



دانشگاه اصفهان
دانشکده مهندسی کامپیوتر

مستند پروژه توصیه‌گر فیلم

درس جبر خطی کاربردی
استاد درس: دکتر پیمان ادیبی

امیرعلی لطفی (۴۰۰۳۶۱۳۰۵۳)

زمستان ۱۴۰۲

توضیحات

در این پروژه، با استفاده از تجزیه مقادیر منفرد (Singular Value Decomposition) و معیار شباهت cosine similarity، یک توصیه‌گر فیلم پیاده شده است.

آماده‌سازی محیط

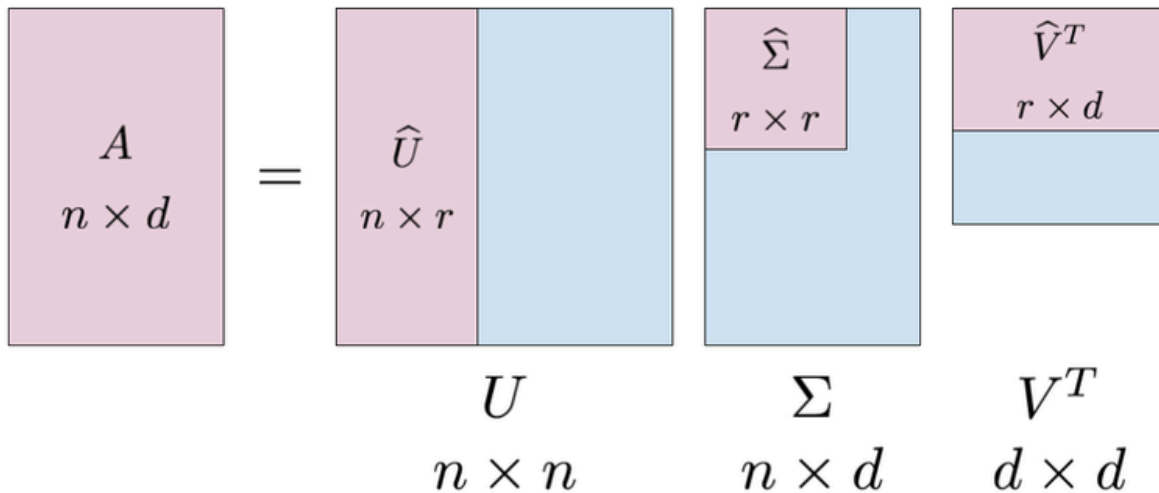
ابتدا دیتاست‌های داده شده را با استفاده از pandas می‌خوانیم:

```
movies = pd.read_csv("../dataset/movies.csv")
ratings = pd.read_csv("../dataset/ratings.csv")
links = pd.read_csv("../dataset/links.csv")
tags = pd.read_csv("../dataset/tags.csv")
```

سپس ماتریس امتیاز کاربر-فیلم را توسط تابع np.pivot_table می‌سازیم:

```
# Creating the user-movie rating pivot table
# The rows are the users, columns are the movies and the values are
the ratings
user_rates = ratings.pivot_table(index='userId', columns='movieId',
values='rating', fill_value=0)
```

محاسبه SVD



برای محاسبه تجزیه مقادیر منفرد، نیاز است که سه ماتریس U و Σ و V را محاسبه کنیم.
ابتدا مقادیر ویژه و بردارهای ویژه ماتریس $A^T A$ را با استفاده از تابع `np.linalg.eig` محاسبه می‌کنیم:

```
ATA = A.T @ A
eigvals, eigvecs = np.linalg.eig(ATA)
```

سپس برای محاسبه مقادیر منفرد ماتریس A ، از مقادیر ویژه به دست آمده برای ماتریس $A^T A$ جذر می‌گیریم:

```
singular_values = np.sqrt(eigvals)
```

