به نام خدا

تمرین کامپیوتری اول بهینه سازی محدب

دکتر یاسایی

مبین خطیب

۹۹۱۰۶۱۱۴

۱ مقادیر ویژه و بردار های ویژه

۱.۱:

درباره ی این الگوریتم تحقیق کنید و آن را شرح دهید.

ایده اصلی ،انجام تجزیه ،QR نوشتن ماتریس به عنوان حاصل ضرب یک ماتریس متعامد و یک ماتریس مثلثی بالا، ضرب کردن فاکتورها به ترتیب معکوس و تکرار است.

این الگوریتم در واقع روشی برای محاسبه مقدارویژه و بردار ویژه یک ماتریس است برای انجام این کار در مرحله امk که k از صفر شروع میشود خواهیم داشت:$A_k = Q_k R_k$ که در آن Q_kیک ماتریس متعامد است وR_kیک ماتریس بالا مثلثی است سپس$A_{k+1} = R_k Q_k$را تشکیل میدهیم

$A_{k+1} = R_k Q_k = Q_k^{-1} Q_k R_k Q_k = Q_k^{-1} A_k Q_k = Q_k^T A_k Q_k$ در انتها با تکرار محاسبات پس از k مرحلهA_kبازگردانده میشود و از آن مقدار ویژه و بردار ویژه اش را محاسبه میکنیم.

۱.۲.

```
In [3]: # 1.2
        # A : square random matrix with size n
        n = 3
        A = np.random.rand(n, n)
        def QR_eignvalue_eignvector(A, iterations=5000):
            Ak = np.copy(A)
            n = A.shape[0]
            QQ = np.eye(n)
            for k in range(iterations):
                Q, R = np.linalg.qr(Ak)
                Ak = np.dot(R,Q)
                QQ = np.dot(QQ,Q)
            return Ak, QQ
        w,v=eig(QR_eignvalue_eignvector(A))
        print('E-value:', (w))
        print('E-vector', (v))

        E-value: [[ 1.61437946+0.j         -0.45696995+0.j          0.12600377+0.j        ]
         [-0.66676511+0.74526793j -0.66676511-0.74526793j  1.        +0.j        ]]
        E-vector [[[ 1.        +0.j          0.16379051+0.j
           -0.45892119+0.j        ]
          [ 0.        +0.j          0.98649514+0.j
           -0.55575318+0.j        ]
          [ 0.        +0.j          0.        +0.j
            0.69320253+0.j        ]]

         [[-0.06554964-0.64315076j -0.06554964+0.64315076j
           -0.40511812+0.j        ]
          [ 0.68929106+0.j          0.68929106-0.j
           -0.22305979+0.j        ]
          [ 0.14346107-0.29386579j  0.14346107+0.29386579j
            0.88663614+0.j        ]]]
```

۱.۳: درباره ی محدودیت های این الگوریتم و الگوریتم های تقریبی دیگر برای تخمین مقادیر و بردارهای ویژه تحقیق کنید.

هر مرحله تکرار نیاز به محاسبه QR factorization یک ماتریس کامل n × n دارد، یعنی هر مرحله تکرار منفرد دارای کامپلکسیتی $O(n۳)$است. حتی اگر تعداد مراحل را متناسب با n فرض کنیم، یک کامپلکسیتی$O(n۴)$دریافت می کنیم. در واقع می تواند کند باشد و ممکن است برای برخی از ماتریس ها همگرا نباشد، به ویژه برای ماتریس هایی که به خوبی شرطی نشده اند یا دارای مقادیر ویژه متعدد با اندازه های مشابه هستند. محدودیت اصلی این الگوریتم این است که در مرحله اول که در حال حاضر مرحله اول معمولاً پر کردن کامل در ماتریس‌های تنک کلی ایجاد می‌کند. بنابراین نمی توان آن را برای ماتریس های تنک بزرگ، صرفاً به دلیل نیاز به حافظه بیش از حد، اعمال کرد. از سوی دیگر، الگوریتم QR همه مقادیر ویژه (و در نهایت بردارهای ویژه) را محاسبه می کند که به ندرت در محاسبات ماتریس تنک لازم میشود. بعضی از الگوریتم های مشابه QR algorithm:

Arnoldi iteration:

این الگوریتم برای محاسبه مبنای جزئی برای زیرفضای کریلوف یک ماتریس استفاده می شود و اغلب همراه با الگوریتم QR استفاده می شود.

Lanczos iteration:

مانند Arnoldi iteration، Lanczos iteration برای محاسبه مبنای جزئی برای زیرفضای کریلوف یک ماتریس استفاده می‌شود

:Implicitly Restarted Arnoldi Method (IRAM)

این نوعی از Arnoldi iteration است که از روش راه‌اندازی مجدد برای بهبود نرخ هم‌گرایی استفاده می‌کند.

