The Long Tail and Recommender Systems

Hits vs. Niche

- Giant retailers of books, music, etc.
- Question: are most sales being generated
  - by a small set of items that are enormously popular ("hits", blockbusters), or
  - by a much larger population of items that are each individually less popular ("niche products" appealing to a small segment of the audience)?
- Answer:
  - Despite stereotype of media business: only blockbusters matter
  - We observe that the total sales volume of *unpopular items, taken together, is* very significant

The Long Tail

- Chris Anderson (2004)
  - Internet-based distribution is driving the media and entertainment industries to a "long tail" of obscure products driving the bulk of sales
- Amazon or Netflix sell an astronomical diversity of products (no restrictions imposed by physical stores) even when very few of them generate much volume on their own
- Quantifying the importance of the Long Tail comes down to an analysis of power laws

Visualizing the Long Tail

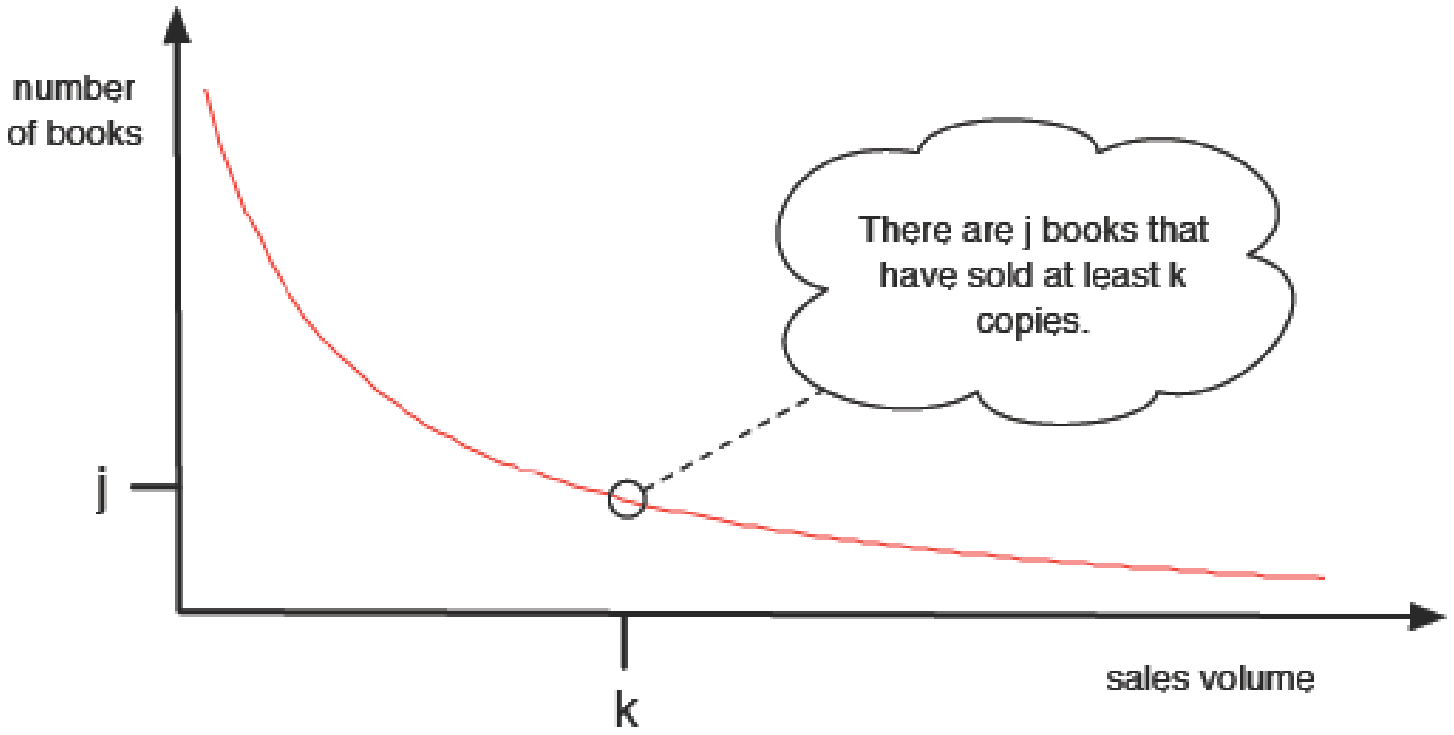
- Our original definition of the popularity curve:

As a function of k, what fraction of items have popularity exactly k?

- Modify our original definition slightly:

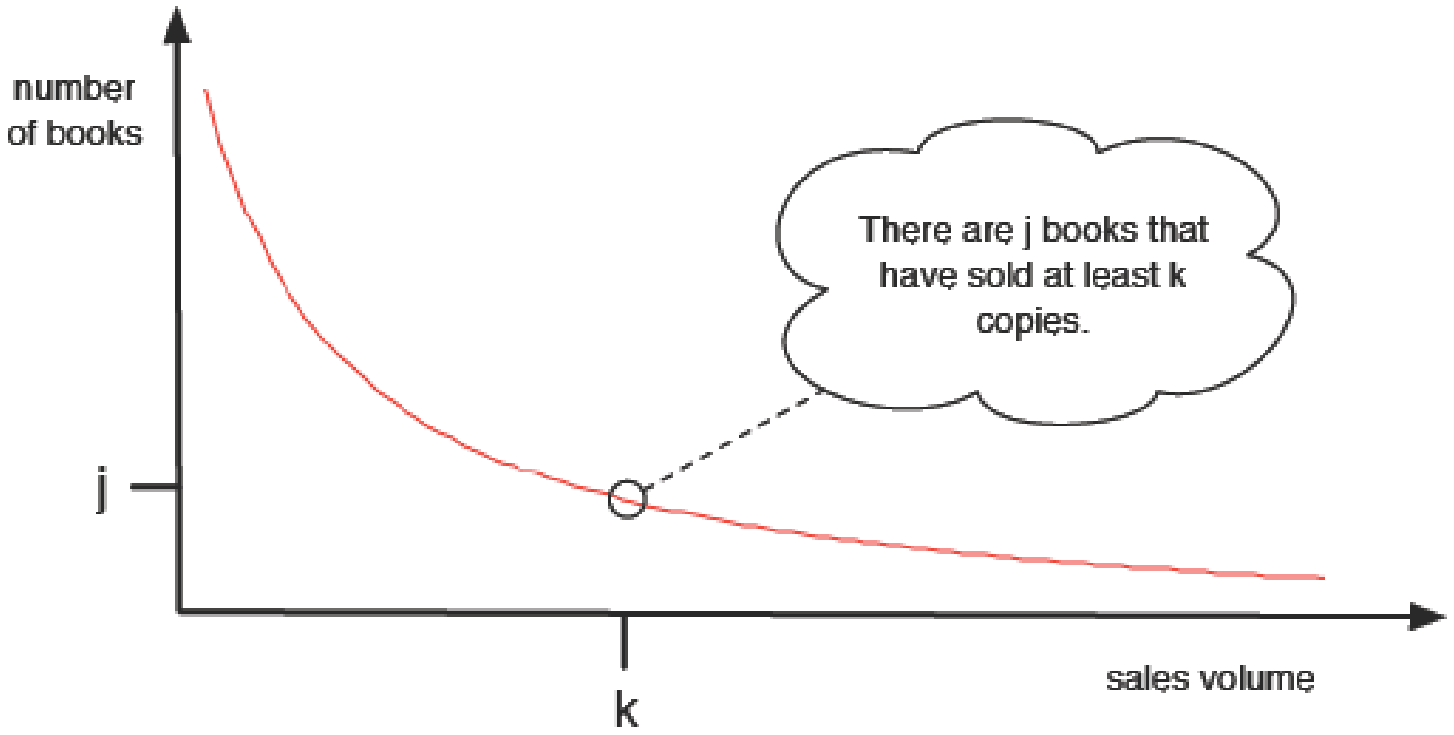
As a function of k, what number of items have popularity at least k?

Visualizing the Long Tail



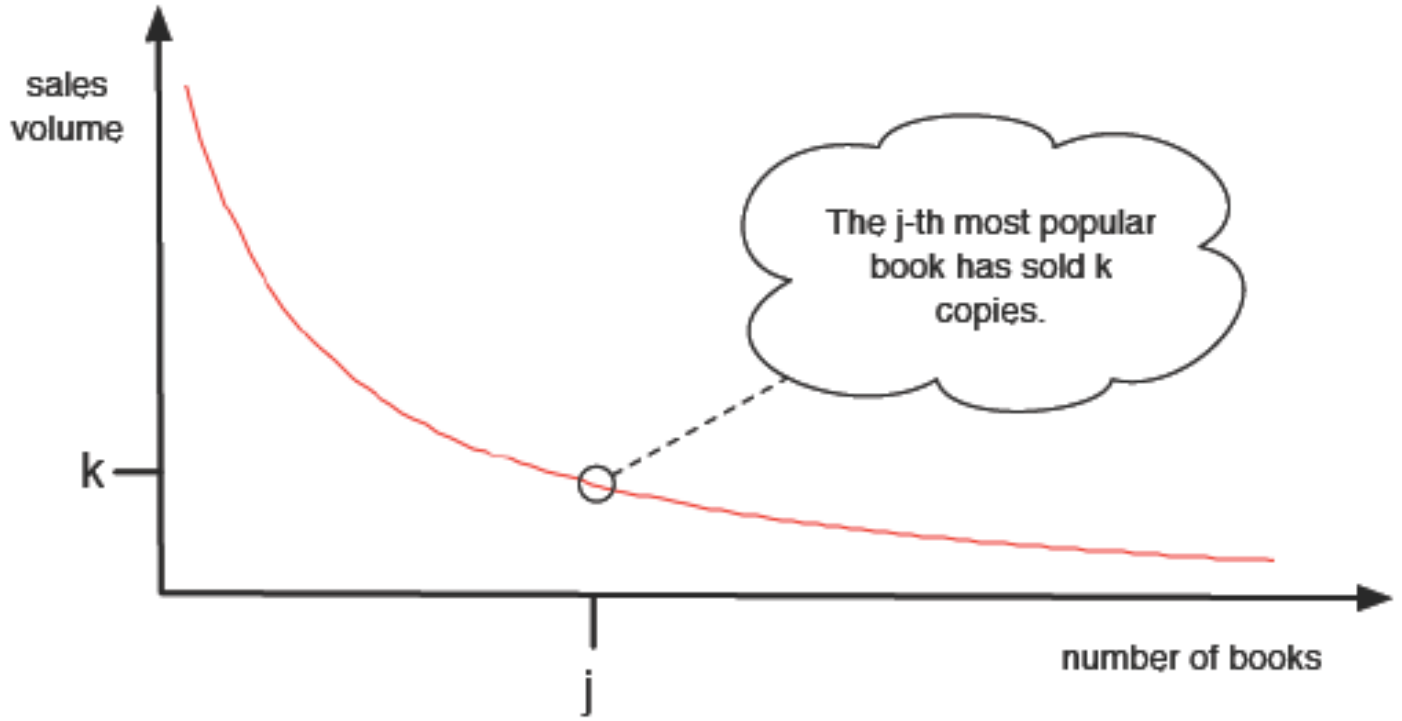
- If the original function was a power-law, then this new one is too (we showed that in previous lecture)

Visualizing the Long Tail



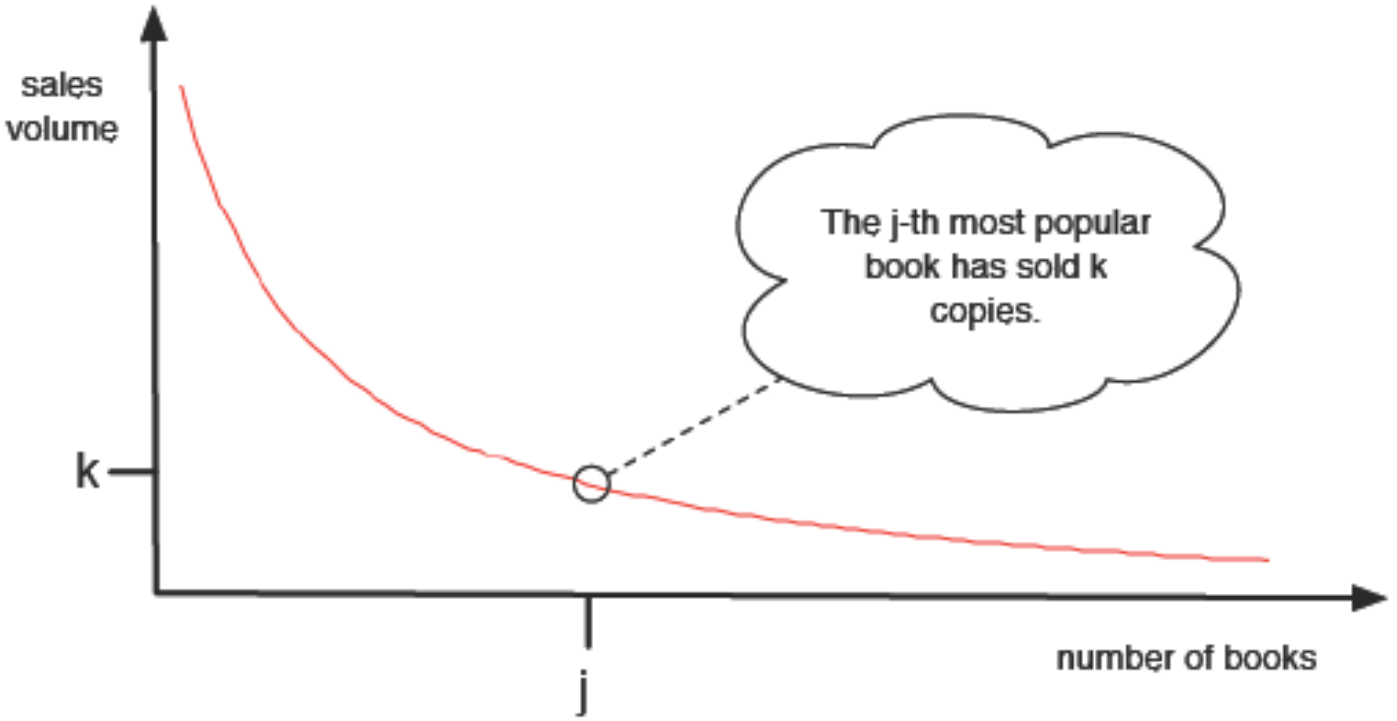
- Question we ask: As you look at larger and larger sales volumes, how few books do you find?
- Questions we want to ask: As you look at less and less popular items, what sales volumes do you see?

Visualizing the Long Tail



- Interchange the roles of the x- and y-axes
- New curve says: The j-th most popular book has sold k copies

Visualizing the Long Tail



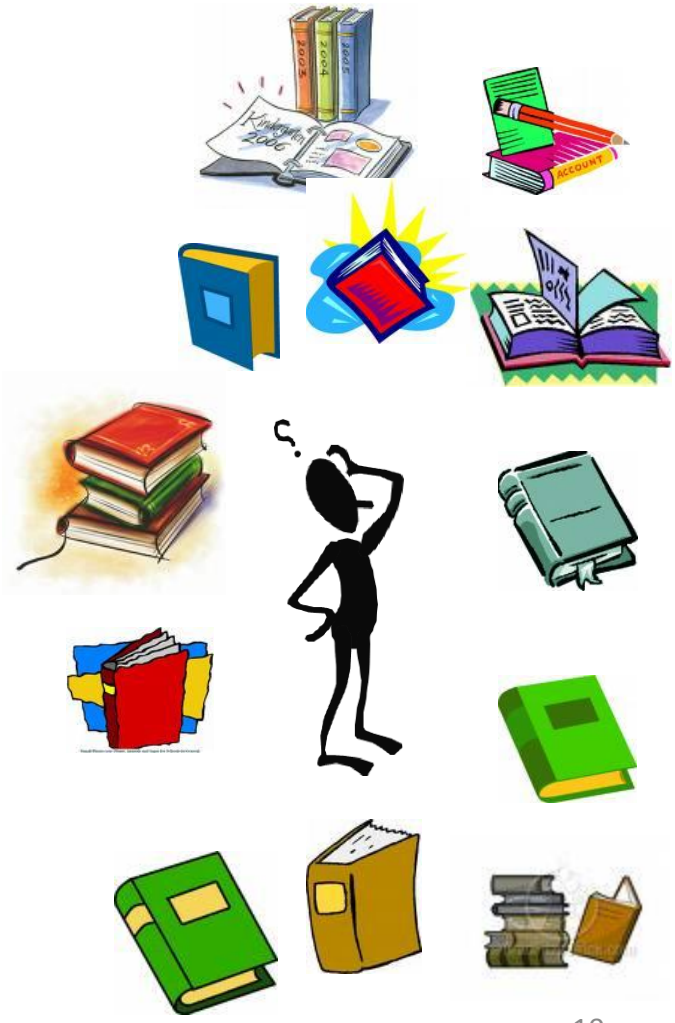
- Order by “sales rank” and look at the popularity as we move out to larger and larger sales ranks -> into the niche products
- The characteristic shape gives the name “Long Tale”
- Significantly more area under the right (niche products) compared to the left part (hits)

Need for Recommendations

- Companies make money from a giant inventory of niche products when customers are aware of these products and have some reasonable way to explore them
- Recommender systems that companies like Amazon and Netflix have popularized can be seen as integral to their business strategies
  - They expose people to items that may not be generally popular, but which match user interests as inferred from their history of past purchases

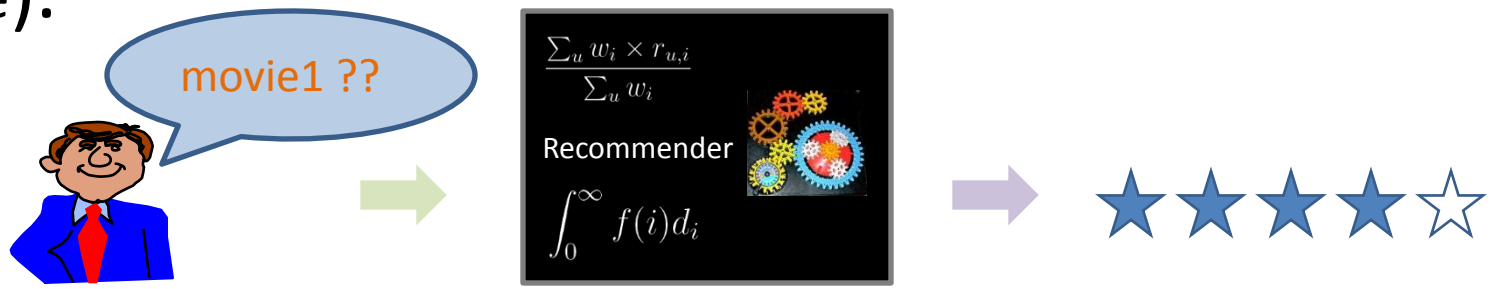
Recommender Systems

- Need For Recommenders
  - Rapid Growth of Information
  - Lots of Options for Users
- Input Data
  - A set of users $U=\{u_1, \ldots, u_N\}$
  - A set of items $I=\{i_1, \ldots, i_M\}$
  - The rating matrix $R=[r_{u,i}]_{NxM}$

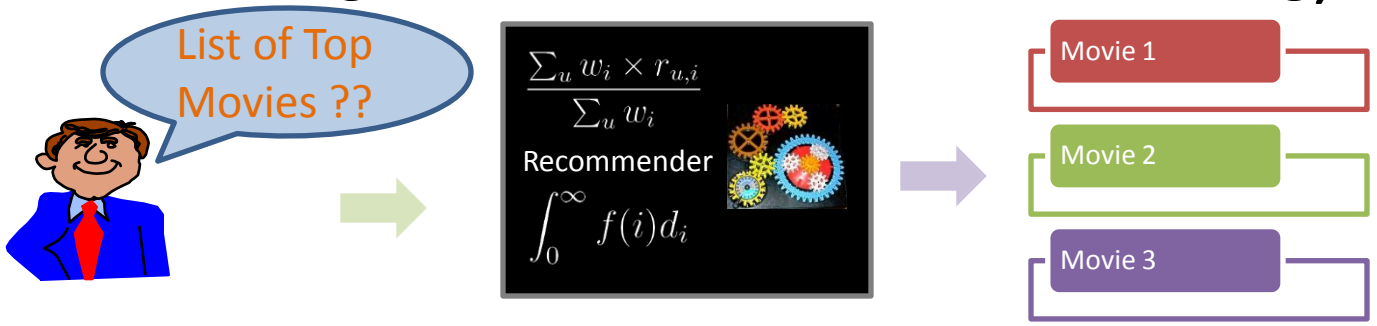


Problem Definitions in RSs

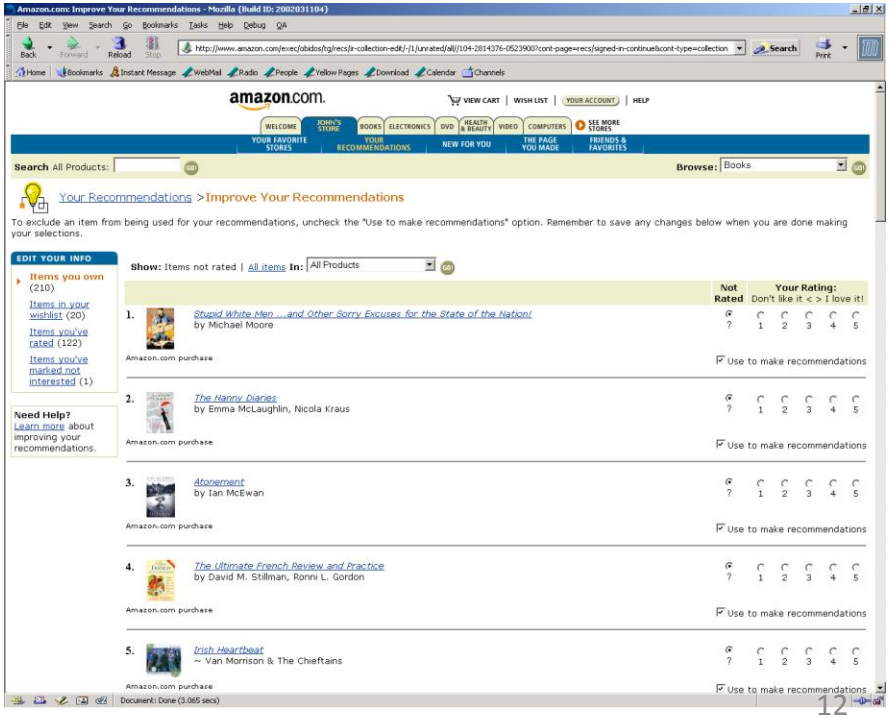
- Predicting the rating on a target item for a given user (i.e. Predicting John's rating on Star Wars Movie).



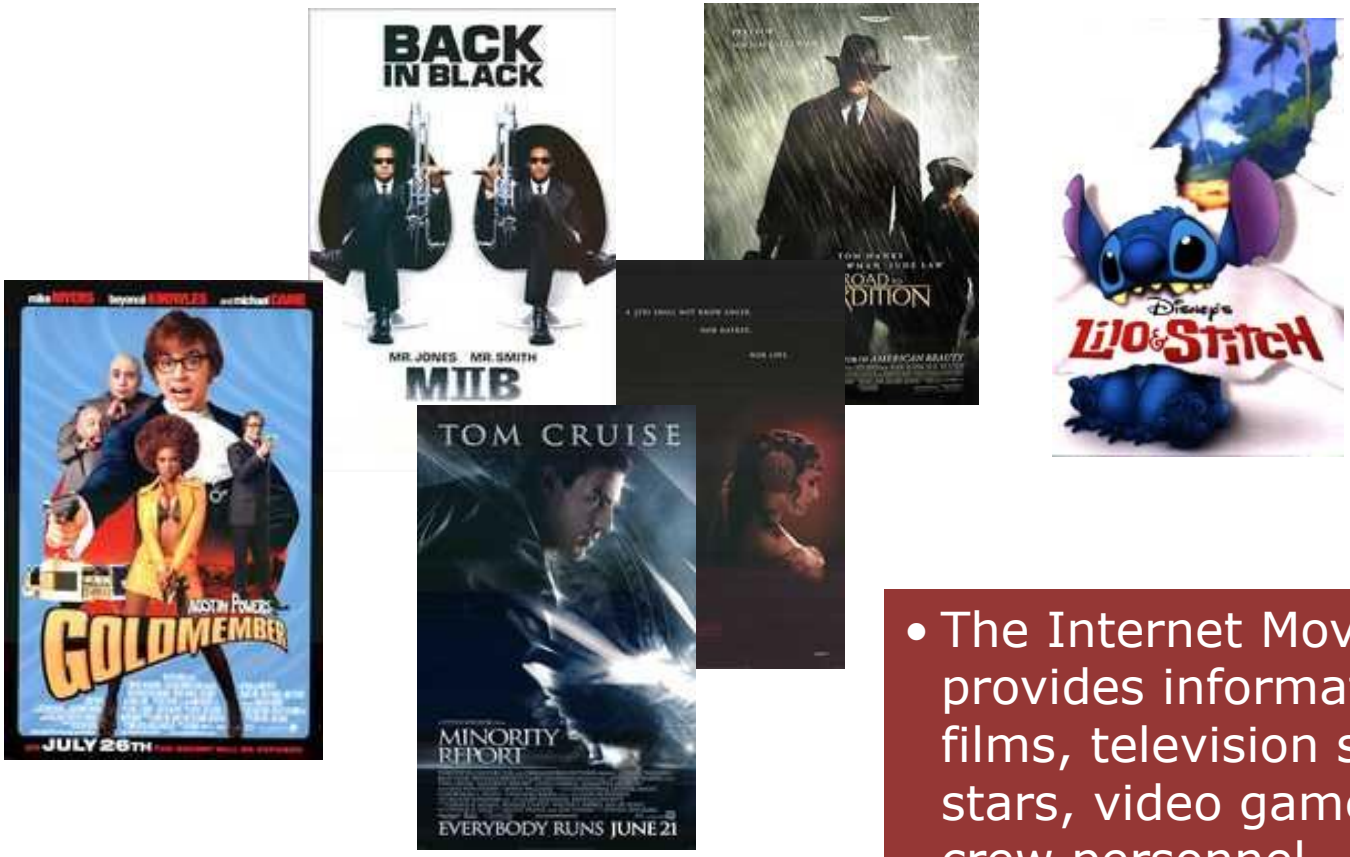
- Recommending a List of items to a given user (i.e. Recommending a list of movies to John for watching).



