به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس یادگیری ماشین

تمرین پنجم

نام و نام خانوادگی	امیرحسین پورداود
شماره دانشجویی	۸۱۰۱۰۱۱۲۰
تاریخ ارسال گزارش	۱۴۰۲.۱۱.۰۸

فهرست

پاسخ ۱ – پرسش

/ /

Q1/

پاسخ ۲ – پرسش

Q2/

$$J_1(w) = \frac{(\mu_1-\mu_2)^2}{\sigma_1^2+\sigma_2^2} \Rightarrow w^* = (\Sigma_1+\Sigma_2)^{-1}(\mu_1-\mu_2)$$

$$y = w^T x \rightarrow \begin{cases} \tilde{\mu}_1 = w^T\mu_1 \\ \tilde{\mu}_2 = w^T\mu_2 \end{cases}$$

$E\{x\}=0$

$$\tilde{s}_1^2 = \sum_{i=1}^{n} (w^T x_i)(w^T x_i)^T = w^T \underbrace{\left(\sum_{i=1}^{n} x_i x_i^T\right)}_{s_1^2} w = w^T s_1^2 w$$

$$\tilde{s}_2^2 = w^T s_2^2 w$$

$$J_2(w) = \frac{(\tilde{\mu}_1-\tilde{\mu}_2)^2}{\tilde{s}_1^2+\tilde{s}_2^2} = \frac{(w^T\mu_1 - w^T\mu_2)^2}{w^T(s_1^2+s_2^2)w} = \frac{w^T\overbrace{(\mu_1-\mu_2)(\mu_1-\mu_2)^T}^{s_B}w}{w^T\underbrace{(s_1^2+s_2^2)}_{s_W}w}$$

$$\max_w \frac{w^T s_B w}{w^T s_W w} \equiv \max_w w^T s_B w \quad s.t. \; w^T s_W w = K$$

$$\text{Lagrangian} = w^T s_B w - \lambda(w^T s_W w - K)$$

$$\nabla_w L = 2 s_B w - 2\lambda s_W w = 0 \Rightarrow \underbrace{s_W^{-1} s_B}\, w^* = \lambda w^*$$

eigenvalue

w همان بردار ویژه متناظر با $s_W^{-1} s_B$ می باشد.

$$\lambda w = s_W^{-1}(\mu_1-\mu_2)(\mu_1-\mu_2)^T w$$

مقیاس $\quad a a^T u = \alpha a \rightarrow \lambda w = s_W^{-1}\cdot\alpha\cdot(\mu_1-\mu_2)$

$$\rightarrow w = \frac{\alpha}{\lambda} s_W^{-1}(\mu_1-\mu_2) \Rightarrow w^* = (\Sigma_1+\Sigma_2)^{-1}(\mu_1-\mu_2) \; ☑$$

پاسخ ۳ – پرسش

۳-۱. قسمت ۱

تحلیل مولفه اصلی (PCA):

- مشکل: داده های با ابعاد بالا می تواند منجر به افزایش پیچیدگی محاسباتی و نیاز به حافظه در هنگام محاسبه ماتریس کوواریانس شود.

- راه حل: از الگوریتم های کارآمد یا تکنیک های کاهش ابعاد مانند PCA تصادفی برای مدیریت مجموعه داده های بزرگ استفاده میتوان استفاده کرد. علاوه بر این، روش‌های PCA افزایشی را نیز می‌توان برای داده‌های آنلاین یا جریانی استفاده کرد.

تحلیل تشخیص خطی (LDA):

- مشکل: مشابه PCA، LDA در فضاهای با ابعاد بالا به دلیل محاسبه ماتریس های پراکندگی با چالش هایی مواجه است.

- راه حل: روش های منظم سازی را می توان برای تثبیت محاسبات ماتریس های پراکندگی به کار برد. همچنین، تکنیک‌های تجزیه ارزش ویژه را می‌توان برای مدیریت کارآمدتر سناریوهای با ابعاد بالا بهینه کرد.

