ارائه راهکاری نوین برای تولید سامانه‌های توصیه‌گر با استفاده از روش تجزیه نامنفی ماتریس

نوشین شاهرخی*، سمیه عربی نرئی
دانشگاه خوارزمی، دانشکدهٔ علوم ریاضی و کامپیوتر، گروه علوم کامپیوتر

دریافت ۹۷/۰۷/۱۷ پذیرش ۹۸/۰۸/۰۶

چکیده

تجزیه نامنفی ماتریس یک رویکرد جدید برای کاهش ابعاد داده‌ها است. در این روش با اعمال محدودیت نامنفی بودن داده‌های ماتریس، ماتریس به اجزایی تجزیه می‌شود که این اجزا تفسیر پذیرتر هستند و داده‌ها را به بخش‌هایی تقسیم می‌کنند که داده‌های موجود در این بخش‌ها ارتباط خاصی با هم دارند. در این مقاله از این خاصیت تجزیه نامنفی ماتریس، برای تجزیه ماتریس امتیازات کاربران به کالاها در سامانه‌های توصیه‌گر استفاده می‌کنیم. بدین‌ترتیب که ماتریس امتیازات را تجزیه می‌کنیم به‌گونه‌ای که کاربران با علایق مشابه تشخیص داده می‌شوند. در این مقاله به‌منظور کمینه‌سازی اختلاف بین ماتریس اصلی و فاکتورهای تجزیه، از روش منظم‌سازی استفاده می‌کنیم به‌طوری که ضرایبی از نرم فاکتورهای تجزیه را در معادله تجزیه اعمال می‌کنیم که در یک فرایند به‌روز رسانی ضربی، داده‌های فاکتورهای تجزیه را کنترل می‌کنند. نتایج عددی روی مجموعه داده‌های موی‌لنز[1] نشان‌گر دقت بیش‌تر روش پیشنهادی ما در پیش‌بینی امتیازات کاربران به کالاها است.

واژه‌های کلیدی: تجزیه نامنفی ماتریس، سامانه‌های توصیه‌گر، کم‌ترین مربعات تکراری، به‌روز رسانی ضربی، پردازش داده.

مقدمه

امروزه با توجه به حجم عظیم داده‌ها به‌دنبال روش‌هایی هستیم که بتوانند خلاصه مفیدی را از داده‌ها استخراج کنند. روش‌هایی مانند پردازش سیگنال[2]، تحلیل داده[3]، داده کاوی[4]، تشخیص الگو[5] و یادگیری ماشین[6] این هدف را دنبال می‌کنند. این روش‌ها به‌عنوان روش‌های کاهش ابعاد شناخته می‌شوند که یک چالش بزرگ در تحلیل داده‌های چند متغیره هستند. در واقع در این روش‌ها دو هدف عمده دنبال می‌شود. اول این‌که ابعاد داده‌ها کاهش داده شوند و دوم اینکه مؤلفه‌های اصلی و اطلاعات پنهان در داده‌ها تشخیص داده شوند.

در بسیاری از کاربردها از جمله تولید سامانه‌های توصیه‌گر، داده‌ها به‌صورت ماتریس قابل نمایش هستند که در این صورت کاهش ابعاد می‌تواند به‌وسیلهٔ روش‌های جبری تجزیه ماتریس مانند تحلیل مؤلفه اصلی PCA[7]، تجزیه و تحلیل خطی LDA[8]، تحلیل مؤلفه مستقل ICA[9] و چندسازی برداری VQ[10] انجام گیرد. اما اشکال عمده در این

*نویسنده مسئول Nushin_shahrokhi@yahoo.com

1. MovieLens
2. Signal Processing
3. Data Analysis
4. Data Mining
5. Pattern Recognition
6. Machine Learning
7. Principal Component Analysis
8. Linear Discriminant Analysis
9. Independent Component Analysis
10. Vector Quantization

روش‌ها ظاهر شدن داده‌های منفی در فاکتورهای تجزیه است. در حالی که بسیاری از داده‌های واقعی مانند تصاویر، متن و طیف‌های صوتی را می‌توان به ماتریس‌های نامنفی تبدیل کرد که از تجزیه آنها به عوامل نامنفی نتایج جالبی به‌دست می‌آید. برخلاف روش‌های نام‌برده در بالا، تجزیه نامنفی ماتریس از محدودیت نامنفی بودن عناصر تجزیه استفاده می‌کند تا عامل‌های تجزیه قابل تفسیر باشند. امروزه تجزیه نامنفی ماتریس به‌عنوان یک ابزار برای تحلیل داده و پردازش سیگنال شناخته می‌شود [۱].

در این مقاله علاوه بر منظم‌سازی درایه‌های ماتریس‌های تجزیه، ترکیبی از الگوریتم‌های کم‌ترین مربعات تکراری و به‌روز رسانی ضربی را استفاده می‌کنیم. ممکن است در طول عملیات به‌روز رسانی ضربی، اندازه داریه‌های ماتریس‌های تجزیه افزایش یابد که این افزایش موجب افزایش خطای محاسبات در هر مرحله می‌شود. برای جلوگیری از بزرگ شدن بیش از حد درایه‌های ماتریس‌های تجزیه، ضرایبی را به مقدار خطای حاصل از تخمین درایه نظیر آن سطر و ستون اضافه می‌کنیم تا در هر مرحله علاوه بر جلوگیری از بزرگ شدن خطا، از بزرگ شدن درایه‌ها ماتریس‌های تجزیه نیز جلوگیری کنیم. نتایج عددی ما نشان‌دهندهٔ افزایش دقت روش پیشنهادی در تجزیه ماتریس امتیازات کاربران و افزایش صحت پیش‌بینی امتیازات کاربران به کالاها است.

سامانه‌های توصیه گر

سامانه‌های توصیه‌گر[1] الگوریتم‌های نسبتاً ساده‌ای هستند که با کاوش در اطلاعات مرتبط با کاربران از بانک اطلاعاتی مربوط و بررسی انتخاب‌های کاربران در گذشته، الگوهایی را در داده‌ها پیدا می‌کنند که با توجه به آن الگوهای رفتاری، برای هر کاربر، توصیه مناسب را ارائه می‌دهند. سامانه‌های محتوا محور[2] و سامانه‌های مشارکت محور[3] محور[3] دو دسته مهم از سامانه‌های توصیه‌گر هستند. سامانه‌های محتوا محور براساس خصوصیات کالاها و مشابهت بین آنها و نیز علایق کاربر (در صورت وجود)، توصیه‌هایی به کاربر ارائه می‌کنند. در سامانه‌های مشارکت محور، به کاربر اقلامی توصیه می‌شود که دیگران در گذشته با تمایلات و ترجیحات مشابه او این اقلام را پسندیده‌اند. یعنی بر اساس رابطهٔ بین کاربران و کالاها، اقلام جدید به کاربر توصیه می‌شود. در این روش‌ها، خود کالا اهمیتی ندارد و بر اساس انتخاب کاربران دیگر و انتخاب‌های گذشته خود کاربر، به او پیشنهادهای جدیدی ارائه می‌شود. در این نوع از سامانه‌های توصیه‌گر، از اطلاعات مربوط به ویژگی کاربران یا کالاها استفاده نمی‌کنند بلکه اطلاعات کاربران در یک ماتریس رتبه‌بندی قرار می‌گیرد به‌طوری‌که در آن، رتبه‌هایی که کاربران به کالاها داده‌اند، به‌عنوان درایه‌های ماتریس ذخیره می‌شود و اگر کاربری به یک کالا امتیازی نداده باشد درایه متناظرش در ماتریس را صفر درنظرمی‌گیریم [۲].

در این مقاله بر سامانه‌های توصیه‌گر مشارکت محور تمرکز می‌کنیم.

