فصل ششم
شناسایی الگو
تولید ویژگی
Feature Generation, Data transformation and Dimensionality Reduction

محمدجواد فدائی‌اسلام

Introduction-Transform

- هدف از تبدیل(Transform) انتقال ویژگی‌ها به فضای جدید است.
- اگر تبدیل به درستی انتخاب شود. ویژگی‌ها در فضای جدید با یک فشرده‌سازی بالای اطلاعات مواجه خواهند شد.
- به این فعالیت گاهی تکنیک کاهش بعد (dimensionality reduction) هم گفته می‌شود.

تعریف ترانهاده – متعامد

$$A = \begin{bmatrix} 1 & -2 & 5 \\ 1 & 0 & 4 \end{bmatrix}$$

$$A^{\mathrm{T}} = \begin{bmatrix} 1 & 1 \\ -2 & 0 \\ 5 & 4 \end{bmatrix}$$

$\boldsymbol{A^T}$

T denotes the Transpose operation

ترانهاده

A real matrix is called orthogonal if $A^{-1} = A^T$

متعامد

تعریف: بردار ویژه – مقدار ویژه

- اگر برداری به نام v در رابطه زیر صدق کند

$$Av = \lambda v$$

صدق کند، آنگاه v را بردار ویژه (eigenvector) و λ را مقدار ویژه (eigenvalue) متناظر با آن بردار برای ماتریس A می‌نامند.

یافتن بردار ویژه – مقدار ویژه

- برای یافتن مقادیر ویژه ماتریس A با ابعاد $n \times n$، باید معادله $Av = \lambda v$ را برای اسکالر(های) λ حل کنیم. می‌توانیم از تساوی $Av = \lambda Iv$ استفاده نماییم که I ماتریس همانی است.

$$Av = \lambda Iv$$
$$Av - \lambda Iv = 0$$
$$(A - \lambda I)v = 0$$

- اگر $\det(A - \lambda I) = 0$ باشد، آنگاه یک حل غیربدیهی برای v وجود خواهد داشت.

بردار ویژه – مقدار ویژه (حل مثال)

$$A = \begin{bmatrix} 2 & -4 \\ -1 & -1 \end{bmatrix}$$

$$\begin{aligned} \det \begin{bmatrix} 2-\lambda & -4 \\ -1 & -1-\lambda \end{bmatrix} \\ &= (2-\lambda)(-1-\lambda) - (-4)(-1) \\ &= \lambda^2 - \lambda - 6 \\ &= (\lambda - 3)(\lambda + 2). \end{aligned}$$

$$\lambda_1 = 3, \lambda_2 = -2$$

$$(A - 3I)v = 0$$

$$\begin{bmatrix} 2-3 & -4 \\ -1 & -1-3 \end{bmatrix} v = 0, \begin{bmatrix} -1 & -4 \\ -1 & -4 \end{bmatrix} v = 0, v_1 = \begin{bmatrix} 4 \\ -1 \end{bmatrix},$$