۳-۲. قسمت ۲

متریک: اطلاعات متقابل (MI) بین داده و برچسب:

بهبودها:

- MI می‌تواند وابستگی‌های غیرخطی بین ویژگی‌ها و برچسب‌ها را ثبت کند و آن را برای انواع خاصی از توزیع‌های داده مناسب‌تر می‌کند.
- ممکن است معیار آموزنده تری از رابطه بین متغیرها در مقایسه با ماتریس کوواریانس ارائه دهد.

معایب:

- از نظر محاسباتی نسبت به روش‌های مبتنی بر کوواریانس سنتی نیازمندتر است.
- حساسیت به انتخاب binning یا گسسته سازی در فرآیند تخمین.

- MI ممکن است به داده های بیشتری برای تخمین دقیق نیاز داشته باشد، به ویژه در فضاهای با ابعاد بالا.

انتخاب متریک به ماهیت داده ها و روابط زیربنایی بین متغیرها بستگی دارد. در حالی که MI ممکن است مزایایی در گرفتن وابستگی های غیر خطی ارائه دهد، با افزایش هزینه های محاسباتی و چالش هایی در تنظیم پارامترها همراه است.

پاسخ ۴ – پرسش

۴-۱. قسمت ۱

$$p(x|z) = \mathcal{N}(x|Wz+\mu, \sigma^2 I) \qquad p(z) = \mathcal{N}(0, I)$$

$$p(x) = \int p(x,z)\,dz \qquad , \quad p(x) \overset{?}{=} \mathcal{N}(x|\mu, WW^T+\sigma^2 I)$$

a)

$$p(x) = \int p(x|z)\,p(z)\,dz$$

$$E[z]=0$$

$$E(x) = E\{Wz+\mu+\varepsilon\} = \mu$$

$$Cov(x) = E\{(x-\mu)(x-\mu)^T\} = E\{(Wz+\cancel{\mu}-\cancel{\mu}+\varepsilon)(Wz+\cancel{\mu}-\cancel{\mu}+\varepsilon)^T\}$$

$$= E\{Wz(Wz)^T\} + E\{\varepsilon\varepsilon^T\}$$

$$= E\{Wzz^TW^T\} + E\{\varepsilon\varepsilon^T\}$$

$$= W\underbrace{E\{zz^T\}}_{I}W^T + \sigma^2 I = WW^T + \sigma^2 I \quad ☑$$

۴-۲. قسمت ۲

b)

$$p(z|x) = \frac{p(x|z)\,p(z)}{p(x)}$$

پاسخ ۵ – پرسش

۵-۱. قسمت ۱

Q5/

$C_1 = \{x_1, x_2, x_4\} \qquad C_2 = \{x_3, x_5\}$

a)

$\mu_i = \frac{1}{n_i}\sum_{x \in C_i} x_i \Rightarrow \mu_1 = [\frac{5}{3}, \frac{2}{3}] \qquad \mu_2 = [\frac{13}{4}, 1]$

$d_1 = 2{,}13$ (1) $\qquad d_1 = 3{,}4$

$d_2 = 1{,}79$ (1) $\qquad d_2 = 3{,}4$

$d_3 = 0{,}68$ (1) $\qquad d_3 = 2{,}02$

$d_4 = 3{,}40 \qquad d_4 = 2{,}02$ (2)

$d_5 = 3{,}59 \qquad d_5 = 2{,}02$ (2)

$C_1 = \{x_1, x_2, x_3\} \qquad C_2 = \{x_4, x_5\}$

$\Rightarrow \mu_1 = [\frac{1}{2}, \frac{2}{3}] \qquad \mu_2 = [5, 1]$

$d = \|x - c_i\|_2$

$d_1 = 1{,}42$ (1) $\qquad d_1 = 5{,}1$

$d_2 = 0{,}83$ (1) $\qquad d_2 = 5{,}1$

$d_3 = 1{,}2$ (1) $\qquad d_3 = 3{,}64$

$d_4 = 4{,}55 \qquad d_4 = 1$ (2)

