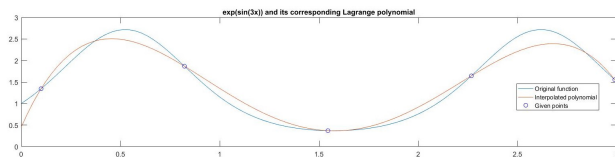


پروژه نهایی جبر خطی

فاطمه نائینیان (۸۱۰۱۹۸۴۷۹)، علی هادی زاده مقدم (۸۱۰۱۹۸۴۸۵)
دانشکده مهندسی برق و کامپیوتر، پردیس دانشکده‌های فنی، دانشگاه تهران

Email: Fatemeh.Naeinian@gmail.com - hadizadeh.ali@ut.ac.ir

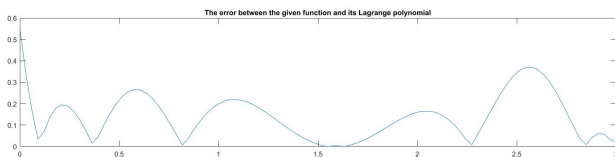
فهرست مطالب	
۱	مقدمه
۲	سوال اول
۳	سوال دوم
۴	سوال سوم
۵	سوال چهارم
۶	سوال پنجم
۷	پیوستگی
۸	سوال دوم: نمودار خطای گرام-اشمیت
۹	سوال سوم: نمودار واریانس تجمعی
۱۰	سوال سوم: دقت در ۲۵۰ مقدار ویژه برتر
۱۱	سوال سوم: دقت در ۵۰۰ مقدار ویژه برتر
۱۲	سوال سوم: تصویر نمونه اول
۱۳	سوال سوم: تصویر نمونه دوم
۱۴	سوال چهارم: تصویر اصلی
۱۵	سوال چهارم: تصویر به ازای مقادیر مختلف تکین
۱۶	سوال چهارم: تصویر به ازای ۳۰ درصد بزرگترین مقادیر تکین
۱۷	سوال پنجم: فیلم های پیشنهادی به کاربر نمونه با نرم اقلیدسی
۱۸	سوال پنجم: فیلم های پیشنهادی به کاربر نمونه با ضریب به هم پیوستگی
۱۹	سوال پنجم: امتیاز دهی ما به فیلم های نمونه
۲۰	سوال پنجم: فیلم های پیشنهادی به ما با نرم اقلیدسی
۲۱	سوال پنجم: فیلم های پیشنهادی به ما با ضریب به هم پیوستگی
۲	سوال اول: نمودار $f(x)$ و منحنی تخمین زده شده
۳	سوال دوم: تخمین مدل به روش Pseudo Inverse
۴	سوال دوم: الگوریتم QR
۵	سوال دوم: الگوریتم QR
۶	سوال دوم: تخمین مدل به روش گرام-اشمیت



شکل ۱: سوال اول: نمودار $f(x)$ و منحنی تخمین زده شده

ج.

در این بخش مقدار خطا را نمایش میدهیم. خطا به صورت قدر مطلق منحنی درونیابی شده و منحنی اصلی در هر نقطه تعریف می شود.



شکل ۲: سوال اول: نمودار خطای بین دو منحنی

۳. سوال دوم

آ.

در سوال دوم ابتدا به توضیح الگوریتم میپردازیم. برای تخمین پارامتر ها میخواهیم از الگوریتم حداقل مربعات استفاده کنیم. در این الگوریتم به دو روش متفاوت پاسخ ها را مقایسه خواهیم کرد. در روش اول با الگوریتم شبه معکوس Pseudo Inverse پاسخ را پیدا میکنیم. در این الگوریتم داده های x و y را با کمک تابع `normrnd` تولید میکنیم.

$$\begin{aligned} x_1 &= \text{normrnd}(0, 1, [1, N]) \\ x_2 &= \text{normrnd}(0, 1, [1, N]) \\ x_3 &= x_1 - x_2 + \text{normrnd}(0, 1e-20, [1, N]) \end{aligned}$$

آنها را به شکل یک ماتریس مینویسیم.

$$X = [x_1; x_2; x_3]$$

حال با کمک این ماتریس و فرمول اصلی داده شده $(y = 2x_1 + 3x_2 - 4x_3)$ مقادیر y ها را به دست می آوریم و به صورت یک ماتریس مینویسیم.

حال داده ها را به دو دسته `train` و `test` تقسیم میکنیم.

