« به نام او »

فاز اول پروژه ماشین لرنینگ

دکتر امینی

علیرضا صفرخانی(شماره دانشجویی: 97101983)

سجاد هاشم بیکی(شماره دانشجویی: 98107077)

زمستان1401

سوال 1)

سیستم های توصیه گر،الگوریتم هایی هستند که سعی میکنند با استفاده از دیتاهای موجود ایتم ها را به کاربران پیشنهاد بدهند.برای مثال این دیتاها میتواند شامل سوابق خرید یک کاربر در فروشگاه،بازدید های کابر از کالا و یا فیلم های دیده شده توسط کاربر و امتیازی که کاربربه آنها داده است، باشد.

این الگوریتم ها با مشاهده این دیتاها و یادگیری الگوهای موجود در دیتا،سعی میکنند براساس سلایق و اولویت های کاربران،ایتم ها را(ایتم شامل هرچیزی میتواند باشد مثلا فیلم،کالا،موسیقی،خدمات و...) به آنها پیشنهاد بدهند.

دو دسته اصلی سیستم های توصیه گر عبارتند از:

content-based و collaborative filtering

سوال 2)

برخی ازکاربرد های سیستم های توصیه گر عبارتند از:

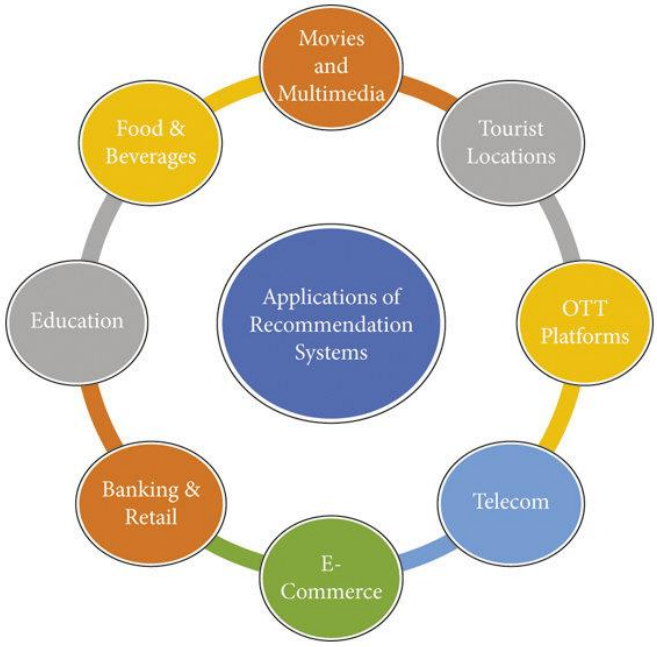
- پیشنهاد فیلم و سریال به کاربران که نمونه آن را در پلتفرم هایی مثل نتفلیکس،آمازون پرایم و.. میتوان مشاهده کرد.

- پیشنهاد موزیک وپادکست به کاربران که برای مثال در اسپاتیفای میتوان نمونه آن را دید.

- پیشنهاد کالاها در فروشگاه های انلاین مانند امازون

- یافتن افراد مشابه و پشنهاد آنها به کاربران درشبکه های اجتماعی مانند اینستاگرام

- و....



سوال 3)

چالش ها:

- یکی از چالش ها cold start است.به این معنا که وقتی یک کاربر جدید وارد سیستم میشود(مثلا وقتی تازه اکانت خود را در نتفلیکس یا اسپاتیفای ساخته اید)،دیتای مناسبی وجود ندارد تا بتوان براساس آن ایتم ها را به کاربر پشنهاد داد.البته راه حل هایی برای این چالش وجود دارد،مانند کاری که اسپاتیفای در شروع میکند.ابتدا سوالات مختلفی را از کاربر میپرسد تا به یک شناخت نسبی برسد.

-چالش دیگر سنجش عملکرد سیستم توصیه گر است.معیارهای مختلفی برای سنجش وجود دارد که بایستی با توجه به شرایط انتخاب شود.

