فرهنگ و اندیشهٔ ریاضی

سال ۴۱، شمارهٔ ۷۰ (بهار و تابستان ۱۴۰۱) صص. ۹۷ تا ۱۱۲

کاربرد تجزیه‌های ماتریسی در سامانه‌های پیشنهادگر

عفت گلپر رابوکی، معصومهٔ کیاهی

چکیده. با توجه به انبوه اطلاعات در شبکهٔ جهانی اینترنت، وجود سامانه‌های پیشنهادگر که کالاها را به‌شکل خودکار و هوشمندانه به کاربران پیشنهاد دهند، کاملاً ضروری به نظر می‌رسد. یکی از چالش‌های مطرح در این نوع سامانه‌ها، تُنکی ماتریس کاربر–کالا است که باعث می‌شود سامانه نتواند پیشنهاد مناسبی به کاربر ارائه دهد و درنتیجه کارایی آن کاهش پیدا می‌کند. الگوریتم پیشنهادی ما برای رفع این مشکل، ترکیب پالایش مشارکتی مبتنی‌بر حافظه و پالایش مشارکتی مبتنی‌بر مدل است. برای این منظور از روش‌های کاهش بُعد استفاده می‌کنیم که از طریق فشرده‌سازی ماتریس‌ها تقریبی کم‌رتبه از آن به دست می‌دهد و علاوه‌بر تشخیص داده‌های کم‌اهمیت و حذف آن‌ها، ساختار داخلی داده‌ها نیز بهتر مشخص می‌شود. به همین دلیل، روش‌های تجزیهٔ مقدار تکین (SVD) و تجزیهٔ نیمه‌گسسته (SDD) را مورد بررسی قرار می‌دهیم و نتایج حاصل را مقایسه می‌کنیم. نتایج به‌دست آمده نشان می‌دهد که هرچند SVD کمترین خطا را دارد، ولی SDD با خطایی نزدیک به SVD، از نظر زمان اجرا و به‌ویژه حافظهٔ موردنیاز به‌صرفه‌تر است.

۱ مقدمه

امروزه استفاده از شبکه‌های اجتماعی گسترش یافته است. کاربران با جستجوی در این شبکه‌ها می‌توانند کالای مورد نظر خود را خریداری کنند. بااین‌حال، به‌دلیل گستردگی شبکه‌ها این کار وقت‌گیر است و ازاین‌رو، استفاده از سامانه‌های پیشنهادگر[1] نیز در شبکه‌های اجتماعی گسترش یافته است. سامانه‌های پیشنهادگر در اواسط سال‌های ۱۹۹۰، از زمانی که اولین مقالات در زمینهٔ پالایش مشارکتی مطرح شد، به یکی از موضوعات مهم تحقیقاتی تبدیل شدند. به‌کارگیری این

عبارات و کلمات کلیدی: سامانه‌های پیشنهادگر، پالایش مشارکتی، ماتریس کاربر–کالا، کاهش بُعد، تجزیهٔ مقدار تکین، تجزیهٔ نیم‌گسسته

نوع مقاله: ترویجی؛ تاریخ دریافت: ۱۳۹۷/۱۱/۲۷؛ تاریخ پذیرش: ۱۳۹۸/۹/۲

[1] recommender systems

doi: 10.30504/mct.2022.335

© ۱۴۰۱ (انجمن ریاضی ایران)

سامانه‌ها در تجارت الکترونیک بسیار رایج شده است به‌گونه‌ای‌که تعداد زیادی از فروشگاه‌های مجازی و تارنماهایی که خدمات مختلف به کاربران عرضه می‌کنند، از آن‌ها استفاده می‌کنند. از مهم‌ترین تارنماهایی که از سامانه‌های پیشنهادگر استفاده می‌کنند می‌توان به آمازن با بیش از صد میلیون کاربر اشاره کرد. هدف سامانه‌های پیشنهادگر، در اصل، رتبه‌بندی کالاهای سامانه، به‌لحاظ نزدیکی به علاقه‌مندی‌های کاربران است تا در هنگام ارائهٔ پیشنهاد، کالاهایی با رتبهٔ بالاتر را به کاربر پیشنهاد دهند. بدون شک اساسی‌ترین مؤلفه در سامانه‌های پیشنهادگر، فنون و الگوریتم‌های آن است که مهم‌ترینِ آن‌ها سامانه‌های پیشنهادگر مبتنی‌بر پالایش مشارکتی[۱]، پالایش محتوا-محور[۲]، پالایش معاشرت-محور[۳]، پالایش دانش-محور[۴]، پالایش آگاه از بافت[۵] ، و پالایش ترکیبی[۶] هستند [۹].

روش پالایش مشارکتی نسبت به سایر روش‌ها کاربرد بیشتری دارد. این روش از الگوریتم نزدیک‌ترین همسایه‌ها[۷] برای پیشنهاد به کاربران استفاده می‌کند. سامانهٔ پیشنهادگر مبتنی‌بر پالایش مشارکتی از پایگاه داده‌ای شامل اولویت‌های کاربران برای کالاها استفاده می‌کند تا علاقهٔ یک کاربر جدید را نسبت به موضوعات یا محصولات دیگر پیش‌بینی کند. فرض کنید m کاربر $\{u_۱, u_۲, \ldots, u_m\}$ و n کالا $\{i_۱, i_۲, \ldots, i_n\}$ در دست باشند. ماتریس کاربر-کالای R از مرتبهٔ $m \times n$ با درایه‌هایی که امتیاز[۸] کاربران به کالاها است، تشکیل می‌شود. برای هر کاربرِ u_i، فهرستی از کالاهای I_{u_i} وجود دارد که به آن‌ها امتیاز داده است و با توجه به اولویت‌های او از طریق رفتارش استنباط شده است. امتیازها می‌توانند به‌صورت صریح در مقیاس ۱ تا ۵ و یا ضمنی باشند. هدف هر سامانهٔ پیشنهادگر، پیش‌بینی جاهای خالی در یک ماتریس کاربر-کالا است [۵]. چون کاربرها به تعداد کمی از کالاها امتیاز می‌دهند، R یک ماتریس تُنک است. الگوریتم‌های جبر خطی روش‌های مناسبی برای کار با ماتریس‌های تُنک ارائه می‌دهند. در این مقاله، با استفاده از تجزیه‌های ماتریسی و کاهش بُعد[۹] ماتریس R، راه‌حل‌های موجود در مواجهه با تنکی ماتریس کاربر-کالا را مرور می‌کنیم.

