Latent Factors and SVD

TWSM, BIPM SS2022

Netflix Prize 1

Netflix provided 100M ratings (from 1 to 5) of 17K movies by 500K users. Netflix then posed a “quiz” which consists of a bunch of question marks plopped into previously blank slots, and your job is to fill in best-guess ratings in their place.

Netflix Prize 2

Imagine for a moment that we have the whole shebang–8.5 billion ratings and a lot of weary users. Presumably there are some **generalities** to be found in there, something more concise and descriptive than 8.5 billion completely independent and unrelated ratings.

For instance, any given movie can, to a rough degree of approximation, be described in terms of some **basic attributes** such as overall quality, whether it's an action movie or a comedy, what stars are in it, and so on. And every user's preferences can likewise be roughly described in terms of whether they tend to rate high or low, whether they prefer action movies or comedies, what stars they like, and so on.

Netflix Prize 3

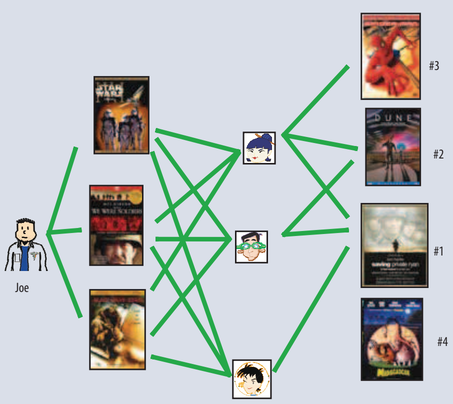
And if those basic assumptions are true, then a lot of the 8.5 billion ratings ought to be explainable by a lot less than 8.5 billion numbers, since, for instance, a single number specifying how much action a particular movie has may help explain why a few million action-buffs like that movie.

A fun property of machine learning is that this reasoning works in reverse too: If meaningful generalities can help you represent your data with fewer numbers, finding a way to represent your data in fewer numbers can often help you find meaningful generalities. **Compression is akin to understanding.**

Recommendation Engines

- **content filtering** approach
  - creates a profile for each user or product (a movie profile could include attributes regarding its genre, the participating actors, its box office popularity, and so forth. User profiles might include demographic information or answers provided on a suitable questionnaire.)
  - associate users with matching products.
  - requires gathering external information that might not be available or easy to collect.
  - Music Genome Project, Pandora.com
- **collaborative filtering** relies only on past user behavior
  - for example, previous transactions or product ratings
  - not requiring the creation of explicit profiles.
  - "cold start problem": inability to address new products and users.

Collaborative filtering: neighborhood methods

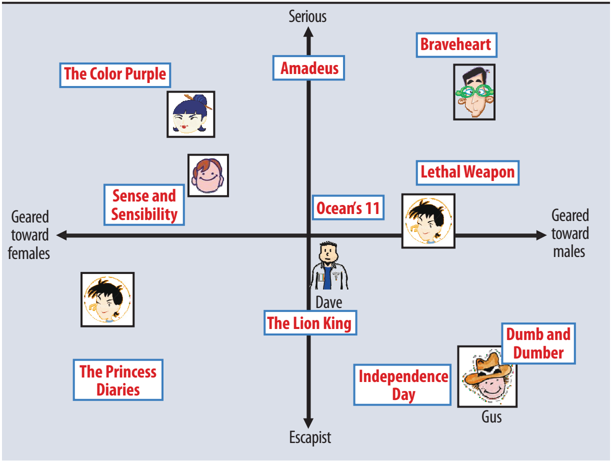


The user-oriented neighborhood method. Joe likes the three movies on the left. To make a prediction for him, the system finds similar users who also liked those movies, and then determines which other movies they liked. In this case, all three liked Saving Private Ryan, so that is the first recommendation. Two of them liked Dune, so that is next, and so on.

