

گزارش کتبی سمینار داده کاوی

موضوع: سیستم‌های پیشنهاد دهنده

پژوهشگران:

امیرحسین بینش

شهریار شهبازی

محمد مهدی صابرماهانی

استاد راهنما:

دکتر مزلقانی

بهمن ۹۹

۱. مقدمه

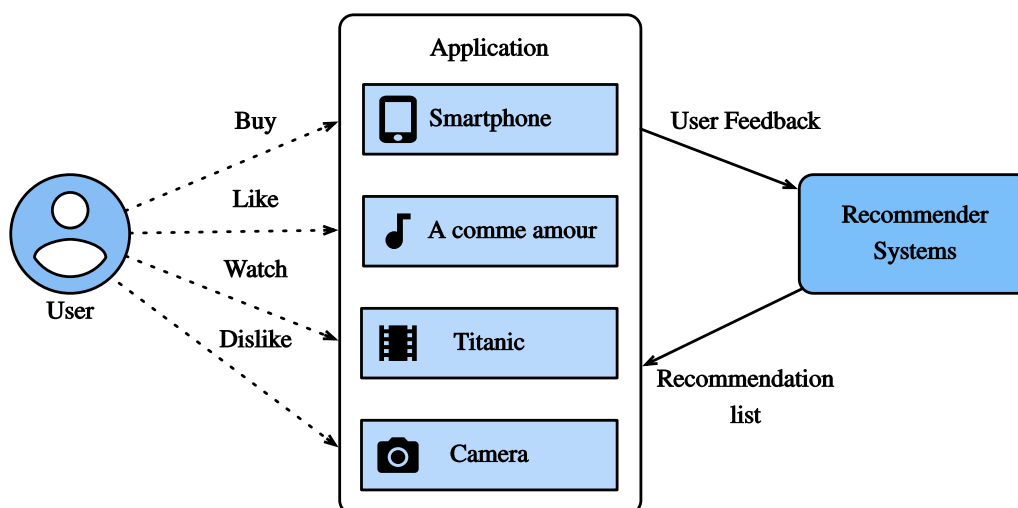
۱.۱ سیستم پیشنهاد دهنده چیست؟

سیستم‌های پیشنهاد دهنده، زیرمجموعه‌ای از فیلتر اطلاعات^۲ هستند که هدفشان، تخمین امتیاز کاربران به آیتم‌ها هست. با تخمین زدن امتیاز هر کاربر به هر آیتم، سیستم‌های پیشنهاد دهنده می‌توانند، امتیازات برتر را به عنوان "پیشنهاد" به کاربر مربوطه نمایش دهند تا هر کاربر بهترین آیتم‌ها را به توجه به سلیقه‌ی خود مشاهده کند.

در بحث سیستم‌های پیشنهاد دهنده، سه مورد اساسی وجود دارد:

- **کاربران:** کاربرانی که از سیستم استفاده می‌کنند و به آنها پیشنهاد داده می‌شود
- **آیتم‌ها:** موارد موجود در سیستم که به کاربران پیشنهاد داده می‌شود
- **امتیازات:** مجموعه امتیازهایی که کاربران به آیتم‌های مختلف می‌دهند

همانطور که در شکل ۱ دیده می‌شود، کاربر پس از ورود به سیستم، می‌تواند فعالیت‌های انجام دهد و آیتم‌هایی را مشاهده کند و - احتمالا - به چندتا از آنها امتیاز بدهد. سیستم پیشنهاد دهنده با بررسی تاریخچه کاربر، سلیق احتمالی کاربر را استخراج می‌کند و آیتم‌های جدید مفیدی به او ارائه می‌دهد.



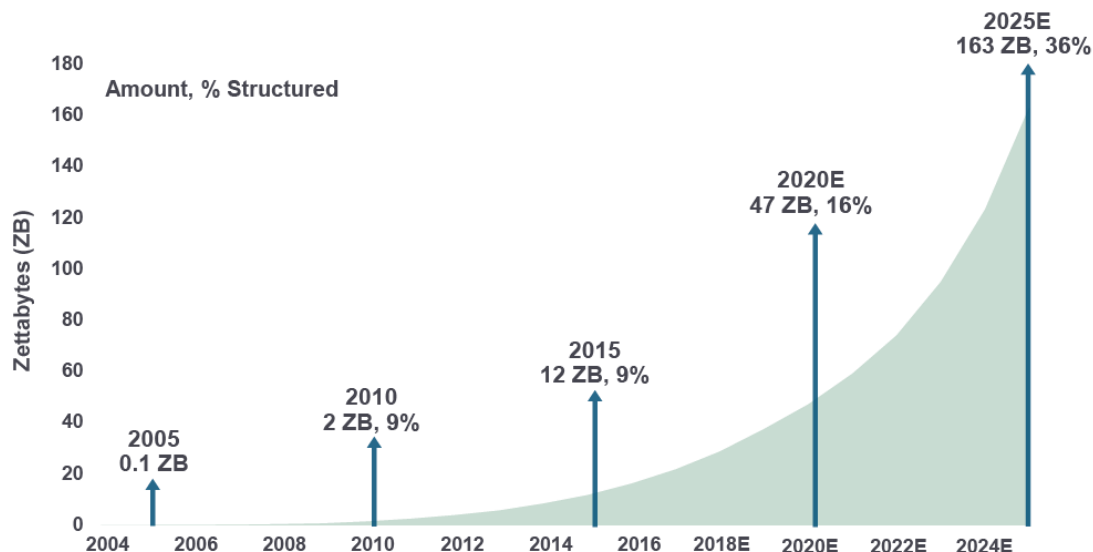
شکل ۱ - نحوه تعامل سیستم‌های پیشنهاد دهنده با کاربران

¹ Recommender system

² Information filtering

۱.۲ چرا از سیستم‌های پیشنهاد دهنده استفاده می‌کنیم؟

با زیاد شدن داده‌های موجود در اینترنت (شکل ۲) نیاز به سیستمی احساس می‌شود که بتواند بهترین نتایج را به کاربران پیشنهاد بدهد.



شکل ۲ - پیش‌بینی افزایش داده در اینترنت و درصد داده‌های ساخت یافته

در ابتدا سیستم‌های بازیابی اطلاعات^۱ معرفی شدند که برای زمانی مناسب هستند که کاربران در جستجوی موردی هستند که می‌دانند چیست. موتورهای جستجو^۲ نمونه‌ای از این سیستم‌ها هستند که با ورودی مستقیم از کاربر، مفیدترین و مرتبط‌ترین صفحات را در اینترنت پیدا می‌کنند و به کاربر نمایش می‌دهند. این سیستم‌ها معمولاً برای داده‌های ساخت‌نیافته^۳ استفاده می‌شود، چرا که برای پایگاه‌داده‌های ساخت یافته، جستجو با زبان‌هایی مانند SQL آسان است.

کاربرانی که نمی‌دانند چه چیزی می‌خواهند، با استفاده از موتورهای جستجو به نتایج شخصی نمی‌رسند. برای مثال اگر کاربری فیلم‌های رئال دوست دارد، در حالی که بازخورد مردم از فیلم‌های این سبک خوب نباشد، عموماً نمی‌تواند به نتیجه‌ی خوبی با استفاده از موتورهای جستجو برسد. به این دلیل سیستم‌های جدیدی ساخته شدند که با جای بازیابی، اطلاعات موجود را فیلتر می‌کرد.

با افزایش داده‌های موجود در اینترنت، وبسایت‌ها و شبکه‌های اجتماعی ایجاد شدند که اطلاعات و محتوای فراوانی از هر نوع دارند. برای راضی نگه داشتن کاربران، این سایت‌ها باید بتوانند محتوای خود را با سلیقه‌ی هر کاربر وفق دهند تا کاربران به استفاده از آنها ادامه بدهند. برای مثال برای شخصی حامی

¹ Information retrieval

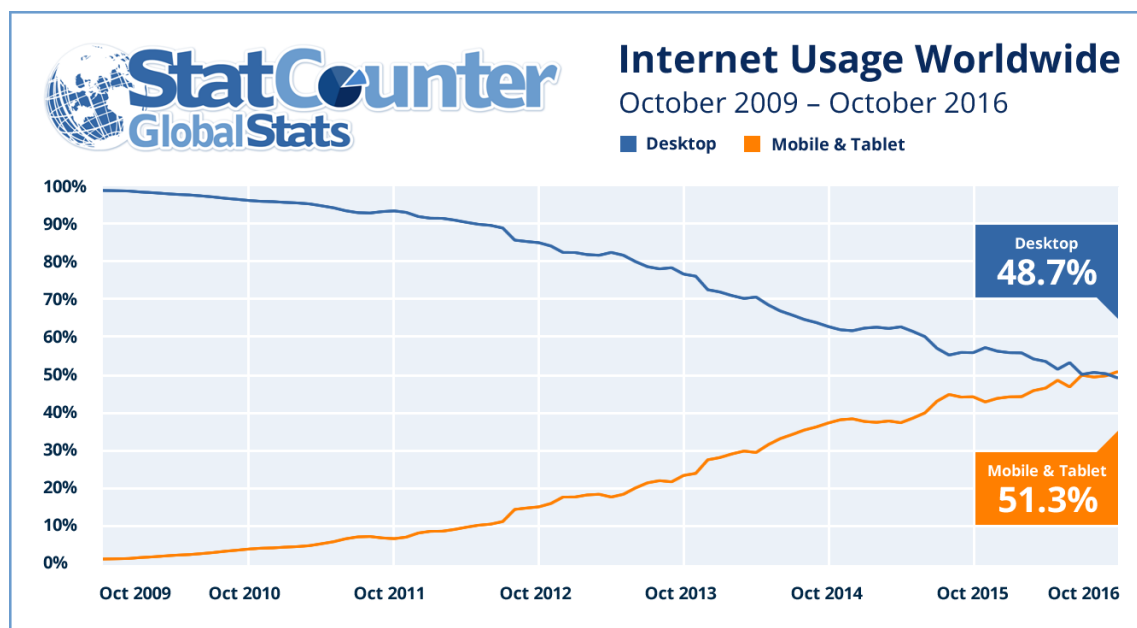
² Search engines

³ Unstructured

دموکراسی در توییتر، توییت‌هایی نمایش داده می‌شود که حامی دموکراسی باشد و اگر این شخص از سرمایه‌داری تنفر دارد، این توییت‌ها در صفحه اصلی^۱ او دیده نمی‌شوند.

