باسمه تعالی

دانشکده علوم مهندسی

نیم‌سال اوّل ۰۵ – ۱۴۰۴

پروژه درس جبر خطی

دکتر هادی امیری

طراح بخش نظری:

علیرضا فرهی قهی

طراح بخش عملی:

سجاد نجف خانی

دیدی که یار، جُز سَرِ جور و سِتَم نَداشت بِشْکَسْت عَهْد، وَز غَمِ ما، هیچ غَم نَداشت
یا رَب مَگیرَش اَرْچِه دِلِ چون کَبوتَرَم اَفْکَنْد و کُشْت و عِزَّتِ صیْدِ حَرَم نَداشت
بَر مَن، جَفا زِ بَخْتِ مَن آمَد وَگَرنَه یار حاشا که رَسْمِ لُطْف و طَریقِ کَرَم نَداشت
با این هَمِه، هَر آن‌که نَه خواری کَشید اَز او هَر جا که رَفْت، هیچ کَسَش مُحْتَرَم نَداشت
ساقی بیار بادِه و با مُحْتَسِب بِگو اِنْکارِ ما مَکُن که چُنین جام، جَم نَداشت
هَر راهرو که رَه به حَریمِ دَرَش نَبُرْد مِسْکین بُرید وادی و رَه دَر حَرَم نَداشت
«حافظ»! بِبَر تو گویِ فِصاحَت که مُدَّعی هیچش هُنَر نَبود و خَبَر نیز هَم نَداشت

— خواجه حافظ شیرازی

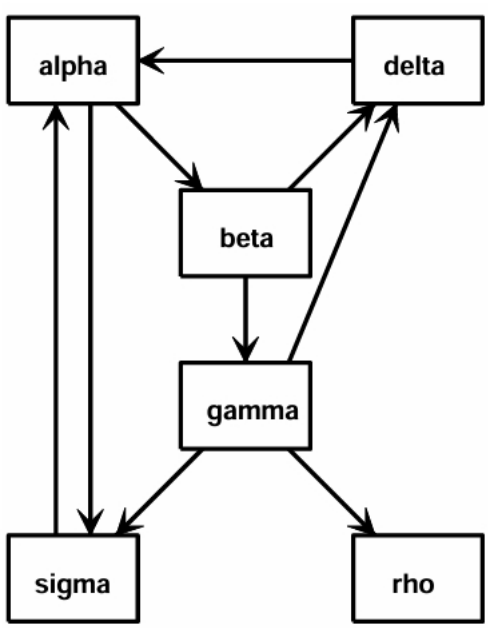
۱ مقدمه و طرح مسئله

امروزه موتورهای جستجو نظیر گوگل، دروازه‌ی اصلی ما برای دسترسی به اطلاعات در میان میلیاردها صفحه وب محسوب می‌شوند. اما پرسش بنیادین این است: یک موتور جستجو چگونه تصمیم می‌گیرد که کدام صفحه را در رتبه‌ی نخست نمایش دهد؟

فرض کنید عبارت «جبر خطی» را جستجو می‌کنید و ۱۰,۰۰۰ صفحه حاوی این عبارت بازیابی می‌شوند. چالش اصلی، یافتن صفحات مرتبط نیست، بلکه تشخیص این نکته است که کدام صفحات مهم‌تر یا معتبرتر هستند.

اینترنت را می‌توان هم‌چون یک گراف جهت‌دار عظیم مدل‌سازی کرد که در آن:

- **گره‌ها (Nodes):** همان صفحات وب هستند.
- **یال‌ها (Edges):** پیوندها (Hyperlinks) از یک صفحه به صفحه‌ی دیگر را نشان می‌دهند.



شکل ۱: نمایشی از یک شبکه وب ۶ تایی

در این پروژه، ما با بهره‌گیری از ابزارهای جبر خطی، ساختار پنهان این گراف را تحلیل کرده و صفحات را رتبه‌بندی می‌کنیم. بدین منظور، دو الگوریتم انقلابی زیر بررسی خواهند شد:

۱. **PageRank**: الگوریتمی که هسته‌ی اولیه‌ی گوگل بر پایه آن بنا شد.

۲. **HITS**: الگوریتمی که هم‌زمان توسعه یافت و ساختار وب را به دو دسته‌ی «مرجع» و «هاب» تفکیک می‌کند.

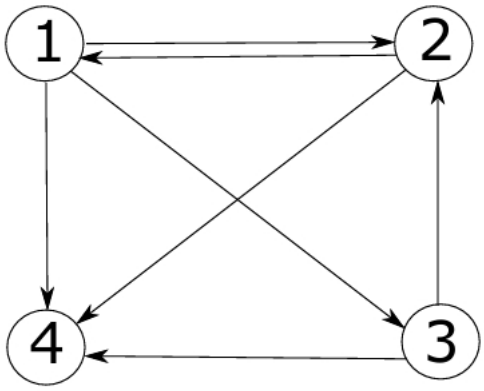
۲ بخش نظری

۱.۲ مدل‌سازی وب به عنوان گراف

فرض کنید مجموعه‌ای متشکل از n صفحه وب در اختیار داریم. این گراف را با یک ماتریس مجاورت (Adjacency Matrix) به نام A با ابعاد $n \times n$ نمایش می‌دهیم. برای سازگاری محاسبات ماتریسی با بردارهای ستونی، از تعریف **ستون‌ـمبنا** استفاده می‌کنیم:

$$A_{ij} = \begin{cases} ۱ & \text{اگر صفحه } j \text{ به صفحه } i \text{ لینک داده باشد} \\ ۰ & \text{در غیر این صورت} \end{cases}$$

سؤال ۱ (۵ نمره). ماتریس مجاورت شبکه وب نشان داده شده در شکل ۲ را بنویسید.



شکل ۲: شبکه وب مربوط به سؤال ۱

۲.۲ از گراف به احتمال: زنجیره مارکوف

در این مدل، فرض بر این است که کاربری در صفحه j حضور دارد و با احتمالی برابر، روی یکی از لینک‌های خروجی کلیک می‌کند تا به صفحه‌ای دیگر منتقل شود. برای تشکیل **ماتریس انتقال احتمال** (Probability Transition Matrix) به نام M، باید ستون‌های ماتریس مجاورت را نرمال‌سازی کنیم.

اگر $c_j = \sum_{i=۱}^{n} A_{ij}$ بیانگر مجموع درایه‌های ستون j (تعداد کل لینک‌های خروجی از صفحه j) باشد، درایه‌های ماتریس M به صورت زیر تعریف می‌شوند:

$$M_{ij} = \begin{cases} \frac{A_{ij}}{c_j}, & c_j \neq ۰ \\ ۰, & c_j = ۰ \end{cases}$$

این رابطه بیان می‌کند که اگر صفحه‌ای دارای لینک خروجی باشد، احتمال انتخاب هر لینک برابر $\frac{۱}{c_j}$ است؛ اما اگر صفحه‌ای هیچ لینک خروجی نداشته باشد (بن‌بست)، ستون مربوط به آن فعلاً تماماً صفر باقی می‌ماند.

سؤال ۲ (۵ نمره). ماتریس M را برای گراف سؤال ۱ تشکیل دهید. نشان دهید که مجموع درایه‌های هر ستون این ماتریس (برای ستون‌هایی که مخرج آن‌ها صفر نیست) برابر با ۱ می‌شود.

۳.۲ الگوریتم PageRank و ماتریس گوگل

در بخش پیشین مشاهده شد که اگر یک صفحه فاقد لینک خروجی باشد (مانند گره ۴ در سؤال ۲)، مجموع ستون مربوط به آن در ماتریس احتمال، صفر می‌شود. این پدیده دو مشکل اساسی ایجاد می‌کند:

۱. ماتریس، خاصیت استوکستیک ستونی خود را از دست می‌دهد.

۲. احتمال حضور کاربر در آن گره به مرور زمان «محو» شده و به صفر میل می‌کند.

اما خاصیت استوکستیک ستونی دقیقاً به چه معناست؟

تعریف ۱.۲. یک ماتریس مربعی حقیقی $G \in \mathbb{R}^{n\times n}$ را **استوکستیک ستونی** (Column Stochastic) می‌نامیم، هرگاه دو شرط زیر برقرار باشند:

۱. **نامنفی بودن:** تمام درایه‌های ماتریس نامنفی باشند:

$$\forall i,j: \quad G_{ij} \geq ۰$$

۲. **نرمال بودن ستون‌ها:** مجموع درایه‌های هر ستون دقیقاً برابر ۱ باشد:

$$\forall j \in \{۱,\dots,n\}: \quad \sum_{i=۱}^{n} G_{ij} = ۱$$

قضیه ۱.۲. اگر G یک ماتریس استوکستیک ستونی باشد، آنگاه عدد حقیقی $\lambda = ۱$ همواره یکی از مقادیر ویژه ماتریس G است.

سؤال ۳ (۲۰ نمره). قضیه ۱.۲ را اثبات کنید.

اکنون برای دستیابی به یک رتبه‌بندی پایدار، باید ماتریس را طی دو مرحله اصلاح کنیم.

۱.۳.۲ گام اول: رفع بن‌بست‌ها (Dead Ends)

فرض می‌کنیم کاربری که به یک صفحه بن‌بست می‌رسد، مرورگر را نمی‌بندد؛ بلکه با احتمالی برابر، یکی از تمام صفحات موجود در وب را به تصادف انتخاب کرده و به آن می‌پرد. بنابراین، اگر ستون j–ام ماتریس M تماماً صفر بود، کلیه درایه‌های آن ستون را با عدد $\frac{۱}{n}$ جایگزین می‌کنیم. ماتریس اصلاح‌شده را S می‌نامیم.

سؤال ۴ (۵ نمره). ماتریس S را برای گراف سؤال ۱ تشکیل دهید. آیا اکنون جمع تمام ستون‌های این ماتریس برابر ۱ است؟

۲.۳.۲ گام دوم: پرش تصادفی و ماتریس گوگل

هنوز یک چالش نظری باقی است: ممکن است کاربر در یک «حلقه» (Cycle) میان چند صفحه گرفتار شود و هرگز به صفحات دیگر نرسد. برای حل این معضل، «لری پیج» و «سرگی برین» مدلی تحت عنوان **موج‌سوار تصادفی** (Random Surfer) را پیشنهاد دادند:

- کاربر با احتمال p (معمولاً $p = ۰/۸۵$) یکی از لینک‌های موجود در صفحه جاری را دنبال می‌کند.
- کاربر با احتمال $۱ - p$ خسته شده و یک آدرس جدید را به تصادف در نوار آدرس تایپ می‌کند (Teleportation).

این ایده ما را به **ماتریس گوگل** (G) رهنمون می‌سازد:

$$G = p \cdot S + (۱ - p) \cdot \frac{۱}{n} \mathbb{J}_{n \times n}$$

که در آن $\mathbb{J}$ ماتریسی است که تمام درایه‌های آن ۱ است. مقدار p یک ضریب بین ۰ و ۱ است.

سؤال ۵ (۱۵ نمره). ثابت کنید اگر S یک ماتریس استوکستیک ستونی باشد، آنگاه ماتریس G نیز لزوماً استوکستیک ستونی خواهد بود.

سؤال ۶ (۵ نمره). ماتریس G را برای گراف سؤال ۱ با در نظر گرفتن $p = ۰/۸۵$ تشکیل دهید.

۳.۳.۲ رتبه‌بندی نهایی و مقادیر ویژه

اکنون که ماتریس G را در اختیار داریم، رتبه‌بندی صفحات وب (بردار $\mathbf{x}$) در واقع همان توزیع احتمال حضور کاربر در بلندمدت است. از آنجا که $\mathbf{x}$ بیانگر احتمال است، مجموع درایه‌های آن باید برابر ۱ باشد

$(\sum x_i = ۱)$.

این توزیع در حالت تعادل صدق می‌کند:

$$G\mathbf{x} = \mathbf{x}$$

این رابطه دقیقاً بیانگر تعریف **بردار ویژه** برای مقدار ویژه $\lambda = ۱$ است (با شرط نرمال‌سازی $\sum x_i = ۱$).

سؤال ۷ (۲۰ نمره). چرا اطمینان داریم که ماتریس G حتماً دارای یک بردار ویژه با درایه‌های مثبت برای مقدار ویژه ۱ است؟

سؤال ۸ (۱۵ نمره). بردار ویژه متناظر با $\lambda = ۱$ را برای سؤال قبل بیابید و صفحات را بر اساس مقدار درایه‌های این بردار رتبه‌بندی کنید.

۴.۳.۲ محاسبه عملی: روش توانی

همان‌طور که در درس آموختید، برای یافتن بردار ویژه متناظر با بزرگترین مقدار ویژه، روش توانی کارآمدترین الگوریتم برای ماتریس‌های بزرگ و تنک (اسپارس) محسوب می‌شود.

سؤال ۹ (۲۰ نمره). در سؤال ۳ ثابت کردید که عدد ۱ همواره یک مقدار ویژه برای ماتریس استوکستیک ستونی است. اکنون ثابت کنید که هیچ مقدار ویژه‌ای نمی‌تواند بزرگتر از ۱ باشد. به بیان دقیق‌تر، نشان دهید اگر λ مقدار ویژه ماتریس استوکستیک ستونی G باشد، آنگاه:

$$|\lambda| \leq ۱$$

در الگوریتم PageRank، این روش معادل شبیه‌سازی رفتار موج‌سوار تصادفی در طول زمان است. با شروع از یک توزیع احتمال اولیه $\mathbf{x}_۰$، رابطه تکراری زیر را دنبال می‌کنیم:

$$\mathbf{x}_{k+۱} = G\mathbf{x}_k$$

این تکرار تا زمانی که تفاوت بین دو بردار متوالی ناچیز شود، ادامه می‌یابد.

سؤال ۱۰ (۱۰ نمره). برای گراف سؤال ۱، الگوریتم فوق را با بردار اولیه $\mathbf{x}_۰ = [\frac{۱}{۴}, \ldots, \frac{۱}{۴}]^T$ تا زمانی‌که تفاوت بین هر درایه دو بردار متوالی از $۱۰^{-۲}$ کمتر شود، اجرا کنید.

۴.۲ الگوریتم HITS

تقریباً هم‌زمان با گوگل، جان کلاینبرگ در شرکت IBM الگوریتم دیگری را برای رتبه‌بندی صفحات وب پیشنهاد داد. برخلاف PageRank که به هر صفحه تنها یک امتیاز اختصاص می‌دهد، الگوریتم HITS برای هر صفحه دو امتیاز متفاوت قائل است:

۱. **امتیاز مرجع (Authority Score) - بردار a:** نشان می‌دهد که محتوای صفحه چقدر معتبر و ارزشمند است (مانند وب‌سایت اصلی یک دانشگاه).

۲. **امتیاز هاب (Hub Score) - بردار h:** نشان می‌دهد که این صفحه چقدر خوب به مراجع معتبر لینک داده است (مانند صفحه‌ای که فهرستی از بهترین منابع آموزشی را گردآوری کرده است).

۱.۴.۲ روابط بازگشتی هاب و مرجع

ایده‌ی اصلی الگوریتم HITS به شرح زیر است:

- یک **مرجع خوب**، صفحه‌ای است که توسط هاب‌های خوبِ بسیاری لینک شده باشد.
- یک **هاب خوب**، صفحه‌ای است که به مراجع خوبِ بسیاری لینک داده باشد.

اگر ماتریس مجاورت A را در نظر بگیریم، این تعاریف را می‌توان به زبان ریاضی ترجمه کرد. اما توجه کنید که اگر صرفاً امتیاز هاب‌ها را جمع بزنیم، اعداد در هر مرحله بزرگ و بزرگتر می‌شوند. برای جلوگیری از این مشکل، در هر مرحله یک **ضریب تناسب** (یا نرمال‌سازی) در نظر می‌گیریم. بنابراین اگر α و β ضرایب ثابت باشند، در حالت تعادل داریم:

$$\mathbf{a} = \alpha A\mathbf{h} \quad \text{(امتیاز مرجعیت متناسب است با مجموع امتیاز هاب‌های ورودی)} \tag{۱}$$

$$\mathbf{h} = \beta A^T\mathbf{a} \quad \text{(امتیاز هاب بودن متناسب است با مجموع امتیاز مراجع خروجی)} \tag{۲}$$

سؤال ۱۱ (۱۰ نمره). با جایگذاری معادله (۲) در (۱) و بالعکس، نشان دهید که $\mathbf{a}$ و $\mathbf{h}$ باید در روابط زیر صدق کنند:

$$(AA^T)\mathbf{a} = \lambda\mathbf{a}$$

$$(A^T A)\mathbf{h} = \mu \mathbf{h}$$

سؤال ۱۲ (۱۵ نمره). برای گراف شبکه وب (سؤال ۱):

۱. ماتریس AA^T را تشکیل دهید. بزرگترین مقدار ویژه آن (λ_{max}) و بردار ویژه متناظر با آن ($\mathbf{a}$ یا همان امتیاز Authority) را بیابید.

۲. ماتریس $A^T A$ را تشکیل دهید. بزرگترین مقدار ویژه آن (λ_{max}) و بردار ویژه متناظر با آن ($\mathbf{h}$ یا همان امتیاز Hub) را بیابید.

۳. کدام صفحه «مرجع» برتر و کدام صفحه «هاب» برتر است؟

۲.۴.۲ ارتباط عمیق با تجزیه مقادیر منفرد (SVD)

در انتهای درس جبر خطی با «تجزیه مقادیر منفرد» یا SVD آشنا شدید. طبق این تجزیه، هر ماتریس دلخواه A را می‌توان به صورت زیر بسط داد:

$$A = U\Sigma V^T$$

اکنون می‌خواهیم نشان دهیم که الگوریتم HITS، در ماهیت خود، چیزی جز محاسبه SVD ماتریس وب نیست!

سؤال ۱۳ (۵ نمره). در بخش‌های قبل ثابت کردیم که:

- بردار امتیاز مرجع ($\mathbf{a}$) بردار ویژه اصلی ماتریس AA^T است.
- بردار امتیاز هاب ($\mathbf{h}$) بردار ویژه اصلی ماتریس $A^T A$ است.

از طرفی در درس آموختیم که بین مؤلفه‌های تجزیه مقدار منفرد ($A = U\Sigma V^T$) و بردارهای ویژه ماتریس‌های AA^T و $A^T A$ رابطه مستقیمی برقرار است.

با توجه به این مقدمه، جملات زیر را کامل کنید:

«در الگوریتم HITS، بردار مرجعیت وب‌سایت‌ها ($\mathbf{a}$)، دقیقاً همان اولین ستون ماتریس ..... (بردار منفرد چپ) در تجزیه SVD است و بردار هاب ($\mathbf{h}$)، دقیقاً همان اولین ستون ماتریس ..... (بردار منفرد راست) است.»

۳ بخش عملی: پیاده‌سازی روی داده‌های واقعی

در این بخش، قرار است الگوریتم‌های فوق را روی یک شبکهٔ اجتماعی واقعی اجرا کنید. برای این منظور، فایل `wiki-Vote` را از این آدرس دریافت کنید. فایل مورد نظر، حاوی اطلاعات مربوط به گرافی جهت‌دار با ۸۲۹۷ رأس و ۱۰۳۶۸۹ یال است.

می‌دانیم که ویکی‌پدیا با مشارکت داوطلبان از سراسر جهان نوشته می‌شود. بخش کوچکی از مشارکت‌کنندگان ویکی‌پدیا مدیران هستند؛ کاربرانی با دسترسی به ویژگی‌های فنی اضافی که به نگهداری سامانه کمک می‌کنند. برای این‌که کاربری به مدیر تبدیل شود، یک «درخواست مدیریت» ارائه می‌دهد و جامعهٔ ویکی‌پدیا از طریق یک بحث عمومی یا رأی‌گیری تصمیم می‌گیرد که چه کسی به مدیریت ارتقا یابد.

شبکه‌ای که دانلود کرده‌اید، شامل تمامی داده‌های رأی‌گیری ویکی‌پدیا از آغاز تأسیس تا ژانویه ۲۰۰۸ است. گره‌های شبکه نشان‌دهندهٔ کاربران ویکی‌پدیا هستند و وجود یک یال جهت‌دار از گره i به گره j نشان می‌دهد که کاربر i به کاربر j رأی داده است.

مرحلهٔ ۱: آماده‌سازی داده‌ها

ماتریس مجاورت مربوط به شبکهٔ مورد نظر را تشکیل دهید و آن را `AdjMatrix` بنامید.

سؤال ۱۴ (۱۰ نمره). چنانچه شبکهٔ ما یک گراف متراکم باشد (یعنی تقریباً همهٔ گره‌ها به هم وصل باشند)، برای ذخیره‌سازی ماتریس مجاورت حدوداً چه مقدار حافظه مورد نیاز است؟

ذخیره‌سازی ماتریسی با چنین ابعادی، به‌خودی‌خود دشوار نیست؛ اما باید توجه داشت که فراخوانی‌ها و عملیات متعددی روی آن صورت می‌گیرد. این‌جاست که به اهمیت شناخت بعضی ماتریس‌های خاص و ویژگی‌های آن‌ها، مثل ماتریس‌های اسپارس، پی می‌بریم.