-چالش دیگری که با آن مواجه هستیم تنک (sparse) بودن ماتریس utility میباشد.چون بطور مثال همه کاربران به فیلم هایی که دیده اند امتیاز نمیدهند(یا مسلما همه کاربران همه فیلم ها را ندیده اند) و این موضوع باعث تنک شدن ماتریس خواهد شد.

- چالش مهم دیگراین است که سلایق کاربر ممکن است درطول زمان تغییر کند ویا حتی بعضی سلایق موقتی باشند.(برای مثال به سفر رفتید و موزیک های مناسب سفر را در اسپاتیفای پلی میکنید،قاعدتا بعد از اتمام سفر نیازی به توصیه شدن این سبک موسیقی نمیباشد.)

سوال 4)

در روش های مختلف ریکامند کردن معیار Similarity حائز اهمیت بسیاری است. روش های مبتنی بر شباهت انواع گوناگونی دارند.

یک دسته از روش ها مربوط به حالتی است که شباهت بین کاربران مختلف محاسبه شود و اگر شباهت از یک ترشهولد مشخص بیشتر باشد یعنی این دو کاربر علایق مشابه دارند و سپس تعدادی از کاربرانی که بیشترین شباهت را با کاربر موردنظر ما دارند انتخاب می شوند و مجموع وزن دار اختلاف میانگین نمراتشان با نمره ای که به یک ایتم خاص می دهند محاسبه شده و در نهایت با میانگین نمرات خود کاربر موردنظر جمع شده و پیش بینی می شود نمره ای که کاربر به یک ایتم خاص احتمالا می دهد چه نمره ای است. انتظار می رود نمره ای مشابه با افرادی باشد که سلیقه نزدیک به این فرد دارند.

دسته دیگری از روش ها مبتنی بر شباهت بین ایتم هاست. یعنی برای یک کاربر خاص بر اساس نمراتی که به ایتم های قبلی داده است تصمیم می گیریم که برای ایتم مشابه ان ها چه نمره ای خواهد داد و تصمیم می گیریم آن را به او توصیه بکنیم یا خیر.

خود معیار شباهت هم انواع گوناگون دارد. مثلا یکی از شباهت های ساده و معروف معیار شباهت کسینوسی یا همان پیرسون کورلیشن می باشد. معیار های بسیار دیگری هم وجود دارد که هم می توان شباهت بین هر دو کاربر را با ان ها سنجید و هم شباهت بین دو ایتم. یکی از ایده های جدید تر هم استفاده از شباهت ساختاری می باشد که داده ها را به صورت شبکه در می اورند و روی آن شبکه تعریف های شباهت را ارائه می دهند که در بسیاری از موارد از شباهت های کلاسیک بهتر عمل کرده است.

سوال 5)

در این سوال به انواع روش های هیبریدسازی سیستم های ریکامندر می پردازیم. به صورت موردی آن ها را نام برده و برای هر کدام توضیح مختصری هم ارائه می دهیم:

1) Weighted :

در این روش امتیاز ها یا آرای حاصل از چندین تکنیک ریکامندیشن به صورت وزن دار با یکدیگر ترکیب شده تا در نهایت فقط یک امتیاز بدست آید که بر اساس آن ریکامندیشن انجام شود. مثلا یک حالت ساده این می تواند باشد که فقط ترکیب خطی شان لحاظ شود. البته به صورت کلی مزیت این روش این است که از چندین روش ریکامندیشن در آن استفاده شده اما فرض مهمی که دارد این است که مقادیر نسبی عدد های حاصل از روش های مختلف کم و بیش به صورت یونیفرم هستند اما می دانیم در واقعیت و در عمل این فرض همیشه صحیح نیست.

2) Switching:

در این روش سیستم ریکامندر با توجه به وضعیتی که در آن قرار دارد بین تکنیک های مختلف ریکامندیشن سوییچ می کند. این رویکرد می تواند سیستم کلی را پیچیده تر کند چرا که تعیین این شرط که چه موقع سیستم بین روش های مختلف سوییچ کند خودش پارامتر های جدید وارد مسئله می کند. اما از طرفی این حساس بودن سیستم به نقاط قوت و ضعف یک روش که باعث می شود روش را عوض کند نکته مثبتی است که به نتیجه خوبی منتهی می شود.

