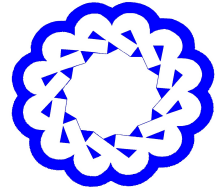


موجک‌ها و جبرخطی

<http://wala.vru.ac.ir>



دانشگاه ولیعصر (عج)

رفسنجان

بهبود برخی از روش‌های حل مسئله‌ی تکمیل ماتریس فائزه آقامحمدی^آ، فاطمه شاکری^آ

گروه ریاضی کاربردی، دانشکده ریاضی و علوم کامپیوتر، دانشگاه صنعتی امیرکبیر، استان تهران، ایران

چکیده

یکی از روش‌های جدید بازیابی اطلاعات، تکمیل ماتریس می‌باشد و از آنجایی که اغلب داده‌ها از قبیل صوت، تصویر، فیلم و داده‌های عددی قابل تبدیل به ماتریس می‌باشند این روش برای حل مسائل بازیابی اطلاعات بسیار مفید است. مسئله‌ی تکمیل ماتریس براساس مینیمم‌سازی رتبه‌ی ماتریس ناقص، به تکمیل ماتریس و بازیابی اطلاعات از دست‌رفته می‌پردازد به طوری که رتبه‌ی ماتریس تکمیل شده مینیمم شود. تاکنون روش‌ها و الگوریتم‌های متعددی نظیر روش‌های مبتنی بر نرم مرکزی، روش‌های مبتنی بر رتبه و روش‌های موسوم به خودضربی برای حل این مسئله ارائه شده است. روش‌های مبتنی بر نرم مرکزی به علت نیاز به محاسبه‌ی تجزیه مقدار منفرد در هر تکرار از الگوریتم ارائه شده برای حل مسئله، دارای پیچیدگی محاسباتی زیادی بوده و خصوصاً در ابعاد بزرگ ناکارآمد می‌باشند. موجک‌ها و جبرخطی (۱۴۰۰) ©

اطلاعات مقاله

تاریخچه مقاله:
دریافت شده: ۵ تیر ۱۳۹۸
پذیرفته شده: ۳۰ فروردین ۱۳۹۹
دسترسی آنلاین: ۱۱ اردیبهشت ۱۴۰۰
ادیتور رابط: فاطمه پنجه علی
بیک

کلمات کلیدی:

مسئله‌ی تکمیل ماتریس
رتبه‌پایین، روش‌های
مبتنی بر رتبه، روش
خودضربی، پیچیدگی
محاسباتی، روش جهت
متناوب ضرایب.

همچنین به دلیل عدم وجود الگوریتم‌های مفید و کارآمد برای به‌روز رسانی رتبه‌ی ماتریس، روش‌های مبتنی بر رتبه که اساس آن‌ها تجزیه‌ی ماتریس به چند عامل می‌باشد روی داده‌های واقعی مانند تصاویر، دارای خطای بیشتری نسبت به سایر روش‌ها می‌باشند و روش‌های خودضربی (SR)^۱، خصوصاً دو روش تقریب کمترین مربعات (SR-LS)^۲ و روش رتبه‌پایین (SR-LR)^۳ دارای نتایج بهتری هستند. اما عیب بزرگ روش‌های خودضربی خصوصاً روش رتبه‌پایین، پیچیدگی محاسباتی و صرف زمان زیاد برای رسیدن به همگرایی می‌باشد. در این مقاله به ارائه‌ی چند راهکار برای افزایش سرعت روش‌های موسوم به خودضربی می‌پردازیم. نتایج عددی حاصل از اجرای راهکارهای معرفی شده در مقاله روی این روش‌ها نشان دهنده‌ی افزایش سرعت و بهبود کارایی این روش‌ها از نظر پیچیدگی محاسباتی می‌باشد.

۱. مقدمه

پایه‌ی بسیاری از علوم تجربی، جمع‌آوری داده‌ها و اطلاعات و نتیجه‌گیری بر اساس آن‌ها می‌باشد. اما جمع‌آوری اطلاعات همراه با مشکلات زیادی نظیر از بین رفتن داده‌ها و اختلال در داده‌ها به‌وسیله‌ی عوامل خارجی می‌باشد و قطعاً جمع‌آوری مجدد داده‌ها در صورت از بین رفتن و تغییر بخشی از آن‌ها، بسیار پرهزینه و گاهی اوقات ناممکن می‌باشد. بنابراین در عصر اطلاعات، بازیابی و بازسازی اطلاعات تغییر یافته با عوامل خارجی یکی از مسائل مهمی است که با آن سر و کار داریم. یکی از روش‌های جدید بازیابی اطلاعات که در سال‌های اخیر توجه بسیاری از پژوهشگران رشته‌ها و علوم مختلف مانند رادار، شناسایی سیستم، تشخیص حرکات در ویدئو، کاهش مدل، تشخیص الگو، یادگیری ماشین، آمار، شیمی، نظریه سیستم، بهینه سازی گسسته، مترام سازی داده‌ها و ... را به خود جلب کرده است مسئله‌ی تکمیل ماتریس رتبه‌پایین می‌باشد [۹، ۴، ۱، ۵]. اساس این مسئله، تکمیل ماتریس و بازیابی اطلاعات از دست‌رفته می‌باشد به‌طوری که رتبه‌ی ماتریس تکمیل شده مینیمم شود. فرض رتبه‌پایین بودن در مسئله‌ی تکمیل ماتریس فرض مفید و کارآمدی است؛ زیرا ستون‌های ماتریس‌های مرتبط به بسیاری از داده‌ها متعلق به یک زیرفضا با بعد پایین می‌باشند [۳] که همین امر منجر به تشکیل یک ماتریس رتبه‌پایین می‌

*نویسنده مسئول

آدرس ایمیلها: yas1365@aut.ac.ir (فائزه آقامحمدی)، f.shakeri@aut.ac.ir (فاطمه شاکری).

<http://doi.org/10.22072/wala.2020.110303.1233>

موجک‌ها و جبرخطی (۱۴۰۰) ©

¹self-representation

²least square self-representation

³low rank self-representation

شود. به‌عنوان مثال در خدمات ارائه‌ی فیلم، ماتریس امتیاز فیلم همیشه ناقص است زیرا افراد معمولاً به تعداد کمی از مجموعه‌ی فیلم‌ها امتیاز می‌دهند. اما از آنجا که خریداران مختلف ممکن است با سلیقه‌ی شبیه به هم به فیلم‌های مختلف امتیاز یکسان بدهند، ماتریس امتیاز فیلم می‌تواند رتبه‌پایین باشد. با کامل کردن ماتریس امتیاز فیلم، می‌توان بسته‌های سفارشی تولید کرد. در مسائل پردازش تصویر نیز ماتریس پیکسل‌های یک تصویر می‌تواند رتبه‌پایین باشد؛ زیرا ستون‌ها و ردیف‌های مختلف تصویر ممکن است دارای وضوح یا بافت یکسان باشند. ویژگی رتبه‌پایین بودن ماتریس پیکسل، به ما در از بین بردن نویزها و بازیابی قسمت‌های مبهم تصویر نیز کمک می‌کند [۳].

در این مقاله حروف بزرگ و ایتالیک به صورت A نشانگر یک ماتریس در فضای $\mathbb{R}^{m \times n}$ ، A^T و A_{ij} به ترتیب ترانهاده و درایه‌ی (i, j) ماتریس A می‌باشند. P_Ω عملگر تصویر روی مجموعه‌ی Ω ، زیر مجموعه‌ای از مجموعه‌ی اندیس‌های ماتریس A ، به صورت زیر تعریف می‌شود [۶]

$$P_\Omega(\cdot) : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n},$$

$$P_\Omega(A_{ij}) = \begin{cases} A_{ij} & (i, j) \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \quad (1.1)$$

$\text{Rank}(\cdot)$ بیانگر رتبه‌ی ماتریس، $\|\cdot\|_*$ موسوم به نرم مرکزی، برابر با مجموع مقادیر منفرد ناصفر یک ماتریس می‌باشد. D_τ عملگر کاهش مقدار منفرد^۴ بوده و به صورت زیر تعریف می‌شود [۲]

$$D_\tau(A) = U D_\tau(\Sigma) V^T, \quad D_\tau(\Sigma) = \text{diag}(\{\sigma - \tau\}_+). \quad (2.1)$$

که

$$\{\sigma - \tau\}_+ = \begin{cases} \sigma - \tau & \sigma > \tau, \\ 0 & \text{otherwise.} \end{cases} \quad (3.1)$$

⁴singular value thresholding (SVT)

$A = U\Sigma V^T$ تجزیه‌ی مقدار منفرد ماتریس $A \in \mathbb{R}^{m \times n}$ با $\text{Rank}(A) = r$ بوده و $\Sigma = \text{diag}(\{\sigma_i\}_{1 \leq i \leq r})$ می‌باشد که $\sigma_i, i = 1, \dots, r$ مقادیر منفرد ماتریس A می‌باشند. همچنین $\text{QR}(\cdot)$ عملگر محاسبه‌ی عامل متعامد در تجزیه‌ی QR یک ماتریس دلخواه می‌باشد.

با استفاده از ویژگی رتبه‌پایین بودن ماتریس، مسئله‌ی تکمیل ماتریس را می‌توان به صورت زیر بیان کرد

فرض کنیم $W \in \mathbb{R}^{m \times n}$ ماتریس داده‌ها با N درایه‌ی مشخص باشد، اگر Ω مجموعه‌ی اندیس‌های درایه‌های شناخته شده با $|\Omega| = N$ و دارای توزیع تصادفی و یکنواخت باشد، آنگاه مسئله‌ی تکمیل ماتریس رتبه‌پایین را می‌توان به صورت زیر ارائه داد [۹]

$$\begin{aligned} \min_X \text{Rank}(X), \\ \text{s.t. } P_\Omega(X) = P_\Omega(W), \end{aligned} \quad (۴.۱)$$

مسئله‌ی (۴.۱) از نظر پیچیدگی محاسباتی غیرچندجمله‌ای و یک مسئله‌ی نامحدب می‌باشد [۶، ۲]. از آنجایی که رتبه‌ی یک ماتریس برابر با تعداد مقادیر منفرد ناصفر ماتریس می‌باشد، می‌توان تابع $\text{Rank}(\cdot)$ را با تابع $\|X\|_*$ که صورت تخفیف یافته‌ی محدب^۵ تابع هدف مسئله‌ی (۴.۱) می‌باشد جایگزین کرد [۲]. با جایگزینی تابع $\|X\|_*$ در مسئله‌ی (۴.۱)، مسئله‌ی تکمیل ماتریس به صورت زیر نوشته می‌شود

$$\begin{aligned} \min_X \|X\|_*, \\ \text{s.t. } P_\Omega(X) = P_\Omega(W). \end{aligned} \quad (۵.۱)$$

بنابراین برای یافتن جواب مسئله‌ی تکمیل ماتریس می‌توان به جای مسئله‌ی (۴.۱)، مسئله‌ی (۵.۱) را

⁵convex relaxation

حل کرد. حل این مسئله با استفاده از الگوریتم تکراری تجزیه‌ی مقدار منفرد به صورت زیر می‌باشد

$$\begin{cases} X^k = D_\tau(Y^{k-1}) \\ Y^k = Y^{k-1} + \delta P_\Omega(W - X^k) \end{cases} \quad (۶.۱)$$

که $\tau > 0$ ، δ طول گام مثبت و Y^0 مقدار اولیه می‌باشد. اما این الگوریتم به خاطر محاسبه‌ی تجزیه‌ی مقدار منفرد و ضرب ماتریسی در هر تکرار آن، در ابعاد بزرگ بسیار پرهزینه می‌باشد [۱۰]. با فرض رتبه‌پایین بودن ماتریس می‌توان نتیجه گرفت که بین برخی سطرها و ستون‌های ماتریس یک رابطه‌ی خطی برقرار است و با استفاده از این رابطه‌ی خطی می‌توان درایه‌های ناشناخته‌ی آن را محاسبه کرد [۱۰]. راهکارها و روش‌های تکمیل ماتریس و تقریب یک ماتریس رتبه‌پایین نیز بر اساس همین وابستگی بین ستون‌ها - که ناشی از ساختار رتبه‌پایین این مسائل است - ارائه شده است [۳].