در روش Pseudo Inverse به دلیل اینکه ماتریس X معکوس پذیر نیست، ابتدا معادله را در X^T ضرب میکنیم. حال معادله معکوس پذیر می شود و میتوان θ را پیدا کرد.

$$\begin{aligned} y &= X\theta \\ X^T y &= X^T X \theta \\ \theta &= (X^T X)^{-1} X^T y \end{aligned}$$

حال که θ ها را به دست آوردیم. داده های `test` را مدل میکنیم و y پیش بینی شده برای آنها را به دست می آوریم. سپس روی نمودار y اصلی و y پیش بینی شده را نمایش می دهیم. در این بخش به دلیل اینکه بتوان y ها را بهتر مقایسه کرد و به شکل واضح تری آنها را ببینیم، y ها را `sort` کردیم و با کمک `index` داده های مرتب شده داد های تخمین زده شده را نیز مرتب کردیم و سپس نمایش دادیم.

چکیده—در این پروژه، به بررسی عملی تعدادی از مباحث درس جبر خطی می پردازیم. این مباحث عبارتند از درونیابی معادلات غیر چندجمله ای به کمک چندجمله ای های لاگرانژ، تخمین ارتباط یک خروجی به ورودی های اندازه گیری شده، کاهش حجم تصویرها به کمک PCA و SVD و تحلیل علایق کاربران در یک سامانه ای رای دهی به فیلم.

کلمات کلیدی—PCA، SVD، حداقل مربعات

۱. مقدمه

در پروژه جاری ابتدا درونیابی لاگرانژ را بررسی میکنیم. به صورتی که با داشتن چند نقطه از یک منحنی میتوانیم یک منحنی پیدا کنیم که از تمامی نقاط عبور کند و برای سایر نقاط نیز پیش بینی خوبی از نتایج داشته باشد. در سوال دوم میخواهیم با داشتن ورودی و خروجی های یک مدل، بهترین تخمین را از آن مدل به دست بیاوریم. برای این کار الگوریتم حداقل مربعات را به دو روش پیاده سازی میکنیم. در سوال سوم یک مجموعه داده از عکس ها داریم که میخواهیم با کمک روش PCA حجم داده ها کاهش دهیم تا در محاسبات صرفه جویی شود. همچنین در بعضی از مواقع صرفاً با اطلاعات اصلی میتوانیم مساله را حل کنیم و به جزئیات نیازی نداریم. در سوال چهارم نیز یک عکس دلخواه را انتخاب کرده و با کمک الگوریتم SVD ابعاد را کاهش میدهیم. این روش کمک میکند تا عکس را فشرده سازی کنیم و بزرگترین مقادیر تکین را نگه میداریم. در سوال پنجم نیز از یک مجموعه داده، ۲۰ فیلم برتر را پیدا میکنیم و سپس با توجه به این ۲۰ فیلم به کاربران فیلم پیشنهاد میدهیم.

لازم به ذکر است در این پروژه در تمامی کارها از جمله نوشتن گزارش و کد زدن، هردو حضور فعال داشتیم و در راستای پیشبرد پروژه سهم یکسانی داریم.

۲. سوال اول

آ.

ابتدا به شرح و تفضیل درونیابی لاگرانژ میپردازیم. در این روش با داشتن نقاطی از یک منحنی، میتوانیم یک چند جمله ای برای آن درونیابی کنیم. با داشتن n نقطه میتوانیم یک چندجمله ای حداکثر از مرتبه $n-1$ به آن برازش کنیم. فرض کنید داده های زیر را داریم:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

با کمک داده ها برای هر داده تابع زیر را به صورت زیر حساب میکنیم.

$$f_j(x) = \frac{(x-x_1) \dots (x-x_k)}{(x_j-x_1) \dots (x_j-x_k)} = \prod_{1 \leq m \leq n, m \neq j} \frac{(x-x_m)}{(x_j-x_m)}$$

حال چند جمله ای لاگرانژ را به شکل زیر تعریف میکنیم.

$$y = \sum_{j=1}^k y_j f_j(x)$$

حال در این سوال ۵ نقطه داده شده است. پس میتوانیم به منحنی با حداکثر درجه ۴ به نقطه ها برازش کنیم. با کمک MATLAB دو تابع به نام `Lagrange.m` و `f.m` تعریف کردیم که در تابع `f.m` برای هر نقطه چند جمله ای متناظر آن را پیدا میکند. سپس در تابع `Lagrange.m` مجموع حاصل ضرب y_i ها و f_i ها را محاسبه میکنیم.

ب.

حال در بازه ۰ تا ۳ با کمک `linspace` صد نقطه ایجاد میکنیم و به منحنی درونیابی شده میدهیم. حال خروجی منحنی لاگرانژ و خروجی اصلی تابعی که آن را تخمین زده ایم را نمایش میدهیم.

where $\langle \mathbf{e}_i, \mathbf{a}_i \rangle = \|\mathbf{u}_i\|$. This can be written in matrix form:

$$A = QR$$

where:

$$Q = [\mathbf{e}_1 \ \cdots \ \mathbf{e}_n]$$

and

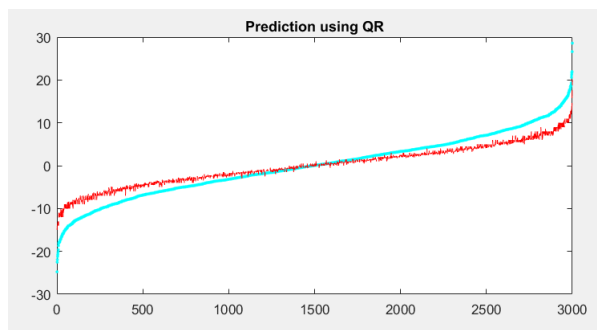
$$R = \begin{bmatrix} \langle \mathbf{e}_1, \mathbf{a}_1 \rangle & \langle \mathbf{e}_1, \mathbf{a}_2 \rangle & \langle \mathbf{e}_1, \mathbf{a}_3 \rangle & \cdots & \langle \mathbf{e}_1, \mathbf{a}_n \rangle \\ 0 & \langle \mathbf{e}_2, \mathbf{a}_2 \rangle & \langle \mathbf{e}_2, \mathbf{a}_3 \rangle & \cdots & \langle \mathbf{e}_2, \mathbf{a}_n \rangle \\ 0 & 0 & \langle \mathbf{e}_3, \mathbf{a}_3 \rangle & \cdots & \langle \mathbf{e}_3, \mathbf{a}_n \rangle \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \langle \mathbf{e}_n, \mathbf{a}_n \rangle \end{bmatrix}$$

شکل ۵: سوال دوم : الگوریتم QR

حال با داشتن Q و R میتوانیم Θ را پیدا کنیم. برای این کار باید به روش زیر عمل کنیم.

$$\Theta = R^{-1} Q^T y$$

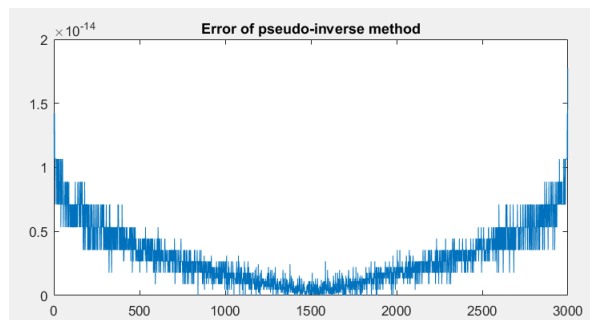
سپس با کمک داده های تست، y ها را پیش بینی میکنیم و نمایش میدهیم.



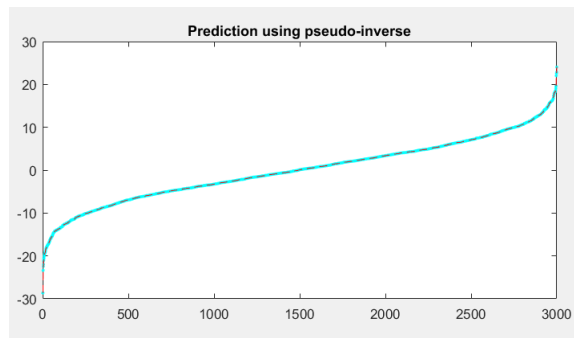
شکل ۶: سوال دوم : تخمین مدل به روش گرام-اشمیت

د. میبینیم در روش گرام-اشمیت خطای بیشتری وجود دارد ولی با این حال این تخمین میتواند خوب عمل کند.

ه. برای محاسبه خطا، برای هر داده yi، قدرمطلق تفاضل خروجی اصلی و تخمین زده شده را به دست می آوریم. حال نمودار خطا را برای روش Pseudo Inverse به دست می آوریم.



شکل ۷: سوال دوم : نمودار خطای Pseudo Inverse



شکل ۳: سوال دوم : تخمین مدل به روش Pseudo Inverse

مشاهده می شود تخمین ما به خوبی عمل کرده است و خروجی پیش بینی شده با خروجی اصلی مطابقت دارد و خطای بسیار کمی مشاهده می شود.

ج.

حال روش گرام-اشمیت را بررسی میکنیم. در این روش ماتریس X را به دو ماتریس Q و R تفکیک میکنیم. برای این کار الگوریتم زیر را پیاده سازی میکنیم.