3) Mixed:

در این رویکرد از چند تکنیک ریکامندیشن همزمان استفاده شده و نتایج حاصل و ریکامندها هم به صورت همزمان ارائه می شوند بنابراین خروجی فقط یک چیز نیست بلکه خروجی هر روش جداگانه ارائه می گردد. برای مواقعی که تعداد زیادی توصیه همزمان برای ارائه نیاز باشد و عملی هم باشد روش خوبی است.

4) Feature Combination:

در این رویکرد از فیچرهایی که از منابع دیتای گوناگون از سیستم های ریکامندیشن مختلف آمده اند استفاده شده و همه فیچرها با هم برای یک الگوریتم ریکامندیشن استفاده می شوند و به این صورت انگار که به نوعی از ترکیبشان در یک روش استفاده کرده ایم. این روش هم حساسیت کمتر نسبت به تعداد کاربران خواهد داشت و هم شباهت بین ایتم ها را از روی ویژگی ها در نظر می گیرد.

5) Cascade:

در این روش دو سیستم ریکامندر پشت سر هم قرار می گیرند یعنی ابتدا یک روش اعمال شده و سپس نتیجه آن به روش بعدی منتقل شده و به نوعی در آن سیستم نتیجه قبلی بهبود می یابد تا در نهایت سیستم کلی بهترین ریکامندیشن را انجام دهد. در واقع این روش در سیستم دوم فقط مواردی را لحاظ می کند که در سیستم اول پیشنهاد شده اند و موارد دور ریخته شده یا با اولویت کمتر را کاری نمی کند. همچنین نسبت به نویز مقاومت خوبی دارد.

6) Feature Augmentation:

در این روش خروجی یک تکنیک خودش به عنوان فیچر ورودی به یک سیستم ریکامندر دیگر داده می شود. تفاوتش با روش شماره 4 این است که آنجا فیچرهای خام از منابع مختلف ترکیب می شدند ولی اینجا ابتدا یک روش اعمال شده و نتیجه اش به عنوان فیچر جدیدی به سیستم بعدی داده می شود که میتواند نتیجه را بهبود دهد.

7) Meta-level:

در این روش کل یک مدل که با یک ریکامندر یاد گرفته شده است به عنوان ورودی به یک ریکامندر دیگر داده می شود. تفاوت با روش قبلی این است که در انجا یک مدل یاد گرفته می شد و سپس آن مدل تعدادی فیچر تولید می کرد که آن فیچر ها به ریکامندر سیستم بعدی داده می شد ولی در اینجا کل مدل به عنوان ورودی داده می شود. حسن این روش این است که آن مدل یادگرفته شده در واقع یک خلاصه و چکیده از علایق کاربر را نشان می دهد و وقتی به سیستم بعدی داده شود بهتر از حالتی که فقط دیتای خام نمره های کاربر را داریم عمل می کند.

سوال 6)

الف) در هر مسئله ریکامندیشن تعدادی کاربر یا یوزر داریم و تعدادی هم آیتم داریم که هر یک از کاربران به آن آیتم ها بر اساس علاقه شان نمره داده اند و یا اصلا نمره نداده اند. برای نشان دادن این ارتباط جفت جفت بین کاربر و آیتم های مختلف از جدولی استفاده می شود که ماتریس یوتیلیتی نامیده می شود. هر درایه آن نشان می دهد نمره یا میزان علاقه یک کاربر به یک آیتم چقدر است.

ب) بله در عمل معمولا این ماتریس اسپارس است یعنی بیشتر درایه هایش خالی است یعنی نمی دانیم مقدارشان چقدر است چرا که افراد کاربر در دنیای واقعی معمولا درصد بسیار کمی از محصولات را نمره دهی می کنند.

ج) این جمله غلط است. هدف این نیست که تک تک خانه های خالی ماتریس مقدارش پیش بینی شود. بلکه در بسیاری از اپلیکیشن ها تنها ضرورت دارد در هر سطر ماتریس فقط آن مواردی که احتمالا نمره های بالایی دارند پیش بینی شود. این کار منطقی هم هست چون کاربر به هرحال حتی اگر همه موارد هم برایش پیش بینی شود و لیست بلندی به او پیشنهاد شود فقط موارد بالای لیست را که به آنها علاقه بیشتری دارد نگاه می کند و تا انتهای لیست بررسی نمی کند.