علاوه بر این کاربران وقت کافی برای مرور و مشاهده همه‌ی محتوای یک سایت یا شبکه اجتماعی را ندارند؛ پس برای استفاده بهینه از وقت، فقط باید مطالبی را ببیند که مهم‌تر و با ارزش‌تر هستند تا بتواند در زمان کمتری، اطلاعات بیشتری دریافت کند.

نکته‌ی دیگر، محدودیت صفحه‌های نمایش است. با ظهور گوشی‌های هوشمند، استفاده از کامپیوتر-های خانگی در چند سال اخیر، رشد سابق خود را ندارد (شکل ۳). صفحه‌های گوشی‌های هوشمند کوچکتر هستند و طراحان رابط کاربری^۲ در نمایش داده‌ها محدودیت دارند. با کاهش آیتم‌های صفحه نمایش با استفاده از روش‌های فیلتر کردن، این مشکل قابل حل کردن است.



شکل ۳ - مقایسه سهم استفاده از اینترنت بین موبایل و کامپیوترهای شخصی

۱.۳ سیستم‌های پیشنهاد دهنده چگونه به ما کمک می‌کنند؟

سیستم‌های پیشنهاد دهنده هم با کاربران اینترنت کمک می‌کنند و هم به سرویس دهنده‌ها و صاحبان کسب و کارهای اینترنتی.

کاربران:

¹ Feed

² User interface

- زمان کمتری برای جستجوی آیتم موردنظر سپری می‌کنند؛ چون سیستم پیشنهاد دهنده با پیشنهادهای هوشمندانه نیاز کاربران به جستجوهای زنجیره‌ای را از بین می‌برد.
- تصمیم‌گیری ساده‌تری خواهند داشت؛ چون سیستم‌های پیشنهاد دهنده با حذف بسیاری از آیتم و نشان دادن مفیدترین آیتم‌ها، کاربران را از شک نسبت به انتخاب آیتم‌ها رها می‌کند.
- آیتم‌های باکیفیت تری انتخاب می‌کنند؛ چون پیشنهادهای انجام شده می‌تواند علاوه بر سلیقه‌ی کاربران، بر اساس کیفیت خود آیتم‌ها باشد و به این صورت، آیتم‌های با امتیاز یا کیفیت کم نمایش داده نمی‌شود، مگر اینکه خود کاربر دقیقاً همان آیتم را بخواهد.

سرویس دهندگان:

- افزایش رضایت کاربران؛ افزایش رضایت کاربران یکی از مهم‌ترین عوامل موفقیت یک کسب و کار اینترنتی است. این قدم اول در مسیر موفقیت‌های بعدی است؛ چون بدون کاربران راضی، هیچ کسب و کاری قادر به ادامه نخواهد بود.
- افزایش نرخ بازگشت کاربران؛^۱ با رضایت کاربران، نرخ بازگشت آنها به اپلیکیشن یا وبسایت افزایش پیدا می‌کند. درصد مهمی از کاربران، بعد از اینکه اپلیکیشنی را نصب می‌کنند در صورت نارضایتی، آن را حذف می‌کنند و دیگر به سمت آن برنمی‌گردند. ایجاد یک حس خوب در کاربران جدید با پیشنهاد آیتم‌های جذاب و مفید، باعث می‌شود آنها دوباره به اپلیکیشن یا وبسایت برگردند.
- افزایش درآمد؛ در نهایت با عملی شدن دو مورد قبل، یک کسب و کار موفق خواهد شد و به درآمد خواهد رسید.

۱.۴ کاربرد های سیستم‌های پیشنهاد دهنده در عمل

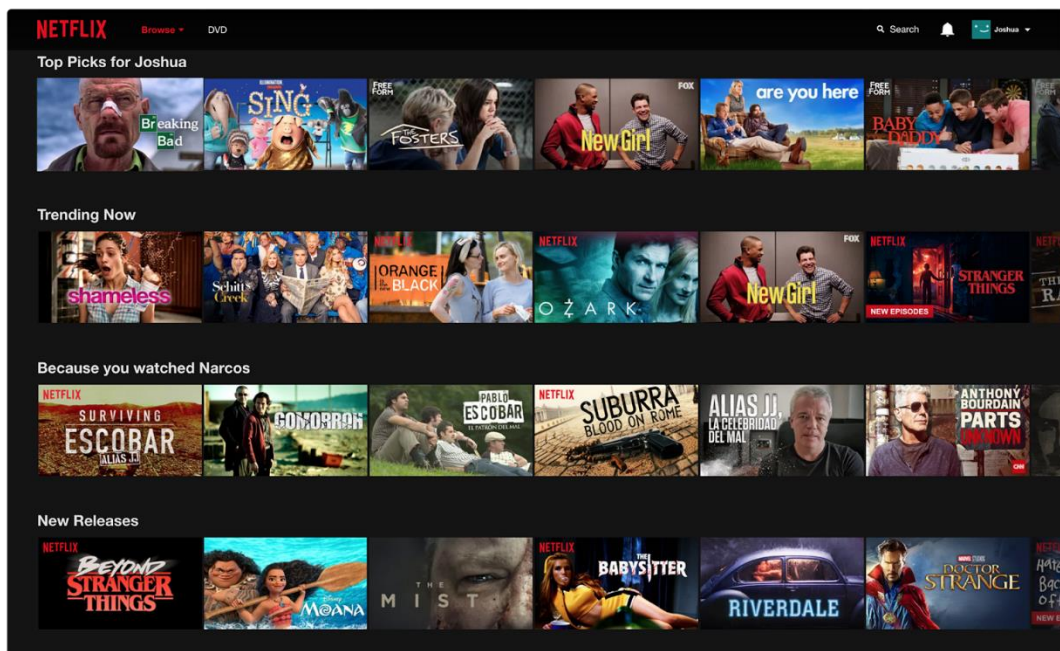
سیستم‌های پیشنهاد دهنده در وبسایت‌ها و اپلیکیشن‌های زیادی کاربرد دارند. در این قسمت، بعضی از کاربردهای این سیستم‌ها در واقعیت را مرور می‌کنیم و مواردی را نیز معرفی خواهیم کرد.

۱.۴.۱ پیشنهاد دهنده ویدئو

این سیستم‌ها با توجه به نوع ویدئوی تماشا شده و اطلاعات آن ویدئو، می‌توانند ویدئوهای مشابه را پیشنهاد دهند. علاوه بر این با استخراج شباهت کاربران می‌توانند ویدئو-های جدیدی که یک گروه احتمالاً دوست داشته باشند را نیز پیشنهاد دهند. در شکل ۴ نمونه‌ای از این سیستم‌ها، نتفلیکس^۲، دیده می‌شود.

^۱ User retention rate

^۲ Netflix



شکل ۴ - رابط کاربری نتفلیکس

در شکل ۴، رابط کاربری نتفلیکس شامل ۴ بخش زیر می‌باشد:

- بهترین پیشنهادهای شخصی
- بهترین پیشنهادهای کل جامعه در زمان حال
- پیشنهاد فیلم‌های شبیه یکی از فیلم‌هایی که کاربر خیلی دوست داشته
- جدیدترین فیلم‌های منتشر شده

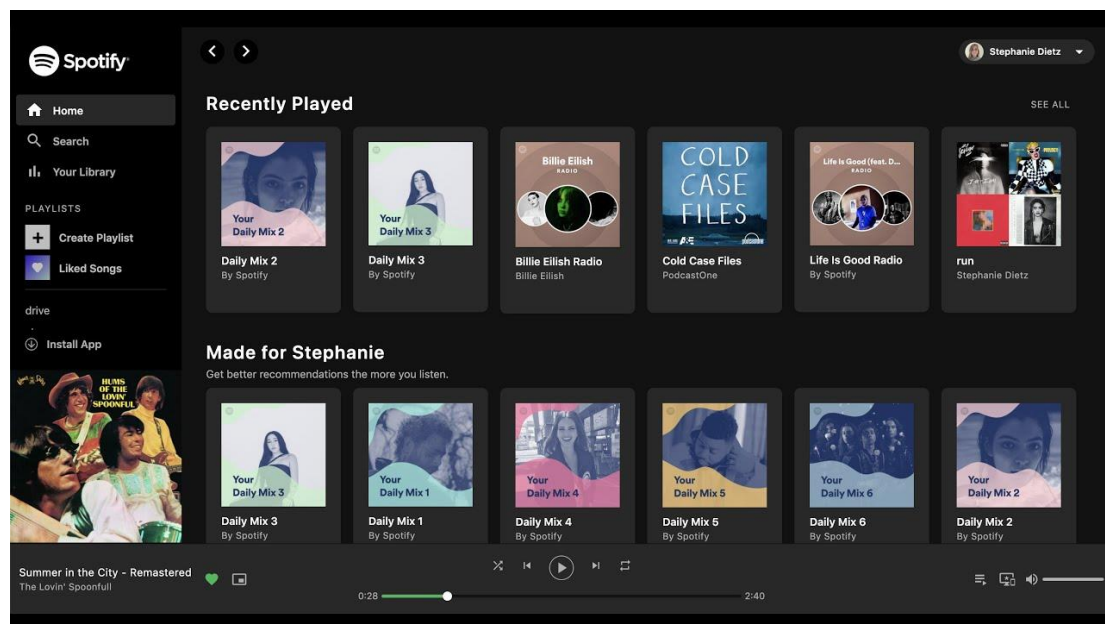
از نمونه‌های دیگر سیستم‌های پیشنهاد دهنده فیلم، می‌توان به یوتیوب^۱ اشاره کرد که مواردی مانند زمان فیلم (مثلا شاید یک فرد فقط فیلم‌های کوتاه دوست داشته باشد) را هم در نظر می‌گیرد.

