

یک روش مبتنی بر خوشه بندی برای بهبود کارایی در سیستم های توصیه گر

مشارکت محور

چکیده

در سیستم های توصیه گر مشارکت محور، محصولات به عنوان ویژگی شناخته می شوند و از کاربران درخواست می گردد تا به محصولات خریداری شده، رأی دهند. با آموختن رتبه بندی ارائه شده توسط کاربران، سیستم توصیه گر می تواند محصولات جالبی را به کاربران توصیه نماید. با این حال، معمولاً محصولات بسیار زیادی در تجارت الکترونیک وجود دارد و ممکن است این راه کار چندان کارآمد نباشد، به خصوص در زمان هایی که لازم است هر محصول قبل از ساخت در سیستم توصیه گر قرار گیرد. ما یک روش جدید را پیشنهاد می کنیم که یک الگوریتم خوشه بندی خودکار را برای کاهش ابعاد مرتبط با تعداد محصولات در بردارد. محصولات مشابه در خوشه ای یکسان قرار می گیرند و محصولات نامشابه، در خوشه های متفاوتی قرار می گیرند. کارهای پیشنهادی سپس با نتیجه خوشه بندی انجام می گیرد. در نهایت، انتقال مجدد انجام شده و لیست مرتب شده از محصولات پیشنهادی به هر کاربر پیشنهاد می گردد. با روش پیشنهادی، زمان پردازش برای تهیه پیشنهاد کاهش می یابد. نتایج آزمایش ها، نشان می دهد که کارایی سیستم های پیشنهادی می تواند، بدون به خطر انداختن کیفیت توصیه ها بهبود یابد.

کلمات کلیدی: سیستم توصیه کننده فیلتر مشارکتی، نمودار همبستگی، خوشه سازی خودساز، کاهش ابعاد، الگوریتم

رتبه بندی

1. مقدمه

با توجه به توسعه سریع تجارت الکترونیکی، امروزه تعداد زیادی خریدار و فروشنده آنلاین وجود دارد و مقدار زیادی محصولات وجود دارد که کاربران می‌توانند برای خرید انتخاب کنند. به هر حال، وظیفه بررسی و انتخاب محصولات مناسب از جمله تعداد زیادی از محصولات، نه تنها گیج‌کننده، بلکه وقت‌گیر است. سیستم‌های توصیه‌گر [1،2] برای کمک به مردم در یافتن محصولات جالب‌توجه و ذخیره زمان جستجوی محصولات کمک می‌کنند. برای یک کاربر، چنین سیستمی می‌تواند از تجربیات ذخیره‌شده برای تمام مشتری‌ها و توصیه‌های یک لیست اولویت از محصولات به کاربران استفاده کنند. در طول چند سال گذشته، سیستم‌های پیشنهاددهنده به سرعت در حال تحول هستند. بسیاری از سیستم‌های توصیه‌گر توسعه یافته‌اند.

در اصل می‌توان آن‌ها را در دو دسته طبقه‌بندی کرد: محتوا محور و مشارکت محور. اگر چه گرایش به سمت سیستم‌های ترکیبی [3] در سال‌های اخیر افزایش یافته است. سیستم توصیه‌گر مبتنی بر محتوا [4] به کاربر بر اساس محتوایی که ممکن است شامل دسته‌ها و یا ویژگی‌های دیگر محصولات باشد، پیشنهاد می‌دهد. همچنین ممکن است به عادات، منافع و یا تنظیمات کاربران مراجعه کند. با تجزیه و تحلیل این داده‌ها با برخی از فناوری‌ها از جمله مدل‌سازی بیزین [8،9]، محتوای مبتنی بر سیستم پیشنهاددهنده جذاب تر خواهد بود. به طور کلی، سیستم‌های مبتنی بر محتوا نیاز به اطلاعات دقیق در مورد محصولات و کاربران ندارند. محصولات جدید را نیز می‌توانند به کاربران توصیه نمایند. با این حال، رسیدن به اطلاعات مورد نیاز سخت و یا زیاد است. محصولات یا کاربران دارای ویژگی‌های خود هستند. مشکل جمع‌آوری ویژگی‌ها از تمام محصولات و کاربران در هر صورت وجود دارد. علاوه بر این، اطمینان از اینکه یک محصول یا یک کاربر، را می‌توان با داده‌های جمع‌آوری شده نمایش داد، مشکل است. یک سیستم توصیه‌گر مشارکت محور، نیازی به جزئیات اطلاعات درباره ویژگی‌های محصولات یا کاربران ندارد. به جای آن، آن با اجرای تعاملات بین اطلاعات کاربران و محصولات توصیه می‌کند. همواره، اطلاعات تعامل به عنوان رتبه‌بندی کاربر برای محصولات خریداری شده بیان می‌شود. با یادگیری این رتبه‌بندی، چنین سیستم پیشنهاددهنده‌ای، می‌تواند محصولات

را بر اساس نظرات ثبت شده سایر کاربران و ترجیحات خود کاربر مدنظر انجام دهد. به طور کلی، سیستم های توصیه گر مشارکت محور، ساده و موثر هستند و در جامعه تجارت، جذاب تر و عملی تر هستند.

معمولاً محصولات بسیار زیادی برای بررسی در سیستم های توصیه گر وجود دارند و این امر می تواند بسیار ناکارآمد باشد، اگر هر محصول نیاز داشته باشد قبل از دریافت توصیه بررسی گردد. روش کاهش ابعاد برای تولید سریع توصیه باکیفیت بالا و مجموعه هایی با مقیاس بزرگ استفاده می گردد. در [26]، یک روش K-means به نام الگوریتم خوشه بندی K-means برای گروه بندی کاربران درون گروه های متفاوت ارائه شده است. همسایگان هر کاربر به دنبال تقسیم بندی کاربران، برای دریافت توصیه از سیستم توصیه گر از نتایج سایر کاربران هم گروه با خود بهره می گیرند. ژو و همکارانش [27] یک رویکرد مبتنی بر Smooth ارائه کردند که از خوشه های تولید شده از داده های آموزشی برای تهیه خوشه ها و همسایگان هر خوشه بهره می گیرد. Honda و همکارانش [28]، یک روش خوشه بندی برای انتخاب همسایگان بر اساس یک نظریه تعادل ساختاری ارائه کردند. کاربران و محصولات بر اساس متعادل کردن گراف ترکیبی، به خوشه هایی تقسیم می شوند.

Be و همکارانش [29]، کاربران را در خوشه هایی مطابق با صفات آنان خوشه بندی کردند، به عنوان مثال، بر اساس جنسیت، سن و شغل. سپس ماتریس امتیاز کاربر به هر کالا را تجزیه و تحلیل کرده و شباهت کاربران به یکدیگر را محاسبه کردند. سیستم iExpand [30]، مشارکت محور بودن سیستم های توصیه گر را توسط علایق کاربران و رتبه بندی شخصی هر کاربر انجام داد. معرفی یک طرح سه لایه ای، برای کمک به فهم تعاملات بین کاربران، محصولات و کاربران توسط آنان انجام شد. LDA، برای تقسیم بندی محصولات به خوشه هایی ارائه گردید. در [31]، خوشه بندی فازی در کاربران، به تشکیل گروه های کاربران منجر شد. یک بردار خصوصیت گروهی برای هر کاربر ساخته می شود. هر بردار خصوصیت کاربر را می توان به عنوان یک بردار نمایش دهنده میزان اهمیت و کارایی محصولات برای آن کاربر در نظر گرفت. Sarwat و همکارانش [32]، سیستم توصیه گری را پیشنهاد دادند که از یک طبقه بندی کننده با سه نوع نرخ مبتنی بر مکان استفاده می کند. در PRM2 [33]، منابع شخصی، شباهت علایق فردی و نفوذ فردی تجمع

شده و توسط روش SVD، کاهش بعد یافته و مورد استفاده قرار می‌گیرند. BiFu [34] یک مفهوم از محصولات محبوب را شناسایی کرده و برای توصیه در اختیار سیستم های توصیه گر قرارداد.

