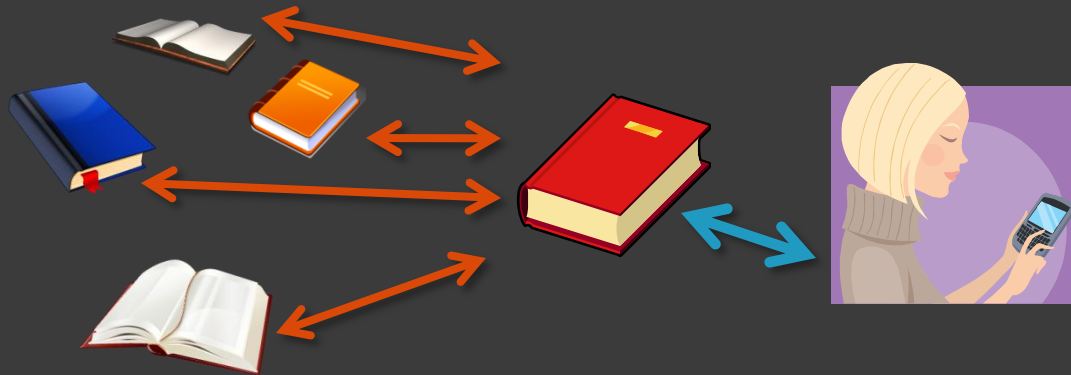
amazon



Distributed CF

amazon

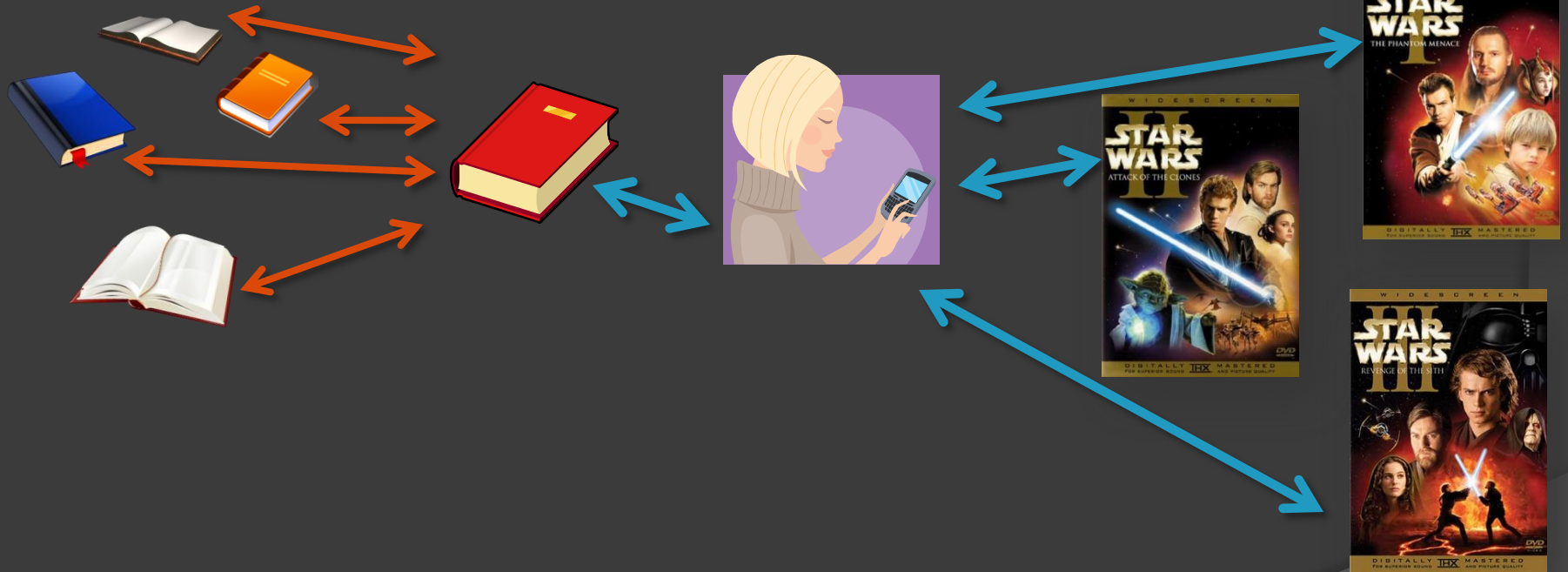
NETFLIX



Distributed CF

amazon

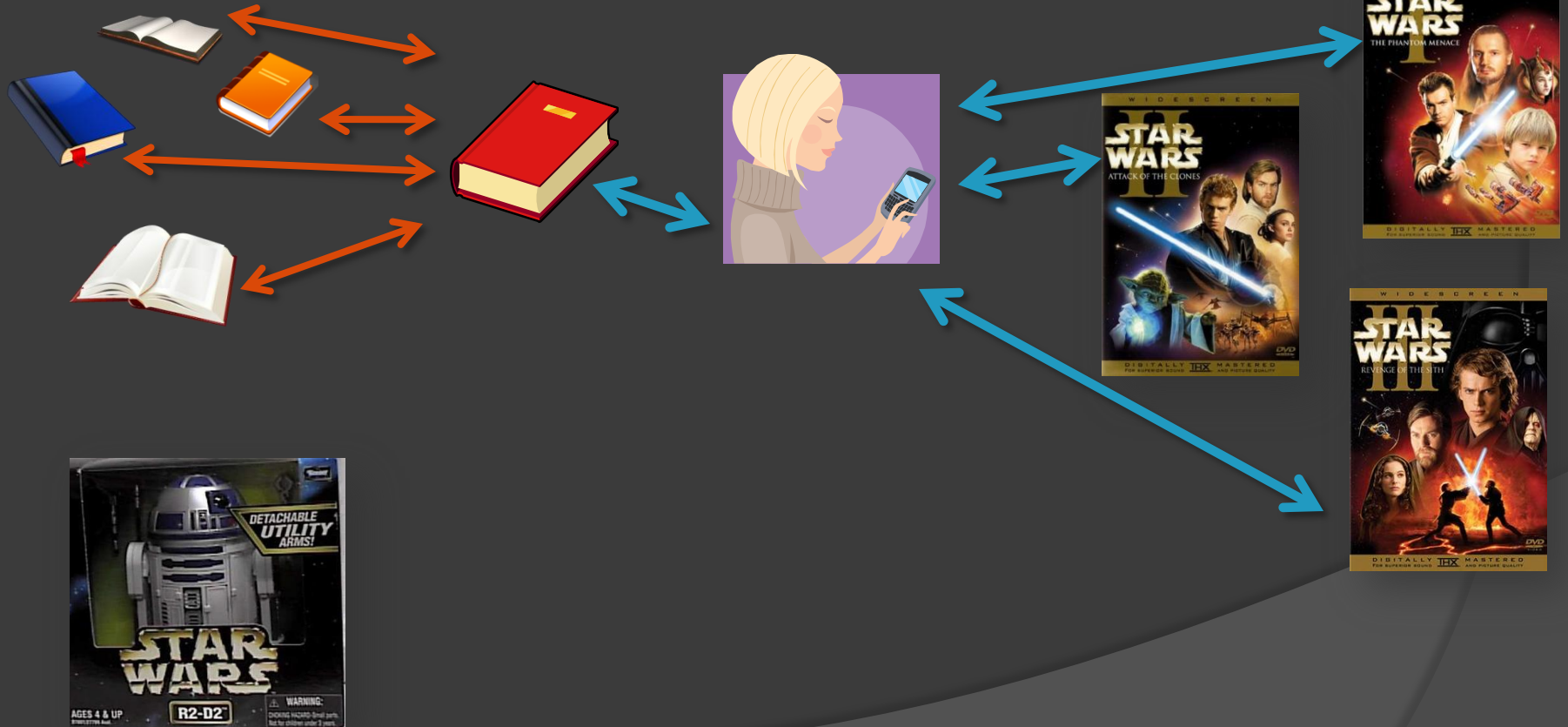
