امیرحسین کارگران خوزانی -- شماره دانشجویی: ۹۹۲۰۱۱۱۹

1. Content based recommendation:

یکی از روش‌های طراحی سیستم‌های توصیه‌گر می‌باشد. این روش بر اساس توصیف آیتم و ترجیحات پروفایل کاربر عمل می‌کند. در واقع این filtering خود به یک مدل دسته بند تقسیم می‌ شود که بر اساس لایک‌ها و دیس‌لایک‌های کاربران و ویژگی‌ آیتم‌ها یاد می‌گیرد که به هر کاربر چه چیزی را پیشنهاد کند. از تاریخچه تعاملات کاربر با سیستم و ترجیحاتی که مستقیما از کاربر پرسیده می‌شود می‌توان پروفایل کاربر را ساخت. یکی از پر استفاده‌ترین الگوریتم‌ها در این زمینه tf-idf می‌باشد. سیستم براساس ویژگی‌ آیتم‌ها یک پروفایل برای هر کاربر می‌سازد که در آن هر ویژگی یک عدد وزن دار است و نشان می‌دهد آن کاربر چقدر به آن ویژگی علاقه دارد. یکی از مشکلاتی که برای این توصیه‌گر وجود دارد محدودیت آن در توصیه‌گری تنها برای یک content است و الزاما نمی‌توان از یک content برای پیشنهاد دادن بقیه content ها استفاده کرد، به همین منظور روش‌های hybrid پیشنهاد می‌شوند.

مثال: برای کاربر می‌توان بر اساس انواع ژانر‌های موسیقی و انواع آرتیست‌ها پروفایل ساخت که مشخص کند هر کاربر چقدر به هر یک از این فیچرها علاقه دارد. از آن طرف هر موسیقی را می‌توان بر اساس آرتیست و ژانر آن توصیف کرد، حال می‌توان به کاربر موسیقی‌های جدیدی را با توجه به پروفایل کاربر پیشنهاد داد.

1. Collaborative filtering:

یکی از روش‌های طراحی سیستم‌های توصیه‌گری که به صورت گسترده استفاده می‌شود، می‌باشد. در این روش یک فرض وجود دارد و آن این است که کاربران همانند گذشته رفتار می‌کنند.

این روش توصیه‌گر بر اساس اطلاعات rate هایی که کاربران مختلف به item های مختلف دادند عمل می‌کند که خود دو روش مبتنی بر مدل و مبتنی بر حافظه است. در روش مبتنی بر مدل با فرض یک مدل generative سعی در توجیه روابط میان کاربران و آیتم‌ها می‌شود. یکی از روش‌های مبتنی بر حافظه معروف الگوریتم‌های بر اساس کاربر هستند. در این الگوریتم‌ها سعی می‌شود که مشابه‌ترین کاربرها به کاربری که میخواهیم به آن آیتم پیشنهاد دهیم پیدا شود و محبوب‌ترین آیتم‌های آن کاربران را به کاربر مورد نظر پیشنهاد کنیم. در این صورت می‌توان یک ماتریس crosstab برای کاربرها و آیتم‌ها و rating آن‌ها تشکیل داد و از روی آن شباهت بین دو کاربر را با یک معیار شباهت پیدا کرد.

مثال: تعداد کاربران زیادی به تعداد فیلم‌های زیادی در گذشته rate داده‌اند، حال برای پیشنهاد دادن به یک کاربر نیاز است که ببینیم آن کاربر چقدر به بقیه کاربرهایی که فیلم‌هایی مانند او rate داده‌است شبیه است و فیلم‌های محبوب این دسته کاربران را به او معرفی کنیم.

1. Nearest neighbours:

K Nearest Neighbors خود یک روش یادگیری ماشین تحت نظارت برای دسته بندی است که نقاط را با استفاده از نزدیکی‌ آن‌ها به بقیه نقاط دسته‌بندی می‌کند. این روش می‌تواند همگام با روش collaborative filtering به طراحی سیستم توصیه‌گر کمک کند. بدین ترتیب که به هر کاربر آیتم‌های خریداری شده همسایگان نزدیک آن کاربر را پیشنهاد می‌دهیم. یا آیتم‌هایی که نزدیک به آیتم خریداری شده توسط آن کاربر است را پیشنهاد می‌دهیم. که این همسایگی را می‌توان با استفاده از ماتریس crosstab بیان شده در قسمت قبل مدل کرد.

