در این گزارش به بررسی انواع مختلف پالایش مشارکتی یعنی پالایش مشارکتی بر اساس کاربر (User-based collaborative filtering) و پالایش مشارکتی بر اساس آیتم (Item-based collaborative filtering)، اجرا و تست آن‌ها می‌پردازیم.

شماره‌ی ۳ گزارش پیشرفت پروژه

پالایش مشارکتی سیستم پیشنهاد دهنده | ۱۸ اردیبهشت ۱۴۰۰

فائزه سادات سعیدی نژاد

در گزارش قبلی بصورت کلی در رابطه با پالایش مشارکتی صحبت کردم، در این گزارش اما می‌خواهم در رابطه با دو نوع پالایش مشارکتی یعنی پالایش مشارکتی بر اساس کاربر[1] و پالایش مشارکتی بر اساس آیتم[2] صحبت کنم.

پالایش مشارکتی بر اساس کاربر

در این پالایش مشارکتی، بر اساس امتیازهایی که به محتواها داده‌اید، کاربران شبیه به شما را پیدا می‌کند و محتواهایی که آن کاربران دوستشان داشته‌اند و شما هنوز ندیده‌اید را به شما پیشنهاد می‌کند. اما چطور این کار انجام می‌شود؟ با استفاده از یک آرایه‌ی دو بعدی یا همان ماتریس[3] که در یک بعد آن کاربران و در بعد دیگر، فیلم‌ها هستند و امتیاز کاربران به فیلم‌ها در ماتریس مشخص است. مانند شکل زیر:

	Indiana Jones	Star Wars	Empire Strikes Back	Incredibles	Casablanca
Bob	4	5			
Ted					1
Ann		5	5	5	

حال با استفاده از شباهت کسینوسی[4]، شباهت بین هرکدام از این کاربران را باهم بدست می‌آوردیم، برای به دست آوردن شباهت کسینوسی بین دو کاربر، باید فقط محتواهایی را در نظر بگیریم که هردو کاربر دیده‌اند در این مثال Ann و Bob که هردو فیلم Star Wars را دیده‌اند پس شباهت کسیسنوسی بین آن‌ها یک است. به همین ترتیب یک ماتریس جدید می‌سازیم برای شباهت کسینوسی بین کاربران:

	Bob	Ted	Ann
Bob	1	0	1
Ted	0	1	0
Ann	1	0	1

مشکل این روش این است که در صورت داشتن پراکندگی داده[5]، اگر دو کاربر یک محتوا را دیده باشند و امتیازهای بسیار متفاوت از یگدیگر به آن محتوا داده باشند، در نهایت شباهت کسینوسی این دو کاربر همچنان یک می‌شود و این باعث می‌شود که نتایج عجیبی داشته باشیم. در ادامه‌ی این روش، باید همسایه‌های کاربر مورد نظر را نگاه کنیم و شباهت کسینوسی‌شان را مقایسه کنیم، که در این مثال نزدیکترین همسایه به Bob، Ann است و در آخر Ted که هیچ شباهتی به Bob ندارد. سپس محتواهایی که Ann

[1] User-based collaborative filtering
[2] Item-based collaborative filtering
[3] Matrix
[4] Cosine similarity
[5] Sparsity

به آن‌ها امتیاز داده است را مرتب‌سازی براساس امتیازی که از Ann گرفته‌اند می‌کنیم و در آخر براساس اینکه Bob محتواها را دیده یا خیر، فیلتر می‌شوند.

قسمت عملی

با استفاده از برنامه‌ی Spyder، کدهای فولدر CollaborativeFiltering را باز می‌کنیم. فایلی که با آن کار داریم و می‌خواهیم اجرایش کنیم SimpleUserCF.py است. همانطور که مشاهده می‌کنید، فایل اجرایی فایل بسیار کوچک و کم کدی است، دلیلش این است که از کتابخانه‌ی SurpriseLib که در گزارش قبلی به آن پرداختیم، داریم استفاده می‌کنیم. همانطور که در کد مشاهده می‌کنید، از شباهت کسینوسی استفاده می‌کنیم برای بدست آوردن شباهت بین کاربران.

```
sim_options = {'name': 'cosine', 'user_based': True }
```