:Jacobi-Davidson

این الگوریتم برای محاسبه چند بردار ویژه و مقادیر ویژه یک ماتریس بزرگ و پراکنده استفاده می شود.

:Power method

یک روش تکراری برای یافتن بردار ویژه غالب یک ماتریس است.

:Inverse iteration

این اصلاح روش توان است که برای یافتن بردار ویژه مربوط به یک مقدار ویژه خاص استفاده می شود.

:Rayleigh quotient iteration

بسط روش توان است که برای یافتن بردار ویژه مربوط به نزدیکترین مقدار ویژه به مقدار هدف معین استفاده می شود.

مشکلات کلی این روش ها مانند الگوریتم QR مسئله همگرایی،حافظه مورد نیاز، دقت و مسئله کامپلکسیتی است.

۲ SVD و پردازش تصویر

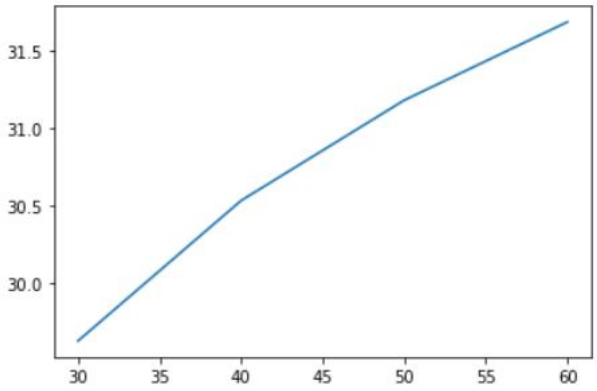
۲.۱: PSNR: معیاری مخفف Peak to noise signal ratio و معیاری برای سنجش کیفیت فشرده سازی است.بلوک PSNR حداکثر نسبت سیگنال به نویز را در دسی بل بین دو تصویر محاسبه می کند. این نسبت به عنوان یک اندازه گیری کیفیت بین تصویر اصلی و فشرده استفاده می شود. هر چه PSNR بالاتر باشد، کیفیت تصویر فشرده شده یا بازسازی شده بهتر است.مقادیر معمولی برای PSNR در فشرده‌سازی تصویر و ویدیو با اتلاف بین ۳۰ تا ۵۰ دسی‌بل است، مشروط بر اینکه عمق بیت ۸ بیت باشد، جایی که بالاتر بهتر است. کیفیت پردازش تصاویر ۱۲ بیتی زمانی بالا در نظر گرفته می شود که مقدار PSNR از ۶۰ دسی بل بالاتر باشد. بهبود تصویر یا بهبود کیفیت بصری یک تصویر دیجیتال می تواند ذهنی باشد. گفتن اینکه یک روش تصویری با کیفیت بهتر ارائه می دهد، می تواند از فردی به فرد دیگر متفاوت باشد. به همین دلیل، لازم است اقدامات کمی/تجربی برای مقایسه اثرات الگوریتم‌های بهبود تصویر بر کیفیت تصویر ایجاد شود.

با استفاده از مجموعه ای از تصاویر آزمایشی، الگوریتم های مختلف بهبود تصویر را می توان به طور سیستماتیک مقایسه کرد تا مشخص شود آیا یک الگوریتم خاص نتایج بهتری ایجاد می کند یا خیر. معیار مورد بررسی نسبت پیک سیگنال به نویز است. اگر بتوانیم نشان دهیم که یک الگوریتم یا مجموعه‌ای از الگوریتم‌ها می‌توانند یک تصویر شناخته‌شده تخریب‌شده را تقویت کنند تا شباهت بیشتری به تصویر اصلی داشته باشند، آنگاه می‌توانیم با دقت بیشتری نتیجه بگیریم که الگوریتم بهتری است.