Define the projection:

$$\text{proj}_{\mathbf{u}} \mathbf{a} = \frac{\langle \mathbf{u}, \mathbf{a} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle} \mathbf{u}$$

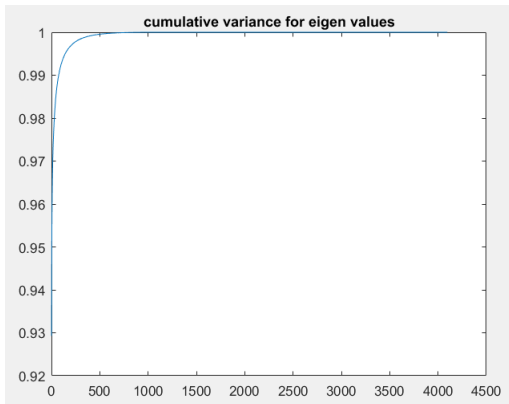
then:

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{a}_1, & \mathbf{e}_1 &= \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \\ \mathbf{u}_2 &= \mathbf{a}_2 - \text{proj}_{\mathbf{u}_1} \mathbf{a}_2, & \mathbf{e}_2 &= \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \\ \mathbf{u}_3 &= \mathbf{a}_3 - \text{proj}_{\mathbf{u}_1} \mathbf{a}_3 - \text{proj}_{\mathbf{u}_2} \mathbf{a}_3, & \mathbf{e}_3 &= \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|} \\ &\vdots & &\vdots \\ \mathbf{u}_k &= \mathbf{a}_k - \sum_{j=1}^{k-1} \text{proj}_{\mathbf{u}_j} \mathbf{a}_k, & \mathbf{e}_k &= \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|} \end{aligned}$$

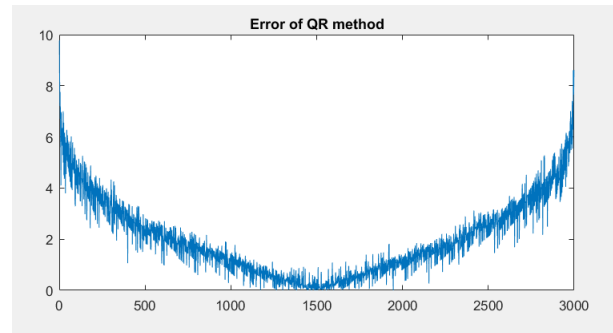
We can now express the \mathbf{a}_i 's over our newly computed orthonormal basis:

$$\begin{aligned} \mathbf{a}_1 &= \langle \mathbf{e}_1, \mathbf{a}_1 \rangle \mathbf{e}_1 \\ \mathbf{a}_2 &= \langle \mathbf{e}_1, \mathbf{a}_2 \rangle \mathbf{e}_1 + \langle \mathbf{e}_2, \mathbf{a}_2 \rangle \mathbf{e}_2 \\ \mathbf{a}_3 &= \langle \mathbf{e}_1, \mathbf{a}_3 \rangle \mathbf{e}_1 + \langle \mathbf{e}_2, \mathbf{a}_3 \rangle \mathbf{e}_2 + \langle \mathbf{e}_3, \mathbf{a}_3 \rangle \mathbf{e}_3 \\ &\vdots \\ \mathbf{a}_k &= \sum_{j=1}^k \langle \mathbf{e}_j, \mathbf{a}_k \rangle \mathbf{e}_j \end{aligned}$$

شکل ۴: سوال دوم : الگوریتم QR



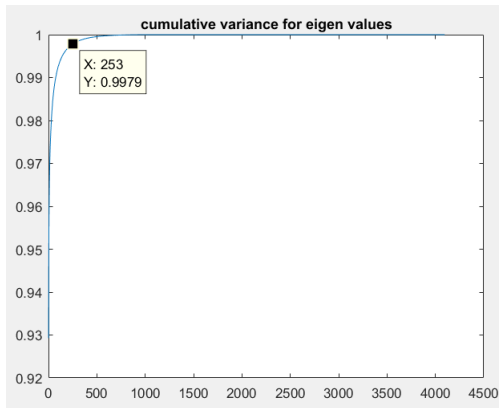
شکل ۹: سوال سوم : نمودار واریانس تجمعی



شکل ۸: سوال دوم : نمودار خطای گرام اشمیت

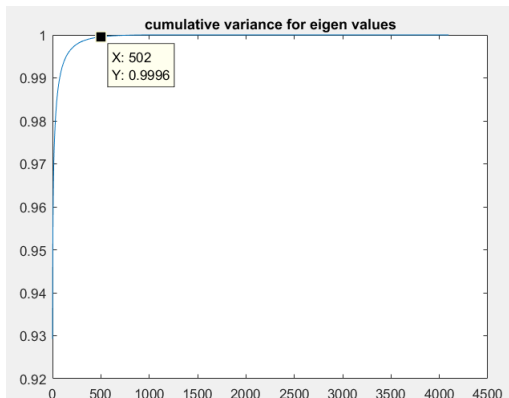
با کمک روش آرنج با توجه به شکل زیر میبینیم که در حدود ۲۵۰ مقدار ویژه برتر به دقت خوبی میرسیم.

۴. سوال سوم



شکل ۱۰: سوال سوم : دقت در ۲۵۰ مقدار ویژه برتر

همینطور مشاهده می شود که در حدود ۵۰۰ مقدار ویژه برتر به دقت حدود ۱ میرسیم. بنابراین مناسب ترین تعداد مقادیر ویژه ۵۰۰ تا است.