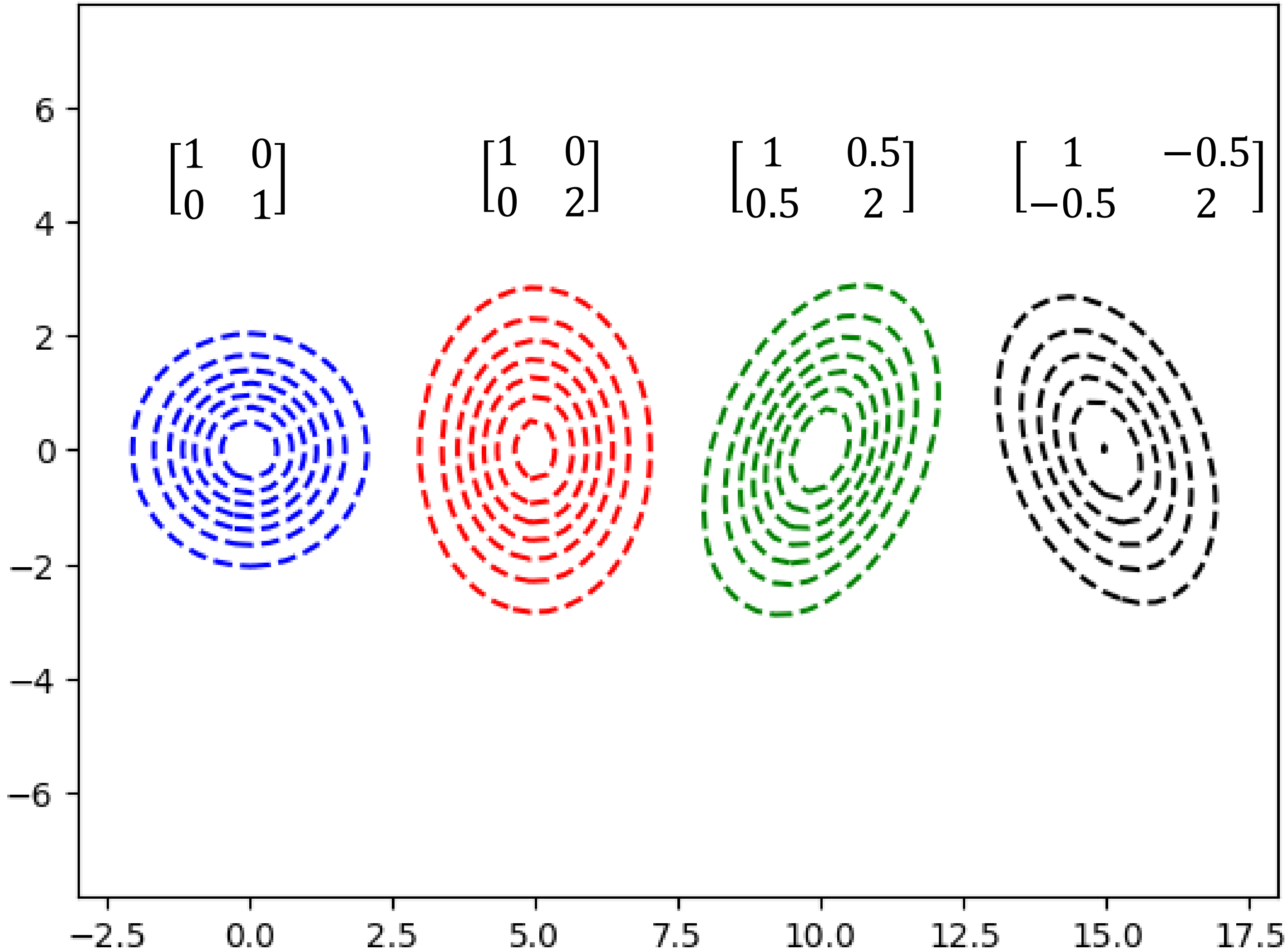
$$\begin{bmatrix} 2+2 & -4 \\ -1 & -1+2 \end{bmatrix} v = 0, \begin{bmatrix} +4 & -4 \\ -1 & 1 \end{bmatrix} v = 0, v_2 = \begin{bmatrix} -4 \\ -4 \end{bmatrix}$$

بردار ویژه ماتریس کواریانس

- بردارهای ویژه ماتریس کواریانس برهم عمود هستند.
- بردار ویژه متناظر با مقدار ویژه بیشتر، نماینده جهتی است که پراکندگی داده‌ها در آن سو بیشتر است.

پراکندگی داده برای ماتریس‌های کوواریانس مختلف

Covariance Matrix



```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal
```

```python
print("Covariance Matrix")
mu1 = np.array([0.0 , 0.0])
mu2 = np.array([5 , 0])
mu3 = np.array([10 , 0])
mu4 = np.array([15 , 0])

covmat1 = np.array([[1, 0], [0 , 1]])
covmat2 = np.array([[1, 0], [0 , 2]])
covmat3 = np.array([[1, 0.5], [0.5 , 2]])
covmat4 = np.array([[1, -0.5], [-0.5 , 2]])

xx, yy = np.meshgrid(np.linspace(-3, 18, 70), np.linspace(-4, 4, 70))
pos = np.dstack((xx, yy))
pdf1 = multivariate_normal(mu1, covmat1).pdf(pos)
pdf2 = multivariate_normal(mu2, covmat2).pdf(pos)
pdf3 = multivariate_normal(mu3, covmat3).pdf(pos)
pdf4 = multivariate_normal(mu4, covmat4).pdf(pos)

contour1 = plt.contour(xx, yy, pdf1, colors='blue' , linestyles='dashed' )
contour2 = plt.contour(xx, yy, pdf2, colors='red', linestyles='dashed')
contour3 = plt.contour(xx, yy, pdf3, colors='green', linestyles='dashed')
contour4 = plt.contour(xx, yy, pdf4, colors='black', linestyles='dashed')
plt.axis('equal')
plt.show()
```