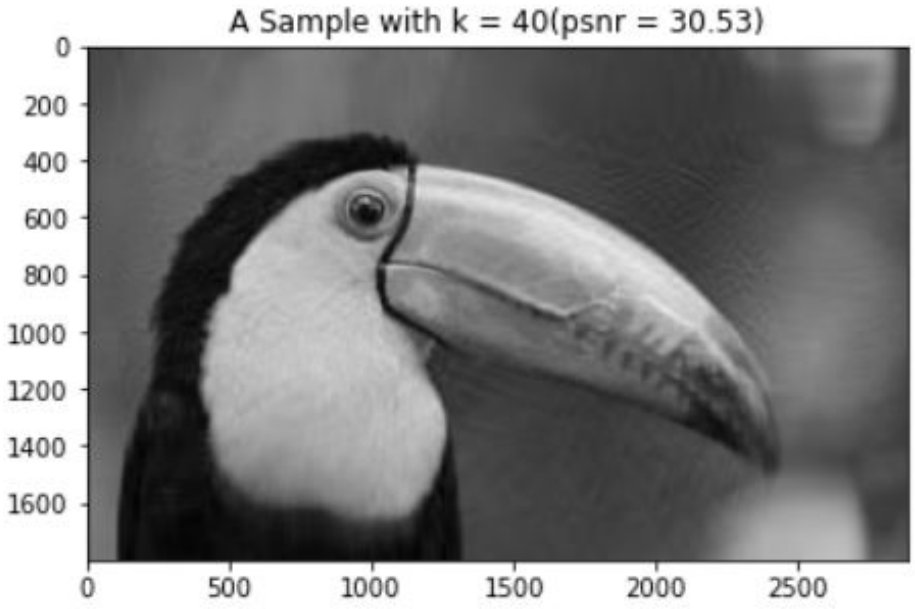
با دادن چند نمونه و محاسبه psnr برای داده های با رنک ۳۰تا ۷۰ و با استپ ده تایی نمودار psnr و همچنین دو سمپل با رنک ۴۰ و رنک ۶۰ رسم کرده ایم در نهایت میبینیم که برای رنک ۶۰ وضوح تصاویر بهتر است طبیعتا همین طور است چون فشرده سازی کمتری در تصویر رخ داده و psnr برای رنک ۶۰ بیشتر است

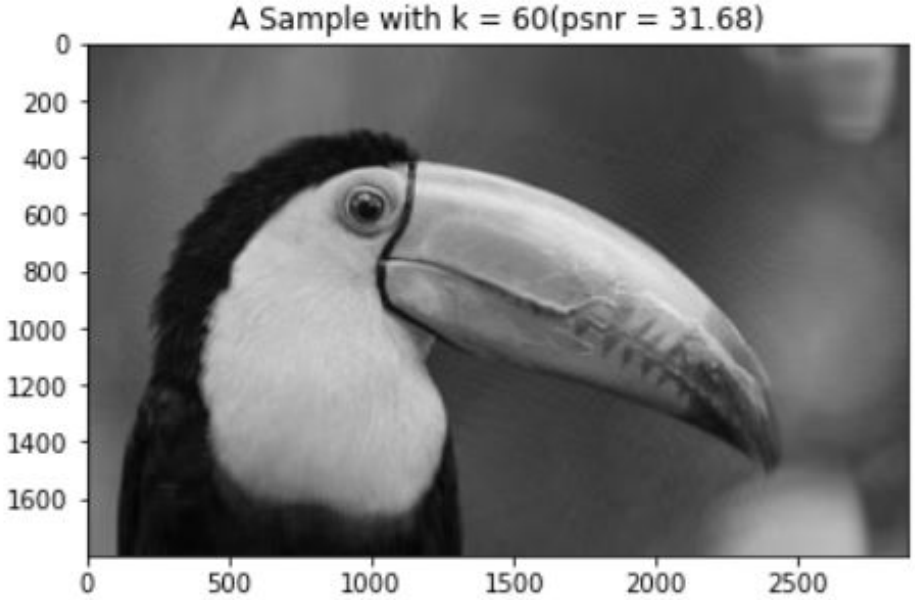
نمودار :psnr

```
[29.62911717241382, 30.53378956349421, 31.18050213280293, 31.684829519493274]
```



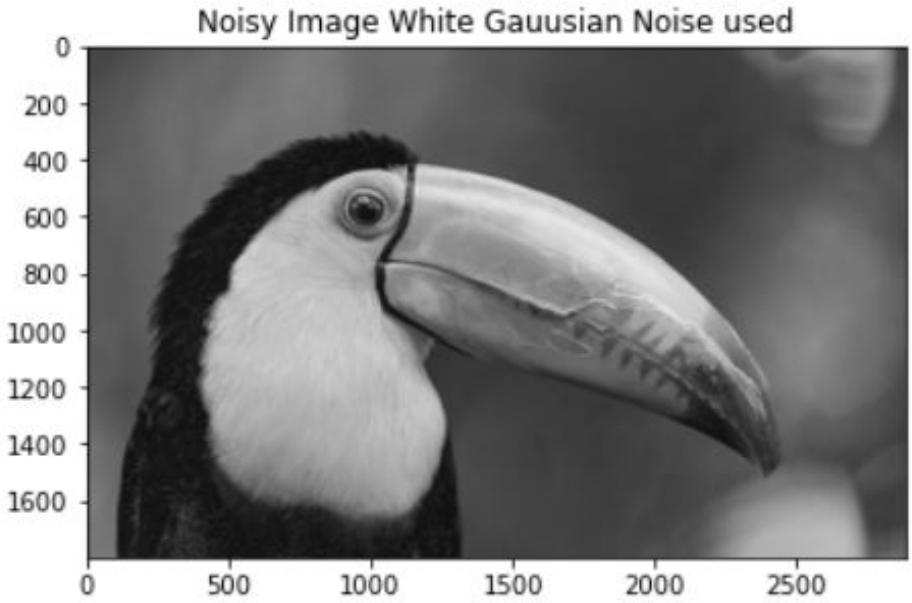
و حال شکل نمونه ها را با هم میبینیم:





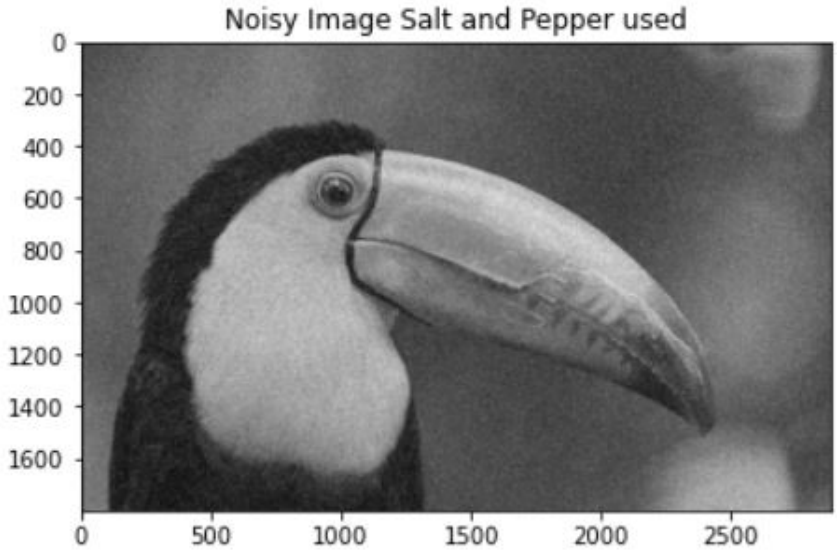
۲.۲: نویز گاوسی به وجود آمده همراه با مقدار psnr آن:

```
32.26295682186223
```



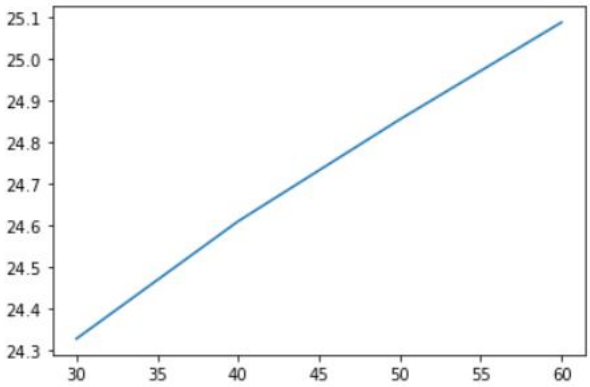
نویز نمک و فلفل به وجود آمده همراه با مقدار psnr آن:

```
35.020034962282374
```