۱.۴.۲ پیشنهاد دهنده موسیقی

این سیستم‌ها برخلاف پیشنهاد دهنده‌های فیلم، برپایه امتیاز کاربران به آهنگ‌های نیستند؛ چون آهنگ‌ها کوتاه هستند، امتیاز دادن به تک تک آنها کار دشواری برای کاربران است و آنها را از هدف اصلی اپلیکیشن دور می‌کند. به همین خاطر این سیستم‌ها فقط با مانیتور کردن تاریخچه موسیقی‌های گوش داده شده‌ی هر کاربر و تعداد تکرار آنها، سلیق کاربران را مدل می‌کند.

¹ YouTube

از یکی از مثال‌های موفق این پیشنهاد دهنده‌ها می‌توان به اسپاتیفای^۱ اشاره کرد (شکل ۵)



شکل ۵ - رابط کاربری اسپاتیفای

همانطور که در شکل ۵ مشاهده می‌شود، اسپاتیفای با ساختن میکس روزانه^۲ مخصوص هر کاربر، لیست موسیقی‌هایی از ژانرها و هنرمندانی می‌سازد که کاربر از آنها خوشش می‌آید.

۱.۴.۳ پیشنهاد دهنده اجتماعی

شبکه‌های اجتماعی در سال‌های اخیر رشد قابل توجهی داشته‌اند. الگوریتم‌های اجتماعی با شبیه‌سازی گراف روابط کاربران می‌توانند به کاربران، اشخاصی که احتمالاً می‌شناسند را پیشنهاد دهند. اما شبکه‌های اجتماعی که بر پایه دوست‌یابی هستند نمی‌توانند از گراف اجتماعی استفاده کنند؛ چرا که به دنبال افراد شبیه هستند، نه افراد آشنا. سیستم‌های پیشنهاد دهنده می‌توانند افرادی که سلیقه‌ی یکسانی دارند را به هم معرفی کنند تا یک شبکه‌ی دوستیابی را پایه‌گذاری کنند.

علاوه بر این، برای پیشنهاد محتوا در صفحات اجتماعی نیز، می‌توان از سیستم‌های پیشنهاد دهنده استفاده کرد. برای مثال قسمت کاوش^۳ در اینستاگرام^۴، محتوای مشابه به سلیقه‌ی کاربران را نشان می‌دهد که دوستان کاربران آنها را منتشر نکردند.

¹ Spotify

² Daily mix

³ Explore

⁴ Instagram

۱.۴.۴ پیشنهاد دهنده کالا

در سایت‌هایی مثل آمازون^۱ که برای خرید کالا بصورت دیجیتالی هستند، دو نوع پیشنهاد برای هر کالا وجود دارد:

- کالاهای مشابه: فرض کنید یک کاربر می‌خواهد یک تلفن همراه بخرد و در حال جستجو است. سیستم پیشنهاد دهنده، تلفن‌های مشابه به هر تلفن را در صفحه خودش اضافه می‌کند، مثلاً اگر یک گوشی حافظه ۶۴ گیگابایت دارد، مدل ۳۲ گیگابایتی آن را نیز پیشنهاد می‌دهد.
- کالاهایی که دیگران پس از خرید این کالا خریده‌اند: برای مثال گوشی، این مورد می‌تواند کاور، گلس و یا احتمالاً شارژر مربوط به این گوشی باشد.

دو مورد بالا در سایت آمازون مربوط به سلیق کاربر نیست. موارد مربوط به سلیق کاربر در صفحه اصلی قابل مشاهده است که به دلیل ذات خرید با مواردی مثل پیشنهاد فیلم فرق دارد؛ چون احتمالاً کسی که گوشی موبایل خریده، دیگر تا مدتی نیازی به گوشی موبایل ندارد، برای همین این موارد باید مدنظر قرار گیرند و یک مدل داخلی ساخته شود از اینکه هر کاربر چه کالایی را دارد و چه چیزهایی نیاز دارد.

۱.۴.۵ پیشنهاد دهنده بازی و یافتن بازیکن برای بازی^۲

پیشنهاد بازی، در حالت کلی مانند پیشنهاد فیلم است. یکی از فروشگاه‌های بازی دیجیتالی، استیم^۳، بازی‌های مختلف را بر اساس سلیق کاربر، مقدار پول موجود در کیف پول کاربر، دوستان کاربر و غیره پیشنهاد می‌دهد.

علاوه بر این در بازی‌های ویدئویی آنلاین، باید بازیکنانی پیدا شوند تا بازی بصورت عادلانه و مساوی انجام بگیرد و کاربران اوقات خوبی داشته باشند. یکی از این بازی‌ها، اوراچ^۴ نام دارد که یک بازی ۵ در برابر ۵ است. برای هر دور بازی باید ۱۰ بازیکن آماده پیدا شود و طوری در دو تیم قرار گیرند که تعادل بازی حفظ شود. علاوه بر این با توجه به نوع بازی هر بازیکن، بازیکنان هم تیمی باید طوری پیشنهاد شوند، که مکمل هم باشند و هر کدام در یک نقش خوب بازی کنند.

^۱ Amazon

^۲ Matchmaking

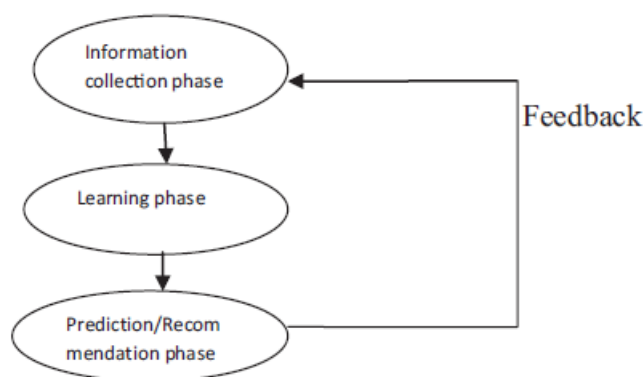
^۳ Steam

^۴ Overwatch

۱.۵ مراحل اصلی سیستم‌های پیشنهاد دهنده

همانطور که در شکل ۶ دیده می‌شود، سیستم‌های پیشنهاد دهنده سه فاز اصلی دارند:

- جمع‌آوری اطلاعات
- یادگیری
- پیش‌بینی/پیشنهاد



شکل ۶ - فازهای سیستم پیشنهاد دهنده

در فاز جمع‌آوری اطلاعات، سیستم با مشاهده فعالیت‌های کاربر و ذخیره سازی امتیازات کاربر و دیگر اطلاعات مربوط مربوط به آیتم‌ها، اطلاعات خام سلیقه‌ی کاربران را استخراج می‌کند.

در فاز یادگیری، با توجه به اطلاعات جمع‌آوری شده و روش‌های یادگیری، یک مدل برای نمایش سلیقه‌های هر کاربر ساخته می‌شود. این مرحله در بعضی از روش‌ها انجام نمی‌گیرد و مدلی ساخته نمی‌شود و مستقیم با داده‌ها کار می‌کند (روش تنبل^۱)

در نهایت پیش‌بینی امتیازات انجام می‌شود و با توجه به پیش‌بینی‌های انجام شده توسط مدل، بهترین آیتم به هر کاربر پیشنهاد می‌شود.

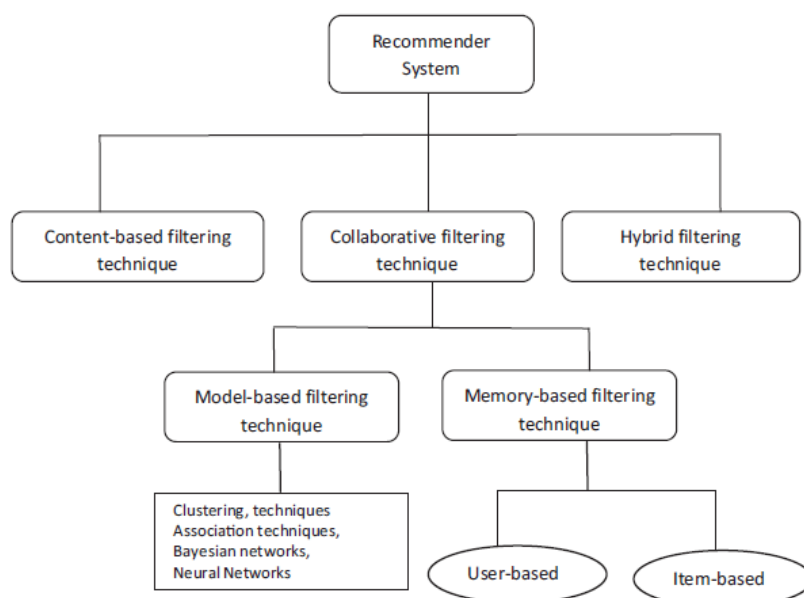
این چرخه پس از ثبت بازخورد کاربر از آیتم پیشنهاد شده ادامه می‌یابد و بهبود پیدا می‌کند.

¹ Lazy learning

۲. روش‌های سیستم‌های پیشنهاد دهنده

در این بخش به مرور روش‌های مرسوم طراحی سیستم‌های پیشنهاد دهنده می‌پردازیم. سیستم‌های پیشنهاد دهنده بصورت کلی به سه دسته تقسیم می‌شوند (شکل ۷):

- فیلتر مبتنی بر محتوا^۱
- فیلتر همکارانه^۲
- ترکیبی^۳



شکل ۷ - روش‌های طراحی سیستم‌های پیشنهاد دهنده

۲.۱ فیلتر مبتنی بر محتوا

این روش بر اساس ویژگی‌های آیتم‌ها یا به عبارتی دیگر، محتوای آن، پیشنهادات خود را انجام می‌دهد. بعد از آنالیز ویژگی‌های آیتم‌های موجود، فیچرهایی از آنها استخراج می‌شود و هر آیتم را با یک بردار فیچر نمایش می‌دهیم.

