
بسم الله الرحمن الرحيم

Netflix Prize Challenge Remake with Modern Collaborative Filtering Techniques

تهیه‌شده توسط:

علیرضا صمیمی

پرنیان ملک‌زاده

علیرضا تنها

پوریا دیانتی

سهیل سرائیان

تیر ماه ۱۴۰۳

مقدمه

مقدمه

سیستم‌های توصیه‌گر [1] به عنوان یکی از ابزارهای حیاتی در عصر اطلاعات مدرن شناخته می‌شوند که به کاربران کمک می‌کنند تا اطلاعات مرتبط و مفید را از بین حجم عظیمی از داده‌ها پیدا کنند. این سیستم‌ها در حوزه‌های مختلفی مانند خرید آنلاین، خدمات رسانه‌ای، شبکه‌های اجتماعی و بسیاری از زمینه‌های دیگر استفاده می‌شوند. هدف اصلی سیستم‌های توصیه‌گر، پیش‌بینی [2] علایق و ترجیحات کاربران بر اساس داده‌های موجود و ارائه پیشنهاداتی است که با سلایق آنها مطابقت داشته باشد.

یکی از موفق‌ترین روش‌ها برای پیاده‌سازی سیستم‌های توصیه‌گر، استفاده از تکنیک‌های فیلترینگ مشارکتی [3] است. این تکنیک‌ها بر اساس بررسی نظرات و ترجیحات گروهی از کاربران و پیشنهاد موارد مشابه به کاربران دیگر عمل می‌کنند. فیلترینگ مشارکتی به دو دسته کلی تقسیم می‌شود: فیلترینگ مشارکتی مبتنی بر حافظه [4] و فیلترینگ مشارکتی مبتنی بر مدل [5]. هر یک از این روش‌ها دارای مزایا و معایب خاص خود هستند و برای بهبود عملکرد سیستم‌های توصیه‌گر، تکنیک‌های ترکیبی [6] نیز مورد استفاده قرار می‌گیرند.

در اکتبر سال ۲۰۰۶، نتفلیکس یک چالش بزرگ به نام "جایزه نتفلیکس [7]" را راه‌اندازی کرد که هدف آن بهبود دقت سیستم توصیه‌گر فیلم‌های این شرکت بود. نتفلیکس مجموعه داده‌ای عظیم شامل نظرات و امتیازات بیش از ۴۸۰,۰۰۰ کاربر و ۱۷,۷۷۰ فیلم را فراهم کرد و از پژوهشگران و توسعه‌دهندگان خواست تا مدل‌هایی ایجاد کنند که بتوانند دقت پیش‌بینی‌ها را بهبود بخشند. معیار ارزیابی این چالش، ریشه میانگین مربعات خطا [8] یا RMSE بود. این مسابقه توانست توجه بسیاری از پژوهشگران در سراسر جهان را به خود جلب کند و منجر به پیشرفت‌های قابل‌توجهی در زمینه الگوریتم‌های توصیه‌گر شد.

نتایج این چالش نشان داد که ترکیب مدل‌های مبتنی بر عوامل نهان [9] و مدل‌های همسایگی [10] می‌تواند بهبود قابل‌توجهی در دقت پیش‌بینی‌ها ایجاد کند. تیم "BellKor’s Pragmatic Chaos" که توانست جایزه یک میلیون دلاری نتفلیکس را برنده شود، از یک مدل ترکیبی پیچیده استفاده کرد که دقت پیش‌بینی‌ها را به میزان ۱۰.۰۵ درصد نسبت به سیستم اصلی نتفلیکس بهبود بخشید.

در این گزارش، به بررسی تکنیک‌های مختلف فیلترینگ مشارکتی، چالش‌های پیش‌روی این تکنیک‌ها و نتایج به دست آمده از چالش نتفلیکس پرداخته می‌شود. همچنین، به تحلیل مدل‌های ترکیبی و راه‌حل‌های ارائه شده برای مشکلات سیستم‌های توصیه‌گر خواهیم پرداخت .

---

[1] Recommender Systems
[2] Predict
[3] Collabrative Filtering
[4] Memory Based
[5] Model Based
[6] Hybrid Techniques
[7] Netflix Prize
[8] Root Mean Squared Error
[9] Laten Factor Models
[10] Neighborhood Models

ویژگی‌ها و چالش‌های فیلترینگ مشارکتی

الگوریتم‌های توصیه‌گر در تجارت الکترونیک [1]معمولاً در محیطی چالش‌برانگیز عمل می‌کنند. یک سیستم توصیه‌گر که توصیه‌های سریع و دقیقی ارائه می‌دهد، توجه مشتریان را جلب کرده و برای شرکت‌ها سودمند است. تولید پیش‌بینی‌ها یا توصیه‌های با کیفیت بالا بستگی به چگونگی مواجهه آنها با چالش‌ها دارد که این چالش‌ها نیز جزء ویژگی‌های وظایف فیلترینگ مشارکتی محسوب می‌شوند.

پراکندگی[2] داده‌ها

ماتریس کاربر-آیتم مورد استفاده در فیلترینگ مشارکتی بسیار پراکنده است و عملکرد پیش‌بینی‌ها را به چالش می‌کشد.
مشکل شروع سرد [3]زمانی رخ می‌دهد که کاربر یا آیتم جدید وارد سیستم می‌شود و یافتن موارد مشابه به دلیل کمبود اطلاعات دشوار است (مشکل کاربر جدید یا آیتم جدید [۲۱, ۲۲]).

مقیاس‌پذیری[4]

با افزایش تعداد کاربران و آیتم‌ها، الگوریتم‌های سنتی با مشکلات جدی مقیاس‌پذیری مواجه می‌شوند. تکنیک‌های کاهش ابعاد مانند SVD می‌توانند این مشکل را حل کنند، اما نیاز به مراحل پرهزینه تجزیه ماتریس [5]دارند. الگوریتم‌های مبتنی بر حافظه مانند الگوریتم همبستگی پیرسون مبتنی بر آیتم [6]می‌توانند مقیاس‌پذیری مطلوبی ارائه دهند.

هم‌نامی[7]

هم‌نامی به تمایل آیتم‌های مشابه به داشتن نام‌های مختلف اشاره دارد. تکنیک‌های SVD و نمایه‌سازی معنایی پنهان [8]یا LSI می‌توانند این مشکل را حل کنند [۲۴, ۲۵].

[1] E-Commerce Recommendation Algorithms
[2] Sparsity
[3] Cold Start Problem
[4] Scalibility
[5] Matrix Factorization
[6] Item Based Pearson Correlation Algorithm
[7] Synonymy
[8] Latent Semantic Indexing

کاربران خاکستری [1]

کاربران خاکستری به کاربرانی گفته می‌شود که نظراتشان با هیچ گروه خاصی مطابقت ندارد و بنابراین از فیلترینگ مشارکتی بهره نمی‌برند. ترکیب روش‌های فیلترینگ محتوا-محور [2] و فیلترینگ مشارکتی می‌تواند به حل این مشکل کمک کند.

حملات شیلینگ [3]

در سیستم‌های باز، کاربران می‌توانند تعداد زیادی توصیه مثبت برای محصولات خود و توصیه منفی برای محصولات رقبا ثبت کنند. روش‌های مختلفی برای ارزیابی و کشف این حملات ارائه شده است [۴۸, ۴۹].

سایر چالش‌ها

سیستم‌های فیلترینگ مشارکتی همچنین نگرانی‌هایی در مورد حفظ حریم خصوصی کاربران ایجاد می‌کنند. میلر و همکاران [۴] و کنی [۵۲] روش‌هایی برای حفاظت از حریم خصوصی کاربران در وظایف توصیه‌گر CF ارائه کرده‌اند.
قابلیت توضیح [4]نیز جنبه مهم دیگری از سیستم‌های توصیه‌گر است. استدلال‌های شهودی مانند "شما این کتاب را دوست خواهید داشت زیرا آن کتاب‌ها را دوست داشتید" برای خوانندگان جذاب و مفید است، حتی اگر دقت توضیحات بالا نباشد [۵۷].

تکنیک‌های فیلتر کردن بر أساس حافظه:

روش‌های فیلتر کردن بر اساس حافظه از کل یا نمونه‌ای از پایگاه داده کاربر-آیتم برای تولید پیش‌بینی‌ها استفاده می‌کنند. در این روش‌ها، کاربران به گروه‌هایی از افراد با علایق مشابه تعلق دارند. با شناسایی همسایه‌ها برای یک کاربر جدید، می‌توان ترجیحات وی را بر روی آیتم‌های جدید پیش‌بینی کرد.

الگوریتم‌های بر اساس همسایگی [5]که معروف‌ترین روش‌های بر اساس حافظه هستند، به صورت زیر عمل می‌کنند: ابتدا شباهت یا وزن بین کاربران یا آیتم‌ها محاسبه می‌شود. سپس با استفاده از میانگین وزنی تمامی امتیازات کاربران یا آیتم‌ها، پیش‌بینی برای کاربر فعال تولید می‌شود. برای تولید توصیه‌های برتر، پس از محاسبه شباهت‌ها، k نزدیک‌ترین کاربران یا آیتم‌ها شناسایی شده و از ترکیب آنها، توصیه‌ها ارائه می‌شوند.

---

[1] Gray Sheep
[2] Content Based Filtering
[3] Shilling Attacks
[4] Explainibility
[5] Neighborhood Based Algorithms

تکنیک‌های فیلتر کردن بر اساس مدل:

تکنیک‌های فیلتر کردن بر اساس مدل از طراحی و توسعه مدل‌های مختلف (مانند الگوریتم‌های یادگیری ماشین و داده‌کاوی) استفاده می‌کنند تا سیستم بتواند الگوهای پیچیده را تشخیص داده و پیش‌بینی‌های هوشمندانه‌ای برای وظایف فیلتر کردن مشارکتی انجام دهد. این روش‌ها به کمک داده‌های آموزشی، مدل‌هایی را ایجاد می‌کنند که سپس برای داده‌های تست یا دنیای واقعی مورد استفاده قرار می‌گیرند.

مدل‌های مختلفی برای CF بر اساس مدل وجود دارند، از جمله مدل‌های بیزی[1]، مدل‌های خوشه‌بندی[2]، و شبکه‌های وابستگی[3]. به عنوان مثال، الگوریتم ساده بیزی از استراتژی نایو بیز [4]برای پیش‌بینی استفاده می‌کند. همچنین مدل‌های بیزی پیشرفته‌تری مانند NB-ELR و TAN-ELR وجود دارند که دقت بالاتری دارند اما به زمان بیشتری برای آموزش نیاز دارند.

مدل‌های خوشه‌بندی نیز برای بهبود مقیاس‌پذیری و عملکرد سیستم‌های CF به کار می‌روند. این مدل‌ها داده‌ها را به خوشه‌هایی تقسیم کرده و پیش‌بینی‌ها را درون هر خوشه انجام می‌دهند.

در کل، تکنیک‌های فیلتر کردن بر اساس مدل نسبت به روش‌های بر اساس حافظه، دقت و مقیاس‌پذیری بهتری دارند و می‌توانند به طور موثرتری با چالش‌های داده‌های پراکنده و حجم بالای داده‌ها مقابله کنند.

سیستم های فیلتر ترکیبی:

سیستم‌های فیلتر کردن ترکیبی با ترکیب فیلتر کردن مشارکتی با سایر تکنیک‌های توصیه‌گر (معمولاً سیستم‌های مبتنی بر محتوا) به منظور ایجاد پیش‌بینی‌ها یا توصیه‌ها کار می‌کنند. این سیستم‌ها سعی می‌کنند با ترکیب ویژگی‌های مختلف، دقت توصیه‌ها را افزایش داده و مشکلات خاص هر روش را برطرف کنند.

[1] Bayesian
[2] Clustering Models
[3] Dependency Networks
[4] Naïve Bayes

تکنیک‌های مختلفی برای ترکیب وجود دارد:

۱. توصیه‌گرهای ترکیبی محتوا-تقویت شده[1]: این روش‌ها از پیش‌بینی کننده‌های محتوایی (مانند نایو بیز) برای پر کردن مقادیر گم‌شده در ماتریس رتبه‌بندی استفاده می‌کنند و سپس از الگوریتم‌های CF مانند همبستگی پیرسون برای انجام پیش‌بینی‌ها استفاده می‌کنند. این روش‌ها مشکلاتی مانند "شروع سرد" و پراکندگی داده‌ها را برطرف می‌کنند.

۲. توصیه‌گرهای ترکیبی وزنی [2]: این سیستم‌ها نتایج چندین تکنیک توصیه‌گر را با توجه به وزن‌های آنها ترکیب می‌کنند. وزن‌ها می‌توانند ثابت یا قابل تنظیم باشند و از روش‌هایی مانند رأی‌گیری اکثریت وزنی برای ترکیب استفاده می‌شود.

