

1. Cold Start:

به طور معمول یک سیستم توصیه گر از پروفایل کاربر و مشخصه‌های آن برای پیشنهاد دادن آیت‌ها استفاده می‌کند که این مشخصه‌ها می‌تواند متناسب مشخصه‌های آیت‌ها باشد. مشخصه‌های پروفایل کاربر ممکن است از محیط استفاده کاربر و یا رفتار قبلی او بدست آمده باشد.

۳ مورد وجود دارد که ممکن است موجب ایجاد cold start شود:

- جامعه جدید: اشاره به شروع یک سیستم توصیه گر از نو دارد. زمانی که اگر چه ممکن است تعدادی آیت‌ها وجود داشته باشد اما هیچ کاربری وجود ندارد و عدم وجود تعاملات کاربران باعث می‌شود که تولید پیشنهادها قابل تکیه خیلی سخت باشد. این مشکل هم شامل مشکل آیت‌ها جدید و هم کاربر جدید است که در ادامه بیان می‌شود.
- آیت‌ها جدید: یک آیت‌ها جدید به سیستم اضافه می‌شود که ممکن است مقداری اطلاعات content برای آن وجود داشته باشد اما هیچ تعاملی با آن وجود ندارد.
- کاربر جدید: یک کاربر جدید ثبت نام می‌کند، و هنوز تعاملی انجام نداده است، پس نمی‌توان به آن پیشنهاد شخصی‌سازی شده داد.

راه کارهای مقابله:

روش Hybrid:

Collaborative filtering یکی از بیشترین الگوریتم‌های آسیب‌پذیر نسبت به این مشکلات است چرا که نیازمند تشکیل ماتریس کاربر-آیت‌ها است. برای جلوگیری از این مشکل می‌توان این روش را با روش Content based ترکیب کرد.

تکمیل پروفایل:

همچنین در مورد کاربران جدید باید گفت که می‌توان به جای استفاده از تعاملات کاربران که اکنون وجود ندارد از محیط کاربر استفاده کرد که این خود شامل زبان یا موقعیت جغرافیایی آن کاربر می‌باشد. راهکار اصولی‌تر در این موارد آن است که از کاربر مستقیماً در ابتدا ترجیحات او پرسیده شود و سعی در کامل‌سازی پروفایل کاربر کنیم.

Feature mapping:

استفاده از روش‌هایی مانند matrix factorization می‌تواند کمک کند. اگر آیت‌ها جدیدی اضافه شد می‌توان بر اساس مشخصات آن (سال، نویسنده، طبقه، و ..) و با توجه به آیت‌های قدیمی، latent factor مناسب را تولید کرد. همچنین برای کاربر جدید نیز همین‌طور است و بر اساس مشخصات آن (ملیت، جنسیت، سن و ...) می‌توان latent factor مناسب تولید کرد.

2. Exploitation:

Exploitation که در مقابل آن exploration قرار می‌گیرد به عنوان یک تریدآف در بسیاری از سیستم‌های مطرح می‌شود.

در این تریدآف باید سیستم تصمیم بگیرد که چقدر می‌خواهد به روش قدیمی خود پایبند باشد و چقدر می‌خواهد روش‌های جدید ناشناخته به امید بهره‌وری سیستم را در پی بگیرد.

طبیعی است که هرچه توصیه‌گر exploit کند، در ارائه پیشنهادهاى جدید خوب عمل نمی‌کند و اگر هم بیشتر بخواهد explore کند ممکن است پیشنهادهاى جدید کیفیت مناسبی نداشته باشند.

راه مقابله:

تنظیم تریدآف: از هر دو روش‌های همزمان استفاده می‌شود و بخشی از exploit بخشی دیگر از آن از explore استفاده می‌کند، این گونه کیفیت لازم گارانتی می‌شود و همچنین در ارائه پیشنهادهاى جدید نیز خوب عمل می‌کند.

استفاده از Thompson sampling: که یک روش heuristic برای انتخاب تصمیمات در مساله exploration-exploitation است.

3. Interpretability

قابلیت تفسیر یکی از مباحث مهمی است که امروزه به خصوص در صنعت هوش مصنوعی و الگوریتم‌های یادگیری ماشین نیز زیاد مطرح می‌شود. اهمیت این بحث معطوف به این است که باید بتوان یک توصیف قابل توجیه ارائه داد که چرا چنین روشی برای حل این مساله مناسب است و چرا این روش کار می‌دهد. این توضیح موجب ایجاد قابلیت اعتماد برای استفاده از آن سیستم می‌شود. در بسیاری از سیستم‌های توصیه‌گر از روش‌هایی مانند الگوریتم‌های یادگیری عمیق، matrix factorization و ... استفاده می‌شود که این الگوریتم‌ها نسبت به بقیه روش‌های از قابلیت تفسیر کمتری برخوردار می‌باشند. هر چه به سمت الگوریتم‌هایی برویم که قابلیت تفسیر بیشتری دارند از پیچیدگی سیستم نیز کم می‌شود. به منظور مقابله می‌توان از ترکیب این روش‌ها با روش‌های دیگر استفاده کرد تا قابلیت تفسیر افزایش یابد و همچنین تا جایی که ممکن است منطق هر محاسبه مشخص باشد.

4. Scalability:

با افزایش دادگان امروزه مقیاس‌پذیری یکی از مباحث مهم است چرا که به سرعت تعداد کاربران و آیت‌ها در حال افزایش است. برای مثال فرض کنید من به برداری برای هر کاربر تشکیل می‌دهم که نشان می‌دهد آیا به انواع ژانرها یا کارگردان‌ها و بازیگران و .. هر کدام چقدر علاقه دارد.

از آن طرف به هر فیلم نیز یک بردار با همان اندازه می‌سازم که نشان می‌دهد هر ژانر چقدر در این فیلم وجود دارد و آیا فلان کارگردان و بازیگر در آن هستند.

ضرب داخلی این دو بردار می‌تواند نشان دهد چقدر این کاربر نسبت به آن فیلم جذب می‌شود. اما مشکل چیست؟ مشکل آن است که تعداد این فیچرهای دستی در حال افزایش است و با اضافه شدن بازیگران و کارگردان‌های جدید و مفاهیم جدید دوباره این بردار باید بزرگتر شود و برای همه کاربران و فیلم‌ها نیز از نو مجدد ساخته شود. این روش مثالی از یک روش غیر مقیاس‌پذیر است.

راه مقابله:

در یادگیری ماشین هم سعی می‌شود که چنین بردارهایی ساخته شود اما فیچرهای آن به صورت دستی وارد نمی‌شود و در واقع یک embedding ای از آن کاربر یا فیلم را نشان می‌دهد که این بردارها سایز یکسانی دارند. این گونه سیستم‌ها نسبت به اضافه شدن کاربران و آیتم‌ها تا جایی که سرعت یادگیری این embedding‌ها معقول باشد مقیاس‌پذیر است. همچنین از روش‌های پردازش massive مانند استفاده از spark و .. نیز می‌توان این دادگان را هندل کرد تا مشکلی در مقیاس‌پذیری آن‌ها بوجود نیاید.