Recommender systems:
Part 1

El Habib NFAOUI (elhabib.nfaoui@usmba.ac.ma)

2020-2021

Course outline

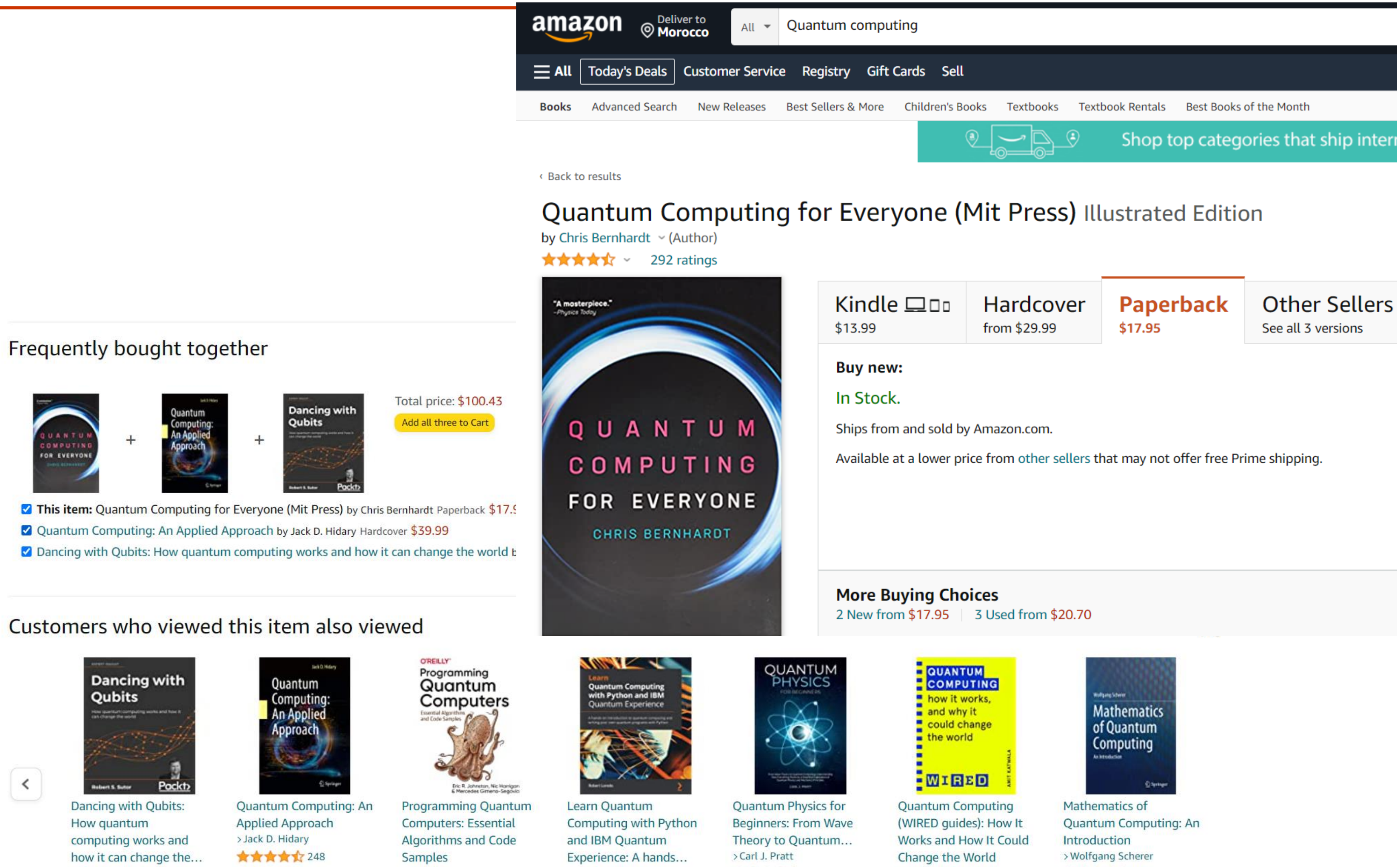
- Introduction
- Content-Based recommendation
- Collaborative filtering (or collaborative recommendations)
- Evaluation of the recommendation systems
- Demos and Hands-on exercises