د) به صورت کلی دو رویکرد زیر برای تشکیل و پر کردن ماتریس یوتیلیتی وجود دارد:

1) یک روش این است ما از افراد بخواهیم به محصولات یا محتوا یا فیلم نمره دهند. مشکل این روش این است که خیلی از افراد تمایل به ثبت نمره هایشان ندارند و همچنین آن مواردی هم که نمره شان را در اختیار داریم در واقع نمره ها بایاس هستند چرا که فقط نمره های افرادی را داریم که تمایل داشته اند نمره بدهند. نمره دهی هم معمولا بین 0 تا 5 است که معمولا با تعدادی ستاره نمره می دهند.

2) در این روش ما خودمان با توجه به رفتار و فعالیت افراد تصمیم می گیریم که به یک ایتم خاص علاقه دارند یا ندارند. در این رویکرد معمولا به صورت 0 و 1 جدول را پر می کنیم. 1 برای وقتی است که یک فرد ایتمی را خریده یا فیلمی را تماشا کرده و ما حدس می زنیم آن را دوست دارد. عدد 0 هم به این معناست که اصلا نمره برای ان ایتم نداریم. البته مشکل این روش هم این است که ممکن است کسی فیلمی را ببیند ولی از آن خوشش نیاید اما ما فرض را بر این گرفته ایم که دوست دارد.

سوال7)

a)

در روش content-based شباهت ایتم ها را با توجه به ویژگی آنها میسنجیم.برای مثال اگر دو ایتم Aو B مشابه باشند و یک کاربر ایتم A را دوست داشته باشد(مثلا ایتم یک فیلم است و کاربر به فیلم A امتیاز بالا داده)،پس احتمالا ایتم B را نیز دوست دارد.بنابراین ایتم B میتواند یک پیشنهاد به کاربر باشد.

b)

1-کارگردان 2-مجموعه بازیگران 3-ژانر 4-سال ساخت 5-امیتازفیلم(مثلا امیتازIMDB)

6-کشورسازنده 7-شرکت سازنده 8-میزان فروش 9- نویسنده 10-بودجه ساخت و میزان سود دهی

c)

Movie1= [0 1 1 0 1 1 0 1 3a]
Movie2= [1 1 0 1 0 1 1 0 4a]

در اینجا یکی از ویژگی ها عددی است(امیتاز) و ویژگی دیگر به صورت صفرو یکی(boolean).
برای محاسبه کوسینوس زاویه میان دو بردار،فیچرهای عددی را میتوان اسکیل کرد.ضریب a را به عنوان فاکتوراسیکل در نظر میگیریم و براساس این پارامتر کوسینوس زاویه بین دو بردار با بدست میاوریم.

$$Cos = (2 + 12a^2)/sqrt(25 + 125a^2 + 144a^4)$$

-مثلا به ازای1 = a کسینوس برابر 0.816 و به ازای a=2 برابر با 0.940 خواهد بود.بنابراین بسته به این که چه مقداری برای پارامترa انتخاب کنیم مقدار کسینوس متفاوت خواهد بود.

(d

از بین فیلم هایی که کاربر مورد نظر دیده است،در 20 درصد انها جولیا رابرتز بازی کرده است.بنابراین مولفه مربوط به این بازیگر در پروفایل کاربر برابر با 0.2 خواهد بود.

(e

ابتدا امتیازی که هر کاربر داده است را نرمالایز میکنیم(میانگین امتیازات داده شده توسط هرکاربر را از امتیازی که کاربر به هر فیلم داده است را کم میکنیم)،و سپس میانگین میگیریم.عدد بدست امده مولفه بازیگر مورد نظر در پروفایل کاربر خواهد بود.

U: ((3 – 3) + (4 – 3) +(5 – 3)) / 3 =1

V: ((2 – 4) + (3 – 4) +(5 – 4)) / 3 =-2/3 = -0.67

-بنابراین مولفه مربوط به بازیگر جولیا رابرتز در پروفایل کاربران U و V به ترتیب برابر با 1 و0.67- میباشد.