۲. روش‌های حل مسئله‌ی تکمیل ماتریس

در این بخش، به معرفی برخی روش‌های موجود برای حل مسئله‌ی تکمیل ماتریس پرداخته و مزایا و معایب این روش‌ها را بررسی می‌کنیم.

۱.۲. روش‌های مبتنی بر رتبه

روش‌های مبتنی بر رتبه که بر اساس تجزیه‌ی ماتریس ارائه می‌شوند، دسته‌ی مهمی از روش‌های حل مسئله‌ی تکمیل ماتریس می‌باشند. این روش‌ها ماتریس ناقص $m \times n$ با رتبه‌ی $r < \min(m, n)$ را با استفاده از تجزیه‌ی ماتریس به دو یا سه عامل، بازیابی می‌کنند که ضرب آنها ماتریس تکمیل شده می‌باشد و به دلیل حذف محاسبه‌ی تجزیه‌ی مقدار منفرد در هر تکرار یا محاسبه تجزیه‌ی مقدار منفرد روی یک ماتریس با ابعاد کوچک‌تر، دارای پیچیدگی محاسباتی کمی می‌باشند [۹، ۶]. اما نکته‌ی حائز اهمیت در مورد این دسته از روش‌ها این است که با وجود ناقص بودن ماتریس، برای بازیابی آن با استفاده از این روش‌ها نیازمند شناسایی رتبه‌ی ماتریس می‌باشیم که تاکنون الگوریتم کارامدی برای آن ارائه نشده است [۱۱]. بنابراین خطای روش‌های مبتنی بر رتبه با وجود پیچیدگی محاسباتی کم و دقت بالای آنها روی داده‌های مصنوعی با رتبه‌ی معلوم، به دلیل عدم وجود الگوریتم‌های مفید و کارآمد برای به‌روز رسانی رتبه‌ی ماتریس، روی داده‌های واقعی مانند تصاویر افزایش می‌یابد و در نتیجه این روش‌ها فاقد کارایی

لازم برای بازیابی ماتریس رتبه پایین می‌باشند. در این مقاله به معرفی یکی از مهم‌ترین روش‌های مبتنی بر رتبه بسنده می‌کنیم.

۱.۱.۲. روش تجزیه‌ی سه‌تایی سریع (FTF)

برای کاهش هزینه‌ی محاسباتی تجزیه‌ی مقدار منفرد در روش تکراری (۶.۱) در [۶] یک روش تجزیه‌ی سه‌تایی سریع^۶ ارائه شده است. در این روش با قرار دادن $X = LMR$ به طوری که $L \in \mathbb{R}^{m \times r}$ ، $M \in \mathbb{R}^{r \times r}$ و $R \in \mathbb{R}^{r \times n}$ که $r \ll \min(m, n)$ و $L^T L = I$ و $RR^T = I$ ، مسئله‌ی (۵.۱) به یک مسئله‌ی مینیمم‌سازی کوچک‌تر روی ماتریس $M \in \mathbb{R}^{r \times r}$ تبدیل و یک مدل بر اساس روش جهت متناوب ضرایب^۷ برای حل آن ارائه می‌گردد. با اعمال $X = LMR$ در مسئله‌ی (۵.۱)، مسئله به صورت زیر خواهد بود

$$\begin{aligned} & \min_{M, L, R} \|M\|_* \\ \text{s.t. } & P_{\Omega}(LMR - W) = \circ, \\ & L^T L = I, \\ & RR^T = I. \end{aligned} \quad (1.2)$$

مسئله‌ی (۱.۲) با استفاده از روش جهت متناوب ضرایب و استفاده از متغیر کمکی $Z = LMR$ قابل حل می‌باشد. با افزودن متغیر کمکی، مسئله (۱.۲) به صورت زیر بازنویسی می‌شود

$$\begin{aligned} & \min_{M, L, R, Z} \|M\|_* \\ \text{s.t. } & P_{\Omega}(Z - W) = \circ, \\ & Z = LMR \\ & L^T L = I, \\ & RR^T = I. \end{aligned} \quad (2.2)$$

^۶Fast Tri Factorization

^۷ Alternating direction method of multipliers (ADMM)

لاگرانژین افزوده‌ی مسئله‌ی (۲.۲) به شکل زیر تعریف می‌شود

$$\ell_{\mu}(M, L, R, X, Y, \mu) = \|M\|_* + \langle Y, Z - LMR \rangle + \frac{\mu}{2} \|Z - LMR\|_F^2, \quad (۳.۲)$$

$$\text{s.t. } P_{\Omega}(Z) = P_{\Omega}(W),$$

که Y ضریب لاگرانژ و μ ضریب جریمه می‌باشد. مراحل تکراری حل مسئله با استفاده از روش جهت متناوب ضرایب در [۶] به صورت زیر ارائه شده است

$$\begin{cases} M^{k+1} &= D_{\mu^{-1}} \left((L^k)^T (Z^k + Y^k/\mu) (R^k)^T \right), \\ Z^{k+1} &= L^k M^{k+1} R^k - Y^k/\mu + P_{\Omega} \left(W - L^k M^k R^k + Y^k/\mu \right), \\ L^{k+1} &= QR \left((Y^k/\mu + Z^k) (R^k)^T \right), \\ (R^{k+1})^T &= QR \left((Y^k/\mu + Z^k)^T L^{k+1} \right), \\ Y^{k+1} &= P_{\Omega} \left(Y^k + \mu (Z^{k+1} - L^{k+1} M^{k+1} R^{k+1}) \right), \end{cases} \quad (۴.۲)$$

۲.۲. روش خودضربی

با وجود کارایی ویژگی رتبه‌پایین بودن ماتریس برای بازیابی بسیاری از داده‌ها، داده‌های زیادی وجود دارند که ستون‌های آن‌ها متعلق به چندین زیرفضا با بعد پایین می‌باشند. در این صورت با افزایش تعداد زیرفضاها، ماتریس اصلی ممکن است رتبه‌بالا یا حتی رتبه کامل باشد و استفاده از ویژگی رتبه‌پایین بودن ماتریس مفهومی ندارد. برای حل این دسته از مسائل در منبع [۳] روش موسوم به روش خودضربی ارائه شد که علاوه بر بازیابی ماتریس‌های رتبه‌پایین برای بازیابی ماتریس‌های رتبه‌بالای تشکیل شده از چندین زیرفضای با بعد پایین، نیز مناسب است. ایده‌ی اصلی روش خودضربی این است که می‌توان برای هر ماتریس $X \in \mathbb{R}^{m \times n}$ با رتبه‌ی $r < \min(m, n)$ ، ماتریس ضرایب غیرهمانی $S \in \mathbb{R}^{n \times n}$ را به نحوی یافت که $X = XS$ [۳]. بنابراین اگر فرض کنیم که X تکمیل ماتریس ناقص W می‌باشد با قرار دادن $X = XS$ و مینیم کردن نرمی از ماتریس S به جای نرم ماتریس X می‌توان مسئله‌ی تکمیل ماتریس X

را به صورت زیر بیان کرد [۳]

$$\begin{aligned} & \min_{X,S} \|S\|_{\ell_S} \\ \text{s.t. } & X = XS, \\ & P_{\Omega}(X) = P_{\Omega}(W), \end{aligned} \quad (5.2)$$

یا

$$\begin{aligned} & \min_{X,S,E} \|S\|_{\ell_S} + \lambda \|E\|_F^2 \\ \text{s.t. } & X = XS + E, \\ & P_{\Omega}(X) = P_{\Omega}(W), \end{aligned} \quad (6.2)$$

که $\|\cdot\|_{\ell_S}$ نرم اعمال شده دلخواه روی ماتریس S بوده و λ ضریب جریمه‌ای است که از بزرگ شدن عبارت دوم جلوگیری می‌کند. با انتخاب مناسب λ و اعمال یک نرم محدب روی ماتریس S مسئله‌ی بالا یک جواب یکتا خواهد داشت. با اعمال سه نرم مرکزی، نرم ۲ و نرم ۱ روی ماتریس، مسئله‌ی بالا به مسائل خودضربی رتبه‌پایین، کمترین مربعات و تنک (SR-Sp) ^۸ منجر می‌شود و می‌توان ثابت کرد که جواب مسائل فوق معادل جواب مسئله‌ی تکمیل ماتریس می‌باشد. در این مقاله به اختصار مسئله‌ی موسوم به رتبه‌پایین

$$\begin{aligned} & \min_{X,S,E} \|S\|_* + \lambda \|E\|_F^2 \\ \text{s.t. } & X = XS + E, \\ & P_{\Omega}(X) = P_{\Omega}(W), \end{aligned} \quad (7.2)$$

⁸sparse self-representation

را بررسی می‌کنیم. حل مسئله‌ی (۷.۲) با استفاده از روش جهت متناوب ضرایب روی لاگرانژین افزوده‌ی مسئله‌ی فوق ارائه شده است [۳]. لاگرانژین افزوده‌ی مسئله‌ی فوق به صورت زیر ارائه می‌شود