1. Introduction

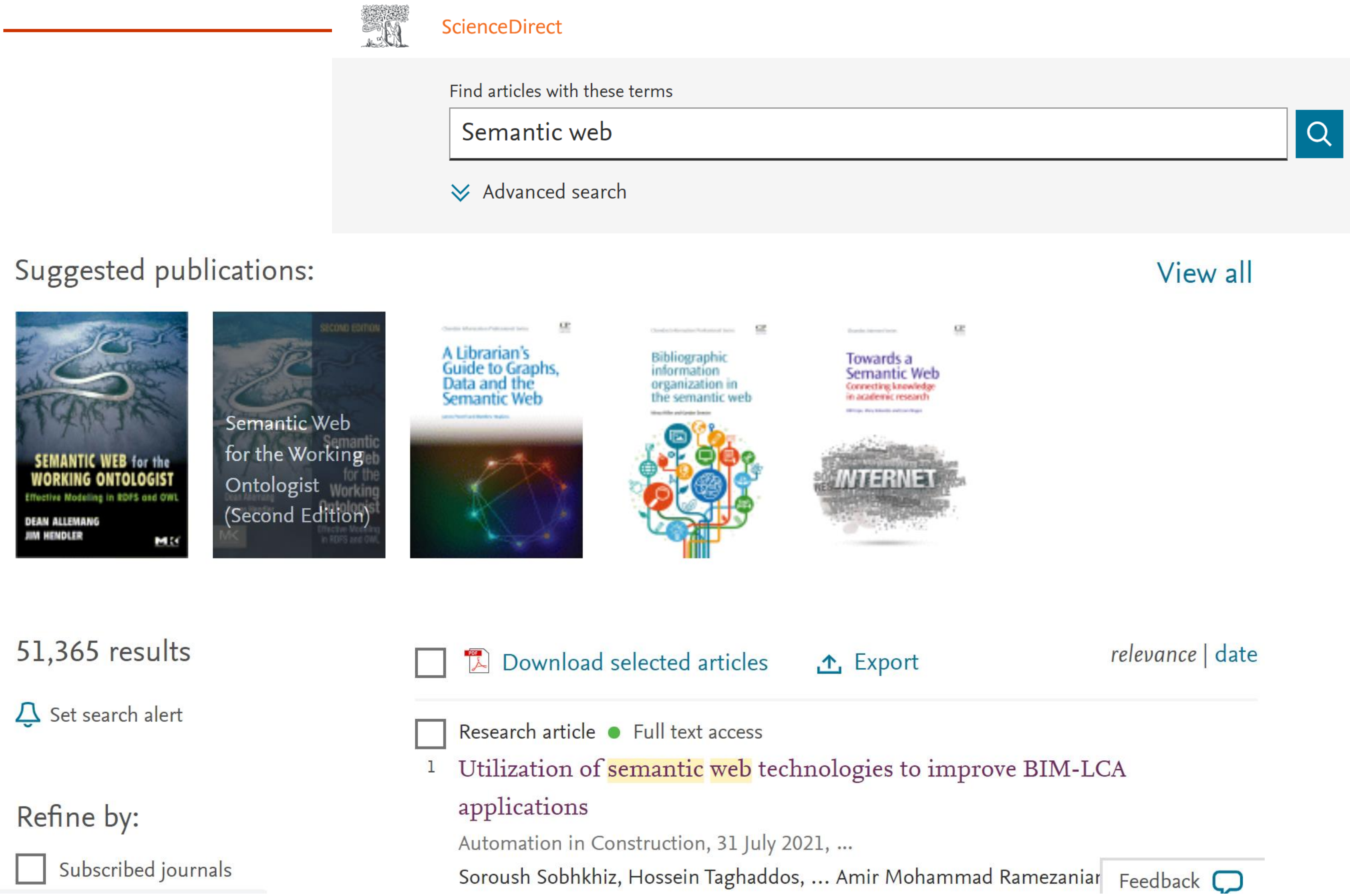
- Recommender systems are widely used on the Web for recommending products and services to users. Most e-commerce Web sites have such systems. These systems serve two important functions.
  - First, **they help users deal with the information overload** by giving them personalized recommendations. For example, given thousands of movies, a recommender system selects and recommends some movies to each user that he/she will most likely enjoy watching.
  - Second, they **help businesses make more profits**.