برای کاهش ابعاد ماتریس رتبه‌بندی، K-means [35] برای گروه‌بندی کاربران و محصولات درون خوشه‌هایی استفاده شده است. همچنین از روش smoothing and fusion برای غلبه بر پراکندگی داده‌ها و تنوع امتیاز استفاده می‌گردد. ICRRS [36]، یک الگوریتم تکراری است که نه تنها مبتنی بر ارزیابی، بلکه مبتنی بر مقایسه و تقریب زنی است. سیستم های توصیه گر مبتنی بر کاهش ابعاد ذکر شده در بالا، دارای برخی معایب هستند. برخی از سیستم ها [29، 32]، نیاز به ویژگی‌های اضافی دیگری در مورد کاربران و یا محصولات دارند. این ویژگی‌ها معمولاً برای دستیابی به کاربرد عملی ضروری است. به عنوان مثال، [26]، [27] و [31]، نیاز به خوشه‌های بیشتری برای پیشبرد هدف دارند. همچنین شباهت اندازه‌گیری شده، اکثراً برای سیستم های اتخاذ شده در زمینه کاهش ابعاد مورد استفاده قرار می‌گیرد. نادیده گرفتن واریانس خوشه‌بندی، ممکن است منجر به نتایج نادرستی گردد. در این مقاله، یک رویکرد مبتنی بر خوشه‌بندی خودساخته برای اعمال پیشنهاد با هدف کاهش ابعاد مرتبط با تعداد محصولات ارائه شده است. محصولات مشابه در هر خوشه‌بندی، برای انجام توصیه در همان خوشه مورد استفاده قرار می‌گیرند. نمودار همبستگی نشان می‌دهد که ارتباط متقابل میان گروه محصولات ایجادشده ضروری و مهم است. الگوریتم random walk، اجراشده و لیست اولویت‌های محصولات، گروه‌بندی شده و مورد استفاده قرار می‌گیرند. پس از آن، لیست اولویت، مجدداً به لیست اولویت‌های فردی از محصولات تبدیل می‌گردد. در نهایت، با روش پیشنهادی، خوشه‌بندی کاملاً بر اساس ماتریس اولویت دهی بدون نیاز به جمع‌آوری صفات اضافی درباره محصولات و کاربران انجام می‌گیرد. خوشه‌ها به صورت خودکار تولید شده و یک تعداد خوشه از پیش تعیین شده توسط کاربر ایجاد می‌گردد. علاوه بر این، در هنگام نیاز به اندازه‌گیری شباهت برای خوشه‌بندی، ما از واریانس مراکز خوشه‌ها استفاده می‌کنیم. با توجه به کاهش ابعاد در تعداد محصولات، زمان پردازش برای ساخت توصیه توسط رویکرد ما بسیار کاهش می‌یابد. نتایج تجربی نشان می‌دهد که بهره‌وری سیستم پیشنهادی را می‌توان بدون کاهش کیفیت، تا حد زیادی بالا برد. ادامه مقاله به شرح زیر سازمان‌دهی شده است. مسئله‌ای که باید حل شود، در بخش 2 نشان داده شده است. یک سیستم توصیه گر مشارکت

محور، ItemRank [17]، در بخش 3 معرفی شده است. روش پیشنهادی ما برای بهبود کارایی ItemRank، با جزئیات در بخش 4 ارائه شده است. یک مثال در بخش 5 نشان داده شده است. نتایج آزمایش‌ها در بخش 6 نمایش داده شده است. در نهایت، بخش پایانی، نتیجه‌گیری را نشان می‌دهد.

2. بیان مسئله

فرض کنید یک مجموعه از N کاربر به صورت $1 < i < N$ وجود دارد و یک مجموعه از M محصول p_j ، $1 < j < M$ وجود دارد. یک کاربر u_i ، ارزیابی خود از محصولات p_j را با تهیه یک نرخ r_{ij} نمایش می‌دهد. همواره، یک مقدار بزرگ‌تر برای نمایش یک بازخورد از کاربر ضروری است. اگر کاربر u_i ، اولویت و یا امتیازی را برای محصول p_j وارد نکند، $r_{ij}=0$. چنین اطلاعاتی را می‌توان توسط ماتریس R زیر نمایش داد.

$$R = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_N \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1M} \\ r_{21} & r_{22} & \dots & r_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ r_{N1} & r_{N2} & \dots & r_{NM} \end{bmatrix} \quad (1)$$

که یک ماتریس N در M است. توجه داشته باشید که

$$R_i = [r_{i1} \quad r_{i2} \quad \dots \quad r_{iM}], 1 \leq i \leq N.$$

برای راحتی، هر سطر، یک رکورد کاربر نامیده می‌شود و هر ستون، نشانگر یک محصول است. هدف از سیستم توصیه گر مشارکت محور، رتبه‌بندی ماتریس داده است که در آن یک لیست اولویت پیش‌بینی شده از محصولات به هر کاربر تهیه می‌شود.

3. ItemRank

ItemRank [17] یکی از روش‌های اساسی برای مشارکت بودن سیستم های توصیه گر است. آن یک الگوریتم امتیازدهی مبتنی بر random-walk، برای پیشنهاد محصولات مطابق با اولویت کاربر استفاده می‌کند. ItemRank

در مطالعات ما استفاده شده است، زیرا دارای پیچیدگی کمتر و کارایی بالاتر در مصرف پول و هزینه محاسباتی نسبت به سایر سیستمها است. ماتریس اولویت نمایش داده شده در رابطه (1)، دارای فرآیند ItemRank با دو گام است، ایجاد گراف همبستگی و random walk [38]. در گام ایجاد گراف همبستگی، یک گراف همبستگی ساخته می‌شود. هر محصولی به عنوان یک محصول در داخل گراف است و یک گراف همبستگی از اولویت‌های داده شده ساخته می‌شود. هر محصولی، به عنوان یک نود در گراف است. یال بین هر دو نود، P_i و P_j ، دارای وزن W_{ij} است که نشانگر تعداد کاربرانی است که به هر دو محصول، امتیاز یکسانی داده بودند. توجه داشته باشند یک کاربر U_k ، اولویت هر دو محصول را تعیین می‌کند. زمانی که گراف همبستگی کامل شد، ماتریس همبستگی زیر ایجاد می‌گردد:

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1M} \\ w_{21} & w_{22} & \dots & w_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ w_{M1} & w_{M2} & \dots & w_{MM} \end{bmatrix} \quad (2)$$

که یک ماتریس M در M است. هر ستون W ، نرمال سازی می‌گردد. در مرحله random walk، یک سری از random walk ها اجرا می‌گردد. برای بررسی وضعیت هر کاربر، $S^i(0)$ را داریم:

$$S_i(0) = \left[1/M \quad 1/M \quad \dots \quad 1/M \right]^T \quad (3)$$

که برداری با سائز M است. عملیات زیر

$$S_i(t+1) = \alpha W S_i(t) + (1 - \alpha) R_i^T \quad (4)$$

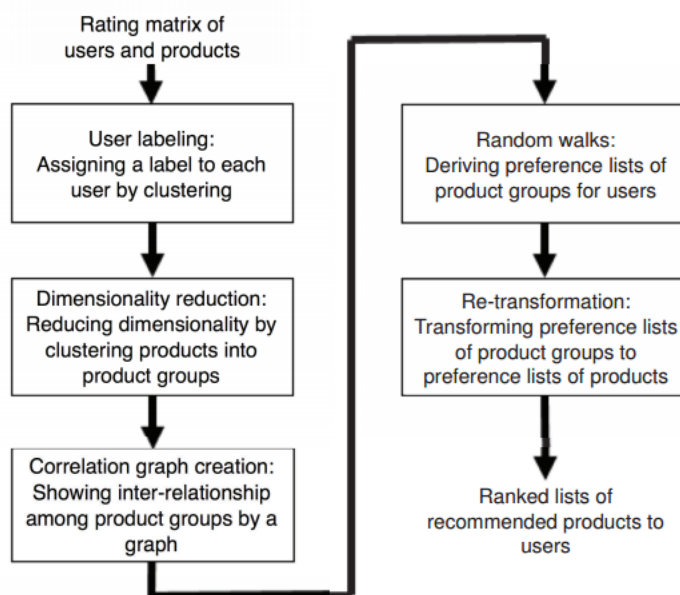
مکرراً برای $t=0,1,2,\dots$ اجرا می‌گردد تا همگرایی ایجاد گردد. توجه داشته باشید که $\alpha \in [0, 1]$. همواره، یک همگرایی قابل قبول، بعد از تقریباً 20 تکرار ایجاد می‌گردد. بنابراین، ضروری است که رابطه (4) را حدود 20 بار برای هر کاربر اجرا کنیم. اگر S^i بردار همگرایی باشد، اولویت پیشگویی محصولات کاربر U_i قابل محاسبه است. سپس می‌توان محصولاتی را برای کاربر U_i توصیه کرد. محصولات متعلق به بزرگ‌ترین عنصر در S_i ، اولین توصیه است و دومین عنصر بزرگ در S_i ، دومین توصیه است.

4. روش پیشنهادی

Itemrank مسئله کارایی را مدنظر قرار می‌دهد. زیرا ممکن است تعداد زیادی محصولات در تجارت الکترونیک وجود داشته باشد. ماتریس W ، که دارای سایز $M * M$ است، می‌تواند شدیداً بزرگ باشد. تکثیر W با $S_i(t)$ در رابطه (4)، زمان زیادی را می‌گیرد و روش ItemRank را برای مقیاس بزرگ مسئله ناکارآمد می‌کند. ما یک الگوریتم خوشه‌بندی خودساخته (SCC)، [39،40] اجرا کرده‌ایم که عمل کاهش بعد را برای تولید خوشه‌ها استفاده می‌کند. کار پیشنهادی سپس با خوشه‌بندی مجدد ادامه می‌یابد. در نتیجه، کارایی ItemRank، می‌تواند بهبود یابد. در مقایسه با سایر روش‌های کاهش بعد [41، 42، 43، 44]، SCC دارای مزایای زیادی است. خوشه‌بندی به صورت اتوماتیک انجام می‌گیرد و تعیین تعداد خوشه‌ها توسط کاربر، مورد نیاز نیست. در کنار آن، زمانی که شباهت خوشه‌ها اندازه‌گیری گردید، هر دوی مراکز و واریانس خوشه‌ها محاسبه می‌گردند. در پایان، اندازه‌گیری شباهت بهتر از بررسی مراکز در سایر روش‌ها است.