$d_5 = 4{,}69 \qquad d_5 = 1$ (2)

باتوجه به اینکه داده ها تغییری نکردند کلاس و مراکز ثابت می مانند پس کلاسترینگ بصورت زیر خواهد بود

$C_1 = \{x_1, x_2, x_3\} \qquad C_2 = \{x_4, x_5\}$

$\boxed{1}$

$\Rightarrow \mu_1 = [\frac{1}{2}, \frac{2}{3}] \qquad \mu_2 = [5, 1]$

۵-۲. قسمت ۲

b)

$\mu_i = \frac{1}{n_i}\sum_{x \in C_i} x_i \Rightarrow \mu_1 = [\frac{5}{3}, \frac{2}{3}] \qquad \mu_2 = [\frac{13}{4}, 1]$

$d = \|x - C_i\|_1$

$d_1 = 3$ 1	$d_1 = 4,25$
$d_2 = 2,33$ 1	$d_2 = 4,25$
$d_3 = 0,83$ 1	$d_3 = 2,75$
$d_4 = 4$	$d_4 = 2,75$ 2
$d_5 = 4,67$	$d_5 = 2,75$ 2

$C_1 = \{x_1, x_2, x_3\} \qquad C_2 = \{x_4, x_5\}$

$\Rightarrow \mu_1 = [\frac{1}{2}, \frac{2}{3}] \qquad \mu_2 = [5, 1]$

$d_1 = 1,83$ 1	$d_1 = 6$
$d_2 = 1,17$ 1	$d_2 = 6$
$d_3 = 1,67$ 1	$d_3 = 4,5$
$d_4 = 5,17$	$d_4 = 1$ 2
$d_5 = 5,83$	$d_5 = 1$ 2

باتوجه به اینکه داده ها تغییر نمی کنند و کلاس و مراکز ثابت می مانند پس کلاسترینگ بصورت زیر خواهد بود