- Due to the problem rich nature and abundance of applications, numerous research papers have been published on recommender systems in the fields of computer science, information systems, marketing, and management science.

Example 1: Amazon Recommendations



Example 2: ScienceDirect Recommendations



The value of recommendations

- Netflix: 2/3 of the movies watched are recommended
- Google News: recommendations generate 38% more clickthrough
- Amazon: 35% sales from recommendations
- Choicestream: 28% of the people would buy more music if they found what they liked.

Data acquisition (Sources of information): What data to use?

1. Explicit Data
   - Customer Ratings
   - Feedback
   - Demographics
   - Physiographics
   - Ephemeral Needs

2. Implicit Data
   - Purchase History
   - Click or Browse History

3. Product Information
   - Product Taxonomy
   - Product Attributes
   - Product Descriptions

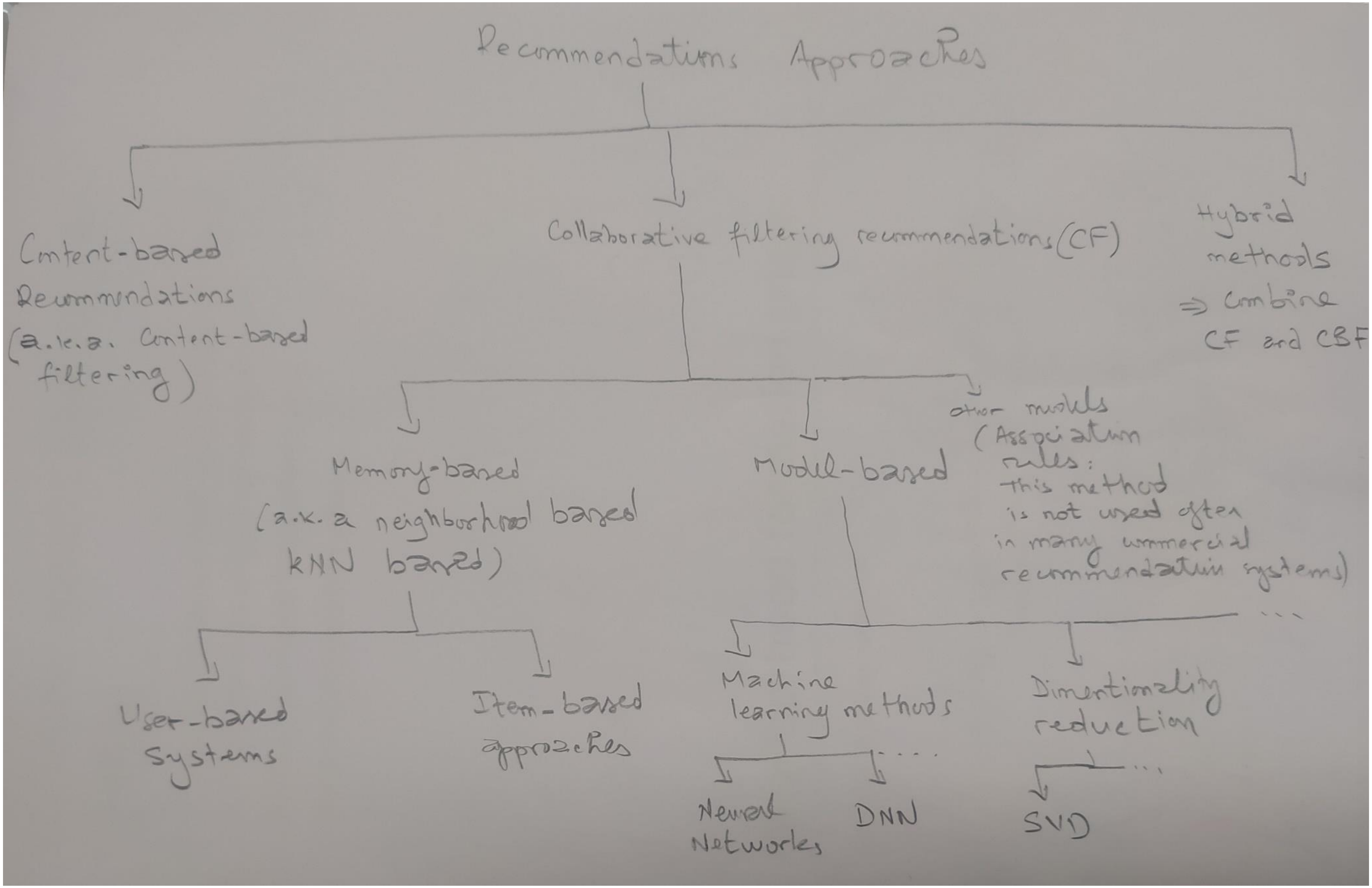
Data

- User behaviors data

Behavior	User	Size
Page view	All user	Very Large
Watch video	All user	Large
Favorite	Register user	Middle
Vote	Register user	Middle
Add to playlist	Register user	Small
Facebook like	Register user	Small
Share	Register user	Small
Review	Register user	Small

- User profiles/preferences

2. Recommendations Approaches



2. Recommendations Approaches

- There are two basic approaches to recommendations:
  - **Content-based recommendations**: The user will be recommended items **similar to the ones the user preferred in the past**.
  - **Collaborative filtering** (or **collaborative recommendations)**: The user will be recommended items **that people with similar tastes and preferences liked in the past**.

