Recommender Systems

COMP 30030 Introduction to Artificial Intelligence

Dr. Michael O'Mahony

michael.omahony@ucd.ie

Overview

- The Long Tail and Information Discovery
- Recommender Systems Overview
- Collaborative Filtering Algorithms
- Evaluation Methodology and Metrics
- Advantages and Limitations

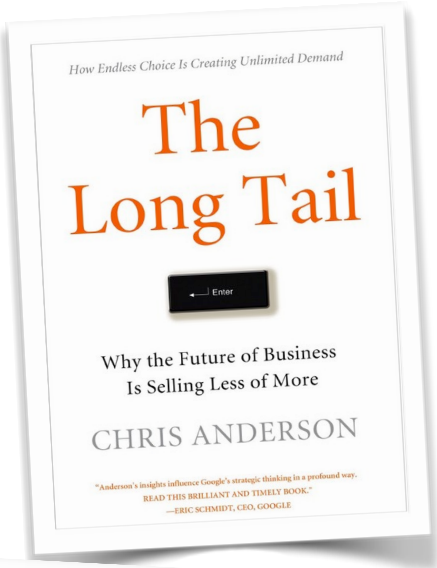
The Long Tail and Information Discovery

The Long Tail

- A new economic model for the media and entertainment industries.
- In the past, these have been hit-driven industries:
  - The average US movie theatre will not show a film unless it can attract at least 1,500 people over a two-week run; that's the screen rental cost.
- Online stores such as Amazon, iTunes, Netflix etc. carry much greater inventory levels… Provide customers with access to niche content in a manner that is revolutionising sales.
- See: *The Long Tail*, Chris Anderson.

The Long Tail

- Physical vs digital stores: finite vs infinite inventory.
- Popularity no longer has the monopoly on profitability
- Products with low demand/low sales volume can collectively make up a market share that rivals the relatively few bestsellers and blockbusters, if the store/distribution channel is large enough.
- Recommender systems help to drive demand down the long-tail – **enable findability**