(۸.۲)

$$\ell_{\mu}(X, S, E, Y, \mu) = \|S\|_* + \lambda \|E\|_F^2 + \langle Y, X - XS - E \rangle + \frac{\mu}{\gamma} \|X - XS - E\|_F^2,$$

$$\text{s.t. } P_{\Omega}(X) = P_{\Omega}(W),$$

که $Y \in \mathbb{R}^{m \times n}$ ماتریس ضرایب لاگرانژ، μ پارامتر جریمه و E ماتریس خطا می‌باشد. با حل مسئله‌ی مینیم‌سازی (۸.۲)، فرمول‌های تکراری زیر برای به‌روز رسانی متغیرها به دست می‌آید

$$\begin{cases} S^{k+1} &= D_{(\mu\eta)^{-1}}(S^k - \nabla_S^k / (\mu\eta)), \\ X^{k+1} &= X^k - t \nabla_X^k; \text{ s.t. } P_{\Omega}(X^{k+1}) = P_{\Omega}(M), \\ E^{k+1} &= \frac{\mu}{\gamma\lambda + \mu} (X^{k+1} - X^{k+1} S^{k+1} + Y^k / \mu), \\ Y^{k+1} &= Y^k + \mu (X^{k+1} - X^{k+1} S^{k+1} - E^{k+1}), \end{cases} \quad (9.2)$$

که ∇_S^k و ∇_X^k به ترتیب گرادیان ماتریس‌های S و X در S^k و X^k بوده و به صورت زیر تعریف می‌شوند

$$\begin{aligned} \nabla_S^k &= \mu (-X^k)^T (X^k - X^k S^k - E^k + Y^k / \mu), \\ \nabla_X^k &= (X^k - X^k S^{k+1} - E^k + Y^k / \mu) (I - S^{k+1})^T. \end{aligned} \quad (10.2)$$

روش‌های خودضربی روش‌هایی با سرعت متوسط بوده و فاقد عیوب روش‌های مبتنی بر رتبه و نرم مرکزی می‌باشند. بنابراین اگر بتوان سرعت اجرای این الگوریتم‌ها را افزایش داد و بر سرعت متوسط این روش‌ها غلبه کرد، می‌توان از قابلیت بالای بازیابی این روش‌ها استفاده کرد. در این مقاله چند راهکار برای افزایش سرعت روش‌های خودضربی ارائه شده است. این راهکارها از طریق خنثی کردن پیچیدگی محاسباتی تجزیه‌ی مقدار منفرد به‌کار رفته در هر مرحله‌ی الگوریتم حل روش خودضربی رتبه‌پایین و نیز ایده‌ی تقسیم مسئله‌ی تکمیل ماتریس به دو زیرمسئله، سرعت همگرایی روش‌های خودضربی را افزایش می‌دهند. یادآوری می‌کنیم که دسته‌ی دیگری از روش‌های ارائه شده برای حل مسئله‌ی تکمیل ماتریس،

روش‌های مبتنی بر مینیمم‌سازی نرم مرکزی می‌باشد که به علت محاسبه‌ی تجزیه‌ی مقدار منفرد در هر تکرار پیچیدگی محاسباتی بالایی دارند و برای مسائل با ابعاد بزرگ مناسب نیستند [۲، ۷].

۳. روشهای پیشنهادی

۱.۳. روش سه عاملی خودضربی رتبه‌پایین (SR-TriLR)

برای کاهش هزینه‌ی محاسباتی ناشی از تجزیه‌ی مقدار منفرد در روش خودضربی رتبه‌پایین قصد داریم که با استفاده از ایده‌ی روش تجزیه‌ی سه تایی سریع [۶]، ماتریس ضرایب S در روش خودضربی را با یک تقریب به صورت LML^T جایگزین کنیم که $L \in \mathbb{R}^{n \times r'}$ ، $M \in \mathbb{R}^{r' \times r'}$ ، $r' \ll \min(m, n)$ یک عدد صحیح دلخواه بوده و $L^T L = I$. این جایگزینی باعث می‌شود که تجزیه‌ی مقدار منفرد و محاسبه‌ی نرم مرکزی روی ماتریس M با ابعادی بسیار کوچک‌تر از ماتریس S انجام گیرد که منجر به کاهش پیچیدگی محاسباتی مسئله می‌شود. با اعمال این جایگزینی روی روش خودضربی رتبه‌پایین و استفاده از متغیر کمکی $Z = LML^T$ مسئله به صورت زیر خواهد بود

$$\begin{aligned} & \min_{X, Z, L, M, E} \|M\|_* + \lambda \|E\|_F^2 \\ \text{s.t. } & X = XZ + E, \\ & Z = LML^T, \\ & L^T L = I, \\ & P_\Omega(X) = P_\Omega(W). \end{aligned} \quad (1.3)$$

به دلیل سرعت پایین روش خودضربی رتبه‌پایین نسبت به دو روش دیگر خودضربی، این ایده را فقط روی روش رتبه‌پایین که به خاطر وجود محاسبه‌ی تجزیه‌ی مقدار منفرد در هر تکرار دارای پیچیدگی محاسباتی بیشتری می‌باشد پیاده می‌کنیم.

قضیه ۱.۳. جواب مسئله (۱.۳)، یک تقریب از جواب مسئله‌ی مینیمم‌سازی رتبه‌ی X ، (مسئله‌ی (۴.۱))، می‌باشد.

اثبات: فرض کنید $X = U\Sigma V^T$ تجزیه‌ی مقدار منفرد ماتریس X باشد که $U^T U = U U^T = I$ ، $V V^T = V^T V = I$ ، $\text{diag}(\Sigma) = (\sigma_1, \dots, \sigma_r)$ و r رتبه‌ی ماتریس X است. هدف ما مینیمم‌سازی تابع زیر می‌باشد

$$\|M\|_* + \frac{\lambda}{\nu} \|E\|_F^2 = \|M\|_* + \frac{\lambda}{\nu} \|XLML^T - X\|_F^2 \quad (۲.۳)$$

برای به‌دست آوردن مینیمم تابع فوق کافی است که صفر عضو مجموعه‌ی زیرگرادیان تابع هدف رابطه (۲.۳) باشد بنابراین زیرگرادیان تابع (۲.۳) را به‌دست آورده و قرار می‌دهیم

$$\begin{aligned} \circ & \in \partial \|M\|_* + \lambda (XL)^T (XLML^T - X) L \\ \circ & \in \frac{1}{\lambda} \partial \|M\|_* + L^T U \Sigma^2 U L (M - I) \end{aligned} \quad (۳.۳)$$

با استفاده از رابطه (۳.۳) ماتریس M به‌صورت زیر به دست می‌آید [۳]

$$M = \left(I_+ - \frac{1}{\lambda} L^T V_+ \Sigma_+^{-2} V_+^T L \right) = L^T V_+ \left(I_+ - \frac{1}{\lambda} \Sigma_+^{-2} \right) V_+^T L \quad (۴.۳)$$

و در نتیجه

$$\begin{aligned} \|M\|_* &= \left\| L^T V_+ \left(I_+ - \frac{1}{\lambda} \Sigma_+^{-2} \right) V_+^T L \right\|_* \\ &= \left\| I_+ - \frac{1}{\lambda} \Sigma_+^{-2} \right\|_* \\ &= \sum_{\sigma_+} \left(1 - \frac{1}{\lambda} \sigma_+^{-2} \right) \end{aligned} \quad (۵.۳)$$

که Σ_+ ماتریس قطری مقادیر منفرد ماتریس X که قطر آن بزرگ‌تر از $1/\sqrt{\lambda}$ ، σ_+ مقادیر منفردی از ماتریس X که مقدار آنها بزرگ‌تر از $1/\sqrt{\lambda}$ (اعضای روی قطری اصلی Σ_+) و ماتریس V_+ ماتریس بردارهای منفرد مرتبط با Σ_+ و I_+ یک ماتریس همانی می‌باشد که $\text{size}(I_+) = \text{size}(\Sigma_+)$. در رابطه‌ی (۵.۳)، $\|M\|_*$ را به‌صورت تابعی از بزرگ‌ترین مقادیر منفرد ماتریس X ، جواب مسئله‌ی

(۵.۱) به دست آوردیم. با توجه به رابطه‌ی (۵.۳) با کاهش بزرگ‌ترین مقادیر منفرد ماتریس X مقدار نرم مرکزی ماتریس M کاهش می‌یابد، بنابراین با نادیده گرفتن عبارت $\|E\|_F^2$ مینیمم‌سازی تابع هدف مسئله‌ی (۱.۳) با مینیمم‌سازی مسئله‌ی (۵.۱) معادل می‌باشد، در نتیجه اگر ماتریس X جواب مسئله‌ی (۱.۳) باشد، در این صورت یک تقریب از جواب مسئله‌ی (۵.۱) و لذا تقریبی از جواب مسئله (۴.۱) می‌باشد. \square

لاگرانژین افزوده‌ی مسئله‌ی (۱.۳) به صورت زیر تعریف می‌شود

$$\begin{aligned} \ell_\mu(X, Z, L, M, E, Y_1, Y_2, \mu) = & \|M\|_* + \lambda \|E\|_F \\ & + \langle Y_1, X - XZ - E \rangle + \frac{\mu}{\gamma} \|X - XZ - E\|_F^2 \\ & + \langle Y_2, Z - LML^T \rangle + \frac{\mu}{\gamma} \|Z - LML^T\|_F^2 \end{aligned} \quad (۶.۳)$$

$$\text{s.t. } L^T L = I,$$

$$P_\Omega(X) = P_\Omega(W),$$

به طوری که $Y_2 \in \mathbb{R}^{n \times n}$, $Y_1 \in \mathbb{R}^{m \times n}$ ضرایب لاگرانژ و μ ضریب جریمه می‌باشد. روش جهت متناوب ضرایب با در نظر گرفتن (۶.۳)، مسئله‌ی (۱.۳) را به صورت تکراری زیر حل می‌کند

$$M^{k+1} = \arg \min_M \|M\|_* + \frac{\mu}{\gamma} \|Z^k - L^k M (L^k)^T + Y_2^k / \mu\|_F^2. \quad (۷.۳)$$

در زیر مسئله‌ی مینیمم‌سازی (۷.۳)، می‌توان با استفاده از عملگر کاهش مقدار منفرد جواب ماتریس M را به صورت زیر ارائه داد

$$M^{k+1} = D_{\mu^{-1}}(Z^k + Y_2^k / \mu). \quad (۸.۳)$$

با حل زیرمسئله‌ی زیر نیز جواب بهین ماتریس Z^{k+1} در مسئله‌ی (۶.۳) به دست می‌آید

$$Z^{k+1} = \arg \min_Z \frac{1}{\gamma} \|X^k - X^k Z - E^k + Y_{\gamma}^k / \mu\|_F^2 + \frac{1}{\gamma} \|Z - L^k M^{k+1} (L^k)^T + Y_{\gamma}^k / \mu\|_F^2, \quad (9.3)$$

و با استفاده از روش گرادیان می‌توان به جواب صریح دست یافت

$$Z^{k+1} = Z^k - t_1 \nabla_Z^k, \quad (10.3)$$

که ∇_Z^k گرادیان Z در Z^k بوده و به شکل زیر به دست می‌آید

$$\nabla_Z^k = (Z^k - L^k M^{k+1} (L^k)^T + Y_{\gamma}^k / \mu) - (X^k)^T (X^k - X^k Z^k - E^k + Y_{\gamma}^k / \mu), \quad (11.3)$$