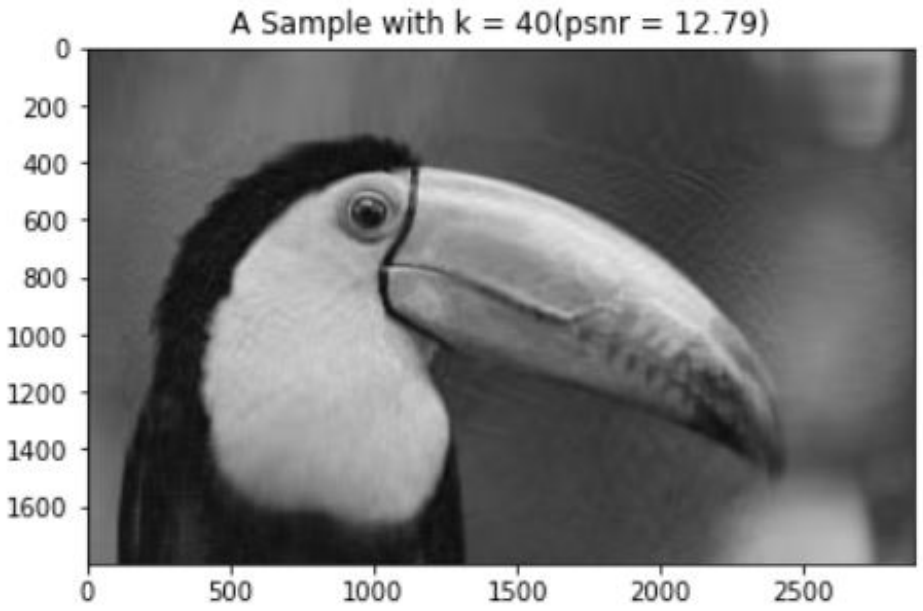


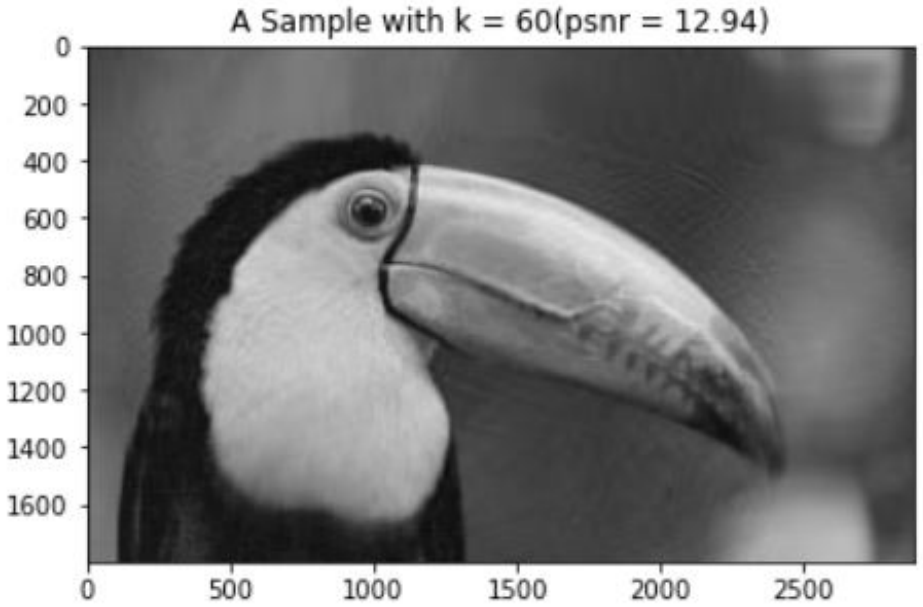
نمودار psnr برای نویز گاوسی به ازای k های مختلف:

```
[24.326455013556707, 24.60885405812776, 24.853748314718665, 25.08719766428547]
```



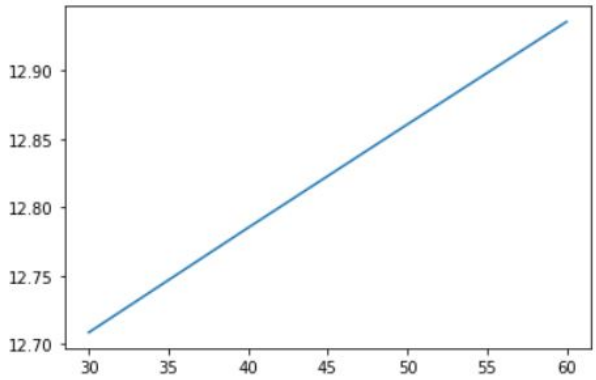
مثال هایی برای نویز نمک و فلفل:





نمودار psnr به ازای k های مختلف برای نویز نمک و فلفل:

```
[12.70842554524815, 12.785006141592714, 12.86057793381115, 12.935684655413947]
```



با دادن چند نمونه و محاسبه psnr برای داده های با رنک ۳۰تا ۷۰ و با استپ ده تایی نمودار psnr و همچنین دو سمپل با رنک ۴۰ و رنک ۶۰ رسم کرده ایم در نهایت میبینیم که برای رنک ۶۰ وضوح تصاویر بهتر است طبیعتا همین طور است چون فشرده سازی کمتری در تصویر رخ داده و psnr برای رنک ۶۰ بیشتر است(هم برای نویز گاوسی و هم برای نویز نمک و فلفل) در نهایت میبینیم که برای رنک ۶۰ وضوح تصاویر بهتر است طبیعتا همین طور است چون فشرده سازی کمتری در تصویر رخ داده و برایpsnr رنک ۶۰ بیشتر است این روش به نظر با توجه به تصاویر مشاهده شده از نمونه های دو نویز برای نویز گاوسی بهتر است و وضوح بیشتری در تصاویر گاوسی نسبت به تصاویر دارای نویز نمک و فلفل مشاهده میشود

۳ کاهش بعد داده ها به روش PCA

:۳.۱

- $PCA \equiv SVD(Cov(X)) = SVD(XX^T/(n-۱))$
- $cov(X_i, X_j) = E[(X_i - \bar{X}_i)(X_j - \bar{X}_j)^T]$
- $cov(X) = E[(X - \bar{X})(X - \bar{X})^T] = E[(\tilde{X})(\tilde{X})^T]$

که در واقع این کواریانس ماتریس است که میزان وابستگی فیچرها نسبت به همدیگر را می گوید X کل دیتاست ماست (در واقع شامل همه ویژگی هاست)،$\bar{X}$نیز میانگین ردیفی است.