روش ما شامل پنج گام است. برچسب‌گذاری کاربر، کاهش بعد، ایجاد گراف همبستگی، random walk و انتقال مجدد که در شکل 1 نشان داده شده است. در کام برچسب‌گذاری کاربر، SCC، برای نگاشت برچسب کلاس به کاربر برای کمک به گام دوم انجام می‌گیرد تا مرحله دوم بتواند کار خود را به طور کارا انجام دهد. در گام کاهش بعد، SCC، مجدداً برای خوشه‌بندی تعدادی از گروه محصولات استفاده می‌گردد. محصولات مشابه، به گروه محصولات یکسانی تعلق دارند و محصولات غیرمشابه، به محصولات متفاوتی تعلق دارند. از آنجایی که تعداد گروه محصولات کوچک‌تر از تعداد محصولات است، ابعاد درگیر کاهش خواهد یافت. سپس یک گراف همبستگی که نشانگر رابطه داخلی بین گروه محصولات هست، ایجاد می‌گردد. بر اساس گراف همبستگی، یک سری از random walk ها اجرا شده و یک لیست اولویت از گروه محصولات برای هر کاربر در گام چهارم ایجاد می‌گردد. در نهایت، در گام انتقال مجدد، لیست اولویت گروه محصولات به لیست اولویت تکی محصولات منتقل می‌گردد و یک لیست رتبه‌بندی شده از محصولات برای هر کاربر ایجاد می‌گردد.

در این مطالعه، ما ItemRank [17] را به عنوان هدف نهایی استفاده کرده‌ایم. به هر حال، روش ما می‌تواند با سایر روش‌ها برای کاهش ابعاد و بهبود کارایی آن‌ها کار کند.



شکل 1. بررسی اجمالی از رویکرد ما.

4.1 خوشه‌بندی خودساخته (SCC)

مجموعه X با n پارامتر $x_1, x_2, \dots, x_n$ با $\mathbf{x}_i = \langle x_{i1}, x_{i2}, \dots, x_{ip} \rangle$ برای $1 < i < n$ ، نشان داده شده است.

هدف الگوریتم SCC، گروه‌بندی این الگوها درون یک مجموعه از کلاسترها است، تا الگوهای یکسان در یک گروه و الگوهای غیریکسان درون گروه‌های مختلفی قرار گیرند. اگر K ، تعداد خوشه‌های جاری موجود با نام $G_1, G_2, \dots, \text{and } G_K$ باشد. هر خوشه G_j ، دارای میانه $\mathbf{m}_j = \langle m_{j1}, m_{j2}, \dots, m_{jp} \rangle$ و واریانس

$\sigma_j = \langle \sigma_{j1}, \sigma_{j2}, \dots, \sigma_{jp} \rangle$ است که حاصل میانگین و واریانس تمام الگوهای موجود است. اگر S_j ، سایز کلاستر G_j باشد، تعداد الگوهای شامل G_j خواهد بود. بنابراین ما داریم $K=0$ و این مورد نشانگر عدم وجود هیچ خوشه‌ای است.

برای هر الگوی x_i ، $1 < i < n$ ، ما اعضای آن با درجه x_i را محاسبه کرده‌ایم که درون خوشه $\mu_{G_j}(x_i)$ وجود دارد.

$$\mu_{G_j}(\mathbf{x}_i) = \prod_{q=1}^p \exp \left[- \left(\frac{x_{iq} - m_{jq}}{\sigma_{jq}} \right)^2 \right] \quad (5)$$

$$\mu_{G_j}(\mathbf{x}_i) \geq \rho \quad (6)$$

که در آن $0 \leq \rho \leq 1$ ، یک حد آستانه است. توجه داشته باشید که علامت "رو" نقش مهمی را در این الگوریتم خوشه‌بندی بازی می‌کند. مقدار بزرگ برای علامت "رو" در خوشه‌های کوچک‌تر استفاده می‌شود، اگر چه مقدار کوچک آن در خوشه‌های بزرگ است. همان طور که "رو" افزایش می‌یابد، تعداد خوشه‌ها نیز افزایش می‌یابد. دو مورد ممکن است رخ دهد. اولاً، هیچ خوشه‌ای بر روی X_i نباشد. در این مورد، ما فرض می‌کنیم که X_i مشابه با سایر خوشه‌ها نیست و خوشه جدیدی با $h=K+1$ ایجاد می‌گردد:

$$\mathbf{m}_h = \mathbf{x}_i, \quad \sigma_h = \sigma_0 \quad (7)$$

به طوری که در آن $\sigma_0 = \langle \sigma_0, \sigma_0, \dots, \sigma_0 \rangle$ یک بردار ثابت تعریف شده توسط کاربر است. البته، تعداد خوشه‌ها افزایش می‌یابد و سائز خوشه G_h و S_h باید برابر با یک باشد.

$$K = h, \quad s_h = 1. \quad (8)$$

دوماً، خوشه‌هایی که در X_i وجود دارد، دارای تست مشابهی هستند. در این مورد، خوشه G_t خوشه‌ای با بزرگ‌ترین اعضا است.

$$t = \arg \max_{1 \leq j \leq K} (\mu_{G_j}(\mathbf{x}_i)). \quad (9)$$

X_i باید مشابه با خوشه G_t باشد و m_t و G_t باید تغییر یابند. تغییرات خوشه G_t به صورت زیر هستند:



$$\sigma_{tj} = \sqrt{A - B} + \sigma_0, \quad (10)$$

$$A = \frac{(s_t - 1)(\sigma_{tj} - \sigma_0)^2 + s_t \times m_{tj}^2 + x_{ij}^2}{s_t}, \quad (11)$$

$$B = \frac{s_t + 1}{s_t} \left(\frac{s_t \times m_{tj} + x_{ij}}{s_t + 1} \right)^2, \quad (12)$$

$$m_{tj} = \frac{s_t \times m_{tj} + x_{ij}}{s_t + 1} \quad (13)$$

for $1 \leq j \leq p$, and

$$s_t = s_t + 1. \quad (14)$$

توجه داشته باشید که K، در این مورد تغییری نمی‌کند. این فرایند تا زمانی که الگوها قابل اجرا باشند، ادامه می‌یابد. مطابق با آن، ما K خوشه برای X محاسبه کرده‌ایم.

4.2 گام 1: برچسب‌گذاری کاربر

برای کاهش ابعاد به طور موثر، نیاز به تعریف برچسب کلاس برای کاربران داریم. ایده، گروه‌بندی کاربران درون خوشه‌هایی است [32]. کاربران مشابه، درون خوشه‌هایی گروه‌بندی شده و کاربران غیرمشابه، درون خوشه‌های مختلفی قرار می‌گیرند. سپس تمام کاربران در یک گروه، برچسب یکسانی را دریافت می‌کنند. ما از الگوریتم SCC برای این هدف استفاده کردیم. الگوریتم‌های خوشه‌بندی دیگری [35، 46، 45] نیز می‌توانند این کار را انجام دهند. اما، آن‌ها نیاز دارند تا ای پیش در مورد تعداد کلاس‌ها تصمیم‌گیری کنند. با الگوریتم SCC، ما فقط نیاز داریم تا برخی ثوابت معنادار را در طول فرآیند خوشه ایجاد کنیم.

برای اعمال SCC، ما شباهت ایجادشده میان کاربران را به صورت رتبه‌بندی شده ایجاد کردیم. با این حال، مردم دارای شخصیت‌های مختلفی هستند و برای رأی دادن به محصولات از سلاقی مختلف خود استفاده می‌کنند. در واقع برخی سخاوتمند بوده و نمرات بیشتری را در نظر می‌گیرند و در مقابل برخی سخاوتمند نیستند و نمرات کمتری را در نظر می‌گیرند. حال می‌توان با توجه به رتبه‌بندی یک کاربر، برای دیگران توصیه در نظر گرفت. فرض کنید که دو کاربر شبیه به یکدیگر هستند و شکل موج ایجادشده توسط رتبه‌بندی آن‌ها یکسان است. بنابراین ما برای رتبه دهی به کاربران از رابطه زیر استفاده می‌کنیم:

$$Q_i = \sum_{k=1}^M r_{ik},$$

$$x_{ij} = \frac{r_{ij}}{Q_i}, 1 \leq j \leq M \quad (15)$$

برای $1 < i < N$ داریم، $\mathbf{x}_i = \langle x_{i1}, x_{i2}, \dots, x_{iM} \rangle$, $1 \leq i \leq N$, and $\mathbf{X} = \{\mathbf{x}_i | 1 \leq i \leq N\}$.

ما الگوریتم SCC را بر روی $\mathbf{X}$ اعمال می‌کنیم. فرض کنید که z کلاستر $G_1, G_2, \dots, G_z$ وجود داشته باشد. هر کلاستر به عنوان یک کلاس است و ما z کلاس داریم که دارای برچسب‌های $c_1, c_2, \dots, c_z$ هستند. برای تمام کلاسترها در G_z ، $1 < z < Z$. برچسب کلاس c_j را داریم. در نتیجه، ما مجموعه R را با مجموعه دیگر R' که شامل N مؤلفه است ایجاد

کرده‌ایم: $(\mathbf{R}_1, y_1), (\mathbf{R}_2, y_2), \dots, (\mathbf{R}_N, y_N)$ where $y_i \in \{c_1, c_2, \dots, c_z\}$.

4.3 کاهش ابعاد

در این گام، ما ابعاد M را کاهش دادیم که مرتبط با محصولات مشابه بود. برای هر محصول p_j ، $1 < j < M$ است. ما یک

$$\mathbf{x}_j = \langle x_{j1}, x_{j2}, \dots, x_{jz} \rangle$$

الگوی ویژگی ایجاد کردیم.

$$x_{jk} = P(c_k | p_j) = \frac{\sum_{d=1}^N r_{dj} \times \delta_{dk}}{\sum_{d=1}^N r_{di}}, 1 \leq k \leq z \quad (16)$$

for $1 \leq j \leq M$, where δ_{dk} is defined as

$$\delta_{dk} = \begin{cases} 1, & \text{if } y_d = c_k; \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

$$\mathbf{Y} = \{\mathbf{x}_i | 1 \leq i \leq M\}.$$

بنابراین، ما ویژگی $x_1, x_2, \dots, x_M$ را برای هر مؤلفه z ایجاد کردیم. داریم

در نهایت، ما الگوریتم SCC را بر روی $\mathbf{Y}$ اجرا کردیم. فرض کنید که q خوشه ایجاد کرده‌ایم. توجه داشته باشید که محصولات شامل خوشه‌های مشابه هستند. این قابل قبول است که خوشه‌ها مسئول تمام محصولات هستند. از آنجا که در آن، q کلاستر وجود دارد، یک کاربر باید M مؤلفه را انتخاب کند. ما می‌توانیم ابعاد M را کاهش داده و یک q با ابعاد کوچک‌تر داشته باشیم. T ماتریس کاهش بعد یافته است:

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1q} \\ t_{21} & t_{22} & \dots & t_{2q} \\ \vdots & \vdots & \vdots & \vdots \\ t_{M1} & t_{M2} & \dots & t_{Mq} \end{bmatrix} \quad (18)$$

where

$$t_{ij} = \mu_{G_j}(\mathbf{x}_i) \quad (19)$$

که درجه عضویت $\mathbf{x}_i$ را در خوشه G_j در رابطه (5) نمایش می‌دهد. سپس ما R کاهش بعد یافته را درون یک ماتریس $N \times M$ قراردادیم تا

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_N \end{bmatrix} = \mathbf{R}\mathbf{T} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \vdots \\ \mathbf{R}_N \end{bmatrix} \mathbf{T} \quad (20)$$

که در آن $\mathbf{B}$ یک ماتریس $N \times q$ است. توجه داشته باشید که

$$\mathbf{B}_i = [b_{i1} \quad b_{i2} \quad \dots \quad b_{iq}] \quad (21)$$

برای سهولت، ما ستون $\mathbf{B}$ را به عنوان یک گروه محصولات در نظر گرفته‌ایم. بنابراین، ما q گروه از محصولات را داریم که با نام $g_1, g_2, \dots, \text{and } g_q$ به ترتیب نشان داده شده‌اند. در این راستا، کاربر باید M مؤلفه را ذخیره کند و هر مؤلفه متعلق به یک محصول است. Itemrank با ماتریس R کار می‌کند. اما روش ما، با ماتریس $\mathbf{B}$ کار می‌کند. از آنجایی که q ، همواره از M بزرگ‌تر است، ابعاد می‌توانند کاهش یافته و به طور چشمگیری بهبود یابند.

4.4 گام 3: ایجاد گراف همبستگی

در این گام، ما یک گراف همبستگی ایجاد کرده‌ایم که نشانگر روابط بین محصولات گروهی q است [47]. چون ما با $\mathbf{B}$ کار می‌کنیم، گراف ما دارای تفاوت‌هایی خواهد بود. هر گروه از محصولات، به عنوان یک گره هستند و بنابراین، ما q گره در گراف داریم. وزن w_{ij} بین گره g_i با $1 \leq i < j \leq q$ به صورت زیر تعریف می‌گردد:

calculated by

$$w_{ij} = \begin{cases} 0, & \text{if } i = j \\ \sum_{k=1}^N \text{limit}(\frac{b_{ki}}{b_{kj}}), & \text{otherwise} \end{cases} \quad (22)$$

where

$$\text{limit}(\frac{a_1}{a_2}) = \begin{cases} 0, & \text{if } a_1 = 0 \text{ or } a_2 = 0 \\ \frac{a_1}{a_2}, & \text{if } a_1 < a_2 \\ 1, & \text{otherwise} \end{cases} \quad (23)$$

برای موردی که $a_1 < a_2$ ، داریم: $w_{ij} = \frac{a_1}{a_2} < 1$. زیرا دارای اولویت کمتری برای کاربر است. برای موردی

که $a_1 \geq a_2$ ، ما داریم، $w_{ij} = \frac{a_1}{a_2} \geq 1$ تا کاربر را برای داشتن اولویت بهتر تشویق کنیم. به هر حال، اگر

W_{ij} خیلی بزرگ باشد، برخی از معیارهای گروهی، بر دیگری غلبه خواهند کرد. بنابراین ما یک محدودیت بر روی W_{ij} ایجاد می‌کنیم. زمانی که گراف همبستگی کامل شد، ما ماتریس همبستگی زیر را داریم:

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1q} \\ w_{21} & w_{22} & \dots & w_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ w_{q1} & w_{q2} & \dots & w_{qq} \end{bmatrix} \quad (24)$$

که در آن، ماتریس، یک ماتریس q در q است. هر ستون از W ، نرمال می‌گردد:

$$Q_j = \sum_{k=1}^q w_{kj},$$

$$w_{ij} = \frac{w_{ij}}{Q_j}, \quad 1 \leq i \leq q$$

for $1 \leq j \leq q$.

4.5 گام چهارم: random walk

در این گام، یک سری از الگوریتم‌های random walk [48,38,17,49] ارائه شده است

$$V_i(0) = [1/q \quad 1/q \quad \dots \quad 1/q]^T \quad (25)$$

که دارای سائز برداری برابر با q است.

$$\mathbf{V}_i(t+1) = \alpha \mathbf{W} \mathbf{V}_i(t) + (1 - \alpha) \mathbf{B}_i^T \quad (26)$$

توجه داشته باشید که $\mathbf{W}$ یک ماتریس در رابطه (24) است و $\mathbf{B}_i$ برداری در رابطه (21) است. همانند آنچه در [17] نشان داده شده است، آلفا، در رابطه (4) انتخاب شده تا برابر با 0.85 باشد. $\mathbf{V}_i$ لیست اولویت ایجاد شده برای سهولت گروه بندی است.

4.6 گام پنجم: Re-Transmission

هر بردار $\mathbf{V}_i$ ، برای کاربر u_i است. مقدار q ، نشانگر گروه بندی محصولات است. به هر حال، ما نیاز داریم تا گروه بندی شخصی را برای هر کاربر رعایت کنیم. رابطه (19) دارای درجه x است. بنابراین داریم:

$$Q_k = \sum_{j=1}^M t_{jk},$$

$$t_{jk} = \frac{t_{jk}}{Q_k}, 1 \leq j \leq M$$

برای $1 \leq k \leq q$ است. برای هر سطر، این محاسبات شامل محصولات P_j است. بنابراین داریم:

$$S_i[j] = t_{j1}V_i[1] + t_{j2}V_i[2] + t_{j3}V_i[3] + \dots + t_{jq}V_i[q] \quad (27)$$

که در آن $S_i[j]$ ، زامین مؤلفه S_i است و $V_i[k]$ در آن $1 < k < q$ ، زامین مؤلفه V_i است. توجه داشته باشید که t_{jk} ، محصول مشابه با محصولات p_j است. بنابراین $t_{jk}V_i[k]$ ، اولویت محصول p_j را برای کاربر u_i در مورد گروه محصولات g_k نشان می دهد. در مجموع، نتایج در رابطه (27) نشان داده شده است. نتایج اولویت محصولات p_j برای کاربر u_i است. در نهایت، ما لیست اولویت را در لیست S_i قراردادیم.

4.7 آنالیز پیچیدگی

ما آنالیز پیچیدگی محاسباتی راه کار پیشنهادی را در این بخش انجام داده ایم. در گام برچسب گذاری کاربر، ما شباهت بین کاربرها و خوشه های هر یک را محاسبه کردیم. بازگو می کنیم که N ، تعداد کاربران را نشان می دهد و M ، تعداد

محصولات را نشان می‌دهد. پیچیدگی زمانی این گام برابر است با $O(NzM)$. در گام کاهش ابعاد، ما شباهت بین الگوهای بار و خوشه‌های موجود را بررسی کردیم. زیرا تعداد الگوهای ویژگی برابر با M است و تعداد محصولات گروه‌بندی شده برابر با q است. در گراف همبستگی ایجادشده، هر وزن w_{ij} نشانگر رابطه (22) است.

جدول 1. یک ماتریس رتبه دهی R

Table 1: A user-product rating matrix R

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}
u_1	5	0	2	0	5	2	3	0	4	0	0	5
u_2	0	4	0	3	0	3	3	0	2	0	4	0
u_3	4	0	2	0	4	0	2	2	3	0	0	4
u_4	3	4	0	4	0	0	3	0	2	0	4	0
u_5	0	2	5	3	0	4	0	5	0	4	2	0
u_6	5	0	0	0	4	3	0	2	4	2	0	5
u_7	0	4	0	3	2	2	3	0	0	0	3	0
u_8	2	3	4	0	0	4	0	5	0	4	0	0
u_9	4	0	2	2	4	0	0	0	3	0	0	5
u_{10}	0	2	4	0	0	3	0	4	0	4	2	2

بنابراین، پیچیدگی این مرحله به صورت $O(q^2)$ است. در مرحله random walk، رابطه (26) دارای یک روند تکراری است. برای هر تکرار، $WV_i(t)$ محاسبه می‌گردد. زیرا 20 تکرار قبل از همگرایی لازم است. در نهایت، برای یک کاربر، رابطه (27)، دارای زمان محاسباتی M است. پیچیدگی این مرحله به صورت $O(NqM)$ است. بنابراین پیچیدگی زمانی $O(NzM) + O(Mqz) + O(q^2) + O(Nq^2) + O(NqM)$. Since $z < N$, راه کار ما برای تمام محصولات به صورت است.

5. مثال

در این بخش، ما یک مثال برای نمایش راه کار پیشنهادی ارائه داده‌ایم. ماتریس R مورد بررسی در جدول 1 نشان داده شده است. توجه داشته باشید که 10 کاربر و 12، 10 محصول وجود دارد و $M=12$ ، 65 است. برای مثال، کاربر

u_1 ، دارای 7 نرخ تولید محصول، به صورت $p_1, p_3, p_5, p_6, p_7, p_9, \text{ and } p_{12}$ است. در مرحله برچسب گذاری

کاربر، برچسب C_1 ، به کاربر $u_1, u_3, u_6, \text{ and } u_9$ اعطا می گردد و برچسب کلاس C_2 ، به کاربران

$u_2, u_4, \text{ and } u_7$ ، و برچسب C_3 به کاربران $u_5, u_8, \text{ and } u_{10}$ اعطا می گردد.

جدول 2. ویژگی الگوها در گام دوم

$x_1 = \langle 0.783, 0.130, 0.087 \rangle$	$x_2 = \langle 0.000, 0.632, 0.368 \rangle$
$x_3 = \langle 0.316, 0.000, 0.684 \rangle$	$x_4 = \langle 0.133, 0.667, 0.200 \rangle$
$x_5 = \langle 0.895, 0.105, 0.000 \rangle$	$x_6 = \langle 0.238, 0.238, 0.524 \rangle$
$x_7 = \langle 0.357, 0.643, 0.000 \rangle$	$x_8 = \langle 0.222, 0.000, 0.778 \rangle$
$x_9 = \langle 0.778, 0.222, 0.000 \rangle$	$x_{10} = \langle 0.143, 0.000, 0.857 \rangle$
$x_{11} = \langle 0.000, 0.733, 0.267 \rangle$	$x_{12} = \langle 0.905, 0.000, 0.095 \rangle$

در مرحله کاهش ابعاد، ما الگوهای ویژگی را به صورت نمایش داده شده در جدول 2 در نظر گرفته ایم. پس از اجرای

الگوریتم SCC، سه خوشه G_1, G_2, G_3 به دست می آیند. ماتریس کاهش بعد یافته T به صورت زیر است:

$$= \begin{bmatrix} 0.982 & 0.109 & 0.120 \\ 0.028 & 0.891 & 0.219 \\ 0.091 & 0.100 & 0.963 \\ 0.066 & 1.000 & 0.149 \\ 0.983 & 0.060 & 0.056 \\ 0.126 & 0.382 & 0.821 \\ 0.189 & 0.764 & 0.080 \\ 0.040 & 0.082 & 0.977 \\ 0.945 & 0.140 & 0.083 \\ 0.018 & 0.065 & 0.908 \\ 0.024 & 0.938 & 0.118 \\ 0.938 & 0.039 & 0.076 \end{bmatrix} \quad (28)$$

با اجرای رابطه (20)، ما ماتریس R را کاهش داده و به ماتریس B می رسیم که در جدول 3 نشان داده شده است.

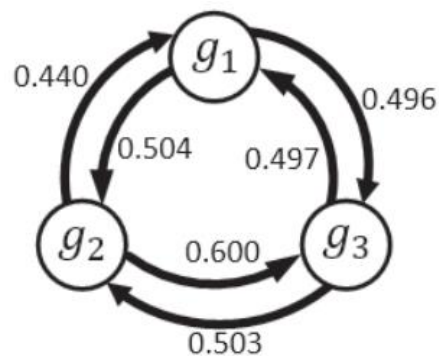
توجه داشته باشید که سه گروه محصول با عناوین g_1, g_2, g_3 وجود دارد. ما گراف همبستگی ایجاد شده در گام قبل را

داریم. با اجرای رابطه (22)، ما گراف همبستگی را همانند آنچه در شکل 2 نشان داده شده است، داریم.

$$W = \begin{bmatrix} 0.000 & 0.440 & 0.497 \\ 0.504 & 0.000 & 0.503 \\ 0.496 & 0.560 & 0.000 \end{bmatrix}. \quad (29)$$

جدول 3. ماتریس کاهش بعد یافته B

	g_1	g_2	g_3
u_1	19.294	4.855	5.395
u_2	3.236	14.031	4.666
u_3	15.086	3.144	5.292
u_4	5.872	14.212	2.711
u_5	1.529	9.353	17.738
u_6	17.806	2.979	7.763
u_7	3.162	12.549	3.673
u_8	3.186	5.487	16.551
u_9	15.697	3.491	3.552
u_{10}	2.952	5.870	14.682



شکل 2. گراف همبستگی

در مرحله random walk، ما رابطه (26) را برای هر کاربر اجرا می‌کنیم. برای نمونه، پس از هر همگرایی، ما

را داریم. در نهایت، در گام re-transformation، ما V_i را به S_i تبدیل کردیم. برای مثال، V_9 به S_9 برای کاربر u_9 تبدیل می‌شود. مؤلفه‌های S_9 در جدول 4 نشان داده شده است.

چون p_6 ، دارای مقدار اولویت بزرگتری است، 0.126 در S_9 و p_6 ، اولین توصیه برای کاربر u_9 و سپس p_4 و p_3 خواهد بود.

جدول 4. مشتق شده S_9 برای کاربر U

p_1	p_2	p_3	p_4	p_5	p_6
0.109	0.108	0.110	0.115	0.099	0.126
p_7	p_8	p_9	p_{10}	p_{11}	p_{12}
0.097	0.105	0.106	0.095	0.102	0.095

6. نتایج آزمایش‌ها

برای ارزیابی کارایی الگوریتم مبتنی خوشه‌بندی خودساخته (SCC)، ما یک مجموعه از آزمایش‌ها را بر روی چندین مجموعه داده انجام دادیم. برای همگرایی، ما روش SCC خود را در این بخش ارزیابی کردیم. ما همچنان، روش SCC را یا سایر راه‌کارهای مبتنی بر فیلترگذاری مقایسه کردیم. سه متریک برای مقایسه بر روی دقت توصیه‌ها ارائه شده است: درجه توافق (DOA)، میانگین خطا و خطای میانی ریشه. یک اعتبارسنجی برای آزمایش‌ها ارائه شده است. در این آزمایش‌ها، ورودی، مجموعه داده‌ها با 5 زیرمجموعه متفاوت است. سپس عملیات پنج بار تکرار می‌گردد. هر زمان، چهار زیرمجموعه از آن پنج زیرمجموعه، به عنوان داده‌های آزمایشی شناخته می‌شوند. اگر P ، مجموعه تمام محصولات باشد، L_i ، مجموعه شامل محصولات کاربر u_i است که دارای مجموعه آموزشی است و T_i ، مجموعه‌ای شامل محصولات کاربر u_i است. DOA به صورت زیر تعریف می‌گردد:

$$DOA = \frac{\sum_{u_i \in U} DOA_i}{|U|} \quad (30)$$

که در آن، U مجموعه تمام کاربران و $|U|$ ، سایز U است و DOA_i ، اندازه‌گیری برای کاربر u_i با رتبه تعیین شده است. DOA_i به صورت زیر محاسبه می‌گردد.