۳. توصیه‌گرهای ترکیبی سوئیچینگ[3]: این سیستم‌ها بین تکنیک‌های مختلف توصیه‌گر سوئیچ می‌کنند، مثلاً زمانی که سیستم CF نمی‌تواند با اطمینان کافی توصیه‌ای بدهد، سیستم مبتنی بر محتوا فعال می‌شود.

۴. توصیه‌گرهای ترکیبی مخلوط[4]: این سیستم‌ها چندین تکنیک توصیه‌گر را همزمان برای یک کاربر اعمال کرده و نتایج را ترکیب می‌کنند.

ترکیب تکنیک‌های CF با سیستم‌های دیگر می‌تواند دقت توصیه‌ها را بهبود بخشد، به ویژه در مواردی که یک الگوریتم CF معمولی نمی‌تواند توصیه‌های رضایت‌بخشی ارائه دهد. با این حال، سیستم‌های ترکیبی معمولاً به اطلاعات خارجی نیاز دارند و پیاده‌سازی آن‌ها پیچیده‌تر است.

معیارهای ارزیابی

برای ارزیابی کیفیت یک سیستم توصیه‌گر، از معیارهای مختلفی استفاده می‌شود که بسته به نوع کاربردهای فیلتر کردن مشارکتی متفاوت هستند. طبق Herlocker و همکاران [۶۰]، این معیارها به چند دسته عمده تقسیم می‌شوند: معیارهای دقت پیش‌بینی، معیارهای دقت طبقه‌بندی، و معیارهای دقت رتبه‌بندی. در ادامه، برخی از معیارهای پرکاربرد برای CF معرفی می‌شوند.

[1] Content Boosted
[2] Weighed Hybrid
[3] Switching
[4] Mixed

میانگین خطای مطلق [1] و میانگین خطای مطلق نرمال شده [2]

میانگین خطای مطلق یا MAE که در ادبیات تحقیق CF به طور گسترده‌ای استفاده می‌شود، میانگین تفاوت مطلق بین پیش‌بینی‌ها و امتیازات واقعی را محاسبه می‌کند:

$$MAE = \frac{\sum_{i,j}|p_{i,j} - r_{i,j}|}{n}$$

که در آن n تعداد کل امتیازات است، $p_{i,j}$ امتیاز پیش‌بینی شده برای کاربر i بر روی آیتم j و $r_{i,j}$ امتیاز واقعی است. هرچه MAE کمتر باشد، پیش‌بینی دقیق‌تر است.

سیستم‌های توصیه‌گر مختلف ممکن است از مقیاس‌های عددی متفاوتی برای امتیازدهی استفاده کنند. میانگین خطای مطلق نرمال شده NMAE یا MAE را نرمال می‌کند تا خطاها به صورت درصدی از مقیاس کامل بیان شوند:

$$NMAE = \frac{MAE}{r_{max} - r_{min}}$$

که در آن r_{max} و r_{min} به ترتیب بالاترین و پایین‌ترین مقادیر امتیازدهی هستند.

میانگین مربعات خطا

ریشه میانگین مربعات خطا یا RMSE به دلیل استفاده در جایزه نتفلیکس برای عملکرد توصیه فیلم‌ها محبوبیت بیشتری پیدا کرده است:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i,j}(p_{i,j} - r_{i,j})^2}$$

که در آن n تعداد کل امتیازات است، $p_{i,j}$ امتیاز پیش‌بینی شده برای کاربر i بر روی آیتم j، و $r_{i,j}$ امتیاز واقعی است. RMSE تاثیر خطاهای بزرگ‌تر را نسبت به خطاهای کوچک‌تر تقویت می‌کند.

[1] Mean Absolute Error
[2] Normalized Mean Absolute Error

در ادامه این گزارش، به بررسی مقالات مختلف در زمینه موضوع این گزترش، پرداخته می‌شود.

الگوریتم CDL

معرفی الگوریتم CDL[1] و دلایل انتخاب آن

الگوریتم CDL یکی از روش‌های نوین و هیبریدی در سیستم‌های توصیه‌گر [2]است که با ترکیب یادگیری عمیق و فیلترینگ مشارکتی[3]، بهبود قابل‌توجهی در دقت و کارایی توصیه‌ها ایجاد می‌کند. این الگوریتم که توسط تیمی از پژوهشگران دانشگاه علم و فناوری هنگ کنگ توسعه یافته، به منظور حل مشکلات موجود در روش‌های فیلترینگ مشارکتی سنتی ارائه شده است. یکی از مشکلات اصلی فیلترینگ مشارکتی، پراکندگی داده‌ها و کمبود اطلاعات است که CDL با استفاده از یادگیری عمیق و بهره‌گیری از اطلاعات محتوایی، این مشکل را به‌طور موثری کاهش می‌دهد. در الگوریتم CDL، یادگیری نمایشی عمیق برای اطلاعات محتوایی و فیلترینگ مشارکتی برای ماتریس امتیازات کاربران به صورت همزمان انجام می‌شود که منجر به بهبود قابل توجه در دقت پیش‌بینی‌ها[4] می‌شود. آزمایشات گسترده بر روی سه مجموعه داده [5]واقعی از حوزه‌های مختلف نشان داده‌اند که CDL توانایی بالایی در بهبود دقت توصیه‌ها و پیشی گرفتن از روش‌های موجود دارد. مدل CDL به گونه‌ای طراحی شده است که می‌تواند با استفاده از مدل‌های یادگیری عمیق مختلف مانند شبکه‌های بولتزمن عمیق [6]یا DBN، شبکه‌های عصبی مکرر [7]یا RNN، و شبکه‌های عصبی پیچشی [8] یا CNN گسترش یابد، که باعث انعطاف‌پذیری[9] و تعمیم‌پذیری[10] بالای این مدل می‌شود. همچنین، استفاده از مدل‌های یادگیری عمیق باعث می‌شود که CDL توانایی استخراج نمایش‌ها[11]ی عمیق و موثر از اطلاعات محتوایی را داشته باشد، که به بهبود دقت توصیه‌ها و کاهش تاثیر مشکلاتی مانند پراکندگی داده‌ها و مشکل شروع سرد [12]کمک می‌کند. به‌طور کلی، استفاده از الگوریتم CDL می‌تواند به شرکت‌ها و کاربران کمک کند تا از تجربیات بهتری در استفاده از سیستم‌های توصیه‌گر بهره‌مند شوند.

توضیح مدل و مسئله به زبان ساده

در این مقاله، مشابه کار در [۳۴] مسئله توصیه‌گر با استفاده از بازخورد ضمنی [13]کاربران بررسی می‌شود. بازخورد ضمنی به داده‌هایی اشاره دارد که به طور غیرمستقیم از رفتار کاربران به دست می‌آید، مانند این که چه مقالات یا فیلم‌هایی را یک کاربر در کتابخانه شخصی خود ذخیره کرده است.

---

[1] Collabrative Deep Learning
[2] Recommender Systems
[3] Collabrative Filtering
[4] Predictions
[5] Dateset
[6] Deep Boltzmann Networks
[7] Recurrent Neural Networks
[8] Convolutional Neural Networks
[9] Flexibility
[10] Generalizability
[11] Representations
[12] Cold Start Problem
[13] Implicit Feedback

ما از یک مجموعه شامل J آیتم (مانند مقالات یا فیلم‌ها) استفاده می‌کنیم که به صورت یک ماتریس J-by-S به نام X_c نمایش داده می‌شود. هر سطر در این ماتریس نشان‌دهنده یک آیتم است که به صورت یک بردار کیسه کلمات[1] ساخته شده است، و این بردار بر اساس یک واژگان با اندازه S ساخته شده است.

همچنین، یک ماتریس باینری I-by-J به نام R تعریف می‌کنیم که در آن I تعداد کاربران و J تعداد آیتم‌ها است. در این ماتریس، اگر کاربر i آیتم j را در کتابخانه شخصی خود داشته باشد، مقدار R_{ij} برابر با ۱ و در غیر این صورت ۰ است. هدف ما این است که با استفاده از بخشی از امتیازات موجود در R و اطلاعات محتوایی X_c، سایر امتیازات را پیش‌بینی کنیم. این مدل به طور خاص برای توصیه فیلم‌ها و مقالات طراحی شده است، اما می‌تواند برای توصیه‌های دیگر نیز استفاده شود.

ماتریس X_c به عنوان ورودی پاک[2] به شبکه SDAE داده می‌شود، در حالی که ماتریس دیگری با نویز به نام X_0 نیز وجود دارد. خروجی هر لایه از شبکه SDAE به صورت یک ماتریس J-by-Kl نمایش داده می‌شود که هر سطر آن یک بردار برای آیتم j است. W_l و b_l به ترتیب ماتریس وزن‌ها و بردار بایاس لایه l ام هستند. برای راحتی، از W^+ برای اشاره به مجموعه تمامی لایه‌های ماتریس وزن‌ها و بایاس‌ها استفاده می‌کنیم. لازم به ذکر است که یک شبکه SDAE با $L/2$ لایه معادل یک شبکه L-لایه است.

فرمولاسیون بهینه‌سازی

شبکه SDAE مسئله بهینه‌سازی زیر را حل می‌کند:

$$\min_{\{W_l\},\{b_l\}} |X_c - X_L|_F^2 + \lambda \sum_l |W_l|_F^2$$

که در آن λ یک پارامتر تنظیمی[3] و $|\cdot|_F$ نشان‌دهنده نرم فروبنیوس[4] است.

[1] Bag of Words
[2] Clean Input
[3] Regularization Parameter
[4] Frobenius Norm

فرآیند مولد Bayesian SDAE

فرض کنیم که ورودی پاک X_c و ورودی نویزدار[1] X_0 هر دو مشاهده شده‌اند. فرآیند مولد به صورت زیر تعریف می‌شود:

۱. برای هر لایه l از شبکه SDAE:

- برای هر ستون n از ماتریس وزن W_l:

$$W_{l,*n} \sim N\left(0, \beta_w^{-1} I_{K_l}\right)$$

- بردار بایاس b_l:

$$b_l \sim N\left(0, \beta_w^{-1} I_{K_l}\right)$$

- برای هر سطر j از X_l :

$$X_{l,j*} \sim N\left(\sigma\left(X_{l-1,j*} W_l + b_l\right), \beta_s^{-1} I_{K_l}\right)$$

۲. برای هر آیتم j:

$$X_{c,j*} \sim N\left(X_{L,j*}, \beta_n^{-1} I_J\right)$$

فرآیند مولد CDL

با استفاده از Bayesian SDAE به عنوان یک جزء، فرآیند مولد CDL به صورت زیر تعریف می‌شود:

۱. برای هر لایه l از شبکه SDAE:

- برای هر ستون n از ماتریس وزن W_l:

$$W_{l,*n} \sim N\left(0, \beta_w^{-1} I_{K_l}\right)$$

- بردار بایاس b_l:

$$b_l \sim N\left(0, \beta_w^{-1} I_{K_l}\right)$$

- برای هر سطر j از X_l :

$$X_{l,j*} \sim N\left(\sigma\left(X_{l-1,j*} W_l + b_l\right), \beta_s^{-1} I_{K_l}\right)$$

۲. برای هر آیتم j:

- ورودی پاک:

[1] Noisy Input

$$X_{c,j*} \sim N\left(X_{L,j*}, \beta_n^{-1} I_J\right)$$

- بردار آفست آیتم:

$$\epsilon_j \sim N(0, \beta_v^{-1} I_K)$$

سپس بردار آیتم نهان به صورت زیر تنظیم می‌شود:

$$v_j = \epsilon_j + X_{L/2,j*}^T$$

۳. بردار کاربر نهان برای هر کاربر i:

$$u_i \sim N(0, \beta_u^{-1} I_K)$$

۴. امتیاز R_{ij} برای هر جفت کاربر-آیتم (i,j):

$$R_{ij} \sim N\left(u_i^T v_j, C_{ij}^{-1}\right)$$

به‌روزرسانی پارامترها

برای به‌روزرسانی پارامترهای u_i و v_j از روش صعود هماهنگ استفاده می‌شود:

$$u_i \leftarrow (VC_iV^T + \beta_u I_K)^{-1} VC_iR_i$$
$$v_j \leftarrow \left(UC_jU^T + \beta_v I_K\right)^{-1} \left(UC_jR_j + \beta_v f_e\left(X_{0,j*}, W^+\right)^T\right)$$

که در آن $U = u_i i = 1^I$ ، $V = v_j j = 1^J$ ، C_i یک ماتریس قطری است و R_i یک بردار ستونی حاوی تمامی امتیازات کاربر i است.