  The simplest and original implementation of this approach [93] recommends to the active user the items that other users with similar tastes liked in the past. The similarity in taste of two users is calculated based on the similarity in the rating history of the users.

  - There are also many approaches that **combine** collaborative and content based methods. These methods are called the **hybrid** methods. They typically work in one of the following ways :

  1. Implementing collaborative filtering and content-based methods separately and combining their predictions just like ensemble techniques in classification learning
  2. Incorporating some content-based characteristics into a collaborative approach
  3. Incorporating some collaborative characteristics into a content-based approach
  4. Constructing a general unifying model that incorporates both content based and collaborative characteristics

3. Content-based recommendations

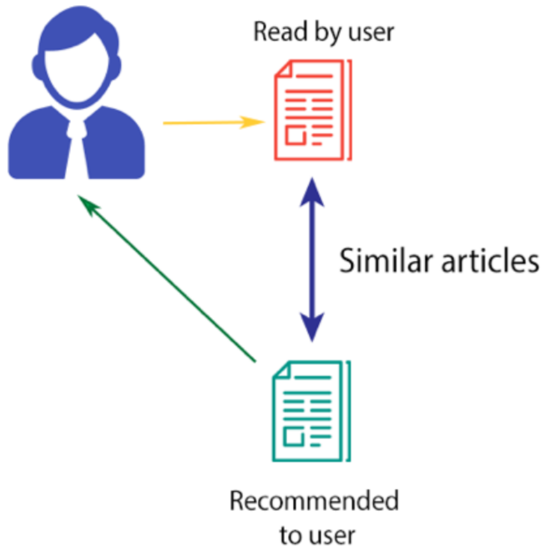
2. **Content Based Systems** recommend items similar to those a user has liked (browsed/purchased) in the past.

OR

Recommendations are based on the content of items rather on other user's opinion.

User Profiles: Create user profiles to describe the types of items that user prefers.

e.g. User1 likes sci-fi, action and comedy.



Methods of a aggregating inputs

- Content-based filtering
  - recommendations based on item descriptions/features, and profile or past behavior of the "target" user only.
- Collaborative filtering
  - look at the ratings of like-minded users to provide recommendations, with the idea that users who have expressed similar interests in the past will share common interests in the future.

4. Collaborative Filtering

- The key characteristic of CF is that it predicts the utility of items for a particular user based on the items previously rated or purchased by other like-minded users. It often utilizes only consumer–product interaction data and ignores consumer and product attributes.
- Collaborative recommender systems make suggestions of items on **the basis of similarities in consumption behavior of users**. This approach is considered **more powerful than content-based filtering** and remains predominant in RS research.

- Collaborative filtering methods can be grouped in the two general classes of **memory-based** and **model-based** methods:

  - **Memory-based** use the ratings to compute similarities between users or items (the "memory" of the system) that are successively exploited to produce recommendations.
  - **Model-based** use the ratings to estimate or learn a model and then apply this model to make rating predictions.

4. 1 Collaborative Filtering Methods

- Collaborative filtering methods can be grouped in the two general classes of **memory-based** and **model-based** methods.
  - In **memory-based (a.k.a.** neighborhood based, K-Nearest Neighbor (kNN) based, or heuristic-based) collaborative filtering, the user-item ratings stored in the system are directly used to predict ratings for new items. This can be done in two ways known as user-based or item-based recommendation:
    - **User-based systems**, such as GroupLens, Bellcore video, and Ringo, evaluate the interest of a user u for an item I using the ratings for this item by other users, called neighbors, that have similar rating patterns. The neighbors of user u are typically the users v whose ratings on the items rated by both *u* and *v*, are most correlated to those of *u*.
    - **Item-based approaches**, on the other hand, predict the rating of a user *u* for an item *i* based on the ratings of *u* for items similar to *i*. In such approaches, two items are similar if several users of the system have rated these items in a similar fashion.

4. 1 Collaborative Filtering Methods

- In contrast to neighborhood-based systems, which use the stored ratings directly in the prediction, **model-based approaches** use these ratings **to learn a predictive model**. The general idea is to model the user-item interactions with factors representing latent characteristics of the users and items in the system, like the preference class of users and the category class of items. This model is then trained using the available data, and later used to predict ratings of users for new items. Model-based approaches for the task of recommending items are numerous and include Singular Value Decomposition, Machine learning, Deep Neural Networks, Bayesian Clustering, Latent Semantic Analysis, Latent Dirichlet Allocation, Maximum Entropy, and others.

4.2 Memory-based Collaborative Filtering

- Memory-based approach (which is also called the kNN approach) utilizes the **entire user-item database** (the utility matrix) to generate predictions directly, i.e., there is **no model building**.
- Once the k nearest “neighbors” are found, predictions are made based on some kind of aggregation of the values from these neighbors.
- Memory-based methods are widely used in industry (many commercial systems).

2			4	5	
5		4			1
		5		2	
	1		5		4
		4			2
4	5		1		

Fig. 1. Example of a Utility Matrix

	user	Toy Story (1995);1	GoldenEye (1995);2	Four Rooms (1995);3	Get Shorty (1995);4	Copycat (1995);5	Twelve Monkeys (1995);7	Babe (1995);8	Dead Man Walking (1995);9	Richard III (1995);10	...	Cool Runnings (1993);1035	Hamlet (1996);1039
0	1	5	3	4	3	3	4	1	5	3	...	0	0
1	2	4	0	0	0	0	0	0	0	2	...	0	0

4.2.1 User-based Collaborative Filtering

						sim(u,v)
2			4	5		NA
5		4			1	
		5		2		
	1		5		4	
		4			2	
4	5		1			NA

Similarity between the target user u and user v

Target user u (also called a **visitor**)

4.2.1 User-based Collaborative Filtering

- A typical **user-based kNN collaborative filtering method** consists of two primary phases:
  - **Neighborhood formation phase**
  - **Recommendation phase**.

Neighborhood formation phase:

Pearson's correlation coefficient:

$$sim(\mathbf{u},\mathbf{v}) = \frac{\sum_{i\in C}(r_{\mathbf{u},i}-\bar{r}_{\mathbf{u}})(r_{\mathbf{v},i}-\bar{r}_{\mathbf{v}})}{\sqrt{\sum_{i\in C}(r_{\mathbf{u},i}-\bar{r}_{\mathbf{u}})^2}\sqrt{\sum_{i\in C}(r_{\mathbf{v},i}-\bar{r}_{\mathbf{v}})^2}}$$

where C is the set of items that are co-rated by users **u** and **v** (i.e., items that have been rated by both of them), $r_{\mathbf{u},i}$ and $r_{\mathbf{v},i}$ are the ratings (or weights) given to item i by the target user **u** and a possible neighbor **v** respectively, and $\bar{r}_{\mathbf{u}}$ and $\bar{r}_{\mathbf{v}}$ are the average ratings (or weights) of **u** and **v** respectively. (

Recommendation phase:

- Once the k nearest “neighbors” are found, predictions are made based on some kind of aggregation of the values from these neighbors.

- Below the formula to compute the rating prediction of item *i* for target user **u** (there are other possible formulas in the literature for this purpose):

$$p(\mathbf{u}, i) = \bar{r}_{\mathbf{u}} + \frac{\sum_{\mathbf{v} \in V} sim(\mathbf{u}, \mathbf{v}) \times (r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})}{\sum_{\mathbf{v} \in V} |sim(\mathbf{u}, \mathbf{v})|}$$

where V is the set of k similar users,

$r_{\mathbf{v},i}$ is the rating of user **v** given to item i,

$\bar{r}_{\mathbf{u}}$ and $\bar{r}_{\mathbf{v}}$ are the average ratings of user **u** and **v** respectively to compensate for subjective judgment (some users are generous and some are picky).

The idea is that different users may have different “baselines” around which their ratings are distributed.

sim(**u**, **v**) is the Pearson correlation described above.

Recommendation phase:

- The formula basically computes the preference of all the neighbors weighted by their similarity and then adds this to the target user's average rating.

- Note that in case the utility matrix is **so sparse that no neighbors are found,** the mean rating of the user is predicted. Also, it may happen that the predicted rating is beyond 5 or below 1, so in such situations the predicted rating is set to 5 or 1 respectively.

Recommendation phase:

- Once the ratings are predicted, we simply choose **those highly rated items to recommend to the user**.

User-based CF: Example

Utility matrix:

						sim(u,v)
2			4	5		NA
5		4			1	
		5		2		
	1		5		4	
?	?	4	?	?	2	
4	5		1			NA

Neighborhood formation phase:

						sim(u,v)
2			4	5		NA
5		4			1	0.87
		5		2		
	1		5		4	
		4			2	
4	5		1			NA

						sim(u,v)
2			4	5		NA
5		4			1	0.87
		5		2		1
	1		5		4	
		4			2	
4	5		1			NA

						sim(u,v)
2			4	5		NA
5		4			1	0.87
		5		2		1
	1		5		4	-1
		4			2	
4	5		1			NA

Recommendation phase:

						sim(u,v)
2			4	5		NA
5		4			1	0.87
		5		2		1
	1		5		4	-1
3.51*	3.81*	4	2.42*	2.48*	2	
4	5		1			NA

4.2.2 Item-based Collaborative Filtering

This approach is conceptually the same as user-based CF except that the similarity is calculated on the items rather than the users. Since most of the time the number of users can become much larger than the number of items, this method offers a more scalable recommendation system because the items' similarities can be precomputed and they will not change much when new users arrive (if the number of users N is significantly large).

4.3 Evaluation of the recommendation systems

- The evaluation of a recommendation system can be executed:
  - **Offline** (using only the data in the utility matrix)
  - **or Online (using the utility matrix data and the new data provided in real time by each user using the website)**.

- Two **offline** tests often used to evaluate recommendation systems:
  - Root Mean Square Error on ratings (RMSE)
  - and ranking accuracy.

- For all the evaluations in which k-fold cross-validation is applicable, a k-fold cross-validation can be performed to obtain more objective results. The utility matrix will then be divided into k folds.