$$check_order_i(p_j, p_k) = \begin{cases} 1, & \text{if } PP_i^j \geq PP_i^k \\ 0, & \text{otherwise} \end{cases} \quad (31)$$

به طوری که در آن، PP_i^k و PP_i^k نشانگر اولویت پیشنهادی برای کاربر u_i و محصول p_j است. سپس DOA_i به صورت زیر فرموله بندی می گردد:

$$DOA_i = \frac{\sum_{p_j \in T_i \wedge p_k \in \mathcal{NW}_i} check_order_i(p_j, p_k)}{|T_i| \times |\mathcal{NW}_i|} \quad (32)$$

where

$$\mathcal{NW}_i = \mathcal{P} - (\mathcal{L}_i \cup T_i) \quad (33)$$

که در آن مجموعه محصولات کاربر u_i ، دارای مجموعه آموزشی تست است. توجه داشته باشید که مقدار بزرگتر DOA_i ، نشانگر اولویت سیستم پیشنهادی است. یک سیستم پیشنهادی دارای DOA برابر با 100٪ است. برای محاسبه MAE یا $RMSE$ ، ما اولویت پیشگویی را به امتیازات پیشگویی تبدیل می کنیم. اجازه دهید، r_{ij} ، نشانگر امتیاز پیشگویی برای اولویت محصول p_i برای کاربر u_j باشد. سپس r_{ij} ، به صورت زیر محاسبه می گردد.

$$\hat{r}_{ij} = r_{a,i} + \frac{\sum_{k=1 \wedge k \neq i}^N [Sim(u_i, u_k) \times (r_{kj} - r_{a,k})]}{\sum_{k=1 \wedge k \neq i}^N Sim(u_i, u_k)} \quad (34)$$

طوری که $r_{a,i}$ ، میانگین موجود در L_i است و میانگین در L_k نیز وجود دارد که با r_{kj} نشان داده شده است.

$$Sim(u_i, u_k) = \frac{\mathbf{S}_i \cdot \mathbf{S}_k}{\sqrt{\mathbf{S}_i \cdot \mathbf{S}_i} \sqrt{\mathbf{S}_k \cdot \mathbf{S}_k}} \quad (35)$$

مجموعه های MAE و $RMSE$ به صورت زیر تعیین می گردند:

$$MAE = \frac{\sum_{i=1}^N \sum_{j=1 \wedge r_{ij} \in T_i}^M |r_{ij} - \hat{r}_{ij}|}{\sum_{i=1}^N |T_i|}, \quad (36)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1 \wedge r_{ij} \in T_i}^M (r_{ij} - \hat{r}_{ij})^2}{\sum_{i=1}^N |T_i|}}. \quad (37)$$

یک مقدار MAE و RMSE نشانگر اولویت یک سیستم توصیه گر است. یک سیستم توصیه گر ایده آل، دارای هیچ خطایی نیست و در آن $MAE=0$ است و $RMSE=0$ است. توجه داشته باشید که در هر یک از آزمایش‌ها زیر، لیست اولویت پیشنهادی از مجموعه آموزشی مشتق شده است. همچنین، هیچ overlap ای بین داده‌های آموزشی و تست نیست.

6.1 مجموعه داده

پنج مجموعه داده، MovieLens، Yahoo Movie، Amazon Video، BookCrossing و Epinions برای آزمایش‌ها این بخش استفاده شده‌اند. مجموعه داده MovieLens [51]، به صورت عمومی موجود است و توسط موسسه تحقیقاتی GroupLens برای اجرا در سیستم‌های توصیه گر ارائه شده است. آن شامل 943 کاربر، 1682 محصول و 100000 رتبه دریافتی از کاربر است. به عبارت دیگر، $N=943$ و $M=1682$ است. هر سطر از مجموعه داده به صورت یک، سه تایی (U_i, p_j, r_{ij}) نشان داده شده است. بنابراین، این مجموعه داده دارای 100000 عنصر است. مجموعه داده Yahoo Movie، شامل اولویت جامعه Yahoo Movie است و شامل 7642 کاربر و 11916 محصول و 221367 اولویت برای هر کاربر است. هر سطر از آن، به صورت یک سه تایی (U_i, p_j, r_{ij}) نشان داده شده است. مجموعه داده ویدئوی آمازون، شامل تقسیم‌بندی ویدئوهای آمازون است که در طی 18 سال، شامل 143.7 میلیون تا سال 2014 است. آن شامل 2978 کاربر، 31102 محصول و 99816 اولویت برای هر کاربر است. هر سطر از آن نشانگر یک سه تایی است که در طی چهار هفته جمع‌آوری شده است. در نهایت مجموعه داده پایانی، شامل 4981 کاربر، 9846 محصول (کتاب) و 109018 اولویت برای هر کاربر است. هر عنصر در این مجموعه داده با یک سه تایی (U_i, P_j, r_{ij}) نمایش داده می‌شود که در آن $r_{ij} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. مجموعه داده Epinions از Paolo massa در طی 5 هفته جمع‌آوری شده است که شامل 2322 کاربر، 4571 محصول و 136984 اولویت برای کاربر است. هر ورودی از مجموعه داده به عنوان یک سه تایی (U_i, p_j, r_{ij}) است که در آن r_{ij} عضو $\{1, 2, 3, 4, 5\}$ است. ویژگی‌های این پنج مجموعه داده در جدول 5 خلاصه شده است.

جدول 5. ویژگی‌های مجموعه داده استفاده شده برای آزمایش‌ها

	# of users	# of products	# of ratings
MovieLens	943	1682	100000
Yahoo Movie	7642	11916	221367
Amazon Video	2978	31102	99816
BookCrossing	4981	9846	109018
Epinions	2322	4571	136984

6.2 نتایج و بحث

ما اثربخشی، دقت و کارایی، راه‌کار پیشنهادی SCC را توسط مقایسه با سایر روش‌های فیلترگذاری شامل Itemrank نشان داده‌ایم. همچنین iExpand [30]، PRM2 [33]، BiFu [34] و ICRRS [36]. همان طور که قبلاً توضیح داده شد، ItemRank از هیچ تکنیکی استفاده نمی‌کند و iExpand از روش LDA برای گروه‌بندی محصولات استفاده می‌کند و PRM2 از روش SVD برای تولید یک نمایش کاهش بعد یافته استفاده می‌کند و BiFu از روش K-means، برای خوشه‌بندی کاربران و محصولات بهره می‌گیرد و ICRRS از روش کاهشی که برای تجزیه ارزیابی استفاده می‌شود، بهره می‌گیرد. برای یک مقایسه امن، ما یک برنامه برای iExpand، PRM2، BiFu، ICRRS و SCC نوشتیم. تمام برنامه‌های نوشته شده از C++ در ویژوال استودیو 2013 بهره بردند. ما از کامپیوتری با Intel®Core(TW) i7-4790k CPU، با 4 GHz و 32GB RAM استفاده کردیم که دارای ویندوز 7 برای اجرای برنامه‌ها بود. برای سهولت، ما از تعداد خوشه یکسانی برای گروه‌بندی کاربران و محصولات iExpand و BiFu استفاده کردیم. چون LDA و روش K-means نیاز به خوشه‌هایی برای تصمیم‌گیری پیشرفته دارند. در هر مورد ما از SCC برای یافتن تعداد خوشه‌ها استفاده کردیم. جدول 6 نشانگر تعداد خوشه‌ها برای خوشه‌بندی کاربران و محصولات تقسیم‌بندی شده درون SCC، iExpand و BiFu است. برای مثال، محصولات MovieLens دارای 25 خوشه برای iExpand، BiFu و SCC هستند.

جدول 7 و جدول 8 و جدول 9 نشانگر مقایسه MAE، RMSE و DOA هستند. بین ItemRankهای ایجادشده، Item Rank، iExpand و BoFu و ICRRS و SCC وجود دارند. توجه داشته باشید که این جداول دارای مقداری هستند که به صورت عملی جمع آوری شده است.