فیچرها در این روش اکثراً با کمک متای داده برای هر داده تعیین می‌شود، ولی روش‌های دیگر با کمک یادگیری ماشین می‌تواند فیچرهای اصلی هر آیتم را استخراج کند. تا الان روش کاملی برای استخراج اتوماتیک فیچرهای پراهمیت از آیتم‌هایی مثل فیلم ابداع نشده است.

¹ Content-based filtering

² Collaborative filtering

³ Hybrid

روش‌های مبتنی بر محتوا، با دامنه وابسته هستند؛ چون استخراج فیچرها در هر زمینه با دیگری متفاوت است و تخصص خاص خودش را می‌خواهد.

این روش با استخراج فیچرها و مشاهده نمرات کاربران، می‌تواند هر کاربر را نیز با یک بردار نشان دهد که نماینده سلیقه‌ی آن کاربر است. این بردار شامل فیچرهای استخراج شده است. جدول ۱ و ۲ بردار مربوط به یک کاربر و یک فیلم را در یک سیستم فرضی پیشنهاد فیلم با روش مبتنی بر محتوا نشان می‌دهد.

user	action	horror	romance	classic
John doe	0.9	0.8	0.1	0.3

جدول ۱ - بردار نمایش کاربر

movie	action	horror	romance	classic
My story	0.0	0.0	0.9	1.0

جدول ۲ - بردار نمایش فیلم

همانطور که دیده می‌شود کاربر مربوط به جدول یک، از فیلم‌های اکشن و ترسناک خوشش می‌آید و فیلم مربوط به جدول شماره دو عاشقانه و کلاسیک است. پس روش مبتنی بر محتوا می‌تواند با کمک این بردارها و تکنیک‌هایی مثل tf-idf و یا naïve bayes مدلی برای هر کاربر بسازد و اگر احتمال اینکه آن کاربر از یک فیلم خوشش بیاید زیاد بود، آن را به او پیشنهاد بدهد.

مزایای فیلتر مبتنی بر محتوا

- توانایی پیشنهاد آیتم‌هایی که امتیازی ندارند؛ با توجه به اینکه این روش، برای پیشنهاد آیتم‌ها از محتوای آن استفاده می‌کند، بدون دانستن امتیاز آن و فقط با داشتن اطلاعاتی درباره محتوا می‌تواند، آن را پیشنهاد بدهد.
- سازگاری با تغییر سلیقه؛ روش مبتنی بر محتوا به راحتی می‌تواند با در نظر نگرفتن امتیازات ثبت شده قدیمی، خود را با تغییر سلیقه‌ی سریع وفق دهد و بر خلاف روش مبتنی بر مدل نیازی به زمان اضافی ندارد.
- حریم خصوصی کاربران؛ این روش می‌تواند بدون استفاده از اطلاعات شخصی کاربران مانند، زبان و موقعیت، آیتم‌های مشابهی را بدون مشکل پیشنهاد دهد و به این اطلاعات وابسته نیست.

معایب فیلتر مبتنی بر محتوا

- وابستگی به متا؛ این روش نیاز دارد تا متای کاملی از آیتم‌ها جمع‌آوری شود که حتی بعضی از آنها توسط ماشین، قابل برداشت نیست. برای مثال اگر یک بازیگر بعد از

چند سال به یک فیلم بازگشته باشد، معمولا برای طرفداران خوشایند است، ولی روش مبتنی بر محتوا این ریزکاری‌ها را متوجه نمی‌شود.

- نیاز به دانش زمینه‌ای دارد؛ استخراج محتوا علاوه بر هزینه، نیاز به تخصص هم دارد. برای مثال یک متخصص موسیقی باید درباره آهنگ‌های شاد و غمگین بداند و بداند چگونه این موارد را می‌توان از آهنگ استخراج کرد.
- نمی‌تواند آیتم‌های مرتبط ولی غیر مرتبط به پیشنهاد دهد

۲.۲ فیلتر مبتنی بر همکاری

این روش بدون توجه به محتوای آیتم‌ها، با استفاده از امتیازات داده شده توسط کاربران، شباهت آیتم‌ها و کاربران را شناسایی می‌کند. این شباهت‌ها در پیش‌بینی استفاده شده و بر اساس همسایگی کاربران هم سلیقه و یا آیتم‌های شبیه به هم پیش‌بینی از امتیاز صورت خواهد گرفت.

اساس این روش، ماتریس user-item است که نمونه‌ای از آن در جدول ۳ آورده شده است. این روش، با استفاده از این ماتریس، عملیات شناسایی کاربران مشابه یا آیتم‌های مشابه را انجام می‌دهد و با توجه به آن مدل‌سازی انجام می‌گیرد.

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	5		3		1
User 2	4	4		1	1
User 3	2		1	4	4
User 4	1		1	5	

جدول ۳ - نمونه ماتریس user-item

این روش به دو نوع تقسیم می‌شود:

- مبتنی بر مدل (یادگیرنده مشتاق)^۱
- مبتنی بر حافظه (یادگیرنده تنبل)

در روش مبتنی بر مدل، با استفاده از ماتریس user-item یک مدل، استخراج می‌شود که هزینه نگهداری همه‌ی جدول و محاسبات آنلاین کم شود. برای مثال این روش‌ها می‌توانند با خوشه بندی کاربران، کاربران را در دسته‌های مختلف قرار دهند که نماینده سلیق آنها می‌باشد.

در روش مبتنی بر حافظه، که در مسائل بزرگ کاربرد ندارد، کل ماتریس user-item نگهداری می‌شود و همسایگی‌های کاربران یا آیتم‌ها در لحظه محاسبه می‌شوند. این روش تمام جزئیات را نگهداری می‌کند، ولی با توجه به اینکه یک ماتریس m در n (تعداد کاربران و n تعداد آیتم‌ها) را

¹ Eager learner

ذخیره می‌کند، هزینه‌ی نگهداری و محاسبات آنلاین با افزایش کاربران و آیتم‌ها بصورت خطی بالا می‌رود.

مزایای فیلتر همکارانه

- می‌تواند آیتم‌های مرتبط ولی با شباهت کم پیشنهاد دهد؛ مثلاً کیف لپ‌تاپ و لپ‌تاپ
- نیازی به محتوا ندارد؛ مثلاً می‌تواند بدون متا، متوجه شود یک عکس، خنده‌دار است یا جدی. این کار اینگونه انجام می‌شود که پس از مشاهده اینکه یک عکس را، همه‌ی افرادی که عکس خنده‌دار می‌بینند، دوست داشتند، پس حتماً این عکس هم خنده‌دار است.

معایب فیلتر همکارانه

- مشکل شروع سرد و احتیاج به جامعه‌ی فعال و مسئولیت‌پذیر؛ در ابتدای کار آیتم‌ها که نمره‌ای ندارند، عملاً سیستم هیچ پیشنهادی نمی‌تواند بدهد. در بخش ۵ به طور جزئی به این مسئله خواهیم پرداخت.
- مقیاس ناپذیری: همانطور که گفته شد در روش‌های مبتنی بر حافظه، کل ماتریس user-item نگهداری می‌شود که این، این روش را به شدت مقیاس ناپذیر می‌کند. علاوه بر این در روش‌های مبتنی بر مدل نیز برای ساخت مدل، به زمان خیلی زیادی احتیاج داریم که آپدیت کردن مدل را هزینه‌بر می‌کند.
- اعتماد: این روش بر پایه اعتماد به رای کاربران، پیشنهادات خود را انجام می‌دهد. در صورتی که گروه بزرگی از کاربران به یک آیتم حمله کنند و به آن بی‌دلیل رای منفی بدهند، این روش فرض می‌کند این آیتم محتوای بدی دارد و آن را به دیگران پیشنهاد نمی‌دهد.

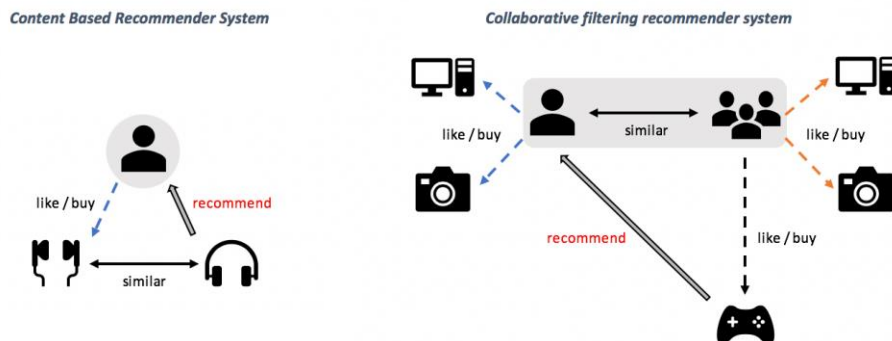
۲.۳ مقایسه

شکل ۸، نحوه‌ی کار دو روشی که با آن اشاره شد را بررسی می‌کند.

در این مثال، همانطور که دیده می‌شود، در روش فیلتر مبتنی بر محتوا، کاربر یک هندزفری سفارش داده است و از آنجایی که این کالا شبیه به هدفون است، سیستم هدفون را نیز به آن کاربر پیشنهاد می‌دهد.

این نکته حائز اهمیت است که سیستم با توجه به اطلاعاتی از کالا در اختیار دارد، می‌تواند کالا‌های شبیه را تشخیص دهد و از پیشنهاد کالا‌های تکراری پرهیز کند. این ویژگی روش فیلتر مبتنی بر

محتوا در مواردی مانند پیشنهاد اخبار مفید است؛ چرا که به کاربر خبری که دیده شده را دوباره نشان نمی‌دهد و تازگی خود را حفظ می‌کند.



شکل ۸ - مقایسه دو روش مبتنی بر محتوا (چپ) و مبتنی بر همکاری (راست)

در روش همکاری‌ها، یک کاربر یک کامپیوتر و یک دوربین خریده است. سیستم با پیدا کردن کاربران مشابه که کامپیوتر و دوربین خریده‌اند، متوجه می‌شود که اکثریت این گروه، دسته بازی نیز خریداری کرده‌اند، بخاطر همین، این دسته را به کاربر مدنظر پیشنهاد می‌دهد.