و $t_1 < 1 / (\|X^k\|_F^2 + 2)$ نیز طول گام می‌باشد. فرمول بندی مسئله‌ی بهینه‌سازی برای یافتن جواب X نیز به صورت زیر می‌باشد

$$X^{k+1} = \arg \min_X \frac{\mu}{\gamma} \|X - X Z^{k+1} - E^k + Y_{\gamma}^k / \mu\|_F^2, \quad (12.3)$$

و با اعمال روش گرادیان بر مسئله‌ی (۱۲.۳) داریم

$$X^{k+1} = X^k - t_2 \nabla_X^k; \\ \text{s.t. } P_{\Omega}(X^{k+1}) = P_{\Omega}(M), \quad (13.3)$$

که $t_2 < 1 / \|I - Z^{k+1}\|_F^2$ طول گام بوده و

$$\nabla_X^k = (X^k - X^k Z^{k+1} - E^k + Y_{\gamma}^k / \mu) (I - Z^{k+1})^T, \quad (14.3)$$

گرادیان X در X^k می‌باشد. برای یافتن ماتریس L در مسئله (۶.۳)، مسئله‌ی بهینه‌سازی به صورت زیر ارائه می‌شود

$$L^{k+1} = \arg \min_L \|Z^{k+1} - LM^k(L^k)^T + Y_2/\mu\|_F^2, \quad (15.3)$$

و با استفاده از فرمول زیر می‌توان یک جواب صریح برای ماتریس L ارائه داد

$$\begin{aligned} LM^{k+1}(L^k)^T &= Z^{k+1} + Y_2^k/\mu, \\ LM^{k+1} &= (Z^{k+1} + Y_2^k/\mu)L^k. \end{aligned} \quad (16.3)$$

اگر تعریف کنیم $P = Z + Y_2/\mu$ ، داریم

$$L^{k+1} = QR(L^k M^{k+1}) = QR(P^k L^k), \quad (17.3)$$

با استفاده از جواب‌های به دست آمده تا این مرحله ماتریس E را می‌توان به صورت زیر به روز رسانی کرد

$$E^{k+1} = \arg \min_E \lambda \|E\|_F^2 + \frac{\mu}{2} \|X^{k+1} - X^{k+1}Z^{k+1} - E + Y_1^k/\mu\|_F^2, \quad (18.3)$$

و با صفر قرار دادن گرادیان، ماتریس خطای E را مستقیم و به صورت زیر محاسبه کرد

$$E^{k+1} = \frac{\mu}{2\lambda + \mu} (X^{k+1} - X^{k+1}Z^{k+1} + Y_1^k/\mu). \quad (19.3)$$

دو ضریب لاگرانژ Y_1 و Y_2 نیز از رابطه‌ی زیر به دست می‌آیند

$$\begin{aligned} Y_1^{k+1} &= Y_1^k + \mu (X^{k+1} - X^{k+1}Z^{k+1} - E^{k+1}), \\ Y_2^{k+1} &= Y_2^k + \mu (Z^{k+1} - L^{k+1}M^{k+1}(L^{k+1})^T). \end{aligned} \quad (20.3)$$

با توجه به محاسبات انجام شده، الگوریتم حل مسئله‌ی مینیمم‌سازی (۶.۳) به صورت زیر ارائه می‌شود

Algorithm 1: Low Rank Tri Factorization Self Representation method (TriSR-LR)

Input: $\Omega, P_{\Omega}(W), \mu, \mu_{max}, \rho, \text{max-iter}, k = 0, \varepsilon, \lambda.$

Initialization: $M = 0, L = I, X = P_{\Omega}(W), Z = 0, E = 0, Y_1 = 0, Y_2 = 0.$

Output: $X = X^{(k)}, Z = Z^{(k)}, E = E^{(k)}.$

while not converged and $k < \text{max-iter}$ do

- Fix the others and update M by Eq. (3.8).
- Fix the others and update Z by Eq. (3.10).
- Fix the others and update X by Eq. (3.13).
- Fix the others and update L by Eq. (3.17).
- Fix the others and update E by Eq. (3.19).
- Update the multipliers Y_1 and Y_2 by Eq. (3.20).
- Check the convergence condition

$$c_1 = \left\| X^{(k+1)} - X^{(k+1)}Z^{(k+1)} - E^{(k+1)} \right\|_F^2 / \|M\|_F^2$$

$$c_2 = \max \left\{ \left\| Z^{(k+1)} - Z^{(k)} \right\|_F^2, \left\| X^{(k+1)} - X^{(k)} \right\|_F^2, \left\| E^{(k+1)} - E^{(k)} \right\|_F^2 \right\} / \|M\|_F^2$$

- **if** $c_1 < \varepsilon$ and $c_2 < \varepsilon$, converged;
- **else** $\mu = \min(\rho\mu, \mu_{max})$ and $k = k + 1.$

end while

ایده‌ی قضیه‌ی زیر از [۸] گرفته شده است.

قضیه ۲.۳. فرض کنید $\{T_j\}_{j=1}^{\infty}$ دنباله‌ی تولید شده از الگوریتم ۱ باشد که

$$T_j = (X_j, Z_j, L_j, M_j, E_j, Y_{\setminus j}, Y_{\setminus j})$$

اگر دنباله‌ی $\{T_j\}_{j=1}^{\infty}$ کراندار بوده و $\lim_{j \rightarrow \infty} (T^{j+1} - T^j) = 0$ ، آن‌گاه هر نقطه‌ی حدی دنباله‌ی $\{T_j\}_{j=1}^{\infty}$ به صورت

$$T^* = (X^*, Z^*, L^*, M^*, E^*, Y_{\setminus}^*, Y_{\setminus}^*)$$

در شرایط *KKT* زیر

$$\nabla \ell_{X^*} = Y_{\setminus}^* (I - Z^*)^T = 0$$

$$\nabla \ell_{Z^*} = -(X^*)^T Y_{\setminus}^* + Y_{\setminus}^* = 0$$

$$\nabla \ell_{L^*} = -Y_{\setminus}^* L^* (M^*)^T = 0 \quad (21.3)$$

$$\nabla \ell_{E^*} = \lambda E^* - Y_{\setminus}^* = 0$$

$$0 \in \partial \ell_{M^*} = \partial \|M^*\|_* + (L^*)^T Y_{\setminus}^* L^*$$

و همچنین شدنی بودن مسئله که معادل است با

$$X^* - X^* Z^* - E^* = 0$$

$$Z^* - L^* M^* (L^*)^T = 0 \quad (22.3)$$

صدق می‌کند و بنابراین اگر دنباله‌ی فوق همگرا باشد، به نقطه‌ی *KKT* مسئله همگرا خواهد بود.

اثبات: با استفاده از روابط به دست آمده در الگوریتم ۱ داریم

$$\begin{aligned}
 Y_{\checkmark}^{k+1} - Y_{\checkmark}^k &= \mu (X^{k+1} - X^{k+1} Z^{k+1} - E^{k+1}), \\
 Y_{\checkmark}^{k+1} - Y_{\checkmark}^k &= \mu (Z^{k+1} - L^{k+1} M^{k+1} (L^{k+1})^T), \\
 &\circ \in \frac{1}{\mu} \partial \|M^{k+1}\|_* + M^{k+1} - (L^k)^T (Z^k + Y_{\checkmark}^k / \mu) L^k, \\
 Z^{k+1} - Z^k &= -t_{\checkmark} (Z^k - L^k M^{k+1} (L^k)^T + Y_{\checkmark}^k / \mu) \\
 &\quad + t_{\checkmark} (X^k)^T (X^k - X^k Z^k - E^k + Y_{\checkmark}^k / \mu), \\
 X^{k+1} - X^k &= t_{\checkmark} (X^k - X^k Z^{k+1} - E^k + Y_{\checkmark}^k / \mu) (I - Z^{k+1})^T, \\
 L^{k+1} - L^k &= (Z^{k+1} + Y_{\checkmark}^k / \mu) L^k (M^{k+1})^{-1} - L^k, \\
 E^{k+1} - E^k &= \frac{\mu}{\checkmark \lambda + \mu} (X^{k+1} - X^{k+1} Z^{k+1} + Y_{\checkmark}^k / \mu) - E^k.
 \end{aligned} \tag{۲۳.۳}$$

از آنجایی که $\lim_{j \rightarrow \infty} (T^{j+1} - T^j) = \circ$ با توجه به رابطه‌ی (۲۳.۳) داریم

$$\begin{aligned}
 X^j - X^j Z^j - E^j &\longrightarrow \circ, \\
 Z^j - L^j M^j (L^j)^T &\longrightarrow \circ, \\
 Y_{\checkmark}^j (I - Z^j)^T &\longrightarrow \circ, \\
 - (X^j)^T Y_{\checkmark}^j + Y_{\checkmark}^j &\longrightarrow \circ, \\
 - Y_{\checkmark}^j L^j (M^*)^T &\longrightarrow \circ, \\
 \checkmark \lambda E^j - Y_{\checkmark}^j &\longrightarrow \circ, \\
 \partial \|M^j\|_* + (L^j)^T Y_{\checkmark}^j L^j &\longrightarrow \circ,
 \end{aligned} \tag{۲۴.۳}$$

و بدین ترتیب نتیجه مورد نظر حاصل می‌شود. \square

۲.۳. تقسیم ماتریس به دو زیرماتریس

نتایج عددی حاصل از اجرای روش‌های حل مسئله‌ی تکمیل ماتریس نشان‌دهنده‌ی این امر است که با بزرگ‌شدن ابعاد مسئله، هزینه‌ی محاسباتی و زمان لازم برای حل مسئله بالا می‌رود. یک ایده برای کاهش زمان حل مسئله، تقسیم ماتریس به چند زیرماتریس می‌باشد. یک ماتریس را می‌توان به صورت‌های مختلفی به چند زیرماتریس تبدیل کرد. اما نکته‌ای که باید در تعیین زیرماتریس‌ها در نظر داشت و نتایج عددی ناموفق به دست آمده توسط نگارنده آن را تایید می‌کنند، حساسیت و وابستگی جواب مسئله به ستون‌های ماتریس می‌باشد زیرا ایده‌ی حل مسئله‌ی تکمیل ماتریس رتبه‌پایین، بر اساس وابستگی بین ستون‌های ماتریس ارائه شده است. بنابراین باید زیرماتریس‌ها به گونه‌ای تعیین شود که ستون‌های ماتریس دست نخورده بماند و تعیین زیرماتریس باید بر اساس تقسیم ستون‌ها به چند زیرماتریس باشد. اما برای تقسیم ستون‌ها می‌توان راهکارهای زیر را ارائه داد

- تقسیم تصادفی ستون‌ها در دو دسته‌ی مساوی در دو زیرماتریس و بازیابی جداگانه‌ی زیرماتریس‌ها و قرار دادن در ماتریس اصلی و به دست آوردن خطا
- بازیابی یک زیرماتریس با نصف ستون‌های ماتریس که به تصادف انتخاب شده‌اند و قرار دادن جواب به دست آمده در ماتریس اصلی و سپس بازیابی کل ماتریس و به دست آوردن خطا
- تقسیم ماتریس به دو بلوک با ابعاد مساوی، بازیابی جداگانه‌ی بلوک‌ها و قرار دادن در ماتریس اصلی و به دست آوردن خطا
- بازیابی یک بلوک با نصف ستون‌های ماتریس و قرار دادن جواب به دست آمده در ماتریس اصلی و سپس بازیابی ماتریس و به دست آوردن خطا

نتایج عددی به دست آمده کارایی روش اول و سوم را تایید می‌کند البته ما در بررسی روش اول ستون‌های ماتریس را به صورت یک در میان در دو زیرماتریس قرار دادیم. نتایج عددی به دست آمده در روش دوم و چهارم نشان‌گر کاهش دقت، کاهش سرعت و عدم کارایی این دو راهکار می‌باشد.

