باسمه تعالی

تاریخ: ۱۴۰۴/۰۹/۲۲

پروژه‌ی دوم درس مبانی علم داده

تحلیل و فشرده‌سازی ماتریس امتیاز فیلم‌ها با PCA و SVD (با دو روش پر کردن داده‌های گمشده) + توصیه‌گر ساده

در این تمرین شما یک ماتریس «کاربر–فیلم» می‌سازید، بخشی از امتیازها را برای ارزیابی کنار می‌گذارید ،(Validation) سپس با دو روش SVD و PCA (به دو شیوه‌ی محاسبه‌ی PCA ماتریس را با رتبه‌ی کم تقریب می‌زنید و کیفیت پیش‌بینی امتیازهای کنارگذاشته‌شده را با RMSE مقایسه می‌کنید. در پایان برای یک کاربر، چند فیلم پیشنهادی ارائه می‌دهید و معنی «جهت‌های اصلی واریانس سلیقه کاربران» را تفسیر می‌کنید.

فایل‌های لازم (ضمیمه)

۱. **movielens.csv**
ستون‌ها: userId, movieId, rating, timestamp, title, genres
هر سطر = یک امتیاز ثبت‌شده‌ی کاربر به یک فیلم

۲. **movielens_val_indices.csv**
ستون‌ها: row_inds, col_inds
ایندکس‌های ماتریس A که باید به عنوان Validation کنار گذاشته شوند.

۳. **movie_map.pkl**
دیکشنری نگاشت movieId $\rightarrow$ عنوان فیلم برای گزارش توصیه‌ها و تفسیر مؤلفه‌ها.