The Karhunen-Loève Transform

تبدیل کارهانن لویی یا PCA

principal component analysis (PCA)

تجزیه به مولفه‌های اصلی

این تبدیل داده‌ها را به فضایی می‌برد که محورهای جدید عمود بر هم یا مستقل از هم هستند.

گام‌ها برای یافتن PCA

- **گام اول**: استانداردسازی
- **گام دوم**: محاسبه ماتریس کواریانس
- **گام سوم**: محاسبه بردارویژه و مقدار ویژه ماتریس کواریانس برای یافتن مولفه‌های اصلی

گام‌های محاسبه PCA

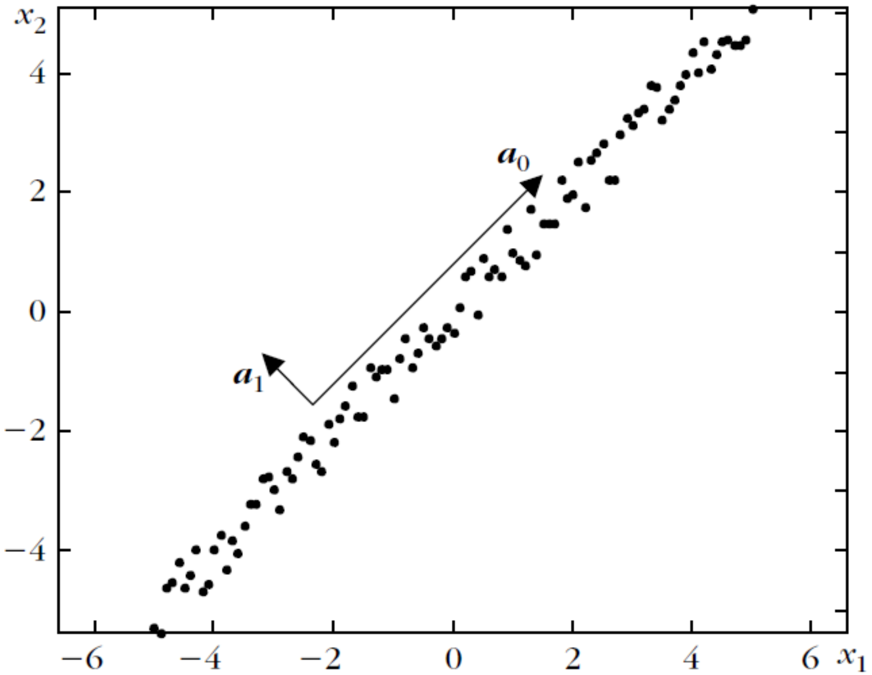
اگر بین دامنه متغیرهای اولیه تفاوت زیادی وجود داشته باشد، آن دسته از متغیرهایی که دامنه بزرگتر دارند بر متغیرهایی با دامنه کوچک غالب می شوند که منجر به بایاس یا تمایل به یک سو می‌شود. بنابراین، استانداردسازی داده‌ها می‌تواند از این مشکل جلوگیری کند.

$$z = \frac{value - mean}{standard\ deviation}$$

ماتریس کوواریانس یک ماتریس متقارن است.