U_{m*r} : یک ماتریس متعامد حاوی بردارهای ویژهXX^Tاست $\Sigma r * r$: یک ماتریس مورب حاوی ریشه های مربع مقادیر ویژه XX^T است

$V^T n * r$: یک ماتریس متعامد حاوی بردارهای ویژه X^TXاست

۳.۲: در PCA با استفاده از SVD ، ماتریس داده ورودی X به سه ماتریس تجزیه می شود:$X = UV^T$که در آن U و V ماتریس های متعامد هستند و σ یک ماتریس مورب از مقادیر منفرد است.

ستون‌های U مربوط به بردارهای منفرد سمت چپ ،X و ستون‌های V با بردارهای منفرد راست X مطابقت دارند. مقادیر منفرد در $diag(\sigma)$ نشان‌دهنده اهمیت نسبی هر یک از بردارهای منفرد است.

ستون‌های U یک مبنای متعارف برای فضای ستون X تشکیل می‌دهند، که فضایی است که توسط ستون‌های X پوشانده شده است. این بدان معنی است که هر ستون X را می‌توان به صورت ترکیب خطی از ستون‌های U بیان کرد و ستون‌ها از U به صورت مستقل خطی هستند.

وقتی PCA را انجام می‌دهیم، می‌خواهیم مجموعه جدیدی از متغیرها را پیدا کنیم که مهم‌ترین اطلاعات را در داده‌های اصلی ثبت می‌کنند. این متغیرهای جدید اجزای اصلی هستند که ترکیب خطی متغیرهای اصلی هستند. جزء

اصلی اول ترکیب خطی متغیرهای اصلی است که بیشترین واریانس را دارد، جزء اصلی دوم ترکیب خطی است که دومین واریانس بزرگ را دارد و غیره.

ستون‌های U بردارهای ویژه ماتریس کوواریانس X هستند و مقادیر منفرد مربوطه با ریشه‌های مربع مقادیر ویژه متناظر متناسب هستند. بنابراین، ستون‌های U مجموعه‌ای از بردارهای پایه متعارف را برای فضای پوشانده شده توسط مؤلفه‌های اصلی X ارائه می‌کنند.

به عبارت دیگر، ستون‌های U را می‌توان به‌عنوان پایه‌ای برای فضای مقصد (فضای اجزای اصلی) استفاده کرد، زیرا جهت‌های حداکثر واریانس را در داده‌های اصلی نشان می‌دهند و بنابراین مجموعه‌ای از بردارهای پایه را ارائه می‌دهند که می‌توان از آنها استفاده کرد. برای ساخت ترکیب های خطی از متغیرهای اصلی که مهم ترین اطلاعات را در داده ها جمع آوری می کند.

ماتریس U ماتریسی است که ستون های آن بردارهای منفرد سمت چپ هستند، سیگما یک ماتریس مورب است که عناصر مورب آن مقادیر منفرد هستند، و VT ماتریسی است که ستون های آن بردارهای منفرد سمت راست هستند.

اجزای اصلی با گرفتن حاصل ضرب نقطه ای ماتریس داده X با ماتریس VT محاسبه می شوند. این معادل نمایش داده ها بر روی بردارهای منفرد سمت راست است که جهات بیشترین واریانس را در داده ها تعریف می کنند.

ماتریس U نشان دهنده رابطه بین داده های اصلی و اجزای اصلی است. هر ردیف U نشان دهنده وزنی است که متغیر مربوطه در داده های اصلی در هر جزء اصلی دارد.

ماتریس سیگما حاوی مقادیر منفرد است که نشان دهنده مقدار واریانس توضیح داده شده توسط هر جزء اصلی است. مقادیر منفرد به ترتیب کاهشی در امتداد قطر ماتریس مرتب شده اند.

ماتریس VT خود مؤلفه های اصلی را نشان می دهد. هر ردیف از VT یک جزء اصلی را نشان می دهد که ترکیبی خطی از متغیرهای اصلی است. اجزای اصلی متعامد با یکدیگر هستند و بر اساس مقدار واریانسی که در داده ها توضیح می دهند مرتب می شوند.