۴. نتایج عددی

برای بررسی نتایج روش‌های تکمیل ماتریس آن‌ها را روی داده‌های مصنوعی (ماتریس تصادفی) $W \in \mathbb{R}^{m \times n}$ و داده‌های واقعی (تصویر) پیاده کردیم. داده‌های مصنوعی شامل ماتریس‌های تصادفی

δ	$r' = ۴۵$			$r' = ۴۹$			$r' = ۵۰$		
	e	k	T(s)	e	k	T(s)	e	k	T(s)
۰/۳	۳/۹e-۱	۲۰۰	۴	۱/۷e-۱	۲۰۰	۵	۱/۲e-۴	۹۵	۲
۰/۴	۳/۵e-۱	۲۰۰	۴	۱/۵e-۱	۲۰۰	۵	۱/۲e-۴	۸۰	۲
۰/۵	۳/۴e-۱	۲۰۰	۴	۱/۴e-۱	۲۰۰	۴	۱/۱e-۴	۶۸	۲
۰/۶	۲/۹e-۱	۲۰۰	۴	۵/۴e-۳	۲۰۰	۵	۲/۳e-۴	۵۷	۲
۰/۷	۹/۵e-۱	۲۰۰	۴	۲/۹e-۲	۲۰۰	۵	۲/۸e-۴	۴۶	۱
۰/۸	۹/۱e-۱	۲۰۰	۴	۷/۹e-۲	۲۰۰	۵	۳/۱e-۴	۲۰۰	۵

جدول ۱: نتایج عددی حاصل از پیاده‌سازی روش تجزیه‌ی سه‌تایی سریع روی ماتریس تصادفی با بعد ۵۰۰ و $r = ۵۰$ با $r' \leq r$

متعلق به یک زیرفضای رتبه‌پایین و ماتریس‌های تصادفی متعلق به چندین زیرفضای رتبه‌پایین (حاوی چندین زیرماتریس رتبه‌پایین) می‌باشد. داده‌های واقعی نیز شامل بازیابی تصاویر می‌باشد. در همه‌ی نتایج، r رتبه‌ی ماتریس بوده و $\delta = N/(mn)$ که N تعداد درایه‌های معلوم ماتریس و mn تعداد کل درایه‌های ماتریس می‌باشد. همچنین e خطای نسبی محاسبه شده را نشان می‌دهد و مقدار آن برابر با $\frac{P_{\Omega^c}(W-X)}{P_{\Omega^c}(W)}$ می‌باشد که Ω^c مجموعه‌ی اندیس‌های درایه‌های نامعلوم ماتریس، X تکمیل ماتریس W ، k تعداد تکرارهای هر روش تا رسیدن به همگرایی، k_{max} حداکثر تعداد تکرار برای هر روش (در همه‌ی نتایج k_{max} را برابر با ۲۰۰ در نظر گرفته‌ایم) و T زمان صرف شده برای محاسبات هر روش می‌باشد. در همه‌ی نتایج، ماتریس تصادفی $W \in \mathbb{R}^{m \times n}$ با رتبه‌ی دلخواه r و ستون‌های متعلق به یک زیرفضا از ضرب دو ماتریس تصادفی $A \in \mathbb{R}^{m \times r}$ و $B \in \mathbb{R}^{r \times n}$ تولید شده است. همچنین ماتریس تصادفی $W \in \mathbb{R}^{m \times n}$ با رتبه‌ی دلخواه r و ستون‌های متعلق به l زیرفضا با استفاده از ماتریس‌های تصادفی $A_i \in \mathbb{R}^{m \times r_i}$ و $B_i \in \mathbb{R}^{r_i \times n_i}$ که $r_1 + r_2 + \dots + r_l = r$ و $n_1 + n_2 + \dots + n_l = n$ به صورت $W = [A_1 B_1, A_2 B_2, \dots, A_l B_l]$ تولید شده است.

۱.۴. نتایج عددی حاصل از پیاده‌سازی روش‌های مبتنی بر رتبه و روش‌های خودضربی

۱.۱.۴. ماتریس‌های تصادفی با ستون‌های متعلق به یک زیرفضا

در این بخش برای بررسی تأثیر انتخاب رتبه روی نتایج روش‌های مبتنی بر رتبه، سه روش تجزیه‌ی سه‌تایی سریع [۶]، مقیاس شده‌ی سریع‌ترین جهت کاهش متناوب (SASD) [۹] و تجزیه‌ی قوی

⁹scaled alternating steepest descent

δ	$r' = 51$			$r' = 55$			$r' = 60$		
	e	k	T(s)	e	k	T(s)	e	k	T(s)
۰/۳	$2/2e-4$	۹۷	۳	$4/7e-3$	۱۰۹	۳	$1/4e-2$	۲۰۰	۶
۰/۴	$1/6e-4$	۷۹	۲	$1/7e-4$	۷۹	۲	$2/6e-4$	۲۰۰	۶
۰/۵	$1/6e-4$	۶۲	۲	$1/6e-4$	۶۲	۲	$1/7e-4$	۶۱	۲
۰/۶	$2/4e-4$	۵۶	۲	$2/4e-4$	۵۶	۲	$2/4e-4$	۵۶	۲
۰/۷	$2/8e-4$	۴۶	۱	$2/8e-4$	۴۶	۲	$2/8e-4$	۴۶	۲
۰/۸	$3/1e-4$	۲۰۰	۵	$3/1e-4$	۲۰۰	۵	$3/3e-4$	۲۰۰	۶

جدول ۲: نتایج عددی حاصل از پیاده‌سازی روش تجزیه‌ی سه‌تایی سریع روی ماتریس تصادفی با بعد 500 و $r = 50$ با $r' > r$

δ	$r' = 45$			$r' = 49$			$r' = 50$		
	e	k	T(s)	e	k	T(s)	e	k	T(s)
۰/۳	$3/8e-1$	۲۰۰	۴	$1/7e-1$	۲۰۰	۴	$6/8e-5$	۱۰۸	۳
۰/۴	$3/4e-1$	۲۰۰	۴	$1/5e-1$	۱۸۷	۴	$2/5e-5$	۴۵	۱
۰/۵	$3/3e-1$	۲۰۰	۳	$1/4e-1$	۲۰۰	۳	$1/9e-5$	۲۸	۱
۰/۶	$3/5e-1$	۱۹۷	۳	$1/5e-1$	۲۰۰	۳	$1/4e-5$	۲۰	۱
۰/۷	$4/0e-1$	۱۷۱	۳	$1/7e-1$	۲۰۰	۳	$6/9e-6$	۱۶	۰/۶۵
۰/۸	$4/7e-1$	۱۸۳	۳	$2/0e-1$	۲۰۰	۳	$5/3e-6$	۱۳	۰/۴۱

جدول ۳: نتایج عددی حاصل از پیاده‌سازی روش مقیاس شده‌ی سریع‌ترین جهت کاهش متناوب روی ماتریس تصادفی با بعد 500 و $r = 50$ با $r' \leq r$

δ	$r' = 51$			$r' = 55$			$r' = 60$		
	e	k	T(s)	e	k	T(s)	e	k	T(s)
۰/۳	$3/2e-3$	۱۵۶	۳	$1/4e-2$	۲۰۰	۴	$3/2e-2$	۲۰۰	۴
۰/۴	$4/6e-3$	۵۵	۱	$1/2e-2$	۱۱۰	۲	$1/7e-2$	۲۰۰	۴
۰/۵	$2/9e-3$	۴۳	۱	$8/1e-3$	۵۶	۱	$1/7e-2$	۷۵	۱
۰/۶	$4/5e-4$	۲۵	۱	$3/4e-3$	۳۵	۱	$1/3e-2$	۶۱	۱
۰/۷	$4/5e-3$	۱۶	۰/۵۳	$8/9e-3$	۲۰	۱	$1/4e-2$	۳۳	۱
۰/۸	$1/8e-4$	۱۳	۰/۴۵	$4/2e-3$	۱۸	۰/۴۷	$1/1e-2$	۲۶	۱

جدول ۴: نتایج عددی حاصل از پیاده‌سازی روش مقیاس شده‌ی سریع‌ترین جهت کاهش متناوب روی ماتریس تصادفی با بعد 500 و $r = 50$ با $r' > r$

δ	$r' = ۴۵$			$r' = ۴۹$			$r' = ۵۰$		
	e	k	T(s)	e	k	T(s)	e	k	T(s)
۰/۳	۳/۷e-۱	۲۰۰	۴	۱/۶e-۱	۲۰۰	۴	۷/۰e-۳	۲۰۰	۴
۰/۴	۳/۱e-۱	۲۰۰	۴	۱/۳e-۱	۲۰۰	۳	۱/۴e-۴	۱۲۹	۱
۰/۵	۲/۸e-۱	۲۰۰	۳	۱/۲e-۱	۲۰۰	۳	۷/۵e-۵	۷۰	۲
۰/۶	۲/۷e-۱	۲۰۰	۳	۱/۲e-۱	۲۰۰	۳	۵/۲e-۵	۴۴	۲
۰/۷	۲/۶e-۱	۲۰۰	۳	۱/۱e-۱	۲۰۰	۳	۳/۳e-۵	۳۰	۱
۰/۸	۲/۵e-۱	۲۰۰	۳	۱/۱e-۱	۲۰۰	۳	۲/۹e-۵	۲۱	۱

جدول ۵: نتایج عددی حاصل از پیاده‌سازی روش تجزیه‌ی قوی روی ماتریس تصادفی با بعد ۵۰۰ و $r = ۵۰$ با $r' \leq r$

δ	$r' = ۵۱$			$r' = ۵۵$			$r' = ۶۰$		
	e	k	T(s)	e	k	T(s)	e	k	T(s)
۰/۳	۹/۸e-۳	۲۰۰	۳	۳/۱e-۲	۲۰۰	۴	۸/۳e-۲	۲۰۰	۴
۰/۴	۷/۲e-۴	۱۴۲	۱	۱/۹e-۲	۲۰۰	۲	۲/۵e-۲	۲۰۰	۴
۰/۵	۹/۱e-۳	۹۰	۱	۱/۵e-۲	۱۲۰	۱	۳/۳e-۲	۱۵۸	۱
۰/۶	۲/۵e-۳	۵۷	۱	۱/۰e-۲	۸۵	۱	۳/۵e-۲	۱۷۰	۱
۰/۷	۱/۶e-۲	۳۰	۱	۲/۲e-۲	۴۵	۲	۳/۶e-۲	۶۵	۱
۰/۸	۸/۴e-۴	۲۴	۰/۵۳	۱/۳e-۲	۴۴	۱	۳/۱e-۲	۵۰	۱

جدول ۶: نتایج عددی حاصل از پیاده‌سازی روش تجزیه‌ی قوی روی ماتریس تصادفی با بعد ۵۰۰ و $r = ۵۰$ با $r' > r$

δ	SR-Sp			SR-LS			SR-LR		
	e	k	T(s)	e	k	T(s)	e	k	T(s)
۰/۳	۹/۴e-۳	۷۸	۱۳	۹/۶e-۳	۷۷	۱۳	۱/۴e-۲	۷۳	۲۳
۰/۴	۶/۸e-۳	۶۴	۱۱	۷/۰e-۳	۶۳	۱۱	۸/۵e-۳	۶۰	۱۸
۰/۵	۵/۹e-۳	۵۲	۹	۶/۰e-۳	۵۱	۹	۷/۴e-۳	۴۹	۱۶
۰/۶	۵/۳e-۳	۴۲	۸	۵/۳e-۳	۴۱	۷	۶/۱e-۳	۴۰	۱۲
۰/۷	۴/۰e-۳	۳۵	۶	۴/۱e-۳	۳۴	۶	۴/۸e-۳	۳۳	۱۰
۰/۸	۲/۲e-۳	۳۲	۶	۲/۳e-۳	۳۱	۵	۲/۱e-۳	۳۱	۹

جدول ۷: نتایج عددی حاصل از پیاده‌سازی روش‌های خودضربی تنک، کمترین مربعات و رتبه‌پایین روی ماتریس تصادفی با بعد ۵۰۰ و $r = ۵۰$

(PF) ^{۱۰} [۹] را روی ماتریس تصادفی با ستون‌های متعلق به یک زیرفضا با ابعاد $m = n = 500$ با $r = 50$ و $\delta = 0.3, 0.4, \dots, 0.8$ روی مجموعه‌ی تصادفی و یکنواخت Ω با آستانه‌ی خطای 10^{-5} به ازای $r' = 45, 49, 50, 51, 55, 60$ پیاده کردیم و نتایج را با نتایج حاصل از روش‌های خودضربی مقایسه کردیم. جداول ۱-۷ نتایج به دست آمده از اجرای سه روش مبتنی بر رتبه و جدول ۸ نتایج به دست آمده از روش‌های خودضربی می‌باشد. نتایج عددی جداول ۱-۷ نشان می‌دهد که به ازای هر $r' < r$ خطای بازیابی، بسیار زیاد می‌باشد و تنها برای $r' \geq r$ می‌توان به خطای قابل قبولی دست یافت که با مینیم کردن رتبه در تناقض می‌باشد و اساس و فرض مورد استفاده برای حل مسئله زیر سؤال می‌رود زیرا با این شرایط نمی‌توان به جوابی با کمترین رتبه‌ی ممکن رسید در صورتی که مسئله‌ی ما یافتن یک ماتریس رتبه‌پایین می‌باشد. از طرف دیگر الگوریتم مناسبی هم برای یافتن رتبه‌ی اصلی ماتریس وجود ندارد و الگوریتم‌های مورد استفاده برای به روز رسانی رتبه بسیار ضعیفند، همچنین در مسائل واقعی به رتبه‌ی اصلی ماتریس دسترسی نداریم. پس با همه‌ی این دلایل روش‌های مبتنی بر رتبه روش‌های مناسبی برای حل مسئله‌ی تکمیل ماتریس نمی‌باشند. اما بررسی نتایج جدول ۷ نشان می‌دهد که با حل مسئله با روش‌های خودضربی به خطای قابل قبولی کمتر از خطای ذکر شده در ستون‌های اول و دوم جداول ۱، ۳، و ۵ دست یافته‌ایم. پس تا اینجا روش‌های خودضربی از لحاظ میزان خطا و کارایی نسبت به روش‌های مبتنی بر رتبه برتری دارد و تنها اشکال آن‌ها پایین بودن سرعتشان می‌باشد.

۲.۱.۴. پیاده‌سازی روی تصویر

در این بخش نیز نتایج پیاده‌سازی روش‌های مبتنی بر رتبه را روی تصویر طبیعت که در شکل ۱ نمایش داده شده با نرخ نمونه‌ی $\delta = 0.1, 0.2, \dots, 0.8$ اجرا کرده و نتایج عددی را در جداول ۹ و ۱۰ نمایش دادیم. همچنین تصاویر بازیابی شده‌ی قسمت‌های (ب) و (ج) شکل ۱ با $\delta = 0.3$ و $\delta = 0.5$ با استفاده از روش‌های مبتنی بر رتبه و روش‌های خودضربی در شکل‌های ۵-۲ نمایش داده شده است. مقایسه‌ی نتایج به دست آمده در جداول ۹ و ۱۰ نشان می‌دهد که دقت روش‌های خودضربی مخصوصاً دو روش SR-LR و SR-LS بیش از سه روش معرفی شده‌ی مبتنی بر رتبه و تجزیه می‌باشد و مشاهده‌ی تصاویر بازیابی شده نیز مؤید دقت پایین روش‌های مبتنی بر رتبه در مقایسه با تصاویر بازیابی شده از روش‌های خودضربی می‌باشد.

¹⁰power factorization

از آنجایی که تصویر می‌تواند ارائه دهنده‌ی یک مثال کوچک از مسائل واقعی تکمیل ماتریس باشد و با کنار هم قرار دادن نتایج به‌دست آمده از بازیابی ماتریس‌های تصادفی با رتبه‌ای کوچک‌تر از رتبه‌ی واقعی و نتایج به‌دست آمده از بازیابی تصویر با استفاده از روش‌های مبتنی بر رتبه و توجه به این مسئله که تاکنون الگوریتم کارآمدی برای به‌دست آوردن رتبه ارائه نشده ضرورت ارائه‌ی روش‌های مطمئن‌تر و دقیق‌تر افزایش می‌یابد. همان‌گونه که نتایج عددی نیز نشان می‌دهد عیب سه روش خودضربی سرعت پایین آن‌هاست که این عیب با راهکارهایی که در بخش‌های قبل برای افزایش سرعت روش‌های خودضربی ارائه دادیم از بین رفته است.

۲.۴. پیاده‌سازی روش سه‌عاملی خودضربی رتبه‌پایین روی ماتریس‌های تصادفی

برای بررسی کارایی روش سه‌عاملی خودضربی رتبه‌پایین، این روش و روش خودضربی رتبه‌پایین را روی ماتریس‌های متعلق به یک زیرفضا با ابعاد $m = n = ۱۰۰۰, ۲۰۰۰, ۳۰۰۰$ با رتبه‌ی $r = ۱۰, ۵۰, ۱۰۰$ و روی ماتریس‌های متعلق به چندین زیرفضا با $l = ۵$ و ابعاد $m = n = ۱۰۰۰, ۲۰۰۰$ و $r = ۵۲, ۲۰۰$ با نرخ نمونه‌ی $\delta = ۰/۳, ۰/۴, \dots, ۰/۶$ پیاده کردیم. جدول ۱۰ نتایج عددی حاصل از پیاده‌سازی دو روش خودضربی رتبه‌پایین و روش سه‌عاملی خودضربی رتبه‌پایین روی ماتریس‌های تصادفی متعلق به یک زیرفضا و جدول ۱۱ نتایج عددی حاصل از پیاده‌سازی روش خودضربی رتبه‌پایین و روش سه‌عاملی خودضربی رتبه‌پایین روی ماتریس‌های متعلق به چندین زیرفضا می‌باشد. بررسی نتایج عددی نشان می‌دهد که دو روش خودضربی رتبه‌پایین و سه‌عاملی خودضربی رتبه‌پایین دقیقاً با یکدیگر معادل بوده و به‌طور تقریبی خطا و تعداد تکرارهای هر دو روش در همه‌ی نتایج به‌دست آمده بر یکدیگر منطبق می‌باشند. ولی از لحاظ سرعت، سرعت روش سه‌عاملی خودضربی رتبه‌پایین تقریباً $۱/۷$ برابر روش خودضربی رتبه‌پایین می‌باشد.

۳.۴. پیاده‌سازی ایده‌ی تبدیل مسئله‌ی تکمیل ماتریس به چندین زیرمسئله

برای بررسی کارایی دو روش تبدیل ماتریس به دو زیر ماتریس (SMC)^{۱۱} و تبدیل ماتریس به دو بلوک (BMC)^{۱۲} این روش‌ها را تا رسیدن به خطای نسبی معین $e = ۱۰^{-۳}$ اجرا کرده و نتایج را با نتایج به‌دست آمده از مسئله‌ی تکمیل ماتریس معمولی (MC)^{۱۳} مقایسه می‌کنیم و هر سه ایده را را روی

^{۱۱}sub matrix completion

^{۱۲}block matrix completion

^{۱۳}matrix completion

δ	PF				SASD				FTF			
	r	e	k	T(s)	r	e	k	T(s)	r	e	k	T(s)
۰/۱	۲	۳/۱e-۱	۲۰۰	۰/۳۱	۲	۲/۸e-۱	۲۰۰	۰/۶۴	۲	۲/۳e-۱	۲۰۰	۰/۱۷
۰/۲	۲	۱/۷e-۱	۲۰۰	۰/۱۷	۲	۱/۸e-۱	۲۰۰	۰/۳۳	۲۰	۱/۶e-۱	۸۵	۰/۲۱
۰/۳	۲	۱/۶e-۱	۱۳۷	۰/۱۶	۲	۱/۹e-۱	۳۵	۰/۰۶	۲۰	۱/۴e-۱	۲۰۰	۰/۲۶
۰/۴	۵	۱/۴e-۱	۲۰۰	۰/۱۴	۵	۱/۶e-۱	۲۰۰	۰/۶۳	۲۰	۱/۴e-۱	۲۰۰	۰/۲۷
۰/۵	۵	۱/۲e-۱	۲۰۰	۰/۱۲	۵	۱/۵e-۱	۱۴۲	۰/۱۸	۲۰	۱/۵e-۱	۲۰۰	۰/۲۸
۰/۶	۵	۱/۲e-۱	۱۸۵	۰/۱۲	۵	۱/۷e-۱	۷۰	۰/۰۷	۲	۲/۰e-۱	۲۰۰	۱
۰/۷	۵	۱/۲e-۱	۲۰۰	۰/۱۲	۱۰	۱/۳e-۱	۱۲۹	۰/۳۹	۲	۲/۰e-۱	۲۰۰	۰/۰۸
۰/۸	۱۰	۹/۰e-۲	۸۱	۰/۰۸	۱۵	۱/۳e-۱	۱۵۳	۰/۳۶	۲	۱/۹e-۱	۲۰۰	۰/۰۶

جدول ۸: نتایج عددی حاصل از پیاده‌سازی روش‌های مبتنی بر رتبه‌ی تجزیه‌ی سه‌تایی سریع، مقیاس شده‌ی سریع‌ترین جهت کاهش متناوب و تجزیه‌ی قوی روی تصویر طبیعت با بعد ۱۲۷×۷۹

δ	SR-Sp			SR-LS			SR-LR		
	e	k	T(s)	e	k	T(s)	e	k	T(s)
۰/۱	۶/۰e-۱	۱۴۴	۱	۲/۶e-۱	۱۰۵	۰/۳۸	۲/۶e-۱	۱۱۰	۱
۰/۲	۳/۲e-۱	۱۴۲	۱	۱/۵e-۱	۱۲۹	۰/۵۱	۱/۵e-۱	۱۳۲	۱
۰/۳	۱/۹e-۱	۱۲۴	۰/۴۶	۱/۴e-۱	۹۹	۰/۳۵	۱/۴e-۱	۱۰۴	۱
۰/۴	۱/۶e-۱	۱۰۴	۰/۳۸	۱/۲e-۱	۱۰۳	۰/۳۷	۱/۲e-۱	۱۰۹	۱
۰/۵	۱/۲e-۱	۹۷	۰/۳۵	۱/۰e-۱	۱۰۰	۰/۳۶	۱/۰e-۱	۱۰۶	۱
۰/۶	۹/۰e-۲	۹۳	۰/۳۳	۹/۰e-۲	۹۷	۰/۳۵	۱/۰e-۱	۱۰۲	۱
۰/۷	۹/۰e-۲	۸۹	۰/۳۲	۹/۰e-۲	۹۳	۰/۳۵	۹/۰e-۲	۹۸	۱
۰/۸	۸/۰e-۲	۸۶	۰/۳۳	۸/۰e-۲	۹۰	۰/۳۴	۹/۰e-۲	۹۳	۱

جدول ۹: نتایج عددی حاصل از پیاده‌سازی روش‌های خودضربی تنک، کمترین مربعات و رتبه‌پایین روی تصویر طبیعت با بعد ۱۲۷×۷۹

n	r	δ	SR-LR			TriSR-LR		
			e	k	T(s)	e	k	T(s)
۱۰۰۰	۱۰	$1/2e-1$	$5/3e-3$	۷۳	۴۱	$5/3e-3$	۷۴	۴۲
	۵۰	$3/9e-1$	$5/5e-3$	۴۴	۴۴	$5/5e-3$	۴۵	۲۷
	۱۰۰	$5/7e-1$	$5/8e-3$	۴۲	۴۱	$5/8e-3$	۴۳	۲۶
۲۰۰۰	۱۰	$6/0e-2$	$8/2e-3$	۸۹	۸۹۲	$8/3e-3$	۸۹	۵۲۷
	۵۰	$2/5e-1$	$5/0e-3$	۵۴	۵۲۸	$5/1e-3$	۵۴	۳۲۳
	۱۰۰	$3/9e-1$	$5/3e-3$	۴۵	۴۴۷	$5/3e-3$	۴۵	۲۹۲
۳۰۰۰	۱۰	$4/0e-2$	$9/0e-2$	۱۰۷	۲۳۱۳	$9/0e-2$	۱۰۸	۹۰۷
	۵۰	$1/6e-1$	$5/3e-3$	۶۷	۱۴۷۶	$5/3e-3$	۶۷	۵۶۷
	۱۰۰	$2/6e-1$	$5/6e-3$	۵۶	۱۲۵۹	$5/5e-3$	۵۶	۶۴۸

جدول ۱۰: نتایج عددی حاصل از پیاده‌سازی روش سه‌عاملی خودضربی و خودضربی رتبه‌پایین برای ماتریس‌های تصادفی با ستون‌های متعلق به یک زیرفضا با ابعاد $m = n = 1000, 2000, 3000$

n	r	δ	SR-LR			TriSR-LR		
			e	k	T(s)	e	k	T(s)
۱۰۰۰	۵۲	$0/3$	$6/2e-3$	۶۶	۶۷	$6/2e-3$	۶۶	۳۶
		$0/4$	$4/3e-3$	۵۷	۵۶	$4/3e-3$	۵۷	۳۲
		$0/5$	$3/6e-3$	۴۹	۴۹	$3/6e-3$	۴۹	۲۷
۲۰۰۰	۲۰۰	$0/3$	$3/1e-1$	۸۱	۷۷۷	$3/1e-1$	۸۱	۴۸۴
		$0/4$	$1/3e-2$	۷۰	۶۸۴	$1/3e-2$	۷۰	۴۱۲
		$0/5$	$8/9e-3$	۶۰	۶۳۴	$8/9e-3$	۶۰	۴۰۱
		$0/6$	$4/9e-3$	۵۵	۵۵۹	$4/9e-3$	۵۵	۳۳۸

جدول ۱۱: نتایج عددی حاصل از پیاده‌سازی روش سه‌عاملی خودضربی و خودضربی رتبه‌پایین برای ماتریس‌های تصادفی با ستون‌های متعلق به چندین زیرفضا با ابعاد $m = n = 1000, 2000$

n	r	δ	MC		SMC		BMC	
			k	T(s)	k	T(s)	k	T(s)
۱۰۰۰	۱۰	۰/۱۲	۹۹	۱۱۱	۱۱۳	۵۳	۱۱۲	۶۴
	۵۰	۰/۳۹	۶۷	۸۱	۷۸	۳۶	۷۹	۴۸
	۱۰۰	۰/۵۷	۶۵	۷۳	۸۰	۴۰	۸۱	۴۶
۲۰۰۰	۱۰	۰/۰۸	۱۰۴	۱۰۵۰	۱۱۳	۴۰۱	۹۹	۴۲۸
	۵۰	۰/۲۵	۷۴	۶۵۷	۸۴	۲۹۰	۸۴	۳۴۸
	۱۰۰	۰/۳۹	۶۶	۶۷۸	۷۸	۲۷۳	۷۸	۳۳۶

جدول ۱۲: اعمال راهکارهای افزایش سرعت روی روش تنک و مقایسه‌ی آن از نظر سرعت محاسبات با مسئله‌ی تکمیل ماتریس روی ماتریس‌های تصادفی با ستون‌های متعلق به یک زیرفضا با ابعاد $m = n = ۱۰۰۰, ۲۰۰۰$

n	r	δ	MC		SMC		BMC	
			k	T(s)	k	T(s)	k	T(s)
۱۰۰۰	۱۰	۰/۱۲	۹۸	۱۰۶	۱۱۱	۴۶	۱۱۰	۶۳
	۵۰	۰/۳۹	۶۷	۷۲	۷۸	۳۵	۷۹	۴۵
	۱۰۰	۰/۵۷	۶۵	۷۰	۸۰	۳۵	۸۱	۴۸
۲۰۰۰	۱۰	۰/۰۸	۱۰۴	۹۱۶	۱۱۴	۳۵۵	۹۹	۴۲۴
	۵۰	۰/۲۵	۷۴	۶۶۶	۸۴	۲۸۹	۸۴	۳۴۸
	۱۰۰	۰/۳۹	۶۷	۶۳۶	۷۸	۲۶۸	۷۸	۳۴۰

جدول ۱۳: اعمال راهکارهای افزایش سرعت روی روش کمترین مربعات و مقایسه‌ی آن از نظر سرعت محاسبات با مسئله‌ی تکمیل ماتریس روی ماتریس‌های تصادفی با ستون‌های متعلق به یک زیرفضا با ابعاد $m = n = ۱۰۰۰, ۲۰۰۰$

n	r	δ	MC		SMC		BMC	
			k	T(s)	k	T(s)	k	T(s)
۱۰۰۰	۱۰	۰/۱۲	۹۶	۱۰۳	۱۲۳	۵۷	۵۷	۶۷
	۵۰	۰/۳۹	۶۶	۷۷	۷۸	۳۸	۳۹	۴۵
	۱۰۰	۰/۵۷	۶۵	۷۴	۸۰	۳۷	۴۰	۵۰
۲۰۰۰	۱۰	۰/۰۸	۱۰۲	۹۲۰	۱۲۴	۳۸۴	۴۹	۴۱۲
	۵۰	۰/۲۵	۷۳	۷۶۱	۸۴	۲۹۰	۴۲	۳۶۱
	۱۰۰	۰/۳۹	۶۶	۶۳۱	۷۸	۲۶۸	۳۹	۳۴۵

جدول ۱۴: اعمال راهکارهای افزایش سرعت روی روش سه عاملی خودضربی رتبه‌پایین و مقایسه‌ی آن از نظر سرعت محاسبات با مسئله‌ی تکمیل ماتریس روی ماتریس‌های تصادفی با ستون‌های متعلق به یک زیرفضا با ابعاد $m = n = ۱۰۰۰, ۲۰۰۰$

n	r	δ	MC		SMC		BMC	
			k	T(s)	k	T(s)	k	T(s)
۱۰۰۰	۵۰	۰/۳۹	۶۸	۹۸	۸۶	۴۷	۷۵	۳۲
	۱۰۰	۰/۵۷	۶۵	۹۳	۸۵	۴۳	۷۲	۳۱
۲۰۰۰	۵۰	۰/۲۵	۷۶	۸۳۸	۸۹	۳۵۵	۷۹	۳۰۷
	۱۰۰	۰/۳۹	۶۸	۷۵۱	۸۲	۲۸۸	۷۲	۲۸۰

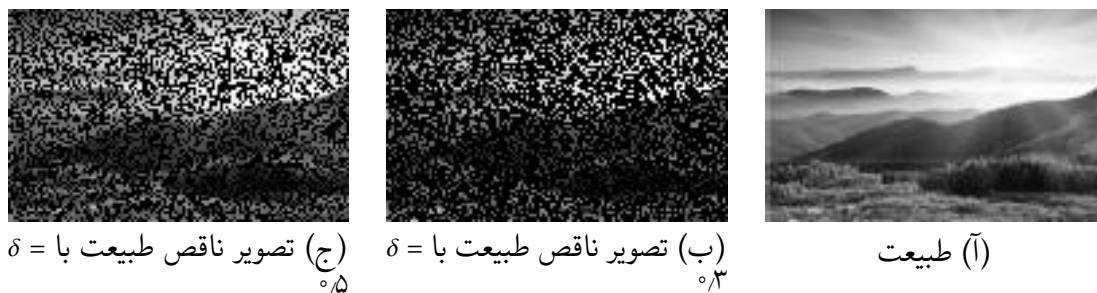
جدول ۱۵: اعمال راهکارهای افزایش سرعت روی روش تنک و مقایسه‌ی آن از نظر سرعت محاسبات با مسئله‌ی تکمیل ماتریس روی ماتریس‌های تصادفی با ستون‌های متعلق به چندین زیرفضا با ابعاد $m = n = ۱۰۰۰, ۲۰۰۰$

n	r	δ	MC		SMC		BMC	
			k	T(s)	k	T(s)	k	T(s)
۱۰۰۰	۵۰	۰/۳۹	۷۱	۱۰۲	۸۷	۴۶	۷۶	۳۲
	۱۰۰	۰/۵۷	۶۸	۹۹	۸۶	۴۴	۷۲	۳۱
۲۰۰۰	۵۰	۰/۲۵	۷۸	۸۶۰	۹۱	۳۵۱	۸۰	۳۰۲
	۱۰۰	۰/۳۹	۶۹	۷۶۱	۸۳	۲۹۳	۷۲	۲۸۰