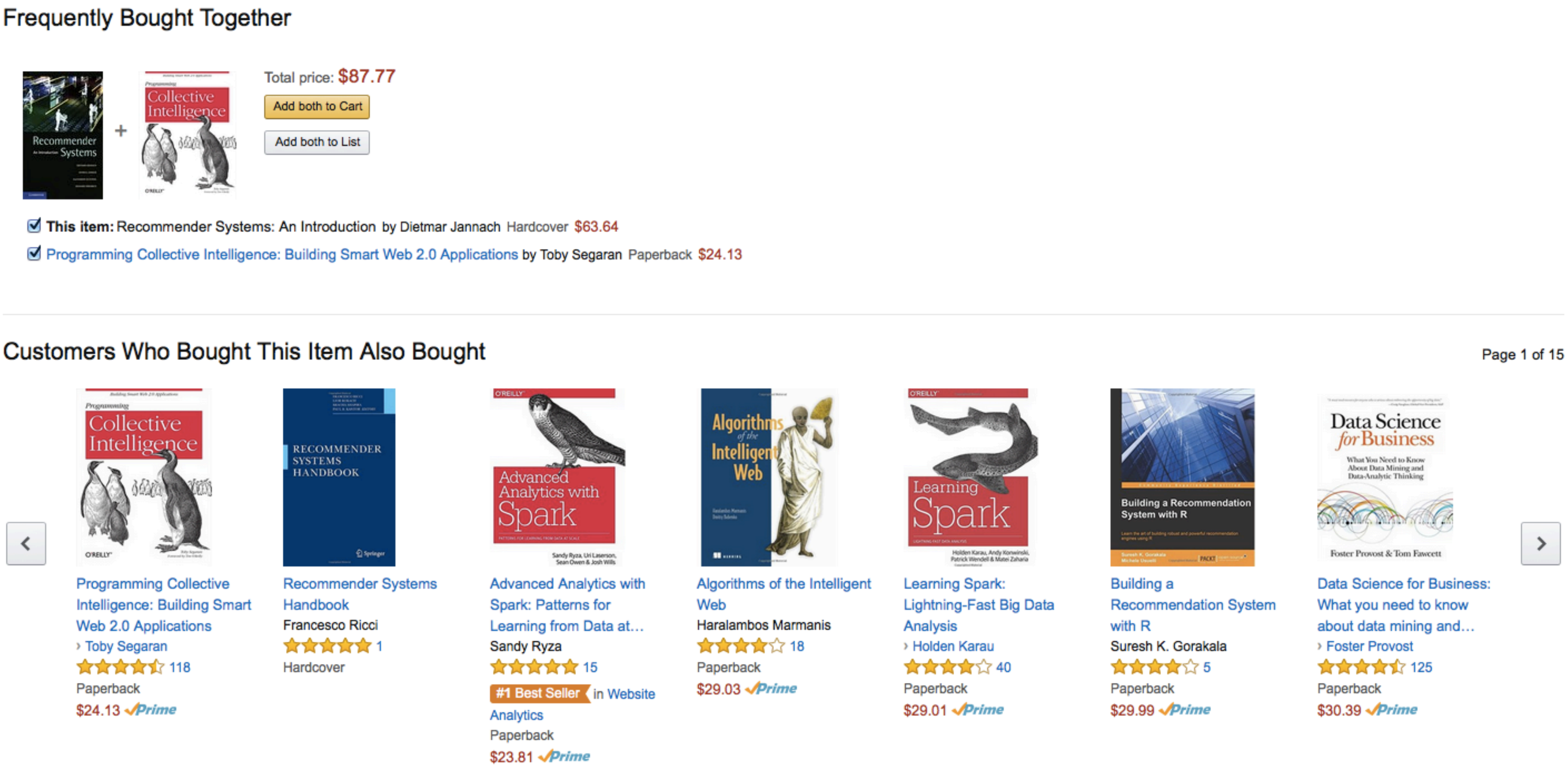
Blockbuster - A (Brief) History…

- 1985 - the first Blockbuster store opens in Dallas, Texas…
- At peak (circa 2004), Blockbuster employed about 60,000 people in over 9,000 stores…
- In 2000, the company declined an opportunity to purchase a new-ish company called Netflix for a modest $50 million…
- Fast forward to 2010… due to competition from Netflix and others, Blockbuster lost significant revenue… filed for bankruptcy…