NETFLIX



Distributed CF

amazon

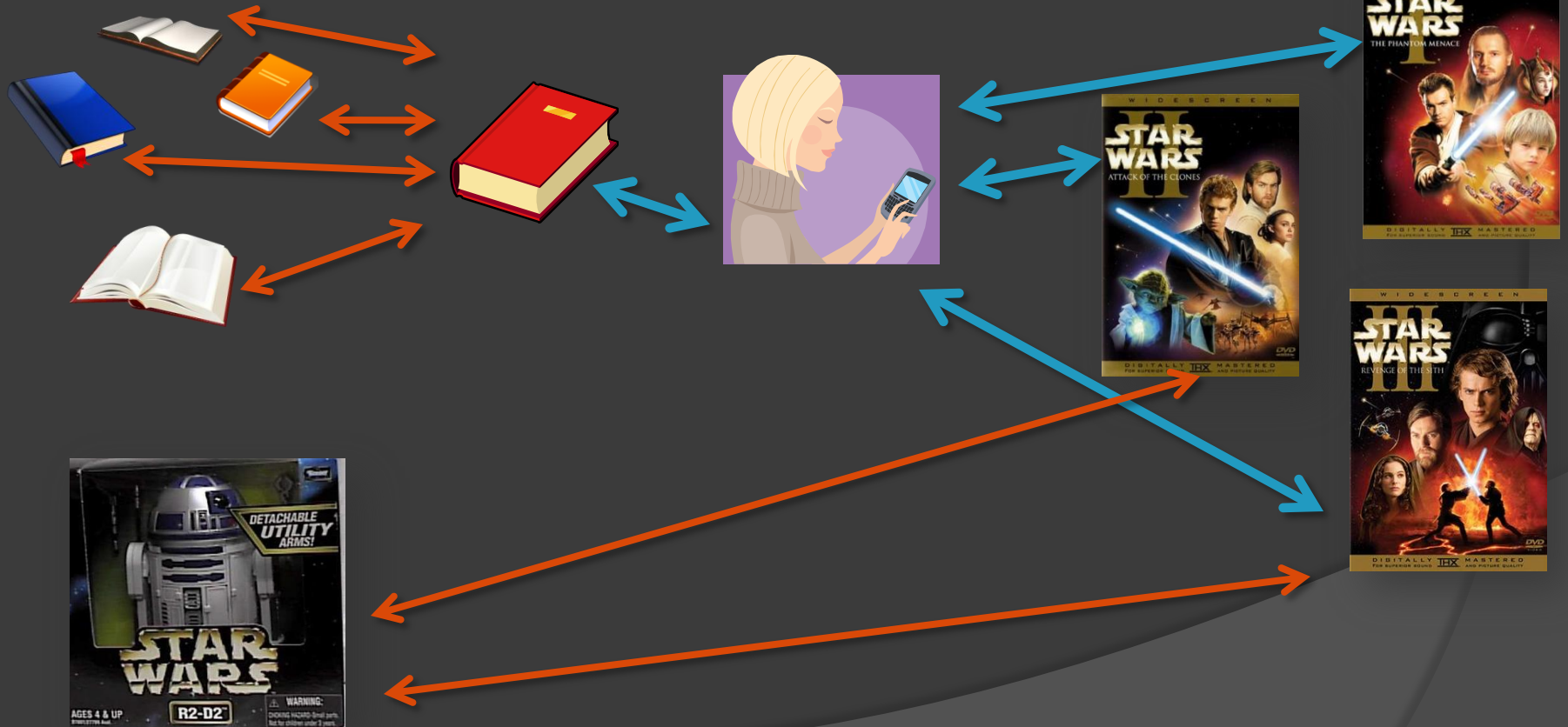
NETFLIX



Distributed CF

amazon

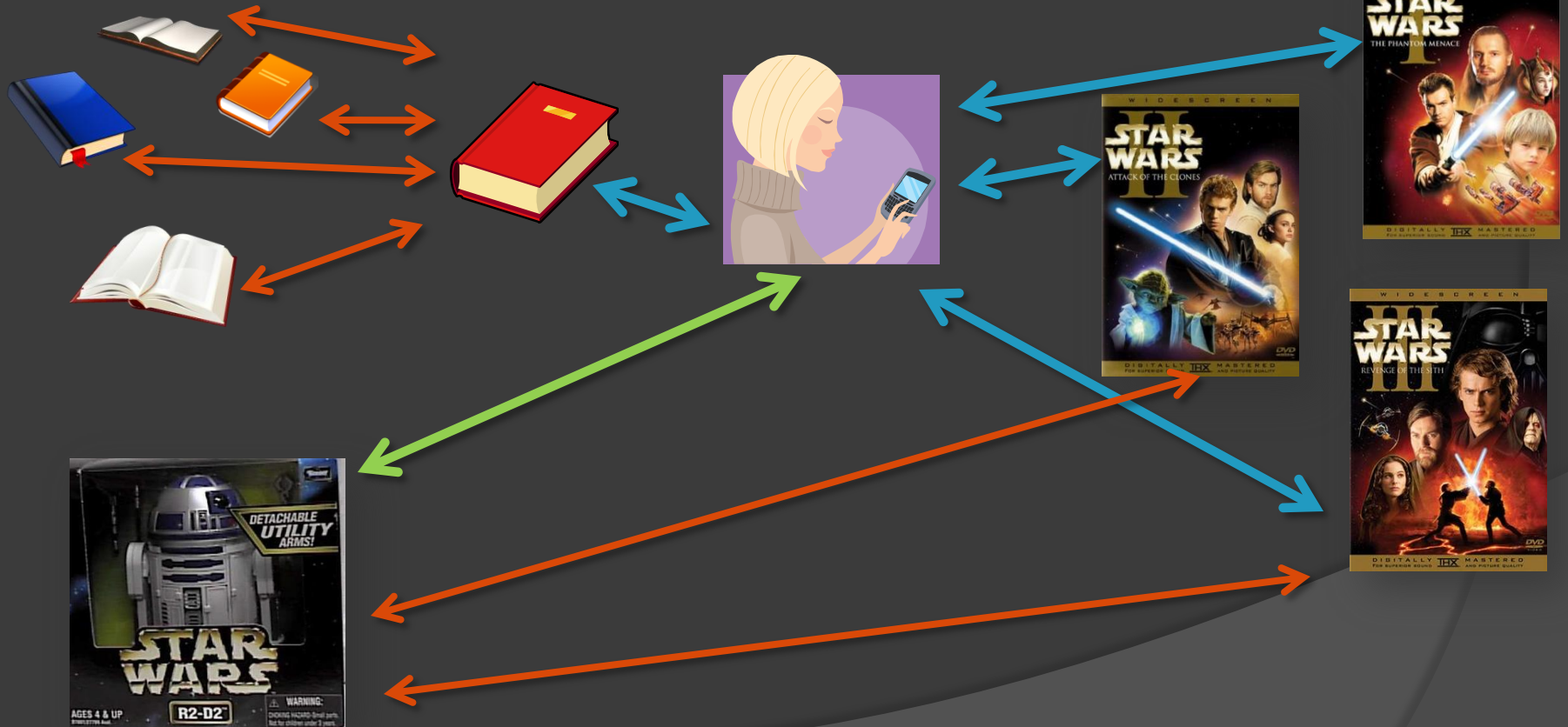
NETFLIX



Distributed CF

