

به نام خدا



دانشگاه ارومیه

دانشکده کامپیوتر

"فشرده سازی تصویر با تقریب رتبه پایین"

جبر خطی کاربردی

استاد : دکتر صباحی

گردآورنده : سلوی بهنام اصل

۹۷۴۴۲۱۰۰۸

نیمسال اول ۱۴۰۱-۱۴۰۲

امروزه، تجزیه مقادیر منفرد در بسیاری از شاخه‌های علوم و نجوم کاربرد دارد. این روش، در یادگیری ماشین و آمار توصیفی و مدل‌سازی آماری نیز بسیار مورد استفاده قرار می‌گیرد.

کاربردهای ریاضی SVD شامل محاسبه شبه معکوس، تقریب ماتریس و تعیین رتبه، دامنه و فضای تهی یک ماتریس است. SVD همچنین در کلیه زمینه‌های علمی و مهندسی بسیار مفید است مانند پردازش سیگنال، حداقل مربعات مناسب داده‌ها و کنترل فرایند.

البته که محاسبه SVD برای یک ماتریس دارای پیچیدگی زمانی خوبی نیست ولی به هر حال این روش یکی از روش‌های شناخته شده در کاهش ویژگی‌ها در داده‌کاوی می‌باشد و کاربرد دیگر آن که فشرده سازی داده است در ادامه مورد بحث قرار خواهد گرفت. در زبان‌های برنامه‌نویسی مانند Python و R می‌توان کتابخانه‌های مختلفی را پیدا کرد که این عملیات را به سادگی برای شما انجام می‌دهند.

به طور خلاصه می‌توان گفت که تجزیه مقادیر منفرد روشی است که یک ماتریس را به سه ماتریس دیگر تجزیه می‌کند. برای مثال، ماتریس  $A$  را می‌توان به صورت زیر نوشت:

$$A = USV^T$$

که در آن:

- $A$  یک ماتریس  $m \times n$
- $U$  یک ماتریس متعامد  $m \times m$
- $S$  یک ماتریس قطری  $m \times n$
- $V$  یک ماتریس متعامد  $n \times n$  است.

در رشته علوم کامپیوتر یکی از چالش‌های متداول در یادگیری ماشین، وجود چند صد متغیر است، در حالی که بسیاری از الگوریتم‌های یادگیری ماشین، اگر با تعدادی بیش از یک مقدار مشخص متغیر کار کنند، با شکست مواجه می‌شوند. این موضوع، استفاده از تجزیه مقادیر تکین را برای کاهش متغیر در یادگیری ماشین ضروری می‌کند.

اگر تعداد و انواع مناسبی از ویژگی‌ها برای حل یک مساله خاص به الگوریتم‌های یادگیری ماشین داده شود، این الگوریتم‌ها به خوبی کار می‌کنند. اما، در صورتی که تعداد ویژگی‌ها (متغیرها) بسیار زیاد باشد، اغلب الگوریتم‌های یادگیری ماشین در حل مسئله دچار مشکل می‌شوند، زیرا با مسئله داده‌های ابعاد بالا (High Dimensional Data) مواجه خواهیم بود. در اینجا است که بحث کاهش ابعاد (Dimensionality Reduction) مطرح می‌شود.

$$a_{ij} \approx \sum_{k=1}^p u_{ik} s_k v_{jk}$$

در معادله بالا یک SVD با کاهش تعداد مقادیر تکین می‌تواند یک ماتریس را بسیار دقیق تقریب بزند. از این ویژگی می‌توان برای فشرده‌سازی داده‌ها با فرم‌های کوتاه شده  $U$ ,  $S$ ,  $V$  به جای  $A$  و برای کاهش متغیر از جایگزینی  $A$  با  $U$  استفاده کرد. البته در پایان، بر اساس معادله  $U = AVS^{-1}$  باید با ضرب  $S$  و  $V$  نتایج را به دستگاه مختصات اصلی تبدیل کرد.

پیاده سازی فشرده سازی تصویر :

این عملیات در ۵ مرحله با استفاده از کتابخانه های ریاضی پایتون بصورت زیر پیاده سازی شده است :

```

1  # -*- coding: utf-8 -*-
2  ***
3  Created on Mon Jan 16 18:50:05 2023
4
5  @author: Asus
6  ***
7  from matplotlib.image import imread
8  import matplotlib.pyplot as plt
9  import numpy as np
10 plt.rcParams['figure.figsize'] = [16,8]
11
12 A = imread('Toyota_FJ_Cruiser.jpg') #open the image
13 X = np.mean(A,2) #convert RGB to grayscale
14
15 img = plt.imshow(X) #show image
16 img.set_cmap('gray')
17 plt.axis('off')
18 plt.show()
19
20 U, Sigma, VT = np.linalg.svd(X, full_matrices=False)
21 Sigma = np.diag(Sigma)
22
23 j=0
24 for r in (5,20,100):
25     #construct approximate image
26     Xapprox=U[:,0:r] @ Sigma[0:r,0:r] @ VT[0:r,:]
27     plt.figure(j+1)
28     j+=1
29     img=plt.imshow(Xapprox)
30     img.set_cmap('gray')
31     plt.axis('off')
32     plt.title('r='+str(r))
33     plt.show()
34

```

در مرحله اول با ضمیمه کردن کتابخانه های Matplotlib و Numpy شروع میکنیم که به ترتیب از Matplotlib برای نشان دادن و رسم کردن تصاویر فشرده شده و از Numpy برای استفاده از فرمول های جبر خطی مثل SVD و استخراج قطر اصلی (diag) استفاده خواهد شد .

```
6      """
7      from matplotlib.image import imread
8      import matplotlib.pyplot as plt
9      import numpy as np
10     plt.rcParams['figure.figsize'] = [16,8]
11
```

] 1

در اسکریپت بالا، از ویژگی rcParams در ماژول plt و ارسال مقدار figure.figsize به عنوان پارامتر استفاده کرده ایم که فهرستی حاوی عرض و ارتفاع پیش فرض نمودار را برمی گرداند. ایندکس اول عرض و ایندکس دوم ارتفاع است.