پیش‌بینی

برای پیش‌بینی امتیازات، از تخمین نقطه‌ای u_i ، W^+ و ϵ_j استفاده می‌شود:

$$E\left[R_{ij}|D\right] \approx E[u_i|D]^T \left(E\left[f_e\left(X_{0,j*}, W^+\right)^T \middle| D\right] + E\left[\epsilon_j|D\right]\right)$$

این فرمول‌ها و توضیحات، الگوریتم CDL را به صورت جامع و ساده تشریح می‌کنند و نشان می‌دهند که چگونه این روش هیبریدی می‌تواند دقت و کارایی سیستم‌های توصیه‌گر را بهبود بخشد.

مجموعه داده‌ها

مجموعه داده نتفلیکس، از دو بخش تشکیل شده است. بخش اول شامل امتیازات و عناوین فیلم‌ها از مجموعه داده چالش نتفلیکس [1]است. بخش دوم شامل خلاصه داستان‌های مربوط به فیلم‌ها است که توسط نویسندگان از IMDB جمع‌آوری شده است. به منظور همخوانی با تنظیمات بازخورد ضمنی در دو مجموعه داده اول، فقط امتیازات مثبت (امتیاز ۵) برای آموزش و تست استخراج شده‌اند. پس از حذف کاربران با کمتر از ۳ امتیاز مثبت و فیلم‌های بدون خلاصه داستان، مجموعه داده نهایی شامل ۴۰۷۲۶۱ کاربر، ۹۲۲۸ فیلم و ۱۵۳۴۸۸۰۸ امتیاز است.

فرآیند پیش‌پردازش

ما از همان روشی که در مقاله [۳۴] آمده است برای پیش‌پردازش اطلاعات متنی (محتوای آیتم‌ها) استخراج شده از عناوین و چکیده‌های مقالات و خلاصه داستان‌های فیلم‌ها استفاده می‌کنیم. پس از حذف کلمات توقف[2]، بالاترین S کلمه‌های متمایز بر اساس مقادیر tf-idf انتخاب می‌شوند تا واژگان را تشکیل دهند. مقدار S برای سه مجموعه داده به ترتیب ۸۰۰۰، ۲۰۰۰۰ و ۲۰۰۰۰ است.

طرح ارزیابی[3]

برای هر مجموعه داده، مشابه [۳۵, ۳۶]، ما به صورت تصادفی P آیتم مرتبط با هر کاربر را برای تشکیل مجموعه آموزش انتخاب کرده و بقیه مجموعه داده را به عنوان مجموعه تست استفاده می‌کنیم. برای ارزیابی و مقایسه مدل‌ها در شرایط پراکندگی[4] و تراکم[5]، مقدار P را به ترتیب ۱ و ۱۰ تنظیم می‌کنیم. برای هر مقدارP ، ارزیابی را پنج بار با مجموعه‌های آموزشی مختلف تصادفی تکرار می‌کنیم و عملکرد متوسط گزارش می‌شود.

معیار ارزیابی

همانند [۳۴, ۲۲, ۳۵]، از معیار Recall به عنوان معیار عملکرد استفاده می‌کنیم زیرا اطلاعات امتیازدهی به صورت بازخورد ضمنی است. به طور خاص، یک ورودی صفر ممکن است به دلیل عدم علاقه کاربر به آیتم یا عدم آگاهی او از وجود آن باشد. از این رو، Percision معیار مناسبی برای ارزیابی عملکرد نیست. مانند بیشتر سیستم‌های توصیه‌گر، امتیازات پیش‌بینی شده آیتم‌های کاندید را مرتب کرده و بالاترین M [6]آیتم را به کاربر هدف توصیه می‌کنیم. معیار recall@M برای هر کاربر به صورت زیر تعریف می‌شود:

---

[1] Challenge Netfilx
[2] Stop Words
[3] Evaluation
[4] Sparsity
[5] Density
[6] Top M

$$\text{recall@M} = \frac{\text{number of items user likes among the top M}}{\text{total number of items user likes}}$$

نتیجه نهایی گزارش شده میانگین Recall برای تمامی کاربران است. یک معیار ارزیابی دیگر میانگین دقت متوسط [1]یا mAP است. دقیقا همانند [۲۱]، نقطه قطع [2]را برای هر کاربر روی ۵۰۰ تنظیم می‌کنیم.

مقایسه کمی

شکل‌های ۴ و ۵ نتایج مقایسه الگوریتم‌های CDL، CTR، DeepMusic، CMF و SVDFeature را با استفاده از سه مجموعه داده تحت شرایط پراکنده (P=۱) و متراکم (P=۱۰) نشان می‌دهند. مشاهده می‌شود که CTR یک خط مبنای [3]قوی است که در تمامی مجموعه داده‌ها از DeepMusic، CMF و SVDFeature بهتر عمل می‌کند، حتی با وجود این که DeepMusic دارای یک معماری عمیق است.

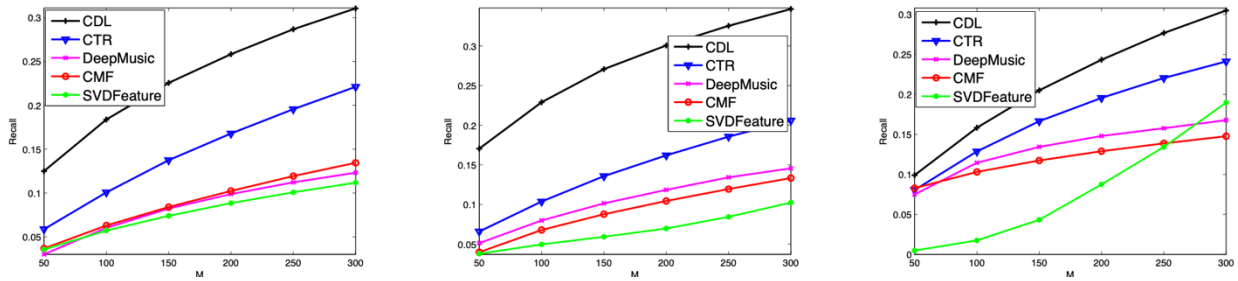
در شرایط پراکنده، CMF اغلب از SVDFeature بهتر عمل می‌کند و گاهی حتی به عملکرد قابل مقایسه‌ای با CTR دست می‌یابد. DeepMusic به دلیل کمبود امتیازات و بیش‌برازش[4]، عملکرد ضعیفی دارد. در شرایط متراکم، SVDFeature برای مجموعه داده‌های citeulike-a و citeulike-t به طور قابل توجهی بهتر از CMF است، اما برای نتفلیکس ضعیف‌تر از CMF عمل می‌کند. DeepMusic همچنان کمی ضعیف‌تر از CTR است.

برای مقایسه مشخص‌تر CDL با CTR، مشاهده می‌شود که برای citeulike-a، CDL دو لایه در شرایط پراکنده بین ۴.۲٪ تا ۶.۰٪ و در شرایط متراکم بین ۳.۳٪ تا ۴.۶٪ بهتر از CTR عمل می‌کند. اگر تعداد لایه‌ها به ۳ (L=۶) افزایش یابد، این اختلاف به ۵.۸٪ تا ۸.۰٪ و ۴.۳٪ تا ۵.۸٪ می‌رسد. به طور مشابه برای citeulike-t، CDL دو لایه در شرایط پراکنده بین ۱۰.۴٪ تا ۱۳.۱٪ و در شرایط متراکم بین ۴.۷٪ تا ۷.۶٪ بهتر از CTR است. با افزایش تعداد لایه‌ها به ۳، این اختلاف به ۱۱.۰٪ تا ۱۴.۹٪ و ۵.۲٪ تا ۸.۲٪ می‌رسد. برای نتفلیکس، CDL دو لایه در شرایط پراکنده بین ۱.۹٪ تا ۵.۹٪ و در شرایط متراکم بین ۱.۵٪ تا ۲.۰٪ بهتر از CTR عمل می‌کند.

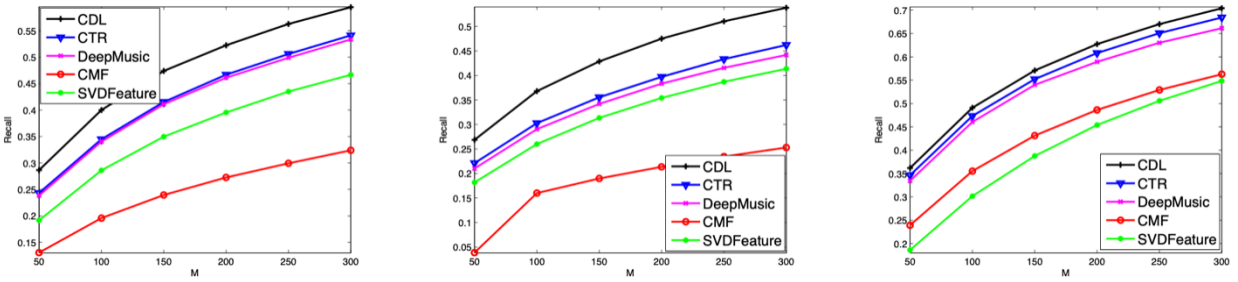
[1] Mean Average Percision
[2] Cutoff Point
[3] Baseline
[4] Overfitting

جدول‌های ۲ و ۳ نتایج recall@۳۰۰ را هنگامی که CDL با تعداد لایه‌های مختلف به سه مجموعه داده اعمال می‌شود، نشان می‌دهند. برای نتفلیکس، recall با افزایش تعداد لایه‌ها افزایش می‌یابد.

با یکپارچه‌سازی بدون درز یادگیری نمایشی عمیق برای اطلاعات محتوایی و فیلترینگ مشارکتی برای ماتریس امتیازات، CDL می‌تواند هم ماتریس امتیازات پراکنده و هم اطلاعات متنی پراکنده را بهتر مدیریت کرده و نمایشی نهان و مؤثرتر برای هر آیتم و در نتیجه هر کاربر بیاموزد.



شکل ۴: مقایسه عملکرد CDL، CTR، DeepMusic، CMF و SVDFeature بر اساس معیار recall@M برای مجموعه داده‌های citeulike-a، citeulike-t و نتفلیکس در شرایط پراکنده. از CDL دو لایه استفاده شده است.



شکل ۵: مقایسه عملکرد CDL، CTR، DeepMusic، CMF و SVDFeature بر اساس معیار recall@M برای مجموعه داده‌های citeulike-a، citeulike-t و نتفلیکس در شرایط متراکم. از CDL دو لایه استفاده شده است.

#layers	۱	۲	۳
Netflix	۲۹.۲۰	۳۰.۵۰	۳۱.۰۱

جدول ۲ : Recall@۳۰۰ در شرایط پراکنده (٪)

#layers	۱	۲	۳
Netflix	۶۹.۲۶	۷۰.۴۰	۷۰.۴۲

جدول ۳ : Recall@۳۰۰ در شرایط متراکم (٪)

الگوریتم CDL به عنوان یک روش نوین و هیبریدی برای بهبود دقت و کارایی سیستم‌های توصیه‌گر معرفی شد. این الگوریتم با ترکیب یادگیری عمیق و فیلترینگ مشارکتی، توانسته است مشکلات مربوط به پراکندگی داده‌ها و کمبود اطلاعات را به طور موثری حل کند. آزمایشات گسترده بر روی سه مجموعه داده واقعی از حوزه‌های مختلف نشان داده‌اند که CDL در مقایسه با روش‌های موجود مانند CTR، DeepMusic، CMF و SVDFeature عملکرد بهتری دارد. همچنین، نتایج نشان می‌دهند که افزایش تعداد لایه‌های شبکه می‌تواند به بهبود دقت توصیه‌ها کمک کند، هرچند باید مراقب بیش‌برازش بود. در نهایت، پیشنهاد می‌شود که در آینده به بررسی و بهبود بیشتر این الگوریتم پرداخته شود تا بتواند در شرایط و حوزه‌های مختلف کارایی بهتری داشته باشد.

تیم برنده رقابت: BellKor

مقدمه

تیم BellKor یک راه‌حل نوآورانه به نام "BellKor's Pragmatic Chaos" را برای برنده شدن جایزه بزرگ نتفلیکس توسعه داد. رویکرد آن‌ها، ترکیبی از پیش‌بینی‌کننده‌های جدید با کارهای قبلی بود که شامل بهبود پیش‌بینی‌کننده‌های پایه، مدل‌های تجزیه ماتریس، دینامیک زمانی، ماشین‌های بولتزمن محدود (RBM) و الگوریتم‌های ترکیب‌کننده است.