به طور کلی، SVD ماتریس داده در PCA برای یافتن مؤلفه‌های اصلی استفاده می‌شود، که ترکیبات خطی متغیرهای اصلی هستند که حداکثر مقدار واریانس در داده‌ها را ثبت می‌کنند. ماتریس‌های Sigma U، و VT برای محاسبه مؤلفه‌های اصلی و درک رابطه بین داده‌های اصلی و مؤلفه‌های اصلی استفاده می‌شوند.

:۳.۳

با توجه به یک ماتریس A با ابعاد$m * n$ که در آن$m >= n$ تجزیه مقدار منفرد آن (SVD) با$A = U\Sigma V^T$داده می شود، که در آن U یک ماتریس متعامد$m * m$است، Σ یک ماتریس مورب$n * n$با ورودی های حقیقی غیر منفی است. ، وV^Tیک ماتریس متعامد$n * n$ است.

برای کاهش بعد A از n به l جایی که$l < n$می‌توانیم یک SVD کوتاه شده انجام دهیم، که در آن فقط l ستون اول U ردیف و ستون‌های اول از Σ ، و l ردیف اولV^T را نگه می‌داریم. در اینجا اگر فرض کنیم U_lاولین ستون برای U هستند و همچنین اولین l ردیف و ستون اول برای Σ ، وV_l^Tاولین l ردیفV^Tباشد. سپس، SVD کوتاه شده A توسط$A \approx U_l \Sigma_l V_l^T$داده می شود.

تقریب$A \approx U_l \Sigma_l V_l^T$بهترین تقریب رتبه‌ـl از نظر نرم فروبنیوس است. همچنین می توان نشان داد که این تقریب تفاوت بین A و هر ماتریس رتبه‌ـl ماتریس B را از نظر نرم،$||A - B_۲||$به حداقل می رساند.

حال SVD کوتاه شده را می توان برای فشرده سازی داده ها استفاده کرد، جایی که ماتریس اصلی A با تقریب بعدی پایین تر آن$U_l \Sigma_l V_l^T$نشان داده می شود.

:۳.۴

یکی از الگوریتم‌هایی که مبتنی بر روش‌های خطی مانند PCA (تحلیل مؤلفه اصلی) است، آنالیز تشخیص خطی (LDA) است.

یک الگوریتم یادگیری نظارت شده است که برای کارهای طبقه بندی استفاده می شود. ترکیبی خطی از ویژگی‌ها را پیدا می‌کند که بهترین کلاس‌ها را در داده‌ها جدا می‌کند. مشابه LDA PCA، داده ها را روی فضایی با ابعاد پایین تر پخش می کند. با این حال، بر خلاف برچسب‌های کلاس نقاط داده را هنگام یافتن طرحی که بهترین کلاس‌ها را از هم جدا می‌کند، در نظر می‌گیرد.

اغلب در تشخیص تصویر و بینایی ماشین و همچنین در پردازش زبان طبیعی و وظایف طبقه بندی متن استفاده می شود.

یک مدل آماری است که برای مدل‌سازی موضوعی در پردازش زبان طبیعی (NLP) استفاده می‌شود. این یک مدل احتمالی مولد است که امکان کشف موضوعات پنهان در مجموعه بزرگی از متن را فراهم می کند.

ایده اصلی پشت LDA این است که اسناد به صورت ترکیبی از موضوعات نمایش داده می شوند و هر موضوع به صورت توزیع بر روی کلمات نمایش داده می شود. به عبارت دیگر، LDA فرض می‌کند که اسناد با انتخاب توزیعی از موضوعات، و سپس تولید کلمات از هر یک از آن موضوعات ایجاد می‌شوند.

مدل LDA دارای دو جزء اصلی است: مجموعه ای از توزیع های موضوعی، و مجموعه ای از توزیع های کلمه. توزیع موضوع برای هر سند به عنوان بردار احتمالات نشان داده می شود، جایی که هر عنصر نشان دهنده احتمال وجود یک موضوع خاص در سند است. به طور مشابه، توزیع کلمه برای هر موضوع به عنوان بردار احتمالات نشان داده می شود، جایی که هر عنصر نشان دهنده احتمال مرتبط شدن یک کلمه خاص با موضوع است.

معمولاً برای کارهایی مانند طبقه بندی اسناد، بازیابی اطلاعات و سیستم های توصیه استفاده می شود. این به طور گسترده ای در حوزه های مختلف، مانند تجزیه و تحلیل رسانه های اجتماعی، مراقبت های بهداشتی و مالی استفاده شده است.

برای استفاده از LDA، باید مجموعه ای از داده های متنی را به عنوان ورودی ارائه کرد. سپس مدل با به‌روزرسانی مکرر آنها بر اساس داده‌های متنی مشاهده‌شده، توزیع‌های موضوع و کلمات را یاد می‌گیرد. هنگامی که مدل ترین شد، می توان از آن برای استنتاج توزیع موضوع اسناد جدید یا توصیه اسناد مرتبط به یک پرس و جو استفاده کرد.