در مرحله دوم فایل تصویر را (که باید نام دقیق تصویر دلخواه را به عنوان ورودی به آن داد) میخوانیم و در پارامتر A ذخیره میکنیم .

سپس با تابع Mean میتوانیم از سه عدد رنگ های قرمز ، سبز و آبی RGB هر پیکسل که بین صفر و ۲۵۵ می باشد میانگین بگیریم که بعداً رنگ تصویر را به حالت سفید سیاه دریاوریم و محاسبات آسان تری داشته باشیم و آن را به شکل زیر در ۲ بعد ذخیره می کنیم .

```
11
12     A = imread('Toyota_FJ_Cruiser.jpg')#open the image
13     X = np.mean(A,2);#convert RGB to grayscale
14
```

] 2

در مرحله سوم تصویر پردازش شده را به متغیر img می دهیم . میانگین اعدادی که از تصویر رنگی به دست آوردیم را به رنگ خاکستری متناظر میکنیم تا حالت سفید سیاه اجرا شود . سپس محور های افقی و عمودی تصویر را خاموش میکنیم که نمایش داده نشود و تصویر را نمایش می دهیم .

```
14
15     img = plt.imshow(X)#show image
16     img.set_cmap('gray')
17     plt.axis('off')
18     plt.show()
19
```

] 3

در مرحله چهارم وارد عملیات ریاضی شده و عکس ماتریس شده را به تابع SVD می دهیم تا با فرمول های LinAlg ( Linear Algebra ) بتواند مقادیر  $U$  ,  $S$  ,  $VT$  را حساب کند و آن هارا جایگذاری کند .

```
19
20 U , Sigma , VT = np.linalg.svd(X,full_matrices=False)
21 Sigma=np.diag(Sigma)
22 ] 4
```

برای در نظر گرفتن ماتریس های غیرمربعی حالت دیفالت تابع که True میباشد را به حالت full\_matrices=False تغییر می دهیم . سپس قطر اصلی سیگما را در متغیر بصورت ارایه یک بعدی ذخیره میکنیم .

در مرحله آخر حلقه ای درست می کنیم که به ازای رتبه های دلخواه مان (  $r$  , Rank ) فشرده سازی را پیاده سازی کند که تفاوت تقریب با رتبه های مختلف را متوجه بشویم :

```
22
23 j=0
24 for r in(5,20,100):
25     #construct approximate image
26     Xapprox=U[:, :r] @ Sigma[0:r, :r] @ VT[:, :]
27     plt.figure(j+1)
28     j+=1
29     img=plt.imshow(Xapprox)
30     img.set_cmap( 'gray')
31     plt.axis( 'off')
32     plt.title( 'r='+str(r))
33     plt.show()
34 ] 5
```

برای محاسبه تقریب با توجه به رتبه مد نظر, به تعداد  $r$  ستون اول از  $U$  را جدا کرده , در سیگما به ابعاد  $r \times r$  ضرب کرده و نهایتا به تعداد  $r$  سطر اول از  $V$  Transposed جدا کرده و همگی را در هم ضرب میکنیم و در متغیر  $X$  تقریبی (X approximate) ذخیره میکنیم .

سپس جواب تقریبی مان را بصورت سفید سیاه نمایش میدهیم و رتبه هر کدام را بالای تصویرش به نمایش در می اوریم .

نتایج فشرده سازی تصویر به روش SVD و تقریب رتبه پایین ماتریس را بصورت زیر مشاهده می کنیم :

با Rank=5 :

r=5



با Rank=20 :

r=20



با Rank=100 :



به درستی مشاهده میشود که تنها با نگه داشتن ۲۰ ستون اول  $U$  و ۲۰ سطر اول  $V_t$  میتوان به تقریب خوبی از جواب رسید و تصویری را به ما ارائه میدهد که محتوی آن تا حد خوبی قابل تشخیص است و البته حجم آن نیز به شدت فشرده شده .

در تقریب با رتبه ۱۰۰ نیز تصویر به وضوح مناسبی رسیده که اگر حجم آن را حساب کنیم تنها به اندازه ۱۲٪ از سایز اصلی تصویر حافظه اشغال می کند و توانستیم حجم داده مان را بیش از ۸ برابر فشرده کنیم در حالیکه کاملاً کیفیت قابل قبولی را ارائه می کند .

با این حساب با در نظر گرفتن حجم محدود حافظه های فیزیکی و همینطور هزینه زیاد آنها، فشرده سازی حجم عظیم از تصاویر در دیتابیس ها ، امری اجتناب ناپذیر و مهم است؛ از این رو، فشرده سازی تصاویر اولیه در برخی مواقع مفید بوده و حتی ضروری می باشد و می بایست با توجه با اهمیت موضوع و کیفیت مورد نیاز ، ذخیره و نگهداری شوند که روش مذکور به حد قابل قبولی از عهده چنین کاری بر می آید.