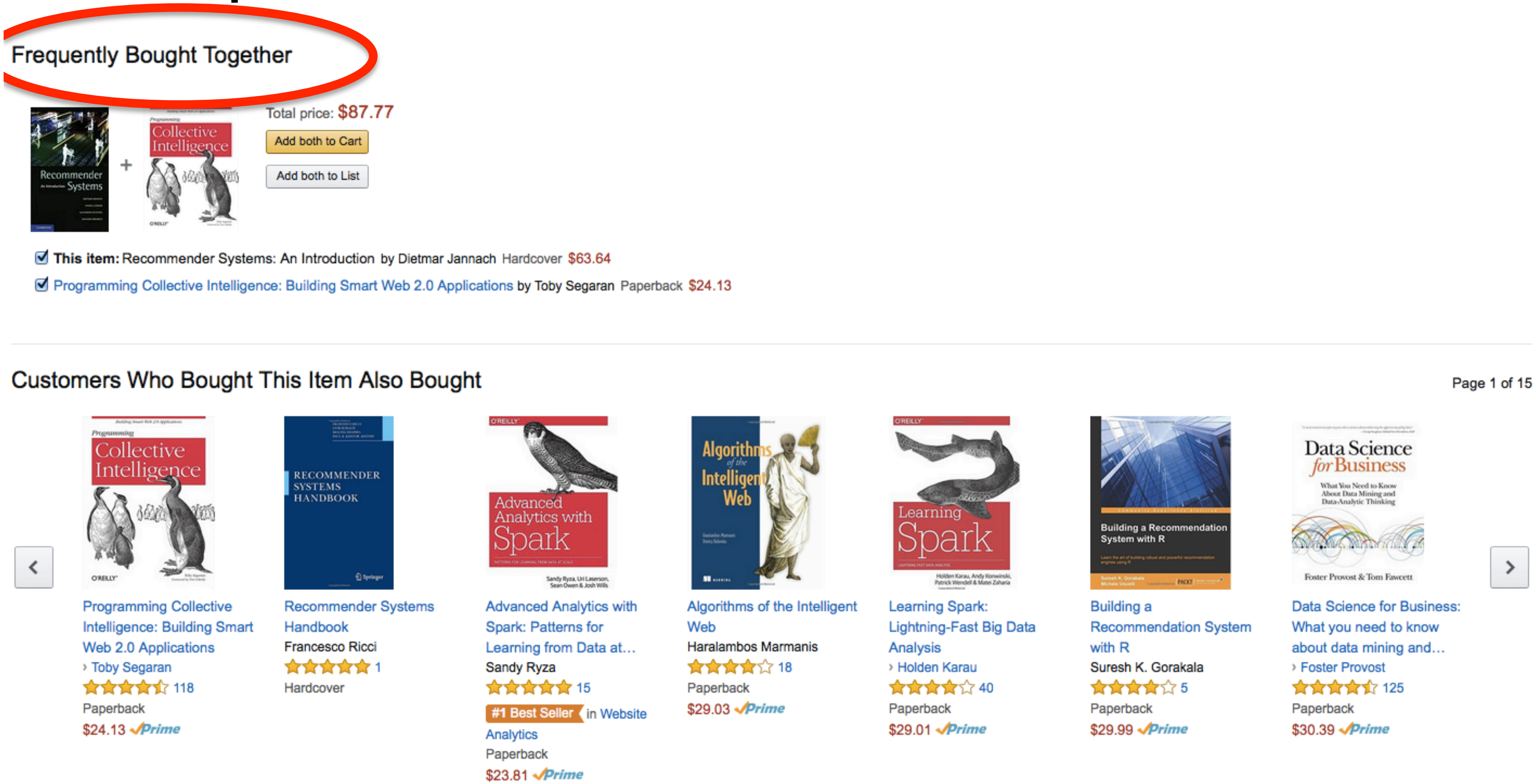
Recommender Systems – Amazon

Amazon Example…

Amazon Example…



Amazon Example…

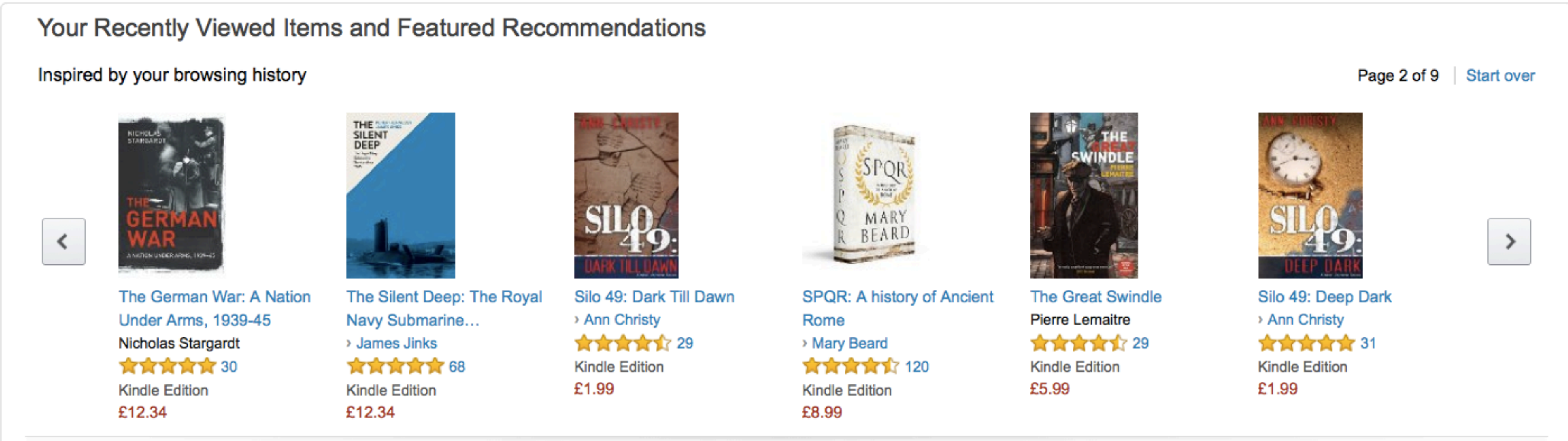


Amazon Example…

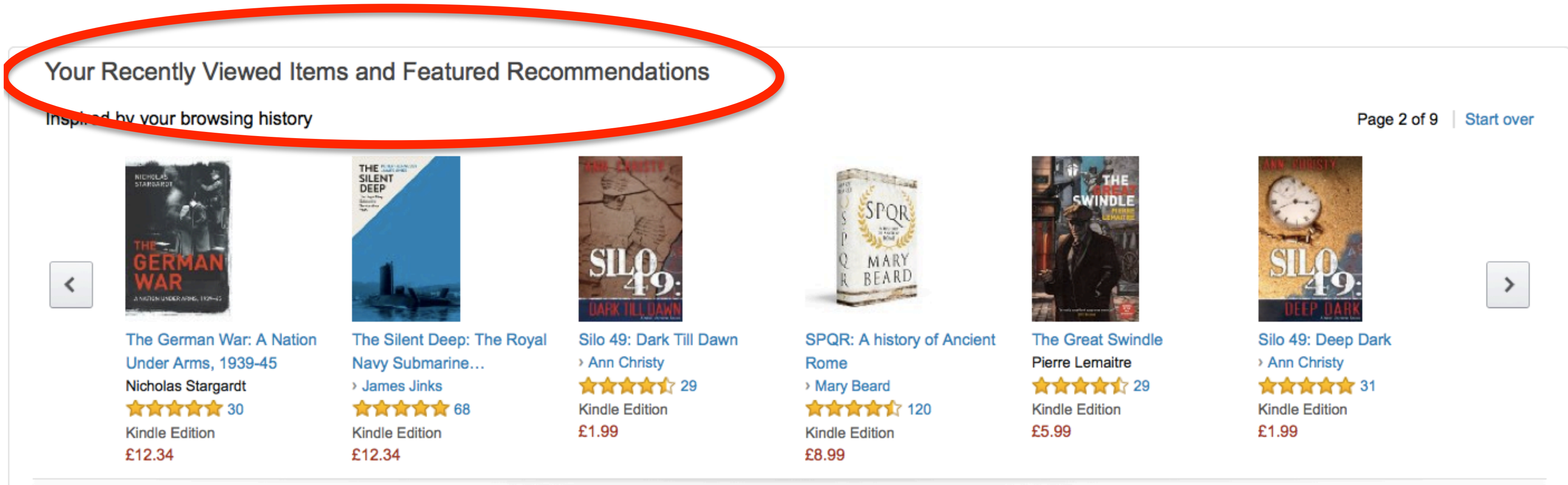
Amazon Example…

Non-personalised Recommendations

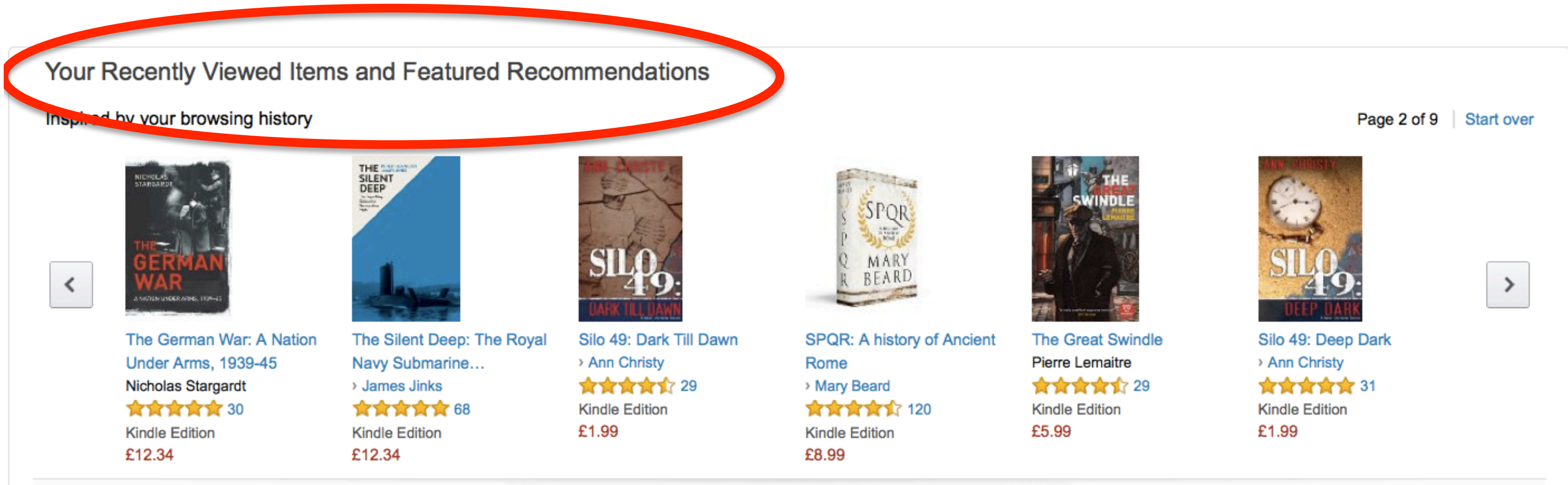
Amazon Example…



Amazon Example…



Amazon Example…



Personalised Recommendations

Recommender Systems Overview

Recommender Systems

- Recommendations can be personalised or non-personalised:
  - Tailored to the particular interests of users or generic
- Approaches to recommendation:
  - Collaborative filtering (CF), content-based, demographic, hybrid, group RS, conversational RS… This lecture, focus on CF
- Sources of recommendation knowledge:
  - Implicit/explicit user preferences, product metadata, product reviews, tweets, FB posts/likes…
- Recommendation output:
  - Ranked lists of items, predicted ratings, explanations (why are these items recommended?)

Content-based Recommendation

- Items are recommended which are are similar in content to previously selected items
- Recommendations are based on a description of the **content** of items as opposed to users' **opinions** about items
- Comparisons between items are calculated over the features associated with each item
  - For example, in the movie domain, features include actors, genres, director, plot summary…
  - Based on the range of feature values associated with a user's past choices, further movies with similar feature values are recommended
  - A *more like this* approach to recommendation

Collaborative Filtering (CF)

- CF – automates the “word-of-mouth” process
- Assists users to make choices based on the opinions of other users
- The underlying heuristic: people who agreed or disagreed on items in the past are likely to agree or disagree on future items
- Predictions and recommendations are made for users by combining the preferences of similar users in the system:
  - Note: content descriptions are not required – recommendations are based only on user preferences

Collaborative Filtering (CF)

- Let's look at a typical problem…
- Suppose we had the following preference data…

	The Quiet Man	Casino	Star Wars	Top Gun	Dallas: The Movie
Eamon	3		3	4	2
Sharon	4	1	4	2	4
John	5	2		2	5
Trisha	2	5	3		1
Mike	?	2	4	1	5

- Q – based on this preference data, would user *Mike* like or dislike *The Quiet Man*?

Collaborative Filtering (CF)

- Let's look at a typical problem…
- Suppose we had the following preference data…

	The Quiet Man	Casino	Star Wars	Top Gun	Dallas: The Movie
Eamon	3		3	4	2
Sharon	4	1	4	2	4
John	5	2		2	5
Trisha	2	5	3		1
Mike	?	2	4	1	5

- Q – based on this preference data, would user *Mike* like or dislike *The Quiet Man*?

Collaborative Filtering (CF)

- Let's look at a typical problem…
- Suppose we had the following preference data…

	The Quiet Man	Casino	Star Wars	Top Gun	Dallas: The Movie
Eamon	3		3	4	2
Sharon	4	1	4	2	4
John	5	2		2	5
Trisha	2	5	3		1
Mike	?	2	4	1	5

- Q – based on this preference data, would user *Mike* like or dislike *The Quiet Man*?

Collaborative Filtering (CF) Algorithms

User-based CF Algorithm

Collaborative Filtering (CF)

- Many CF algorithms proposed (user-based, item-based, matrix factorisation, other model-based approaches)…
- Focus on user-based CF – similar to the example we looked at earlier…
- Remember CF approaches operate over a set of user preferences; no content descriptions are required

User-based CF Algorithm

Prediction Algorithm: objective is to predict the rating of a *target item* for a given user (*active user*)

```
1. Data:
       Obtain preference data from users

2. Similarity Computation:
       Compute the similarity between the active user and all
       other users in the system which have rated the target item

3. Neighbourhood Formation:
       Select a subset consisting of the most similar users to the
       active user which have rated the target item

4. Make a prediction for the active user by aggregating the
   neighbour's ratings for the target item
```

Different approaches used in the above steps …

Preference Data

- Need to acquire (lots of) user preferences
- Types of ratings:
  - Scalar ratings: numerical ratings (e.g. 1 - 5 stars for movies) or ordinal ratings (e.g. strongly agree, agree, neutral, disagree, strongly disagree)
  - Binary ratings: e.g. votes up/down, like/dislike or agree/disagree
  - Unary ratings: e.g. purchased/not purchased a book, visited/not visited a web page (note that not purchasing a book does not imply dislike)
- Ratings are gathered explicitly or implicitly