تجزیهٔ نامنفی ماتریس

ماتریس نامنفی $\mathrm{A} \in \mathrm{R}^{\mathrm{m}\times\mathrm{n}}$ و عدد طبیعی $1 \le k \le \min(m,n)$ را درنظر بگیرید. هدف پیدا کردن ماتریس‌های نامنفی $\mathrm{W} \in \mathrm{R}^{\mathrm{m}\times\mathrm{k}}$ و $H \in R^{k\times n}$ است به‌طوری‌که تابع زیر کمینه شود:

$$\mathrm{f}(\mathrm{W},\mathrm{H}) = \frac{1}{2}\|\mathrm{A}-\mathrm{WH}\|_{\mathrm{F}}^{2}. \qquad (۱)$$

W و H را ماتریس‌های تجزیه گوییم.

1. Recommender Systems
2. Content Filtering
3. Collaborative Filtering

در تجزیه نامنفی ماتریس به‌طور کلی باید دو ویژگی اساسی را برقرار کرد:

۱. بعد داده‌های اصلی باید کاهش یابد؛

۲. اجزای اصلی، مفاهیم پنهان و ویژگی‌های برجسته، به‌طور مناسب شناسایی شوند. (ویژگی‌ها، می‌توانند بخشی از چهره در داده‌های تصویری، موضوعات در داده‌های متنی، ویژگی کالاها در سامانه‌های توصیه‌گر و... باشند.)

یکی از کاربردهای NMF استفاده از آن در سامانه‌های توصیه‌گر است. در این سامانه‌ها تعداد زیادی کالا و کاربر داریم که کاربران می‌توانند میزان رضایت خود از کالاهایی که استفاده کرده‌اند را به‌وسیلهٔ رتبه‌دهی به کالاها با امتیاز مثلاً یک تا پنج که به‌ترتیب بیان‌گر حداقل و حداکثر رضایت است، اعلام کنند. به این ترتیب ماتریس داده که درایه‌هایش امتیازهای داده‌شده به‌وسیلهٔ کاربران است، در اختیار است. در این سامانه‌ها هدف، ارائه پیشنهاد و توصیه کالاها به کاربران، با استفاده از اطلاعات دیگر کاربران است. NMF با دسته‌بندی کاربران (به‌طوری که کاربرانی که نظرات مشابه دارند دریک گروه قرار بگیرند)، برای توصیه کالا به کاربر مشخص، از نظرات کاربرانی که با این کاربر، در یک گروه قرار گرفته‌اند، استفاده می‌کند و کالایی که مورد علاقه آنها است را به این کاربر پیشنهاد می‌کند.

الگوریتم‌های اصلی تجزیه نامنفی ماتریس

روش‌های تکراری مختلفی برای تجزیه نامنفی ماتریس پیشنهاد شده‌اند. در این باره می‌توان به پژوهش‌های لین[1] (۲۰۰۵) [۳]، [۴]، بری[2] و همکاران وی (۲۰۰۷) [۵]، و سپس کیم و پارک[3] (۲۰۰۷) [۶]، [۷]، اشاره کرد [۱].

روش‌های مختلف NMF به‌دنبال یافتن راهی هستند که در هر تکرار، تقریب بهتری برای H و W حاصل شود یا سرعت همگرایی روش، بیش‌تر باشد. برای یافتن ماتریس‌های W و H باید معادلهٔ (۱) را کمینه کنیم. این مسئله، به‌دلیل مجهول بودن W و H نامحدب است، از این‌رو، دارای کمینه سراسری نیست. بنابراین، هدف به‌دست آوردن کمینه موضعی است. در[۹] اثبات شده است که اگر (W,H) کمینهٔ موضعی مسئله مذکور باشد، آن‌گاه شرایط KKT بدین‌صورت برقرار است:

$$\begin{aligned} \mathrm{w} \le 0, \quad & H \le 0, \\ \frac{\partial f}{\partial H} \le 0, \quad & \frac{\partial f}{\partial W} \le 0, \\ \frac{\partial f}{\partial H} .* H = 0, \quad & \frac{\partial f}{\partial W} .* W = 0, \end{aligned}$$

که نماد $*$. به معنای ضرب آدامار[4] یا مؤلفه به مؤلفه است. از مقدمات جبرخطی نتیجه می‌شود:

$$\frac{\partial f}{\partial H} = W^T(WH - A), \qquad \frac{\partial f}{\partial W} = (WH - A)H^T.$$

بنابراین الگوریتم‌های NMF درجهت فراهم کردن شرایط بالا به‌وجود می‌آیند.

به‌طورکلی می‌توان الگوریتم‌های اصلی NMF را به سه دسته تقسیم کرد:

الگوریتم‌های به‌روزرسانی ضربی[5]، الگوریتم‌های نزول گرادیان[6] و الگوریتم‌های کم‌ترین مربعات[7] [۵].

1. Chin-Jen Lin
2. Michael Berry
3. Hyunsoo Kim and Haesun Park
4. Hadamard
5. Multiplicative Update Algorithms
6. Gradient Descent Algorithms
7. Least Squares Algorithms

الگوریتم‌های به‌روز رسانی ضربی

پرکاربردترین روش برای مسئله NMF را می‌توان الگوریتم‌های به‌روزرسانی ضربی دانست که به‌وسیلهٔ لی و سونگ (۲۰۰۱) پیشنهاد شده‌است [۱۰].

$$W = rand(m,k)$$
$$H = rand(k,n)$$
$$for\ i = 1{:}\ t$$
$$H = H.*\ (W^T A)./(W^T WH + 10^{-9})$$
$$W = W.*\ (AW^T)./(WHH^T + 10^{-9})$$
$$end$$

به‌طوری که نماد / . به‌معنای تقسیم مؤلفه به مؤلفه است. در هر مرحله مخرج را با 10^{-9} جمع می‌کنیم تا از تقسیم کردن بر صفر، جلوگیری کنیم.

همگرایی روش به‌روزرسانی ضربی به یک کمینه موضعی

اگر ماتریس‌های اولیه H و W مثبت باشند، با توجه به قوانین به‌روزرسانی و با فرض این‌که ماتریس A دارای سطر یا ستون صفر نیست (در صورت وجود سطر یا ستون صفر، بدون کاستن از کلیت مسئله می‌توان آن را حذف کرد.)، واضح است که در هر تکرار نیز ماتریس‌های H و W مثبت باقی می‌مانند. اگر دنباله (W, H) به (W^*, H^*) همگرا شود و $W > 0$ و $H > 0$ در این صورت داریم:

$$\frac{\partial f(W^*, H^*)}{\partial H} = 0, \qquad \frac{\partial f(W^*, H^*)}{\partial W} = 0.$$

عبارت اول را ثابت می‌کنیم، عبارت دوم نیز به طریق مشابه اثبات می‌شود:

فرمول به‌روزرسانی H را می‌توان بدین‌صورت بازنویسی کرد:

$$H = H + [H./(W^T WH)].*\ [W^T(A - WH)]. \qquad (۲)$$

زیرا:

$$\begin{aligned} H &= H + [H./(W^T WH)].*\ [W^T(A - WH)] \\ &= H + [H./(W^T WH).*\ W^T A] - [H./(W^T WH)\ .*\ (W^T WH)] \\ &= H + [H./(W^T WH).*\ W^T A] - H \\ &= H./(W^T WH).*\ W^T A. \end{aligned}$$

عنصر H_{ij} را در نظر بگیرید. فرض کنید پس از به‌روزرسانی، ماتریس H حاصل، بسیار نزدیک به ماتریس H قبل از به‌روزرسانی باشد و هم‌چنین $H_{ij} > 0$ ، از تساوی (۲) داریم:

$$\frac{H_{ij}}{W^T WH_{ij}}\left(W^T A_{ij} - W^T WH_{ij}\right) = 0.$$

از آن‌جاکه $H_{ij} > 0$، نتیجه می‌شود:

$$W^T A_{ij} - W^T WH_{ij} = 0.$$

یعنی $(\frac{\partial f}{\partial H})_{ij} = 0$. بنابراین زمانی‌که دنبالهٔ (W, H) به (W^*, H^*) همگرا شود، چون در ابتدا فرض کردیم که ماتریس‌های اولیه مثبت باشند، پس برای هر i و j داریم $H_{ij} > 0$ و در نتیجه برای هر i و j، $(\frac{\partial f}{\partial H})_{ij} = 0$. یعنی $(\frac{\partial f}{\partial H}) = 0$.