برای محاسبه ماتریس U نیاز است تا ابتدا از فرمول زیر پیروی کنیم:

$$U = [u_1, u_2, \dots, u_r]$$

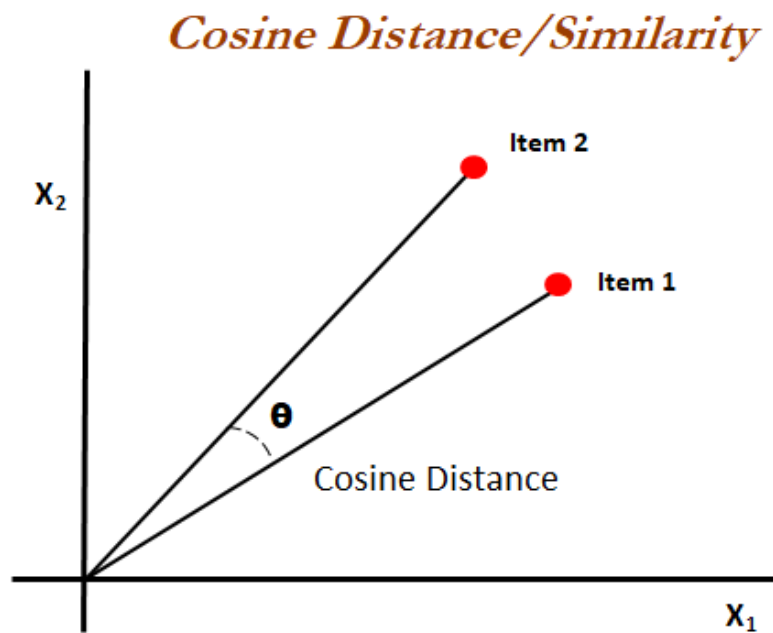
$$u_i = \frac{Av_i}{\sigma_i}$$

با استفاده از کد زیر می‌توان ماتریس V و U و S را حساب کرد:

```
V = eigvecs  
S = singular_values  
U = np.matmul(A, eigvecs) / singular_values
```

محاسبه Cosine Similarity

این فاکتور معیاری است برای تخمین شباهت دو بردار از روی زاویه بین آنها:



$$\text{similarity}(x_1, x_2) = \cos(\theta)$$

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

پیشنهاد فیلم

برای پیشنهاد یک فیلم به یک کاربر مشخص، ابتدا نیاز است تا سلیقه آن کاربر را به دست آورد. سپس می‌توان شبیه‌ترین سلیقه‌ها را در بین باقی کاربرها جستجو کرد. سپس از روی کاربران با سلیقه مشابه، فیلم‌هایی که آن‌ها مشاهده می‌کنند را به کاربر هدف پیشنهاد داد.

برای اینکار ابتدا نیاز است تا یک شاخص برای سلیقه کاربران داشته باشیم. ستون‌های ماتریس U در تجزیه SVD ماتریس user_rates مقادیر latent features می‌باشند. این به این معنی است که ویژگی‌هایی که قابل اندازه‌گیری نبوده، در این ماتریس به صورت ویژگی‌هایی مخفی نمایان شده‌اند.

هر ستون در ماتریس U نشان‌دهنده latent features برای هر کاربر می‌باشد. با مقایسه این بردارها توسط معیار cosine similarity، می‌توان کاربرانی را که نزدیک‌ترین سلیقه به کاربر هدف دارند را یافت.

```
# Choose a target user for recommendation
target_user_index = int(input("Enter a user ID to get recommendations: "))

user_latent_feature_vector = U[target_user_index, :]

# Calculate cosine similarity between the target user and all other users
user_similarity_scores = np.array([cosine_similarity(user_latent_feature_vector,
U[i, :]) for i in range(U.shape[0])])

# Find the indices of users most similar to the target user
similar_users_indices = np.argsort(user_similarity_scores)[::-1]

# Exclude the target user from the similar users
similar_users_indices = similar_users_indices[similar_users_indices !=
target_user_index]
```

```
# Recommend movies based on the most similar users
recommended_movies = np.dot(U[similar_users_indices, :] * S, VT)

# Exclude movies the target user has already rated
movies_already_rated =
user_rates.iloc[target_user_index][user_rates.iloc[target_user_index] > 0].index
recommended_movies[:, movies_already_rated] = 0

recommended_movie_ids =
recommended_movies[target_user_index,:].argsort()[::-1][:10]
```

نمایش داده‌های فیلم‌های انتخاب شده:

```
movies[movies["movieId"].isin(recommended_movie_ids)]
```