1. Latent factor methods:

latent matrix factorization یکی از روش‌های قوی در ساخت سیستم‌های توصیه‌گر است. به خصوص که زمانی که در مسابقه Netflix نشان داد که از روش‌های دیگر عملکرد بهتری دارد بیشتر از قبل مطرح شد.

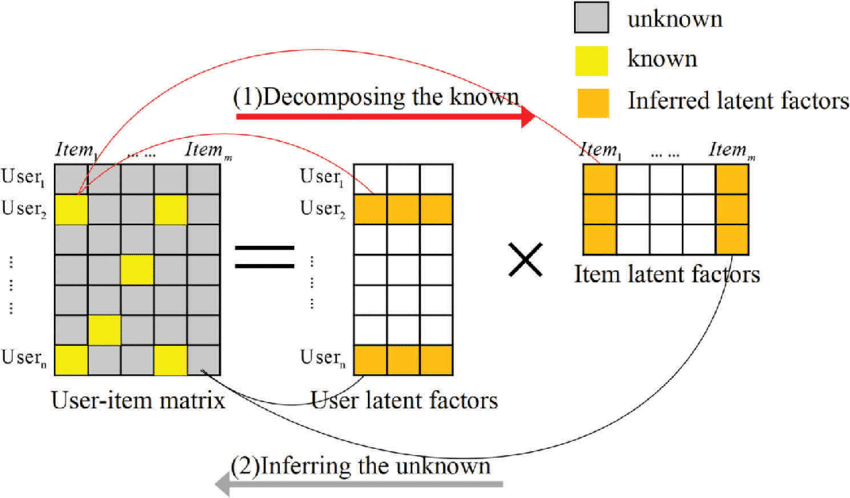
در این روش سعی می‌شود که به ماتریس crosstab بیان شده که شامل آیتم و کاربرهاست نگاه شود ولی همزمان از تاثیر هر دو کاربر و آیتم استفاده شود و آن‌ها را به یک فضای یکسان نگاشت کند. در ادامه در قسمت matrix factorization بیشتر توضیح داده شده است.

1. Matrix factorization:

یک روش از کلاس روش‌های collaborative filtering می‌باشد. که با decompose کردن ماتریس crosstab کاربر-آیتم بحث شده کار می‌کند و آن را به صورت ضرب دو ماتریس مستطیلی با دایمنشن کمتر بدل می‌کند. همان گونه که در قبل هم گفته شد خانواده این روش در مسابقه Netflix عملکرد مناسبی از خود نشان دادند. جواب‌های پیش‌بینی می‌توانند با استفاده از اضافه کردن وزن‌های منظم‌ساز به latent factor ها بر اساس مشهور بودن آیتم‌ها و میزان فعال بودن کاربر بهبود یابد. یکی از مشهوریترین این روش‌ها روش‌های decomposition بر اساس SVD است. شکل زیر شهود مناسبی از این decomposition می‌دهد. ([منبع شکل](http://dx.doi.org/10.1080/13658816.2017.1400550))

دو ماتریس decompose شده یکی ماتریس نهفته کاربران و دیگری ماتریس نهفته آیتم‌هاست.

هر سطر i ام در ماتریس نهفته کاربران نشان دهنده کاربر i و هر سطر i ام در ماتریس نهفته آیتم‌ها نشان دهنده آیتم i در ماتریس اصلی است.