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

مثال ۱



$a_0 = [0.7045,\ 0.7097]^T$

$a_1 = [-0.7097,\ 0.7045]^T$

$\lambda_0 = 17.26, \quad \lambda_1 = 0.04$

$\lambda_0 >> \lambda_1$

- داده‌ها مربوط به معادله $x_2 = x_1 + \epsilon$ است که ϵ نویز را نشان می‌دهد.
- بردار a_i نشان‌دهنده بردار ویژه و λ_i نشان‌دهنده مقدار ویژه است.
- در امتداد محور a_0 داده‌ها بیشترین واریانس را دارند.
- با توجه به اطلاعاتی که PCA ارائه می دهد (اختلاف زیاد بین دو مقدار ویژه) ابعاد مجموعه تقریباً یک است.
- خط a_0 تقریباً موازی با خط $x_1 = x_2$ است.

مثال ۲

The correlation matrix of a vector $\boldsymbol{x}$ is given by

$$R_x = \begin{bmatrix} 0.3 & 0.1 & 0.1 \\ 0.1 & 0.3 & -0.1 \\ 0.1 & -0.1 & 0.3 \end{bmatrix}$$

Compute the KL transform of the input vector.

The eigenvalues of R_x are $\lambda_0 = \lambda_1 = 0.4$, $\lambda_2 = 0.1$. Since the matrix R_x is symmetric, we can always construct orthonormal eigenvectors. For this case we have

$$\boldsymbol{a}_0 = \frac{1}{\sqrt{6}} \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{a}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}, \quad \boldsymbol{a}_2 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \end{bmatrix} = \underbrace{\begin{bmatrix} 2/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{3} & -1/\sqrt{3} & -1/\sqrt{3} \end{bmatrix}} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \end{bmatrix}$$

ماتریس تبدیل دارای سطر و ستون‌های متعامد با اندازه واحد است (orthonormal).

مثال ۲ – ادامه – کاهش بعد

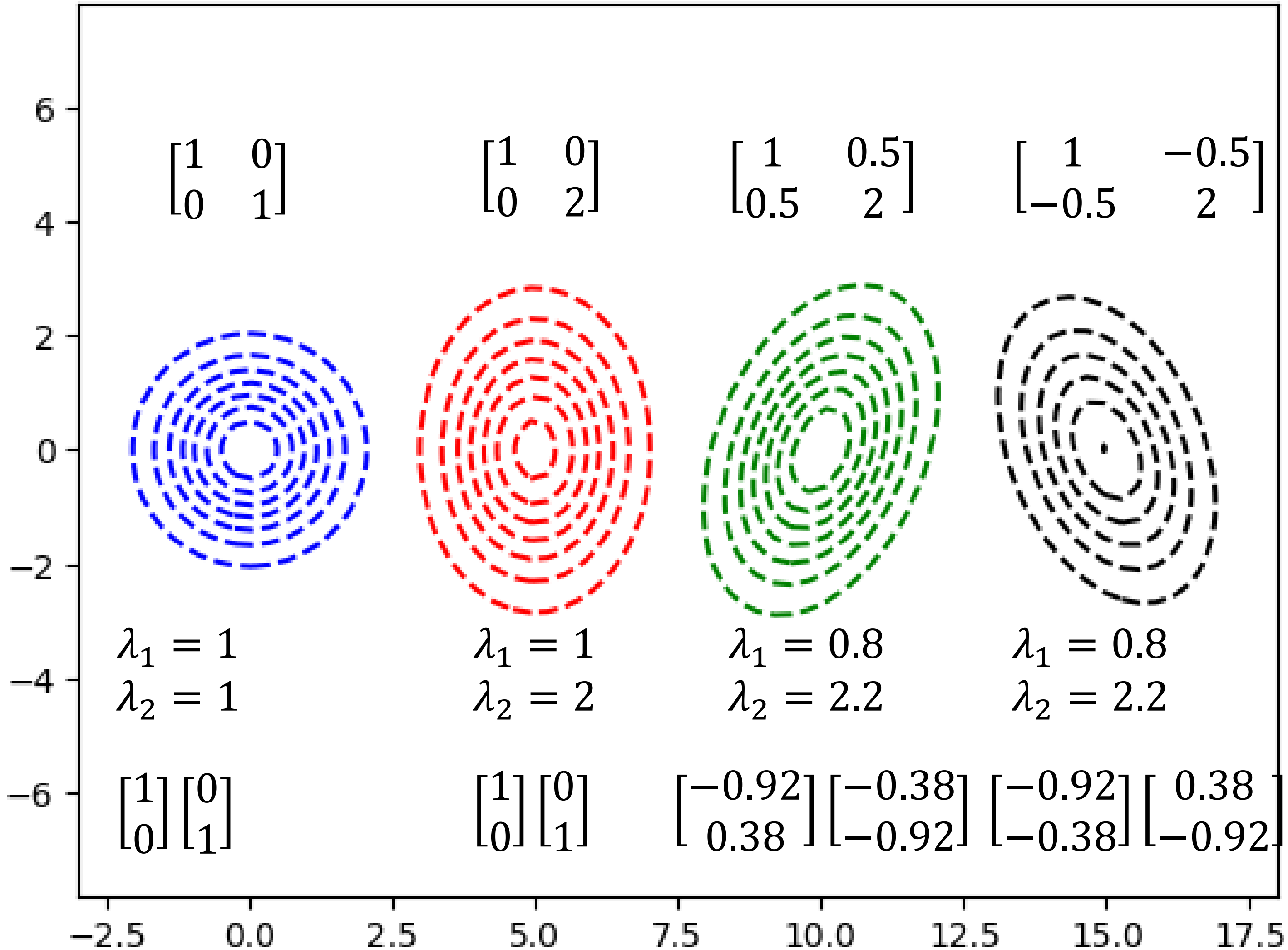
$$\begin{bmatrix} y(0) \\ y(1) \\ \text{[redacted]} \end{bmatrix} = \begin{bmatrix} 2/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ & \text{[redacted]} & \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \end{bmatrix}$$