شکل ۱۱: سوال سوم : دقت در ۵۰۰ مقدار ویژه برتر

نمودار خطا را برای روش گرام-اشمیت به شکل زیر داریم.

شکل ۸: سوال دوم : نمودار خطای گرام اشمیت

آ.

ابتدا به شرح و تفضیل الگوریتم pca می پردازیم.

در این الگوریتم برای فشرده سازی داده ها، بعد داده ها را کاهش میدهیم. حتی الامکان سعی می شود تا داده های زیادی از دست نرود. بدین منظور ابتدا مقادیر ویژه و بردار های ویژه را پیدا میکنیم.

$$X^T X = W L W$$

سپس بزرگترین مقادیر ویژه را نگه میداریم. برای این کار از یک ماتریس قطری F استفاده میکنیم که برای مقادیر ویژه مدنظر قطر آن برابر ۱ و در غیر این صورت قطر آن برابر ۰ است.

$$X_{new} = X W F W^T$$

برای انتخاب تعداد مناسب مقادیر ویژه از روش آرنج استفاده می شود. در این ابتدا واریانس تجمعی را محاسبه میکنیم. سپس آن را نمایش میدهیم. با پیدا کردن بهترین تعداد مقدار ویژه، کاهش بعد را اعمال میکنیم و برای نمونه چند تصویر را نمایش میدهیم.

ب.

ابتدا هزار تصویر را وارد MATLAB میکنیم. آنها را reshape میکنیم و به یک بردار سطری تبدیل میکنیم و همه آنها را پشت هم قرار داده و یک ماتریس کلی X به دست می آوریم.

ج.

مقادیر ویژه و بردار های ویژه را به دست می آوریم و آنها را sort میکنیم. حال واریانس تجمعی را برای مقادیر ویژه به دست می آوریم و نمایش میدهیم.

$$newImage = US_{new}V$$

ب.

حال عکس یک صخره را انتخاب کرده ایم که سایز بزرگتری دارد و آن را با imcrop به سایز 300×300 میرسانیم.



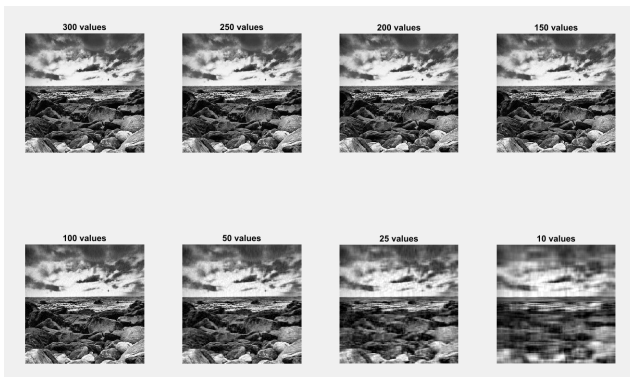
شکل ۱۴: سوال چهارم: تصویر اصلی

ج.

با تابع my_svd که در بخش قبل نوشته ایم، این ماتریس را به U و V تفکیک میکنیم. در این تابع مقادیر ویژه را برای AA^T و A^TA محاسبه میکنیم و سپس آنها را sort میکنیم و از مقادیر ویژه نیز جذر میگیریم. و سپس به صورت خروجی به فایل اصلی برمیگردانیم.

د.

حال به ازای تعداد مختلفی از مقادیر تکین، تصویر را رسم میکنیم.



شکل ۱۵: سوال چهارم: تصویر به ازای مقادیر مختلف تکین

در کل ۳۰۰ مقدار تکین داریم. طبیعتاً هرچه مقادیر تکین کمتری استفاده کنیم از کیفیت عکس کم می شود. ولی نکته قابل توجه این است که چون مقادیر تکین بزرگتر را نگه میداریم، مقادیر تکینی که حذف می شوند تاثیر کمتری بر تصویر دارند. با توجه به شکل بالا

حال برای ۶ حالت ۴۰ و ۷۰ و ۱۵۰ و ۲۵۰ و ۵۰۰ و ۴۰۹۶ مقدار ویژه بزرگ تر، چند تصویر را رسم میکنیم تا به وضوح تاثیر تعداد مقدار ویژه را ببینیم.



شکل ۱۲: سوال سوم: تصویر نمونه اول

مشاهده می شود که در حالت ۴۰ و ۷۰ و ۱۵۰ تصاویر اطلاعات زیادی از دست داده اند. اما برای ۲۵۰ تا حدی قابل قبول است. و در حالت ۵۰۰ تصویر تفاوت زیادی با حالتی که همه مقادیر ویژه استفاده شده است ندارد.



شکل ۱۳: سوال سوم: تصویر نمونه دوم

۵. سوال چهارم

آ.