روشن است که وقتی مشتقات جزئی بالا صفر شوند، چون ماتریس‌های W و H نامنفی هستند، شرایط KKT برقرار است و به‌ازای ماتریس‌های W^* و H^* حاصل، کمینهٔ مورد نظر به‌دست می‌آید. بنابراین درصورتی‌که تضمین کنیم ماتریس‌های اولیه مثبت هستند، قوانین به‌روزرسانی ضربی به یک کمینه موضعی همگرا می‌شوند [۹].

الگوریتم‌های نزول گرادیان

الگوریتم‌های نزول گرادیان بدینصورت به‌روزرسانی می‌شوند:

$$W = rand(m,k)$$
$$H = rand(k,n)$$
$$for\ i = 1{:}\ t$$
$$H = H - \alpha \frac{\partial f}{\partial H}$$
$$W = W - \beta \frac{\partial f}{\partial W}$$
$$end$$

که پارامترهای طول گام α و β به الگوریتم وابسته هستند و چنان انتخاب می‌شوند که در هر تکرار گرادیان به صفر نزدیک‌تر شود و با صفر کردن گرادیان شرایط KKT را فراهم نمایند. اما نکتهٔ قابل توجه، چگونگی انتخاب طول گام‌ها است. در برخی الگوریتم‌ها این پارامترها را در ابتدا برابر با مقدار یک قرار می‌دهند و سپس در هر تکرار مقدار آنها را نصف می‌کنند، اگرچه این روش ساده است اما روشی بهینه نیست و تضمین نمی‌کند که درایه‌های ماتریس‌های به‌روزرسانی شده، نامنفی باشند. یک روش رایج که در بسیاری از الگوریتم‌های نزول گرادیان برای رفع این مشکل به‌کار می‌رود، یک برنامه ساده است که در روند به‌روزرسانی، بعد از هر تکرار، با بررسی درایه‌های ماتریس حاصل، درایه‌های منفی با نزدیک‌ترین مقدار نامنفی، یعنی صفر، جای‌گزین می‌شوند [۵]. البته در این صورت تحلیل همگرایی روش سخت‌تر است. درحالت کلی الگوریتم‌های مبتنی بر روش‌های گرادیان، همگرایی مطلوبی ندارند [۸].

الگوریتم‌های کم‌ترین مربعات تکراری

آخرین دسته از الگوریتم‌های NMF الگوریتم‌های کم‌ترین مربعات تکراری هستند. در این الگوریتم‌ها ابتدا یکی از ماتریس‌ها مثلاً W را با اعداد تصادفی نامنفی مقداردهی می‌کنیم و سپس تنها مجهول دستگاه $(WH = A)$ یعنی ماتریس H را از حل دستگاه به‌روش کم‌ترین مربعات به‌دست می‌آوریم و درایه‌های منفی آن را با نزدیک‌ترین مقدار نامنفی، یعنی صفر جای‌گزین کرده و پس از آن، ماتریس H را معلوم در نظر می‌گیریم و دوباره از حل دستگاه به‌روش کم‌ترین مربعات ماتریس W را تخمین می‌زنیم و درایه‌های منفی آن را با صفر جای‌گزین می‌کنیم. این روند را ادامه می‌دهیم تا حاصل‌ضرب WH به‌اندازهٔ دلخواه به ماتریس A نزدیک شود [۵].

$$W = rand(m,k)$$
$$for\ i - 1{:}\ t$$
$$solve\ for\ H\ in\ matrix\ equation$$
$$W^T W H = W^T A$$
$$set\ all\ negative\ elements\ in\ H\ to\ 0$$
$$solve\ for\ W\ in\ matrix\ equation$$
$$H H^T W^T = H A^T$$

$$Set\ all\ negative\ elements\ in\ W\ to\ 0$$

$$end$$

این الگوریتم موجب تسریع محاسبات می‌شود اما تحلیل همگرایی آن سخت است. معمولاً این الگوریتم به‌عنوان یک راه‌انداز برای الگوریتم‌های دیگر به‌کار می‌رود [۸].

عدم یکتایی و منظم‌سازی

تجزیه یک ماتریس به عوامل نامنفی لزوماً یکتا نیست، زیرا اگر WH یک تجزیه نامنفی از ماتریس A باشد، برای هر ماتریس قطری با درایه‌های بزرگ‌تر از صفر مانند D داریم $WH = WD^{-1}DH$ که ماتریس‌های WD^{-1} و DH تجزیه نامنفی دیگری برای ماتریس A است. گاهی فرمول مسئله NMF گسترش می‌یابد تا شامل محدودیت‌هایی روی W و H شود. این محدودیت‌ها معمولاً برای جبران عدم یکتایی ماتریس‌های تجزیه اعمال می‌شوند. معادلهٔ (۳) شکل گسترش یافته معادله (۱) است. در حالت کلی یک مسئله بهینه‌سازی مقید را می‌توان با واردن کردن شکل مناسبی از قیود مسئله به تابع هدف، به یک مسئله نامقید تبدیل کرد. در واقع معادلهٔ (۳) شکل نامقید مسئله NMF است:

$$F(W,H) = \frac{1}{2}\|A - WH\|_F^2 + \alpha j_1(W) + \beta j_2(H). \qquad (۳)$$

به‌طوری‌که ضرایب α و β پارامترهای کوچک منظم‌سازی هستند. j_1 و j_2 را می‌توان توابع مختلفی در نظر گرفت که در ادامه تعدادی از آنها را معرفی می‌کنیم.

می‌توان گفت یکی از رایج‌ترین انتخاب‌ها برای توابع مذکور، مربع نرم فروبنیوس است:

$$j_1(W) = \|W\|_F^2.$$

تابع مذکور، در عمل منظم‌سازی را روی ستون‌های W اجرا می‌کند. زیرا:

$$\|W\|_F^2 = \sum_i \|W^i\|_F^2.$$

این شکل از منظم‌سازی به‌عنوان منظم‌سازی تیخونف در مسئله معکوس شناخته می‌شود. در حالت کلی می‌توانیم $j_1(W) = \|W\|_F^2$ را به‌صورت $\mathrm{j_1(W) = \|LW\|_F^2}$ بازنویسی کنیم که در آن L یک عملگر منظم‌سازی است. گزینه‌های دیگر L می‌تواند عملگرهای لاپلاس باشند [۵].

یکی از قیودی که روی H می‌توان اعمال کرد، بدین‌صورت تعریف می‌شود:

$$j_2(H) = \frac{1}{n}\sum_i \left\|(I-T)H^{i^T}\right\|_2^2 = \frac{1}{n}\|(I-T)H^T\|_F^2.$$

که در آن n تعداد ستون‌های ماتریس داده‌ها یعنی ماتریس A بوده و T عملگر پیچیدگی است که بدین‌صورت درنظر گرفته‌می‌شود:

$$T = \begin{pmatrix} \delta & 0 & 0 & & 0 & 0 & 0 & 0 & \cdots & 0 \\ \gamma\delta & \delta & 0 & & 0 & 0 & 0 & 0 & \cdots & 0 \\ \gamma^2\delta & \gamma\delta & \delta & & 0 & 0 & 0 & 0 & \cdots & 0 \\ \gamma^3\delta & \gamma^2\delta & \gamma\delta & & \delta & 0 & 0 & 0 & \cdots & 0 \\ \gamma^4\delta & \gamma^3\delta & \gamma^2\delta & & \gamma\delta & \delta & 0 & 0 & \cdots & 0 \\ 0 & \gamma^4\delta & \gamma^3\delta & & \gamma^2\delta & \gamma\delta & \delta & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & & 0 & \cdots & 0 & \gamma^4\delta & \gamma^3\delta & \gamma^2\delta & \gamma\delta & \delta \end{pmatrix}.$$

و در این جا، $0 < \gamma < 1$ عاملی است که محدوده هموارسازی موضعی را تعیین می‌کند و $\delta = 1 - \gamma$.