کتابخانه‌های مجاز / غیرمجاز

مجاز

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
```

برای محاسبات خطی :

- np.linalg.svd
- np.linalg.eigh یا np.linalg.eig ✓ (فقط برای PCA مبتنی بر کوواریانس)

غیرمجاز (مهم)

- هر نوع PCA/SVD آماده: sklearn.decomposition.PCA، TruncatedSVD، Surprise، implicit و ...

گام ۰) بارگذاری داده‌ها

فرآیند انجام کار

- سه فایل را بخوانید.
- ابعاد دیتافریم‌ها و چند سطر اول را چاپ کنید.
- movie_map.pkl، نوع و اندازه‌ی آن را چاپ کنید.

خروجی مورد انتظار

- ratings shape و columns
- val_idx shape و columns
- اندازه‌ی movie_map

گام ۱) ساخت ماتریس User–Item و ساخت Fit/Validation

فرآیند انجام کار

- ماتریس A را با شکل (num_users, num_movies) بسازید:
  - سطر = userId
  - ستون = movieId
  - مقدار = rating
  - خانه‌های بدون امتیاز را 0 بگذارید (فعلاً فقط به‌عنوان نمایش "missing"، نه امتیاز واقعی)
- با استفاده از movielens_val_indices.csv:
  - A_{fit} بسازید: یک کپی از A که در اندیس‌های validation مقدار را 0 می‌کنید.
  - بردار y_val_true را از A استخراج کنید (امتیازهای واقعی کنارگذاشته‌شده).

خروجی مورد انتظار

- A shape, A nonzeros, A sum
- A_fit sum و اینکه آیا واقعاً در validation صفر شده: all(A_fit[rows, cols]==0)
- چند مقدار اول/آخر y_val_true

سوالات تحلیلی

۱. چرا جدا کردن Validation با «صفر کردن خانه‌ها در A_{fit}» منطقی است؟
۲. اگر برخی کاربران/فیلم‌ها در Validation هیچ نمونه‌ای نداشته باشند، چه اثری روی ارزیابی‌های "سطح کاربر" می‌گذارد؟

گام ۲) دو روش پر کردن خانه‌های خالی (Imputation)

شما باید دو نسخه از ماتریس بسازید:

(A) پر کردن با صفر: fill(0)

- همان A_{fit} را به عنوان ماتریس پرشده در نظر بگیرید.

(B) پر کردن با میانگین کاربر: fill(user-mean)

- میانگین هر کاربر را فقط از امتیازهای مشاهده‌شده (غیرصفر) حساب کنید.
- سپس هر جای صفر (گمشده) را با میانگین همان کاربر پر کنید.
- اگر کاربری هیچ امتیازی نداشت، از یک میانگین کلی (global mean) استفاده کنید.

خروجی مورد انتظار

- مقدار global_mean
- نمونه‌ای از user_means و اینکه در ماتریس mean-fill دیگر صفر گمشده نداریم

سوالات تحلیلی

۱. چرا fill(0) می‌تواند یک "سوگیری" جدی وارد کند؟
۲. چرا fill(user-mean) ممکن است به طرز عجیبی خطا را کم کند؟ این الزاماً یعنی "مدل یاد گرفته"؟

گام ۳) SVD و پیش‌بینی امتیازها

فرآیند انجام کار

- روی ماتریس پرشده (هر کدام از دو روش fill) SVD بزنید:

$$A \approx U_k \, \Sigma_k \, V_k^T$$

- برای مقادیر مختلف k (مثلاً $k = 5, 10, 15, \ldots, 100$) ماتریس را بازسازی کنید و برای validation فقط در rows, cols مقدار پیش‌بینی را بردارید.
- RMSE را فقط روی validation حساب کنید.
- نمودار RMSE برحسب k را برای هر دو fill رسم کنید (دو منحنی روی یک نمودار).

خروجی مورد انتظار

- نمودار SVD: RMSE vs k

سوالات تحلیلی

۱. آیا همیشه با افزایش k باید RMSE بهتر شود؟ اگر در عمل دیدید بعد از یک نقطه بدتر شد، یک توضیح علمی بدهید .
۲. چرا منحنی fill(user-mean) ممکن است خیلی پایین‌تر از fill(0) بیفتد؟

گام ۴) PCA با دو روش

شما باید PCA را به دو روش پیاده‌سازی و مقایسه کنید:

۴-۱) PCA مبتنی بر کوواریانس (Covariance PCA)

فرآیند انجام کار

- داده را center کنید .

- ماتریس کوواریانس را بسازید .
- با np.linalg.eigh/eig بردارهای ویژه را بگیرید.
- با k مولفه، بازسازی انجام دهید.
- validation RMSE را برای kهای مختلف حساب کنید.

۴-۲) PCA از طریق SVD (PCA via SVD)

فرآیند انجام کار

- همان داده‌ی center شده را مستقیم با SVD تجزیه کنید و مولفه‌های اصلی را استخراج کنید.
- بازسازی و RMSE را مثل بالا حساب کنید.

خروجی مورد انتظار

- نمودار PCA(cov): RMSE vs k (برای دو fill)
- نمودار PCA(SVD): RMSE vs k (برای دو fill)
- نمودار مقایسه‌ای PCA(cov) vs PCA(SVD) (حداقل برای حالت mean-fill)

سوالات تحلیلی

۱. چرا انتظار داریم PCA(cov) و PCA(SVD) (روی داده‌ی center شده) نتایج بسیار نزدیک بدهند؟
۲. اگر دقیقاً یکسان نشدند، دو دلیل ممکن چیست؟

گام ۵) انتخاب بهترین k

فرآیند انجام کار

- برای هر روش و هر نوع fill، best_k = argmin RMSE(k) را گزارش کنید.

خروجی مورد انتظار

چاپ best_k و best_RMSE برای:

- SVD + fill(0)
- SVD + fill(mean)
- PCA(cov) + fill(0)
- PCA(cov) + fill(mean)

سوالات تحلیلی

۱. چرا ممکن است best_k در حالت mean-fill کوچک‌تر باشد؟
۲. اگر best_k در fill(0) خیلی بزرگ‌تر/کوچک‌تر شد، تفسیر کنید.

گام ۶) بخش Recommender: توصیه فیلم به یک کاربر

فرآیند انجام کار

- کاربری را انتخاب کنید که بیشترین تعداد نمونه در validation دارد.
- با بهترین مدل انتخاب‌شده:
  - برای همان کاربر، امتیازهای پیش‌بینی‌شده‌ی فیلم‌های "ندیده" را حساب کنید.
  - Top-10 فیلم پیشنهادی را چاپ کنید.
- یک نمودار "True vs Predicted" برای همان کاربر رسم کنید:
  - فقط روی امتیازهایی که برای آن کاربر در validation وجود دارد.

خروجی مورد انتظار

- validation count برای کاربر انتخابی
- جدول/پرینت Top-10 توصیه
- نمودار مقایسه True و Pred

گام ۷) چه جهت‌هایی بیشترین واریانس رفتار کاربران را توضیح می‌دهند؟

فرآیند انجام کار

- از PCA روی داده‌ی center شده، مولفه‌ی اول (PC1) را بگیرید.
- وزن‌های PC1 روی فضای فیلم‌ها را بررسی کنید:
  - Top-10 فیلم با وزن بیشترین مقدار مثبت
  - Top-10 فیلم با وزن کمترین مقدار (منفی‌ترین)
- از movie_map اسم فیلم‌ها را چاپ کنید.

خروجی مورد انتظار

- نمودار وزن‌های PC1
- لیست Top مثبت/منفی با movieId و (در صورت امکان) نام فیلم

سؤال تحلیلی

۱. معنی وزن مثبت/منفی در PC1 چیست؟
۲. چرا برخی فیلم‌ها قدر مطلق وزن بزرگ‌تری دارند؟ این چه چیزی درباره‌ی "فیلم‌های تعیین‌کننده در تفاوت سلیقه‌ها" می‌گوید؟