- ماتریس تبدیل معکوس پذیر است، یعنی در تبدیل فضا اطلاعات از دست نمی‌رود.

- با حذف بردار ویژه‌های کم اهمیت می‌توان به کاهش بعد رسید، در این صورت با از دست رفتن داده مواجهیم.

مثال۳: مقدار ویژه و بردار ویژه برای ماتریس‌های کواریانس مختلف

Covariance Matrix



مثال ۳ – ادامه

```
print(covmat1)
print(np.linalg.eig(covmat1))
```

```
[[1 0]
 [0 1]]
(array([1., 1.]), array([[1., 0.],
       [0., 1.]]))
```

مثال ۴ – اثر ویژگی وابسته

```
np.set_printoptions(precision=2)
x = np.array([[1, 3, 4],
     [2,  6, 8],
     [3,  7, 10],
     [4, -1, 3],
     [5, 11, 16],
     [6, 12, 18],
     [7, 56, 63],
     [8, -9, -1],
     [9,  9, 18],
     [10, 33, 43]])
covx = np.cov(np.transpose(x))
print(covx)
eigmatx = np.linalg.eig(covx)
print(eigmatx)
```

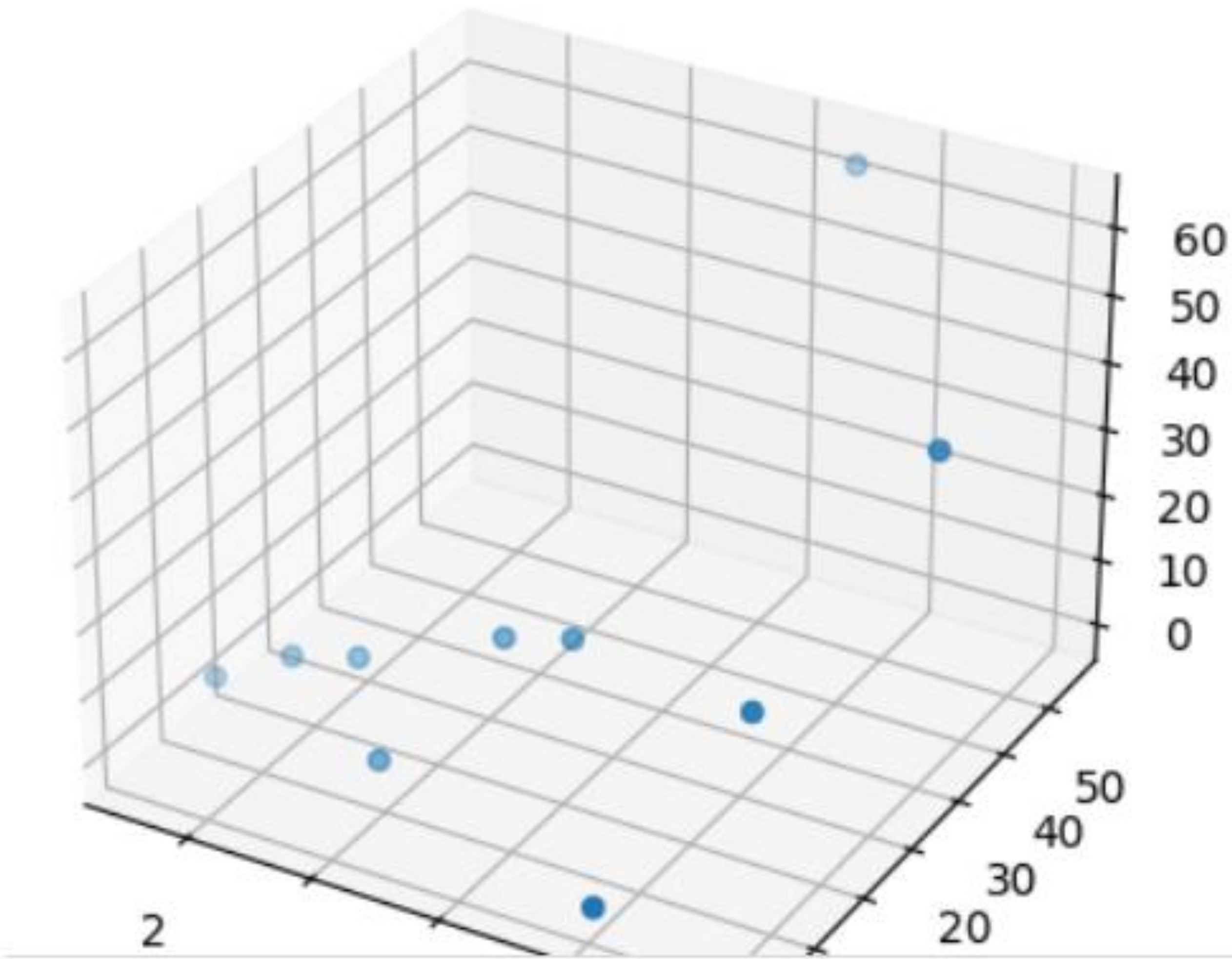
$$z = x + y$$

```
[[  9.17  21.28  30.44]
 [ 21.28 348.23 369.51]
 [ 30.44 369.51 399.96]]
(array([7.46e+02, 1.10e+01, 4.62e-14]), array([[-0.05, -0.81, -0.58],
       [-0.68,  0.45, -0.58],