$C_1 = \{x_1, x_2, x_3\} \qquad C_2 = \{x_4, x_5\}$

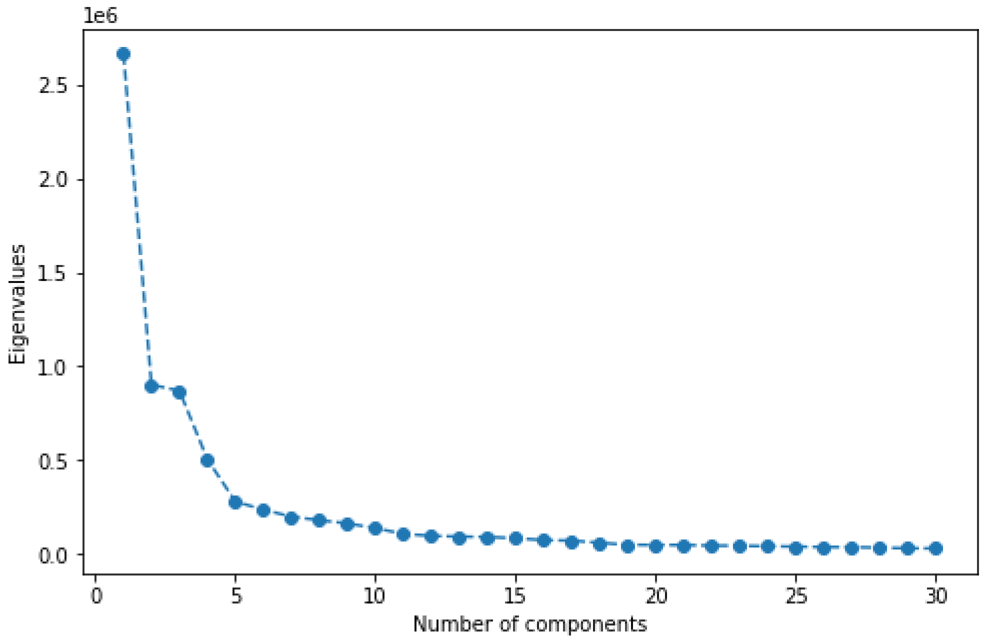
$\boxed{1}$

$\Rightarrow \mu_1 = [\frac{1}{2}, \frac{2}{3}] \qquad \mu_2 = [5, 1]$

پاسخ ۶– شبیه سازی

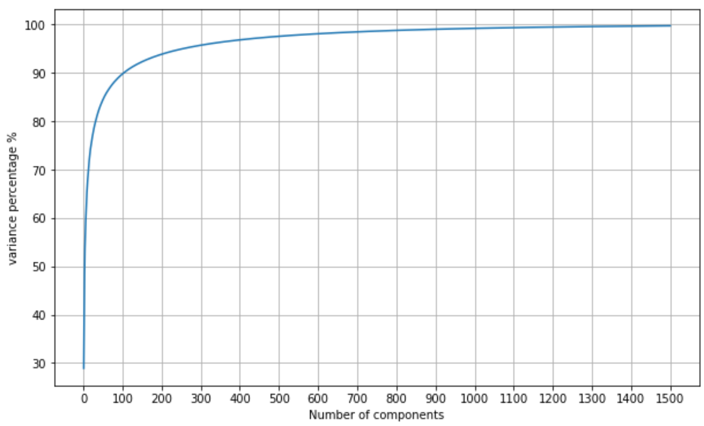
۶-۱. قسمت ۱

مقدار مقادیر ویژه را برای ۳۰ تای اول بصورت زیر رسم میکنیم:



اکنون برای بدست آوردن مقدار بهینه برای تعداد component ها، باید میزان واریانس پوششی در تعداد انتخاب کامپوننت ها دخیل کنیم.

همانطور که میدانیم مقادیر ویژه درصدی از واریانس پوششی را نشان میدهند، به همین منظور ما به ترتیب بزرگی مقادیر ویژه از ابتدا دو به دو بصورت تجمیعی با هم جمع کرده و در انتها به مجموع همه مقادیر ویژه ها تقسیم میکنیم که درصد پوشش واریانس بر اساس تعداد انتخاب کامپوننت ها بدست آید که بصورت زیر قابل نمایش است:

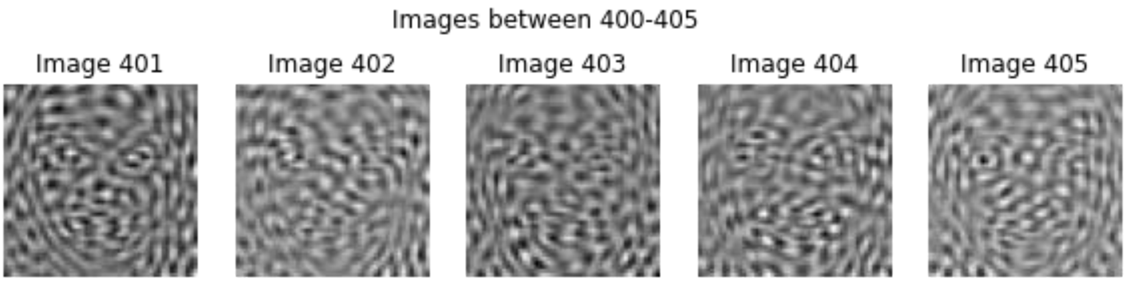


ما در شکل بالا برای ۱۵۰۰ تای اول رسم کردیم که همانطور که مشخص است تا این حدود نیز کاملا همپوشانی دارد. (تعداد کل مقادیر ویژه حدود ۲۵۰۰ میباشد)

۶-۲. قسمت ۲

پنج eigenface اول و آخر را بصورت زیر رسم میکنیم:

۵ تا از ۴۰۰ تا ۴۰۵ نیز رسم میکنیم:



همانطور که مشاهده میشود، با انتخاب eigenface متناسب با مقدار ویژه بالاتر، تصویر واضح تر بوده و ویژگی های بیشتری در آن پیداست.

