Data Mining

Collaborative Filtering – Latent Factor Models

Examples from "Data Science Concepts and Practice" 2018 MK : Vijay Kotu and Bala Deshpande

Terri Hoare – November 2023

Collaborative Filtering
Matrix Factors

The ratings matrix discussed so far has information on the user, item, and the strength of user-item interaction. For every user, the ratings information is at the level of individual item, that is, every individual movie, track, product. The number of items usually ranges from thousands to millions of unique values. If one were to ask someone what movies or books they would prefer to watch or read, the answer is usually in the form of some generalized dimension. For example, expressing the preference for science fiction movies, movies directed by Christopher Nolan, movies with strong female leads, crime novels or 1920s literature. The subsequent follow-up might be providing specific item examples which belong to those groupings, for example titles like, Interstellar, Iron Lady, The Great Gatsby, etc. The generalized categories can be a predefined genre, works by a creator, or sometimes it might be a yet-to-be named or vague categorization. Perhaps the user just likes a set of items with no specific generalized category.

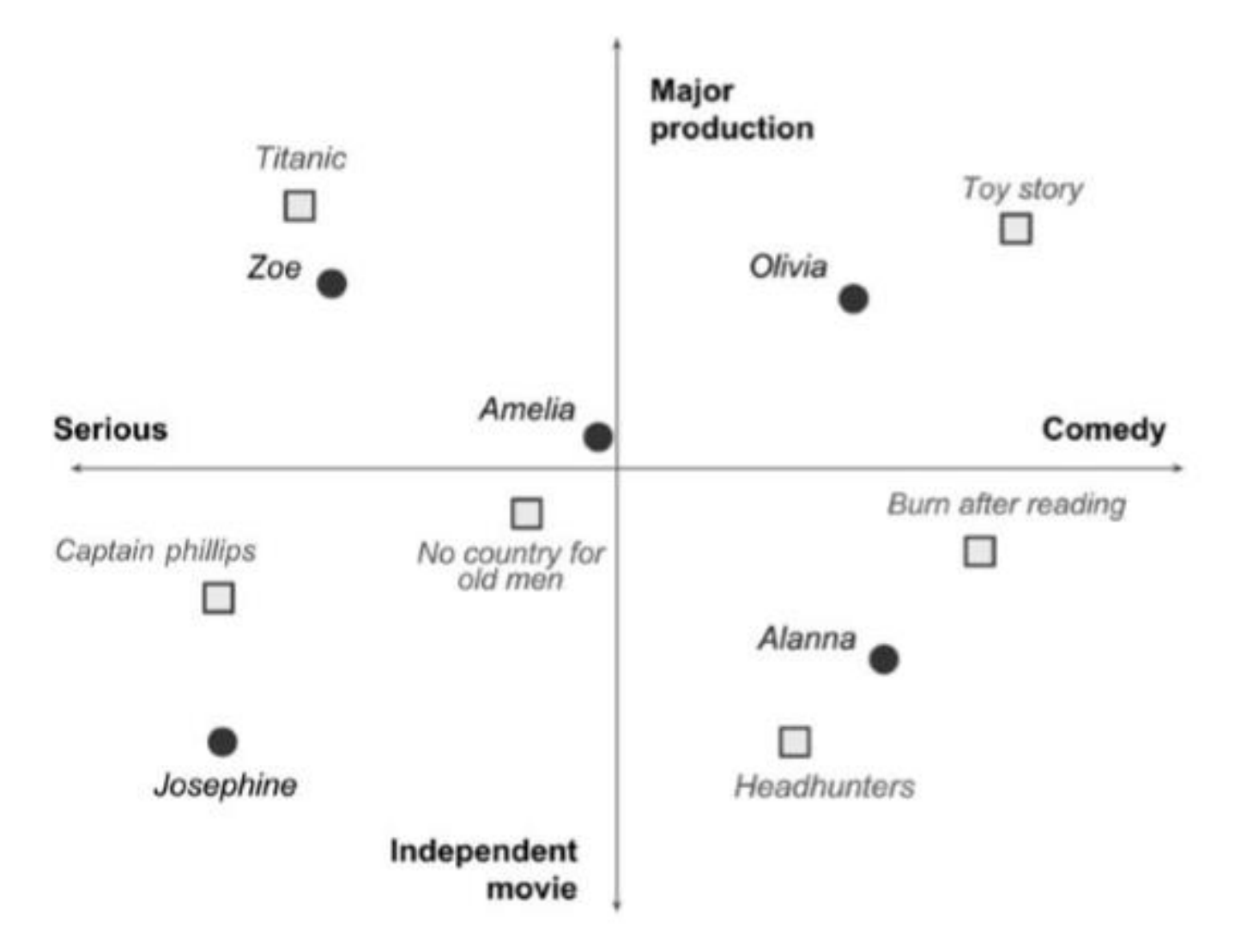
Collaborative Filtering
Matrix Factors

The Latent Factor model, like the neighbourhood methods, uses just the ratings matrix as the only input. It tries to generalize and explain the ratings matrix with a set of latent factors or generalized dimensions. The factors, which usually range from a dozen to a few hundred, are automatically inferred from the ratings matrix, as long as the number of factors is specified. There is no separate input to the model that provides pre-classified genres or factors. In fact, there are no interpretable names for the latent factors. The inferred factors might resemble genre like classification (science fiction or family movies) and in some cases they are just uninterpretable groupings of items. It will be interesting to research and generalize why a group of items are rated highly against a factor. Once the factors are discovered, the model associates an item's membership towards a factor and a user's inclination towards the same factor. If the user and item are near to each other when plotted against a factor or a set of factors, then there is a strong user-item preference.

Collaborative Filtering
Matrix Factors

Next slide shows the users (circles) and items (squares) mapped against two illustrative factors: production scale and comedic content. The items or movies are plotted on the chart based on how movies express themselves against the latent factors. The users are plotted based on their preferences against the same latent factors. From the chart, it can be concluded that the user Amelia prefers the movie No country for Old men because both the user and the movie are close to one another when expressed against the latent factors. Similarly, Alanna prefers the movie Headhunters instead of Titanic. The user-item preference is calculated by the dot product of the user vector and the item vector expressed latent factors. The similarity between the user and the item vectors dictates the preference of the user to the item (Koren et al., 2009).

Collaborative Filtering
Items and Users in Latent Factor Space



Collaborative Filtering
Matrix Factors

Matrix Factorization is a technique to discover the latent factors from the ratings matrix and to map the items and the users against those factors. Consider a ratings matrix R with ratings by n users for m items. The ratings matrix R will have nxm rows and columns. The matrix R can be decomposed into two thin matrices P and Q. P will have nxf dimensions and Q will have mxf dimensions where f is the number of latent factors. In the example used in the previous slide, there are two latent factors. The matrix R can be decomposed in such a way that the dot product of the matrix P and transposed Q will yield a matrix with nxm dimensions that closely approximates the original ratings matrix R.

Implementing a matrix factorization-based recommender from scratch is an effort intensive process. The RapidMiner Recommenders extension offers prebuilt operators to implement Biased Matrix Factorization (BMF) recommendation engines (*Mihelčić et al.*, 2012).

.