اگر کد را اجرا کنید متوجه می‌شوید که کد بسیار سریع اجرا می‌شود و دلیلش این است که دیگر مثل قبل نیاز نیست که دقت[6] سیستم پیشنهاد دهنده را اندازه بگیرد در نتیجه نیاز نیست که برای هر کدام از محتواها، امتیاز احتمالی‌ای که کاربر مورد نظر ممکن است به آن بدهد را پیش بینی کند. تنها کاری که نیاز است در پالایش مشارکتی انجام بگیرد، ساختن ماتریس‌های شباهت[7] است. به همین دلیل است که از این سیستم پیشنهاد دهنده در کمپانی‌های بسیار بزرگ با تعداد داده‌های زیاد مورد استفاده قرار می‌گیرد. همانطور که در خروجی مشاهده می‌کنید، این کد به ما لیست فیلم‌های پیشنهادی به علاوه‌ی امتیاز آن‌ها براساس اینکه کاربر مشابه کاربر مورد نظر ما چقدر از آن‌ها خوشش آمده، بصورت مرتب شده از بهترین پیشنهادات به بدترین، می‌دهد.

```
Computing the cosine similarity matrix...
Done computing similarity matrix.
Computing the cosine similarity matrix...
Done computing similarity matrix.
Inception (2010) 3.3
Star Wars: Episode V - The Empire Strikes Back (1980) 2.4
Bourne Identity, The (1988) 2.0
Crouching Tiger, Hidden Dragon (Wo hu cang long) (2000) 2.0
Dark Knight, The (2008) 2.0
Good, the Bad and the Ugly, The (Buono, il brutto, il cattivo, Il)
(1966) 1.9
Departed, The (2006) 1.9
Dark Knight Rises, The (2012) 1.9
Back to the Future (1985) 1.9
Gravity (2013) 1.8
Fight Club (1999) 1.8

In [2]:
```

[6] Accuracy
[7] Similarity matrix

توجه شود که برای داشتن یک سیستم پیشنهاد دهنده‌ی واقعی، همین اطلاعات کافی و راضی کننده است، اما برای بهتر کردن نتیجه‌ی پیشنهادات راه‌ها و فوت و فن‌های[8] مختلفی است که گزارش بعدی به آن می‌پردازم.

پالایش مشارکتی بر اساس آیتم

در این روش سیستم پیشنهاد دهنده، به جای اینکه ببیند کاربران شبیه شما چه چیزهایی را دوست داشته‌اند تا به شما پیشنهاد دهد، به آیتم‌های شبیه آیتم‌هایی که شما دوست داشته‌اید توجه می‌کند و آن آیتم‌ها را پیشنهاد می‌دهد.

چند دلیل وجود دارد برای اینکه پیشنهاد بر اساس آیتم بهتر از پیشنهاد بر اساس کاربر است، اولین دلیل این است که سلیقه کاربران ممکن است در طول زمان تغییر کند اما یک آیتم هیچ‌وقت تغییر نمی‌کند، پس توجه به شباهت چیزهایی که تغییر نمی‌کنند نتیجه بهتری را دارد. دلیل دیگری این است که تعداد آیتم‌ها کمتر از کاربران است، پس کار کردن با آن‌ها سریعتر و راحت‌تر است زیرا که ماتریس شباهت آن کوچکتر است. استفاده از پیشنهاد دهنده بر اساس آیتم همچنین تجربه کاربری[9] خوبی هم هست برای کاربرانی که تازه وارد سیستم شده‌اند، بدین صورت که پس از اینکه آیتم مورد علاقه‌شان را پیدا کردند، آیتم‌های شبیه به آن به کاربر پیشنهاد داده می‌شود و دیگر نیاز به پیدا کردن کاربران شبیه به کاربر موردنظر نیست.