Collaborative filtering: latent factor models

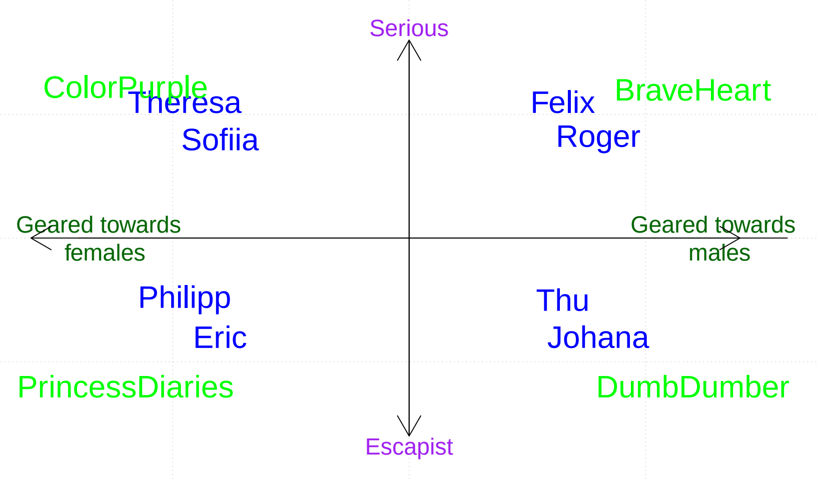
- Tries to explain the ratings by characterizing both items and users on “few” (e.g. 20 to 100) factors inferred from the ratings patterns.
- Such **latent factors** alternative to e.g. human created song genres.
- For movies, the discovered factors might measure
  - obvious dimensions such as comedy versus drama, amount of action, or orientation to children;
  - less well-defined dimensions such as depth of character development or quirkiness;
  - or completely uninterpretable dimensions.
- For users, each factor measures how much the user likes movies that score high on the corresponding movie factor.

Latent factor models



A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes-male versus female and serious versus escapist.

No hidden (latent) factors



Only Movie Rankings available

	Princess Diaries	Dumb Dumber	Color Purple	Brave Heart
Jafar	3	2	4	2
Moritz	2	3	4	2
Felix	2	2	3	3
Rukniya	4	3	4	3
Lilit	3	2	4	2
Philipp	3	1	4	1

Matrix Factorization

Matrix factorization models map both users and items to a joint latent factor space of dimensionality f, such that user-item interactions are modeled as inner products in that space. Accordingly, each item i is associated with a vector $q_i \in R^f$, and each user u is associated with a vector $p_u \in R^f$ such that the dot product approximates the user u's rating of item i:

$$\hat{r}_{ui} = q_i^T \cdot p_u$$

Singular Value Decomposition (SVD)

$$X = U \cdot S \cdot V^T$$

- Left Eigenvectors

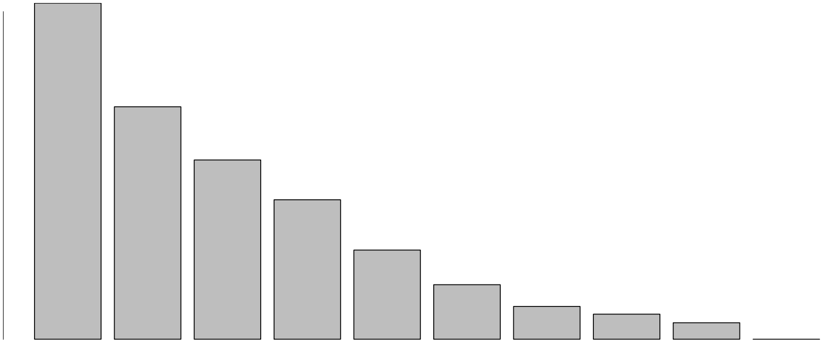
	LF1	LF2	LF3	LF4	LF5	LF6	LF7	LF8
Jafar	2.01	-2.38	-0.57	1.15	-1.09	0.09	-0.59	0.85
Moritz	-0.58	2.55	-0.37	0.62	1.73	0.95	0.06	0.65
Felix	-1.51	0.81	-1.41	0.39	-0.97	-0.12	0.59	0.56
Rukniya	2.42	0.34	1.89	0.32	0.44	-1.43	1.00	0.09
Lilit	2.52	-1.88	-0.73	-0.25	0.67	1.34	0.57	-0.70
Philipp	1.36	1.37	-2.76	-1.13	0.20	-1.06	-0.66	-0.49
Clemens	-3.37	-1.27	-0.02	2.39	0.47	-0.47	-0.15	-0.71
Jonas	-2.69	-0.62	0.06	-2.22	-1.04	0.25	0.64	-0.03
Eric	-1.01	-1.40	1.85	-1.71	1.15	-0.20	-0.88	0.30
Thu	0.87	2.47	2.06	0.43	-1.56	0.65	-0.57	-0.53