ساختار این مقاله چنین است: در بخش دوم، با روش سامانه‌های پیشنهادگر مبتنی بر پالایش مشارکتی و چگونگی ارائهٔ پیشنهاد آشنا می‌شویم. در بخش سوم، به بررسی برخی از روش‌های کاهش بُعد می‌پردازیم. در بخش چهارم یک روش پیشنهادگر براساس تجزیه‌های ماتریسی و پالایش

[1] collaborative filtering [2] content-based filtering [3] social-based filtering [4] knowledge-based filtering [5] context-aware filtering [6] hybrid filtering [7] nearest neighbors algorithm [8] rate [9] dimensionality reduction

مشارکتی بیان می‌کنیم. در بخش پایانی، روش گفته‌شده را برای مجموعه‌داده‌های پایگاه مووی‌لنز[1] پیاده‌سازی می‌کنیم.

۲ سامانه‌های پیشنهادگر مبتنی بر پالایش مشارکتی

یکی از مهم‌ترین فنون در سامانه‌های پیشنهادگر، روش پالایش مشارکتی است که پایه و مبنای کار در بسیاری از راهکارهای دیگر نیز به شمار می‌رود. روش کار این الگوریتم در واقع به همان صورتی است که ما در تصمیم‌گیری‌های روزمره عمل می‌کنیم. برای مثال، کالایی را می‌خریم که بیشتر مورد پسند دیگران واقع شده باشد. در روش پالایش مشارکتی بیشتر تجربهٔ دیگران مدنظر قرار می‌گیرد تا تجربهٔ خود فرد. به‌طورکلی، روش پالایش مشارکتی از طبیعت انسان‌ها الهام می‌گیرد: «انسان‌ها عقایدشان را با دیگران در میان می‌گذارند.» ایدهٔ اصلی در این سامانه‌ها این است که اگر کاربران در گذشته علایق مشابهی داشته‌اند (اگر فیلم یا کتاب همانندی را دیده یا خریده‌اند)، در آینده هم سلیقه‌های مشابهی خواهند داشت [۱۳].

الگوریتم‌های موجود در سامانه‌های مبتنی بر پالایش مشارکتی به دو دستهٔ کلی تقسیم می‌شوند: الگوریتم‌های پالایش مشارکتی مبتنی بر حافظه[2] و الگوریتم‌های پالایش مشارکتی مبتنی بر مدل[3]. الگوریتم‌های مبتنی بر حافظه با استفاده از ماتریس رتبه‌بندی، پیشنهادهای را براساس نزدیک‌ترین همسایگان ارائه می‌کنند. الگوریتم‌های مبتنی بر مدل، پیشنهادات را براساس مدلی ارائه می‌کنند که طبق پیشنهادهای کاربران ساخته شده است. این الگوریتم‌ها از ماتریس رتبه‌بندی برای ایجاد یک مدل استفاده می‌کنند و سپس با این مدل پیشنهادها را ارائه می‌دهند. الگوریتم‌های مبتنی بر حافظه نسبت به الگوریتم‌های مبتنی بر مدل نتیجهٔ بهتر و دقیق‌تری به‌دست می‌دهند و هنگامی‌که ماتریس ارزیابی مرتباً تغییر می‌کند، مناسب‌تر عمل می‌کنند. ازسوی‌دیگر، این الگوریتم‌ها زمان محاسباتی زیادی نیاز دارند که این امر باعث می‌شود تا در پایگاه داده‌های بزرگ از الگوریتم‌های تقریبی مبتنی بر مدل استفاده شود.

۱.۲ الگوریتم پالایش مشارکتی مبتنی برحافظه

الگوریتم های پالایش مشارکتی مبتنی بر حافظه برای انجام پیش‌بینی، از همه یا جزئی از پایگاه دادهٔ کاربر–کالا استفاده می‌کنند. هر کاربر بخشی از یک گروه مرکب از افراد با علاقه‌های مشابه است. با

[1] MovieLens Datasets:http://www.grouplens.org/node/73 [2] memory based collaborative filtering [3] model based collaborative filtering

تعیین آنچه که همسایگی یک کاربر جدید (یا کاربر فعال) نامیده می‌شود، اولویت‌ها فرد در انتخاب کالاهای جدید پیش‌بینی می‌شود. الگوریتم پالایش مشارکتی مبتنی بر همسایگی، یکی از الگوریتم‌های پالایش مشارکتی مبتنی بر حافظه است که الگوریتم Kـنزدیک‌ترین همسایه‌ها (KNN[1]) نام دارد. در این الگوریتم دو رویکرد وجود دارد. رویکرد اول دیدگاه کاربرـکاربر و شامل سه گام است:

۱. براساس یک معیار، مانند تشابه کسینوسی[2] یا همبستگی پیرسن[3] ، برای کاربر u تعداد K همسایه انتخاب می‌شود که بیشترین شباهت را به کاربر u دارند.
۲. به‌ازای همهٔ کالاهای موجود در سامانه، معیاری برای پیش‌بینی خانه‌های خالی ماتریس کاربرـکالا محاسبه می‌شود.
۳. تعداد N کالا که به‌سلیقهٔ کاربر نزدیک‌ترند به او پیشنهاد داده می‌شود.

رویکرد دوم دیدگاه کالاـکالا و شامل سه گام است:

۱. براساس معیارهای تشابه برای هر کالای i تعداد q همسایه تعیین می‌شود.
۲. درصورتی‌که کاربر u به کالای i تاکنون امتیازی نداده باشد، براساس امتیازاتی که این کاربر به کالاهای همسایه i داده است مقدار پیش‌بینی محاسبه می‌شود.
۳. براساس مقدار پیش‌بینی‌ها، کالاهایی که بیشترین مقدار پیش‌بینی را دارند به کاربر u پیشنهاد می‌شوند.

دراصل، در مورد کاربرـکاربر، ماتریس رتبه‌بندی به‌صورت سطری و در مورد کالاـکالا به‌صورت ستونی تحلیل می‌شود.

ابتدا دو عبارتِ کاربرِ هدف و کاربرِ فعال را تعریف می‌کنیم، سپس سه گام در هر دو رویکرد را با جزئیات بیشتر بررسی می‌کنیم.

کاربر هدف، کاربری است که درنظر داریم برای او، به کالاهایی که امتیاز نداده است، امتیازی پیش‌بینی کنیم. کاربر فعال، کاربری است که به کالای مورد نظر امتیاز داده است و می‌خواهیم از آن امتیاز برای پیش‌بینی امتیاز کاربر هدف استفاده کنیم [۱۰].

گام اول: محاسبهٔ تشابه

محاسبهٔ تشابه بین کالاها و یا کاربران در الگوریتم‌های پالایش مشارکتی مبتنی بر حافظه گامی بسیار مهم است. اندیشهٔ اصلی در الگوریتم‌های پالایش مشارکتی دیدگاه کالاـکالا این است که تشابه بین

[1] K nearest neighbors [2] cosine similarity [3] Pearson correlation

کالای i و کالای j در کاربران فعال، که هر دوی این کالاها را رتبه‌دهی کرده‌اند، محاسبه شود. بدین منظور ماتریس تشابه $\mathbb{W}$ تعریف می‌شود. در اینجا $\mathbb{W}_{i,j}$ میزان تشابه دو کالای رتبه‌دهی‌شده توسط کاربران است و کمیتی است که باید تعیین شود. در الگوریتم پالایش مشارکتی دیدگاه کاربر–کاربر، تشابه $\mathbb{W}_{u,v}$ بین کاربران u و v را که دارای کالاهای مشابه رتبه‌بندی‌شده‌اند، محاسبه می‌کنیم. شیوه‌های مختلفی برای محاسبه تشابه یا وزن بین کاربران و کالاها وجود دارد که پرکاربردترین آن‌ها عبارت‌اند از تشابه مبتنی بر بردار کسینوسی، فاصلهٔ اقلیدسی، و تشابه مبتنی بر همبستگی پیرسن. دو مورد اول و دوم را بررسی می‌کنیم.

تشابه کسینوسی

هر سطر ماتریس رتبه، حاوی اطلاعاتی از یک کاربر و هر ستون مربوط به کالایی است. بنابراین تشابه بین کاربران و یا کالاها را می‌توان به‌عنوان تشابه بین دو بردار تفسیر کرد. فرض کنید x و y دو بردار هم‌اندازه باشند. کسینوس زاویهٔ بین آن‌ها را به‌منزلهٔ تشابه بین آن‌ها درنظر می‌گیرند:

$$\mathbb{W}_{x,y} = \cos(x, y) = \frac{x \cdot y}{\|x\| \, \|y\|}.$$

x و y می‌توانند بردارهای نظیر دو کالا یا کاربر باشند. کاربران مختلف ممکن است از مقیاس‌های رتبه‌دهی مختلفی استفاده کنند و درنتیجه تشابه کسینوسی بردارها مناسب آن‌ها نیست. برای رفع این مشکل، از روش ضریب همبستگی پیرسن استفاده می‌شود.

تشابه براساس همبستگی پیرسن

در این شیوه، تشابه بین دو کاربر u و v، $\mathbb{W}_{u,v}$، و یا تشابه بین دو کالای i و j، $\mathbb{W}_{i,j}$، با محاسبهٔ همبستگی پیرسن یا دیگر تشابه‌های مبتنی بر همبستگی اندازه‌گیری می‌شود. همبستگی پیرسن، محدوده‌ای را که دو متغیر به‌طور خطی با یکدیگر مرتبط‌اند اندازه‌گیری می‌کند و مقدار آن از -۱ تا $+۱$ متغیر است. همبستگی $+۱$ معرّف آن است که دو کاربر کاملاً علایق مرتبط با هم دارند، درصورتی‌که عدد -۱ تضاد علایق دو کاربر را نشان می‌دهد.

در دیدگاه کاربر–کاربر، همبستگی پیرسن بین کاربران u و v با فرمول زیر تعیین می‌شود

$$\mathbb{W}_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^۲}\sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^۲}}, \qquad (۱.۲)$$

که در آن I مجموعهٔ کالاهایی است که هردو کاربر u و v به آن‌ها امتیاز داده‌اند، $r_{u,i}$ و $r_{v,i}$ به‌ترتیب

امتیاز کاربران u و v به کالای i و $\bar{r}_u$ و $\bar{r}_v$ میانگین امتیازهای این کاربران است.

برای رویکرد کالا–کالا، فرض کنید U مجموعهٔ تمام کاربرانی را نشان دهد که به کالاهای i و j امتیاز داده‌اند. همبستگی پیرسن با فرمول زیر تعیین می‌شود

$$\mathbb{W}_{i,j} = \frac{\sum_{u\in U}(r_{u,i}-\bar{r}_i)(r_{u,j}-\bar{r}_j)}{\sqrt{\sum_{u\in U}(r_{u,i}-\bar{r}_i)^2}\sqrt{\sum_{u\in U}(r_{u,j}-\bar{r}_j)^2}}, \qquad (۲.۲)$$

که در آن $r_{u,i}$ و $r_{u,j}$ به‌ترتیب امتیاز کاربر u به کالاهای i و j، و $\bar{r}_i$ و $\bar{r}_j$ میانگین امتیازهای کالاهای i و j است [۱۲]. از تشابه مبتنی بر همبستگی پیرسن، معمولاً در الگوریتم پالایش مشارکتی استفاده می‌شود.

گام دوم: پیش‌بینی و محاسبهٔ پیشنهادها

دستیابی به پیش‌بینی‌ها و پیشنهادها مهم‌ترین گام در سامانهٔ پالایش مشارکتی است. در الگوریتم پالایش مشارکتی مبتنی بر همسایگی، زیرمجموعه‌ای از نزدیک‌ترین همسایه‌های یک کاربرِ هدف براساس تشابه آن‌ها با آن کاربر انتخاب می‌شوند و مجموع وزن‌داری از امتیازدهی آن‌ها برای پیش‌بینی در مورد کاربرِ هدف به‌کار می‌رود.

برای پیش‌بینی در مورد کاربرِ هدف v در یک کالای معین i، می‌توانیم میانگین وزن‌دار سادهٔ همهٔ امتیازدهی‌ها را در مورد آن کالا با فرمول زیر به‌دست آوریم

$$\mathbb{P}_{v,i} = \frac{\sum_{u\in U} r_{u,i}\times \mathbb{W}_{v,u}}{\sum_{u\in U}|\mathbb{W}_{v,u}|}, \qquad (۳.۲)$$

که در آن U مجموعهٔ همه همسایه‌های کاربر v است که به کالای i امتیاز داده‌اند و $\mathbb{W}_{v,u}$ وزن بین کاربر v و کاربر u است. برای دیدگاه کالا–کالا از میانگین وزن‌دار سادهٔ زیر استفاده می‌کنیم

$$\mathbb{P}_{v,i} = \frac{\sum_{j\in N} r_{v,n}\times \mathbb{W}_{i,j}}{\sum_{j\in N}|\mathbb{W}_{i,j}|}, \qquad (۴.۲)$$

که در آن N مجموعهٔ کالاهایی است که کاربر v به آن‌ها امتیاز داده است، کمیّت $\mathbb{W}_{i,j}$ تشابه بین کالاهای i و j است و $r_{v,j}$ رتبه یا امتیاز کاربر v به کالای j است [۱۲].

گام سوم: N پیشنهاد برتر

هدف این گام، پیشنهاد یک مجموعه از N کالا با بالاترین رتبه است که مورد علاقهٔ یک کاربر خاص باشد. روش‌های N پیشنهاد برتر، ماتریس کاربر–کالا را برای کشف روابط بین کاربران و یا کالاهای

مختلف تحلیل می‌کند و از آن‌ها برای محاسبهٔ پیشنهادها استفاده می‌کند [۴].

۲.۲ الگوریتم پالایش مشارکتی مبتنی بر مدل

طراحی و توسعهٔ الگوریتم‌های یادگیری ماشین[۱]، داده کاوی[۲] و غیره به سامانهٔ امکان شناسایی الگوهای پیچیده و پیش‌بینی راهکار هوشمندانه را در زمینهٔ فعالیت‌های پالایش مشارکتی ارائه می‌دهد. الگوریتم‌های پالایش مشارکتی مبتنی بر مدل مانند مدل‌های بیزی[۳]، مدل‌های خوشه‌بندی[۴]، شبکه‌های وابستگی و مدل اندیس‌گذاری معنایی پنهان (LSI[۵]) نیز مورد تحقیق قرار گرفته‌اند تا کمبودهای الگوریتم‌های پالایش مشارکتی مبتنی بر حافظه برطرف شود. به‌منظور کاهش زمان محاسبات و حافظهٔ مورد نیاز از روش‌های پالایش مشارکتی مبتنی بر مدل استفاده می‌شود [۱۴].

روش ارزیابی سامانه‌های پیشنهادگر

متداول‌ترین معیار ارزیابی در سامانه‌های پیشنهادگر، خطای جذر میانگین مربعات (RMSE[۶]) است که به‌صورت زیر تعریف می‌شود

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{(u,i)}(r_{u,i} - \hat{r}_{u,i})^{۲}}{n}} \tag{۵.۲}$$

و در آن $r_{u,i}$ امتیازی است که کاربر u به کالای i داده است، $\hat{r}_{u,i}$ امتیازی است که پیش‌بینی می‌شود و n تعداد پیش‌بینی‌ها است.

نرمال‌سازی

یکی از روش‌های بهبود پیشنهادها در سامانه‌های پیشنهادگر استفاده از روش نرمال‌سازی[۷] است. در سامانه‌های پیشنهادگر مبتنی بر پالایش مشارکتی برای قابل قیاس بودن داده‌های موجود آن‌ها را نرمال می‌کنیم. برای مثال، در ماتریس رتبه‌بندی کاربر–کالا ممکن است بعضی از کاربران از امتیازهای پایین و برخی از امتیازهای بالا استفاده کنند، که باعث می‌شود بعضی از کالاها دارای امتیاز خیلی بالا و بعضی دیگر دارای امتیاز خیلی پایین باشند. برای برطرف کردن این ناهمگونی، تمام عناصر ماتریس رتبه‌بندی را نرمال می‌کنیم تا قابل مقایسه باشند. روش‌های مختلفی برای نرمال‌سازی ماتریس رتبه‌بندی وجود دارد. یکی از متداول‌ترین روش‌ها کم‌کردن میانگین سطر و میانگین ستون از عناصر ماتریس است. برای نرمال‌سازی عناصر ماتریس کاربر–کالای R از مرتبهٔ

[1] machine Learning [2] data mining [3] Bayesian [4] clustering [5] latent semantic indexing [6] root mean square error [7] normalization

$m \times k$ از فرمول زیر استفاده می‌کنیم

$$\begin{aligned}\hat{r}_{u,i} &= r_{u,i} - \alpha\bar{r} - \beta\bar{r}_u - \gamma\bar{r}_i \\ &= r_{u,i} - \alpha(\frac{1}{mk}\sum_{u,i} r_{u,i}) - \beta(\frac{1}{m}\sum_{u} r_{u,i}) - \gamma(\frac{1}{k}\sum_{i} r_{u,i}). \end{aligned} \tag{۶.۲}$$

در فرمول بالا $\bar{r}$ میانگین همهٔ عناصر ماتریس، $\bar{r}_u$ میانگین سطر نظیر کاربر u، $\bar{r}_i$ میانگین ستون نظیر کالای i، و $r_{u,i}$ رتبه یا امتیازی است که کاربر u به کالای i می‌دهد. انتخاب پارامترهای بهینهٔ (α, β, γ) می‌توانند به‌وسیلهٔ فنون یادگیری ماشین یا روش‌های آماری مشخص شوند [۱۳]. درحالت‌کلی $\alpha + \beta + \gamma = 1$. توجه کنید که عناصر ماتریس پس از نرمال‌سازی بین -1 تا $+1$ قرار می‌گیرند، یعنی $-1 \leq \hat{r}_{u,i} \leq +1$. نکتهٔ مهم این است که در آخر، اضافه‌کردن میانگین کم‌شده به رتبه‌هایی که پیش‌بینی کرده‌ایم، ضروری است:

$$\widehat{\mathrm{pred}}(u,i) = \mathrm{pred}(u,i) + \alpha\bar{r} + \beta\bar{r}_u + \gamma\bar{r}_i. \tag{۷.۲}$$

مطالعات نشان می‌دهد که دقت پیش‌بینی پس از نرمال‌سازی رتبه‌ها یا امتیازهای ماتریس کاربر–کالا افزایش می‌یابد [۱۳].

۳ روش‌های کاهش بُعد

در هر مجموعه‌داده که به صورت ماتریسی ذخیره می‌شود اطلاعات زیادی پنهان شده است. کاهش بُعد ابزاری قدرتمند برای پیدا کردن ساختار پنهان داده‌ها است. مزایای کاهش بُعد، کاهش حافظه و حجم محاسبات، تشخیص داده‌های پرت (نوفه)، و کشف روابط و عامل‌های پنهان در ماتریس اولیه است. کاهش بُعد در مرحلهٔ پیش‌پردازش اطلاعات انجام می‌شود. برخی از روش‌های کاهش بُعد، یک تقریب کم‌رتبه را از داده‌های با بُعد بالا استخراج می‌کند [۷].

فرض کنید $A \in \mathbb{R}^{m\times n}$ دارای رتبهٔ r است. تقریبی کم‌رتبه از ماتریس A عبارت است از $A_k = C_k F_k$ که در آن $C_k \in \mathbb{R}^{m\times k}$، $F_k \in \mathbb{R}^{k\times n}$، و k (بسیار) کوچک‌تر از رتبهٔ A است. این حاصل‌ضرب تقریب مناسبی برای عناصر خالی ماتریس کاربر–کالا است. ماتریس C_k حاوی مفاهیمی در ارتباط با کاربر و ماتریس F_k در ارتباط با کالاهاست. از بهترین انواع روش‌های کاهش بُعد می‌توان به روش‌های تجزیهٔ مقدار تکین (SVD) و تجزیهٔ نیم‌گسسته (SDD) اشاره کرد. در این بخش به بیان روش‌های فوق و برخی خواص آن‌ها می‌پردازیم.

۱.۳ تجزیهٔ مقدار تکین (SVD)

در این بخش SVD را معرفی و بعضی خواص آن را بیان می‌کنیم.

قضیه ۱.۳ ([۳]). ماتریس $A \in \mathbb{R}^{m\times n}$ با رتبهٔ r را می‌توان به‌صورت $A = U\Sigma V^T$ تجزیه کرد که در آن $U \in \mathbb{R}^{m\times m}$ و $V \in \mathbb{R}^{n\times n}$ ماتریس‌هایی متعامد و $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$ با $p = \min\{m, n\}$ ماتریسی قطری است به‌طوری‌که

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0, \quad \sigma_{r+1} = \cdots = \sigma_p = 0$$

تجزیهٔ ماتریس A به‌صورت $A = U\Sigma V^T$ را تجزیهٔ مقدار تکین A می‌نامیم.

فرض کنید $A = U\Sigma V^T$ با رتبهٔ r، $k < r$ عدد صحیح مثبت و $A_k = U_k\Sigma_k V_k^T$ باشد که در آن U_k و V_k به‌ترتیب شامل k ستون اول ماتریس‌های U و V، و Σ_k ماتریس قطری $k \times k$ شامل k سطر و k ستون اول Σ است. ماتریس A_k یک تقریب با رتبهٔ k برای ماتریس A است که آن را می‌توان به‌صورت مجموعی از ماتریسی‌های رتبه‌یک نوشت:

$$A_k = \sigma_1 u_1 v_1^T + \cdots + \sigma_k u_k v_k^T, \quad k < r.$$

در اینجا u_j و v_j به‌ترتیب ستون‌های U و V را نشان می‌دهند. ماتریس A_k بهترین تقریب از رتبهٔ k در نرم‌های ۲ و فروبنیوسی برای ماتریس A است. ماتریس U_k بیانگر اطلاعات مربوط به کاربران (تعداد سطرهای آن برابر m، تعداد کاربران) و V_k بیانگر اطلاعات مربوط به کالاها (تعداد ستون‌های آن برابر n، تعداد کالاها) است. برای بررسی تشابه بین کاربران، ماتریس $A_kA_k^T$ و برای بررسی تشابه بین کالاها، ماتریس $A_k^TA_k$ را محاسبه می‌کنیم. در اندیس‌گذاری معنایی پنهان (LSI) از SVD استفاده می‌شود، زیرا روابط و معانی پنهان داده‌ها را به‌خوبی آشکار می‌کند [۷].

۲.۳ تجزیهٔ نیم‌گسسته (SDD)

نخستین بار، اولیری و پیلگ روش SDD را برای فشرده‌سازی تصویر به‌کار بردند [۸]. پس از آن کولدا و اولیری از روش SDD برای LSI در بازیابی اطلاعات استفاده کردند [۶].

تعریف ۲.۳. فرض کنید $A \in \mathbb{R}^{m\times n}$. تجزیهٔ نیم‌گسستهٔ ماتریس A چنین تعریف می‌شود

$$A_k = X_k D_k Y_k^T = \sum_{i=1}^{k} d_i x_i y_i^T, \tag{۱.۳}$$

که در آن $X_k = (x_1 \quad \cdots \quad x_k) \in \mathbb{R}^{m\times k}$، $Y_k = (y_1 \quad \cdots \quad y_k) \in \mathbb{R}^{n\times k}$، و $D_k = \mathrm{diag}(d_1, \ldots, d_k)$ ماتریسی قطری است. بردارهای x_i و y_i شامل اعداد مجموعهٔ $\mathbf{S} = \{-1, 0, 1\}$ و d_iها اعداد حقیقی مثبت‌اند.

روش SDD با استفاده از یک الگوریتم تکراری اجرا می‌شود و معمولاً نمی‌تواند A را از نو تولید کند، حتی اگر $k = n$ باشد. انتخاب k به میزان دقت مورد نیاز بستگی دارد [۶].

این تجزیه، تقریبی از ماتریس را به‌صورت مجموعی از ماتریس‌های رتبه‌یک $d_i x_i y_i^T$ به روش زیر محاسبه می‌کند. تعریف کنید $A_0 = 0$ و در مرحلهٔ iام ماتریس مانده $R_i = A - A_{i-1}$ را درنظر بگیرید. هدفْ محاسبهٔ سه‌گانهٔ (d_i, x_i, y_i) با این شرط است که

$$\min_{\substack{x\in\mathbf{S}^m\\ y\in\mathbf{S}^n\\ d>0}} F_i(d,x,y) \equiv \|R_i - dxy^T\|_F^2. \tag{۲.۳}$$

یادآوری می‌کنیم که $\|B\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |b_{ij}|^2}$ نرم فروبینوسی ماتریس $B = (b_{ij})$، از مرتبهٔ $m \times n$، است. مسئلهٔ (۲.۳) یک مسئلهٔ برنامه‌ریزی عددصحیحِ آمیخته است که می‌توان با حذف d آن را مانند یک مسئلهٔ برنامه‌ریزی عددصحیح صورت‌بندی کرد [۸]. برای این کار، $F_i(d,x,y)$ را به‌صورت زیر بازنویسی می‌کنیم

$$F_i(d,x,y) = \|R_i\|_F^2 - 2dx^T R_i y + d^2\|x\|_2^2\|y\|_2^2. \tag{۳.۳}$$

مقدار بهینهٔ d که با حل $\partial F_i/\partial d = 0$ محاسبه می‌شود عبارت است از $d^* = \dfrac{x^T R_i y}{\|x\|_2^2\|y\|_2^2}$. با جانشانی d^* در (۳.۳)، داریم

$$F_i(d^*,x,y) = \|R_i\|_F^2 - \frac{(x^T R_i y)^2}{\|x\|_2^2\|y\|_2^2}. \tag{۴.۳}$$

بنابراین، جواب مسئلهٔ (۲.۳) هم‌ارز با جواب مسئلهٔ زیر است

$$\max_{\substack{x\in\mathbf{S}^m\\ y\in\mathbf{S}^n}} \tilde{F}_i(x,y) = \frac{(x^T R_i y)^2}{\|x\|_2^2\|y\|_2^2}. \tag{۵.۳}$$

مسئلهٔ (۵.۳) شامل مجهول‌های x و y، مسئله‌ای درجه‌دوم است. اولیری و پیلگ برای محاسبهٔ تقریبی از بردارهای x و y، از یک روش تکراری استفاده کردند. به‌این‌ترتیب‌که، ابتدا یک بردار

ناصفر دلخواه مانند $y \in \mathbf{S}^n$ درنظر گرفتند. با انتخاب مقداری برای y، تنها بردار مجهول x باقی می‌ماند و مسئلهٔ (۵.۳) به مسئله‌ای خطی تبدیل می‌شود. پس از محاسبهٔ x، از نو مسئلهٔ (۵.۳) را برحسب مجهول y حل می‌کنیم. روند تکراری، تا زمانی‌که جواب بهینه حاصل شود، ادامه می‌یابد. با توجه به رابطهٔ (۴.۳) داریم

$$\|R_{i+1}\|_F < \|R_i\|_F$$

یعنی، دنبالهٔ ماتریس‌های مانده اکیداً نزولی و درنتیجه الگوریتم SDD همگرا است. برای yی ثابت، مسئلهٔ (۵.۳) عبارت است از

$$\max_{x \in \mathbf{S}^m} \frac{(x^T s)^2}{\|x\|_2^2}, \tag{۶.۳}$$

که در آن $s = R_i y / \|y\|_2^2$. این مسئله را می‌توان با استفاده از شگرد زیر حل کرد [۸].

فرض کنید بردار x در مسئلهٔ برنامه‌ریزی عددصحیح (۶.۳) دقیقاً J عنصر ناصفر دارد و عناصر بردار s به‌صورت $|s_{i_1}| \geq |s_{i_2}| \geq \cdots \geq |s_{i_m}|$ مرتب شده باشند. دراین‌صورت، جواب مسئلهٔ (۶.۳) چنین است:

$$x_{i_j} = \begin{cases} \operatorname{sign}(s_{i_j}) & 1 \leq j \leq J \\ 0 & J+1 \leq j \leq m \end{cases}$$

ازاین‌رو، تنها m بردار x ممکن برای تعیین جواب بهینهٔ (۶.۳) وجود دارد [۶].

در ادامه الگوریتم اولیری-پیلگ برای یافتن تقریب SDD از مرتبهٔ k را بیان می‌کنیم.

الگوریتم اولیری-پیلگ برای تقریب SDD

ورودی: ماتریس A از مرتبهٔ $m \times n$.

(۱) قرار دهید $R_1 = A$.

برای $i = 1, 2, \ldots, k$ تکرار خارجی زیر را انجام دهید:

(۲) تکرار خارجی

(a) بردار شروع y را به‌گونه‌ای انتخاب کنید که $R_i y \neq 0$

برای $j = 1, 2, \ldots, j_{\max}$ تکرار داخلی زیر را انجام دهید:

(b) تکرار داخلی

$(b۱)$ با بردار ثابت y، جواب مسئله $\max_{x\in\mathbf{S}^m} \frac{(x^T R_i y)^۲}{\|x\|_۲^۲}$ را بیابید.

$(b۲)$ با بردار جواب x از $(b۱)$، جواب مسئله $\max_{y\in\mathbf{S}^n} \frac{(y^T R_i^T x)^۲}{\|y\|_۲^۲}$ را بیابید.

(c) **پایان تکرار داخلی**

(d) قرار دهید $x_i = x$، $y_i = y$، $d_i = \frac{x_i^T R_i y_i}{\|x\|_۲^۲\|y\|_۲^۲}$ و

$$A_i = A_{i-۱} + d_i x_i y_i^T, \quad R_{i+۱} = R_i - d_i x_i y_i^T$$

(۳) پایان تکرار خارجی

(۴) قرار دهید $X_k = (x_۱ \quad x_۲ \quad \cdots \quad x_k)$، $Y_k = (y_۱ \quad y_۲ \quad \cdots \quad y_k)$ و $D_k = \mathrm{diag}(d_۱, \ldots, d_k)$. تجزیهٔ SDD ماتریس A عبارت است از $A_k = X_k D_k Y_k^T$.

انتخاب نقطهٔ شروع برای بردارهای x یا y در الگوریتم SDD مهم است و انتخاب‌های مختلف نتایج متفاوتی را تولید می‌کند. به‌این‌ترتیب، الگوریتم تجزیه‌ای یکتا برای ماتریس A تولید نمی‌کند.

روش‌های مختلفی برای انتخاب بردار اولیهٔ y_i در گام iام الگوریتم SDD وجود دارد [۶]. برای مثال، در گام iام، رویکرد آغازش با ماکسیمم عنصر[۱] بردار y را ستون jام ماتریس همانی، یعنی e_j، انتخاب می‌کند که در آن j، ستون شامل بزرگ‌ترین عنصر ماتریس R_i است.

برخی روش‌های دیگر انتخاب نقطهٔ شروع y_i عبارت‌اند از آغازش دوری[۲]، آغازش آستانه‌ای[۳] و آغازش SVD.

در الگوریتم اولیری–پیلگ دو حلقهٔ تکرار وجود دارد. تعداد تکرار حلقهٔ خارجی برای محاسبهٔ $A_k = X_k D_k Y_k^T$ برابر k از قبل انتخاب می‌شود. برای توقف حلقهٔ داخلی معیارهای مختلفی وجود دارد. برای مثال، حلقهٔ داخلی را می‌توان به یک تعداد مشخص از پیش‌تعیین‌شده تکرار کرد، یا تکرار داخلی را تا زمانی ادامه داد که $\beta := \frac{(x^T R_i y)^۲}{\|x\|_۲^۲\|y\|_۲^۲}$، تقریباً مقدار ثابتی شود. برای مشاهدهٔ جزییات بیشتر به [۶] مراجعه کنید.

۳.۳ مقایسهٔ عملکرد روش‌های کاهش بعد

مهم‌ترین مشخصهٔ SDD، تولید بهترین تقریب از رتبهٔ k در نرم‌های ۲ و فروبنیوسی است. مزیت اصلی SDD نسبت به انواع دیگر تقریب ماتریسی مانند SVD این است که معمولاً تقریبی دقیق‌تر

[1] maximum element initialization [2] cycling initialization [3] threshold initialization

با ذخیره‌سازی کمتر ارائه می‌کند. در SVD حتی اگر ماتریس اولیه تنک باشد، ماتریس‌های U و V ماتریس‌های چگال‌اند. یک ویژگی مهم در SDD فضای حافظهٔ مورد نیاز اندک برای ذخیرهٔ عامل‌های X و Y است. عامل‌های X و Y معمولاً تنک‌اند، به‌علاوه، برای ذخیره‌سازی یک SDD با k جمله، نیاز به ذخیرهٔ k اسکالر مثبت d_i و همچنین $k(m+n)$ درایه از مجموعهٔ $\mathbf{S}$ است. اعداد ۱، ۰ و ۱− به فضای ذخیره‌سازی $\log_2(3) \simeq 2$ بیت نیاز دارند. فضای ذخیره‌سازی SVD از مرتبهٔ k برابر $k(m+n+1)$ است. بنابراین، فضای ذخیره‌سازی مورد نیاز برای ماتریس‌های تنک X و Y با درایه‌هایی از مجموعهٔ $\{-1, 0, 1\}$، به‌طور قابل‌ملاحظه‌ای از فضای ذخیره‌سازی ماتریس‌های U و V که معمولاً ماتریس‌های چگالی‌اند، کمتر است [۶].

۴ ارائه پیشنهاد براساس روش کاهش بُعد و پالایش مشارکتی

همان‌طورکه در بخش نخست بیان شد، یکی از مهم‌ترین چالش‌ها در سامانه‌های پیشنهادگر مبتنی بر پالایش مشارکتی، پراکندگی داده‌ها است که پیشنهاد دادن به کاربر را دچار مشکل می‌کند. سامانهٔ پیشنهادگر مبتنی بر پالایش مشارکتی برای تولید پیشنهادهای با کیفیت بالا، الگوریتم نزدیک‌ترین همسایگی را به‌کار می‌گیرد، اما زمانی‌که داده‌ها خیلی پراکنده باشند و درواقع ماتریس کاربر–کالا تُنک باشد، روش نزدیک‌ترین همسایگی به‌دشواری می‌تواند عناصر خالی ماتریس را پیش‌بینی کند و ممکن است پیشنهادهایی با دقت پایین تولید کند [۱۱].

روش‌های کاهش بُعد تقریب مناسبی برای عناصر خالی ماتریس کاربر–کالا ارائه می‌دهند. همچنین با استفاده از این روش می‌توانیم ارتباطات پنهان بین کاربران و کالاها را به‌دست آوریم. روش‌های کاهش بُعد به ما اجازه می‌دهند که علاوه بر صرفه‌جویی در مصرف حافظه و زمان، پیشنهادهایی با دقت بالا به کاربر ارائه دهیم [۲].

الگوریتم پیشنهادی در این مقاله، ترکیب پالایش مشارکتی مبتنی بر حافظه و پالایش مشارکتی مبتنی بر مدل با استفاده از تجزیه‌های ماتریسی است. به‌طور خاص، روش SDD که پیش از این در این کاربرد بررسی نشده است، مورد تحلیل قرار قرار گرفت. پس از فراخوانی داده‌ها و تشکیل ماتریس کاربر–کالا عناصر خالی را با میانگین هر ستون پر و آن‌را نرمال کردیم. در مرحلهٔ بعد با استفاده از روش‌های کاهش بُعد SVD و SDD،T تقریبی از ماتریس با رتبهٔ پایین‌تر را به‌دست می‌آوریم، بنابراین، یک تقریب اولیه برای عناصر خالی ماتریس کاربر–کالا محاسبه می‌شود. سپس با به‌کارگیری الگوریتم نزدیک‌ترین همسایگی روی ماتریس کاهیده، عناصر خالی ماتریس کاربر–کالا را

پیش‌بینی می‌کنیم [۱]. شباهت بین کاربران در ماتریس کاهیده با استفاده از روش همبستگی پیرسن محاسبه می‌شود و پیش‌بینی صورت می‌گیرد. در آخر میانگین هر ستون که به‌منظور نرمال‌کردن از آن کم شده بود را به ماتریس پیش‌بینی اضافه می‌کنیم و پیشنهاد مناسب به کاربر ارائه می‌شود.

۵ ارزیابی نتایج

در این بخش، الگوریتم پیشنهادی با استفاده از نرم‌افزار متلب[1] و داده‌های مووی‌لنز پیاده‌سازی می‌شود. برای این کار بخشی از مجموعه‌داده‌های مووی‌لنز را که معمولاً در سامانه‌های پیشنهادگر به‌کار می‌رود، انتخاب می‌کنیم.

مجموعه‌داده‌های مووی‌لنز به دو مجموعه‌دادهٔ آموزشی[2] و آزمون[3] تقسیم می‌شوند. یک گروه پژوهشی موسوم به گروپ‌لنز[4] نسبت‌های متفاوتی را برای تقسیم‌بندی داده‌های آموزش و آزمون فراهم کرده است. چندین مجموعه‌دادهٔ متفاوت مووی‌لنز با عناوین 1M 100K و 100K برای موضوع‌های پژوهشی منتشر شده است که در اینجا از بخشی از مجموعهٔ 100K استفاده می‌کنیم [۱۳]. این مجموعه شامل ۱۰۰۰۰۰ امتیاز از ۹۴۳ کاربر به ۱۶۸۲ فیلم است. هر کدام از کاربران حداقل به ۲۰ فیلم امتیاز داده‌اند و ژانرهای این مجموعه‌داده با سامانهٔ IMDB[5] مطابقت دارد. ماتریس امتیازها، R از مرتبهٔ ۹۴۳×۱۶۸۲، شامل امتیازهای ۱ تا ۵ است. این ماتریس به‌شدت تُنک است به‌طوری‌که تقریباً ۶٪ از عناصر ماتریس پر است.

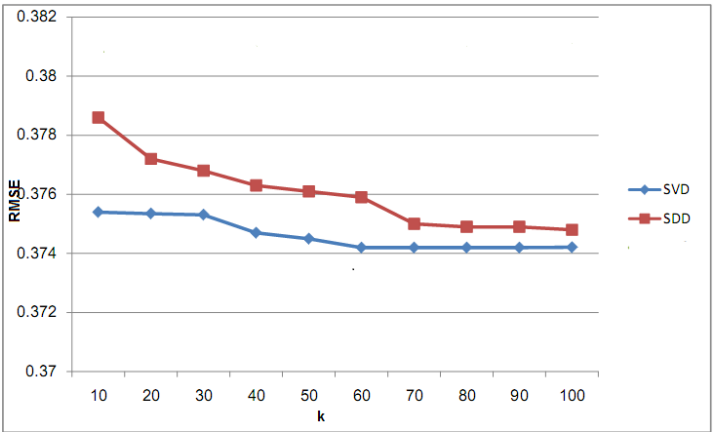
در ادامه عملکرد SVD و SDD را براساس خطا، زمان و حافظهٔ مصرفی آن‌ها مقایسه می‌کنیم. با توجه به شکل ۱، کمترین مقدار خطای پیش‌بینی مربوط به SVD است هر چند SDD نیز خطایی نزدیک به آن دارد. زمان اجرای الگوریتم‌ها برحسب ثانیه به‌ازای مقادیر مختلف $k =$ ۶۰, ۷۰, ۸۰, ۹۰, ۱۰۰ در جدول زیر آمده است

روش	$k =$ ۶۰	$k =$ ۷۰	$k =$ ۸۰	$k =$ ۹۰	$k =$ ۱۰۰
SVD	۱۴۷٫۹۰۱۰	۱۴۶٫۹۰۰۶	۱۴۹٫۲۱۴۷	۱۴۷٫۷۱۵۷	۱۴۹٫۰۸۴۳
SDD	۱۴۵٫۱۸۳۰	۱۴۷٫۶۲۹۶	۱۴۵٫۶۴۴۳	۱۴۸٫۷۴۰۲	۱۴۶٫۲۳۹۵

مشاهده می‌کنیم که زمان مورد نیاز برای روش‌های SVD و SDD به هم نزدیک است. تعداد درایه‌های ناصفر در عامل‌های تجزیه برای $k =$ ۶۰, ۷۰, ۸۰, ۹۰, ۱۰۰ در جدول زیر آمده است. نتایج نشان می‌دهد فضای مورد نیاز برای SDD با توجه به تُنکی عامل‌های آن، کمتر از SVD است.

