Collaborative Filtering

To address some of the limitations of content-based filtering, collaborative filtering uses *similarities between users and items simultaneously* to provide recommendations. This allows for serendipitous recommendations; that is, collaborative filtering models can recommend an item to user A based on the interests of a similar user B. Furthermore, the embeddings can be learned automatically, without relying on hand-engineering of features.

A Movie Recommendation Example

Consider a movie recommendation system in which the training data consists of a feedback matrix in which:

- Each row represents a user.
- Each column represents an item (a movie).

The feedback about movies falls into one of two categories:

- **Explicit**— users specify how much they liked a particular movie by providing a numerical rating.
- **Implicit**— if a user watches a movie, the system infers that the user is interested.

To simplify, we will assume that the feedback matrix is binary; that is, a value of 1 indicates interest in the movie.

When a user visits the homepage, the system should recommend movies based on both:

- similarity to movies the user has liked in the past
- movies that similar users liked

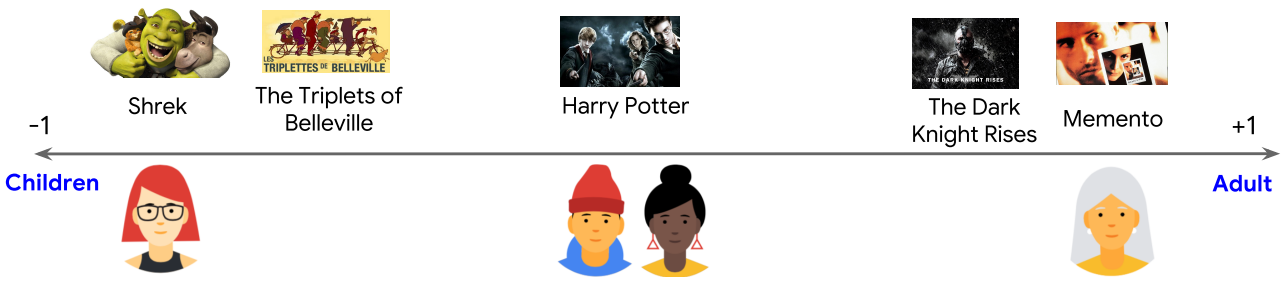
For the sake of illustration, let's hand-engineer some features for the movies described in the following table:

Movie	Rating (https://wikipedia.org/wiki/Motion_Picture_Association_of_Ame

Movie	Rating (https://wikipedia.org/wiki/Motion_Picture_Association_of_Ame
The Dark Knight Rises (http://www.imdb.com/title/tt1345836)	PG-13
Harry Potter and the Sorcerer's Stone (http://www.imdb.com/title/tt0241527/)	PG
Shrek (http://www.imdb.com/title/tt0126029/)	PG
The Triplets of Belleville (http://www.imdb.com/title/tt0286244)	PG-13
Memento (http://www.imdb.com/title/tt0209144/)	R

1D Embedding

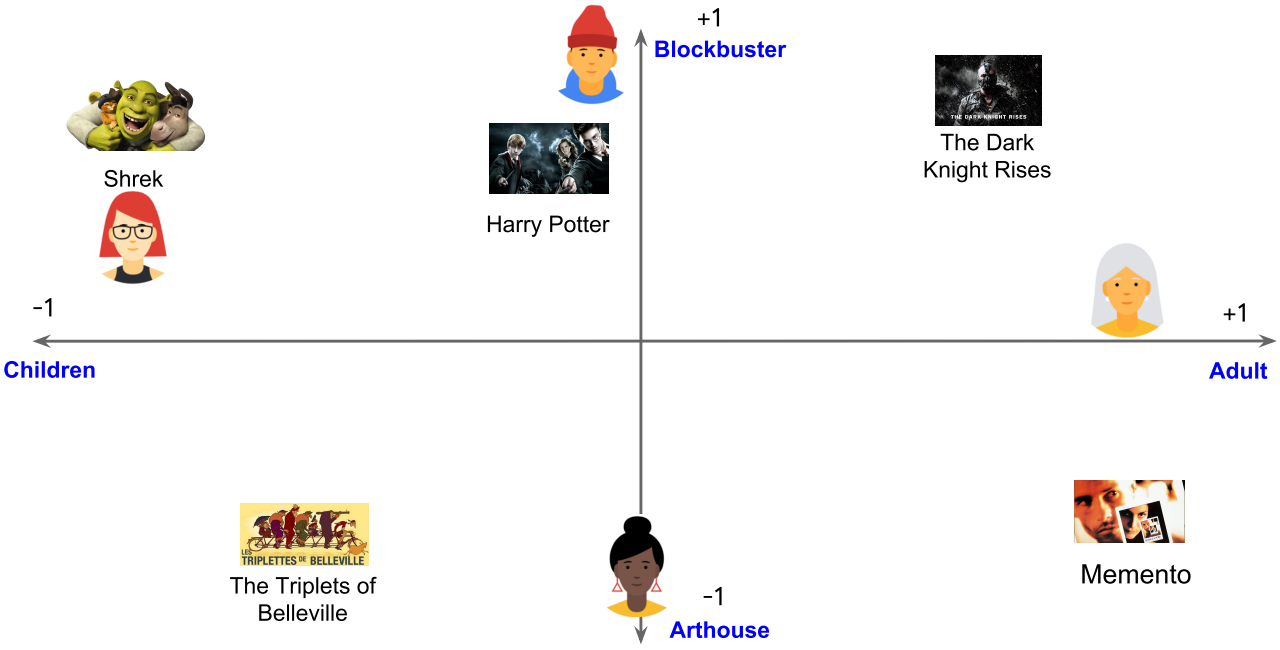
Suppose we assign to each movie a scalar in $[-1, 1]$ that describes whether the movie is for children (negative values) or adults (positive values). Suppose we also assign a scalar to each user in $[-1, 1]$ that describes the user's interest in children's movies (closer to -1) or adult movies (closer to +1). The product of the movie embedding and the user embedding should be higher (closer to 1) for movies that we expect the user to like.