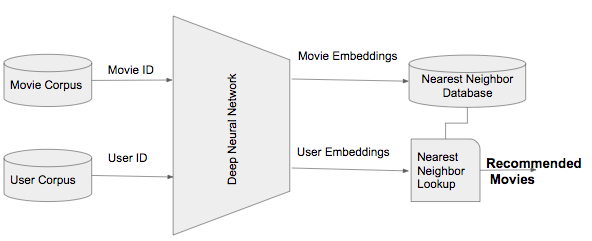


1. Deep learning embedding:

با استفاده از deep learning ها می‌توان به صورت خودکار روابط بین موجودیت‌ها را درک کرد و از روی آن‌ها به طور خودکار یک سری embedding استخراج کرد. و از روی embdeing ها که مانند یک سری ویژگی هستند آیتم‌های شبیه به یکدیگر و کاربران شبیه به یکدیگر را استخراج کرد و به کاربران آیتم‌ها را پیشنهاد داد.

می‌توان از روش Nearest Neighbors برای پیدا کردن و ذخیره آیتم‌ها و کاربران شبیه به یکدیگر اقدام کرد.

از آنجا که به صورت خودکار این ويژگی‌ها ساخته می‌شود و ابعاد بردارهای خروجی آن نیز ثابت است، این روش scalable است. در شکل زیر یک نما از معماری توصیه‌گر برای پیشنهاد فیلم توسط این روش نمایش داده شده است. ([منبع شکل](https://towardsdatascience.com/recommender-systems-from-learned-embeddings-f1d12288f278))



1. Hybrid approaches:

به منظور استفاده از قابلیت‌های هر یک از روش‌ها و غلبه بر مشکلات آن‌ها و استفاده از ويژگی‌های منحصر به فرد آن‌ها می‌توان دو یا چند روش را با یکدیگر ترکیب کرد. فرضا همانگونه که قبلا گفته شد می‌توان برای غلبه بر مشکلات روش‌های مبتنی بر content از روش hybrid استفاده کرد که در یک حالت خاص برای مثال این ترکیب می‌تواند با استفاده از روش collaborative filtering باشد. Netflix یک مثال خوب از نمونه استفاده شده از چنین توصیه‌گرهایی است. تکنیک‌های مختلفی برای انجام عملیات hybrid وجود دارد که می‌توان در ذیل آن‌ها را معرفی کرد:

* Weighted: ترکیب عددی امتیاز دو توصیه‌گر
* Switching: انتخاب بین مولفه‌های توصیه‌گر و اعمال کردن مولفه انتخاب شده
* Mixed: ارائه توصیه‌های متفاوت از توصیه‌گرهای متفاوت
* Feature Combination: ویژگی‌های متفاوت از منابع مختلف با یکدیگر ترکیب می‌شوند و یک توصیه‌گر واحد داده می‌شود.
* Feature Augmentation: محاسبه یک ویژگی یا مجموعه‌ای از ویژگی‌ها که به تکنیک بعدی داده می‌شود. (مثلا اگر کل یک عکس را ویژگی در نظر بگیریم، rotate شده آن یک augmented ویژگی است)
* Cascade: Cascade: به توصیه گران اولویت سختگیرانه داده می شود.
* Meta-level: یک تکنیک توصیه‌گر اعمال می‌شه و خروجی آن به توصیه‌گر بعدی داده میشه.

1. در این [مقاله](https://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf) الگوریتم مسابقه Netflix بیان شده است، از آنجا که این الگوریتم پیچیده بود با وجود جوابی که داد هیچ وقت در Netflix پیاده نشد.

در صورتی که برای مثال مقام سوم این مسابقه ([در اینجا بخوانید](https://sifter.org/simon/journal/20061211.html)) الگوریتم ساده‌تری را در مسابقه استفاده کرده بودند.

برندگان این مسابقه الگوریتم پالایش گروهی را ساخته بودند که می‌توانست سوگیری‌های کاربر و فیلم را بهتر از بقیه مدل‌های موجود، ثبت کند. بنابراین، اگر یک فیلم مشخصی دارای امتیازی بالاتر از میانگین باشد، به تنهایی نمی‌تواند به طور کامل بر الگوریتم توصیه‌گر غالب شود. استفاده از تکنیک‌ قدرتمند Matrix Factorization همراه با neighborhood یکی از دلایل موفقیت این الگوریتم بود و در همین جا نیز Latent factor methods ها بیشتر از پیش مهم شدند.