Recommender Systems & Collective Intelligence

COMP47580

Dr. Michael O'Mahony

michael.omahony@ucd.ie

Recommender Systems

- Personalised vs. non-personalised
- Recommendation Approaches:
  - Collaborative, content-based, social, context-aware, demographic, hybrid, group RS, conversational RS…
- Recommendation Knowledge:
  - User profiles (implicit/explicit preferences)
  - Product metadata (e.g. specs)
  - UGC – consumer reviews, tweets, FB posts/likes…
- Recommendation Output:
  - Ranked product/item suggestions
  - Predictions for products
  - Explanations

Recommender Systems

- Recommender systems help to drive demand down the long-tail; benefits to both consumers and retailers alike

- Consumers:
  - Many more products from which to choose – information overload
  - Recommender systems enable *findability* – assist consumers to find niche products that match their personal tastes

- Retailers:
  - Turning browsers into buyers, promoting cross-selling, customer loyalty
  - Increased sales

- Recommender systems now in widespread use…

Non-personalised Recommendation

- Personalised vs. non-personalised recommendation:
  - Personalised – recommendations tailored to the particular preferences of users
  - Non-personalised – the same recommendations are made for everyone (with certain exceptions)
- This lecture – focus on non-personalised recommenders; widely used in many domains.
- We will explore different approaches to non-personalised recommendation and discuss the pros and cons of these approaches

Predictions & Recommendations

- Distinguish between predictions and recommendations:
  - Predictions:
    - Determine whether a user will like a given item – for example, the recommender predicts 4.5 stars for *Lustrum* by Robert Harris
  - Recommendations:
    - Identify items that a user likes, typically presented in the form of a top-N ranked list
  - Predictions and recommendations are often presented together:

Non-personalised Recommendation

- Main approaches:
  - Aggregated opinion recommenders:
    - Based on user preference data
  - Product association recommenders:
    - Based on relations between items
  - Content-based recommenders (future lecture)
    - Based on descriptions of items

- Begin with aggregated opinion recommenders:
  - How can preference data be obtained?
  - Data can be collected *explicitly* or *implicitly*

Explicit Ratings

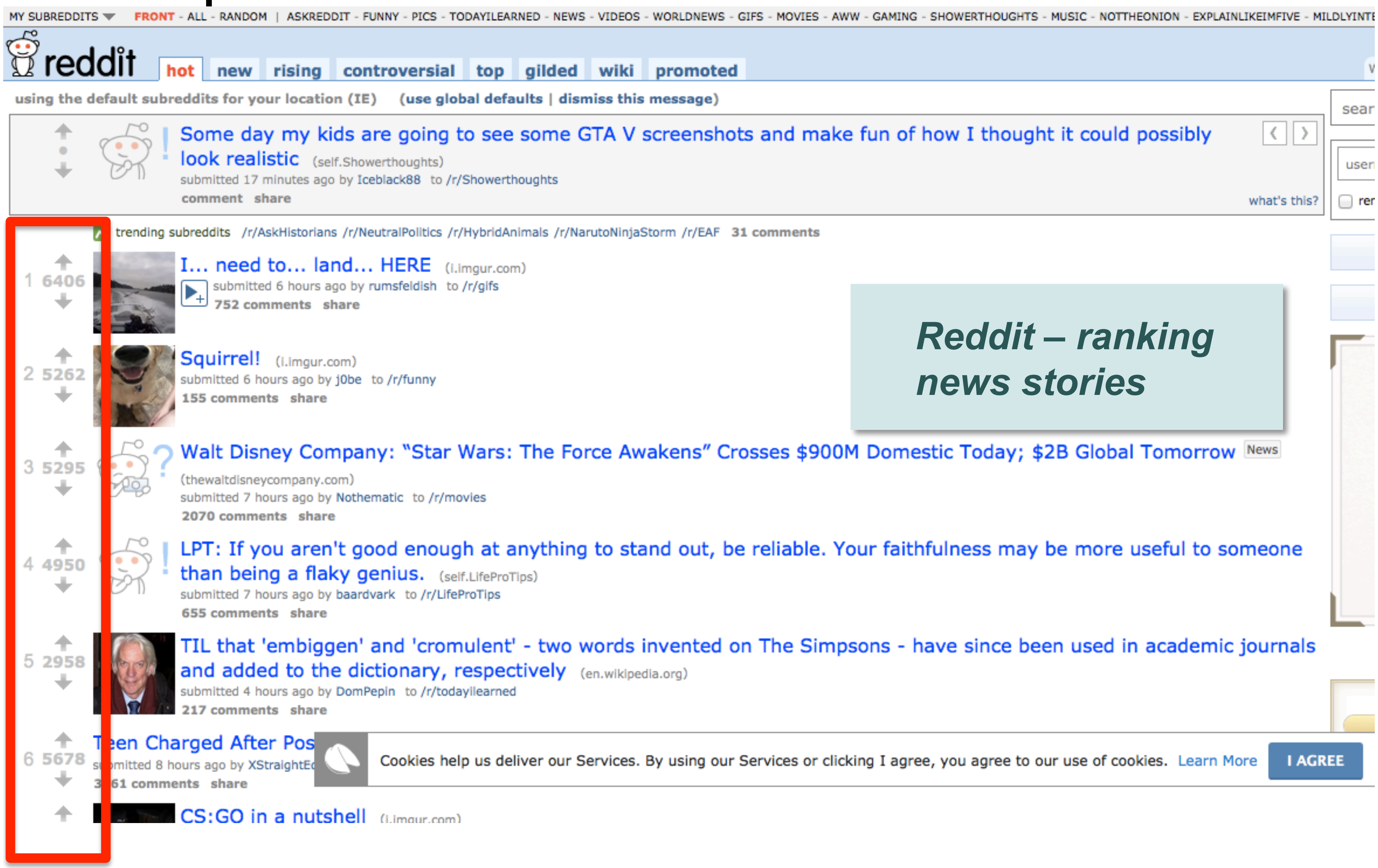
- Users provide ratings – less noisy (compared to implicit data) method of gathering preference data, but still noisy!
- A cost is associated with the provision of ratings – users need to be convinced there is a benefit in order to make the effort
- Examples:
  - Numerical ratings – e.g. 1–5 stars
  - Ordinal ratings – e.g. strongly agree, agree, neutral, disagree, strongly disagree
  - Binary ratings – e.g. votes up or down, agree or disagree
  - Unary ratings – e.g. likes
- Ratings can be provided at the time of consumption, from memory (after the experience), or from expectation (the item has not yet been experienced)
- Issues: reliability, preferences changing with time, malicious ratings

Implicit Ratings

- Not obtained directly from user; instead, ratings are “inferred” from user behaviour/activity
- Examples:
  - Browsing patterns, links followed/not followed
  - TV shows watched, music listened to, etc.
  - Often in the form of unary data (other types possible)
- Removes cost of explicitly gathering ratings
- Every user interaction can potentially contribute
- Implicit data – available in quantity but noisy:
  - No guarantee that web pages visited or music listened to are liked; also e.g. non-visits does not imply dislike
  - Time spent on pages, repeat visits to pages, number of minutes of TV program watched etc. can be leveraged to better infer user preferences

Some examples…

Reddit – ranking news stories



Reddit – ranking news stories

Article #5 has fewer net up votes then Article #6 ??

StackExchange Q&A ranking

Why are “blockbusters” called blockbusters?

83

6

I have just been reading Wiki on the term "Blockbuster" popularly used to describe a highly popular action movie. Here is an excerpt:

> "...a fast-paced exciting entertainment, almost a genre. Audi
> talked about them afterwards, and went back to see them ag

However I still do not understand the root of this film industry term, what does it mean?

terminology

share improve this question

edited Feb 8 '16 at 15:13

asked Feb 5 '16 at 19:54
eYe
7,873 21 73 143

add a comment

4 Answers

active oldest votes

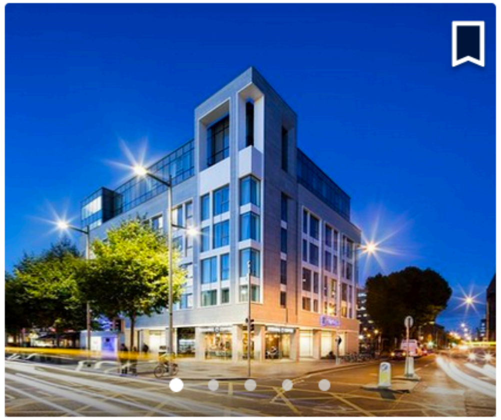
72

Originally the word was used to describe large bombs dropped during World War II that could quite literally destroy an entire city block, or a "blockbuster bomb".

The origins of the term when it comes to film, according to Wikipedia:

> In film, a number of terms were used to describe a hit. In the 1970s these included: "spectacular" (The Wall Street Journal), "super-grosser" (New York Times), and "super-blockbuster" (Variety). In 1975 the usage of "blockbuster" for films coalesced around Steven Spielberg's Jaws and became perceived as something new: a cultural phenomenon, a fast-paced exciting entertainment, almost a genre. Audiences interacted with such films, talked about them afterwards, and went back to

TripAdvisor – ranking hotels, restaurants etc.



Holiday Inn Express Dublin City Centre
Sponsored

Holiday Inn Express

€103

View Deal

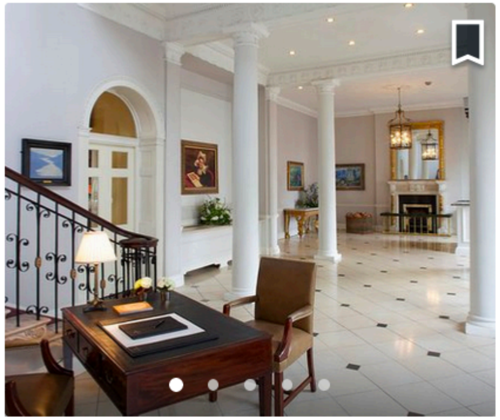
Booking.com
€103

Expedia.ie
€103

Agoda.com
€103

View all 12 deals from €103

Special offer



The Merrion Hotel

Booking.com

€260

View Deal

LHW.com ↗
€250

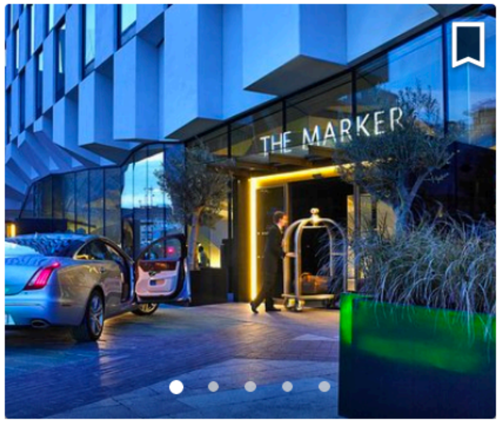
Expedia.ie ↗
€260

travelup.com ↗
€314

View all 12 deals from €250

3,964 reviews
#1 of 155 hotels in Dublin

Free Wifi
Pool
Visit hotel website



The Marker Hotel

Booking.com

~~€227~~
€219

View Deal

LHW.com ↗
€219

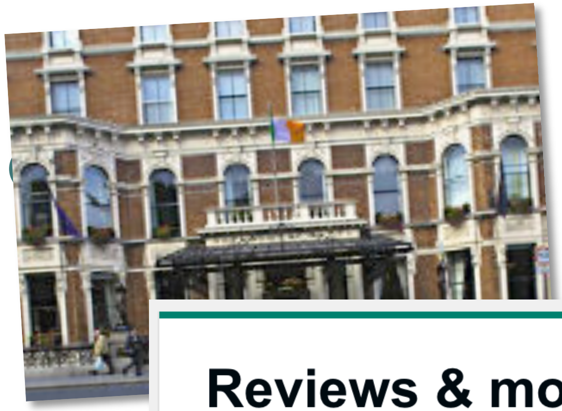
Expedia.ie ↗
€219

getaroom.com ↗
€227

View all 13 deals from €219

4,226 reviews
#2 of 155 hotels in Dublin

Free Wifi
Pool



tripadvisor®

Reviews & more

Write a review

4,927	3,301	130	100
Reviews	Photos	Q+A	Room tips

Traveller rating

- Excellent 3,436
- Good 987
- Average 308
- Poor 117
- Terrible 80

Time of year

- Mar-May
- Jun-Aug
- Sep-Nov
- Dec-Feb

Traveller type

- Families
- Couples
- Solo
- Business
- Friends

Language

- All languages (4,927)
- English (4,702)
- Chinese (Sim.) (88)
- German (75)

More languages

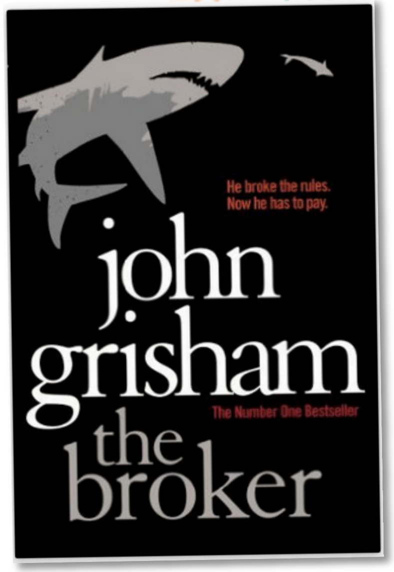
Popular mentions

All reviews · knights bar · fahrenheit restaurant · indigo lounge · real castle · four poster bed · modern amenities · city centre · amazing hotel · bus stop · dublin airport · ireland · wedding · armour · concierge · howth · euro · pub · guinness

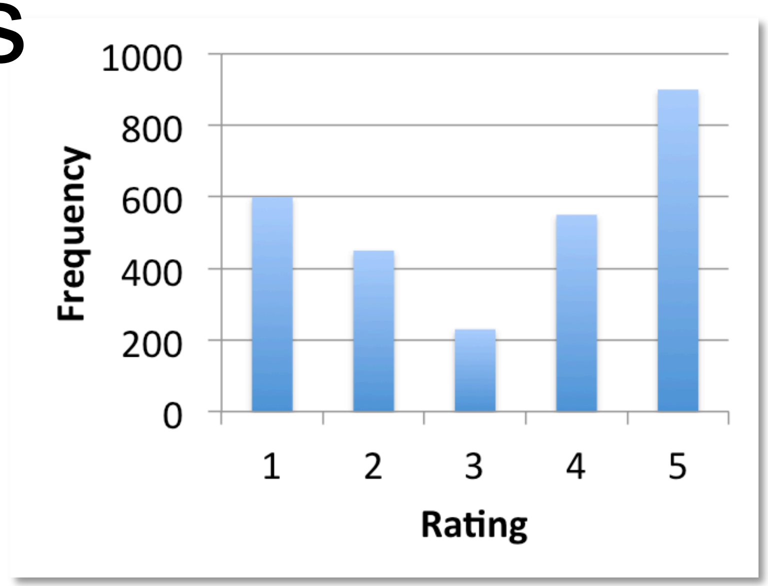
Making Predictions

Some (simple) approaches:

- 5 star rating scales – compute mean rating or percentage of ratings ≥ 4;
- Binary rating scales – percentage of up votes:
  - No indication of item popularlty
  - Shows controversy
- Number of likes:
  - Provides an indication of popularity
  - The number of down votes received are not shown – removes controversy but important information is missing
- Display full distribution of ratings to users:
  - Complex, e.g non-trivial to compare items

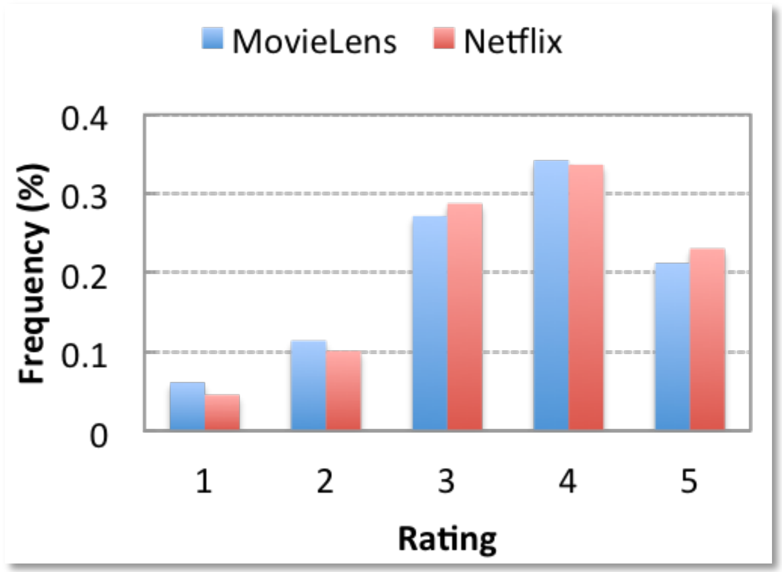


Making Predictions – Average Ratings



Consider 5 star rating scales:

- Need to consider distribution of ratings
- Many users provide ratings – diverse tastes, averages move toward the centre
- Does not provide an indication of popularity – e.g. a book which has received one 5 star rating (average gives max rating!)
- Selection bias – e.g. people usually do not watch movies they think they will not like
- Later look at personalised predictions – based on the preferences of other users with similar tastes



Making Predictions – Average Ratings (aside)

Consider the following scenario in which users' ratings for items are available. Assume the objective is to predict the rating for item 2

	Item 0	Item 1	Item 2
User 0	9	8	
User 1		4	6
User 2			7

- Approach #1 – return the mean rating of item 2 = 6.5
- Approach #2 – consider *user* and *item biases*:
  - For example, predict the rating for user 0 and item 2
  - Overall mean rating (μ): (9+8+4+6+7) / 5 = 6.8
  - Bias user 0 (bu_0): (9+8) / 2 - 6.8 = 1.7
  - Bias item 2 (bi_2): (6+7) / 2 - 6.8 = -0.3
  - Prediction given by: $\mu + bu_0 + bi_2 = 6.8 + 1.7 - 0.3 = 8.2$

Recommendations and Ranking

- What items should be placed at the top of a list?
  - Relevance to users is key
  - Conservative (obvious, generic?) vs. high-risk/reward (interesting, novel?) recommendations
  - Base ranking on predictions?

- Considerations:
  - Business imperatives – e.g. promoting items
  - Domain – e.g. temporal considerations, older items may no longer be relevant (e.g. news articles)
  - Popular, new, hot, rising, controversial items – influence on ranking?

Ranking – Examples…

- **WRONG SOLUTION #1: Score = (# positive ratings) – (# negative ratings)**

  Example:

  \- **item #1** – 600 positive ratings and 400 negative ratings: 60% +

  \- **item #2** – 5,500 positive ratings and 4,500 negative ratings: 55% +

  \- Ranking: **item #2** (score = 1000, but only 55% +) ranked above
  item #1 (score = 200, and 60% +)

Ranking – Examples…

- **WRONG SOLUTION #2: Score = (# positive ratings) / (total # ratings)**

  Can work if many ratings exists for items, but often this is not the case (e.g. new movie releases…)

  Example:

  - **item #1** – 2 positive ratings and 0 negative ratings
  - **item #2** – 100 positive ratings and 1 negative rating

  - Ranking: **item #1** (very few positive ratings) ranked above **item #2** (many positive ratings)

Ranking – Examples…

- **BETTER SOLUTION:**

  Score given by the lower bound of Wilson score confidence interval for a Bernoulli parameter

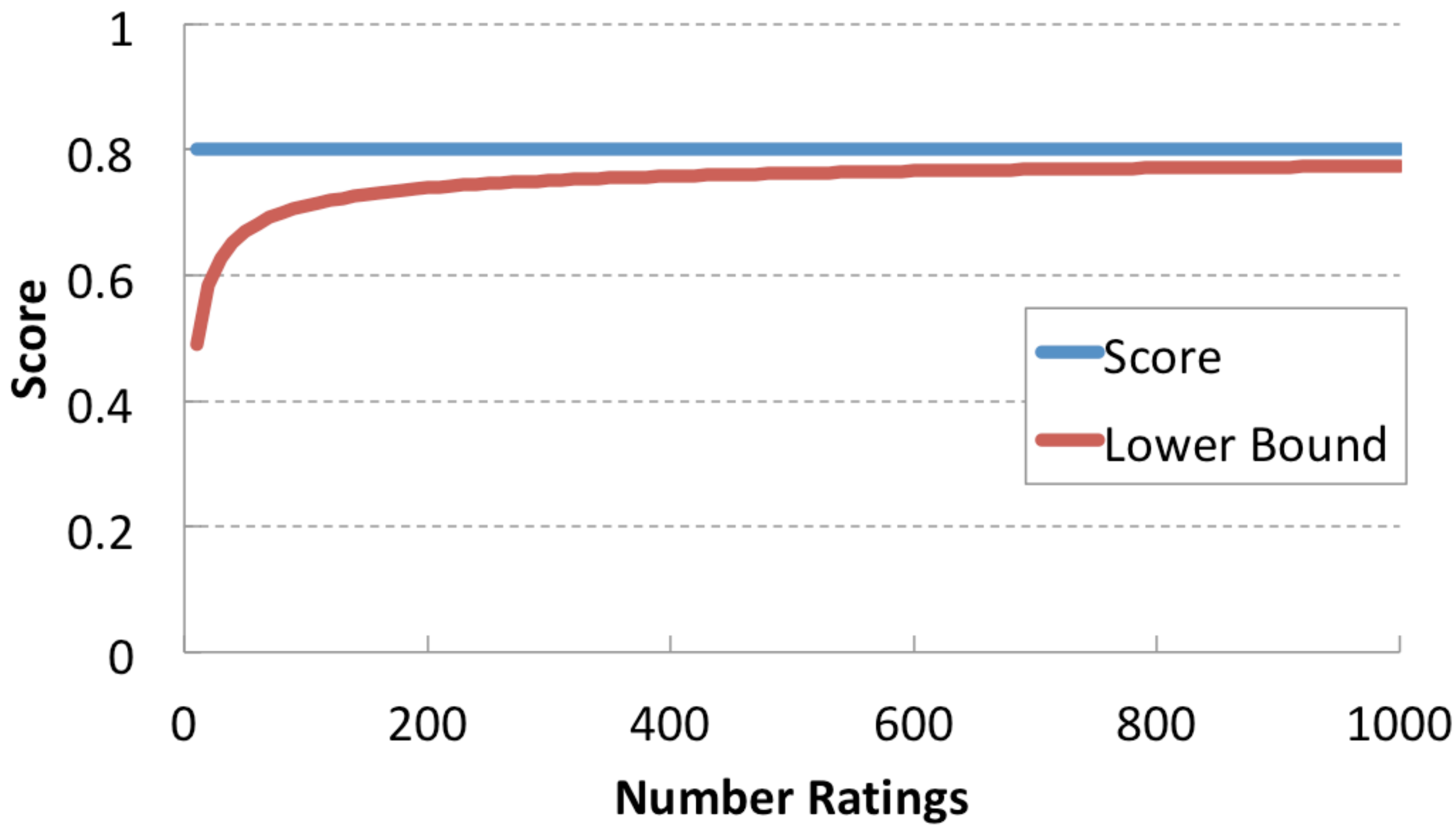
  Balances the proportion of positive ratings with the uncertainty of a small number of observations

  Confidence intervals – given the ratings I have, what is the 95% chance that the fraction of positive ratings is at least X?

- See ***How Not to Sort by Average Rating*** (also available on Moodle):

  http://www.evanmiller.org/how-not-to-sort-by-average-rating.html

Ranking – Examples…



Given the number of ratings, what is the 95% chance that the fraction of positive ratings is at least X?

Ranking – Examples…

- **WRONG SOLUTION #1: Score = (# positive ratings) – (# negative ratings)**

  Example:
  - **item #1** – 600 positive ratings and 400 negative ratings: 60% +
  - **item #2** – 5,500 positive ratings and 4,500 negative ratings: 55% +

  - Ranking: **item #2** (score = 1000, but only 55% +) ranked above
    item #1 (score = 200, and 60% +)

- **Using lower bound of 95% confidence interval**
  - **item #1** = 0.57
  - **item #2** = 0.54
  - now **item #1** ranked above **item #2**…

Hacker News Ranking

- Ranking news articles – temporal considerations important in this domain
- Different variations reported, here is one algorithm:

$$\text{score} = \frac{U - D - 1}{(T + 2)^G}$$

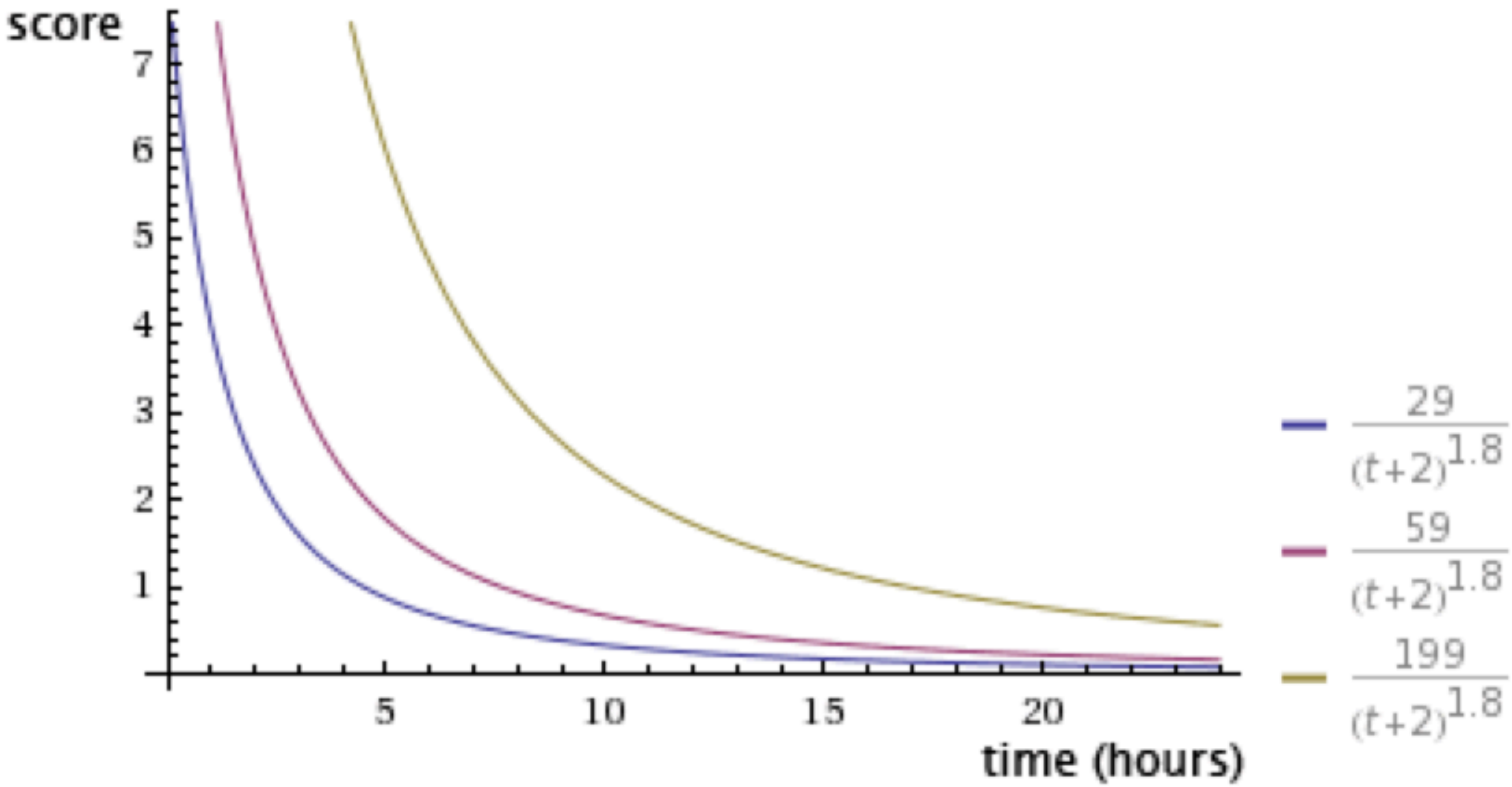
where:

  - *U* = # up votes, D = # down votes (–1 is to negate submitters vote)
  - *T* = time since submission (in hours)
  - *G* = gravity

- Gravity and time have a significant impact on scores:
  - Generally, the score decreases as *T* increases, so older items are penalised
  - The score decreases much faster for older items if gravity is increased

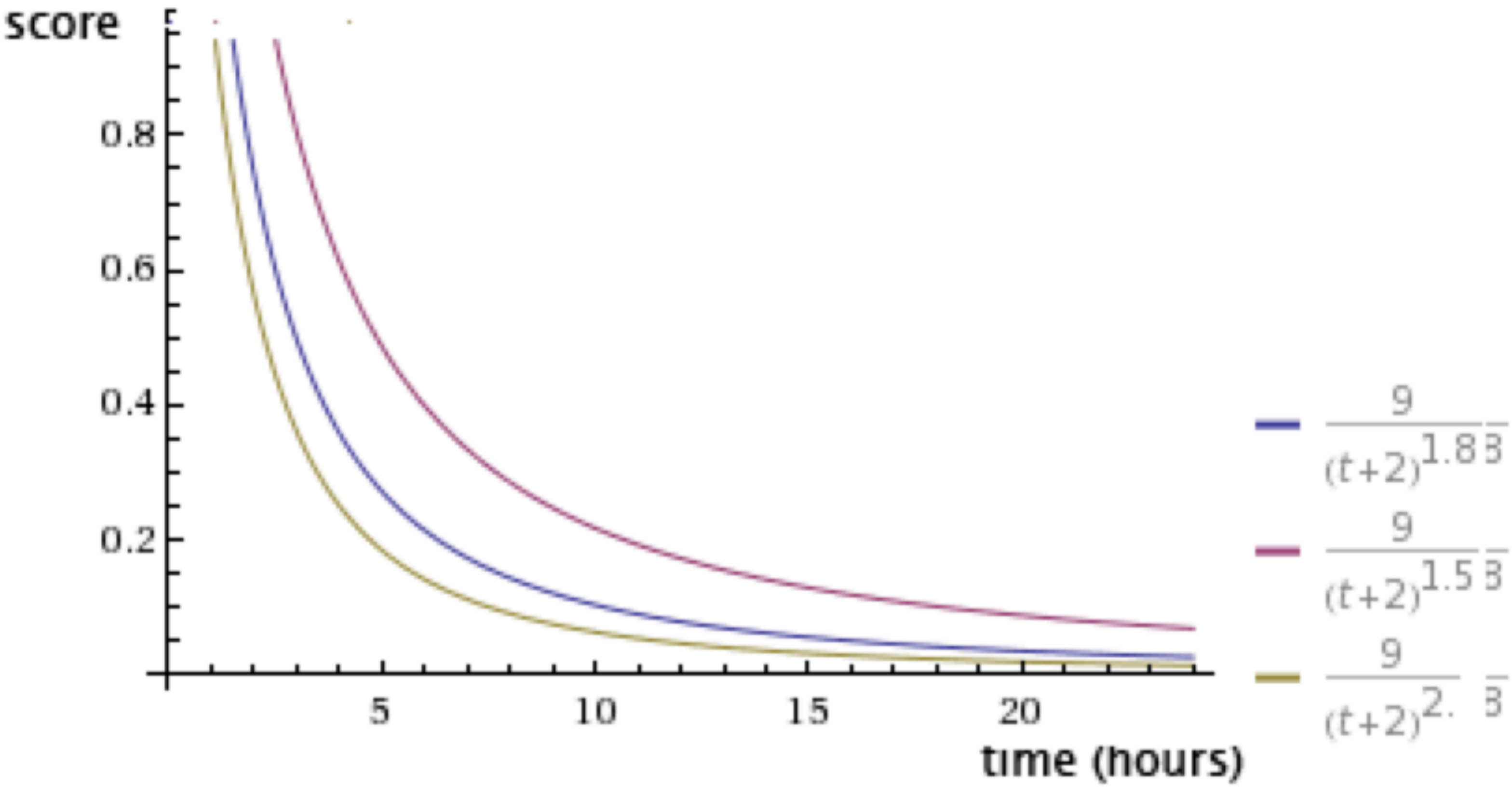
See (also available on Moodle): https://medium.com/hacking-and-gonzo/how-hacker-news-ranking-algorithm-works-1d9b0cf2c08d

Hacker News Ranking Algorithm – Score vs. Time



- Scores decrease significantly with time – day old items have a very low score, even those with a relatively high proportion of up votes…

Hacker News Ranking Algorithm – Score vs. Gravity



- Scores decrease much faster at larger values of gravity.

Reddit News Ranking

- $$\text{score} = \log_{10}\left(\max(1, |U - D|)\right) + \frac{\text{sign}(U - D)\, t}{45,000}$$

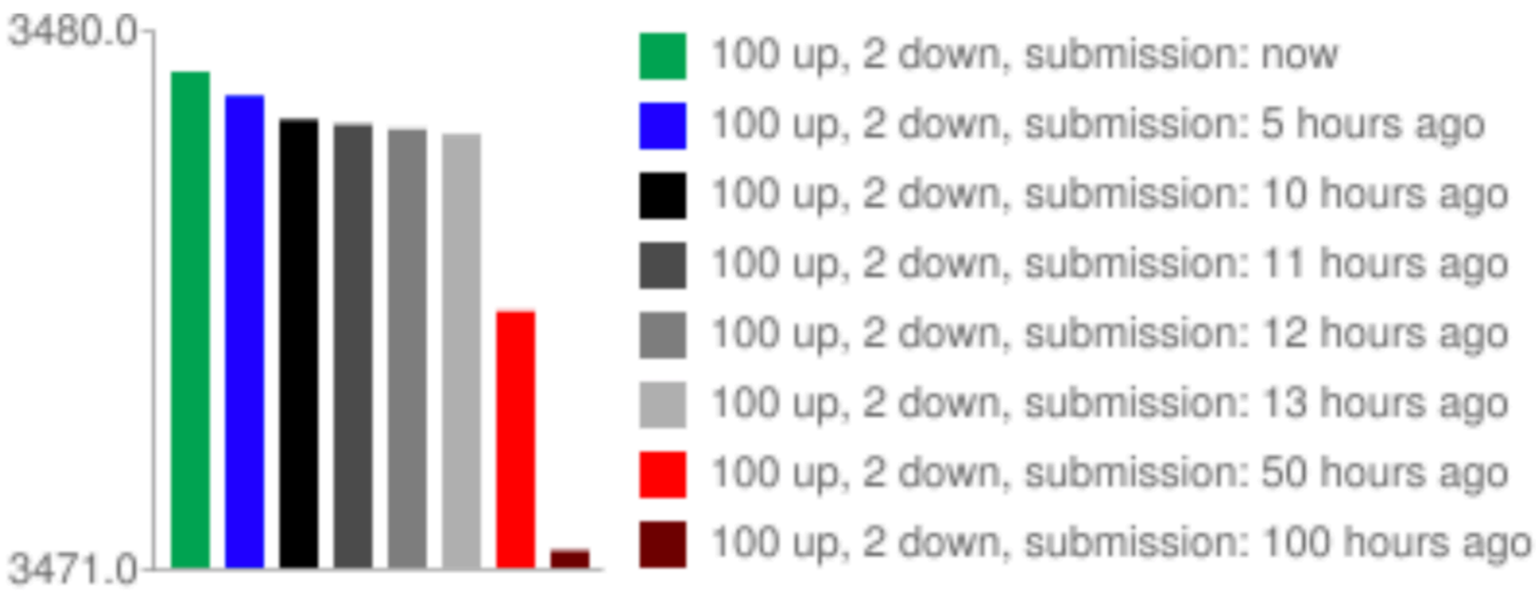
  where:

  - *U* = # up votes, D = # down votes
  - *t* = difference between post time and Reddit epoch

- The algorithm will rank newer stories above older
- Log applied to decrease contribution of later votes
- Articles with more down than up votes are rarely surfaced
- See reading on Moodle

Reddit News Ranking

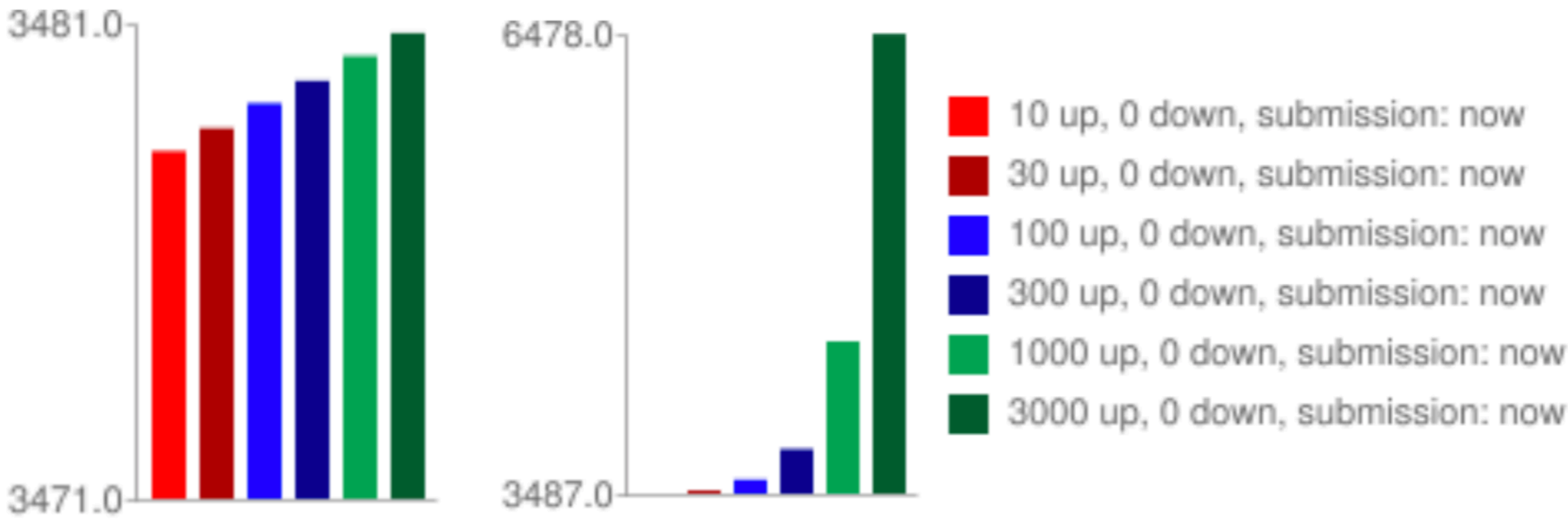
- *Submission time* has an impact on ranking – generally newer stories will rank higher than older stories:



Score vs. submission time (equal numbers of up and down votes)