Preference Data

- Explicit Ratings
  - Users supply ratings:
    - Generally in the form of scalar ratings (e.g. 5 point scale) or binary ratings
    - Good reflection of user preferences
  - Cost – users may tire of providing ratings
  - Users need to be convinced there is a benefit in order to make the effort
  - CF algorithms need many ratings to function accurately
- Implicit Ratings
  - Not obtained directly from user; instead, ratings are “inferred” – e.g. a user’s browsing patterns (time spent on web page, a user’s bookmarks, links followed)
  - Removes cost of explicitly gathering ratings
  - Every user interaction can potentially contribute
  - Can be combined with explicit ratings
  - But: implicit ratings are typically noisy
- Ratings are “stored” in a user-item matrix

User-item Matrix

Items (songs, news articles, movies…)

Users

	i_1	i_2			i_j										i_n
u_1	1	3	..	..	..	..	..	..	..	..	..	..	..	..	2
u_2	5	?	..	..	..	..	..	..	..	..	..	..	..	..	2
	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
u_i	..	..	..	..	$r_{i,j}$	..	..	..	..	..	..	..	..	..	..
	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
u_m	8	5	..	..	..	..	..	..	..	..	..	..	..	..	6

User Similarity

- How do we measure the similarity between users?
- A number of approaches have been considered…
- In most approaches, there needs to be an overlap between two users in order to compute their similarity:
  - No overlap – assume similarity is zero
  - Small overlap – skewed results
  - Large profiles (i.e. users who have rated many items, bots) will have overlaps with many other profiles – problematic, can lead to such profiles being included in neighbourhoods “by default”

Mean Squared Difference

- The Mean Squared Difference in the ratings between users *a* and *i* is computed as:

$$\mathrm{MSD}_{a,i} = \frac{\sum_{j \in I_a \cap I_i} (r_{a,j} - r_{i,j})^2}{|\{j : j \in I_a \cap I_i\}|}$$

  where:
  - I_a is the set of items rated by user *a*
  - $r_{a,j}$ is the rating of user *a* for item *j*

- MSD computes the *difference* in ratings between two users; convert MSD into a *similarity* metric as follows:

$$w_{a,i} = 1 - \frac{\mathrm{MSD}_{a,i}}{(r_{max} - r_{min})^2}$$

  where r_{min} and r_{max} are the minimum and maximum ratings, resp.

Mean Squared Difference

- Summations over co-rated items only
- Results in a similarity value of [0,+1]
- MSD assumes that users rate items according to similar distributions:
  - Critical vs. ‘generous’ raters? Critical raters rarely assign the maximum ratings to items…
  - So while differences in magnitudes between users’ ratings may exist, they may be highly correlated
  - Need a scale-invariant user similarity function to more accurately reflect rating patterns and to capture trends between users’ ratings...