ابتدا به شرح و تفصیل الگوریتم SVD میپردازیم. در این الگوریتم ماتریس مد نظر را به سه ماتریس تفکیک میکنیم.

$$M = USV^T$$

بدین ترتیب داریم، U بردار ویژه های AA^T است. و V بردار ویژه های A^TA است. سپس با توجه به مقادیر ویژه آنها را sort میکنیم. همچنین باید توجه کرد که ماتریس S برابر جذر مقادیر ویژه است. حال برای فشرده سازی تصویر به تعداد مناسب از مقادیر ویژه را نگه میداریم و بقیه آنها را صفر میکنیم.

در این سوال الگوریتم SVD در فایل متلب my_svd.m موجود است. همچنین برای انتخاب مقادیر ویژه خاصی، در فایل selectSin-gularValue.m مقادیر ویژه ای که نیاز نداریم را صفر میکنیم و ماتریس جدیدی برای مقادیر ویژه تولید میکنیم. حال با داشتن U و V و S_{new} میتوان تصویر کاهش بعد یافته را نمایش داد.

کاربران بر اساس فاصله، صرفاً نزدیک‌ترین کاربر را پیدا می‌کنیم. این کار در تابع `fittesteuclid` انجام شده است.

از لحاظ مقایسه، استفاده از ضریب همبستگی پیرسون نسبت به استفاده از فاصله‌ی اقلیدسی مزیت قابل توجهی دارد: بعضی کاربران به طور متوسط امتیاز بهتری به فیلم‌ها می‌دهند، که در میانگین امتیازهای‌شان نمایان می‌شود. هم‌چنین، بعضی کاربران ممکن است که به صورت اغراق‌آمیز به فیلم‌ها رای بدهند، که انحراف معیار امتیازهای‌شان را افزایش می‌دهد. دیدگاهی که به نظر مناسب می‌رسد این است که می‌بایست تاثیر آسان‌گیر یا سخت‌گیر بودن کاربران (میانگین)، و اغراق‌آمیز یا بی‌تفاوت نمره دادن آن‌ها (انحراف معیار) را حذف کنیم، که این کار در فرمول ضریب همبستگی پیرسون صورت می‌گیرد، اما می‌تواند در فاصله‌ی اقلیدسی تاثیر زیادی بگذارد. برای همین، ضریب همبستگی پیرسون به نظر معیار مناسب‌تری است. البته ما در هر حال برای کاربر آزمایشی و هم‌چنین خودمان (بخش‌های آخر صورت سوال) از هر دو معیار برای پیشنهاد فیلم استفاده می‌کنیم. این کار در تابع `interact` صورت گرفته است. این تابع ضمن اعلام شماره‌ی نزدیک‌ترین کاربر (که در این مسئله برای بررسی اعلام می‌شود، وگرنه نباید در یک سامانه‌ی عمومی آن را نشان داد)، فهرستی از ۲۰ فیلم که کاربر منتخب به آن‌ها بیشترین امتیاز را داده، اعلام می‌کند. این کار برای هر دو روش محاسبه‌ی استفاده شده صورت می‌گیرد.

د.

در تابع `fittestpearson` از ضریب همبستگی پیرسون بهره گرفته‌ایم و طبق خروجی برنامه، نتیجه می‌گیریم که انتخاب معیار در فهرست فیلم‌های پیشنهاد شده تاثیر به‌سزایی دارد.

ه.

در تابع `suggest` یک کاربر را به عنوان ورودی می‌گیریم و ۲۰ مورد از فیلم‌هایی که آن کاربر به آن‌ها بیشترین رای را داده، می‌یابیم. در تابع `interact` که در خط ۱۴۳ برای کاربر آزمایشی صدا زده شده، از این تابع به همراه تابع `fittesteuclid` و `fittestpearson` استفاده کرده‌ایم.

```
Suggested user (Euclidean distance): 3963
Suggested movies (Euclidean distance):
Pawnbroker, The (1965) (Drama)
Adventures of Rocky and Bullwinkle, The (2000) (Animation|Children's|Comedy)
Lonely Are the Brave (1962) (Drama|Western)
Big Carnival, The (1951) (Drama)
Hamlet (1990) (Drama)
Blood Simple (1984) (Drama|Film-Noir)
Romeo and Juliet (1968) (Drama|Romance)
Gold Rush, The (1925) (Comedy)
Network (1976) (Comedy|Drama)
Hook (1991) (Adventure|Fantasy)
Inherit the Wind (1960) (Drama)
Modern Times (1936) (Comedy)
Good Morning, Vietnam (1987) (Comedy|Drama|War)
Thelma & Louise (1991) (Action|Drama)
Erin Brockovich (2000) (Drama)
Night to Remember, A (1958) (Action|Drama)
Muppets Take Manhattan, The (1984) (Children's|Comedy)
Muppet Movie, The (1979) (Children's|Comedy)
City Lights (1931) (Comedy|Drama|Romance)
League of Their Own, A (1992) (Comedy|Drama)
```

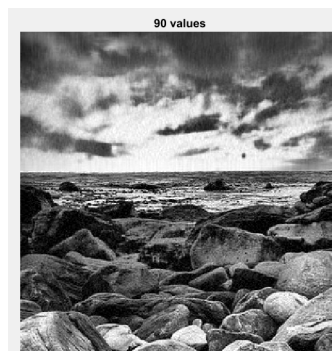
شکل ۱۷: سوال پنجم: فیلم‌های پیشنهادی به کاربر نمونه با نرم اقلیدسی

به خوبی می‌بینیم که اگر مقادیر تکین را تا ۱۰۰ عدد کاهش دهیم، تغییرات تصویر خیلی زیاد نیست و قابل قبول است. اما برای تصاویر با مقادیر تکین کمتر از ۱۰۰ کیفیت کم می‌شود و اطلاعات زیادی از دست می‌رود و تصویر خیلی کم حجم می‌شود.

در نهایت برای ۲۵۰ مقدار تکین ۱۷ درصد کاهش حجم، برای ۲۰۰ مقدار تکین ۳۳ درصد کاهش، برای ۱۵۰ مقدار تکین ۵۰ درصد کاهش، برای ۱۰۰ مقدار ۶۶ درصد کاهش، برای ۵۰ مقدار ۸۳ درصد کاهش حجم داریم.

ه.

حال به ازای فقط ۳۰ درصد از بزرگترین مقادیر تکین، تصویر را رسم می‌کنیم. تصویر ۷۰ درصد کاهش حجم پیدا می‌کند.



شکل ۱۶: سوال چهارم: تصویر به ازای ۳۰ درصد بزرگترین مقادیر تکین

۶. سوال پنجم

آ.

ابتدا در فایل کد `Q5.py` در خط ۴، امتیازهایی که به فیلم‌ها داده شده‌اند را بارگزاری می‌کنیم و سپس در خطوط ۱۹ تا ۲۴ ماتریس امتیازها را تشکیل می‌دهیم. بعد از آن، در خطوط ۲۸ تا ۴۴، ۲۰ فیلم برتر را بر اساس تعداد آرا و سپس میانگین آرا، انتخاب می‌کنیم.

ب.

در خطوط ۴۸ تا ۵۶، کاربرانی که به تمام ۲۰ فیلم برتر رای داده‌اند را انتخاب می‌کنیم و در خطوط ۶۱ تا ۷۴ اطلاعات کاربر آزمایشی را بارگزاری می‌کنیم. از آنجایی که در بخش‌های بعد، تمامی مقایسه‌ها بین کاربران و کاربر آزمایشی، بر اساس ۲۰ فیلم برتر است، با ماتریس‌های سطری‌ای کار می‌کنیم که درایه‌ی شماره‌ی `i` آن، امتیاز کاربر به امین-`i` فیلم برتر است. این موضوع در تشکیل ماتریس `testuser` از ابتدا در نظر گرفته شده است.

ج.

دو مورد ساده سازی در الگوریتم برنامه‌ی مان انجام می‌دهیم: نخست این که نتیجه‌ی مقایسه‌ی فاصله‌ی اقلیدسی کاربران با کاربر آزمایشی، درست مانند وقتی است که به جای فاصله، از مجذور فاصله استفاده کنیم. برای همین با جذر نگرفتن از رابطه‌ی فاصله‌ی اقلیدسی، به محاسبات برنامه سرعت می‌بخشیم. دوم این که از آنجایی که فقط به نزدیک‌ترین کاربر احتیاج داریم، به جای نگه داشتن فهرست مرتب شده‌ی تمام

```
Suggested user (Pearson's r): 5705
Suggested movies (Pearson's r):
Muppet Movie, The (1979) (Children's|Comedy)
Wayne's World (1992) (Comedy)
Fantasia 2000 (1999) (Animation|Children's|Musical)
Toy Story 2 (1999) (Animation|Children's|Comedy)
Fight Club (1999) (Drama)
Ferris Bueller's Day Off (1986) (Comedy)
Drive Me Crazy (1999) (Comedy|Romance)
Christmas Story, A (1983) (Comedy|Drama)
South Park: Bigger, Longer and Uncut (1999) (Animation|Comedy)
Tarzan (1999) (Animation|Children's)
Austin Powers: The Spy Who Shagged Me (1999) (Comedy)
Matrix, The (1999) (Action|Sci-Fi|Thriller)
Other Sister, The (1999) (Comedy|Drama|Romance)
Office Space (1999) (Comedy|Romance)
She's All That (1999) (Comedy|Romance)
Bug's Life, A (1998) (Animation|Children's|Comedy)
Life Is Beautiful (La Vita è bella) (1997) (Comedy|Drama)
Negotiator, The (1998) (Action|Thriller)
Saving Private Ryan (1998) (Action|Drama|War)
Goonies, The (1985) (Adventure|Children's|Fantasy)
```