Left/Right View

- Right Eigenvectors

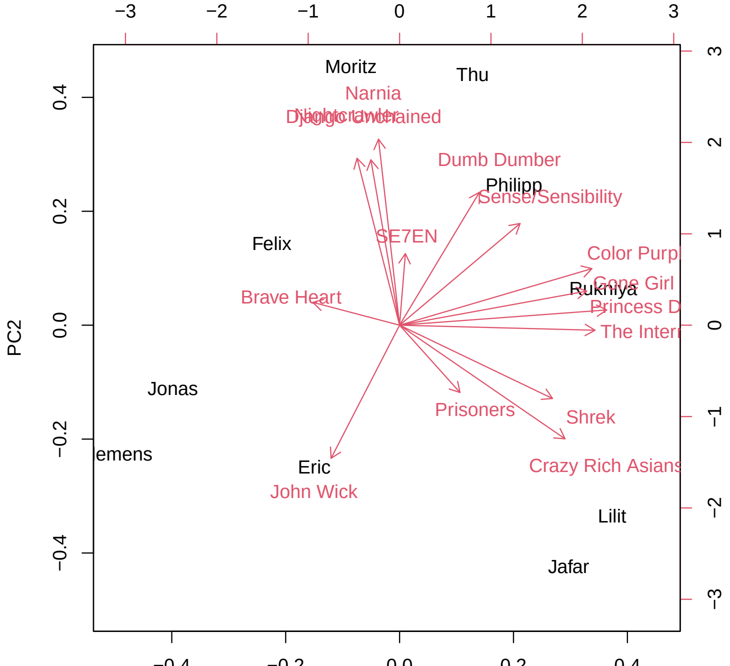
	LF1	LF2	LF3	LF4	LF5	LF6	LF7
Princess Diaries	0.42	0.04	-0.10	0.11	0.01	-0.30	0.41
Dumb Dumber	0.16	0.32	0.30	0.13	-0.25	0.39	0.40
Color Purple	0.39	0.14	-0.06	0.28	0.22	0.04	-0.27
Brave Heart	-0.18	0.06	0.50	-0.12	-0.36	0.14	-0.02
Narnia	-0.04	0.45	-0.08	0.01	-0.26	-0.33	-0.11
Sense/Sensibility	0.24	0.25	0.26	0.37	-0.08	0.11	-0.38
Django Unchained	-0.06	0.40	0.12	-0.40	-0.01	-0.30	0.09
Shrek	0.31	-0.18	0.35	0.01	0.30	0.00	-0.22
SE7EN	0.01	0.17	-0.52	0.08	0.00	0.52	0.14
The Intern	0.40	-0.01	-0.05	-0.19	-0.37	0.06	-0.22
John Wick	-0.14	-0.32	-0.19	0.12	-0.57	0.05	-0.31
Crazy Rich Asians	0.34	-0.28	-0.09	-0.24	-0.19	-0.27	-0.02
Gone Girl	0.38	0.08	-0.19	-0.26	-0.19	0.09	0.11
Prisoners	0.12	-0.16	0.16	-0.55	0.16	0.40	-0.04
Nightcrawler	-0.09	0.41	-0.21	-0.31	0.17	0.07	-0.45

Singular Values

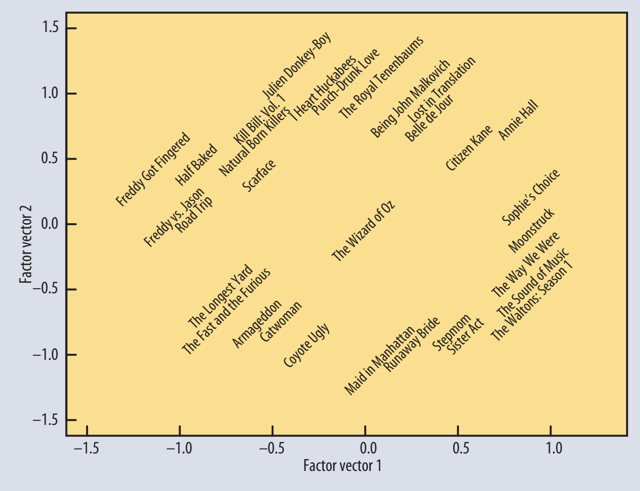


- Missing Values: Stochastic gradient descent
- Regularization

PCA and latent factors



Netflix prize



The first two vectors from a matrix decomposition of the Netflix Prize data. The plot reveals distinct genres, including clusters of movies with strong female leads, fraternity humor, and quirky independent films.

Useful Links

- Netflix Prize: Iterative SVD with a few lines of code
- Latent Semantic Analyis (LSA)