       [-0.73, -0.36,  0.58]]))
```

~ 0

مثال ۴ – پراکندگی داده‌ها

```
ax = plt.axes(projection='3d')
ax.scatter3D(x[:,0], x[:, 1] ,x[:, 2])
```



مثال ۵

```python
np.set_printoptions(precision=2)
x = np.array([[1, 2],
     [2,  4],
     [3,  6],
     [4,  8],
     [5, 10],
     [6, 12],
     [7, 14],
     [8, 16],
     [9, 18],
     [10, 20]])
covx = np.cov(np.transpose(x))
print(covx)
eigmatx = np.linalg.eig(covx)
print(eigmatx)
maxeigv = eigmatx[1]
xx = np.dot(x,maxeigv)
print(xx)
print(xx.round())
```

```
[[ 9.17 18.33]
 [18.33 36.67]]
(array([ 0.   , 45.83]), array([[-0.89, -0.45],
       [ 0.45, -0.89]]))
[[ 0.00e+00 -2.24e+00]
 [ 0.00e+00 -4.47e+00]
 [-1.11e-16 -6.71e+00]
 [ 0.00e+00 -8.94e+00]
 [-3.33e-16 -1.12e+01]
 [-2.22e-16 -1.34e+01]
 [-1.11e-16 -1.57e+01]
 [ 0.00e+00 -1.79e+01]
 [ 1.11e-16 -2.01e+01]
 [-6.66e-16 -2.24e+01]]
[[  0.  -2.]
 [  0.  -4.]
 [ -0.  -7.]
 [  0.  -9.]
 [ -0. -11.]
 [ -0. -13.]
 [ -0. -16.]
 [  0. -18.]
 [  0. -20.]
 [ -0. -22.]]
```

کاهش بعد – جداپذیری در PCA
Dimensionality Reduction
Separability of PCA

- بهینگی تبدیل تجزیه به مولفه‌های اصلی نسبت به حداقل مربعات خطا (MSE) منجر به ویژگی‌هایی جدید با فشردگی بالا می‌شود. از این‌رو PCA را می‌توان ابزاری برای انتخاب m ویژگی غالب در فضای جدید از n ویژگی اولیه در نظر گرفت ($m < n$).
- اگرچه تجزیه به مولفه‌های اصلی ممکن است روش خوبی برای فشرده‌سازی باشد اما در مواردی لزوماً منجر به حداکثر تفکیک‌پذیری در کلاس‌بندی. این معقول است، زیرا کاهش ابعاد با توجه به تعلق داده‌ها به کلاس‌ها بهینه نشده است.

جداپذیری در PCA

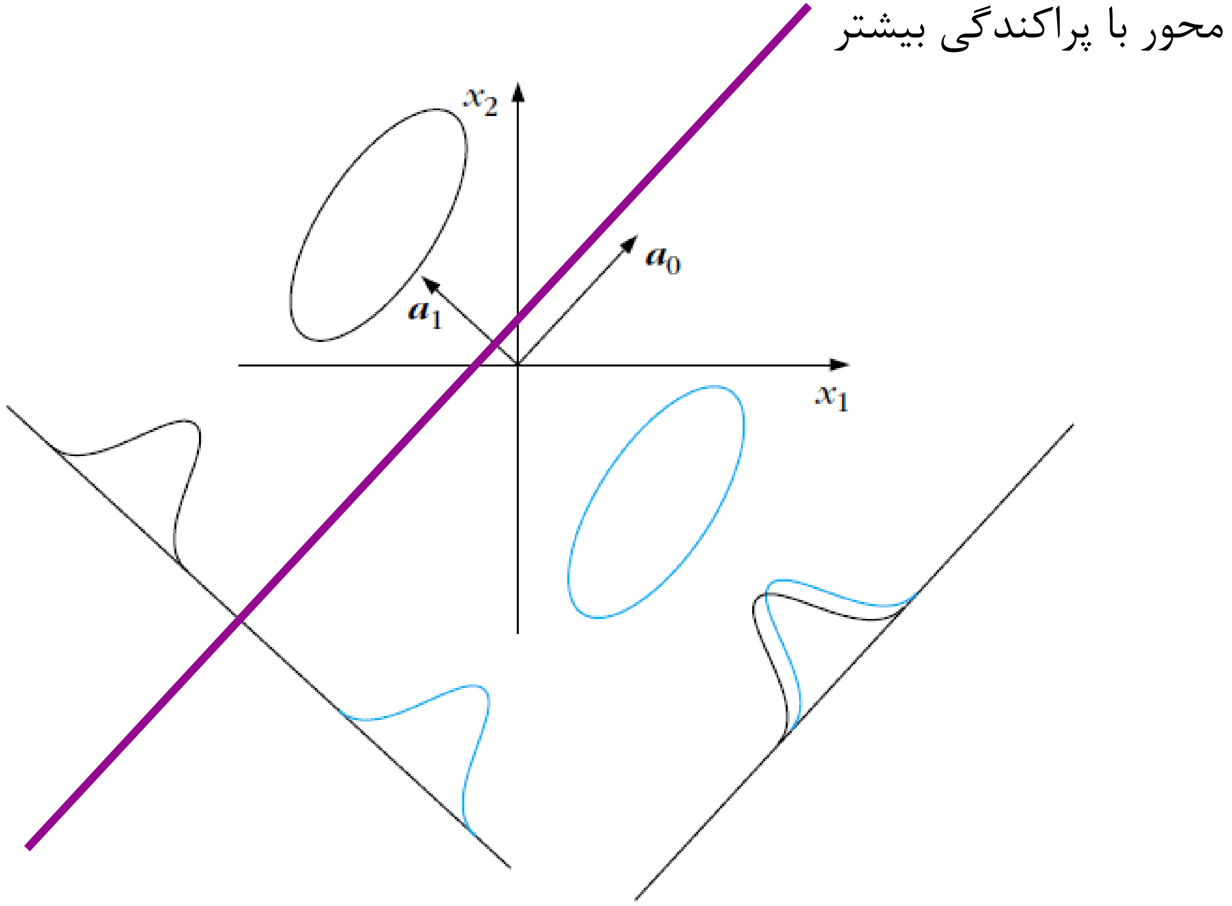
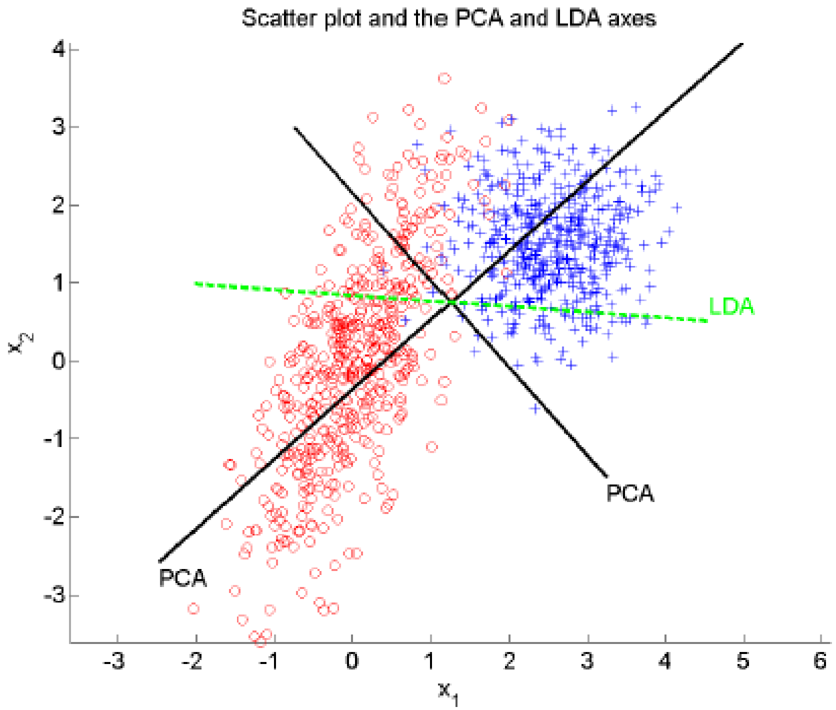


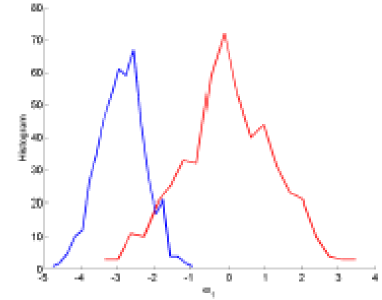
FIGURE 6.1

The KL transform is not always best for pattern recognition. In this example, projection on the eigenvector with the larger eigenvalue makes the two classes coincide. On the other hand, projection on the other eigenvector keeps the classes separated.

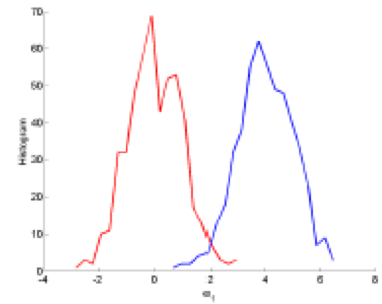
PCA vs LDA



(a) Scatter plot.



(b) Projection onto the first PCA axis.



(c) Projection onto the first LDA axis.

Figure 8: Scatter plot and the PCA and LDA axes for a bivariate sample with two classes. Histogram of the projection onto the first LDA axis shows better separation than the projection onto the first PCA axis.