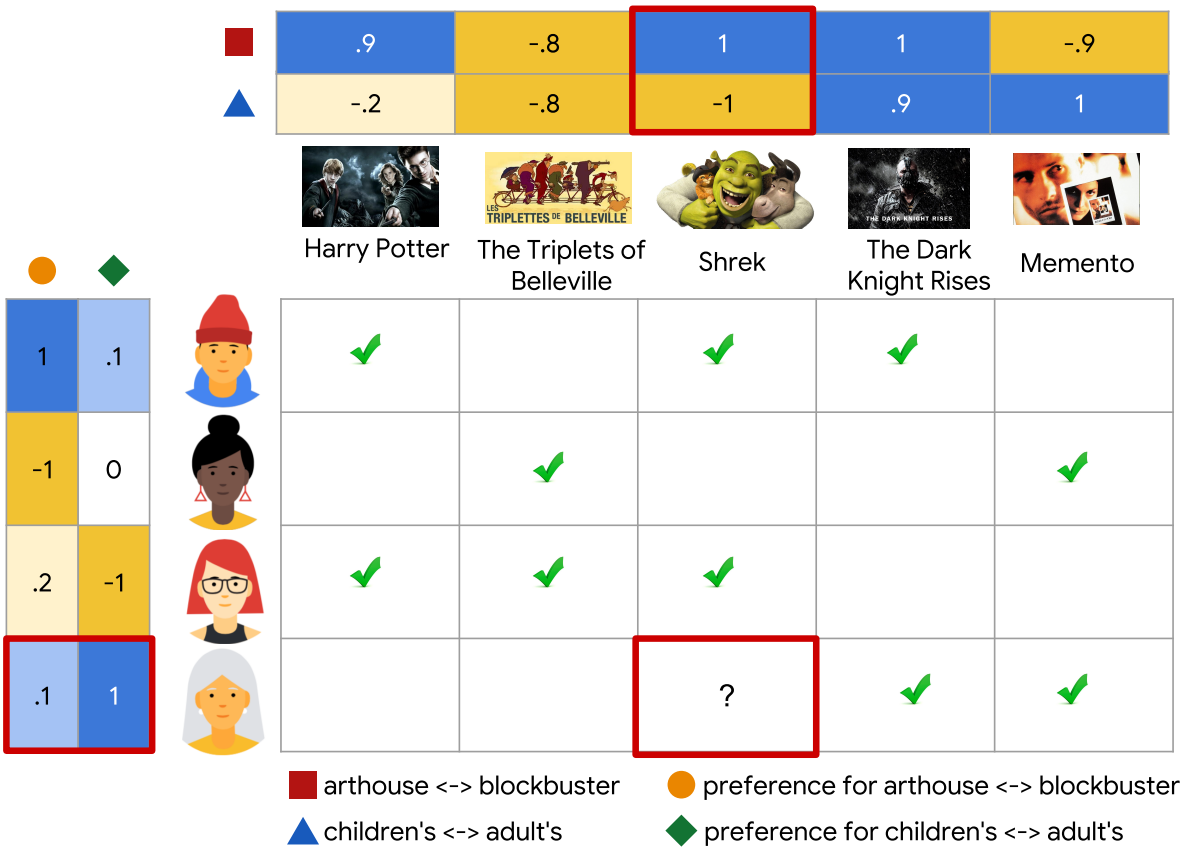
In the diagram below, each checkmark identifies a movie that a particular user watched. The third and fourth users have preferences that are well explained by this feature—the third user prefers movies for children and the fourth user prefers movies for adults. However, the first and second users' preferences are not well explained by this single feature.

2D Embedding

One feature was not enough to explain the preferences of all users. To overcome this problem, let's add a second feature: the degree to which each movie is a blockbuster or an arthouse movie. With a second feature, we can now represent each movie with the following two-dimensional embedding:



We again place our users in the same embedding space to best explain the feedback matrix: for each (user, item) pair, we would like the dot product of the user embedding and the item embedding to be close to 1 when the user watched the movie, and to 0 otherwise.



We represented both items and users in the same embedding space. This may seem surprising. After all, ı
ems are two different entities. However, you can think of the embedding space as an abstract representati

on to both items and users, in which we can measure similarity or relevance using a similarity metric.

In this example, we hand-engineered the embeddings. In practice, the embeddings can be learned *automatically*, which is the power of collaborative filtering models. In the next two sections, we will discuss different models to learn these embeddings, and how to train them.

The collaborative nature of this approach is apparent when the model learns the embeddings. Suppose the embedding vectors for the movies are fixed. Then, the model can learn an embedding vector for the users to best explain their preferences. Consequently, embeddings of users with similar preferences will be close together. Similarly, if the embeddings for the users are fixed, then we can learn movie embeddings to best explain the feedback matrix. As a result, embeddings of movies liked by similar users will be close in the embedding space.

Check Your Understanding

The model recommends a shopping app to a user because they recently installed a similar app. What kind of filtering is this an example of?

- Collaborative filtering
- Content-based filtering

Previous
Advantages & Disadvantages (/machine-learning/recommendation/content-based/summary)

Next
Matrix Factorization (/machine-learning/recommendation/collaborative/matrix)

Last updated 2021-02-05 UTC.