جدول 6. تعدادی از خوشه‌ها برای درگیر کردن کاربران و محصولات

	Products (iExpand, BiFu, SCC)	Users (BiFu, SCC)
MovieLens	25	5
Yahoo Movie	24	7
Amazon Video	26	14
BookCrossing	56	24
Epinions	73	33

جدول 7. مقایسه بین روش‌های متفاوت MAE برای پنج مجموعه داده

	ItemRank	iExpand	PRM2	BiFu	ICRRS	SCC
MovieLens	0.757	0.836	0.757	0.782	1.188	0.757
Yahoo Movie	0.753	0.826	0.754	0.773	1.263	0.754
Amazon Video	1.061	1.080	1.061	1.071	1.900	1.061
BookCrossing	1.210	1.192	1.205	1.250	1.680	1.206
Epinions	0.804	0.952	0.805	0.866	1.244	0.805

جدول 8. مقایسه بین روش‌های متفاوت DOA برای پنج مجموعه داده

	ItemRank	iExpand	PRM2	BiFu	ICRRS	SCC
MovieLens	0.853	0.730	0.615	0.583	0.704	0.976
Yahoo Movie	0.939	0.697	0.567	0.542	0.512	0.979
Amazon Video	0.652	0.721	0.506	0.516	0.495	0.841
BookCrossing	0.715	0.655	0.509	0.515	0.516	0.989
Epinions	0.746	0.576	0.494	0.494	0.485	0.907

جدول 9. مقایسه بین روش‌های متفاوت کارایی (ثانیه) برای پنج مجموعه داده

	ItemRank	iExpand	PRM2	BiFu	ICRRS	SCC
MovieLens	43.83	95.76	13.65	30.67	1.26	0.26
Yahoo Movie	17359.42	4924.84	33.59	5399.62	156.86	7.47
Amazon Video	43678.72	1331.25	16.34	13777.72	56.87	8.99
BookCrossing	9131.29	800.01	16.30	2946.20	29.69	9.48
Epinions	837.08	513.24	20.72	487.90	5.82	3.20

برای MAE و RMSE، مقادیر کوچک‌تر دارای کارایی بهتری هستند. در تضاد با آن، برای DOA، مقادیر بزرگ‌تر دارای اولویت بهتری هستند. همان‌طور که می‌توان دید، ItemRank و SCC دارای اجرای بهتری هستند و MAE و RMSE دارای مجموعه داده بهتری هستند. برای مثال، هر دوی ItemRank و SCC، دارای مقدار کمتری از 1.061 در MAE برای مجموعه داده آمازون هستند و هر دو دارای مقدار کمینه 1.044 در RMSE برای مجموعه داده Epinions هستند. PRM2، دارای اجرای بهتری از MAE و RMSE است. بنابراین برای DOA و SCC، اجرای بهتری در هر پنج شرایط مجموعه داده رخ می‌دهد. جدول 10 نشانگر مقایسه زمان اجرا، بین روش‌های متفاوت هستند. ما می‌توانیم ببینیم که SCC سریع‌تر از سایر روش‌ها اجرا می‌گردد. توجه داشته باشید که ItemRank، دارای کاهش بعد بهتری در مقایسه با سایرین است که در رابطه (4) نشان داده شده است. برای مثال، سایز 31102×31102 برای مجموعه داده ویدئویی آمازون است. مصرف ماتریس بهینه برای آن توصیه می‌گردد. در نتیجه، ItemRank دارای زمان بیشتری در حدود 43678.72 ثانیه است که در جدول نشان داده شده است. به جای آن، SCC حالت خوشه‌بندی خودساخته را برای کاهش ابعاد استفاده کرده است و دارای 31102 گروه و 26 نوع محصول است. بنابراین، سایز ماتریس همبستگی استفاده شده در رابطه (26) نشان داده شده است. در نتیجه، SCC می‌تواند خیلی سریع‌تر اجرا گردد که دو حدود 8.99 ثانیه خواهد بود. BiFu و iExpand دارای کاهش بعد توسط K-means و LDA هستند که دارای مصرف زمان است. بنابراین، iExpand و BiFu کندتر از SCC اجرا می‌شوند. برای مثال، iExpand دارای 1331.25 ثانیه است. ماتریس همبستگی W، در رابطه (4)، دارای سایز $M \times M$ در رابطه (26) است که دارای سایز q در q است. بنابراین SCC باید در مدت زمان $(M/q)^2$ اجرا گردد که سریع‌تر از ItemRank است.

جدول 10. شکست زمان اجرا برای ItemRank

	size	CGC	RW	total (sec)
MovieLens	1682×1682	1.25	42.58	43.83
Yahoo Movie	11916×11916	147.29	17212.13	17359.42
Amazon Video	31102×31102	699.89	42978.83	43678.72
BookCrossing	9846×9846	47.38	9083.91	9131.29
Epinions	4571×4571	5.14	831.95	837.08

SCC باید در مدت زمانی برابر با 4500 برابر سریع تر از ItemRank برای MovieLens اجرا گردد. به هر حال، از جدول 10 دیده می‌شود که SCC، فقط 244 برابر سریع تر از ItemRank اجرا می‌گردد. یک دلیل این است که سایر گام‌ها دارای زمان بیشتری هستند. جدول 11 و جدول 12، نشانگر شکست در زمان اجرای ItemRank و SCC است. توجه داشته باشید که سائز ستون‌ها در جدول نشانگر سائز ماتریس مرتبط با آن است. ItemRank دارای دو گام است: ایجاد گراف همبستگی (CGC) و RandomWalk (RW) و SCC دارای پنج گام است: برچسب‌گذاری کاربر (UL) کاهش ابعاد (DR)، ایجاد گراف همبستگی (CGC)، Random Walk (RW) و re-transformation (RT). بررسی MovieLens نشان می‌دهد. در حالت کلی 43.83 ثانیه است و ItemRank دارای 1.25 در گام CGC است.

جدول 11. شکست زمان اجرای SCC

	size	UL	DR	CGC	RW	RT	total (sec)
MovieLens	25×25	0.03	0.07	0.00	0.05	0.11	0.26
Yahoo Movie	24×24	1.94	3.59	0.01	0.22	1.70	7.47
Amazon Video	26×26	2.45	4.48	0.00	0.11	1.93	8.99
BookCrossing	56×56	1.88	4.21	0.03	0.80	2.56	9.48
Epinions	73×73	0.50	1.24	0.03	0.65	0.79	3.20

در طی کل 0.26 ثانیه، SCC در حدود 0.03 ثانیه است و در گام UL، 0.07 ثانیه می‌کشد. اگر ما به ستون RW نگاه کنیم، در این جدول این ستون‌ها در حدود 850 ثانیه زمان می‌خواهد.

انتخاب یک مقدار مناسب برای مقدار "رو" در SCC وابسته به فرآیند آزمون و خطا است. همان طور که در بخش 4 ذکر شده است، ممکن است در نتیجه الگوریتم خوشه‌بندی تأثیر زیادی در فرآیند توصیه‌گری داشته باشد. مقدار "رو" بزرگ‌تر نشان‌دهنده، تعداد خوشه کوچک‌تر و مقدار "رو" کوچک‌تر نشان‌دهنده تعداد خوشه بزرگ است.

7. نتیجه‌گیری

در یک سیستم فیلترینگ با همبستگی مشترک، مانند ItemRank، محصولات به عنوان یک روش مناسب شناخته شده است، اما معایبی بر آن وارد است. با این حال، معمولاً محصولات تجاری بسیار زیادی وجود دارند که قبلاً مورد بررسی قرار گرفته‌اند. ما یک روش خوشه‌بندی خودساخته برای کاهش ابعاد مرتبط با تعداد محصولات ارائه کرده‌ایم. در نتیجه، زمان پردازش اندازه‌گیری شده توسط روش ما برای تصمیم‌گیری در مورد یک روش توصیه خیلی کمتر شده است. نتایج آزمایش‌ها نشان می‌دهد که کارایی سیستم پیشنهادی بدون داشتن تأثیر بر روی کیفیت توصیه بهبود یافته است.

References

- [1] C. Porcel, J. Moreno, E. Herrera-Viedma, A multi-disciplinar recommender system to advice research resources in university digital libraries, *Expert Systems with Applications* 36 (2009) 12520–12528.
- [2] F. Ricci, L. Rokach, B. Shapira, Introduction to recommender systems handbook, *Recommender Systems Handbook*, Springer, 2011.
- [3] A. Gatzoura, M. Sánchez-Marrè, A case-based recommendation approach for market basket data, *IEEE Intelligent Systems* 30 (1) (2014) 20–27.
- [4] M. Balabanović, Y. Shoham, Fab: Content-based, collaborative recommendation, *Communications of ACM* 40 (3) (1997) 66–72.
- [5] P. Melville, R. J. Mooney, R. Nagarajan, Content-boosted collaborative filtering for improved recommendations, in: 18th National Conference on Artificial Intelligence, Edmonton, Canada, 2002, pp. 187–192.
- [6] S.-L. Huang, Designing utility-based recommender systems for e-commerce: Evaluation of preference-elicitation methods, *Electronic Commerce Research and Applications* 10 (2011) 398–407.
- [7] L. Liu, N. Mehandjiev, D.-L. Xu, Context similarity metric for multidimensional service recommendation, *International Journal of Electronic Commerce* 18 (1) (2013) 73–104.
- [8] M. K. Condliiff, D. D. Lewis, D. Madigan, Bayesian mixed-effects models for recommender systems, in: *ACM SIGIR Workshop on Recommender Systems – Algorithms and Evaluation*, 1999.
- [9] I. Rish, An empirical study of the naive bayes classifier, in: *International Joint Conferences on Artificial Intelligence(IJCAI) Workshop on Empirical Methods in AI*, 2002, pp. 41–46.
- [10] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: An open architecture for collaborative filtering of netnews, in: *ACM 1994 Conference on Computer Supported Cooperative Work*, 1994.
- [11] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, J. Riedl, Grouplens: Applying collaborative filtering to usenet news, *Communications of the ACM* 40 (3) (1997) 77–87.
- [12] B. M. Sarwar, G. Karypis, J. A. Konstan, J. T. Riedl, Application of dimensionality reduction in recommender systems – a case study, in: *ACM WEBKDD workshop*, 2000.
- [13] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, *Journal of Machine Learning Research* 3 (2003) 993–1022.