به طور کلی، پردازش برای پالایش مشارکتی بر اساس آیتم، مانند بر اساس کاربر است تنها تفاوتی که دارد این است که سطر و ستون برعکس هست

	Bob	Ted	Ann
Indiana Jones	4		
Star Wars	5		5
Empire Strikes Back			5
Incredibles			5
Casablanca		1	

در اینجا کاربران بعد ما هستند و ما می‌خواهیم شباهت کسینوسی بین آیتم‌ها را بدست بیاوریم. که بصورت زیر می‌شود:

	Indiana Jones	Star Wars	Empire Strikes Back	Incredibles	Casablanca
Indiana Jones	1	1	0	0	0
Star Wars	1	1	1	1	0
Empire Strikes Back	1	1	1	1	0
Incredibles	1	1	1	1	0
Casablanca	0	0	0	0	1

[8] Tricks
[9] User experience

برای مثال، باب را در نظر بگیرید که تازه وارد سیستم شده است و فقط فیلم Star wars را تماشا کرده و امتیاز بالایی به آن داده، در اینجا سیستم ما باید بگردد و فیلم‌های شبیه به Star wars را پیدا کند بر اساس امتیازهایی که کاربرانی که Star wars را دوست داشته‌اند، به فیلم‌های دیگر داده‌اند. پس فیلم‌های Indiana Jones، Empire strikes back و Incredibles را به کاربر مورد نظر پیشنهاد می‌کند.

قسمت عملی

با استفاده از برنامه‌ی Spyder، کدهای فولدر CollaborativeFiltering را باز می‌کنیم. فایلی که با آن کار داریم و می‌خواهیم اجرایش کنیم SimpleItemCF.py است. همانطور که در کد مشاهده می‌کنید، بسیار زیاد شبیه به SimpleUserCF.py هست به دلیل اینکه به‌طور کلی، این دو روش شبیه به هم هستند. یکی از تفاوت‌هایشان این است که در کد بر اساس آیتم، در sim_options، گزینه‌ی user_based را false دادیم چون می‌خواهیم به کتابخانه‌ی Surprise بگوییم که یک ماتریس شباهت آیتم به آیتم با استفاده از شباهت کسینوسی بسازد. کد را که اجرا کنید، نتایج مبهمی نسبت به پیشنهاد دهنده بر اساس کاربر می‌گیرید، همچنین همانطور که مشاهده می‌کنید بیشتر فیلم‌هایی که پیشنهاد کرده برای دهه‌ی ۹۰ میلادی است، زیرا که کاربر شماره ۸۵ بیشتر فیلم‌های آن دوره را دوست داشته است.

```
Computing the cosine similarity matrix...
Done computing similarity matrix.
Computing the cosine similarity matrix...
Done computing similarity matrix.
James Dean Story, The (1957) 10.0
Get Real (1998) 9.987241120712646
Kiss of Death (1995) 9.966881877751941
Set It Off (1996) 9.963732215657119
How Green Was My Valley (1941) 9.943984081065269
Amos & Andrew (1993) 9.93973694500253
My Crazy Life (Mi vida loca) (1993) 9.938290487546041
Grace of My Heart (1996) 9.926255896645218
Fanny and Alexander (Fanny och Alexander) (1982) 9.925699671455906
Wild Reeds (Les roseaux sauvages) (1994) 9.916226404418774
Edge of Seventeen (1998) 9.913028764691676

In [2]:
```

شرکت آمازون از این روش سیستم پیشنهاد دهنده استفاده می‌کند.

تست پالایش مشارکتی

همانطور که می‌دانید، در پالایش مشارکتی نمی‌توان دقت[10] را اندازه گرفت، زیرا پیش بینی امتیازدهی انجام نمی‌دهیم اما می‌توانیم hit rate را اندازه بگیریم.

[10] Accuracy

قسمت عملی

برمی‌گردیم به Spyder و فولدری که باز کرده بودیم. فایل EvaluateUserCF.py را اجرا می‌کنیم، همانطور که می‌بینید، خیلی سریع اجرا شد زیرا نیاز نبود که برای هر آیتم و هر کاربر پیش بینی امتیاز دهی را انجام دهد.

```
Computing movie popularity ranks so we can measure novelty later...
Estimating biases using als...
Computing the cosine similarity matrix...
Done computing similarity matrix.
Computing the cosine similarity matrix...
Done computing similarity matrix.
Computing the cosine similarity matrix...
Done computing similarity matrix.
HR 0.05514157973174367

In [3]:
```

این مقدار hit rate برای یک سیستم پیشنهاد دهنده مناسب است.