روش	$k =$ ۶۰	$k =$ ۷۰	$k =$ ۸۰	$k =$ ۹۰	$k =$ ۱۰۰
SVD	۷۲۰۶۰	۸۴۰۷۰	۹۶۰۸۰	۱۰۸۰۹۰	۱۲۰۱۰۰
SDD	۱۹۹۹۷	۲۴۰۳۷	۲۶۸۸۷	۲۹۶۵۳	۳۳۴۶۷

[1] MATLAB [2] training data [3] test data [4] http://www.grouplens.org [5] http://www.imdb.com



شکل ۱. نمودار مقایسه RMSE روش‌های کاهش بُعد به ازای مقادیر مختلف k

۶ نتیجه‌گیری

در این مقاله، ابتدا روش‌های سامانه‌های پیشنهادگر را بیان کردیم. از میان این روش‌ها، روش پالایش مشارکتی کاربرد بیشتری نسبت به سایر روش‌ها دارد. بااین‌حال، این روش با چالش‌های زیادی نیز روبه‌رو است. یکی از مهم‌ترین چالش‌ها، پراکندگی داده‌ها است که ارائه پیشنهاد به کاربر را دچار مشکل می‌کند. برای بهبود پیشنهاد در این سامانه‌ها، روش‌های کاهش بُعد SVD و SDD را استفاده کردیم. روش‌های کاهش بُعد ما را قادر می‌سازند تا تقریبی با رتبهٔ پایین به منظور مدیریت ماتریس‌های با اندازه‌های بزرگ محاسبه کنیم و عناصر خالی ماتریس کاربر–کالا را پیش‌بینی کنیم. استفاده از این روش‌ها باعث افزایش دقت پیش‌بینی و همچنین کاهش زمان محاسبه می‌شود. این روش‌ها را روی مجموعه‌داده‌های مووی‌لنز پیاده‌سازی کردیم. نتایج حاصل از به‌کارگیری SVD و SDD نشان می‌دهد که روش‌های SVD و SDD خطاهایی نزدیک به هم تولید می‌کنند. برای برخی مقدار k زمان اجرای SVD و برای برخی دیگر زمان اجرای SDD کمتر است و تقریباً زمان اجرا در این دو الگوریتم نزدیک به هم است. ولی یک تفاوت عمده این است که روش SDD در مقایسه با SVD، فضای ذخیره‌سازی کمتری نیاز دارد.

مراجع

[۱] گلپررابوکی، عفت؛ کیاهی، معصومه، بررسی روش‌های ماتریسی در سیستم‌های پیشنهاددهنده، چهل و ششمین کنفرانس ریاضی ایران، دانشگاه یزد، ۱۳۹۴.

[2] Bokde, D., Girase, S., Mukhopadhyay, D., Matrix factorization model in collaborative filtering algorithms: A survey, *Procedia Computer Science*, **49** (2015), 136–146.
[3] Golub, G., Van Loan, C., *Matrix Computation*, 4th ed., JHU Press, 2013.
[4] Herlocker, J., Konstan, J., Borchers, A., Riedl, J, An algorithmic framework for performing collaborative filtering, in *Proceedings of ACM SIGIR '99*, ACM Press, 1999, 230–237.
[5] Jannach, D., Zanker, M., Felfering, A., Friendrich, G., *Recommender Systems: An Introduction*, Cambridge University Press, Cambridge, 2010.
[6] Kolda, T. G., O'Leary, D. P., A semidiscrete matrix decomposition for latent semantic indexing information retrieval, *ACM Trans. Inf. Syst.*, **16** (1998), 322–346.
[7] Kontostathis, A., Pottenger, W. M., A Mathematical View of Latent Smantic Indexing: Tracing Term Co-occurrences, Lehigh University Technical Report, LU-CSE-02-006, 2002.
[8] O'Leary, D. P., Peleg, S., Digital image compression by outer product expansion, *IEEE Trans. Commu*, **31** (1983), 441–444.
[9] Ricci, F., Rokach, L., Shapira, B., *Recommender systems Handbook*, Springer, New York, 2015.
[10] Sarwar, B. M., Konstan, J.A., Borchers, A., Herlocker, J., Miller, B., Riedl, J., Using filtering agents to improve prediction quality in the groupLens research collaborative filtering system, in *Proceedings of CSCW 98*, Seattle Washington USA, ACM, 1998, 345–354.
[11] Sarwar, B. M., Karypis, G., Konstan, J. A., Riedl, J. T., Application of dimensionality reduction in recommender systems, A Case Study in *ACM Web KDD 2000 Web Mining for E-Commerce Workshop*, Boston, MA, 2000.
[12] Schafer, J. B., Frankowski, D., Herlocker, J., Sen, Sh., Collaborative filtering recommender systems, *The Adaptive Web*, **4321** (2007), 291–324.
[13] Spiegel, S., A Hybrid Approach to Recommender Systems Based on Matrix Factorization, Technical University Berlin, Berlin, 2009.
[14] Xue, G. R., Lin, C., Yang, Q., Xi, W., Zeng, H.J., Yu, Y., Chen, Z., Scalable collaborative filtering using cluster-based smoothing, in *Proc of SIGIR*, 2005, 114-121.

عفت گلپر رابوکی: دانشگاه قم، گروه ریاضی
رایانامه: g.raboky@qom.ac.ir
معصومهٔ کیاهی: دانشگاه قم، گروه ریاضی
رایانامه: raheabrisham416@gmail.com

Farhang va Andishe-ye Riyazi
Volume 41, Number 70, Spring/Summer 2022, Pages 97–112
doi: 10.30504/mct.2022.335

Applications of Matrix Factorization in Recommender Systems

E. Golpar-Raboky[1], M. Kyahi[2]

[1,2]Department of Mathematics, University of Qom, Iran

Abstract. Due to the huge amount of information available online, it seems necessary to have a recommender system that automatically and intelligently suggests to users. Recommendation systems constitute a specific type of information filtering technique that attempt to present items according to the interest expressed by a user. Collaborative Filtering (CF) is the process of evaluating information using the opinion of other people and computes recommendations based on the information about similar items or users. One of the challenging problems is the sparsity problem of the user-item matrix. Collaborative recommenders try to capture relationships among user-user or item-item pairs by reducing the dimensionality of the user or rather item space. Matrix factorization techniques are employed to reduce the dimension of the observed dataset. In our work we want to investigate the well-known Singular Value Decomposition (SVD) and Semi Discrete Decomposition (SDD), which factorize a matrix into three low-dimensional matrices. The comparison of the methods shows that although SVD has the better results than SDD, SDD is efficient in terms of time and especially memory by producing an error close to SVD.

Keywords: recommender systems, collaborative filtering, singular value decomposition, dimensionality reduction, semidiscrete decomposition, user-item rating matrix
Article history: Recieved 16 February 2019; Accepted 23 November 2019

[1]g.raboky@qom.ac.ir
[2]raheabrisham416@gmail.com

ISSN 1022-6443/© Iranian Mathematical Society