(f

cos (m1, m2): $\frac{7.599+100000a^2+15b^2}{\sqrt{6.5+62500a^2+9b^2}\ \sqrt{8.88+160000a^2+25b^2}}$

cos (m1, m3): $\frac{8.552+240000a^2+25b^2}{\sqrt{8.23+360000a^2+25b^2}\ \sqrt{8.88+160000a^2+25b^2}}$

cos (m2, m3): $\frac{7.318+150000a^2+15b^2}{\sqrt{6.50+62500a^2+9b^2}\ \sqrt{8.23+360000a^2+25b^2}}$

(g

a=b=1

cos (m1, m2): 1 →angel=0

cos (m1, m3): 1→ angel=0

cos (m2, m3): 0.9996 → angel=1.62

(h

a=0.01, b=0.5

cos (m1, m2):0.9888→ angel=8.5

cos (m1, m3): 0.9795→ angel=11.62

cos (m2, m3):0.9487→ angel=18.43

(i

mean of f2=416.67

mean of f3=4.3

بنابراین ضریب a و b بدین صورت خواهند بود:

a = 1/mean of f2 = 0.002

b = 1/mean of f3 = 0.232

cos (m1, m2):0.9930→ angel=6.78

cos (m1, m3): 0.9921→ angel=7.2

cos (m2, m3):0.9769→ angel=12.33

(j

mean of f1= 2.8

mean of f2=416.67

mean of f3=4.3

بردار های نرمالایز شده:

	m1	m2	m3
f1	0.18	-0.25	0.07
f2	-16.67	-166.67	183.33
f3	0.7	-1.3	0.7

(k

یک راه برای اسکیل کردن بردار های نرمالایز شده،تقسیم بر انحراف معیار است.با اینکار انحراف معیار هر مولفه برابر یک خواهد بود.و چون نرمالایز شده هستند،میانگین نییز برابر صفر میباشد.(یک راه دیگر برای اسکیل کردن تقسیم بر ماکسیمم است)

داده های قسمت قبل پس از نرمالایز و اسکیل:

	m1	m2	m3
f1	0.986	-1.370	0.383
f2	-0.116	-1.162	1.278
f3	0.707	-1.414	0.707

هر چه زاویه بین دو بردار کمترباشد فاصله کوسینوسی دو بردار کمتر بوده و به معنای شباهت بیشتر دو بردار(دو ایتم مدنظر) خواهد بود.

زاویه بین بردارها پس از نرمالایز کردن:

m1,m2: حدود صفر درجه

m1,m3:حدود 180 درجه

m2,m3:حدود 180 درجه

(l

mean of rates =(4+2+5)/3=3.67

m1= 4-3.67=0.33

m2=2-3.67=-1.67

m3=5-3.67=1.33

(m

ساخت پروفایل کاربر:

f1 = 4*2.98+2*2.55+5*2.87=31.37

f2 =4*400+2*250+5*600=5100

f3 =4*5+2*3+5*5=51

user profile:[31.37,5100 ,51]

همچنین میتوان امتیازات کاربر را اسکیل کرد مثلا بر ماکسیمم که 5 است تقسیم کنیم.که در این صورت داریم:

user profile:[6.274,1020 ,10.2]

سوال 8)

پاسخ این مسئله به صورت دستی نوشته شده و در فایل جداگانه ای تحویل داده شده است.

سوال 9)

بطور کلی،کاهش بعد روشی است که در آن سایز دیتا را کاهش میدهیم و در عین حال اطلاعات اصلی را حفظ میکنیم.

روش های مختلفی برای اینکار وجود دارد که عبارتند از:

SVD

Bayesian Clustering

Probabilistic Latent Semantic Analysis (p-LSA)

Latent Dirichlet Allocation (LDA)

UV-Decomposition

همانطور که میدانیم ماتریسUtility اسپارس است،با استفاده از تجزیه U-V تقریبی از ماتریس Utility میزنیم(با توجه به درایه های موجود در ماتریس) و سپس با محاسبه ضرب ماتریسیUV به تخمینی از درایه های خالی در ماتریس Utility میرسیم.