همانطور که دیده می‌شود مواردی مثل پیشنهاد خرید باتری بعد از خرید موس بی‌سیم را روش همکاری‌ها می‌تواند بدهد. این در حالی است که روش مبتنی بر محتوا، پیشنهاد موس‌های دیگر را می‌دهد (که برای قبل از خرید و انتخاب مناسب است).

۲.۴ روش ترکیبی

همانطور که در بخش ۲.۱ و ۲.۲ ذکر شد، هر دوی این روش‌ها خوبی‌ها و بدی‌های خودشان را دارند. در مسایل واقعی، معمولاً از هر دوی این روش‌ها در کنار هم استفاده می‌شود تا بتوان به حداکثر کارایی دست یافت. مثلاً در مثال آمازون در بخش ۱.۴، دیده شد از هر دوی این روش‌ها برای پیشنهاد استفاده شده است.

۳. مفاهیم فیلتر مبتنی بر همکاری

در این بخش، مفاهیم ابتدایی فیلتر مبتنی بر مدل، از جمله خواسته‌هایی که برآورده می‌کند و ویژگی‌های زمینه‌ای را بررسی می‌کنیم.

۳.۱ خواسته‌های کاربران

- کمک در یافتن آیتم‌های جدید که یک کاربر دوست داشته باشد. در اینترنت با اطلاعات فراوان، کاربران نمی‌توانند همه‌ی آیتم‌ها را بررسی کنند. فیلتر مبتنی بر همکاری با ارائه‌ی چند گزینه‌ی مناسب به کاربران، انتخاب‌های کاربر را محدود می‌کند تا کاربر فقط از میان آنها انتخاب کند.
- مشاوره در مورد یک آیتم خاص
- کمک در یافتن کاربران با سلیقه‌ی مشابه
- کمک به یک گروه برای یک آیتم که همه‌ی آنها دوست داشته باشند. مثلا اگر یک زوج با سلیقه‌های مختلف بخواهند یک فیلم تماشا کنند، فیلتر مبتنی بر همکاری می‌تواند فیلمی را انتخاب کند که هر دو شخص خوششان بیاید.

۳.۲ وظایف سیستم

- پیشنهاد آیتم: فیلتر مبتنی بر همکاری می‌تواند یک لیست از آیتم‌های پیشنهادی برای کاربر نمایش بدهد. این روش معمولا با استفاده از پیش‌بینی امتیاز کاربر به آیتم‌های جدید و سپس رتبه‌بندی بر اساس آن امتیاز صورت می‌گیرد.
- در بعضی از کاربردها، انتخاب چند آیتم مربوط، مهم تر از پیش‌بینی مقدار دقیق است. برای مثال آمازون که کالاهای زیادی دارد، تمام امتیازات را پیش‌بینی نمی‌کند و با کمک روش‌های اکتشافی^۱ و هرس کردن^۲، این کار را سرعت می‌بخشد.
- پیش‌بینی امتیاز: در پیشنهاد فقط کافی است برای هر آیتم، چند گزینه‌ی دیگر برای ارائه داشته باشیم ولی در پیش‌بینی باید آماده‌ی پیش‌بینی هر آیتمی توسط هر کاربر باشیم.
- پیشنهادات مقید: فیلتر مبتنی بر همکاری با توجه به شرایط داده شده، می‌تواند پیشنهادات خود را فیلتر کنند. برای مثال فرض کنید شخصی با دخترش برای دیدن فیلم، پیشنهاد می‌خواهد. شرایط می‌توانند، "نزدیک بودن سینما"، "خشن نبودن فیلم" باشند. علاوه بر این، هر دو فرد باید از این فیلم خوششان بیاید.

¹ Heuristic

² Pruning

۳.۳.۱ توزیع داده

- تعداد آیتم‌ها باید زیاد باشد. اگر آیتم‌ها کم باشند، نیازی به یک سیستم پیشنهاد دهنده نداریم که بر اساس رای کاربران کار کند. مثلاً برای روش مبتنی بر محتوا، با کم بودن تعداد داده‌ها هزینه جمع‌آوری اطلاعات آیتم‌ها کاهش می‌یابد.
- هر آیتم، تعداد امتیاز زیادی دریافت کرده باشد. اگر تعداد امتیازات کم باشد، نتیجه می‌تواند بایاس شود. برای این کار می‌توان یک سیستم داشت که کاربران زیادی داشته باشند و هر کاربر به چند آیتم امتیاز بدهند.

۳.۳.۲ معنی زیرین داده

- هر کاربر در سیستم باید کاربران مشابهی داشته باشد. برای اینکه فیلتر مبتنی بر همکاری به درستی کار کند، باید بتواند کاربران مشابه را شناسایی و استخراج کند و بدون این اطلاعات نمی‌تواند پیشنهادات اختصاصی به کاربران بدهد.
- امتیازات باید وابسته به سلیقه‌ی شخصی باشد. اگر امتیاز آیتم‌ها، بدون نظرات شخصی قابل بدست آوردن باشد، نیازی به دخالت کاربران برای پیشنهاد دادن نیست.
- آیتم‌ها باید همگن باشند. در یک سیستم فیلتر مبتنی بر همکاری، آیتم‌ها فقط باید از نظر شخصی فرق داشته باشند. برای مثال یک کتابفروشی آنلاین، همه‌ی محصولاتش کتاب هستند، قیمت نسبتاً یکسانی دارند و در یک فرمت هستند. تنها چیزی که تفاوت دارد، نظرات مردم نسبت به آنهاست. در یک فروشگاه لوازم خانگی یک تلویزیون ذاتا به یک جاروبرقی فرق دارد و نمی‌توان از فیلتر مبتنی بر همکاری استفاده کرد.

۳.۳.۳ دوام داده

- آیتم‌ها باید دوام داشته باشند. این یعنی که در طول زمان، ارزش آیتم‌ها عوض نشود. برای مثال برای پیشنهاد اخبار، باید اخبار جدید پیشنهاد شود و خبر بعد از یک روز دیگر ارزش قبل را ندارد. در این سیستم‌ها فیلتر مبتنی بر همکاری نمی‌تواند به بهترین شکل عمل کند.
- سلیقه‌ی کاربران نباید به سرعت عوض شود. فیلتر مبتنی بر همکاری برای سازگاری با سلیقه‌ی کاربران به زمان احتیاج دارد. برای مثال، تغییر سریع سلیقه‌ی کاربران، مثلاً در خرید لباس با توجه به مد نمی‌تواند با روش فیلتر مبتنی بر همکاری به پیشنهادهای بهینه دست پیدا کند.

۴. روش‌های فیلتر مبتنی بر همکاری

در این بخش به بررسی روش‌های مختلف فیلتر مبتنی بر همکاری می‌پردازیم. همانطور که در بخش ۲.۲ مطرح شد، روش‌های فیلتر همکارانه مبتنی بر حافظه در مسایل واقعی نمی‌توانند بهیچگی خود را حفظ کنند؛ به همین خاطر در اکثر مسائل واقعی یا از روش مبتنی بر مدل استفاده می‌شود یا از ترکیبی از این دو.

روش‌های مبتنی بر حافظه دو نوع اصلی دارند:

- مبتنی بر کاربر^۱
- مبتنی بر آیتم^۲

۴.۱ روش‌های مبتنی بر کاربر

در این روش، نزدیکترین همسایه‌های هر کاربر از نظر امتیاز داده شده به آیتم‌ها شناسایی می‌شوند و در تخمین امتیاز آن کاربر تاثیر خواهند داشت. این روش مانند نزدیکترین k همسایه عمل می‌کند که نزدیکی را با توجه به امتیازات کاربران اندازه‌گیری می‌کنیم. معیار نزدیکی می‌تواند شباهت کسینوسی یا پیرسون^۳ باشد.

فرمول ساده این روش که میزان نزدیکی را در نظر نمی‌گیرد به شکل زیر است:

$$pred(u, i) = \frac{\sum_{n \in neighbors(u)} r_{ni}}{\text{number of neighbors}}$$

برای دقت بیشتر، می‌توانیم به امتیاز همسایه‌های شناسایی شده با توجه به نزدیکی، وزن بدهیم و آنرا نرمال کنیم:

$$pred(u, i) = \frac{\sum_{n \in neighbors(u)} userSim(u, n) \cdot r_{ni}}{\sum_{n \in neighbors(u)} userSim(u, n)}$$

از آنجایی که مقیاس امتیازدهی کاربران اهمیت دارد، باید برای افزایش دقت، میانگین امتیاز هر کاربر را نیز وارد فرمول کنیم. برای مثال ممکن است یک کاربر خوشبین، به فیلم‌ها امتیاز ۳ تا ۵ بدهد، در صورتی که یک کاربر بدبین امتیاز ۱ تا ۳ بدهد.

$$pred(u, i) = \bar{r}_u + \frac{\sum_{n \in neighbors(u)} userSim(u, n) \cdot (r_{ni} - \bar{r})}{\sum_{n \in neighbors(u)} userSim(u, n)}$$

¹ User-based

² Item-based

³ Pearson correlation

در نهایت برای محاسبه‌ی $userSim()$ می‌توانیم از معیار همبستگی پیرسون کمک بگیریم که به شکل زیر است:

$$userSim(u, n) = \frac{\sum_{i \in CR_{u,n}} (r_{ui} - \bar{r}_u) \cdot (r_{ni} - \bar{r}_n)}{\sqrt{\sum_{i \in CR_{u,n}} (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i \in CR_{u,n}} (r_{ni} - \bar{r}_n)^2}}$$

در فرمول بالا مجموعه CR مجموعه‌ای از آیتم هاست که توسط دو کاربر امتیاز داده شده است.

خروجی فرمول بالا، عددی بین -۱ و ۱ است که به ترتیب، بسیار متفاوت و بسیار شبیه بودن دو کاربر را نشان می‌دهد.