What book should I buy?



What movie should I watch?



- The Internet Movie Database (IMDb) provides information about actors, films, television shows, television stars, video games and production crew personnel.
- Owned by Amazon.com since 1998
- 796,328 titles and 2,127,371 people
- More than 50M users per month.

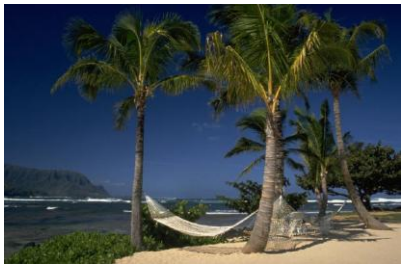
Netflix Prize

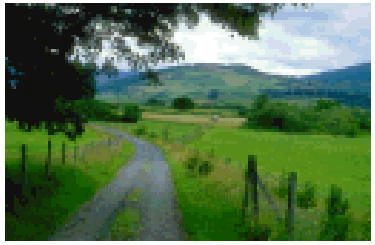
The Nextflix prize story

- In October 2006, Netflix announced it would give a $1 million to whoever created a movie-recommending algorithm 10% better than its own.
- Within two weeks, the DVD rental company had received 169 submissions, including three that were slightly superior to Cinematch, Netflix's recommendation software
- After a month, more than a thousand programs had been entered, and the top scorers were almost halfway to the goal
- But what started out looking simple suddenly got hard. The rate of improvement began to slow. The same three or four teams clogged the top of the leader-board.
- Progress was almost imperceptible, and people began to say a 10 percent improvement might not be possible.
- Three years later, on 21st of September 2009, Netflix announced the winner.

What news should I read?

Where should I spend my vacation?