- [14] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl, Evaluating collaborative filtering recommender systems, *ACM Transactions on Information Systems* 22 (1) (2004) 5–53.
- [15] S. Funk, Netflix, A modified SVD algorithm tends to make a mess of sparsely observed movies or users, <http://sifter.org/~simon/journal/20061211.html>.
- [16] F. Fouss, A. Pirotte, J.-M. Renders, M. Saeens, Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation, *IEEE Transactions on Knowledge and Data Engineering* 19 (3) (2007) 355–369.
- [17] A. Pucci, M. Gori, M. Maggini, A random-walk based scoring algorithm applied to recommender engines, *Lecture Notes in Computer Science – Advances in Web Mining and Web Usage Analysis* 4811 (2007) 127–146.
- [18] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, in: *Neural Information Processing Systems 20 (NIPS'07)*, 2008, pp. 1257–1264.
- [19] N. N. Liu, M. Zhao, Q. Yang, Probabilistic latent preference analysis for collaborative filtering, in: *ACM Conference on Information and Knowledge Management*, 2009, pp. 759–766.
- [20] K.-J. Kim, H. Ahn, Collaborative filtering with a user-item matrix reduction technique, *International Journal of Electronic Commerce* 16 (1) (2011) 107–128.
- [21] M. Jiang, P. Cui, R. Liu, Q. Yang, Fei, W. Zhu, S. Yang, Social contextual recommendation, in: *21st ACM international conference on Information and knowledge management*, 2012, pp. 45–54.
- [22] C. Porcel, A. Tejeda-Lorente, M. Martinez, E. Herrera-Viedma, A hybrid recommender system for the selective dissemination of research resources in a technology transfer office, *Information Sciences* 184 (2012) 1–19.
- [23] A. N. Nikolakopoulos, M. Kouneli, J. Garofalakis, A novel hierarchical approach to ranking-based collaborative filtering, *Communications in Computer and Information Science – Engineering Applications of Neural Networks* 384 (2013) 50–59.
- [24] G. Guo, J. Zhang, D. Thalmann, N. Yorke-Smith, Leveraging prior ratings for recommender systems in e-commerce, *Electronic Commerce Research and Applications* 13 (2014) 440–455.
- [25] H. Cui, M. Zhu, Collaboration filtering recommendation optimization with user implicit feedback, *Journal of Computational Information Systems* 10 (14) (2014) 5855–5862.
- [26] B. M. Sarwar, G. Karypis, J. Konstan, J. Riedl, Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering, in: *5th International Conference on Computer and Information Technology*, 2002.
- [27] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, Z. Chen, Scalable collaborative filtering using cluster-based smoothing, in: *ACM SIGIR Conference*, 2005.

- [27] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, Z. Chen, Scalable collaborative filtering using cluster-based smoothing, in: ACM SIGIR Conference, 2005.
- [28] K. Honda, A. Notsu, H. Ichihashi, Collaborative filtering by user-item clustering based on structural balancing approach, *International Journal of Computer Science and Network Security* 8 (12) (2008) 190–195.
- [29] Q. Ba, X. Li, Z. Bai, Clustering collaborative filtering recommendation system based on SVD algorithm, *IEEE International Conference on Software Engineering and Service Science* (2013) 963–967.
- [30] Q. Liu, E. Chen, H. Xiong, C. H. Q. Ding, J. Chen, Enhancing collaborative filtering by user interest expansion via personalized ranking, *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* 42 (1) (2012) 218–233.
- [31] Y. Cai, H. fung Leung, Q. Li, H. Min, J. Tang, J. Li, Typicality-based collaborative filtering recommendation, *IEEE Transactions on Knowledge and Data Engineering* 26 (3) (2014) 766–779.
- [32] M. Sarwat, J. J. Levandoski, A. Eldawy, M. F. Mokbel, LARS*: An efficient and scalable location-aware recommender system, *IEEE Transactions On Knowledge And Data Engineering* 26 (6) (2014) 1384–1399.
- [33] X. Qian, H. Feng, G. Zhao, T. Mei, Personalized recommendation combining user interest and social circle, *IEEE Transactions on Knowledge and Data Engineering* 26 (7) (2014) 1763–1777.
- [34] D. Zhang, C.-H. Hsu, M. Chen, Q. Chen, N. Xiong, J. Lloret, Cold-start recommendation using bi-clustering and fusion for large-scale social recommender systems, *IEEE Transactions on Emerging Topics in Computing* 2 (2) (2014) 239–250.
- [35] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: 5-th Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, 1967, pp. 281–297.
- [36] M. Allahbakhsh, A. Ignjatovic, An iterative method for calculating robust rating scores, *IEEE Transactions on Parallel and Distributed Systems* 26 (2) (2015) 340–350.
- [37] M. Gori, A. Pucci, ItemRank: A random-walk based scoring algorithm for recommender engines, in: *IJCAI*, 2007, pp. 2766–2771.
- [38] Z. Gyongyi, H. Garcia-Molina, J. Pedersen, Combating web spam with trustrank, in: 30th International Conference on Very Large Data Bases (VLDB), Morgan Kaufmann, 2004, pp. 576–587.
- [39] S.-J. Lee, C.-S. Ouyang, A neuro-fuzzy system modeling with self-constructing rule generation and hybrid SVD-based learning, *IEEE Transactions on Fuzzy Systems* 11 (3) (2003) 341–353.
- [40] J.-Y. Jiang, R.-J. Liou, S.-J. Lee, A fuzzy self-constructing feature clustering algorithm for text classification, *IEEE Transactions on Knowledge and Data Engineering* 23 (3) (2011) 335–349.

- [41] N. Slonim, N. Tishby, The power of word clusters for text classification, in: 23rd European Colloquium on Information Retrieval Research (ECIR-01), 2001.
 - [42] R. Bekkerman, R. El-Yaniv, N. Tishby, Y. Winter, Distributional word clusters versus words for text categorization, *The Journal of Machine Learning Research* 3 (3) (2003) 1183–1208.
 - [43] H. Li, T. Jiang, K. Zhang, Efficient and robust feature extraction by maximum margin criterion, *IEEE Transactions on Neural Networks* 17 (1) (2006) 157–165.
 - [44] J. Yan, B. Zhang, N. Liu, S. Yan, Q. Cheng, W. Fan, Q. Yang, W. Xi, Z. Chen, Effective and efficient dimensionality reduction for large-scale and streaming data preprocessing, *IEEE Transactions on Knowledge and Data Engineering* 18 (3) (2006) 320–333.
 - [45] V. S. Tseng, C.-P. Kao, A novel similarity-based fuzzy clustering algorithm by integrating PCM and mountain method, *IEEE Transactions on Fuzzy Systems* 15 (6) (2007) 1188–1196.
 - [46] Z. Zhang, H. Cheng, S. Zhang, W. Chen, Q. Fang, Clustering aggregation based on genetic algorithm for documents clustering, in: *IEEE Congress on Evolutionary Computation*, 2008, pp. 3156–3161.
 - [47] Z. Huang, W. Chung, H. Chen, A graph model for e-commerce recommender systems, *Journal of the American Society for Information Science and Technology* 55 (3) (2004) 259–274.
 - [48] D. Harel, Y. Koren, On clustering using random walks, *Lecture Notes in Computer Science – Foundations of Software Technology and Theoretical Computer Science* 2245 (2001) 18–41.
 - [49] H. Yildirim, M. S. Krishnamoorthy, A random walk method for alleviating the sparsity problem in collaborative filtering, in: *2008 ACM conference on Recommender systems*, 2008, pp. 131–138.
-
- [50] D. Wu, G. Zhang, J. Lu, A fuzzy preference tree-based recommender system for personalized business-to-business e-services, *IEEE Transactions on Fuzzy Systems* 23 (1) (2015) 29–43.
 - [51] Movielens dataset, <http://www.grouplens.org/node/73#attachments>.
 - [52] Yahoo movie dataset, <https://www.yahoo.com/movies/>.
 - [53] Amazon video dataset, <http://www.amazon.com/>.
 - [54] Bookcrossing dataset, <http://www.bookcrossing.com/>.
 - [55] Epinions dataset, <http://www.epinions.com/>.