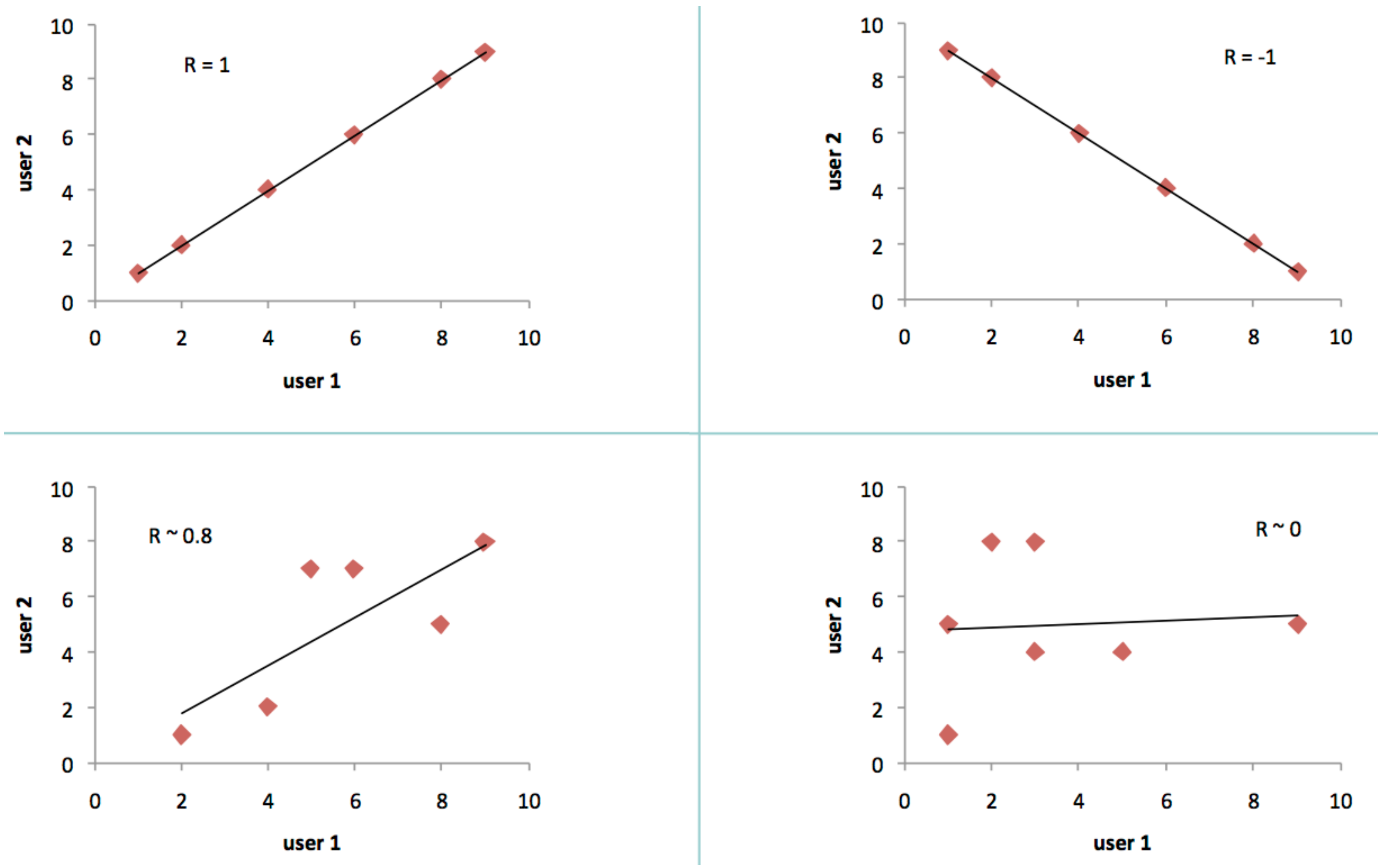
Pearson Correlation

- Given by:

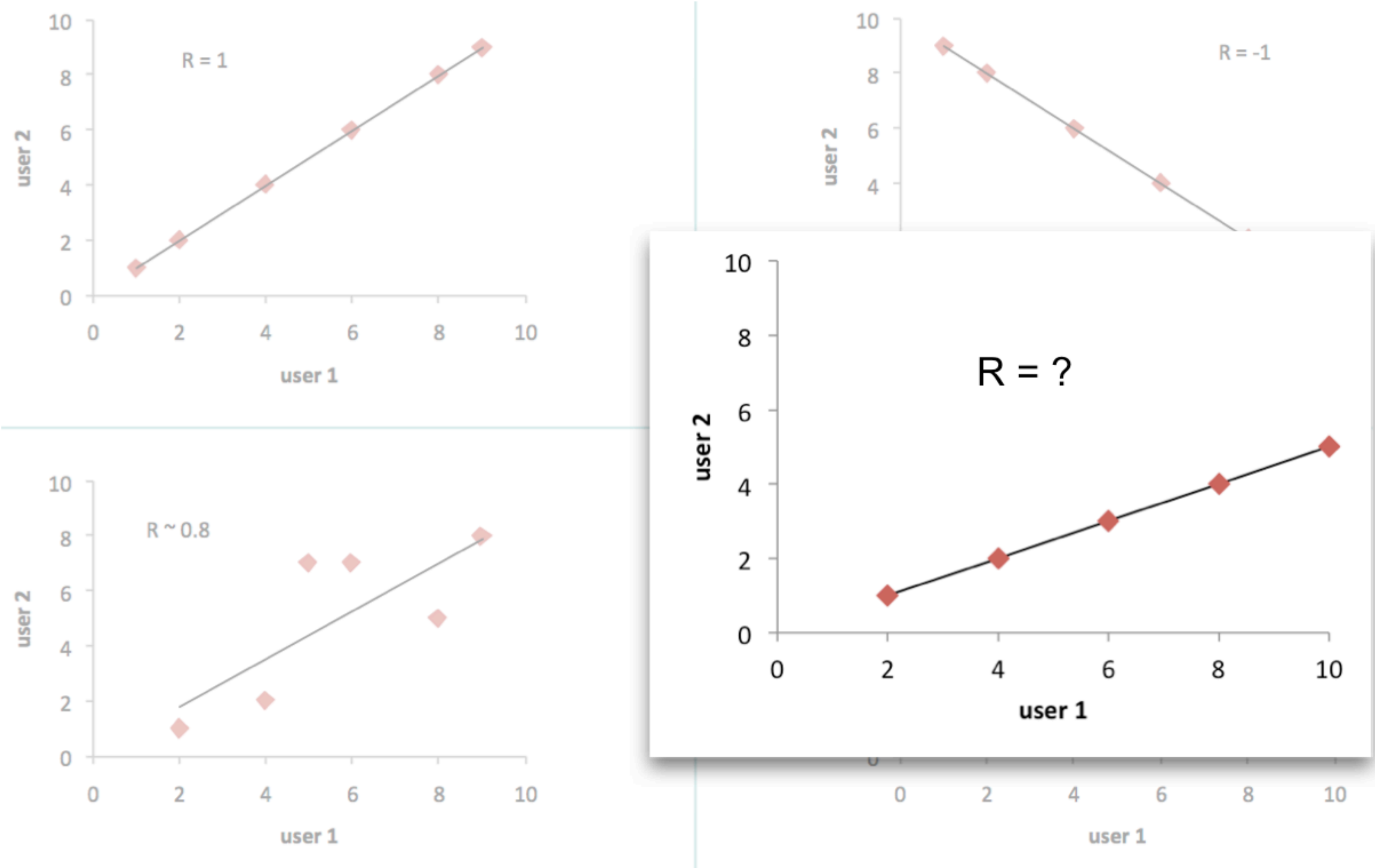
$$w_{a,i} = \frac{\sum_{j \in I_a \cap I_i} (r_{a,j} - \bar{r}_a)(r_{i,j} - \bar{r}_i)}{\sqrt{\sum_{j \in I_a \cap I_i} (r_{a,j} - \bar{r}_a)^2 \sum_{j \in I_a \cap I_i} (r_{i,j} - \bar{r}_i)^2}}$$

  where:
  - I_a is the set of items rated by user *a*
  - $r_{a,j}$ is the rating of user *a* for item *j*
  - $\bar{r}_a$ and $\bar{r}_i$ are the mean ratings of users *a* and *i* **(Note: means are computed over co-rated items only)**
- Summations over co-rated items only
- Results in a value of [-1,+1]
  - +1 indicates total agreement on co-rated items
  - 0 indicates no similarity between users
  - -1 indicates total disagreement − useful, can exploit the preferences of these users for recommendation
- Widely used similarity metric

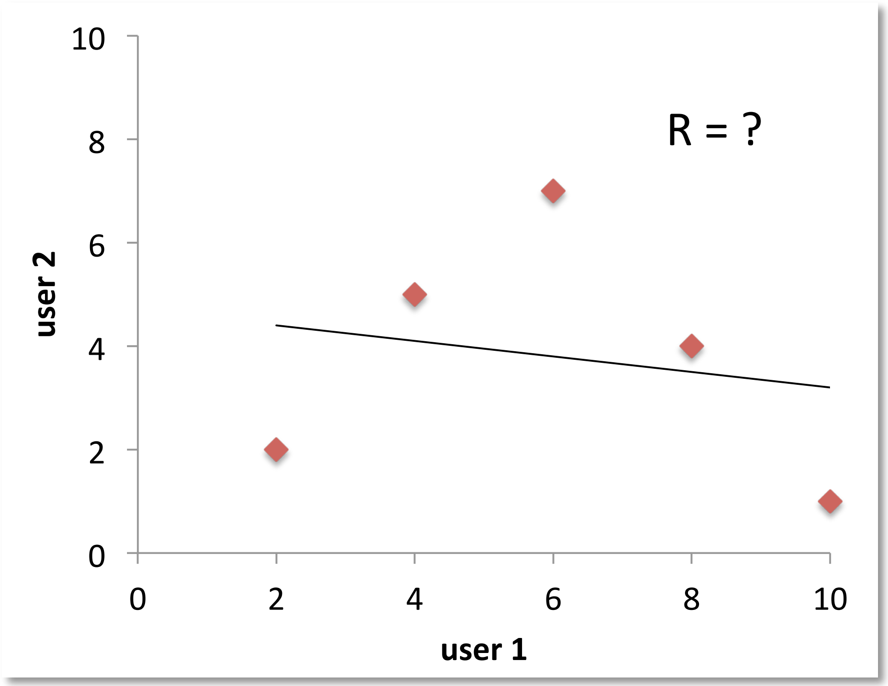
Pearson Correlation Examples



Pearson Correlation Examples



Pearson Correlation Examples



Significance Weighting

- Weights that are calculated over small numbers of co-rated items may provide an unreliable measure of the similarity between users
- Modify similarity weights based on the number of co-rated items (*n*) between users as follows:

$$w'_{a,i} = \begin{cases} w_{a,i} \times \frac{n}{N} & \text{if } n < N \\ w_{a,i} & \text{otherwise} \end{cases}$$

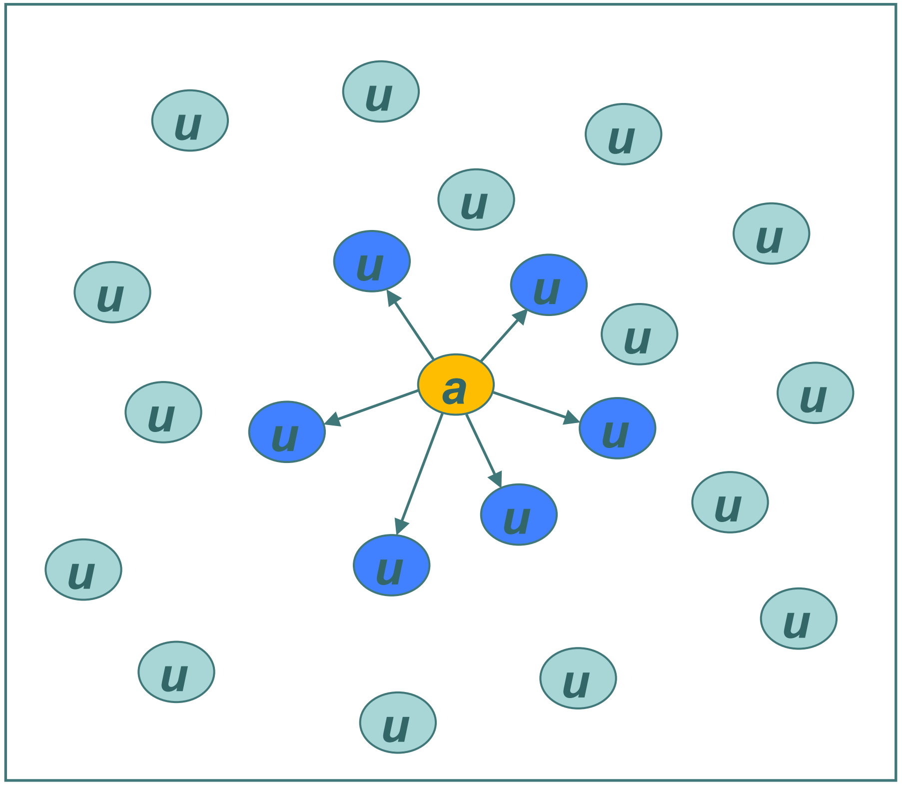
where *N* is a constant (typically set to 50)

- Improves reliability of similarity weights – similarity is modified as a function of the number of co-rated items

Neighbourhood Formation

- Select a subset of users to make predictions for the active user
- Different approaches – here consider ***k* nearest-neighbour (kNN)** approach
  - Form a neighbourhood by selecting the *k* most similar users which have rated the target item
  - Neighbourhood size – determined by experiment:
    - Large *k* can lead to reduced accuracy by diluting the influence of the most similar neighbours
    - Small *k* can result in poor accuracy for users without many (or any) close neighbours

$k = 5$



Making Predictions

- ***Resnick's*** algorithm – deviation from mean approach – target item is assigned a rating that is an adjusted form of the active user's mean rating:

$$p_{a,j} = \bar{r}_a + \frac{\sum_{i=1}^{n} w_{a,i} \; (r_{i,j} - \bar{r}_i)}{\sum_{i=1}^{n} | w_{a,i} |}$$

- Where:
  - $p_{a,j}$ is the predicted rating for the active user *a* on item *j*
  - $\bar{r}_a$ is the mean rating of the active user *a* **(Note: mean is calculated over all items user *a* has rated)**
  - $\bar{r}_i$ is the mean rating of neighbour *i* **(Note: mean is calculated over all items neighbour *i* has rated)**
  - $r_{i,j}$ is the rating of neighbour *i* for item *j*
  - $w_{a,i}$ is the similarity between the active user *a* and neighbour *i*
  - *n* is the number of neighbours

Performance Evaluation

- Live-User Analysis vs Off-line Evaluation
  - Analysis of real usage and prediction/recommendation feedback. A/B testing to evaluate different algorithms/techniques. Most comprehensive approach but expensive.
  - Offline evaluations to test core recommendation algorithm components by using existing user-rating datasets.

- Off-line Evaluation Methodology – *k*-Fold Cross Validation:
  - Randomly partition the data set into *k* subsamples.
  - Of the *k* subsamples, a single subsample is used as test data, the remaining *k*-1 subsamples as training data.
  - Repeat *k* times, using each subsample in turn as the test set.
  - Compute average results (accuracy etc.) over the test set for each fold – then compute average results across all folds.

Evaluation Metric

Mean Squared Error (RMSE) – for a given fold, RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{(a,j) \in T} (p_{a,j} - r_{a,j})^2}$$

where *T* is the test set, *N* is the number of ratings in the test set for which a prediction can be made, and $p_{a,j}$ and $r_{a,j}$ are the predicted and actual ratings for user *a*, item *j*, respectively.

RMSE penalises larger errors over smaller errors.

The smaller the RMSE value, the more accurate the recommender system.

Performance Evaluation

Other Considerations:

- **Coverage** (one definition) – the percentage of ratings in the test set for which predictions can be made
- **Serendipity** – recommend items the user is unaware of but would like
- **Diversity** – ensuring that top-N lists are not comprised of only "similar" items
- **Site performance metrics** – tracking system throughput (items purchased or downloaded, the number of active users etc.)
- **Robustness** – are CF systems vulnerable to manipulation? (**Yes!**)

CF – Advantages

Advantages

- **Quality & Taste**:
  - Human beings have the ability to analyse information based on quality and taste
  - Since CF operates on user preferences, it inherits this ability
  - Other filtering techniques are less able or unable to quantify the quality that is inherent in certain items – e.g. a content-based search may return relevant documents, but some of these may be actually disliked by users…
- **Item Features**:
  - CF algorithms make predictions / recommendations based solely on user preferences
  - Do not rely on any item features that need to be pre-determined and extracted prior to filtering
  - Thus, CF algorithms can be readily implemented in complex domains which contain items that are difficult to analyse by automated processes

CF – Advantages

- **Serendipitous Recommendations**:
  - CF algorithms have the ability to provide serendipitous recommendations – i.e. recommend items the user likes and is unaware of (and unlikely to find unless they are recommended)

<u>Limitations</u>

- **Cold-Start Problem**:
  - CF algorithms rely on relationships between users and items in order to make predictions / recommendations
  - In new applications, ratings are missing / sparse
  - In some applications, new users are required to rate a subset of items…

CF – Limitations

- **Early Rater Problem**:
  - This issue applies to predictions that are sought for items that are new or only recently added to a system.
  - For such items, predictions are problematic, since there are no (or few) ratings contained in the system on which to calculate predictions
  - Similarly, newly registered users are problematic for even established systems, until they have rated at least some of the available items
- **Sparsity Problem**:
  - Typically, recommender system datasets are very sparse, because users will have assigned ratings to only a relatively small number of items (<< 1%).
  - This is particularly true for e–commerce systems, where very large numbers of items are frequently offered for sale (e.g. consider Amazon.com)
  - Thus it may not be possible to make accurate (or any) predictions / recommendations for certain users / items

CF – Limitations

- **Scalability**
  - User-based CF algorithms suffer from serious scalability problems:
    - Computations grow with both the number of users and items
    - With potentially millions of users and items, typical real-world systems running user-based CF algorithms will suffer serious scalability problems
  - Solutions have been proposed to improve scalability performance:
    - Cluster the database into groups of like-minded users from which all neighbours are subsequently drawn…
    - Model-based approaches – item-based CF, matrix factorisation approaches …

Summary

- The long tail and information discovery
- Recommender systems algorithms: collaborative filtering. content-based (many other approaches exist…)
- Focused on the one particular collaborative filtering algorithm – user-based CF
- Evaluation methodology and metrics
- Advantages and limitations