user-item matrix.
- V^T gives information about how much each movie is related to latent features.

Since this traditional decomposition doesn't actually work when our matrices have missing values, we looked at another method for decomposing matrices.

FunkSVD

FunkSVD was a new method that you found to be useful for matrices with missing values. With this matrix factorization you decomposed a user-item (**A**) as follows:

$$A = UV^T$$

Where

- U gives information about how users are related to latent features.
- V^T gives information about how much each movie is related to latent features.

You found that you could iterate to find the latent features in each of these matrices using gradient descent. You wrote a function to implement gradient descent to find the values within these two matrices.

Using this method, you were able to make a prediction for any user-movie pair in your dataset. You also could use it to test how well your predictions worked on a train-test split of the data. However, this method fell short with new users or movies.

The Cold Start Problem

Collaborative filtering using FunkSVD still wasn't helpful for new users and new movies. In order to recommend these items, you implemented content based and ranked based recommendations along with your FunkSVD implementation.

Author's Note

There are so many ways to make recommendations, and this course provides you a very strong mind and skill set to tackle building your own recommendation systems in practice.

NEXT