سؤال ۱۵ (۱۰ نمره). درجهٔ اسپارس بودن ماتریس `AdjMatrix` چقدر است؟

سؤال ۱۶ (۱۰ نمره). آیا این ماتریس متقارن است؟

مرحلهٔ ۲: پیاده‌سازی PageRank

سؤال ۱۷ (۲۵ نمره). تابعی توسعه دهید که پیج‌رنک را به روش توانی محاسبه کند.

سؤال ۱۸ (۲۰ نمره). تابع خود را روی داده‌های `wiki-vote` اجرا کنید. مقدار p (در ساختن ماتریس گوگل) را برابر با ۰٫۸۵ در نظر بگیرید و الگوریتم را تا زمانی که تفاوت بین دو بردار متوالی کمتر از $۱۰^{-۶}$ شود، ادامه دهید. پنج کاربر برتر و مقدار PageRank آن‌ها را گزارش کنید.

توجه: در صورتی که الگوریتم ارائه‌شده توسط شما در مدت‌زمان کمتری نسبت به سایر دانشجویان همگرا شود، نمرهٔ امتیازی در نظر گرفته خواهد شد. میزان امتیاز به مدت‌زمان همگرایی الگوریتم سایرین بستگی دارد.

مرحلهٔ ۳: پیاده‌سازی HITS

سؤال ۱۹ (۲۵ نمره). تابعی توسعه دهید که الگوریتم HITS را اجرا کند.

سؤال ۲۰ (۲۰ نمره). تابع خود را روی مجموعه دادهٔ مربوط به پروژه پیاده کنید. پنج مرجع برتر (یعنی کسانی که بیش از همه رأی‌های معتبر دریافت کرده‌اند) و پنج هاب برتر (یعنی کسانی که به افراد معتبر بیشتری رأی داده‌اند) را گزارش کنید.

یادآوری: در صورتی که الگوریتم ارائه‌شده توسط شما در مدت‌زمان کمتری نسبت به سایر دانشجویان همگرا شود، نمرهٔ امتیازی در نظر گرفته خواهد شد. میزان امتیاز به مدت‌زمان همگرایی الگوریتم سایرین بستگی دارد.

مرحلهٔ ۴: مقایسهٔ نتایج

گاهی در شبکه‌های اجتماعی تصور می‌شود فردی که بیشترین دنبال‌کننده (در این‌جا: بیشترین رأی) را دارد، مؤثرترین فرد شبکه است؛ در حالی‌که می‌دانیم همیشه این‌طور نیست و احیاناً مثال‌های نقضی در زندگی واقعی و دنیای مجازی دیده‌ایم. این امر در این پروژه نیز مشهود است.

سؤال ۲۱ (۱۰ نمره). با استفاده از ماتریس مجاورت، پنج کاربری را که بیشترین تعداد رأی را دارند، بیابید. این لیست را با لیست‌های به‌دست‌آمده از سؤال ۱۸ و ۲۰ مقایسه کنید. دلیل تفاوت را از منظر جبر خطی توضیح دهید.

همچنین شبکه‌های اجتماعی مختلف (مثل اینستاگرام، X (توییتر سابق)، یوتیوب و...) روندهای گوناگونی برای دیده‌شدن و اعتبار بخشیدن به صفحات و پست‌ها دارند. مثلاً چنانچه یک هاب، توییتی از

یک مرجع را ریتوییت کند، به اعتبار هر دو افزوده می‌شود؛ حال آن‌که در اینستاگرام مسئله دقیقاً به این شکل نیست.

سؤال ۲۲ (۲۰ نمره). با استفاده از آن‌چه در سؤال ۱۲ دیدید، بزرگ‌ترین مقدار ویژه و بردار ویژهٔ متناظر با ماتریس‌ها را محاسبه کنید و با مقایسهٔ نتایج با بردار Authority به‌دست‌آمده در سؤال ۲۰، بررسی کنید که آیا (تقریباً) یکسان هستند یا خیر؟

تذکر ۱: محاسبات svd در ماتریس‌های اسپارس، دستورهای خاص جداگانه دارند.

تذکر ۲: برای مقایسهٔ بردارها می‌توانید از مقایسهٔ زاویه‌ها استفاده کنید.