تابع دیگری که می‌تواند به‌کار گرفته شود به‌صورت ذیل است که در آن $vec(.)$ برداری ستونی است که از یک ماتریس با پشت سرهم قرار دادن ستون‌هایش حاصل می‌شود و $\omega = \sqrt{kn} - (\sqrt{kn} - 1)\gamma$ که γ پارامتری است بین صفر و یک که میزان تنک بودن ماتریس H را تنظیم می‌کند [۵].

$$j_2(H) = (\omega \|vec(H)\|_2 - \|vec(H)\|_1)^2.$$

تنکی و پراکندگی داده‌ها

قیود مربوط به تنک بودن را به‌طور مشابه می‌توان برای W و H اعمال کرد. مفهوم تنک بودن وقتی مطرح می‌شود که تعداد داده‌ها به‌طور نسبی اندک است و ماتریس داده صفرهای زیادی دارد. به‌عبارت دیگر، فقط چند ویژگی به‌طور مؤثر برای نشان دادن بردارهای داده نقش دارند. اندازه‌ای که هویر[1] در سال ۲۰۰۴ ارائه داده است، براساس ارتباط بین فضای L_1 و L_2 است و برای $x \in R^n - \{0\}$ بدین‌صورت تعریف می‌شود:

$$Sparsness(x) = \frac{\sqrt{n} - \|x\|_1/\|x\|_2}{\sqrt{n} - 1} = \frac{\sqrt{n} - (\sum_i^n |x_i| / \sqrt{\sum_i^n |x_i^2|})}{\sqrt{n} - 1}. \quad (۴)$$

تابع حقیقی مقدار فوق فقط مقادیر بین صفر و یک را اختیار می‌کند. زیرا می‌دانیم برای هر $x \in R^n$ نابرابری ذیل برقرار است:

$$\|x\|_2 \le \|x\|_1 \le \sqrt{n}\|x\|_2,$$

چون در تابع هویر $x \neq 0$، پس $\|x\|_2 \neq 0$ و با تقسیم طرفین بر $\|x\|_2$ داریم:

$$1 \le \frac{\|x\|_1}{\|x\|_2} \le \sqrt{n},$$

در نتیجه:

$$-\sqrt{n} \le -\frac{\|x\|_1}{\|x\|_2} \le -1,$$

$$0 \le \sqrt{n} - \frac{\|x\|_1}{\|x\|_2} \le \sqrt{n} - 1,$$

$$0 \le \frac{\sqrt{n} - \frac{\|x\|_1}{\|x\|_2}}{\sqrt{n} - 1} \le 1.$$

برای اعمال قیود تنکی، باید پارامترهایی تعریف کنیم که بیان‌گر میزان تنکی ماتریس‌های W و H باشند. بنابراین، بردارهای α_W و α_H را تعریف می‌کنیم که به‌ترتیب، درایه‌هایشان میزان تنک بودن ستون‌های W و H را نشان می‌دهد. یعنی به‌عنوان مثال α_W باید به تعداد ستون‌های W درایه داشته باشد و هر درایه آن بیان‌گر میزان تنکی ستون نظیر آن درایه در W است که به‌وسیلهٔ رابطهٔ (۴) به‌دست می‌آید. یعنی اگر W^i ستون iام ماتریس W باشد، داریم:

$$\alpha_W = (Sparsness(W^1), Sparsness(W^2), \cdots, Sparsness(W^k))^T.$$

با استفاده از این دو پارامتر، دو پارامتر دیگر تعریف می‌کنیم که در عمل، برای انجام محاسبات از آن‌ها استفاده می‌شود [۱۱]:

$$\beta_W = ((1 - \alpha_W)\sqrt{k} + \alpha_W)^2, \beta_H = ((1 - \alpha_H)\sqrt{k} + \alpha_H)^2.$$

1. Hoyer

روش پیشنهادی

اکنون روشی ارائه می‌کنیم که نسبت به‌روش‌های معرفی شده در فصل قبل، دقت بیش‌تری دارد. در این روش علاوه بر منظم‌سازی درایه‌های ماتریس‌های تجزیه، ترکیبی از الگوریتم‌های کمترین مربعات تکراری و به‌روزرسانی ضربی را استفاده می‌کنیم. چنان‌که قبلاً بیان شد، هدف به‌دست آوردن ماتریس‌های نامنفی W و H است، به‌گونه‌ای که داشته باشیم:

$$A \cong WH = \hat{A}$$

هر سطر ماتریس W نشان‌دهندهٔ میزان ارتباط بین کاربر متناظر آن سطر و گروه‌ها است. هم‌چنین هر ستون ماتریس H نشان‌دهندهٔ میزان ارتباط گروه‌ها و کالای نظیر آن ستون است. به‌عبارت دیگر، اگر سطر iام از ماتریس W را در نظر بگیریم، این سطر دارای k درایه است و هرکدام از این درایه‌ها نظیر یک گروه (ویژگی) است، اگر درایه‌ی lام این سطر بزرگ‌تر از سایر درایه‌های دیگر این سطر باشد، احتمال این‌که کاربر iام به گروه lام تعلق داشته باشد، بیش‌تر است. هم‌چنین اگر ستون jام ماتریس H را در نظر بگیریم، این ستون دارای k درایه است و هر کدام از این درایه‌ها متناظر با یک گروه (ویژگی) است، حال اگر درایه‌ی lام این ستون بزرگ‌تر از سایر درایه‌های این ستون باشد، احتمال این‌که اعضای گروه lام به کالای jام علاقه‌مند باشند، بیش‌تر است. بنابراین، برای پیش‌بینی امتیازی که کاربر iام به کالا jام می‌دهد، حاصل‌ضرب سطر iام ماتریس W در ستون jام ماتریس H را محاسبه می‌کنیم. فرض کنیم سطر iام ماتریس W را W_i و ستون jام ماتریس H را با H^j نشان دهیم. بنابراین داریم:

$$\widehat{a_{ij}} = W_i H^j = \sum_{l=1}^{k} w_{il} h_{lj}.$$

اختلاف بین امتیاز واقعی و امتیازی که تخمین‌زده‌ایم را خطا می‌نامیم و در این‌جا از مربع آن استفاده می‌کنیم زیرا امتیاز تخمین‌زده شده، هم می‌تواند بیش‌تر از امتیاز واقعی باشد و هم می‌تواند کم‌تر از آن باشد، از این‌رو، مقدار خطا ممکن است مثبت یا منفی باشد و برای پرهیز از اختلاف علامت، از مربع خطا استفاده می‌کنیم.

$$e_{ij}^2 = (a_{ij} - \widehat{a_{ij}})^2 = (a_{ij} - \textstyle\sum_{l=1}^{k} w_{il} h_{lj})^2 .$$

در واقع برای کمینه کردن (۱) که برابر است با:

$$f(W,H) = \frac{1}{2}\|A - WH\|_F^2 = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - \sum_{l=1}^{k} w_{il} h_{lj})^2.$$

جمع‌وندهای مجموع بالا یعنی $(a_{ij} - \sum_{l=1}^{k} w_{il} h_{lj})^2$ را کمینه می‌کنیم. ممکن است در طول عملیات تکراری، اندازه درایه‌های ماتریس تجزیه افزایش یابد که این افزایش موجب افزایش خطای محاسباتی در هر مرحله می‌شود. برای جلوگیری از بزرگ شدن بیش از حد درایه‌های ماتریس‌های تجزیه، ضریبی از اندازه اقلیدسی سطر و ستونی را که بر اساس آنها تخمین رتبه‌بندی صورت گرفته، به مقدار خطای حاصل از تخمین درایه نظیر آن سطر و ستون اضافه می‌کنیم تا در هر مرحله علاوه بر جلوگیری از بزرگ شدن خطا، از بزرگ‌شدن درایه‌های ماتریس‌های تجزیه نیز جلوگیری کنیم. بنابراین، مقدار خطای جدید را بدین صورت می‌نویسیم:

$$E_{ij}^2 = (a_{ij} - \sum_{l=1}^{k} w_{il} h_{lj})^2 + \beta\left(\|W_i\|^2 + \|H^j\|^2\right)$$
$$= (a_{ij} - \sum_{l=1}^{k} w_{il} h_{lj})^2 + \beta\left(\sum_{l=1}^{k} w_{il}^2 + \sum_{l=1}^{k} h_{lj}^2\right).$$

اکنون، هدف یافتن کمینه موضعی این خطا است. برای این کار لازم است بدانیم که مقادیر w_{il} و h_{lj} را در چه جهتی تغییر دهیم. به‌عبارت دیگر، نیاز داریم که گرادیان را به‌ازای این مقادیر بدانیم و برای فراهم کردن شرایط KKT و در نتیجه یافتن مقدار کمینه، در هر مرحله گرادیان را کاهش دهیم. از این‌رو، مشتق جزئی خطا را نسبت به w_{il} و h_{lj} محاسبه می‌کنیم:

$$\frac{\partial E_{ij}^2}{\partial w_{il}} = -2\left(a_{ij} - \widehat{a_{ij}}\right)\left(h_{lj}\right) + 2\beta w_{il} = -2e_{ij}h_{lj} + 2\beta w_{il},$$
$$\frac{\partial E_{ij}^2}{\partial h_{lj}} = -2\left(a_{ij} - \widehat{a_{ij}}\right)\left(w_{il}\right) + 2\beta h_{lj} = -2e_{ij}w_{il} + 2\beta h_{lj}$$

حال برای یافتن قوانین به‌روزرسانی، ضریبی از عبارت به‌دست آمده را با مقادیر قبلی جمع می‌کنیم. بنابراین، داریم:

$$\widetilde{w_{il}} = w_{il} + 2\alpha\left(e_{ij}h_{lj} + \beta w_{il}\right), \widetilde{h_{lj}} = h_{lj} + 2\alpha\left(e_{ij}w_{il} + \beta h_{lj}\right).$$

با همگرایی دنباله (W, H)، گرادیان صفر شده و از این‌رو، شرایط KKT فراهم می‌شود و الگوریتم به یک کمینهٔ موضعی همگرا می‌شود. پارامتر α عددی مثبت است که می‌تواند در طول برنامه ثابت باشد یا در هر تکرار درصورت نیاز به‌روزرسانی شود. معمولاً این پارامتر را عددی کوچک در نظر می‌گیریم تا مقدار کمینهٔ مورد نظر را از دست ندهیم. معمولاً در عمل پارامتر β را حدود $0/01$ تنظیم می‌کنند. بنابراین، با استفاده از قوانین به‌روزرسانی بالا و انتخاب درست و به اندازه کافی کوچک پارامتر α و انتخاب مناسب پارامتر β عملیات به‌روزرسانی را آنقدر تکرار می‌کنیم که خطا به میزان مطلوب، کاهش یابد.

معیار توقف

شرط‌های مختلفی می‌توانند به‌عنوان معیار توقف در نظر گرفته شوند. مثلاً می‌توان از مقایسه بین ماتریس داده‌ها (A) و ماتریس تقریب حاصل شده از الگوریتم $(\hat{A})$، شرط توقف را تعیین کرد. به این صورت که اختلاف بین درایه‌های ناصفر ماتریس اولیه و درایه‌های نظیرشان در ماتریس تقریب، از مقدار مشخصی تجاوز نکند و این مقدار هم می‌تواند با توجه به مسئله، به‌وسیلهٔ کاربر معین شود. در این روش، معیار توقف را خطای کلی مسئله در نظر می‌گیریم. مربع خطا را با استفاده از این معادله محاسبه می‌کنیم:

$$E_{ij}^2 = (a_{ij} - \sum_{l=1}^{k} w_{il}h_{lj})^2 + \beta\left(\sum_{l=1}^{k} w_{il}^2 + \sum_{l=1}^{k} h_{lj}^2\right).$$

حال اگر خطا، از مقدار مشخصی که به‌وسیلهٔ کاربر تعیین می‌شود (مثلاً ۰/۰۰۰۱) کوچک‌تر باشد، برنامه از عملیات تکراری خارج شده و نتیجه را نمایش می‌دهد.

مقداردهی اولیه

برای شروع الگوریتم‌های NMF ابتدا باید برای ماتریس‌های W و H مقادیر اولیه تعیین کنیم. مقداردهی اولیه ضعیف (مانند شروع با مقدار اولیه تصادفی) اغلب همگرایی آرام و گاهی جواب‌های بی‌ربط و غلطی را نتیجه می‌دهد. اگر روش همگرا باشد، یک مقدار اولیه خوب، می‌تواند به مقدار کافی تعداد تکرارها را کاهش دهد. کارایی بسیاری از الگوریتم‌های NMF تحت تأثیر انتخاب ماتریس‌های اولیه است و از این‌رو، مهم است که روش‌های سازگار و کارا برای مقداردهی اولیه عوامل ماتریسی در دست باشند [۸]. دراین روش به‌جای شروع با ماتریس‌های تصادفی، که بیش‌تر شدن تعداد تکرار روش تکراری و لذا بالا رفتن پیچیدگی محاسباتی و حجم بیش‌تر محاسبات را در پی دارد؛ از الگوریتم ساده و سریع به‌روزرسانی ضربی که در بخش (1.4) معرفی شد، استفاده می‌کنیم تا ماتریس‌های تصادفی را تبدیل به تقریبی از NMF کرده و سپس از این تقریب به‌عنوان مقدار اولیه برای برنامه اصلی استفاده می‌کنیم. از

آن‌جاکه الگوریتم به‌روزرسانی ضربی نسبت به الگوریتم ما، دارای محاسبات کم‌تر و ساده‌تری است، استفاده از این الگوریتم برای یافتن تقریبی از NMF، به صرفه‌تر است. البته می‌توانیم برای شروع، الگوریتم کم‌ترین مربعات متناوب را نیز به‌کار گیریم.

نتایج عددی

در این قسمت قصد داریم تا نتایج حاصل از پیاده‌سازی الگوریتم‌های اساسی NMF که معرفی شدند و الگوریتم ارائه شده در این مقاله را مقایسه و تحلیل کنیم.

مثال ۱.

در سامانه‌های توصیه‌گر مانند نت‌فلیکس[1] یا مووی‌لنز[2] گروهی از کاربران و مجموعه‌ای از کالاها (فیلم ها برای دو سامانه مذکور) وجود دارد. با توجه به این‌که هر کاربر برخی از کالاهای موجود در سامانه را رتبه‌بندی کرده است، پیش‌بینی می‌کنیم که چگونه کاربران، محصولات موردنظر خود را رتبه‌بندی می‌کنند، تا بدین‌وسیله بتوانیم توصیه‌هایی به آنها ارائه دهیم. تمام اطلاعاتی را که درمورد رتبه‌بندی‌های موجود داریم، می‌توان در یک ماتریس نشان داد. فرض کنید اکنون ۹ کاربر و ۸ کالا داریم و رتبه‌بندی و امتیازها اعداد صحیحی از ۱ تا ۵ هستند، ماتریس داده‌ها می‌تواند به‌صورت جدول ۱ باشد که U نماد کاربر و I نماد کالا است. هم‌چنین عدد صفر به این معنا است که کاربر هنوز کالا را رتبه‌بندی نکرده است.

جدول ۱. مثال ۱

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
U_1	0	0	0	5	5	5	0	0
U_2	0	0	0	5	0	5	0	0
U_3	0	0	0	5	5	5	0	0
U_4	5	5	5	0	0	0	0	0
U_5	5	5	5	0	0	0	0	0
U_6	5	5	0	0	0	0	0	0
U_7	0	0	0	0	0	0	5	5
U_8	0	0	0	0	0	0	0	5
U_9	0	0	0	0	0	0	5	5

یکی از روش‌های تخمین رتبه‌دهی‌ها، روش تجزیه نامنفی ماتریس داده‌ها است. چنان‌که در جدول (۱) مشخص است، کاربرها به سه گروه تعلق دارند کاربران ۱، ۲، و ۳ با هم در یک گروه (مثلاً گروه ۱) هستند، کاربران ۴، ۵ و ۶ در یک گروه (مثلاً گروه ۲) و کاربران ۷، ۸ و ۹ در یک گروه (مثلاً گروه ۳) هستند.

باتوجه به درایه‌های ماتریس (که همان رتبه‌دهی کاربران به کالاها است) و مشابهت‌های رفتاری کاربران، می‌بینیم که گروه اول به کالاهای ۴، ۵ و ۶، گروه دوم به کالاهای ۱، ۲ و ۳ و گروه سوم به کالاهای ۷ و ۸ علاقه‌مند هستند.

با توجه به مشابهت رفتاری کاربر ۲ بـه کاربران ۱ و ۳، کاربر ۶ بـه کاربران ۴ و ۵ و کاربر ۸ به کاربران ۷ و ۹ (که با

1. Netflix
2. MovieLens

توجه به همین شباهت‌های رفتاری هم‌گروه شده‌اند)، انتظار داریم این کاربران نیز، نظری مشابه هم‌گروهی‌هایشان داشته باشند. یعنی پس از اجرای الگوریتم انتظار می‌رود درایه‌های (۲,۵)، (۶,۳) و (۸,۷) با امتیاز ۵ مقداردهی شوند.

الگوریتم‌های بیان شده را برای ماتریس بالا به‌ازای $k = 3$ اجرا می‌کنیم. الگوریتم‌های ارائه شده در بخش چهارم، پایه‌ی الگوریتم نرم‌افزار متلب هستند و برای تعداد تکرار به‌اندازهٔ کافی، نتیجهٔ آن تقریباً برابر نتیجه حاصل از نرم‌افزار متلب است. از این‌رو، در تخمین‌های ذیل به‌جای الگوریتم‌های بخش ۴، از نرم‌افزار متلب استفاده کرده‌ایم.

جدول۲. تخمین رتبه‌بندی مثال ۱ با استفاده از نرم‌افزار متلب

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
U_1	0/00	0/00	0/00	5/39	3/94	5/39	0	0/00
U_2	0	0	0	3/94	2/98	3/94	0	0
U_3	0/00	0/00	0/00	5/39	3/94	5/39	0	0
U_4	5/39	5/39	3/94	0	0/00	0	0/00	0
U_5	5/39	5/39	3/94	0	0/00	0	0/00	0
U_6	3/94	3/94	2/98	0	0/00	0	0	0
U_7	0	0	0/00	0	0	0	4/32	5/53
U_8	0	0	0/00	0/00	0/00	0/00	2/42	3/10
U_9	0	0	0/00	0	0	0	4/32	5/53

جدول ۳.تخمین رتبه‌بندی مثال ۱ با استفاده از الگوریتم پیشنهادی

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
U_1	0	0	0	4/99	4/99	4/99	0	0
U_2	0	0	0	4/99	4/99	4/99	0	0
U_3	0	0	0	4/99	4/99	4/99	0	0
U_4	4/99	4/99	4/99	0	0	0	0	0
U_5	4/99	4/99	4/99	0	0	0	0	0
U_6	4/99	4/99	4/99	0	0	0	0	0
U_7	0	0	0	0	0	0	4/99	4/99
U_8	0	0	0	0	0	0	4/99	4/99
U_9	0	0	0	0	0	0	4/99	4/99

با توجه به نتایج در جدول ۲ می‌بینیم که تخمین حاصل از نرم‌افزار متلب برای درایه‌های مذکور، تقریباً امتیاز ۳ را به‌دست آورده است و هم‌چنین برای درایه‌های مجاور آنها و درایه‌های موجود در سطر و ستون نظیر آنها تقریباً امتیاز ۴ را به‌دست آورده است و برای سایر درایه‌های ناصفر دیگر امتیاز بیش‌تر از ۵ یعنی ۵/۳۹ را محاسبه کرده است. در حالی‌که انتظار داریم نه‌تنها برای درایه‌های ناصفر که مقدارشان معلوم است تخمینی نزدیک به‌همان مقدار (در این مثال، عدد ۵) را به‌دست آورد، بلکه برای درایه‌های (۲,۵)، (۶,۳) و (۸,۷) نیز عددی نزیک ۵ تخمین بزند. اکنون با بررسی نتیجه حاصل از الگوریتم پیشنهادی در جدول ۳ می‌بینیم که علاوه بر درایه‌های ناصفر، برای سه درایهٔ مذکور نیز تقریباً امتیاز ۵ را به‌دست آورده است و این دقیقاً همان چیزی است که انتظار داشتیم.

مثال ۲

حال فرض کنیم اطلاعات بیش‌تری از نظرات کاربران داشته باشیم و ماتریس داده‌ها به‌صورت جدول ۴ باشد.

جدول۴. مثال ۲

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
U_1	0	2	1	5	5	5	3	4
U_2	1	0	1	5	0	5	۳	3
U_3	1	2	1	5	5	5	3	0
U_4	5	5	5	3	4	0	2	1
U_5	5	5	5	0	4	3	0	1
U_6	5	5	0	3	0	3	1	2
U_7	2	0	3	1	1	0	5	5

چنان‌که مشاهده می‌کنیم با افزایش داده‌ها هم‌گروهی‌ها تغییر نکرده‌اند. انتظار داریم تخمین الگوریتم‌ها برای درایه‌های ناصفر، تقریباً برابر با مقدار خودشان در ماتریس داده‌ها باشد و هم‌چنین باتوجه به این‌که اکثر درایه‌های ماتریس دارای مقدار ناصفرند و درایه‌های مجاور صفرهای ماتریس، همگی ناصفر می‌باشند، انتظار داریم که الگوریتم‌ها برای همه درایه‌های صفر، تخمینی ناصفر و متناسب با نظرات سایر اعضای گروه مربوط، ارائه دهند.

$$\widehat{a_{11}} = 1, \widehat{a_{22}} = 2, \widehat{a_{25}} = 5, \widehat{a_{38}} = 3/5, \widehat{a_{46}} = 3, \widehat{a_{54}} = 3, \widehat{a_{57}} \sim 2, \widehat{a_{63}} = 5, \widehat{a_{65}} = 4, \widehat{a_{72}} \sim 2, \widehat{a_{76}} = 2, \widehat{a_{85}} = 1, \widehat{a_{87}} = 5, \widehat{a_{91}} \sim 3, \widehat{a_{94}} = 1.$$

دوباره الگوریتم‌های قبل را روی آنها اجرا می‌کنیم.

جدول ۵. تخمین رتبه‌بندی مثال ۲ با استفاده از نرم افزار متلب

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
U_1	0/74	1/52	1/57	5/43	3/55	5/38	3/79	3/01
U_2	0/29	0/95	1/34	4/29	2/65	4/27	3/44	2/92
U_3	1/42	2/11	1/00	5/56	4/00	5/39	2/35	1/17
U_4	5/21	5/34	4/35	1/79	3/07	2/10	1/05	1/82
U_5	5/50	5/56	4/21	1/50	3/04	1/79	0/41	1/15
U_6	3/45	3/66	2/61	2/30	2/75	2/42	0/84	0/94
U_7	0/75	1/05	3/39	0/62	0/39	1/05	4/02	5/34
U_8	2/94	3/10	3/89	0/68	1/38	1/08	2/38	3/56
U_9	1/32	1/67	3/91	0/99	0/83	1/45	4/29	5/69

جدول۶. تخمین رتبه‌بندی مثال ۲ با استفاده از الگوریتم پیشنهادی

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
U_1	0/93	1/98	1/01	4/98	4/96	5/03	3/24	3/73
U_2	1/03	2/04	1/00	4/98	5/03	4/93	2/76	3/23
U_3	0/99	2/01	1/01	4/97	4/99	4/97	2/98	3/47
U_4	4/99	5/00	5/00	2/97	3/96	3/03	1/51	1/47
U_5	5/01	5/00	4/94	2/97	4/01	2/94	1/13	1/06
U_6	4/94	4/96	4/95	3/00	3/97	3/05	1/52	1/48
U_7	2/01	2/18	2/99	1/03	0/97	2/04	4/93	5/02
U_8	2/98	3/01	3/98	0/99	1/14	2/03	4/93	4/95
U_9	2/96	2/98	3/98	0/90	1/05	1/96	4/98	4/99

نتایج نرم‌افزار متلب را با نتایجی که انتظار داشتیم مقایسه می‌کنیم. چنان‌که در جدول ۵ مشاهده می‌کنیم، بعضی از درایه‌هایی که مقدار ناصفر داشتند و انتظار داریم که تخمین آنها نیز به مقادیر اولیه‌شان نزدیک باشد، به میزان چشم‌گیری از مقادیر اصلی‌شان دور هستند. به‌عنوان مثال درایه‌های (۱،۵) و (۶،۱) که مقدار آنها در ماتریس داده‌ها برابر ۵ بوده است، تخمین آن در ماتریس حاصل از نرم‌افزار متلب،حدود ۳/۵ و درایه‌های (۴،۴)، (۵،۶) و (۹،۲) که مقدار اولیه آنها برابر ۳ است، تخمین آن حدود ۱/۷ است. و به‌همین ترتیب درایه‌های (۴،۷)، (۶،۸)، (۷،۱)، (۸،۶)، (۲،۱)، (۴،۸)، (۷،۵) و (۱،۸). درمورد درایه‌های صفرِ ماتریس اولیه نیز تقریباً اکثر این درایه‌های به‌دست آمده با مقدار مورد انتظار دارای اختلاف معناداری هستند. به‌عنوان مثال برای درایه‌های (۲،۵)، (۶،۳) و (۸،۷) که انتظار داشتیم با توجه به نظرات هم‌گروهی‌هایشان، عدد ۵ یا عددی نزدیک به آن را تخمین بزند، به ترتیب امتیازهای ۲/۶۵، ۲/۶۱ و ۲/۳۸ به‌دست آمده است. هم‌چنین برای درایه‌های (۴،۶)، (۵،۴) و (۹،۱) که انتظار امتیازی نزدیک به ۳ داشتیم، به‌ترتیب امتیازهای ۲/۱۰، ۱/۵۰ و ۱/۳۲ تخمین زده شده است. سایر درایه‌ها نیز با رجوع به جدول‌های ۴ و ۵ و مقایسه آنها به روشنی مشخص است. در حالی‌که با بررسی نتایج حاصل از الگوریتم پیشنهادی در جدول ۶، مشاهده می‌کنیم که تخمین تمام درایه‌هایی که در ماتریس اولیه مقدار ناصفر داشته‌اند، تقریباً بسیار نزدیک به همان مقدار اولیه‌شان هستند به‌جز درایه‌های (۱،۷) ، (۱،۸) ، (۲،۷) و (۲،۸) که انتظار داشتیم، به‌ترتیب، مقادیرشان

برابر ۳، ۴، ۳ و۳ باشد، در حالی‌که مقادیر ۳/۲۴، ۳/۷۳، ۲/۷۶ و ۳/۲۳ به‌دست آمده‌اند و علت این اختلاف‌ها نیز آن است که کاربر اول و دوم نظرات مشترک زیادی با یک‌دیگر دارند ولی درمورد کالای هشتم امتیازات آنها دارای اختلاف است، در این وضعیت باتوجه به شباهت و نقاط اشتراکی که بین آنها وجود دارد، برنامه سعی در تعدیل این اختلاف می‌کند و به همان مقدار که از امتیاز درایه (۱,۸) می‌کاهد، به امتیاز درایه (۲,۸)، می‌افزاید. از طرف دیگر نیز این احتمال را می‌دهد که ممکن است در مورد کالای هفتم (که امتیاز کاربر اول و دوم به این کالا نزدیک به امتیازی است که به کالای هشتم داده‌اند) نیز مانند کالای هشتم، اختلاف نظر داشته باشند؛ بنابراین امتیاز کاربر اول (که به کالای هشتم امتیازی بیش‌تر از کاربر دوم داده است) به کالای هشتم را کمی بیش‌تر از ۳، یعنی ۳/۲۴ درنظر می‌گیرد و امتیاز کاربر دوم به این کالا را کمی کم‌تر از ۳، یعنی ۲/۷۶ تخمین می‌زند. و همین طور است برای درایه‌های (۴,۷)، (۴,۸)، (۶,۷) و (۶,۸). درمورد درایه‌های صفر ماتریس اولیه نیز تخمین ها تقریباً برابر با اعدادی است که انتظارشان را داشتیم به‌جز دو درایه (۳,۹) و (۵,۸). درمورد (۳,۹) از آن‌جایی‌که برنامه، نظرات هم‌گروهی‌های کاربر سوم نسبت به کالای نهم را ۳/۷۶ و ۳/۲۳ تخمین زده‌بود، میانگین این دو عدد یعنی ۳/۴۷ را برای نظر کاربر سوم به کالای نهم به‌دست آورده است.

ارزیابی

ارزیابی پیش‌بینی‌ها برای شناخت قدرت مدل‌های پیش‌بینی کننده اهمیت بسیاری دارد و بر اساس ارزیابی‌ها می‌توان دقت یک مدل را بهبود بخشید. در این بخش برنامه‌ها را روی حجم بیش‌تری از داده‌های واقعی اعمال می‌کنیم و برای تحلیل عملکرد سامانه‌های توصیه‌گر، از سه معیار ارزیابی خطا استفاده می‌کنیم و نتایج را روی نمودار نمایش می‌دهیم.

معیارهای خطا

معیارهای خطا، روی داده‌های ناصفر ماتریس داده‌ها اعمال می‌شوند. درواقع، داده‌های حقیقی داده‌های ناصفر هستند و داده صفر به‌معنای عدم وجود داده است. از این‌رو، در محاسبات، خطا را به‌ازای داده‌های ناصفر محاسبه می‌کنیم. فرض کنیم داده‌های ناصفر ماتریس اولیه را در برداری مانند r قرار دهیم و تخمینی که سامانه توصیه‌گر متناظر با بردار داده‌های ناصفر به‌دست می‌آورد را در بردار p ذخیره کنیم و z تعداد درایه‌های بردارهای مذکور باشد، یعنی $p, r \in R^z$.

میانگین خطای مطلق[1] برابر است با میانگین قدرمطلق خطاها که از این رابطه محاسبه می‌شود:

$$MAE_j = \frac{1}{j}\sum_i^j |r_i - p_i| \ , \ j = 1,2, \dots, z.$$

جذر میانگین مربع خطا[2] از این رابطه به‌دست می‌آید:

$$RMSE_j = \sqrt{\frac{1}{j}\sum_i^j (r_i - p_i)^2} \ , \quad j = 1,2, \dots, z.$$

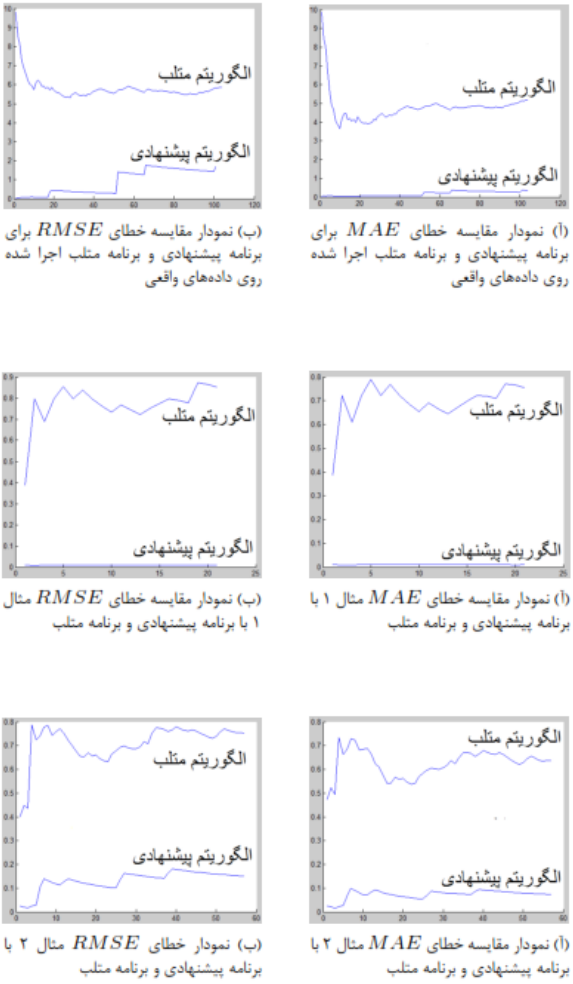
خطای نسبی[3] یکی از معیارهای رایج محاسبه خطا است که از این رابطه محاسبه می‌شود:

$$RE_i = \frac{|r_i - p_i|}{|r_i|} \ , \quad i = 1,2, \dots, z.$$

1. Mean absolute error
2. Root mean square error
3. Relative error

نمایش خطا

نمودارهای مربوط به معیار MAE و $RMSE$ چنان‌که در شکل ۱ نمایش داده شده است، به‌صورت **پیوسته** رسم شده و محور افقی و عمودی، به‌ترتیب، بیان‌گر تعداد داده‌ها و مقدار خطا است. نمودارهای مربوط به خطای نسبی که در شکل ۲ آمده است، به‌صورت نقطه‌ای ترسیم شده‌اند و محور افقی و عمودی، به‌ترتیب، بیان‌گر اندیس داده‌ها و مقدار خطای نظیر هر اندیس است.



(ا) نمودار مقایسه خطای MAE برای برنامه پیشنهادی و برنامه متلب اجرا شده روی داده‌های واقعی

(ب) نمودار مقایسه خطای $RMSE$ برای برنامه پیشنهادی و برنامه متلب اجرا شده روی داده‌های واقعی

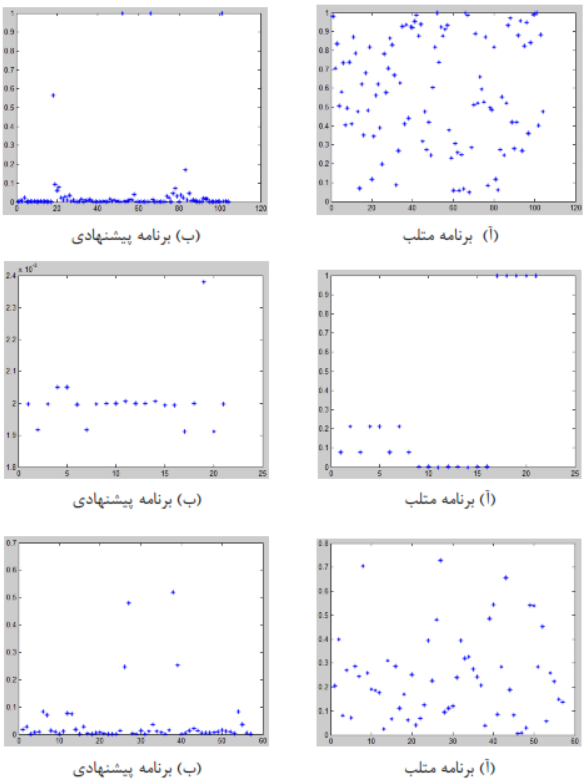
(ا) نمودار مقایسه خطای MAE مثال ۱ با برنامه پیشنهادی و برنامه متلب

(ب) نمودار مقایسه خطای $RMSE$ مثال ۱ با برنامه پیشنهادی و برنامه متلب

(ا) نمودار مقایسه خطای MAE مثال ۲ با برنامه پیشنهادی و برنامه متلب

(ب) نمودار خطای $RMSE$ مثال ۲ با برنامه پیشنهادی و برنامه متلب

شکل ۱. نمودار خطای MAE و $RMSE$



شکل ۲. نمودار خطای نسبی

جمع‌بندی

در دنیای امروز که کاربران با حجم وسیعی از داده‌ها مواجه هستند، تولید سامانه‌هایی که بتوانند بر اساس نیاز و جایگاه کاربران اطلاعات مفیدی را در اختیار آنها قرار دهند اهمیت بسیاری دارد. در میان این سامانه‌ها، سامانه‌های توصیه‌گر از استقبال ویژه‌ای برخوردار شده‌اند، به این علت که بر اساس اطلاعات استخراج شده از داده‌ها علایق و نیازهای کاربران را پیش‌بینی کرده و به آنها توصیه می‌کنند.

در این مقاله یک سامانه توصیه‌گر که از روش تجزیه ماتریس برای ساخت آن استفاده شده است، ارائه شد. به منظور افزایش دقت تجزیه ماتریس، مسئله را به یک مسئله خطای کم‌ترین مربعات تبدیل کرده و با روش به‌روز رسانی ضربی آن را حل کرده‌ایم. از طرفی برای کنترل مقادیر درایه‌های ماتریس‌های تجزیه، از روش منظم‌سازی استفاده کردیم به‌طوری که پارامترهایی را که ضریبی از ماتریس‌های تجزیه هستند، به مسئله خطای کم‌ترین مربعات اضافه نمودیم. ارزیابی‌های انجام شده نشان‌دهندهٔ افزایش دقت در تشخیص و پیش‌بینی امتیاز کاربران به کالاهای جدید است.

با تشکر از دکتر اسمعیل بابلیان که در این مسیر علمی همراه ما بودند.

منابع

1. Fu X., Huang K., Sidiropoulos N. D., and Ma W., "Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications",IEEE Signal Processing Letters, 25(3) (2018) 328-332.
2. Koren Y., Bell R., Volinsky Ch., "Matrix factorization techniques for recommender system", IEEE Computer ,42 Issue 8 (2009) 30-37.
3. Lin C. J., "On the convergence of multiplicative update algorithms for nonnegative matrix factorization", IEEE Transactions on Nueral Network ,18(6)(2005) 1589-1596.
4. Lin C. J., "Projected gradient methods for nonnegative matrix factorization", Nueral Computation ,19(10) (2005) 2756-2779.
5. Berry M. W., Browne M., Langville A. N., Pauca V. P., Plemmons R. J., "Algorithms and applications for approximate nonnegative matrix factorization", Computational Statistics and Data Analysis, 1 (2007) 155-173.
6. Kim H., Park H., "Nonnegative matrix factorization based on alternating nonnegativity-constrained least squares and the active set method", SIAM J. MatrixAnal. Appl., 30(2) (2008) 713-730.
7. Kim H. , Park H., "Sparse non-negative matrix factorization via automating non-negativity-constrained least squares for microarray data analysis", Bioinformatics, 23 (2007), 1495-1502.
۸. یوسفی مهسا، رزقی منصور،" تجزیه نامنفی ماتریس: روشی برای تحلیل داده‌های نامنفی"، فرهنگ و اندیشه ریاضی، شمارهٔ ۵۹ (پاییز و زمستان ۱۳۹۵) ۷۱ تا ۹۰.
9. Bertsekas D. P.,"Nonlinear Programming", 2nd. ed., Athena Scientific, Belmont, Massachusetts, 1999.
10. Lee D. D., Seung H. S., "Algorithms for non-negative matrix factorization", Advanced in Neural Information Processing, 13 (2001).
11. Langville A. N., Meyer C. D., Albright R., Cox J., Duling D.," Algorithms, initializations, and convergence for the nonnegative matrix factorization", CoRR abs/1407.7299 (2014).