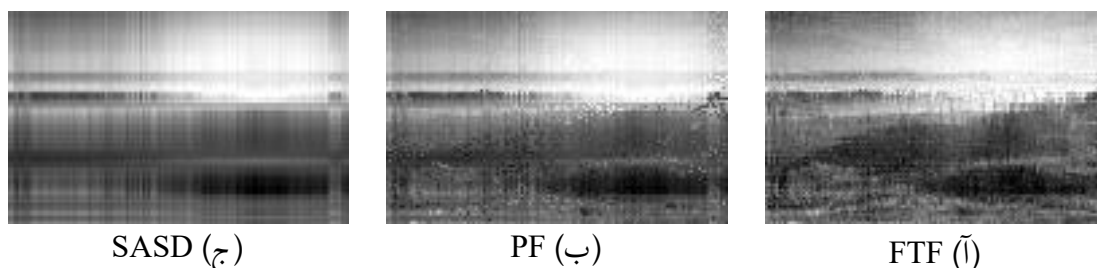
جدول ۱۶: اعمال راهکارهای افزایش سرعت روی روش کمترین مربعات و مقایسه‌ی آن از نظر سرعت محاسبات با مسئله‌ی تکمیل ماتریس روی ماتریس‌های تصادفی با ستون‌های متعلق به چندین زیرفضا با ابعاد $m = n = ۱۰۰۰, ۲۰۰۰$

n	r	δ	MC		SMC		BMC	
			k	T(s)	k	T(s)	k	T(s)
۱۰۰۰	۵۰	۰/۳۹	۶۸	۱۰۰	۸۴	۴۴	۳۵	۳۱
	۱۰۰	۰/۵۷	۶۵	۹۴	۸۲	۴۴	۳۳	۳۰
۲۰۰۰	۵۰	۰/۲۵	۷۶	۸۳۸	۹۰	۳۰۴	۳۸	۲۹۶
	۱۰۰	۰/۳۹	۶۷	۷۴۲	۸۱	۲۹۶	۳۴	۲۷۴

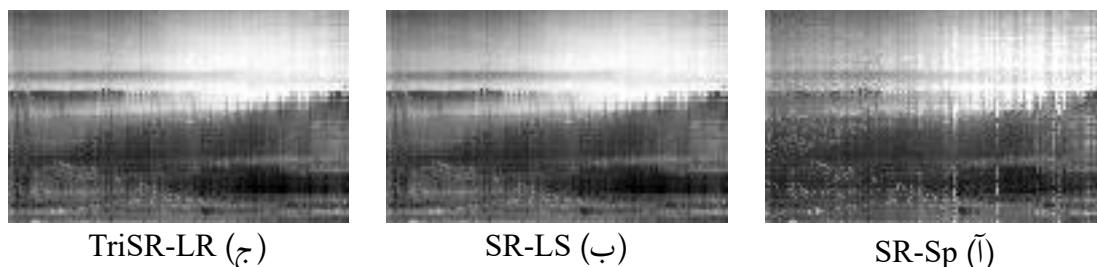
جدول ۱۷: اعمال راهکارهای افزایش سرعت روی روش سه عاملی خودضربی رتبه‌پایین و مقایسه‌ی آن از نظر سرعت محاسبات با مسئله‌ی تکمیل ماتریس روی ماتریس‌های تصادفی با ستون‌های متعلق به چندین زیرفضا با ابعاد $m = n = ۱۰۰۰, ۲۰۰۰$



شکل ۱: تصویر ”طبیعت” همراه با تصاویری که به صورت تصادفی و یکنواخت نقاطی از آن‌ها را حذف کرده‌ایم تا با روش‌های مختلف بازیابی کنیم.



شکل ۲: تصاویر بازیابی شده‌ی تصویر ناقص طبیعت با $\delta = 0.3$ با سه روش مبتنی بر رتبه که در زیر هر تصویر روشی که برای بازیابی استفاده شده مشخص شده است.



شکل ۳: تصاویر بازیابی شده‌ی تصویر ناقص طبیعت با $\delta = 0.3$ با سه روش خودضربی که در زیر هر تصویر روشی که برای بازیابی استفاده شده مشخص شده است.

روش‌های خودضربی تقریب کمترین مربعات و تنک اعمال می‌کنیم. از آنجایی که در بخش قبل نشان دادیم که دو روش خودضربی رتبه‌پایین و روش سه عاملی خودضربی رتبه‌پایین، به جز زمان از همه لحاظ با یکدیگر منطبقند، به جای روش خودضربی رتبه‌پایین راهکارها را روی روش سه عاملی خودضربی رتبه‌پایین پیاده می‌کنیم. کارایی دو راهکار ارائه شده را با اجرای آن‌ها روی ماتریس تصادفی متعلق به یک زیرفضا با ابعاد $m = n = 1000, 2000$ ، رتبه‌ی $r = 10, 50, 100$ و همچنین روی ماتریس‌های تصادفی



SASD (ج)

PF (ب)

FTF (آ)

شکل ۴: تصاویر بازیابی شده‌ی تصویر ناقص طبیعت با $\delta = 0.5$ با سه روش مبتنی بر رتبه که در زیر هر تصویر روشی که برای بازیابی استفاده شده مشخص شده است.



TriSR-LR (ج)

SR-LS (ب)

SR-Sp (آ)

شکل ۵: تصاویر بازیابی شده‌ی تصویر ناقص طبیعت با $\delta = 0.5$ با سه روش خودضربی که در زیر هر تصویر روشی که برای بازیابی استفاده شده مشخص شده است.

متعلق به چندین زیرفضا با ابعاد $m = n = 1000, 2000$ ، رتبه‌ی $r = 50, 100$ با نرخ‌های نمونه‌ی دلخواه بررسی می‌کنیم. جداول ۱۲-۱۴ نتایج حاصل از اعمال راهکارها در تکمیل ماتریس تصادفی با سه روش خودضربی تنک، کمترین مربعات و روش سه‌عاملی خودضربی رتبه‌پایین روی ماتریس‌های تصادفی متعلق به یک زیرفضا و جداول ۱۵-۱۷ نتایج حاصل از اعمال راهکارها در تکمیل ماتریس تصادفی با سه روش خودضربی تنک و تقریب کمترین مربعات و سه‌عاملی خودضربی رتبه‌پایین روی ماتریس‌های متعلق به چندین زیرفضا می‌باشد. بررسی نتایج عددی حاصل از جداول ۱۲-۱۴ نشان می‌دهد که با اعمال راهکار انتخاب یک در میان ستون‌ها شاهد سرعت بیشتری نسبت به بلوک‌بندی ماتریس می‌باشیم و هر دو راهکار نسبت به حل مسئله‌ی تکمیل ماتریس به‌صورت معمول دارای سرعت بیشتری می‌باشند. بررسی نتایج عددی جداول ۱۵-۱۷ نشان می‌دهد که برخلاف ماتریس‌های رتبه‌پایین متعلق به یک زیرفضا با اعمال راهکار بلوک‌بندی روی ماتریس‌های متعلق به چندین زیرفضا شاهد سرعت بیشتری نسبت به انتخاب یک در میان ستون‌ها می‌باشیم، همچنین هر دو راهکار نسبت به حل مسئله‌ی تکمیل ماتریس به‌صورت معمول دارای سرعت بیشتری می‌باشند.

۵. نتیجه‌گیری

در این مقاله به دنبال ارائه‌ی الگوریتم‌های مفیدی برای افزایش سرعت روش‌های خودضربی بودیم تا بتوانیم سرعت عملیات را در آن‌ها افزایش دهیم. نتایج عددی به دست آمده از پیاده‌سازی روش پیشنهادی خودضربی سه‌عاملی رتبه‌پایین نشان می‌دهد که این روش در مقایسه با روش خودضربی رتبه‌پایین دارای سرعت بیشتری بوده و از لحاظ دقت و تعداد تکرارهای لازم تا رسیدن به همگرایی کاملاً با این روش منطبق می‌باشد. همچنین شبیه‌سازی‌های عددی حاصل از اجرای دو ایده‌ی تقسیم بندی ماتریس به دو زیرماتریس نیز مؤید این مطلب است که پیاده‌سازی این ایده‌ها روی ماتریس ناقص باعث افزایش سرعت حل مسئله‌ی تکمیل ماتریس می‌باشد.

مراجع

- [1] E. Adeli-Mosabbeeb and M. Fathy, Non-negative matrix completion for action detection, *Image and Vision Computing*, **39** (2015), 38–51.
- [2] J.F. Cai, E.J. Candès and Z. Shen, A singular value thresholding algorithm for matrix completion, *SIAM J. Optim.*, **20**(4) (2010), 1956–1982.
- [3] J. Fan and T.W. Chow, Matrix completion by least-square, low-rank, and sparse self-representations, *Pattern Recognition*, **71** (2017), 290–305.
- [4] X. Hu, N. Tong, J. Wang, S. Ding and X. Zhao, Matrix completion-based MIMO radar imaging with sparse planar array, *Signal Process.*, **131** (2017), 49–57.
- [5] S.G. Lee and H.G. Seol, A survey on the matrix completion problem, *Trends Math.*, **4** (2001), 38–43.
- [6] Y. Liu, L.C. Jiao and F. Shang, A fast tri-factorization method for low rank matrix recovery and completion, *Pattern Recognition*, **46**(1) (2013), 163–173.
- [7] A. Majumdar and R.K. Ward, Some empirical advances in matrix completion, *Signal Process.*, **91**(5) (2011), 1334–1338.
- [8] Y. Shen, Z. Wen and Y. Zhang, Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization, *Optim. Methods Softw.*, **29**(2) (2014), 239–263.
- [9] J. Tanner and K. Wei, Low rank matrix completion by alternating steepest descent Methods, *Appl. Comput. Harmon. Anal.*, **40**(2) (2016), 417–429.

- [10] J.Y. Wadood, *Image Reconstruction from a Limited Number of Samples: A Matrix-Completion-Based Approach*, Ph.D. thesis, McGill University Libraries, 2011.
- [11] Z. Wen, W. Yin and Y. Zhang, Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm, *Math. Program. Comput.*, 4(24) (2012), 333–361.