شکل ۲۱: سوال پنجم : فیلم های پیشنهادی به ما با ضریب به هم پیوستگی

```
Suggested user (Pearson's r): 5972
Suggested movies (Pearson's r):
Pee-wee's Big Adventure (1985) (Comedy)
No Way Out (1987) (Thriller)
Misery (1998) (Horror)
Bull Durham (1988) (Comedy)
Hoosiers (1986) (Drama)
Dead Calm (1989) (Thriller)
Toy Story 2 (1999) (Animation|Children's|Comedy)
Being John Malkovich (1999) (Comedy)
Ferris Bueller's Day Off (1986) (Comedy)
American Beauty (1999) (Comedy|Drama)
Christmas Story, A (1983) (Comedy|Drama)
Vacation (1983) (Comedy)
Sixth Sense, The (1999) (Thriller)
American Pie (1999) (Comedy)
South Park: Bigger, Longer and Uncut (1999) (Animation|Comedy)
Ideal Husband, An (1999) (Comedy)
```

شکل ۱۸: سوال پنجم : فیلم های پیشنهادی به کاربر نمونه با ضریب به هم پیوستگی

و.

حال به توجه به لیست فیلم های برتر، به آنها امتیاز میدهم.

```
Now it's your turn! How would you rate the following 20 movies?
American Beauty (1999) (Comedy|Drama): 2
Star Wars: Episode IV - A New Hope (1977) (Action|Adventure|Fantasy|Sci-Fi): 5
Star Wars: Episode V - The Empire Strikes Back (1980) (Action|Adventure|Drama|Sci-Fi|War): 1
Star Wars: Episode VI - Return of the Jedi (1983) (Action|Adventure|Romance|Sci-Fi|War): 3
Jurassic Park (1993) (Action|Adventure|Sci-Fi): 4
Saving Private Ryan (1998) (Action|Drama|War): 5
Terminator 2: Judgment Day (1991) (Action|Sci-Fi|Thriller): 1
Matrix, The (1999) (Action|Sci-Fi|Thriller): 2
Back to the Future (1985) (Comedy|Sci-Fi): 3
Silence of the Lambs, The (1991) (Drama|Thriller): 5
Men in Black (1997) (Action|Adventure|Comedy|Sci-Fi): 4
Raiders of the Lost Ark (1981) (Action|Adventure): 3
Fargo (1996) (Crime|Drama|Thriller): 2
Sixth Sense, The (1999) (Thriller): 1
Braveheart (1995) (Action|Drama|War): 1
Shakespeare in Love (1998) (Comedy|Romance): 4
Princess Bride, The (1987) (Action|Adventure|Comedy|Romance): 5
Schindler's List (1993) (Drama|War): 3
L.A. Confidential (1997) (Crime|Film-Noir|Mystery|Thriller): 1
Groundhog Day (1993) (Comedy|Romance): 2
```

شکل ۱۹: سوال پنجم : امتیاز دهی ما به فیلم های نمونه

ز.

با کمک توابعی که در مراحل قبل نوشتیم، از طریق امتیازهایی که به فیلم ها داده ایم، فیلم های توصیه شده به ما را در دو تصویر زیر مشاهده میکنیم.

```
Suggested user (Euclidean distance): 4896
Suggested movies (Euclidean distance):
Running Man, The (1987) (Action|Adventure|Sci-Fi)
Hollywood Knights, The (1980) (Comedy)
Animal House (1978) (Comedy)
Bull Durham (1988) (Comedy)
Fast Times at Ridgemont High (1982) (Comedy)
Longest Day, The (1962) (Action|Drama|War)
Christmas Story, A (1983) (Comedy|Drama)
Airplane! (1980) (Comedy)
American Pie (1999) (Comedy)
Bug's Life, A (1998) (Animation|Children's|Comedy)
Sleeping Beauty (1959) (Animation|Children's|Musical)
Saving Private Ryan (1998) (Action|Drama|War)
Out of Africa (1985) (Drama|Romance)
When Harry Met Sally... (1989) (Comedy|Romance)
Butch Cassidy and the Sundance Kid (1969) (Action|Comedy|Western)
M*A*S*H (1970) (Comedy|War)
This Is Spinal Tap (1984) (Comedy|Drama|Musical)
Young Frankenstein (1974) (Comedy|Horror)
Cool Hand Luke (1967) (Comedy|Drama)
Glory (1989) (Action|Drama|War)
```

شکل ۲۰: سوال پنجم : فیلم های پیشنهادی به ما با نرم اقلیدسی