این معیار با مقایسه تمام آیتم‌هایی که توسط هر دو کاربر امتیاز داده شده است، به شباهت آنها دست پیدا می‌کند.

از چالش‌های این روش می‌توان به موارد زیر اشاره کرد:

- جفت کاربرانی که تعداد کمی آیتم دارند که هر دو به آن امتیاز داده‌اند، می‌توانند باعث چولگی همبستگی بشوند و نتیجه را بایاس کنند. مثلاً اگر دو کاربر فقط به یک آیتم مشترک امتیاز داده‌اند، شباهت زیادی دارند و می‌توانند نتیجه را عوض کنند.
- همبستگی پیرسون نمی‌تواند یک توافق کلی را از یک آیتم لحاظ کند. مثلاً امتیاز بالای دو کاربر از یک فیلم که همه آن را پسندیده‌اند کم ارزش تر از امتیاز بالای دو کاربر از یک فیلم بحث برانگیز است. برای این مشکل می‌توان شباهت را در معکوس محبوبیت کلی یک آیتم ضرب کرد. در این صورت اگر دو فرد هر دو به یک فیلم محبوب امتیاز بالایی داده‌اند، این را به حساب محبوب بودن فیلم می‌آوریم، نه شباهت سلیقه‌ی آن دو کاربر.
- محاسبه‌ی شباهت‌های تک تک کاربرها برای ابعاد بزرگ کار بسیار سختی است. برای حل این مشکل از نمونه‌برداری^۱ و خوشه بندی استفاده می‌کنیم. در نمونه برداری، مقایسه فقط با زیرمجموعه‌ای از کاربران صورت می‌گیرد. در خوشه بندی مقایسه هر کاربر با خوشه‌های دیگر صورت می‌گیرد که تعدادشان به مراتب کمتر از کل کاربران است. ساخت خوشه به نوعی توسعه‌ی یک مدل محسوب می‌شود.

۴.۲ روش‌های مبتنی بر آیتم

این روش به نوعی، ترانهاده‌ی روش مبتنی بر کاربر می‌باشد. روش‌های مبتنی بر کاربر بر اساس شباهت بین کاربران، پیش‌بینی و پیشنهادات خود را انجام می‌دهد، در حالی که روش مبتنی بر آیتم، بر اساس شباهت بین آیتم‌ها این وظیفه را به انجام می‌رساند. برای مثال جدول ۴ را در نظر بگیرید. برای

¹ Subsampling

پیش‌بینی آیتم ۲ برای کاربر ۳ (علامت سوال در جدول)، مشاهده می‌شود که امتیازات آیتم ۳ به آیتم ۲ شبیه‌تر اند تا مثلاً آیتم ۱. حالا با کمک یک میانگین وزن دار روی امتیازات کاربر ۳ می‌توان امتیاز کاربر ۳ به آیتم ۲ را پیش‌بینی کرد. این میانگین فقط روی آیتم‌هایی انجام می‌شود که کاربر سه به آنها رای داده است و وزن بر اساس شباهت با آیتم مورد نظر محاسبه می‌شود.

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	3	
User 2	4	5	5	3
User 3	3	?	4	
User 4	5	3	3	4

جدول ۴

برای مثال فرض کنید به آیتم ۱ وزن ۰.۲۵ و به آیتم ۳ وزن ۰.۷۵ بدهیم. در این صورت امتیاز پیش‌بینی شده به شکل زیر بدست می‌آید:

$$0.25 \times 3 + 0.75 \times 4 = 3.75$$

فرمول این روش بصورت زیر می‌باشد:

$$pred(u, i) = \frac{\sum_{j \in ratedItems(u)} itemSim(i, j) \cdot r_{uj}}{\sum_{j \in ratedItems(u)} itemSim(i, j)}$$

در این روش نیازی به نرمال کردن با توجه به میانگین نیست، چون فقط با امتیازات کاربر هدف کار می‌کنیم.

برای محاسبه‌ی شباهت بین آیتم‌ها روش‌های مختلفی وجود دارد که شباهت کسینوسی تعدیل شده^۱ یکی از بهترین‌ها می‌باشد.

$$itemSim(i, j) = \frac{\sum_{u \in RB_{i,j}} (r_{ui} - \bar{r}_u) \cdot (r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in RB_{i,j}} (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_{u \in BR_{i,j}} (r_{uj} - \bar{r}_u)^2}}$$

در فرمول بالا RB مجموعه‌ای از کاربران است که به هر دو آیتم i و j امتیاز داده‌اند.

تنها تفاوت بین همبستگی پیرسون و شباهت کسینوسی تعدیل شده این است که میانگین در فرمول بالا بر اساس کاربران گرفته می‌شود نه آیتم‌ها.

از چالش‌های روش مبتنی بر آیتم می‌توان به موارد زیر اشاره کرد:

¹ Adjusted cosine similarity

- اندازه‌ی مدل می‌تواند به بزرگی مربع تعداد آیتم‌ها باشد. برای کاهش این اندازه می‌توان:
 - فقط همبستگی‌هایی را ذخیره کرد که بیشتر از k کاربر مشترک دارند.
 - با هرس کردن فقط بهترین n همبستگی را نگهداری کنیم.
 که البته هر دوی این روش‌ها انعطاف سیستم برای پیش‌بینی همه‌ی آیتم‌ها را کم می‌کند.
- آیتم‌هایی که کاربران مشترک کمی دارند - مانند روش مبتنی بر کاربر - می‌توانند در همبستگی‌ها چولگی بوجود بیاورند و در یک پیش‌بینی تاثیر بیش از حد داشته باشند.

۴.۳ روش‌های کاهش بعد

همانطور که مطرح شد، اندازه‌ی ماتریس user-item در مسائل واقعی می‌تواند خیلی بزرگ باشد؛ به همین خاطر می‌توانیم برای مدلسازی با کمک روش‌های کاهش بعد مشکل تنک بودن ماتریس user-item را حل کنیم.

بسیاری از الگوریتم‌های کاهش بعد، پیچیدگی دامنه را با استفاده از نگاشت ماتریس user-item به ابعاد کمتر، کم می‌کنند. این ابعاد را می‌توان نماینده‌ی موضوعات مختلف برای آیتم‌ها یا نماینده سلیقه‌های مختلف برای کاربران در نظر گرفت. پس در واقع این روش‌ها امتیاز کاربران را به سلیقه‌ی آنها نگاشت می‌کند.

توابع نگاشت ذکر شده معمولاً با کمک تکنیک‌های برداری ساده، ابعاد زیرین را استخراج می‌کنند. از این تکنیک‌ها می‌توان به تجزیه بردار پشتیبان^۱ و آنالیز اجزای اساسی^۲ اشاره کرد.

از چالش‌های این الگوریتم می‌توان به موارد زیر اشاره کرد

- محاسبات برداری با توجه با اینکه ماتریس user-item خیلی بزرگ است، زمان زیادی می‌طلبد تا انجام شود و ابعاد پنهان داده‌ها را پیدا کند. به همین خاطر در صورتی که بخواهیم تغییرات سلايق کاربران را مدنظر بگیریم باید دوباره این کار را انجام دهیم و به همین خاطر هزینه‌ی تغییر سلیقه‌ی سریع، گران تمام می‌شود.
- روش‌های عملی بروز کردن، ابعاد پنهان بدون تجزیه دوباره کل ماتریس user-item از توابع اکتشافی استفاده می‌کنند و تا حدی می‌توانند این چالش را حل کنند.

¹ Single vector decomposition

² Principle component analysis

- ابعاد خلاصه شده برای انسان‌ها قابل خواندن و فهم نیست و به همین خاطر بهبود الگوریتم، اشکال زدایی^۱ و نگهداری^۲ سیستم هزینه‌ی بسیار زیادی خواهد داشت.

۴.۴ روش‌های استخراج قواعد انجمنی

این روش با توجه به الگوهایی که مکرراً در ماتریس user-item تکرار شده‌اند مدلی می‌سازد تا با کمک آن پیشنهادات خود را انجام دهد. این روش برای سیستم‌هایی که امتیاز عددی ندارند می‌تواند روش مناسبی باشد.

برای مثال، ممکن است در ماتریس user-item مشاهده شود، کسانی که به آیتم ۱ امتیاز بالایی داده‌اند اکثراً به آیتم ۲ هم امتیاز بالایی داده‌اند. با توجه به این الگو، اگر کسی به آیتم ۱ امتیاز بالایی بدهد، می‌توان آیتم ۲ را نیز به او پیشنهاد کرد.

Rule r : Liking item 1 → Liking item 2

$$Support(r) = \frac{\# \text{ users who liked item 1 and item 2}}{\text{total \# users}}$$

$$Confidence(r) = \frac{\# \text{ users who liked item 1}}{\# \text{ users who like both item 1 and item 2}}$$

این روش با چالش‌های زیر روبرو می‌شود:

- قواعد انجمنی ساده، برای امتیازهای عددی، هر امتیاز را جداگانه و مستقل از هم در نظر می‌گیرد. به این معنی که اگر کسی از ۵ به یک آیتم امتیاز ۲ داده است و به آیتم دیگری امتیاز ۱ داده است، قواعد انجمنی این شباهت را در نظر نمی‌گیرد. برای حل این چالش راه حل‌های زیر پیشنهاد شده‌اند:
 - تقسیم امتیازات به دو دسته بالا و پایین: در این روش، امتیازات ب توجه به میانگین امتیازات هر کاربر به دو دسته‌ی بالای میانگین به معنی اینکه کاربر از آن آیتم خوشش آمده و پایین میانگین تقسیم می‌کنیم.
 - در نظر گرفتن امتیازات بالا: در این روش برای ساختن قواعد انجمنی، فقط قواعدی ساخته می‌شود که شامل آیتم‌هایی است که کاربر به آنها امتیاز بالا داده است.
 - مساوی در نظر گرفتن همه‌ی امتیازات: در این روش امتیاز ۱ با ۵ فرقی ندارد و صرف امتیاز دادن اهمیت دادن.