الگوریتم های مختلفی برای آموزش مدل های LDA وجود دارد، مانند الگوریتم نمونه گیری گیبس و الگوریتم استنتاج تغییرات. انتخاب الگوریتم به الزامات و محدودیت های خاص برنامه بستگی دارد.

LDA چگونه کار می کند:

داده های ورودی: اولین مرحله ارائه مجموعه ای از داده های متنی به عنوان ورودی مدل LDA است.

پیش پردازش: سپس داده های متن با حذف کلمات توقف، علائم نگارشی و سایر عناصر نامربوط یا پر سر و صدا پیش پردازش می شوند.

واژگان: در مرحله بعد، مدل LDA یک واژگان را با شناسایی کلمات منحصر به فرد در تمام اسناد موجود در مجموعه ایجاد می کند.

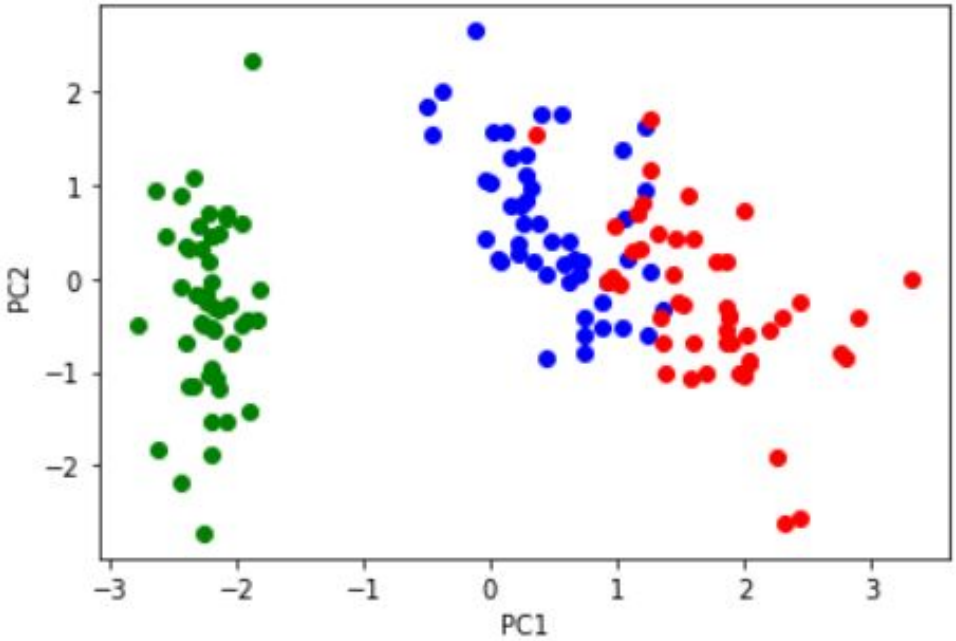
مقداردهی اولیه: مدل LDA توزیع موضوع و کلمه را به صورت تصادفی مقداردهی می کند.

فرآیند تکراری: مدل LDA به صورت تکراری موضوعاتی را به هر کلمه در هر سند، بر اساس موضوع فعلی و توزیع کلمات اختصاص می‌دهد. این مدل احتمال مرتبط شدن هر موضوع با یک کلمه داده شده را محاسبه می کند و کلمه را به محتمل ترین موضوع اختصاص می دهد.

به روز رسانی توزیع موضوع و کلمه: پس از هر تکرار، مدل LDA توزیع موضوع و کلمه را بر اساس داده های متن مشاهده شده به روز می کند.

تکرار: روند تکراری تا زمانی که مدل همگرا شود یا تا زمانی که یک معیار توقف از پیش تعریف شده برآورده شود ادامه می یابد.

خروجی: مدل LDA اون چیز ترین شده و توزیع کلمه رو خروجی می دهد که می تواند برای تجزیه و تحلیل و طبقه بندی داده های متن جدید استفاده شود.

به طور خلاصه، LDA با این فرض کار می کند که هر سند ترکیبی از موضوعات است و هر موضوع توزیعی از کلمات است. این مدل با اختصاص دادن موضوعات به کلمات در هر سند و به‌روزرسانی توزیع‌ها بر اساس داده‌های متن مشاهده‌شده، از یک فرآیند تکراری برای یادگیری موضوع و توزیع کلمات زیربنایی استفاده می‌کند. خروجی LDA می تواند برای کارهای مختلفی مانند طبقه بندی اسناد و بازیابی اطلاعات استفاده شود.

:۳.۵