درصورتی که هرچه مقدار ویژه کم میشود، تصویر ناواضح میشود و ویژگی خاصی در آن مشخص نیست.

۶-۳. قسمت ۳

نتیجه KNN قبل از اعمال PCA:

```
Results for k=1:
  Accuracy (CCR): 37.37%
  Confusion Matrix:
[[215  13  87 141 100  40 203]
 [  9  32   6  12   6   6  16]
 [ 78  12 297 124 118  68 123]
 [131  31 101 586 181  54 359]
 [100  16 100 203 298  46 203]
 [ 36   8  38  93  41 336  82]
 [ 97  18  78 228 135  55 382]]
```

```
Results for k=2:
  Accuracy (CCR): 32.88%
  Confusion Matrix:
[[275  17 119 203  90  28  67]
 [ 13  34   8  15   6   4   7]
 [140  21 320 158 100  42  39]
 [273  59 188 650 133  41  99]
 [189  39 167 261 224  23  63]
 [ 84  18  87 150  63 214  18]
 [176  29 133 324 121  39 171]]
```

نتیجه KNN بعد از اعمال PCA:

در این قسمت ما تعداد کامپوننت ها را برابر با ۴۰ میگیریم چون بیش از ۸۵ درصد از واریانس را پوشش میدهد.

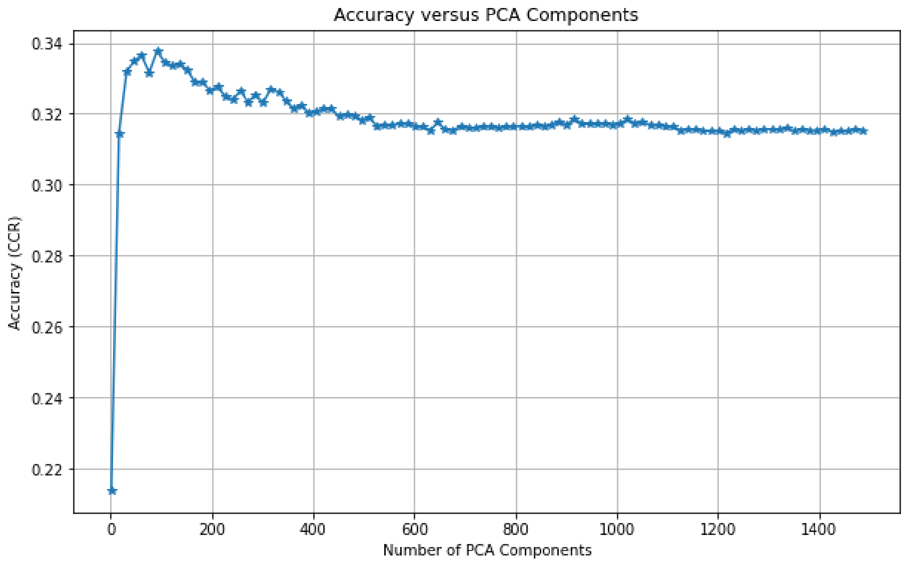
```
Results for k=1:
  Accuracy (CCR): 38.12%
  Confusion Matrix:
[[222  10  88 160 120  44 155]
 [  8  34   6  18   4   5  12]
 [ 82  14 319 132 108  79  86]
 [177  22 118 617 169  66 274]
 [111  15 108 235 288  50 159]
 [ 44   7  49  91  42 351  50]
 [118  16  89 228 121  63 358]]
```

```
Results for k=2:
  Accuracy (CCR): 32.72%
  Confusion Matrix:
[[301  14 127 199  80  35  43]
 [ 18  35  12  14   4   2   2]
 [163  18 348 148  76  44  23]
 [317  50 211 629 126  44  66]
 [218  24 155 296 205  31  37]
 [ 97  15 114 130  50 219   9]
 [199  22 159 313 110  48 142]]
```

درصد کمی بهبود داشتیم.