تیم BellKor با استفاده از روش‌های مختلف و تلفیق آن‌ها توانست به دقت بالاتری در پیش‌بینی رتبه‌بندی فیلم‌ها توسط کاربران دست یابد. این راه‌حل شامل استفاده از مدل‌های مختلفی بود که هر کدام به نحوی به بهبود پیش‌بینی‌ها کمک کردند. در ادامه به توضیح مقدمات و نحوه استفاده از داده‌های نتفلیکس پرداخته می‌شود.

مقدمات

مجموعه داده نتفلیکس شامل بیش از ۱۰۰ میلیون رتبه‌بندی از ۴۸۰,۱۸۹ کاربر روی ۱۷,۷۷۰ فیلم در دوره زمانی بین ۳۱ دسامبر ۱۹۹۹ تا ۳۱ دسامبر ۲۰۰۵ است. این مجموعه داده بسیار بزرگ و پیچیده به تیم BellKor این امکان را داد تا از روش‌های مختلفی برای تحلیل و پیش‌بینی استفاده کنند.

داده‌های مسابقه به دو بخش اصلی تقسیم شدند:

۱. مجموعه آموزشی :که برای آموزش مدل‌ها و پیدا کردن الگوهای مختلف در داده‌ها استفاده می‌شد.
۲. مجموعه نگه‌داشت :که برای ارزیابی مدل‌ها و تعیین دقت پیش‌بینی‌ها استفاده می‌شد.

معیار ارزیابی

معیار ارزیابی اصلی در این مسابقه ریشه میانگین مربع خطا (RMSE) بود. این معیار میزان خطای پیش‌بینی مدل‌ها را اندازه‌گیری می‌کرد و تیم‌ها باید مدل‌هایی را توسعه می‌دادند که کمترین RMSE را داشته باشند.

نشانه‌گذاری و علائم

در این مقاله، برای راحتی در تفکیک کاربران و فیلم‌ها از علائم خاصی استفاده شده است. به طور مثال:

- کاربران با u و v نشان داده می‌شوند.
- فیلم‌ها با i و j نشان داده می‌شوند.
- r_{ui} نشان‌دهنده رتبه‌بندی کاربر u برای فیلم i است.

زمان

یکی از عوامل مهم در پیش‌بینی رتبه‌بندی‌ها، تأثیر زمان بود. تیم BellKor با در نظر گرفتن زمان در مدل‌های خود، توانست تغییرات زمانی در ترجیحات کاربران و محبوبیت فیلم‌ها را بهتر مدل‌سازی کند. زمان به عنوان تعداد روزهای گذشته از یک نقطه مرجع اندازه‌گیری می‌شد و این اطلاعات در مدل‌های پیش‌بینی به کار گرفته می‌شد.

این مقدمات نشان می‌دهد که تیم BellKor چگونه با استفاده از داده‌های بزرگ و پیچیده نتفلیکس و تلفیق روش‌های مختلف، توانست به دقت بالایی در پیش‌بینی رتبه‌بندی فیلم‌ها دست یابد و در نهایت برنده جایزه بزرگ نتفلیکس شود.

پیش‌بینی کننده‌های پایه

پیش‌بینی کننده‌های پایه (Baseline Predictors) برای ثبت اثرات کاربران و آیتم‌ها به طور مستقل استفاده می‌شوند. این پیش‌بینی کننده‌ها شامل تعدادی پارامتر هستند که سعی در مدل‌سازی میانگین رتبه‌بندی‌های کاربران و آیتم‌ها دارند.

فرمول پیش‌بینی پایه به این صورت است:

$$b_i + b_u + \mu = r_{ui}$$

که در آن:

- r_{ui} پیش‌بینی رتبه‌بندی کاربر u برای فیلم i است.
- μ میانگین کل رتبه‌بندی‌ها در مجموعه داده است.
- b_u تعصب (bias) کاربر u است، که نمایانگر تمایل کاربر به رتبه‌بندی بیشتر یا کمتر از میانگین است.
- b_i تعصب (bias) فیلم i است، که نمایانگر تمایل فیلم به دریافت رتبه‌بندی‌های بالاتر یا پایین‌تر از میانگین است.

تخمین پارامترها

برای تخمین پارامترهای b_i و b_u، از تکنیک‌های بهینه‌سازی مانند نزول گرادیان

(gradient descent) استفاده می‌شود. همچنین، برای جلوگیری از بیش‌برازش(overfitting) ، از روش‌های تنظیم (regularization)استفاده می‌شود.

مثال

فرض کنید میانگین کل رتبه‌بندی‌ها μ برابر با ۳.۵ باشد. یک کاربر خاص u معمولاً رتبه‌بندی‌های بیشتری از میانگین به فیلم‌ها می‌دهد (مثلاً تعصب $b_u = ۰.۵$) و یک فیلم خاص i نیز به طور معمول رتبه‌بندی‌های بالاتری دریافت می‌کند (مثلاً تعصب $b_i = ۰.۲$) در این صورت، پیش‌بینی رتبه‌بندی کاربر u برای فیلم i به صورت زیر خواهد بود

$$r_{ui} = ۳.۵ + ۰.۵ + ۰.۲ = ۴.۲$$

پیش‌بینی‌کننده‌های مبتنی بر فراوانی

فراوانی‌ها (Frequencies)

یکی از عوامل مهم در پیش‌بینی رتبه‌بندی‌ها، فراوانی رتبه‌بندی‌هاست. فراوانی به تعداد دفعاتی که یک کاربر در یک روز خاص آیتم‌ها را رتبه‌بندی کرده است، اشاره دارد. این مفهوم به عنوان یک ویژگی مهم در مدل‌های پیش‌بینی‌کننده وارد شده است تا الگوهای رفتاری کاربران را بهتر درک و مدل‌سازی کند.

فراوانی و اثر آن در پیش‌بینی‌ها فرض کنید F_{ui} فراوانی رتبه‌بندی کاربر u برای آیتم i در یک روز خاص باشد. مطالعات نشان می‌دهد که این فراوانی می‌تواند تأثیر قابل توجهی در دقت پیش‌بینی‌ها داشته باشد. به همین دلیل، تیم BellKor تصمیم گرفت تا این عامل را به مدل‌های پیش‌بینی خود اضافه کند.

لگاریتم فراوانی (Log-Frequency)

برای بهتر مدل‌سازی اثرات فراوانی، از لگاریتم فراوانی استفاده می‌شود. لگاریتم فراوانی به صورت F_{ui} تعریف می‌شود که لگاریتم طبیعی فراوانی است. این کار به مدل کمک می‌کند تا بهتر بتواند اثرات غیرخطی فراوانی را ثبت کند.

فرمول جدید با در نظر گرفتن فراوانی

با اضافه کردن فراوانی به مدل‌های پیش‌بینی، فرمول پیش‌بینی به صورت زیر تغییر می‌کند

$$F_{ui} + b_i + b_u + \mu = r_{ui}$$

در این فرمول:

- f_{ui} لگاریتم فراوانی رتبه‌بندی کاربر u برای آیتم i است.
- سایر پارامترها همانند فرمول پایه باقی می‌مانند.

تخمین پارامترها

پارامترهای جدید با استفاده از تکنیک‌های بهینه‌سازی مشابه با پارامترهای پیشین تخمین زده می‌شوند. تنظیم (regularization)همچنان برای جلوگیری از بیش‌برازش مورد استفاده قرار می‌گیرد.

مثال

فرض کنید میانگین کل رتبه‌بندی‌ها μ برابر با ۳.۵ باشد، تعصب کاربر b_u برابر با ۰.۵ و تعصب آیتم b_i برابر با ۰.۲ باشد. اگر فراوانی رتبه‌بندی کاربر u برای آیتم i در یک روز خاص برابر با ۵ باشد، لگاریتم فراوانی f_{ui} برابر با $\ln(۵)$ خواهد بود.

فرمول پیش‌بینی جدید به صورت زیر خواهد بود

$$\text{Ln}(۵) + ۰.۲ + ۰.۵ + ۳.۵ = r_{ui}$$

بهبود دقت پیش‌بینی

با اضافه کردن فراوانی به مدل‌های پیش‌بینی، دقت پیش‌بینی‌ها به طور قابل توجهی بهبود یافته است. این نشان می‌دهد که فراوانی به عنوان یک ویژگی مهم در مدل‌های پیش‌بینی رتبه‌بندی‌ها نقش دارد و می‌تواند به بهبود عملکرد مدل‌ها کمک کند.

مزایای استفاده از فراوانی

- ثبت الگوهای رفتاری کاربران: فراوانی می‌تواند به شناسایی الگوهای رفتاری کاربران کمک کند و مدل‌های پیش‌بینی را دقیق‌تر کند.
- درک بهتر تأثیر زمان: با استفاده از فراوانی، می‌توان تأثیر زمان بر رتبه‌بندی‌ها را بهتر درک کرد.
- بهبود دقت پیش‌بینی: اضافه کردن فراوانی به مدل‌های پیش‌بینی می‌تواند به طور قابل توجهی دقت پیش‌بینی‌ها را افزایش دهد.

مدل‌های تجزیه ماتریس و همسایگی

مدل‌های تجزیه ماتریس (Matrix Factorization)

تجزیه ماتریس یکی از تکنیک‌های کلیدی در سیستم‌های توصیه‌گر است که هدف آن مدل‌سازی روابط پنهان بین کاربران و آیتم‌ها است. این روش با تجزیه ماتریس رتبه‌بندی به دو ماتریس کوچک‌تر که یکی برای کاربران و دیگری برای آیتم‌ها است، سعی می‌کند الگوهای پنهان در داده‌ها را کشف کند. در نتیجه، این مدل‌ها قادر به پیش‌بینی دقیق‌تر رتبه‌بندی‌ها هستند.

چگونگی کارکرد مدل‌های تجزیه ماتریس

در این مدل‌ها، رتبه‌بندی یک کاربر برای یک آیتم خاص به صورت ترکیبی از میانگین کلی رتبه‌بندی‌ها، تعصب‌های خاص هر کاربر و آیتم، و تعاملات بین ویژگی‌های پنهان کاربران و آیتم‌ها محاسبه می‌شود. این ویژگی‌های پنهان به شکل بردارهایی نمایش داده می‌شوند که تعامل بین آن‌ها رتبه‌بندی نهایی را تعیین می‌کند.

بهینه‌سازی پارامترها

برای تخمین پارامترهای مدل که شامل تعصب‌های کاربران و آیتم‌ها و بردارهای ویژگی‌های پنهان هستند، از روش‌های بهینه‌سازی مانند نزول گرادیان استفاده می‌شود. هدف از بهینه‌سازی، کاهش اختلاف بین رتبه‌بندی‌های واقعی و پیش‌بینی‌شده است. این فرآیند شامل به‌روزرسانی مستمر پارامترها بر اساس خطاهای مشاهده‌شده است تا مدل بتواند به تدریج پیش‌بینی‌های دقیق‌تری انجام دهد.

مزایا و معایب مدل‌های تجزیه ماتریس

مزیت اصلی این مدل‌ها توانایی بالا در مدل‌سازی تعاملات پیچیده بین کاربران و آیتم‌ها است که منجر به دقت بالاتر در پیش‌بینی رتبه‌بندی‌ها می‌شود. همچنین این مدل‌ها می‌توانند الگوهای پنهان را به خوبی کشف کنند. از معایب این روش‌ها می‌توان به پیچیدگی محاسباتی و نیاز به زمان بیشتر برای آموزش مدل‌ها اشاره کرد. علاوه بر این، تنظیم دقیق پارامترها برای جلوگیری از بیش‌برازش اهمیت زیادی دارد.

کاربردهای عملی

مدل‌های تجزیه ماتریس به طور گسترده‌ای در سیستم‌های توصیه‌گر استفاده می‌شوند. برای مثال، نتفلیکس، آمازون و اسپاتیفای از این مدل‌ها برای ارائه توصیه‌های شخصی‌سازی‌شده به کاربران خود بهره می‌برند. این مدل‌ها با استفاده از ویژگی‌های پنهان، می‌توانند به دقت بالا در پیش‌بینی رتبه‌بندی‌ها و بهبود تجربه کاربران دست یابند.

مدل‌های همسایگی (Neighborhood Models)

علاوه بر مدل‌های تجزیه ماتریس، مدل‌های همسایگی نیز به طور گسترده در سیستم‌های توصیه‌گر استفاده می‌شوند. این مدل‌ها بر اساس شباهت بین کاربران یا آیتم‌ها کار می‌کنند. در این مدل‌ها، رتبه‌بندی یک آیتم برای یک کاربر بر اساس رتبه‌بندی‌های آیتم‌های مشابه یا کاربران مشابه پیش‌بینی می‌شود. مدل‌های همسایگی به دو دسته اصلی تقسیم می‌شوند:

۱. مدل‌های همسایگی کاربر-محور (User-based):

این مدل‌ها شباهت بین کاربران را اندازه‌گیری می‌کنند و رتبه‌بندی‌های یک کاربر خاص را بر اساس رتبه‌بندی‌های کاربران مشابه پیش‌بینی می‌کنند.

۲. مدل‌های همسایگی آیتم-محور (Item-based):

این مدل‌ها شباهت بین آیتم‌ها را اندازه‌گیری می‌کنند و رتبه‌بندی‌های یک آیتم خاص را بر اساس رتبه‌بندی‌های آیتم‌های مشابه پیش‌بینی می‌کنند.

مزایا و معایب مدل‌های همسایگی

مزیت اصلی مدل‌های همسایگی سادگی و قابل‌توضیح بودن آن‌ها است. این مدل‌ها به راحتی قابل پیاده‌سازی هستند و می‌توانند توضیحات قابل فهمی برای توصیه‌ها ارائه دهند. از معایب این مدل‌ها می‌توان به محدودیت در کشف الگوهای پیچیده و نیاز به محاسبات زیاد برای یافتن همسایه‌های مشابه اشاره کرد.

تکنیک‌های پیشرفته و ترکیب نهایی

مدل‌های پیشرفته تجزیه ماتریس

مدل‌های تجزیه ماتریس از تکنیک‌های پایه در سیستم‌های توصیه‌گر فراتر رفته و شامل مدل‌های پیشرفته‌تری هستند که توانایی بیشتری در مدل‌سازی الگوهای پیچیده در داده‌ها دارند. برخی از این تکنیک‌ها شامل موارد زیر هستند:

۱. دینامیک زمانی:(Temporal Dynamics)

- این تکنیک‌ها تغییرات زمانی در داده‌ها را مدل‌سازی می‌کنند. به عنوان مثال، ترجیحات کاربران ممکن است با گذشت زمان تغییر کنند یا محبوبیت آیتم‌ها ممکن است افزایش یا کاهش یابد. مدل‌های دینامیک زمانی این تغییرات را در نظر می‌گیرند و به این ترتیب می‌توانند پیش‌بینی‌های دقیق‌تری انجام دهند.

۲. ماشین‌های بولتزمن محدود:(RBM)

- این مدل‌ها از شبکه‌های عصبی برای مدل‌سازی تعاملات پیچیده بین کاربران و آیتم‌ها استفاده می‌کنند. ماشین‌های بولتزمن محدود به خوبی قادر به کشف الگوهای پنهان در داده‌ها هستند و می‌توانند به عنوان یک لایه اضافی به مدل‌های تجزیه ماتریس اضافه شوند.

۳. مدل‌های ترکیبی:(Hybrid Models)

- مدل‌های ترکیبی از مزایای چندین تکنیک مختلف بهره می‌برند. به عنوان مثال، ترکیب مدل‌های تجزیه ماتریس با مدل‌های همسایگی می‌تواند منجر به بهبود دقت پیش‌بینی‌ها شود. مدل‌های ترکیبی به کارگیری ویژگی‌های مختلف از مدل‌های متفاوت را ممکن می‌سازند و به این ترتیب می‌توانند نقاط ضعف هر مدل را کاهش دهند.

استفاده از ترکیب مدل‌ها یکی از کلیدهای موفقیت در سیستم‌های توصیه‌گر، ترکیب مدل‌های مختلف به منظور بهره‌گیری از نقاط قوت هر کدام است. ترکیب مدل‌ها می‌تواند به روش‌های مختلفی انجام شود، از جمله:

- **ترکیب خطی:(Linear Blending)**
  - در این روش، پیش‌بینی‌های مختلف از مدل‌های متعدد با استفاده از وزن‌های مختلف ترکیب می‌شوند. وزن‌های هر مدل بر اساس دقت آن تنظیم می‌شوند تا ترکیب نهایی بهترین پیش‌بینی‌ها را ارائه دهد.

- **ترکیب غیرخطی:(Non-linear Blending)**
  - این روش شامل استفاده از تکنیک‌های پیشرفته‌تر مانند شبکه‌های عصبی یا ماشین‌های بردار پشتیبان (SVM)برای ترکیب پیش‌بینی‌ها می‌شود. این مدل‌ها قادر به مدل‌سازی تعاملات پیچیده‌تر بین پیش‌بینی‌های مختلف هستند.

مزایای ترکیب مدل‌ها ترکیب مدل‌ها چندین مزیت دارد:

۱. افزایش دقت:

- با ترکیب مدل‌های مختلف، می‌توان دقت پیش‌بینی‌ها را افزایش داد زیرا نقاط قوت هر مدل می‌تواند نقاط ضعف دیگر مدل‌ها را پوشش دهد.

۲. انعطاف‌پذیری بیشتر:

- ترکیب مدل‌ها انعطاف‌پذیری بیشتری در مواجهه با داده‌های مختلف فراهم می‌کند و می‌تواند به راحتی با تغییرات در داده‌ها سازگار شود.

۳. کاهش بیش‌برازش:(Overfitting)

- استفاده از چندین مدل می‌تواند خطر بیش‌برازش را کاهش دهد، زیرا ترکیب مدل‌ها معمولاً به داده‌های آموزش حساسیت کمتری دارد.

چالش‌ها و ملاحظات در کنار مزایای ترکیب مدل‌ها، چالش‌هایی نیز وجود دارد:

- **پیچیدگی محاسباتی:**
  - ترکیب چندین مدل می‌تواند به پیچیدگی محاسباتی بیشتری منجر شود و زمان و منابع بیشتری برای آموزش و پیش‌بینی نیاز داشته باشد.
- **تنظیم وزن‌ها:**
  - تنظیم دقیق وزن‌های ترکیب مدل‌ها برای دستیابی به بهترین نتایج، نیاز به آزمایش‌ها و بهینه‌سازی‌های زیادی دارد.

کاربردهای عملی تکنیک‌های پیشرفته و ترکیب مدل‌ها در بسیاری از سیستم‌های توصیه‌گر پیشرفته مانند نتفلیکس، آمازون و اسپاتیفای به کار گرفته می‌شوند. این سیستم‌ها با استفاده از مدل‌های ترکیبی قادر به ارائه توصیه‌های شخصی‌سازی‌شده و دقیق به کاربران خود هستند که تجربه کاربری بهتری را فراهم می‌کند.

Neural Collaberative Filtering

مقدمه

در عصر اطلاعات، حجم عظیمی از داده‌ها و اطلاعات به سرعت در حال تولید و انتشار است. این افزایش اطلاعات به چالشی جدی برای کاربران تبدیل شده است زیرا یافتن اطلاعات مورد نیاز از میان انبوه داده‌ها بسیار دشوار شده است. سیستم‌های توصیه‌گر (Recommender Systems) به عنوان راه‌حلی برای این مشکل ارائه شده‌اند. این سیستم‌ها با تحلیل رفتار کاربران و ارائه پیشنهادات شخصی‌سازی شده، به کاربران کمک می‌کنند تا اطلاعات مورد نیاز خود را سریع‌تر و دقیق‌تر پیدا کنند.

با وجود موفقیت‌های چشمگیر سیستم‌های توصیه‌گر، هنوز چالش‌های زیادی در این زمینه وجود دارد. یکی از مهم‌ترین چالش‌ها، نیاز به مدل‌سازی دقیق‌تر و پیچیده‌تر تعاملات بین کاربران و آیتم‌ها است. روش‌های سنتی فیلترینگ مشارکتی، مانند فاکتوریزه کردن ماتریس (Matrix Factorization)، به دلیل استفاده از توابع ساده‌ای مانند ضرب داخلی، نمی‌توانند به طور کامل تعاملات پیچیده را مدل‌سازی کنند. این مدل‌های ساده نمی‌توانند تمامی جوانب و الگوهای پیچیده‌ای که در تعاملات بین کاربران و آیتم‌ها وجود دارد را در بر گیرند، بنابراین نیاز به روش‌های پیشرفته‌تری وجود دارد.

یکی دیگر از مهم‌ترین چالش‌ها در سیستم‌های توصیه‌گر، مشکل شروع سرد (Cold-start) است. این مشکل زمانی رخ می‌دهد که سیستم اطلاعات کافی از تعاملات گذشته کاربر ندارد و بنابراین نمی‌تواند پیشنهادات مناسبی ارائه دهد. برای حل این مشکل، مقاله با استفاده از اطلاعات جمعیتی کاربران و تکنیک‌های مبتنی بر اطلاعات مشترک (Mutual Information) برای انتخاب بهترین ویژگی‌ها استفاده می‌کند. این رویکرد، ترکیبی از روش‌های مختلف برای ارائه پیشنهادات بهتر و دقیق‌تر به کاربران جدید را فراهم می‌کند و به کاهش مشکل شروع سرد کمک می‌کند. به کارگیری تکنیک‌های یادگیری ماشین و شبکه‌های عصبی عمیق نیز می‌تواند بهبود قابل توجهی در عملکرد این سیستم‌ها ایجاد کند.

فیلترینگ هیبریدی با استفاده از شبکه عصبی پرسپترون چند لایه (MLP)

مدل پیشنهادی در این مطالعه از ترکیب دو روش فیلترینگ مشارکتی و فیلترینگ مبتنی بر محتوا با استفاده از شبکه عصبی پرسپترون چند لایه (MLP) بهره می‌برد. این ترکیب به مدل امکان می‌دهد تا از مزایای هر دو روش استفاده کند و دقت پیش‌بینی‌ها را بهبود بخشد. شبکه عصبی پرسپترون چند لایه (MLP) یک نوع شبکه عصبی پیش‌خور است که شامل چندین لایه است که هر کدام از آنها تعدادی نرون دارد. این نرون‌ها توابع غیرخطی را بر روی ترکیبی از ورودی‌ها اعمال می‌کنند و به مدل امکان می‌دهند تا ویژگی‌های پنهان پیچیده‌تری را یاد بگیرد.

در مدل پیشنهادی، ابتدا ویژگی‌های پنهان کاربران و آیتم‌ها با استفاده از فیلترینگ مشارکتی و فیلترینگ مبتنی بر محتوا استخراج می‌شوند. سپس این ویژگی‌ها به عنوان ورودی به شبکه عصبی MLP داده می‌شوند. شبکه عصبی MLP با یادگیری ویژگی‌های پنهان پیچیده می‌تواند عملکرد دقیق‌تری در پیش‌بینی تعاملات کاربر-آیتم ارائه دهد. به این ترتیب، مدل قادر است تعاملات پیچیده بین کاربران و آیتم‌ها را بهتر مدل‌سازی کند و دقت توصیه‌ها را بهبود بخشد.

تکنیک‌های اطلاعات مشترک (Mutual Information) نیز در این مدل برای انتخاب بهترین ویژگی‌ها برای مدل‌سازی استفاده می‌شوند. این تکنیک‌ها به مدل کمک می‌کنند تا ویژگی‌های مهم‌تر و تأثیرگذارتر را شناسایی کرده و از آنها برای بهبود دقت پیش‌بینی‌ها استفاده کند. به کارگیری شبکه عصبی پرسپترون چند لایه در ترکیب با فیلترینگ مشارکتی و مبتنی بر محتوا، یک رویکرد قدرتمند و موثر برای بهبود سیستم‌های توصیه‌گر ارائه می‌دهد.

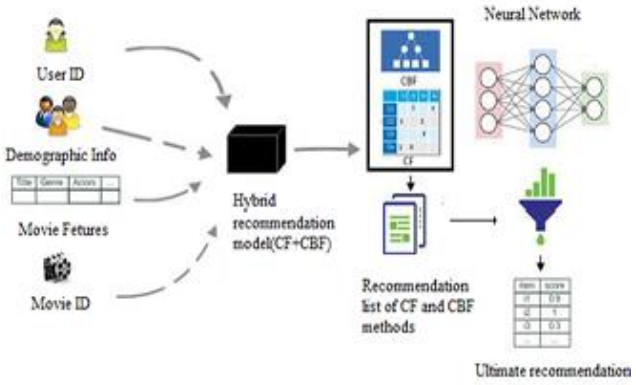


Fig. 1. Research methodology

شکل ۰ : فرآیند کلی مدل پیشنهادی برای سیستم توصیه‌گر هیبریدی

شکل ۰، فرآیند کلی مدل پیشنهادی برای سیستم توصیه‌گر هیبریدی را نشان می‌دهد. این مدل ترکیبی از روش‌های فیلترینگ مشارکتی (Collaborative Filtering - CF) و فیلترینگ مبتنی بر محتوا (Content-Based Filtering - CBF) با استفاده از شبکه عصبی پرسپترون چند لایه (MLP) است.

۱. ورودی‌های اولیه:

- User ID (شناسه کاربر): شناسه‌های منحصر به فردی که هر کاربر را شناسایی می‌کنند.
- Demographic Info (اطلاعات جمعیتی): اطلاعاتی مانند سن، جنسیت و موقعیت جغرافیایی کاربران که می‌تواند به بهبود دقت توصیه‌ها کمک کند.
- Movie Features (ویژگی‌های فیلم): ویژگی‌های محتوایی فیلم‌ها مانند ژانر، کارگردان و بازیگران.

- Movie ID (شناسه فیلم): شناسه‌های منحصر به فردی که هر فیلم را شناسایی می‌کنند.

۲. مدل توصیه‌گر هیبریدی:

- داده‌های ورودی اولیه به یک مدل توصیه‌گر هیبریدی که ترکیبی از فیلترینگ مشارکتی و فیلترینگ مبتنی بر محتوا است، تغذیه می‌شوند. این مدل با تحلیل شباهت‌ها و تعاملات بین کاربران و آیتم‌ها، لیستی از توصیه‌ها را تولید می‌کند.

۳. لیست توصیه‌ها:

مدل هیبریدی لیستی از آیتم‌های پیشنهادی را بر اساس روش‌های CF و CBF تولید می‌کند. این لیست شامل آیتم‌هایی است که بیشترین شباهت را به ترجیحات کاربر دارند.

۴. شبکه عصبی:

- لیست توصیه‌های تولید شده به یک شبکه عصبی پرسپترون چند لایه (MLP) داده می‌شود. این شبکه عصبی با استفاده از ویژگی‌های پنهان پیچیده، دقت پیش‌بینی‌ها را بهبود می‌بخشد و توصیه‌های نهایی را تولید می‌کند.

۵. توصیه نهایی:

- خروجی نهایی شبکه عصبی شامل لیستی از توصیه‌های بهینه برای هر کاربر است. این لیست بر اساس تحلیل‌های پیشرفته و یادگیری ویژگی‌های پنهان پیچیده، دقیق‌ترین پیشنهادات ممکن را ارائه می‌دهد.

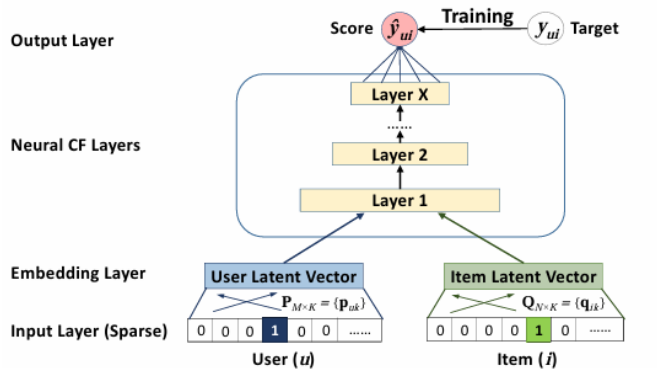
این شکل به وضوح نشان می‌دهد که چگونه مدل پیشنهادی با ترکیب اطلاعات مختلف و استفاده از تکنیک‌های پیشرفته یادگیری ماشین، توصیه‌های بهینه‌ای را به کاربران ارائه می‌دهد. این رویکرد هیبریدی با بهره‌گیری از مزایای هر دو روش CF و CBF و به کارگیری شبکه عصبی MLP، می‌تواند دقت و کارایی سیستم‌های توصیه‌گر را به طور قابل توجهی بهبود بخشد.

فیلترینگ مشارکتی عصبی (NCF)

در سال‌های اخیر، شبکه‌های عصبی عمیق (Deep Neural Networks) در حوزه‌های مختلفی مانند تشخیص گفتار، بینایی کامپیوتری و پردازش زبان طبیعی موفقیت‌های بزرگی کسب کرده‌اند. با این حال، استفاده از این شبکه‌ها در سیستم‌های توصیه‌گر به

نسبت کمتر مورد توجه قرار گرفته است. فیلترینگ مشارکتی عصبی (Neural Collaborative Filtering) روشی است که از قدرت شبکه‌های عصبی برای مدل‌سازی تعاملات پیچیده بین کاربران و آیتم‌ها استفاده می‌کند.

فیلترینگ مشارکتی عصبی یک چارچوب عمومی است که می‌تواند به شکل‌های مختلفی پیاده‌سازی شود. این چارچوب شامل مدل‌های مختلفی مانند GMF، MLP و NeuMF است که هر کدام به نحوی به بهبود عملکرد سیستم‌های توصیه‌گر کمک می‌کنند.



شکل ۱ : ساختار کلی مدل فیلترینگ مشارکتی عصبی

شکل ۱ به خوبی نشان‌دهنده ساختار کلی مدل NCF و نحوه مدل‌سازی تعاملات پیچیده بین کاربران و آیتم‌ها با استفاده از شبکه‌های عصبی عمیق است. بخش‌های مختلف مدل از جمله لایه ورودی، لایه جاسازی، لایه‌های فیلترینگ عصبی، و لایه خروجی است. در ادامه، هر یک از این بخش‌ها به طور مختصر توضیح داده شده‌اند:

لایه ورودی که شامل دو بردار پراکنده (Sparse Vectors) است که هر یک از آنها نمایانگر یک کاربر و یک آیتم هستند. این بردارها به صورت کدگذاری یک-گرمی (One-Hot Encoding) نشان داده می‌شوند. برای مثال، بردار کاربر U شامل چندین صفر و یک است که مقدار یک نشان‌دهنده وجود کاربر در موقعیت خاصی از بردار است.

بخش بعد لایه جاسازی (Embedding Layer) است که بردارهای پراکنده کاربران و آیتم‌ها به بردارهای متراکم پنهان تبدیل می‌شوند. این بردارهای متراکم، ویژگی‌های پنهان کاربران و آیتم‌ها را نمایش می‌دهند.

در اد امه لایه‌های فیلترینگ عصبی (Neural CF Layers) قرار دارد که شامل چندین لایه مخفی هستند که با استفاده از توابع فعال‌سازی، تعاملات پیچیده بین بردارهای جاسازی شده کاربران و آیتم‌ها را مدل‌سازی می‌کنند. این لایه‌ها به صورت لایه ۱، لایه ۲ تا لایه X در تصویر نشان داده شده‌اند. هر لایه یک تابع غیرخطی را بر روی ورودی‌های خود اعمال می‌کند تا ویژگی‌های پیچیده‌تری را استخراج کند.

و در نهایت لایه خروجی شامل یک نورون است که امتیاز پیش‌بینی شده برای تعامل بین کاربر و آیتم را محاسبه می‌کند. این نورون از خروجی‌های لایه‌های قبلی برای محاسبه امتیاز نهایی استفاده می‌کند. این امتیاز پیش‌بینی شده در طی فرآیند آموزش با امتیاز واقعی مقایسه می‌شود تا خطای مدل محاسبه و بهینه‌سازی شود.

در طی فرآیند آموزش، امتیاز پیش‌بینی شده با امتیاز هدف مقایسه می‌شود و مدل با استفاده از الگوریتم‌های بهینه‌سازی (مانند گرادیان نزولی) بهینه‌سازی می‌شود. هدف از فرآیند آموزش کاهش خطای بین امتیاز پیش‌بینی شده و امتیاز واقعی است تا مدل بتواند تعاملات بین کاربران و آیتم‌ها را با دقت بیشتری پیش‌بینی کند.

فاکتوریزه کردن ماتریس عمومی شده (GMF)

مدل (Generalized Matrix Factorization) GMF یکی از مدل‌های اصلی در چارچوب NCF است. این مدل با استفاده از محصول عنصری (Element-wise Product) به جای ضرب داخلی، می‌تواند تعاملات پیچیده‌تری را بین کاربران و آیتم‌ها مدل‌سازی کند. در این مدل، هر کاربر و آیتم با یک بردار ویژگی پنهان (Latent Feature Vector) نمایش داده می‌شوند. فرمول این مدل به صورت زیر است:

$$\hat{y}_{ui} = f(P^T v_u^T, Q^T v_i^I \mid P, Q, \Theta_f)$$

که در آن P و Q به ترتیب بردارهای ویژگی پنهان کاربر (U) و آیتم (I) هستند و Θ_f نشان‌دهنده محصول عنصری است. تابع f به عنوان تابع فعال‌سازی استفاده می‌شود.

پرسپترون چند لایه (MLP)

مدل پرسپترون چند لایه (Multi-Layer Perceptron) MLP با استفاده از لایه‌های مخفی متعدد و توابع غیرخطی مانند ReLU (Rectified Linear Unit)، می‌تواند تعاملات پیچیده بین کاربران و آیتم‌ها را بهتر مدل‌سازی کند. این مدل با استفاده از شبکه‌های عصبی عمیق، قادر به یادگیری ویژگی‌های غیرخطی پیچیده‌تری است که در روش‌های سنتی ممکن نیست. این مدل یکی از مهم‌ترین مدل‌های شبکه عصبی برای یادگیری تعاملات پیچیده بین کاربران و آیتم‌ها است. پرسپترون چند لایه یک نوع شبکه عصبی پیش‌خور (Feedforward Neural Network) است که از چندین لایه تشکیل شده است. هر لایه شامل تعدادی نورون است که هر کدام یک تابع غیرخطی را بر روی ترکیبی از ورودی‌ها اعمال می‌کند.

ساختار MLP شامل سه نوع لایه اصلی است:

۱. **لایه ورودی (Input Layer):** این لایه شامل نورون‌هایی است که ورودی‌های خام را دریافت می‌کنند. در مدل‌های توصیه‌گر، این ورودی‌ها می‌توانند شامل ویژگی‌های کاربر و آیتم باشند.

۲. **لایه‌های مخفی (Hidden Layers):** این لایه‌ها شامل نورون‌هایی هستند که هر کدام یک تابع غیرخطی (مانند ReLU) را بر روی ترکیبی از ورودی‌ها اعمال می‌کنند. هر لایه مخفی به صورت زیر تعریف می‌شود:

$$\phi_i(z_i) = ReLU(W_i^T z_{i-1} + b_i)$$

که در آن W ماتریس وزن‌ها، b بردار بایاس و z_{i-1} ورودی‌های لایه قبلی است.

۳. **لایه خروجی (Output Layer):** این لایه شامل نورون‌هایی است که خروجی نهایی مدل را تولید می‌کنند. در مدل‌های توصیه‌گر، خروجی می‌تواند یک امتیاز پیش‌بینی شده برای تعامل بین کاربر و آیتم باشد.

آموزش MLP شامل بهینه‌سازی پارامترهای مدل (وزن‌ها و بایاس‌ها) با استفاده از داده‌های آموزشی است. روش‌های مختلفی برای آموزش MLP وجود دارد که یکی از معروف‌ترین آنها روش پس‌انتشار خطا (Backpropagation) است. در این روش، خطای پیش‌بینی محاسبه شده و با استفاده از مشتقات زنجیره‌ای، گرادیان خطا نسبت به پارامترهای مدل محاسبه می‌شود. سپس با استفاده از یک الگوریتم بهینه‌سازی (مانند گرادیان نزولی) پارامترها به‌روزرسانی می‌شوند تا خطا کمینه شود.

ترکیب GMF و MLP – (NeuMF)

مدل NeuMF (Neural Matrix Factorization) با ترکیب GMF و MLP، مدلی قوی‌تر و کامل‌تر ارائه می‌دهد که می‌تواند هم از خطی بودن GMF و هم از غیرخطی بودن MLP بهره‌برداری کند. این مدل با استفاده از بردارهای جاسازی جداگانه و ترکیب خروجی‌های لایه‌های مخفی، عملکرد بهتری را نشان می‌دهد. فرمول کلی این مدل به صورت زیر است:

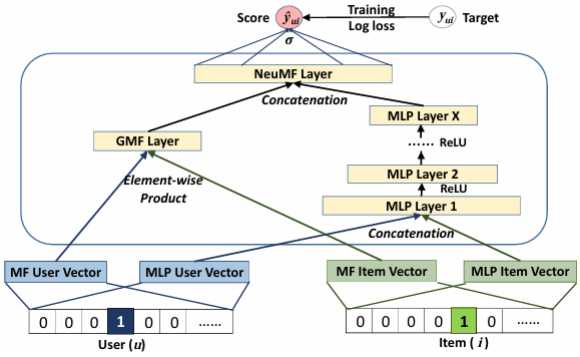
$$\phi^{GMF} = p_u^G \odot q_i^G$$

$$\phi^{MLP} = a_L(W_L^T(a_{L-1}\left(\dots\ a_2(W_L^T\begin{bmatrix}p_u^M\\q_i^M\end{bmatrix}+b_2\right)\dots)) + b_L)$$

$$\hat{y}_{ui} = \sigma(h^T\begin{bmatrix}\phi^{GMF}\\\phi^{MLP}\end{bmatrix})$$

بردارهای ϕ^{GMF} و ϕ^{MLP} با هم ترکیب می‌شوند تا یک بردار ویژگی جامع تشکیل شود و این ترکیب باعث می‌شود که مدل بتواند از هر دو نوع ویژگی خطی و غیرخطی بهره‌برداری کند.

بردار ترکیبی حاصل به یک لایه خروجی متصل می‌شود که با استفاده از وزن‌های (h) و تابع سیگموئید امتیاز پیش‌بینی شده را محاسبه کند. تابع سیگموئید به عنوان تابع فعال‌سازی استفاده می‌شود تا مقدار خروجی در بازه (۰,۱) قرار گیرد که نشان‌دهنده احتمال تعامل مثبت بین کاربر و آیتم است. این فرمول نشان‌دهنده قدرت مدل NeuMF در ترکیب و استفاده از ویژگی‌های مختلف برای پیش‌بینی دقیق‌تر تعاملات بین کاربران و آیتم‌ها است. با این ترکیب، مدل می‌تواند تعاملات پیچیده‌تری را نسبت به مدل‌های سنتی فاکتوریزه کردن ماتریس مدل‌سازی کند. برای درک بهتر این فرآیند ها میتوان به شکل ۲ نیز توجه کرد.



شکل ۲ : ساختار NeuMF

یادگیری NCF

برای یادگیری مدل‌های NCF، از روش بهینه‌سازی log loss با نمونه‌برداری منفی استفاده می‌شود. این روش با در نظر گرفتن ماهیت باینری داده‌های ضمنی، به بهبود عملکرد مدل‌های NCF کمک می‌کند. تابع زیان به صورت زیر تعریف می‌شود:

$$L = -\sum_{(u,i)\in Y} \log(\hat{y}_{ui})\,\hat{y}_{ui} - \sum_{(u,j)\in Y^-} \log(1-\hat{y}_{ui})\,\hat{y}_{uj}$$

که در آن Y مجموعه تعاملات مشاهده شده و Y^- مجموعه نمونه‌های منفی است.

آزمایش‌ها

برای ارزیابی عملکرد مدل‌های NCF، از دو مجموعه داده عمومی MovieLens و Pinterest استفاده شد. از پروتکل ارزیابی Leave-One-Out برای ارزیابی عملکرد توصیه آیتم‌ها استفاده شد. معیارهای ارزیابی شامل Hit Ratio (HR) و Normalized Discounted Cumulative Gain (NDCG) بودند.

پروتکل ارزیابی Leave-One-Out یکی از روش‌های رایج برای ارزیابی سیستم‌های توصیه‌گر است. در این روش، یک تعامل خاص از هر کاربر به عنوان نمونه آزمایشی جدا می‌شود و مدل با استفاده از بقیه داده‌ها آموزش می‌بیند. سپس مدل آموزش دیده برای پیش‌بینی تعامل جدا شده استفاده می‌شود. این پروتکل ارزیابی دقت مدل را در پیش‌بینی تعاملات جدید کاربران اندازه‌گیری می‌کند.

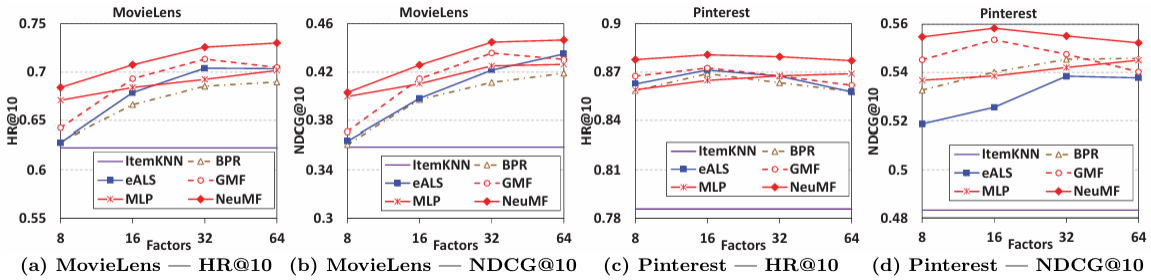
معیارهای ارزیابی:

۱. Hit Ratio (HR): این معیار نشان می‌دهد که آیا آیتم توصیه شده در میان لیست برترین پیشنهادات به کاربر قرار دارد یا خیر

۲. Normalized Discounted Cumulative Gain (NDCG): این معیار دقت مدل را بر اساس موقعیت آیتم توصیه شده در لیست برترین پیشنهادات اندازه‌گیری می‌کند. این معیار به صورت زیر محاسبه می‌شود:

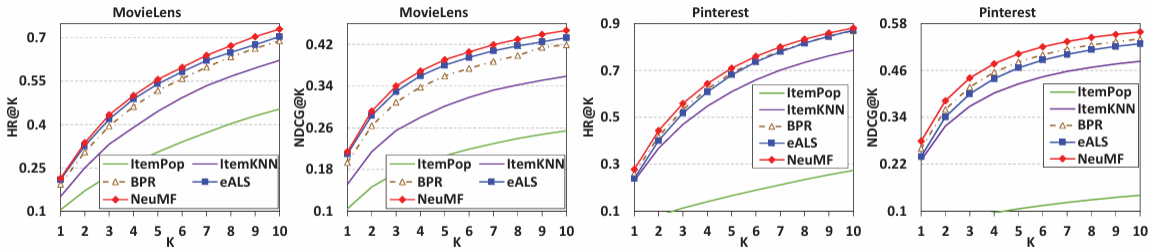
نتایج آزمایش‌ها

نتایج نشان داد که مدل NeuMF با ترکیب GMF و MLP بهترین عملکرد را در بین روش‌های مختلف داشت و بهبودهای قابل توجهی را نسبت به روش‌های پیشرفته موجود مانند BPR و eALS نشان داد. همچنین، مدل‌های NCF با افزایش تعداد نمونه‌های منفی، عملکرد بهتری داشتند و استفاده از لایه‌های عمیق‌تر در مدل MLP عملکرد را بهبود بخشید. که در شکل ۴ و ۵ قابل مشاهده است.



(a) MovieLens — HR@10 (b) MovieLens — NDCG@10 (c) Pinterest — HR@10 (d) Pinterest — NDCG@10

شکل ۴: عملکرد HR@۱۰ و NDCG@۱۰ نسبت به تعداد عوامل پیش‌بینی در دو مجموعه داده.



(a) MovieLens — HR@K (b) MovieLens — NDCG@K (c) Pinterest — HR@K (d) Pinterest — NDCG@K

شکل ۵: ارزیابی توصیه آیتم‌های Top-K که K از ۱ تا ۱۰ در دو مجموعه داده متفاوت است.

نتایج نشان می‌دهند که مدل NeuMF که ترکیبی از GMF و MLP است، در هر دو مجموعه داده MovieLens و Pinterest عملکرد بهتری نسبت به سایر مدل‌های مورد مقایسه دارد. این امر نشان‌دهنده قدرت مدل NeuMF در ترکیب ویژگی‌های خطی و غیرخطی برای مدل‌سازی تعاملات پیچیده بین کاربران و آیتم‌ها است. همچنین، با افزایش تعداد عوامل پیش‌بینی و تعداد آیتم‌های توصیه شده، عملکرد مدل NeuMF به طور پیوسته بهبود می‌یابد که نشان‌دهنده انعطاف‌پذیری و کارایی بالای این مدل در شرایط مختلف است.

یکی از موارد مهم در بهبود عملکرد مدل‌های NCF، تعداد نمونه‌های منفی است. نتایج نشان داد که با افزایش تعداد نمونه‌های منفی، عملکرد مدل بهبود می‌یابد. این امر نشان می‌دهد که استفاده از تعداد کافی نمونه‌های منفی می‌تواند به مدل کمک کند تا تعاملات منفی را بهتر شناسایی کند و دقت پیش‌بینی‌ها را افزایش دهد.

نتیجه‌گیری و کارهای آینده

نتایج نشان می‌دهند که مدل NeuMF، که ترکیبی از دو روش GMF و MLP است، توانسته است به طور قابل توجهی عملکرد بهتری را نسبت به مدل‌های سنتی فیلترینگ مشارکتی مانند BPR، eALS و ItemKNN ارائه دهد. این بهبودها در هر دو معیار HR@۱۰ و NDCG@۱۰ و همچنین در ارزیابی توصیه آیتم‌های Top-K برای دو مجموعه داده MovieLens و Pinterest مشهود است. این نتایج تایید می‌کنند که مدل NeuMF می‌تواند به طور موثری از ویژگی‌های خطی و غیرخطی برای مدل‌سازی تعاملات پیچیده بین کاربران و آیتم‌ها استفاده کند.

یکی از مزایای اصلی مدل NeuMF، توانایی آن در ترکیب ویژگی‌های پنهان کاربران و آیتم‌ها از طریق ساختارهای مختلف شبکه‌های عصبی است. این مدل با استفاده از بردارهای جاسازی جداگانه برای هر یک از مدل‌های GMF و MLP و ترکیب خروجی‌های آنها، قادر است تا ویژگی‌های غنی‌تری را استخراج و تعاملات غیرخطی پیچیده‌تری را مدل‌سازی کند. این امر به خصوص در محیط‌هایی که تعاملات کاربر-آیتم پیچیده‌تر هستند، باعث بهبود دقت پیش‌بینی‌ها و افزایش کارایی سیستم توصیه‌گر می‌شود.

علاوه بر این، نتایج آزمایش‌ها نشان دادند که مدل NeuMF با افزایش تعداد عوامل پیش‌بینی و تعداد آیتم‌های توصیه شده، عملکرد خود را بهبود می‌بخشد. این انعطاف‌پذیری و توانایی مدل در تنظیم بهتر پارامترها، آن را به یک انتخاب مناسب برای کاربردهای مختلف در زمینه‌های مختلف مانند تجارت الکترونیک، شبکه‌های اجتماعی و سیستم‌های توصیه‌گر محتوا تبدیل می‌کند. به طور کلی، مدل NeuMF با بهره‌گیری از قدرت شبکه‌های عصبی عمیق و ترکیب روش‌های مختلف فیلترینگ مشارکتی، یک پیشرفت قابل توجه در زمینه سیستم‌های توصیه‌گر ارائه داده است که می‌تواند در آینده نیز به توسعه و بهبود این سیستم‌ها کمک کند.

کارهای آینده

۱. توسعه یادگیری زوجی (Pairwise Learning): یکی از حوزه‌های تحقیقاتی جذاب در آینده، توسعه مدل‌های NCF برای یادگیری زوجی است. این روش می‌تواند به بهبود عملکرد مدل‌ها در پیش‌بینی رتبه‌بندی آیتم‌ها کمک کند.

۲. مدل‌سازی اطلاعات کمکی: استفاده از اطلاعات کمکی مانند بررسی‌های کاربران، پایگاه‌های دانش و سیگنال‌های زمانی می‌تواند به بهبود دقت و کارایی مدل‌های NCF کمک کند.

۳. کاربردهای عملی: بررسی کاربردهای عملی مدل‌های NCF در حوزه‌های مختلف مانند تجارت الکترونیک، اخبار آنلاین و شبکه‌های اجتماعی می‌تواند به بهبود تجربه کاربران و افزایش کارایی سیستم‌های توصیه‌گر کمک کند.

۴. تحقیق در زمینه بهینه‌سازی: بهینه‌سازی روش‌های آموزش مدل‌های NCF و بررسی تاثیر پارامترهای مختلف مانند تعداد لایه‌ها، نوع توابع فعال‌سازی و تعداد نمونه‌های منفی می‌تواند به بهبود عملکرد این مدل‌ها کمک کند.

A Compact Report of two important papers

این گزارش، گزارشی مختصر در مورد دو منبع ذکرشده در ذیل است:

- Recommendation System Based on Collaborative Filtering
From: Zheng Wen
- Recommendation System for Netflix
Author: Leidy Esperanza MOLINA – FERNÁNDEZ

برای جلوگیری از تکرار اسم منابع از شماره منبع آنها استفاده میشود.

هردو منبع با مقدمه‌ای بر Recommender System شروع میشوند. منبع اول فقط در مورد Collaborative Filtering و منبع دوم هم در مورد Collaborative Filtering و هم Content based Filtering میباشند.
منبع اول با موضوعی تحت عنوان Problem Formulation آغاز میشود. فرض میکنیم که به تعداد N_u تا کاربر و N_m تا فیلم داریم و همچنین ماتریس A که mسطر و u ستون دارد را با نام user-movie matrix تشکیل میدهیم. اگر کاربر u فیلم m را دیده بود و به آن امتیاز داده بود آن امتیاز در خانه‌ی A_{mu} در ماتریس A قرار میگیرد و سایر خانه ها که امتیاز ندارند ۰ میشنوند.

$$A_{mu} = \begin{cases} R_{m,u}, \text{ User } u\text{'s rating on Movie } m & \text{if such rating exists} \\ ? & \text{if no such rating} \end{cases}$$

امتیاز کاربرها به فیلم ها عددی بین ۱ تا ۵ میباشد و عدد ۰ نشان دهنده rate نشد آن فیلم توسط کاربر است.
در هر دو منبع معیاری به نام RMSE(Root Mean Square Error) تعریف شده که برای سنجش کارایی RS به کار میرود و هر منبع از یک روش آن را محاسبه کرده:

$$rmse = \sqrt{\frac{1}{|S_{\text{test}}|} \sum_{(m,u) \in S_{\text{test}}} (R_{m,u} - P_{m,u})^2} \qquad RMSE = \sqrt{\frac{1}{N} \sum (x_i - \hat{x}_i)^2}$$

در روش منبع اول اگر کاربر u به فیلم m امتیاز داده باشد، دوتایی (m,u) عضو مجموعه‌ی S میشوند. |S| همان کاردینالیتی S یا تعداد عضو های مجموعه S میباشد. $R_{m,u}$ امتیاز واقعی و $P_{m,u}$ امتیاز پیشبینی شده میباشند. روش منبع دوم نیز مشابه میباشد.
روش دیگری برای سنجش کارایی RS در منبع دوم به نام MAE(Means Absolute Error) آورده شده که میانگین اشتباهات را حساب میکند:

$$MAE = \frac{1}{n} \sum_{j=1}^{n} |y_j - \hat{y}_j|$$

Collaborative Filtering Algorithms

Item-Based K Nearest Neighbor (KNN) Algorithm

در این روش امتیاز کاربر u به فیلم m را می‌خواهیم پیشبینی کنیم. ای پیشبینی بر اساس امتیاز این کاربر به فیلم های مشابه به فیلم m میباشد. برای این کار یک تابع شباهت یا similarity function تعریف میکنیم:

$$sim(a,b) = \frac{\sum_{u \in U(a) \bigcap U(b)} (R_{a,u} - \bar{R}_u)(R_{b,u} - \bar{R}_u)}{\sqrt{\sum_{u \in U(a) \bigcap U(b)} (R_{a,u} - \bar{R}_u)^2 \sum_{u \in U(a) \bigcap U(b)} (R_{b,u} - \bar{R}_u)^2}},$$

سپس به صورت وزن دار پیشبینی امتیاز را انجام میدهیم:

$$P_{m,u} = \frac{\sum_{j \in N_u^K(m)} sim(m,j) R_{j,u}}{\sum_{j \in N_u^K(m)} |sim(m,j)|},$$

$P_{m,u}$ همان پیشبینی میباشد.

Item-Based EM Algorithm

فیلم ها را به G گروه تقسیم میکنیم و هر فیلم m با احتمال $Q_m(g)$ به گروه g تعلق دارد. فرض بر این است که امتیاز های کاربران برای هر گروه g مستقل از یکدیگر میباشند. اگر U(m) مجموعه کاربرانی که به فیلم m امتیاز داده‌اند و M(u) مجموعه فیلم‌هایی که کاربر u به آنها امتیاز داده باشد:

$$\begin{aligned} Q_m^{(t+1)}(g) &= P(G_m^{(t+1)} = g | R_{u,m}; \mu_{g,u}, \sigma_{g,u}^2) \\ &= \frac{Q_m^{(t)}(g) \Pi_{u \in U(m)} \frac{1}{\sqrt{2\pi}\sigma_{g,u}} exp\left(-\frac{(R_{u,m} - \mu_{g,u})^2}{2\sigma_{g,u}^2}\right)}{\sum_{g'} Q_m^{(t)}(g') \Pi_{u \in U(m)} \frac{1}{\sqrt{2\pi}\sigma_{g',u}} exp\left(-\frac{(R_{u,m} - \mu_{g',u})^2}{2\sigma_{g',u}^2}\right)}, \end{aligned}$$

برای حل معادله E-step نیاز به حل معادله M-step زیر داریم:

$$\max_{\mu_{g,u},\sigma^2_{g,u}} \quad \sum_m \sum_g Q_m(g) \left[log \left\{ \Pi_{u \in U(m)} \frac{1}{\sqrt{2\pi}\sigma_{g,u}} exp \left(-\frac{(R_{u,m} - \mu_{g,u})^2}{2\sigma^2_{g,u}} \right) \right\} - log\left(Q_m(g)\right) \right],$$

که میشود:

$$\max_{\mu_{g,u},\sigma^2_{g,u}} \quad \sum_{m,g} Q_m(g) \sum_{u \in U(m)} \left[-log(\sigma_{g,u}) - \frac{(R_{u,m} - \mu_{g,u})^2}{2\sigma^2_{g,u}} \right].$$

با کمی تغییرات:

$$\mu_{g,u} = \frac{\sum_{m \in M(u)} Q_m(g) R_{u,m}}{\sum_{m \in M(u)} Q_m(g)} \qquad \sigma^2_{g,u} = \frac{\sum_{m \in M(u)} Q_m(g) \left(R_{u,m} - \mu_{g,u}\right)^2}{\sum_{m \in M(u)} Q_m(g)}$$

E-step و M-step را آنقدر تکرار میکنیم تا همگرا شوند و در آخر پیشبینی نهایی از فرمول زیر استفاده میکنیم:

$$P_{u,m} = \sum_g Q_m(g) \mu_{g,u}.$$

در منبع دوم از روش Correlation-based Similarity استفاده میشود. اگر مجموعه U شامل کاربرهایی که به هردو فیلم i و j امتیاز داده اند باشند:

$$S(j,l) = corr_{jl} = \frac{\sum_{i \in U} (v_{ij} - \bar{v}_j)(v_{il} - \bar{v}_l)}{\sqrt{\sum_{i \in U} (v_{ij} - \bar{v}_j)^2} \sqrt{\sum_{i \in U} (v_{il} - \bar{v}_l)^2}}$$

که v_{ij} همان امتیاز کاربر u_i در مجموعه U به فیلم p_j و v_j برابر میانگین امتیازات j امین فیلم میباشد.

در صورتی که رتبه بندی کاربران دارای مقیاس متفاوتی باشد، می توانیم از cosine similarity استفاده کنیم که در آن میانگین امتیاز کاربر از هر جفت رتبه‌بندی شده کم می‌شود:

$$S(j,l) = \frac{\sum_{i \in U} (v_{ij} - \bar{v}_i)(v_{il} - \bar{v}_i)}{\sqrt{\sum_{i \in U} (v_{ij} - \bar{v}_i)^2} \sqrt{\sum_{i \in U} (v_{il} - \bar{v}_i)^2}}$$

Model-based techniques:

در این روش ها معمولا data mining و machine learning به کار میرود و بر مبنای matrix-factorization میباشد.

Principal Component Analysis (PCA)

یک تکنیک ریاضی است که در تجزیه و تحلیل مؤلفه‌های اصلی RS برای کاهش بعد داده‌ها و بهبود فرآیند توصیه‌دهی استفاده می‌شود. در این روش سعی میشود همبستگی‌ها و ویژگی های مشترک بین مجموعه داده ها را پیدا کنیم. وقتی همبستگی قوی وجود دارد، PCA با انتقال داده به یک subspace دیگر با بعد کمتر، dimensionality را کاهش میدهد. این کار باعث افزایش بهره‌وری در محاسبات میشود همچنین میتواند با Cold Start Problem هم مقابله کند.

Probabilistic Matrix Factorization (PMF)

در این روش، ماتریس V، از ضرب دو ماتریس دیگر به وجود می‌آید؛ یکی برای کاربران و یکی برای آیتم‌ها.

$$p(V|U,V,\sigma^2) = \prod_{i=1}^{n}\prod_{j=1}^{m}[\eta(V_{ij}|U_i^T P_j\sigma^2)]^{I_{ij}}$$

$$p(U|\sigma^2) = \prod_{i=1}^{n}\eta(U_i|0,\sigma_U^2 I)$$

$$p(P|\sigma^2) = \prod_{j=1}^{m}\eta(V_j|0,\sigma_P^2 I)$$

Algorithm Sparse SVD

در این روش، هر کاربر و هر فیلم یک feature vector یا بردار ویژگی دارند. هر rating چه مشخص باشد چه نامشخص، از ضرب داخلی این دو بردار به دست می‌آید. اگر N_u شامل همه‌ی بردار های کاربران باشد و n_{ui} تعداد فیلم‌هایی که کاربر i به آنها امتیاز داده و n_{mj} تعداد کاربرانی که به فیلم j امتیاز داده‌اند باشد داریم:

$$\min_{\mathbf{u_i},\mathbf{m_j}} \quad \sum_{(i,j)\in I}(R_{i,j} - \mathbf{u_i}^T\mathbf{m_j})^2 + \lambda\left(\sum_i n_{u_i}\|\mathbf{u_i}\|^2 + \sum_j n_{m_j}\|\mathbf{m_j}\|^2\right),$$

در منبع دوم از روش SVD به شکل زیر استفاده کرده:

$X = U * S * V^t$

در اینجا U همان بردار ویژگی برای کاربران و V برای آیتم‌ها (فیلم‌ها) میباشد.

$$X_{n\times m} = U_{n\times r} \times S_{r\times r} \times V^t_{r\times m}$$

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & & & \\ \vdots & & \ddots & \vdots \\ x_{n1} & & \dots & x_{nm} \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1r} \\ u_{21} & & & \\ \vdots & & \ddots & \vdots \\ u_{n1} & & \dots & u_{nr} \end{bmatrix} \begin{bmatrix} s_{11} & 0 & \dots & 0 \\ & s_{21} & & \\ \vdots & & \ddots & \vdots \\ 0 & & \dots & s_{rr} \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & & & \\ \vdots & & \ddots & \vdots \\ v_{r1} & & \dots & v_{rm} \end{bmatrix}$$

مراجع

[۲۱] A. V. D. Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *NIPS*, pages ۲۶۴۳–۲۶۵۱, ۲۰۱۳.

[۲۲] S. Purushotham, Y. Liu, and C.-C. J. Kuo. Collaborative topic regression with social matrix factorization for recommendation systems. In *ICML*, pages ۷۵۹–۷۶۶, ۲۰۱۲.

[۳۴] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *KDD*, pages ۴۴۸–۴۵۶, ۲۰۱۱.

[۳۵] H. Wang, B. Chen, and W.-J. Li. Collaborative topic regression with social regularization for tag recommendation. In *IJCAI*, pages ۲۷۱۹–۲۷۲۵, ۲۰۱۳.

[۳۶] H. Wang and W. Li. Relational collaborative topic regression for recommender systems. *TKDE*, ۲۷(۵):۱۳۴۳–۱۳۵۵, ۲۰۱۵.

[۴] B. N. Miller, J. A. Konstan, and J. Riedl, "PocketLens: toward a personal recommender system," *ACM Transactions on Information Systems*, vol. ۲۲, no. ۳, pp. ۴۳۷–۴۷۶, ۲۰۰۴.

[۲۱] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the- art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. ۱۷, no. ۶, pp. ۷۳۴–۷۴۹, ۲۰۰۵.

[۲۲] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel, "Probabilistic memory-based collaborative filtering," *IEEE Transactions on Knowledge and Data Engineering*, vol. ۱۶, no. ۱, pp. ۵۶–۶۹, ۲۰۰۴.

[۲۴] T. Landauer, M. Littman, and Bell Communications Research (Bellcore), “Computerized cross-language docu- ment retrieval using latent semantic indexing,” US patent no. ۵۳۰۱۱۰۹, April ۱۹۹۴.

[۲۵] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *Journal of the American Society for Information Science*, vol. ۴۱, no. ۶, pp. ۳۹۱–۴۰۷, ۱۹۹۰.

[۵۲] J. Canny, “Collaborative filtering with privacy via factor analysis,” in *Proceedings of the ۲۵th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. ۲۳۸–۲۴۵, Tampere, Finland, August ۲۰۰۲.

[۵۷] Y. Koren, “Tutorial on recent progress in collaborative filtering,” in *Proceedings of the the ۲nd ACM Conference on Recommender Systems*, ۲۰۰۸.

[۶۰] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Transactions on Information Systems*, vol. ۲۲, no. ۱, pp. ۵–۵۳, ۲۰۰۴.

[۱] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in Proceedings of the ۲۶th International Conference on World Wide Web*, pp. ۱۷۳-۱۸۲, ۲۰۱۷.

[۲] D. Divani Sanandaj and S. H. Alizadeh, "A hybrid recommender system using Multi Layer Perceptron neural network," in *Proceedings of the ۲۰۱۸ Artificial Intelligence and Robotics (IRANOPEN)*, pp. ۱-۱۲, ۲۰۱۸.