¹ Debugging

² Maintenance

روش‌های بالا با گسسته کردن امتیازات به دسته‌های کوچکتر، عملاً روابط عددی بین امتیازات را از قواعد خارج می‌کند.

- در مسایل واقعی، استخراج قواعد از یک ماتریس بسیار بزرگ، خیلی زمانبر است.

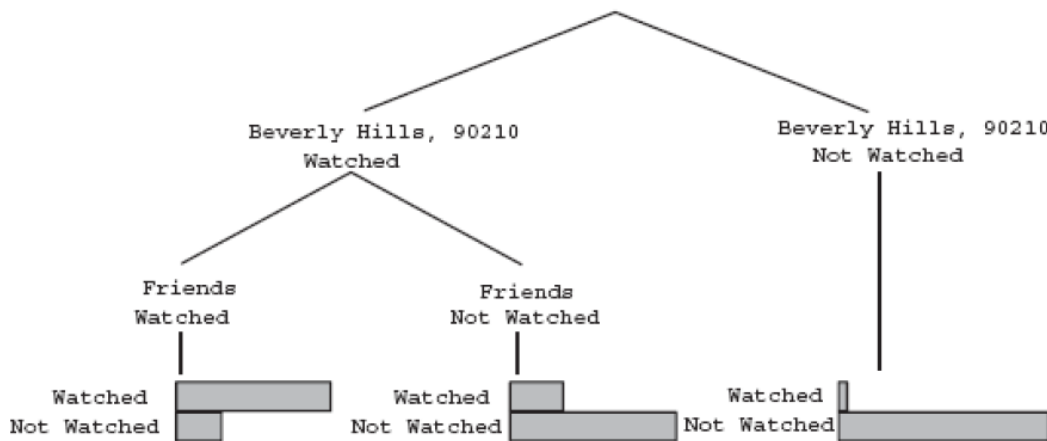
۴.۵ روش‌های احتمالی

الگوریتم‌های احتمالی به شکل صریح از توزیع‌های احتمالاتی هنگام محاسبات تخمین امتیاز یا پیشنهاد آیتم، استفاده می‌کند. این روش‌ها با محاسبه‌ی احتمال امتیاز یک کاربر به یک آیتم، یا امید ریاضی امتیاز آن کاربر به آن آیتم را محاسبه می‌کند و یا به سادگی امتیاز با احتمال بالاتر را مدنظر می‌گیرد.

$$E(r|u, i) = \sum_r r.p(r|u, i)$$

یکی از معروف‌ترین چارچوب‌های احتمالاتی، شبکه‌ی بیز^۱ است که وابستگی‌های احتمالاتی را بین کاربران و آیتم‌ها، بدست می‌آورد.

روش شکل ۹، درختی دیده می‌شود که با استفاده از مدل‌های احتمالاتی استخراج شده است. در این روش برای هر آیتم، یک درخت ساخته می‌شود و با توجه با آن، احتمال رفتارهای کاربر پیش‌بینی می‌شود.



شکل ۹ - مدل احتمالاتی یک آیتم نمایش داده شده توسط یک درخت تصمیم

¹ Bayesian network

شکل ۹، درخت مربوط به احتمال دیده شدن فیلم "Melrose Place" است. برای مثال در شاخه‌ی سمت راست، نشان داده شده که اگر فیلم "Beverly Hills, ۹۰۲۱۰" را ندیده باشد با احتمال خیلی قوی فیلم "Melrose Place" را هم ندیده است.

علاوه بر این روش‌های احتمالاتی دیگری برای خوشه‌بندی و کاهش بعد هم توسعه داده شده است. در روش‌های کاهش بعد احتمالاتی یک متغیر پنهان $p(z|u)$ معرفی می‌شود که نشان دهنده احتمال تعلق کاربر u به کلاس پنهان z است. فرمول زیر، با این روش، احتمال امتیاز یک کاربر به یک آیتم را بدست می‌آورد.

$$p(r|u, i) = \sum_z p(r|i, z).p(z|u)$$

با جاگذاری فرمول بالا در فرمول امید ریاضی:

$$E(r|u, i) = \sum_r r. \sum_z p(r|i, z)p(z|u)$$

روش‌های بیشینه‌سازی امید ریاضی^۱ معرفی شده است که می‌تواند کلاس پنهان z را با کمک توزیع گاوسی بدست بیاورد.

یکی از مزایای الگوریتم‌های احتمالاتی این است که می‌توانند سطح اطمینان خود از پیش‌بینی انجام شده را - علاوه بر پیش‌بینی امتیازات - مدنظر قرار دهند.

¹ Expectation maximization

۵. چالش‌های فیلتر مبتنی بر همکاری

۵.۱ تعداد کم امتیازات

همانطور که در بخش ۴.۱ و ۴.۲ مطرح شد، تعداد کم اشتراکات و به طور کلی نمرات داده شده، می‌تواند در نتیجه تاثیر به سزایی داشته باشد و پیش‌بینی‌ها را بایاس کند. تکنیک‌های زیر برای حل این مشکلات معرفی شده‌اند:

۵.۱.۱ صرف نظر کردن از آیتم‌ها با تعداد کم امتیاز

الگوریتم‌ها در حالت کلی، فقط داده‌هایی که بیش از k امتیاز داشته باشند را در محاسبات دخیل می‌کنند. این روش می‌تواند باعث کاهش پوشش^۱ شود که در بخش ۶ معرفی خواهد شد.

۵.۱.۲ تعدیل محاسبات برای آیتم‌های با تعداد امتیاز کم

این روش با نزدیک کردن امتیاز آیتم‌های به ندرت مشاهده شده به میانگین، تاثیر بایاس را کاهش می‌دهد. برای مثال همبستگی پیرسون برای کاربران با تعداد کم آیتم مشترک، به صفر میل می‌کند. گرچه این روش می‌تواند مفید باشد ولی تنظیم پارامترهای تعدیل می‌توان سخت و ناپایدار باشد.

۵.۱.۳ استفاده از یک باور پیشین^۲

با کمک تخصیص زمینه‌ای و تجربه از داده‌های پیشین، می‌توان با استفاده از داده‌های مصنوعی، توزیع امتیازات را به سمت یک توزیع واقعی سوق داد.

برای مثال با فرض دانستن توزیع امتیازات هر کاربر، می‌توان با ایجاد چند آیتم مصنوعی در داده‌ها، توزیع امتیازات کاربران را به سمت توزیع واقعی، سوق دهیم. همچنین این کار را می‌توان با توزیع امتیازات هر آیتم و اضافه کردن کاربران مصنوعی انجام داد.

۵.۲ پیشنهاد یا پیش بینی

برای پیشنهاد یک آیتم، یک سیستم کافی است اطلاعات زیرمجموعه‌ای از کل آیتم‌ها داشته باشد، ولی احتمالاً نه همه‌ی آیتم‌ها. برای این کار احتیاجی به ذخیره‌ی همه‌ی داده‌ها نیست. برای

¹ Coverage

² Prior belief

پیش‌بینی یک آیت‌خاص، سیستم باید اطلاعات همه‌ی آیت‌ها را ذخیره کند تا در صورتی که بخواهد یک آیت‌خاص که تعداد امتیازات کمی را دارد پیش‌بینی کند، چیزی برای گفتن داشته باشد.

از طرفی برای پیشنهاد دادن، نیاز به پیش‌بینی امتیاز حداقل زیرمجموعه‌ای از آیت‌ها داریم. بنابراین پیش‌بینی یک آیت‌خاص، هزینه‌ی بیشتری از پیشنهاد آیت‌خاص به یک کاربر دارد.

۵.۳ معیارهای اطمینان

برای نمایش آیت‌های پیشنهاد شده، می‌توان از دو معیار استفاده کرد. اولی پیش‌بینی امتیاز کاربر به آن آیت‌خاص است و دوم اطمینان از صحیح بودن آن پیش‌بینی. این نیاز به سبک و سنگین کردن دارد که چقدر به هر کدامیک از این معیارها اهمیت بدهیم. یک سیستم ممکن است فقط پیشنهادات را بر اساس پیش‌بینی امتیاز مرتب‌سازی کند و به سطح اطمینان پیش‌بینی‌ها اهمیتی ندهد.

برای انتخاب معیار اطمینان، باید با توجه به روش انتخاب شده عمل کرد. برای مثال اطمینان در روش مبتنی بر کاربر به امتیاز همسایه‌های آن کاربر بستگی دارد در حالی که در روش مبتنی بر آیت‌خاص، می‌تواند تعداد جفت آیت‌های شبیه را مدنظر قرار دهد.

۵.۴ نمرات صریح و ضمنی

جمع‌آوری امتیازات برای سیستم‌های پیشنهاد دهنده به دو صورت کلی صورت می‌گیرد، صریح و ضمنی.

در روش صریح، به طور مستقیم از کاربر خواسته می‌شود به یک آیت‌خاص که آن را می‌شناسد (مثلاً فیلمی که تماشا کرده) نمره بدهد. این در حالی است که روش‌های ضمنی، فقط فعالیت‌های کاربر را مشاهده می‌کنند و نیازی به امتیازدهی ندارند. برای مثال مبنای کار پیشنهاد دهنده اسپاتیفای که در بخش ۱.۴.۲ معرفی شد، بیشتر روش ضمنی است.

روش‌های صریح از طرف کاربر نیاز به تلاش دارد و در عوض، معیار دقیقی برای سلیقه‌ی کاربر است. روش ضمنی تقریباً هیچ زمانی از کاربر نمی‌گیرد ولی ممکن است دقیق نباشد. برای مثال زمان صرف شده در یک صفحه به عنوان یک معیار امتیاز برای آیت‌خاص موجود در آن صفحه را در نظر بگیرید. ممکن است یک کاربر صفحه‌ای که دوست نداشته را باز کرده باشد و برای استراحت، در حال قهوه خوردن باشد و از سیستم استفاده‌ای نکند. نتفلیکس که در بخش ۱.۴.۱ به آن اشاره شد، در صورتی که کاربر بین تماشای فیلم فعالیت نداشته باشد، پیامی نمایش می‌دهد تا مطمئن شود کاربر در حال تماشای فیلم است و برای مثال خواب نیست.