Reddit News Ranking

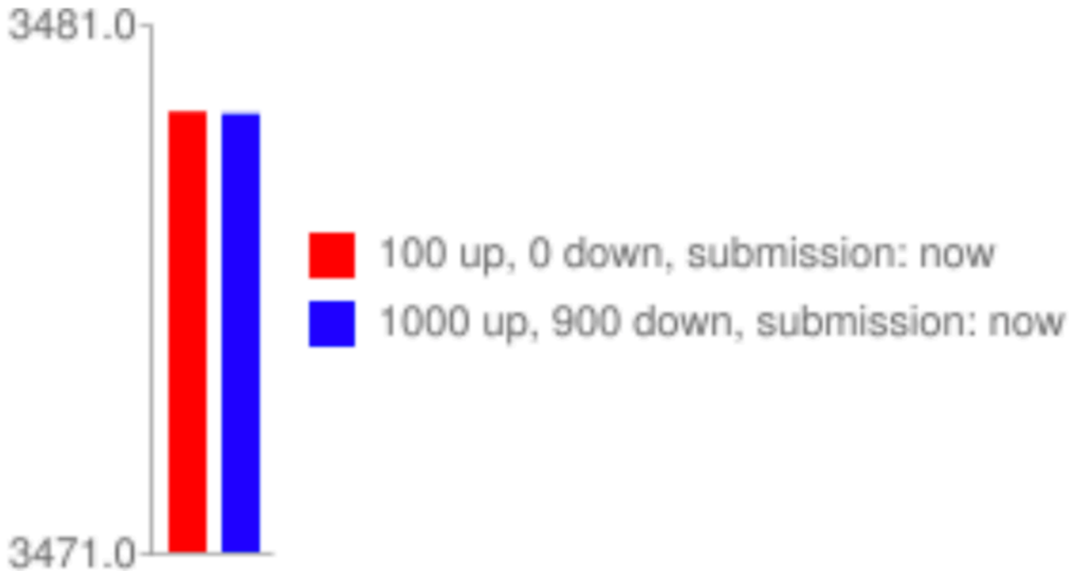
- *Log function* decreases the contribution of later votes (gives higher weight to early votes) :



Scores calculated with (left) and without (right) log function

Reddit News Ranking

- Effect of *down votes* on scores*:*
  - *C*ontroversial stories tend to attract high numbers of both up votes and down votes.
  - Scoring function biased against such stories – tend to receive lower ranking.



Effect of down votes on scores (for same net up votes and submission time)

Product Association Recommendation

- Context in recommender systems is important in a number of respects (location, time, weather, current activity, intent, domain)…
  - Not captured by ranking recommendations on simple average ratings
- Example – I have just bought a printer:
  - I do not want a recommendation for another printer (content-based?) nor a recommendation for the currently top-selling phone!
  - A good recommendation is this instance might be paper or ink cartridges
- So context in this sense is what are the other products that are most commonly associated (e.g. puchased together) with the current product
- Product association recommenders – a la Amazon…

Amazon – product association recommendations (people who bought X also bought…)

People who like X also like Y...

- Association rule mining – discovering interesting (non random) relations between items
- For example, consider supermarket transaction data:
  - {teapot, tea}, {eggs, bacon}, {apples, oranges, milk}
- A rule is defined as an implication of the form: X => Y
  - where X (antecedent) and Y (consequent) are item sets (may contain more than one item) and $X \cap Y = \{ \}$.
- Example rules:
  - {bread, butter} => {milk}
  - {teapot} => {tea}
  - Q – if {teapot} => {tea}, does {tea} => {teapot} also hold?

People who like X also like Y...

- Require large transaction datasets to discover interesting rules
- Which rules should be selected?
- Support:
  - supp(X) = percentage of transactions which contain X
- Confidence:
  - Does liking X imply liking Y?
  - conf(X=>Y) = supp(X and Y) / supp (X) **(1)**
- Sufficient?
  - What if Y is frequently purchased and has no relation to X?
  - Maybe more people like Y when X is not liked…
  - Compute: supp(!X and Y) / supp(!X) **(2)**
  - Divide **(1)** by **(2)** to give the increase in liking Y if X is liked

Example

Assume we have 3 products X, Y, and Z and 5 baskets of items (transactions). The transactions are:

basket #1: X, Y

basket #2: X, Y, Z

basket #3: X, Z

basket #4: Y, Z

basket #5: Y, Z

supp(X) = % of transactions which contain X = 3/5

supp(X and Y) = % of transactions which contain both X and Y = 2/5

supp(X and Y)/supp(X) = 2/3 **(1)**

supp(!X) = % of transactions which do not contain X = 2/5

supp(!X and Y) = % of transactions which do not contain X but do contain Y = 2/5

supp(!X and Y)/supp(!X) = 1 **(2)**

Divide **(1)** by **(2)** to get the increase in liking Y if X is liked = (2/3)/(1) = 2/3

The higher this value, the more liking X implies liking Y

People who like X also like Y...

- Alternatives to support and confidence:
  - Lift, coverage, conviction…
- Apriori algorithm:
  - Widely used approach for frequent item set mining
  - Seminal reference: Agrawal and Srikant, “Fast algorithms for mining association rules in large databases”, Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, 1994
- Other domains:
  - Movies, books, hotel rooms, etc. – people do not watch a number of movies at the same time! What is the dataset (i.e. “baskets of items”)?
    - Consider items in user profiles
    - Can take temporal considerations into account (e.g. out of season vegetables)

Non-personalised Recommendation

- Predictions:
  - Determine whether a user will like a given item
  - Non-personalised averages – can be useful, but care needs to be taken when aggregating ratings

- Recommendations:
  - Identify items that a user likes, typically presented in the form of a top-N ranked list
  - Many ad-hoc approaches – domain dependant, tuning of parameters required
  - Approaches to deal with temporal considerations, smoothing
  - Statistical approaches – confidence intervals
  - Product association recommenders provide (co-occurrence) context, useful in certain situations

Next Topic…

- Personalised recommendation:
  - Predictions and recommendations are tailored to the particular preferences of users:
    - In contrast to non-personalised approaches, where everyone receives the same predictions and recommendations regardless of personal preferences
  - Content-based recommendation:
    - Recommendations are based on content descriptions/meta data