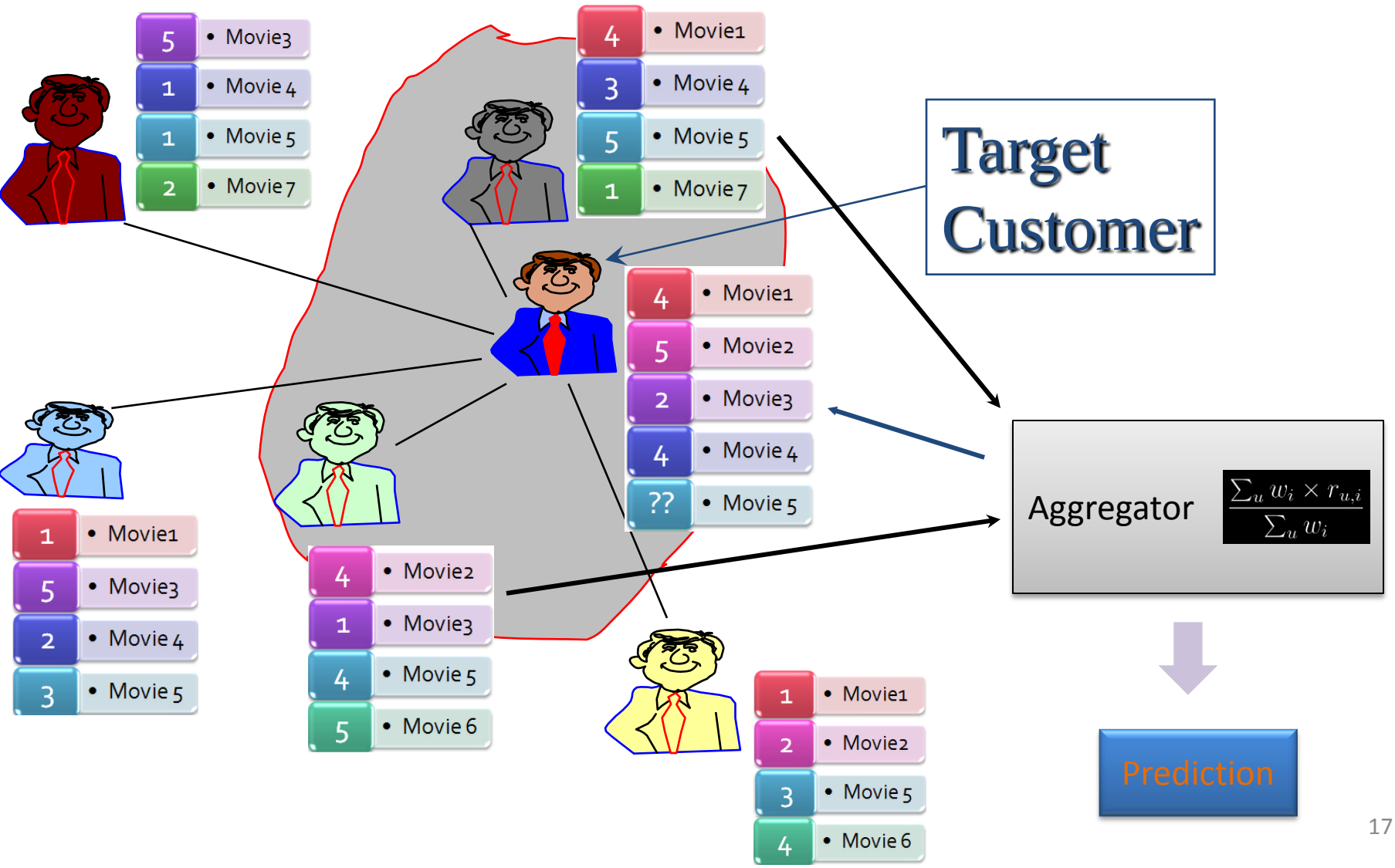
Tripadvisor.com

I would like to escape from this ugly an tedious work life and relax for two weeks in a sunny place. I am fed up with these crowded and noisy places … just the sand and the sea … and some "adventure".

I would like to bring my wife and my children on a holiday … it should not be to expensive. I prefer mountainous places… not too far from home. Children parks, easy paths and good cuisine are a must.

I want to experience the contact with a completely different culture. I would like to be fascinated by the people and learn to look at my life in a totally different way.

Collaborative Filtering



Collaborative Filtering

	Star Wars	Hoop Dreams	Contact	Titanic
Joe	5	2	5	4
John	2	5		3
Al	2	2	4	2
Nathan	5	1	5	?

The problem of collaborative filtering is to predict how well a user will like an item that he has not rated given a set of historical preference judgments for a community of users

User-based CF

- Calculate the similarity (weight) $w_{u,v}$ between the active user *u* and all other users *v*:

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \overline{r}_u)(r_{v,i} - \overline{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \overline{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \overline{r}_v)^2}}$$

  - The $i \in I$ summations are over the items that both the users *u* and *v* have rated

- Generate a top-N recommendation:
  - Find *k* most similar users (nearest neighbors)
  - Identify a set of items, *C,* purchased by the nearest neighbors, together *with* their frequency
  - Recommend the top-*N* most frequent items in *C* that the active user *u* has not already purchased

Evaluation metrics

- Precision of top-N item recommendation:
  - For the active (test) user assume a subset of its ratings as known and the rest as unknown
  - Generate top-N item recommendation based on the known ratings
  - Find the number X of top-N recommender items that also belong in the unknown items
  - Precision is X/N