با روش صریح به تنهایی، مشکل تعداد امتیازات کم، بسیار بیشتر اتفاق می‌افتد و بخاطر همین، معمولاً از روش‌های ترکیبی استفاده می‌شود.

هر چه تعداد امتیازات جمع‌آوری شده بیشتر باشد، پیش‌بینی‌ها دقیق‌تر خواهند بود و سیستم می‌تواند عدم قطعیت بیشتری را مدیریت کند.

یکی از مزایای استفاده از روش ضمنی نسبت به صریح این است که روش صریح نمی‌تواند بین آیتم‌هایی که امتیاز مساوی دارند، تمایزی قائل شود. روش ضمنی با در نظر گرفتن معیارهای جزئی‌تر می‌تواند (هر چند با خطا) ریزکاری‌های سلیقه‌ی کاربران را در نظر بگیرد. برای مثال اگر یک سیستم پیشنهاد موسیقی امتیاز بین ۱ تا ۵ دارد و کاربر دو موسیقی را انتخاب کند و به یکی ۴ بدهد و به دیگری ۴.۵. از آنجایی که سیستم دقت ۰.۵ ندارد، کاربر مجبور است به این موسیقی هم ۴ بدهد و به این صورت روش صریح نمی‌تواند تفاوت این دو را متوجه شود. در صورتی که روش ضمنی با مانیتور کردن فعالیت‌های کاربر متوجه می‌شود که این کاربر به موسیقی دوم، بیشتر گوش می‌دهد.

یکی از چالش‌های سیستم‌های پیشنهاد دهنده، جمع‌آوری امتیازات صریح است. این امتیازات به توجه کاربر نیاز دارند و خیلی از کاربران برای امتیاز دادن وقتی صرف نمی‌کنند و امتیازات غیر دقیق می‌دهند.

یک سیستم فیلتر همکارانه برای موفقیت فقط نیاز دارد در ابتدا چند کاربر فعال داشته باشد که به هر آیتی می‌بینند، رای بدهند و این کار را مکرر و در طولانی مدت انجام دهند.

به علاوه، کاربران با احساس رضایت نسبت به امتیاز دادن می‌توانند باعث ارتقای سیستم بشوند. از این احساسات رضایت می‌توان به موارد زیر اشاره کرد:

- احساس مشارکت در پیشرفت یک اجتماع
- رضایت از با ارزش بودن نظرات خود

راه دیگر برای تشویق کاربران به امتیاز دادن، استفاده از محرک‌هایی مثل امتیاز دادن به کاربران فعال است که بتوانند با آن جایزه بگیرند.

۵.۵ معیار های نمرات

در قسمت ۵.۴ به مشکل درشت‌دانه بودن معیار امتیاز اشاره شد. بصورت کلی هر چه امتیازات ریزدانه‌تر باشند، اطلاعات بیشتری از امتیازات حاصل می‌شود. این ریزدانه بودن می‌تواند برای طراح-های رابط کاربری مشکل ایجاد کند و پیچیدگی رابط کاربری را افزایش دهد.

ریزدانه شدن امتیازات بیشتر از یک حد می‌تواند باعث شود که دیگر ارزش بیشتری به امتیازات اضافه نشود و عدم قطعیت را بالا می‌برد. برای مثال در یک سیستم که باید بین ۱ تا ۱۰۰ به یک فیلم امتیاز بدهیم، اکثراً اگر یک کاربر یک فیلم را دو بار ببیند به آن فیلم امتیازات متفاوتی می‌دهد و بعید است هر دو بار یک امتیاز ثابت به آن بدهد.

در جدول ۵ انواع معیارهای امتیاز دهی نمایش داده شده است.

Rating scale	Description	Example
Unary	Good or “don’t know”	Spotify, Instagram
Binary	Good or bad	YouTube, Reddit
Scalar	Star, 1-10 or 1-5	Amazon, Imdb
Ordinal	Very good, good, bad, very bad	Surveys

جدول ۵ - مقیاس های امتیازدهی صریح

۵.۶ مشکل شروع سرد^۱

شروع سرد به وضعیتی گفته می‌شود که در آن سیستم پیشنهاد دهنده بخاطر نبود امتیازات اولیه قادر به پیش‌بینی و پیشنهاد نیست. این مشکل بصورت کلی در سه سناریو اتفاق می‌افتد:

- ورود یک کاربر جدید: یک کاربر بعد از ثبت نام و ورود به سیستم، هیچ اطلاعاتی از امتیازات به سیستم نمی‌دهند و سیستم نمی‌تواند سلیقه‌ی او را تشخیص بدهد. برای حل این مشکل راه حل‌های زیر پیشنهاد شده‌اند:
 - به کاربران تازه وارد چند آیتم نشان بدهیم و بخواهیم قبل از ورود به سیستم برای بار اول، به آن‌ها امتیاز بدهند.
 - نمایش پیشنهاد‌های غیر شخصی مثل میانگین کل جمعیت تا وقتی که امتیازات کافی جمع‌آوری شده باشد.
 - از کاربر خواسته شود تا سلیقه‌ی خود را مطرح کند. برای مثال اینکه از چه ژانرهای موسیقی خوشش بیاید.
 - اضافه شدن یک آیتم جدید: در صورتی که سیستم آیتم‌های مرغوب و دیده نشده داشته باشد می‌توان از روش‌های زیر استفاده کرد:
 - آیتم‌های جدید را با استفاده از روش‌های غیر همکارانه مثل مبتنی بر محتوا پیشنهاد دهیم تا تعداد امتیازاتش زیاد شود.
 - انتخاب تصادفی آیتم‌های جدید و نمایش آنها به کاربران تا کاربران به آن‌ها رای بدهند.
- تتفلیکس برای مثال در صفحه‌ی اول خود (شکل ۴) یک بخش انتشارات جدید دارد و با این روش، این مشکل را حل می‌کند.

¹ Cold start issue

- اجتماع جدید: در صورتی که یک شبکه یا سایت جدید تشکیل داده باشیم، ماتریس user-item به شرط داشتن کاربر و آیتم، خالی است. برای حل این سناریو، روش‌های زیر پیشنهاد شده‌اند:

- برای تعداد کمی از کاربران اولیه، مشوق‌هایی در نظر بگیریم تا بیشتر امتیاز بدهند و کار سیستم پیشنهاد دهنده را راه بیاندازند.
- تا وقتی ماتریس user-item پر شود، با کمک روش‌های غیر همکارانه پیشنهاد بدهیم تا کاربران از چرخه خارج نشوند.
- با داده‌های آماده از دیگر منابع در این زمینه استفاده کنیم.

۶. ارزیابی

۶.۱ دقت

معیار دقت می‌تواند اختلاف بین پیش‌بینی انجام شده و امتیاز واقعی باشد یا اختلاف بین رتبه‌ی آیتم پیشنهاد شده و ترتیب کلیک شدن روی آن.

ساده‌ترین معیار دقت، میانگین خطای مطلق^۱ است که میانگین قدرمطلق اختلاف امتیاز پیش‌بینی شده و امتیاز داده شده توسط کاربر (در صورت امتیاز دادن) می‌باشد. این بین آیتم‌هایی که در صدر لیست پیشنهادها هستند و در پایین این جدول هستند تمایز قائل نمی‌شود.

روش نیمه‌جان^۲، یک معیار سودمندی است که اشتباهات صدر جدول، وزن نمایی بیشتری نسبت به اشتباهات انتهای جدول پیشنهادات دارند.

۶.۲ روش‌های دیگر

۶.۲.۱ تازگی

این معیار با هدف پیشنهاد آیتم‌هایی مطرح شده که کاربر از آن اطلاعی ندارد. برای مثال پیشنهاد ندادن یک خبر که کاربر آن را قبلاً دیده، تازگی محسوب می‌شود.

۶.۲.۲ سرنوشت‌سازی^۳

^۱ Mean absolute error

^۲ Half-life

^۳ Serendipity

این معیار آیتم‌های جدید و نوآورانه‌ای را پیشنهاد می‌دهد که شباهتی با آیتم‌های امتیاز داده شده توسط کاربر ندارند. برای مثال اگر کاربری اخبار سیاسی دوست دارد به احتمال زیاد اخبار اقتصادی هم دوست دارد.

۶.۲.۳ پوشش

این معیار نشان‌دهنده درصدی از آیتم‌ها است که سیستم پیشنهاد دهنده می‌تواند پیشنهاد دهد. هر چه این مقدار بیشتر باشد، به احتمال زیاد false positive ها بیشتر خواهند شد و به این ترتیب precision کاهش می‌یابد.

۷. جمع بندی

روش‌های مبتنی بر محتوا در مسائل محدودی می‌توانند پیش‌بینی‌های بسیار خوب و مفیدی داشته باشند. متأسفانه هنوز روش ابداع نشده که بتواند بصورت اتوماتیک تمام اطلاعات مربوط به آیتم‌های مثل فیلم را استخراج کند و سلیقه‌ی کاربران را مدل کند.

به همین خاطر روش فیلتر همکارانه با وارد کردن کاربران به چرخه‌ی سیستم‌های پیشنهاد دهنده، بدون نیاز به محتوای آیتم‌های می‌تواند پیش‌بینی‌های خوبی در اکثر زمینه‌ها داشته باشد.

با استفاده از هر دو روش فیلتر همکارانه و فیلتر مبتنی بر محتوا، می‌توانیم در اکثر زمینه‌ها به پیش‌بینی‌های بسیار خوبی دسترسی پیدا کنیم.

Schafer, Ben & J, Ben & Frankowski, Dan & Dan, & Herlocker, & Jon, & Shilad, & Sen, Shilad. (2007). Collaborative Filtering Recommender Systems.

F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh(2015), Recommendation systems:Principles, methods and evaluation