amazon

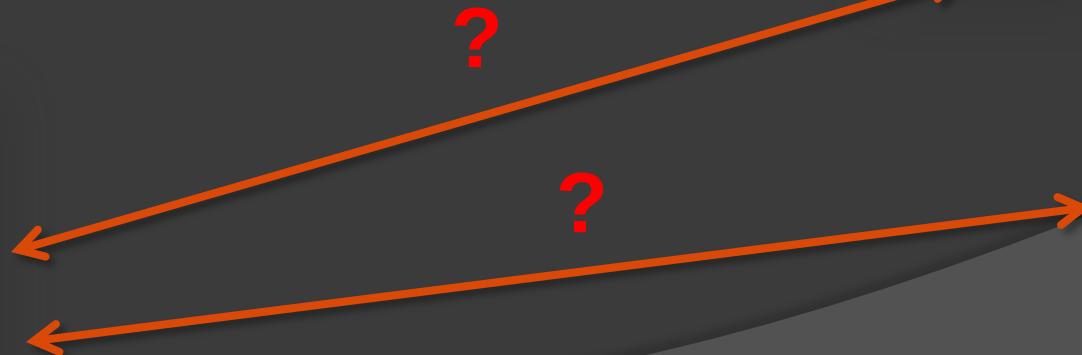
NETFLIX



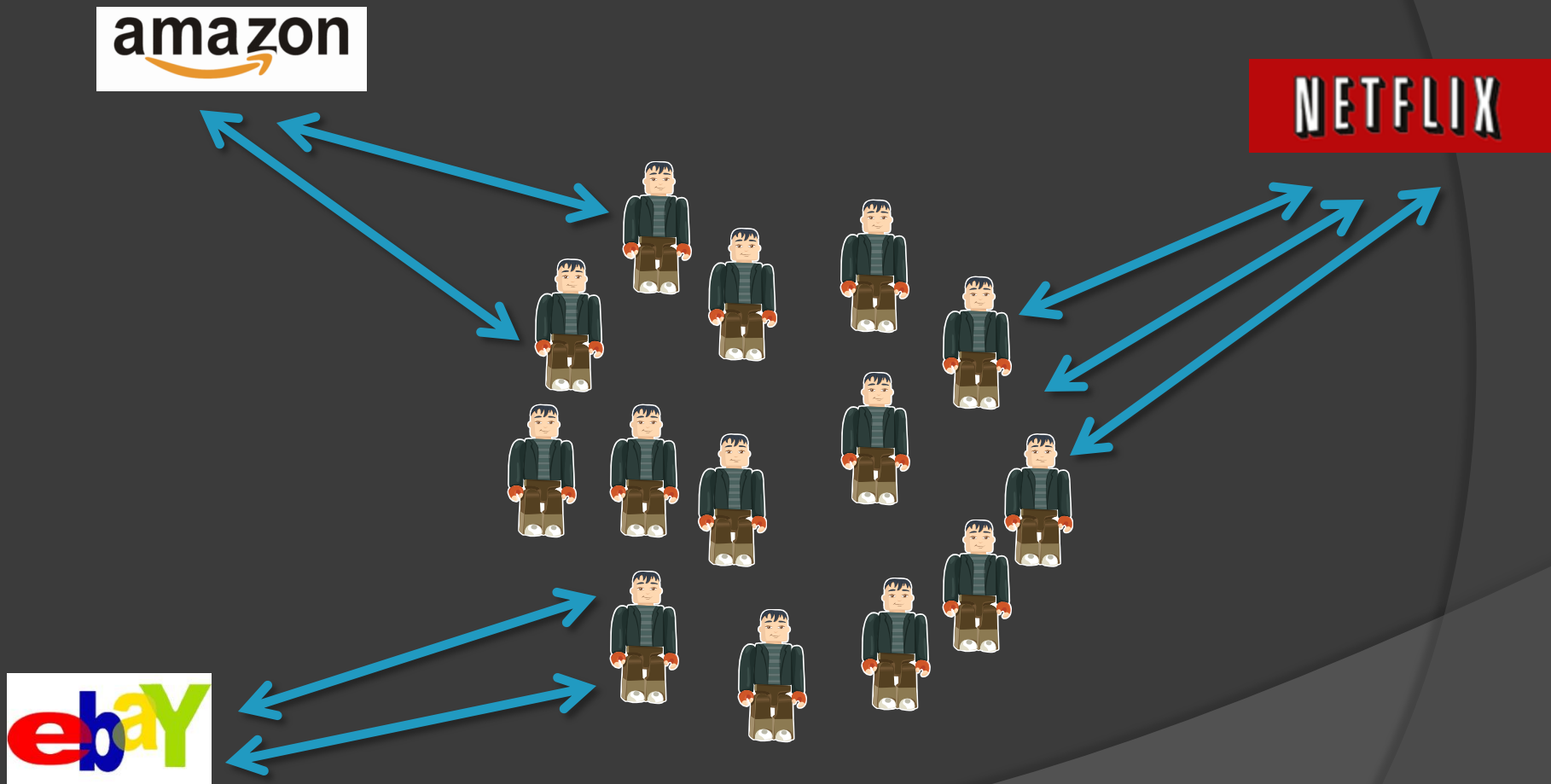
Distributed CF



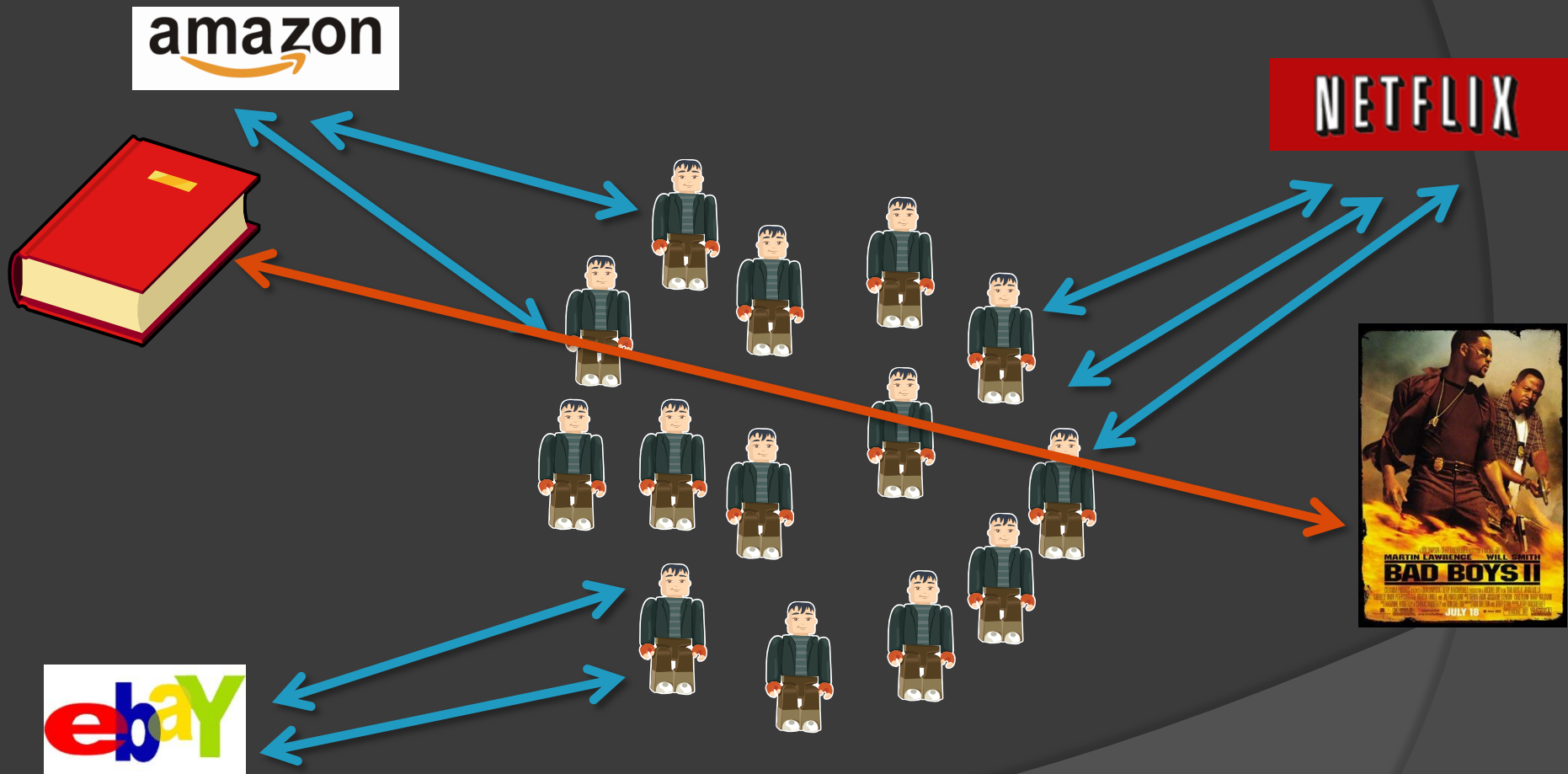
NETFLIX



Shared Users



Shared Users



Shared Items



Shared Items



How To Do It????

Copy all data to one server?

- ⦿ CF algorithm do not scale linear
- ⦿ Privacy
- ⦿ Bandwidth

TAU's Solution

- Join TAU's DB group for more info 😊

TAU's Databases Group

[Tova Milo](#)[DB Group](#)[Projects](#)[Publications](#)[Courses](#)

Tel-Aviv University's DB Group

[School of Computer Science](#)

[Tel Aviv University](#)

Databases Lab, M-20

Schreiber Building

Phone: +972-3-640-5396

- [More group photos](#)
- [Past Members](#)



Dinner at Moses, November 2008

Current Members