۶-۴. قسمت ۴

حال مقدار component ها را از ۱ تا ۱۵۰۰ به فاصله ۱۵ تغییر میدهیم و نمودار CCR طبقه بندی را به ازای مقادیر مختلف کامپوننت ها و الگوریتم KNN رسم میکنیم:



همانطور که مشاهده میشود با افزایش تعداد کامپوننت ها درصد خطا کاهش می یابد، که علت آن زیاد بودن تعداد ویژگی ها می باشد که overfit رخ میدهد.

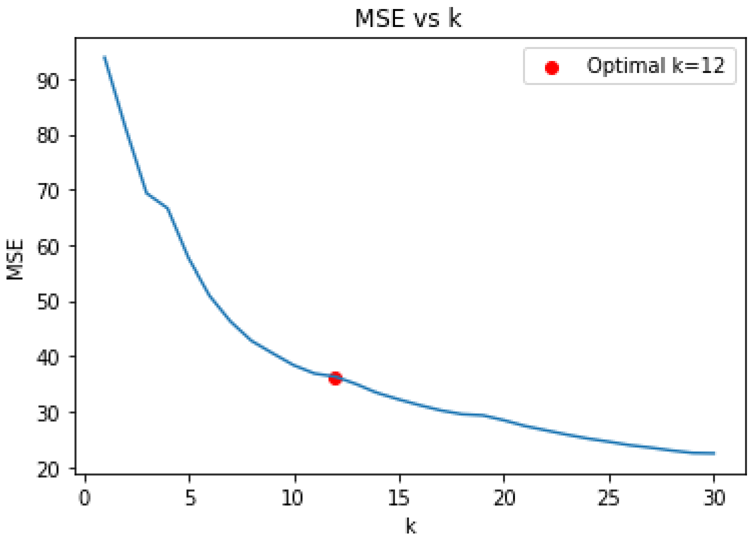
بهترین CCR مربوط به component number = 100 میباشد که ۹۰ درصد واریانس را شامل میشود.

پاسخ ۷ – شبیه سازی

مقادیر خطا بصورت زیر میباشد:

```
[93.83229508196722,
 81.11734342160571,
 69.32670239596469,
 66.61060739806642,
 57.621179066834806,
 50.898548759983186,
 46.256575241698194,
 42.76497793190416,
 40.53441677175284,
 38.378726355611605,
 36.86745376208491,
 36.29489911727617,
 34.937999159310635,
 33.325853299705756,
 32.17346258932324,
 31.13505359394704,
 30.196784363177805,
 29.519056326187474,
 29.310569567044976,
 28.42510508617066,
 27.39145018915511,
 26.605081967213113,
 25.827607187894074,
 25.12198087431694,
 24.546709751996637,
 23.92194094157209,
 23.47336485918453,
 22.954584909625893,
 22.52255884825557,
 22.45190941572089]
```

با رسم مقادیر خطا داریم:



همانطور که مشاهده میشود، نقطه ای که شروع به کم شدن میکند را بعنوان نقطه بهینه انتخاب میکنیم، نحوه انتخاب به این صورت بوده است که یک threshold = 1 تعریف شده است و در صورتی که اختلاف با خطای قبلی کمتر از این مقدار باشد، این نقطه بعنوان زانو انتخاب میشود.

با قرار دادن $k = 12$ و فشرده کردن تصویر داریم: