کاهش ابعاد

سید ناصر رضوی www.snrazavi.ir

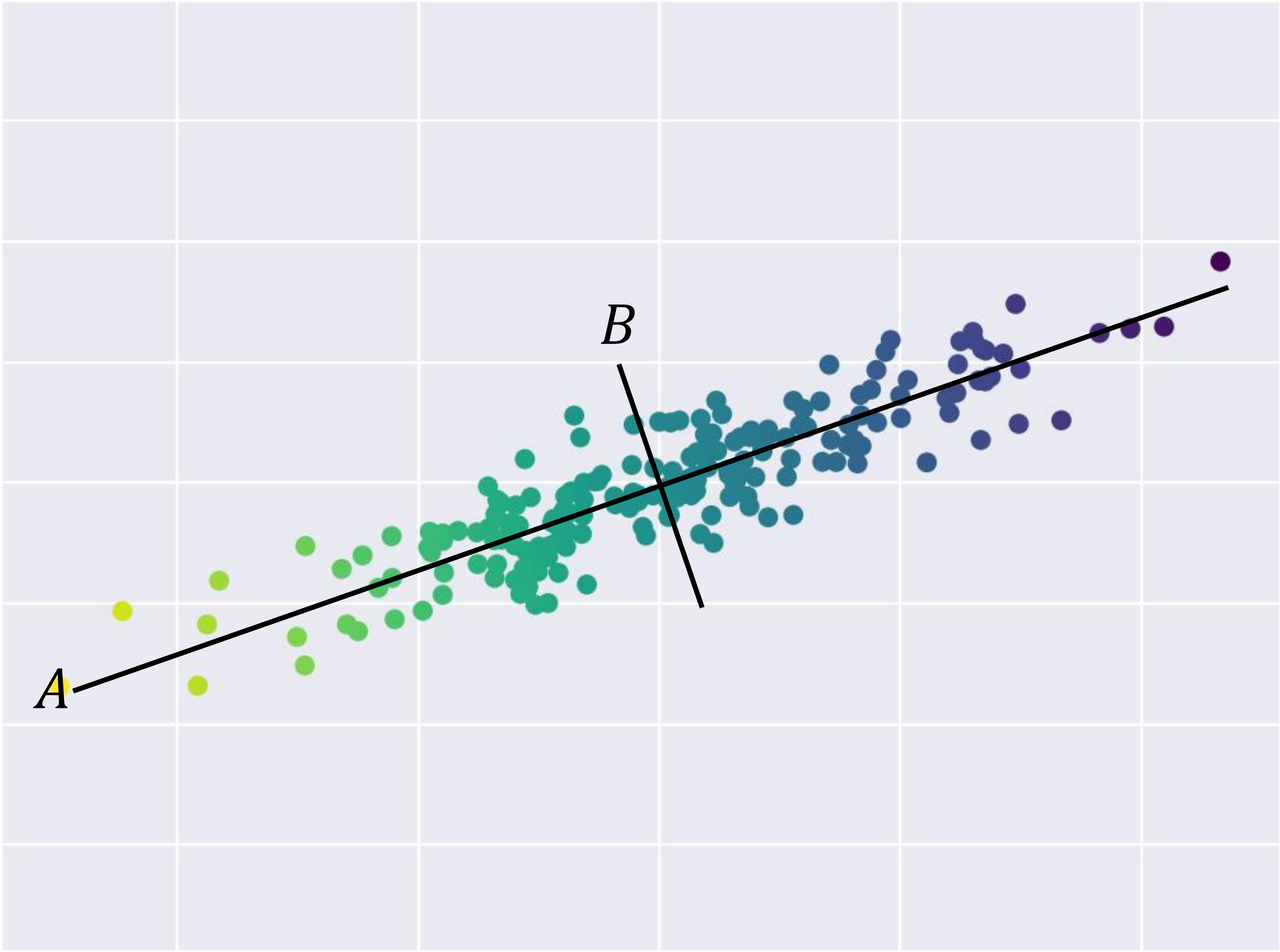
۱۳۹۷

کاهش ابعاد

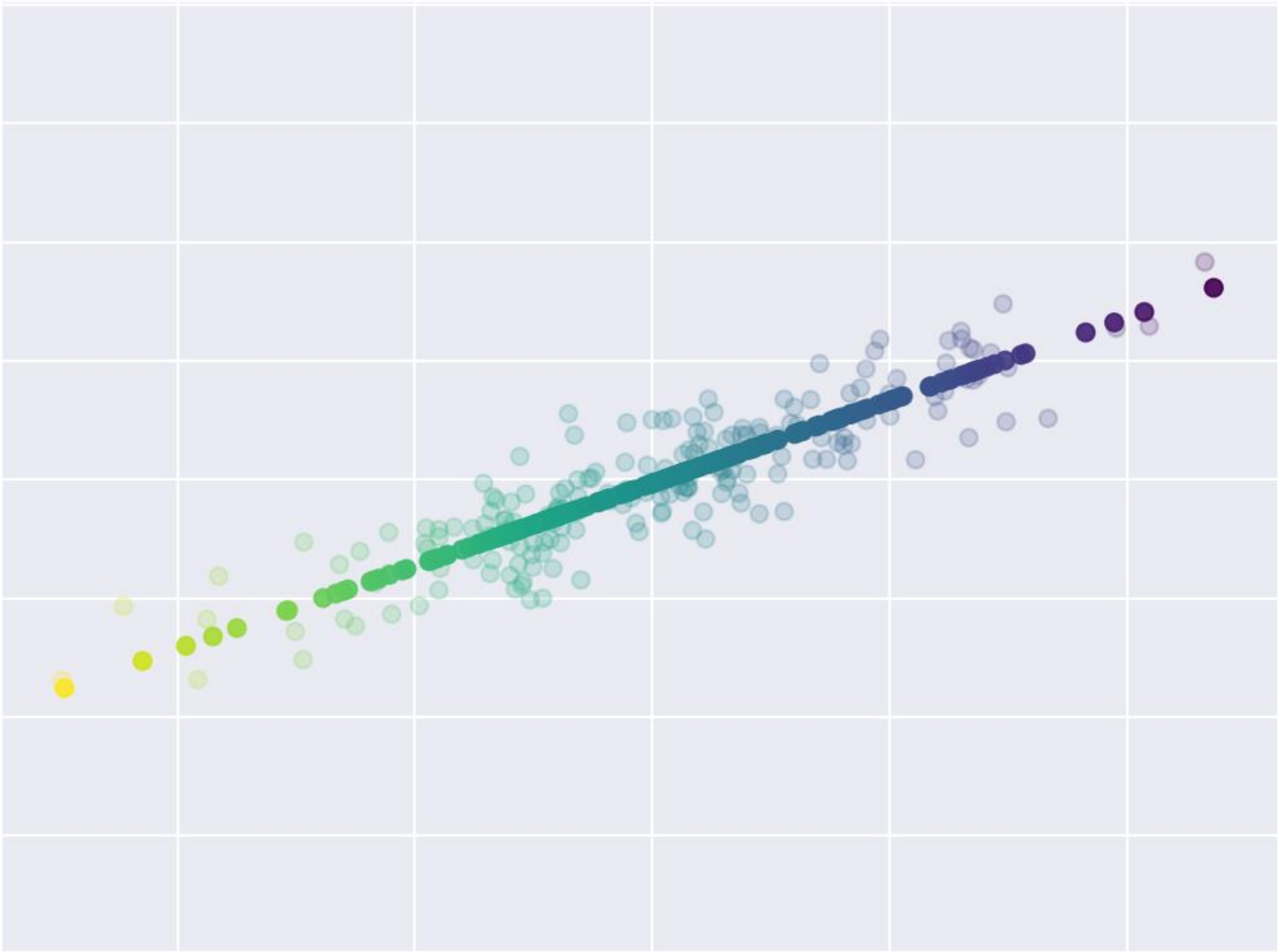
- انگیزه.
  - نمایش داده‌ها
  - فشرده‌سازی داده‌ها
    - کاهش مصرف حافظه
    - استفاده آسان‌تر از مجموعه داده‌ها
    - کاهش هزینه‌های محاسباتی بسیاری از الگوریتم‌ها
    - حذف نویز و افزایش دقت الگوریتم یادگیری
    - ساده‌تر کردن درک نتایج
- روش‌ها.
  - تحلیل مؤلفه‌های اصلی
  - تحلیل فاکتورها
  - تحلیل مؤلفه‌های مستقل

تحلیل مؤلفه‌های اصلی (PCA)

تحلیل مؤلفه‌های اصلی



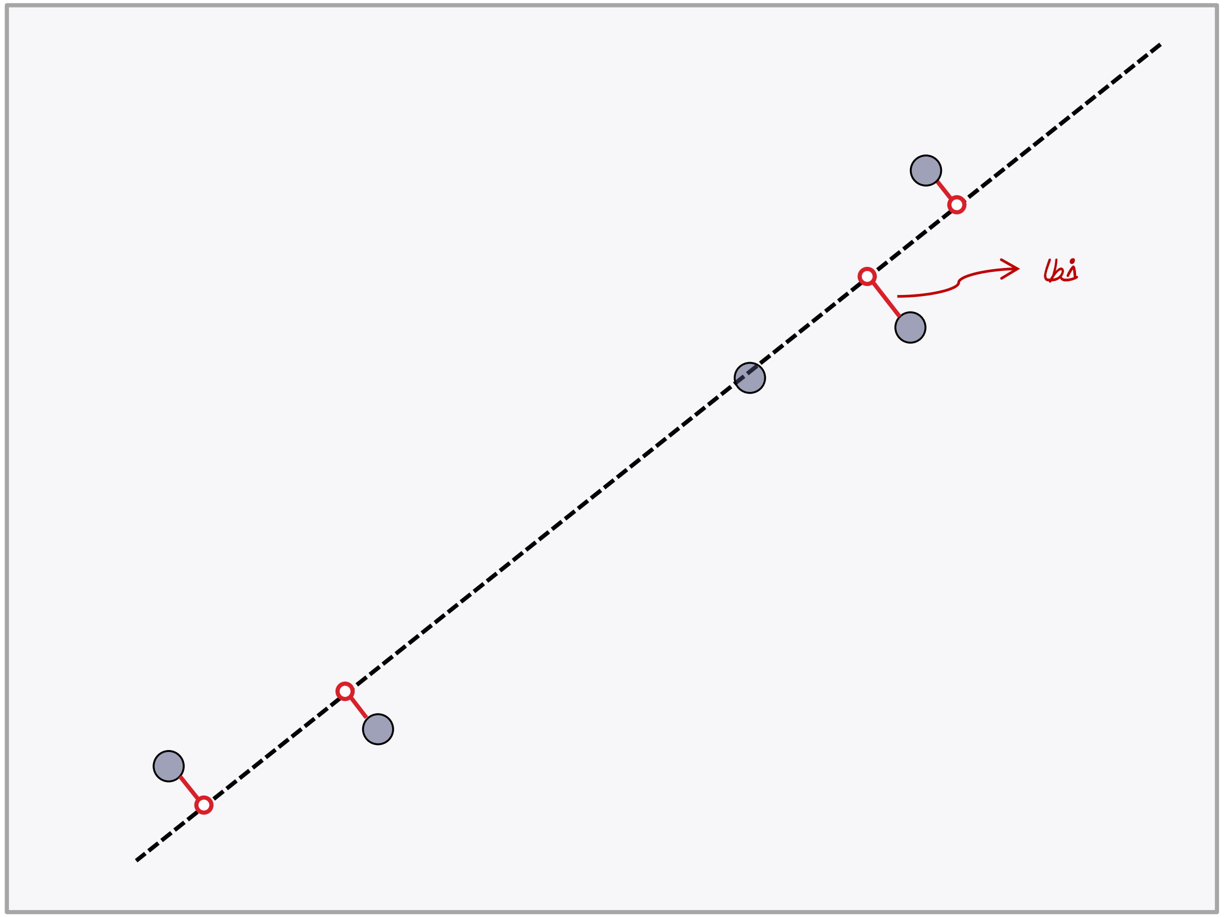
تحلیل مؤلفه‌های اصلی



تحلیل مؤلفه‌های اصلی: بیان مسئله

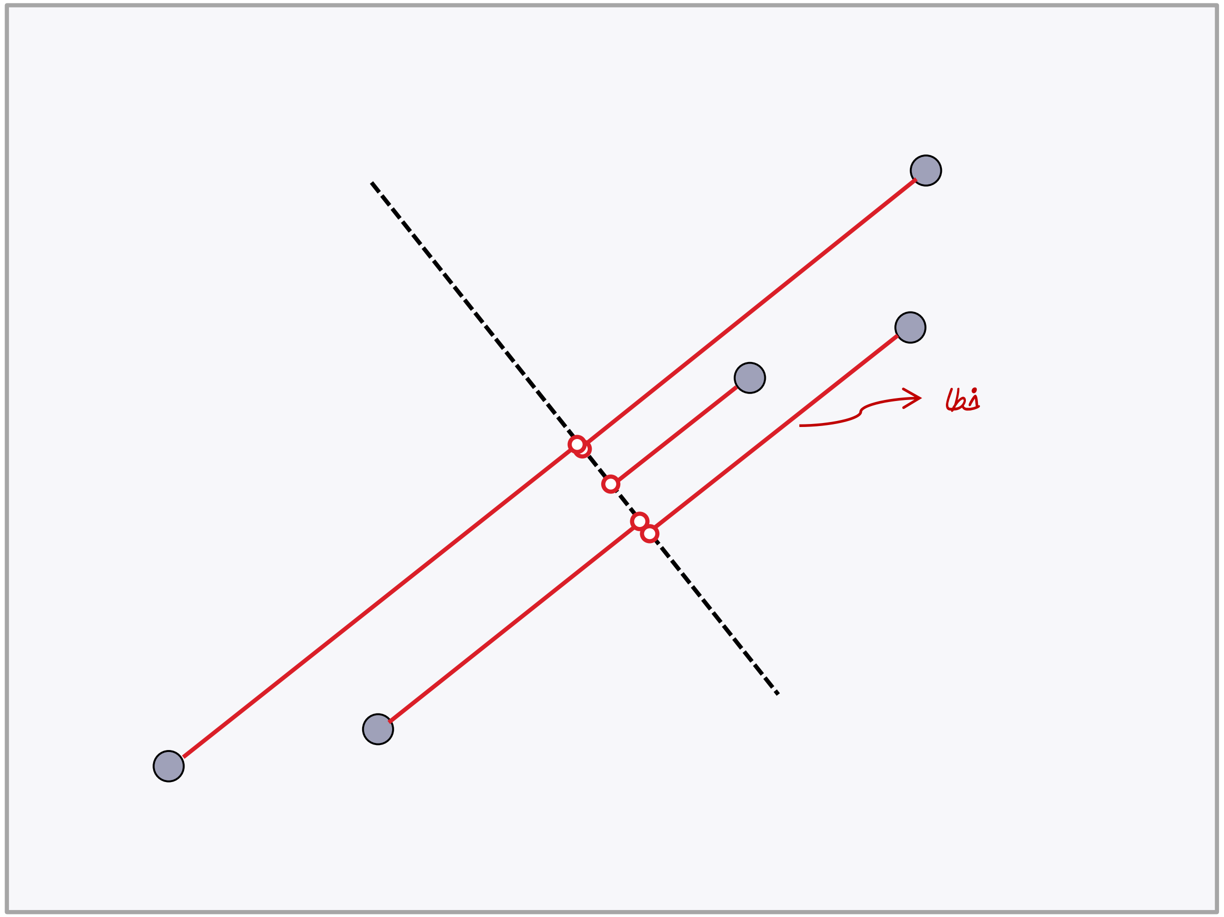
بیان مسئله

- هدف. کمینه‌سازی مجموع مربعات خطای تابش



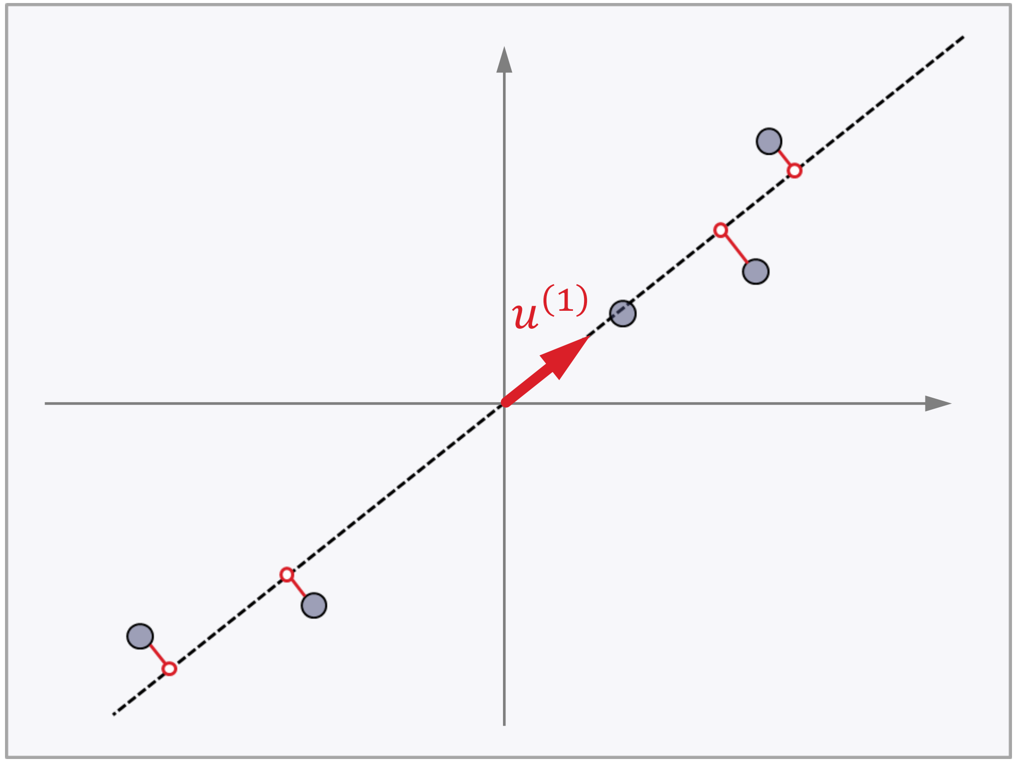
بیان مسئله

- **هدف**. کمینه‌سازی مجموع مربعات خطای تابش



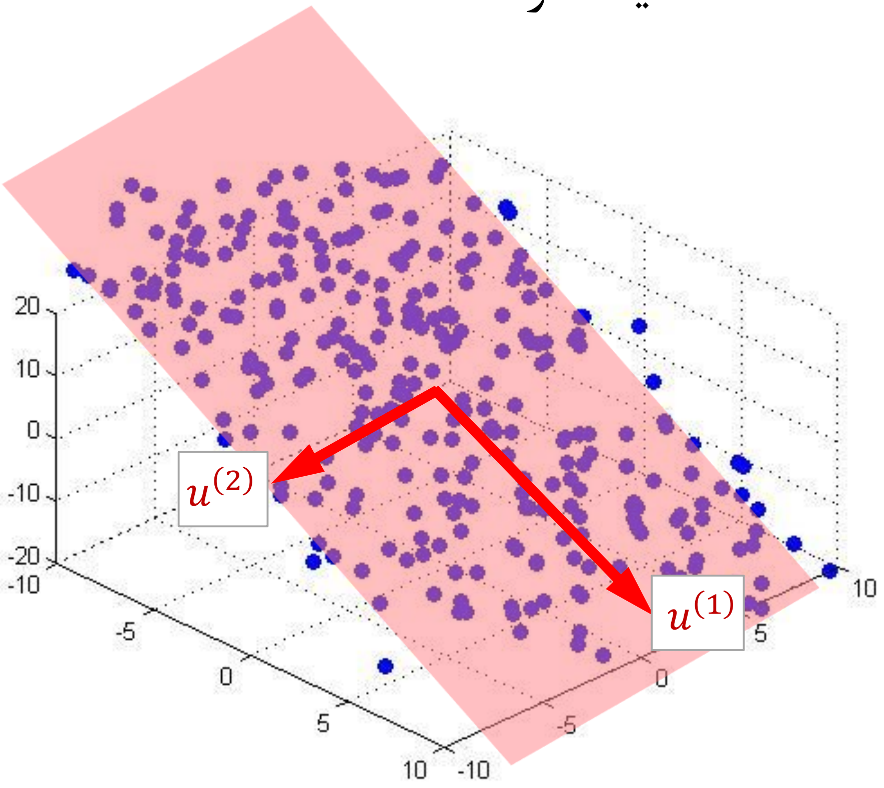
بیان مسئله

- کاهش ابعاد از ۲ به ۱. یافتن یک جهت مانند $u^{(1)} \in \mathbb{R}^2$ به طوری که با تصویر کردن نقاط در آن جهت، مجموع مربعات خطا کمینه گردد.

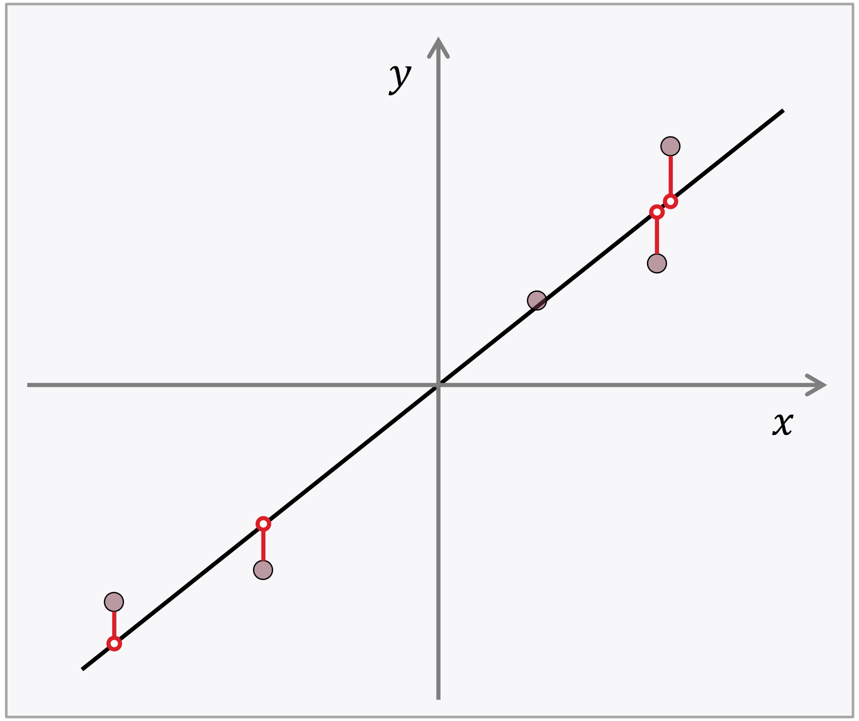


بیان مسئله

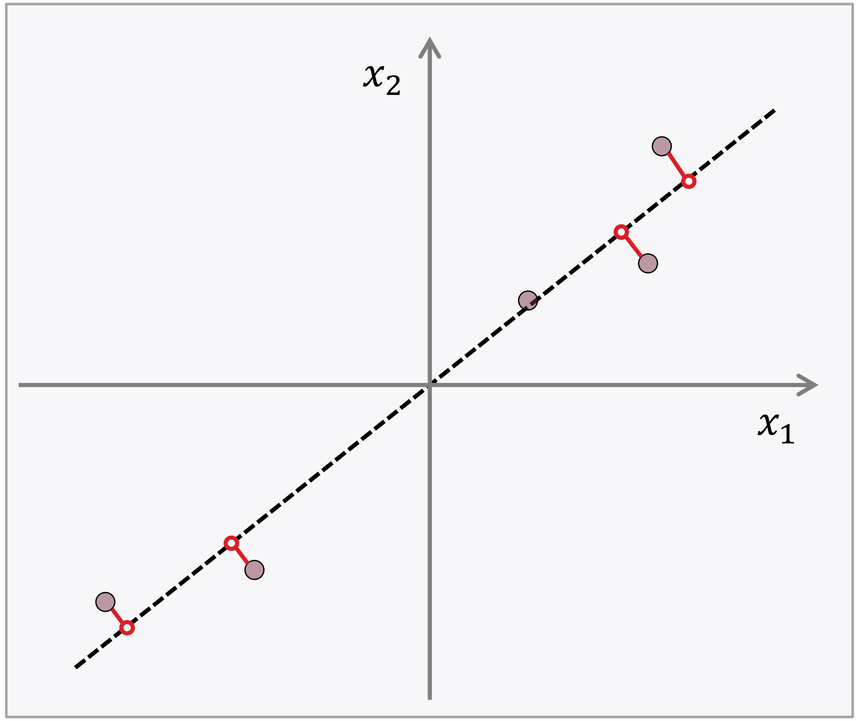
- کاهش ابعاد از n به k. یافتن k بردار متعامد مانند $u^{(1)}, u^{(2)}, \dots, u^{(k)} \in \mathbb{R}^n$ به طوری که با تصویر کردن نقاط در آن جهت‌ها، مجموع مربعات خطا کمینه گردد.



PCA همان رگرسیون است؟



رگرسیون خطی



تحلیل مؤلفه‌های اصلی

الگوریتم PCA

الگوریتم PCA: پیش‌پردازش

- مجموعه آموزشی.

$$\{x^{(1)}, x^{(2)}, x^{(3)}, \cdots, x^{(m)}\}, \qquad x^{(i)} \in \mathbb{R}^n$$

- پیش‌پردازش.

(۱) حذف میانگین.

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}, \qquad x_j^{(i)} = x_j^{(i)} - \mu_j$$

(۲) مقیاس‌بندی. [در صورت نیاز]

$$x_j^{(i)} = \frac{x_j^{(i)} - \mu_j}{s_j}$$

انحراف معیار در ویژگی j

الگوریتم PCA: کاهش ابعاد

- کاهش ابعاد داده‌ها از n به k.
  - محاسبه «ماتریس کوواریانس»:

$$\Sigma = \frac{1}{m} X^T X = \frac{1}{m}\sum_{i=1}^{n} x^{(i)} \left(x^{(i)}\right)^T$$

  - محاسبه «بردارهای ویژه» ماتریس کوواریانس:

$$[U, S, V] = svd(\Sigma)$$

  - انتخاب k بردار اول از ماتریس U:

$$U = \begin{bmatrix} | & | & \cdots & | \\ u^{(1)} & u^{(2)} & \cdots & u^{(n)} \\ | & | & \cdots & | \end{bmatrix}_{n\times n} \Rightarrow U_{reduced} = \begin{bmatrix} | & | & \cdots & | \\ u^{(1)} & u^{(2)} & \cdots & u^{(k)} \\ | & | & \cdots & | \end{bmatrix}_{n\times k}$$

الگوریتم PCA

- محاسبه داده‌های جدید با ابعاد k.

$$z^{(i)}_{k\times 1} = \begin{bmatrix} | & | & \cdots & | \\ u^{(1)} & u^{(2)} & \cdots & u^{(k)} \\ | & | & \cdots & | \end{bmatrix}^T \times x^{(i)}_{n\times 1}$$

$$= \begin{bmatrix} - & u^{(1)} & - \\ - & u^{(2)} & - \\ \vdots & \vdots & \vdots \\ - & u^{(k)} & - \end{bmatrix}_{k\times n} \times x^{(i)}_{n\times 1}$$

پیاده سازی در پایتون

- الگوریتم PCA.
  - پس از حذف میانگین و در صورت نیاز مقیاس‌بندی.

```
def PCA(X, k):
    m = X.shape[0]
    Sigma = (X.T @ X) / m
    U, S, V = svd(Sigma)
    U_reduced = U[:, :k]
    Z = X @ U_reduced
    return Z
```

محاسبه ماتریس کوواریانس

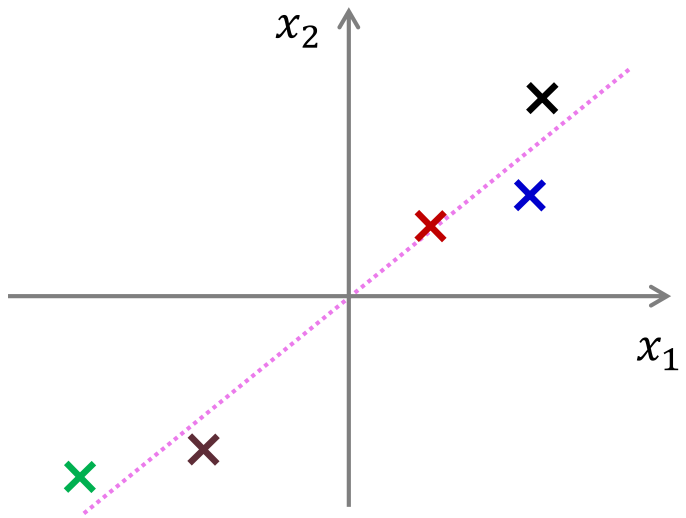
محاسبه تجزیه مقادیر منفرد

انتخاب k مؤلفه اول

محاسبه داده‌های جدید با k بعد

مثال: کاهش ابعاد

$$Z = X \times U_{reduced}$$

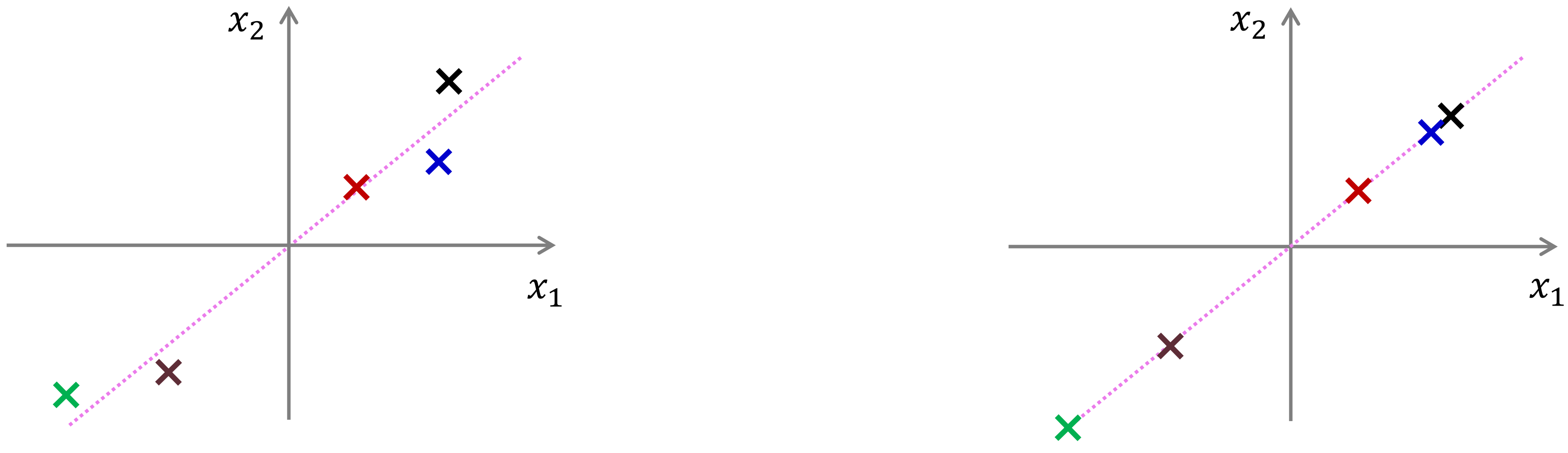


داده‌های اولیه (دو بعدی)

داده‌های جدید (یک بعدی)

مثال: کاهش ابعاد

$$X_{recovered} = Z * U_{reduced}^T + means$$



داده‌های اولیه (دو بعدی)

داده‌های بازسازی شده

الگوریتم PCA: حذف میانگین


```
X = np.array([[1, 1, 1, 0, 0],
              [2, 2, 2, 0, 0],
              [1, 1, 1, 0, 0],
              [5, 5, 5, 0, 0],
              [1, 1, 0, 2, 2],
              [0, 0, 0, 3, 3],
              [0, 0, 0, 1, 1]])

mu = X.mean(axis=0)
X_norm = X - mu
```

داده‌های اولیه

1	1	1	0	0
2	2	2	0	0
1	1	1	0	0
5	5	5	0	0
1	1	0	2	2
0	0	0	3	3
0	0	0	1	1

الگوریتم PCA: محاسبه بردارهای ویژه


```
m = X.shape[0]
Sigma = (X_norm.T @ X_norm) / m
U, S, V = np.linalg.svd(Sigma)

print(S)
```

$$S = \begin{bmatrix} s_{11} & 0 & \cdots & 0 \\ 0 & s_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_{nn} \end{bmatrix}$$

```
[8.72e+00    1.58e+00    6.69e-02    4.79e-16    1.35e-47]
```

الگوریتم PCA: کاهش ابعاد


```
U_red = U[:, :3]

X_proj = X_norm * U_red
```

```
[[ 0.17   1.37  -0.01]
 [-1.44   0.74  -0.02]
 [ 0.17   1.37  -0.01]
 [-6.28  -1.17  -0.06]
 [ 1.76  -1.1    0.57]
 [ 3.33  -1.92  -0.35]
 [ 2.3    0.7   -0.11]]
```

الگوریتم PCA: بازسازی داده های اولیه

```
X_approx = X_proj @ U_red + mu
```

```
[[ 1.00   1.00   1.00   0.00   0.00]
   2.00   2.00   2.00   0.00   0.00]
   1.00   1.00   1.00   0.00   0.00]
   5.00   5.00   5.00  -0.00  -0.00]
   1.00   1.00   0.00   2.00   2.00]
   0.00  -0.00   0.00   3.00   3.00]
  -0.00  -0.00  -0.00   1.00   1.00]]
```

داده‌های اولیه

1	1	1	0	0
2	2	2	0	0
1	1	1	0	0
5	5	5	0	0
1	1	0	2	2
0	0	0	3	3
0	0	0	1	1

انتخاب تعداد مؤلفه‌های اصلی

انتخاب تعداد مؤلفه‌های اصلی

- میانگین مجموع مربعات خطای تابش.

$$\frac{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}-x_{approx}^{(i)}\right\|^2}{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}\right\|^2} \leq 0.01$$

حفظ ۹۹ درصد واریانس

$$\frac{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}-x_{approx}^{(i)}\right\|^2}{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}\right\|^2} \leq 0.05$$

حفظ ۹۵ درصد واریانس

$$\frac{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}-x_{approx}^{(i)}\right\|^2}{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}\right\|^2} \leq 0.10$$

حفظ ۹۰ درصد واریانس

انتخاب تعداد مؤلفه‌های اصلی

یک الگوریتم غیر کارا

$k = 0$

repeat

{

$k = k + 1$

try $PCA(X)$ with k components

compute $U_{reduced}, z^{(1)}, z^{(2)}, \ldots, z^{(m)}, x_{approx}^{(1)}, x_{approx}^{(2)}, \ldots, x_{approx}^{(m)}$

} **until** $\frac{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} - x_{approx}^{(i)}\right\|^2}{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}\right\|^2} \leq 0.01$

انتخاب تعداد مؤلفه‌های اصلی

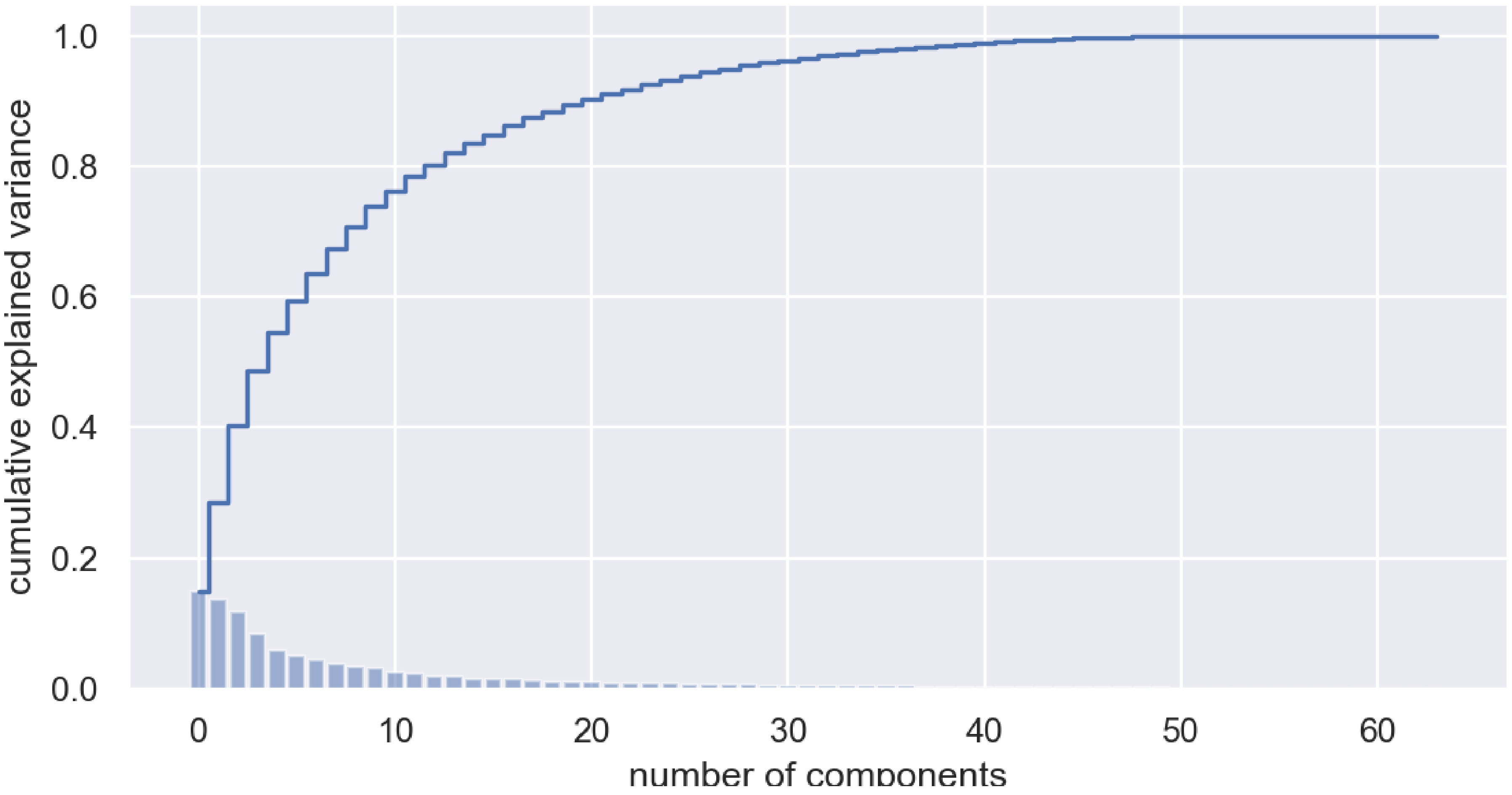

```
m, n = X.shape
X = X - X.mean(axis=0)
Sigma = (X.T @ X) / m
U, S, V = np.linalg.svd(Sigma)

for k in range(1, n + 1):
    total_var = np.sum(S[:k]) / np.sum(S)
    if total_var >= 0.99: break
return k
```

```
[8.72e+00    1.58e+00    6.69e-02    4.79e-16    1.35e-47]
```

$$S = \begin{bmatrix} s_{11} & 0 & \cdots & 0 \\ 0 & s_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_{nn} \end{bmatrix}$$

انتخاب تعداد مؤلفه‌های اصلی: ارقام ۸ در ۸



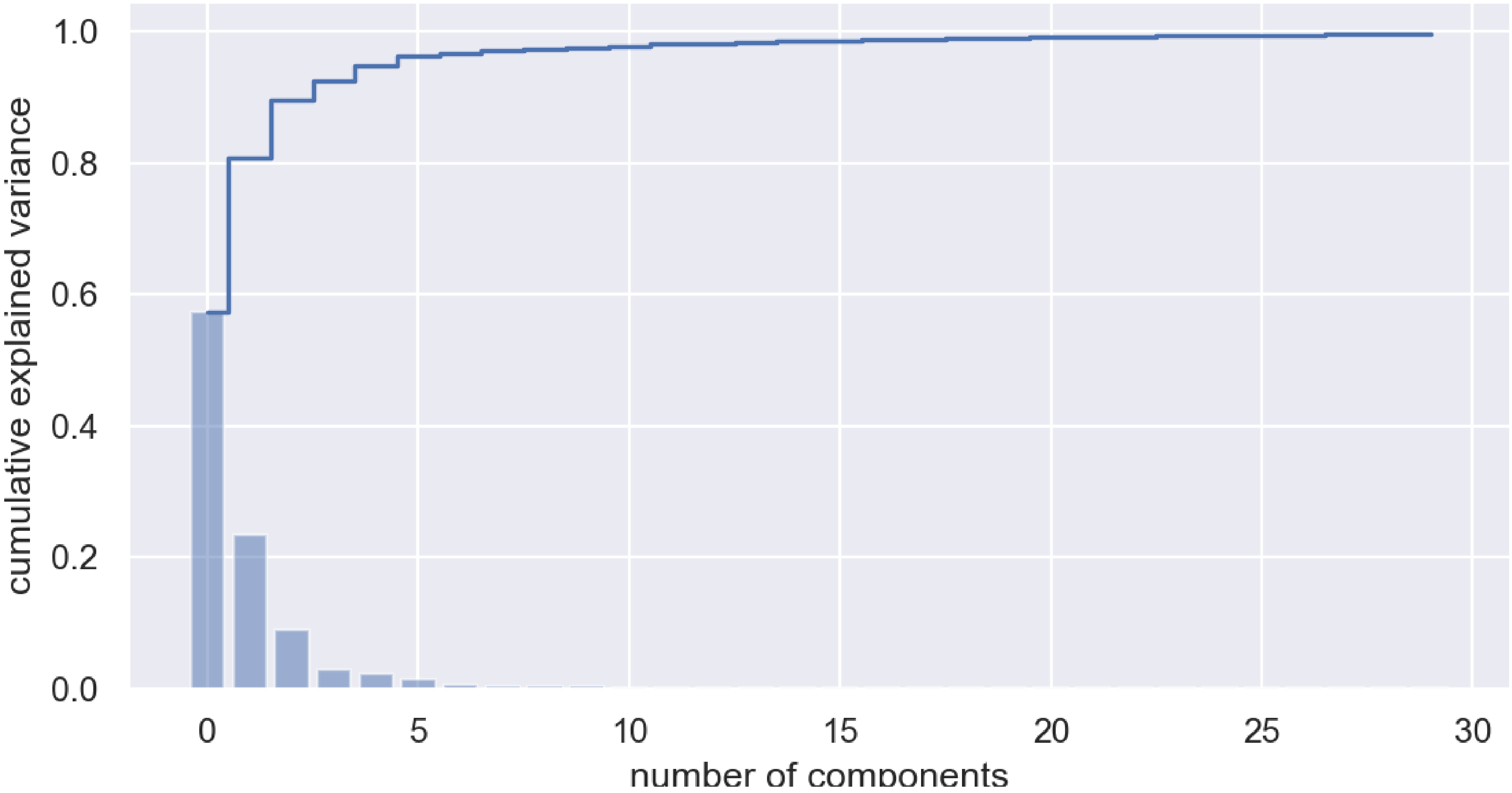
انتخاب تعداد مؤلفه‌های اصلی

- دادهای مربوط به نیمه‌هادی‌ها [۵۹۰ ویژگی]

تعداد مؤلفه‌ها	درصد واریانس	درصد تجمعی
۱	۵۹/۲	۵۹/۲
۲	۲۴/۱	۸۳/۴
۳	۹/۲	۹۲/۵
۴	۲/۳	۹۴/۸
۵	۱/۵	۹۶/۳
۶	۰/۵	۹۶/۸
۷	۰/۳	۹۷/۱
۲۰	۰/۰۸	۹۹/۳

http://archive.ics.uci.edu/ml/datasets/SECOM

انتخاب تعداد مؤلفه‌های اصلی



استفاده نادرست از PCA: مقابله با بیش‌برازش

- روش نادرست.
  - استفاده از $z^{(i)}$ به جای $x^{(i)}$ باعث کاهش تعداد ویژگی‌ها از n به k می‌شود؛
  - در نتیجه، با داشتن ویژگی‌های کمتر، احتمال بیش‌برازش کاهش می‌یابد.

- روش درست.
  - PCA در هنگاه کاهش ابعاد، اطلاعات مربوط به خروجی را استفاده نمی‌کند.
  - برای مقابله با بیش‌برازش از تنظیم استفاده کنید.

$$\min_{\theta} \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$

سخن آخر

- طراحی یک سیستم یادگیری ماشین.
  - ایجاد مجموعه آموزشی به صورت $\{(x^{(1)},y^{(1)}),(x^{(2)},y^{(2)}),\cdots,(x^{(m)},y^{(m)})\}$
  - استفاده از PCA به منظور کاهش ابعاد $x^{(i)}$ ها و به دست آوردن $z^{(i)}$ ها
  - اجرای مرحله آموزش بر روی $\{(z^{(1)},y^{(1)}),(z^{(2)},y^{(2)}),\cdots,(z^{(m)},y^{(m)})\}$
  - آزمایش فرضیه با استفاده از مجموعه آزمایشی: نگاشت $x_{test}^{(i)}$ به $z_{test}^{(i)}$ و محاسبه $h_\theta(z)$ برای هر یک از داده‌های مجموعه آزمایشی.

- یک پرسش مهم. اگر فرآیند فوق را بدون استفاده از PCA انجام دهیم چه می‌شود؟
  - همواره ابتدا فرآیند بالا را بدون استفاده از PCA انجام دهید.
  - اگر به پاسخ مطلوب نرسیدید، آنگاه استفاده از PCA را آزمایش کنید.

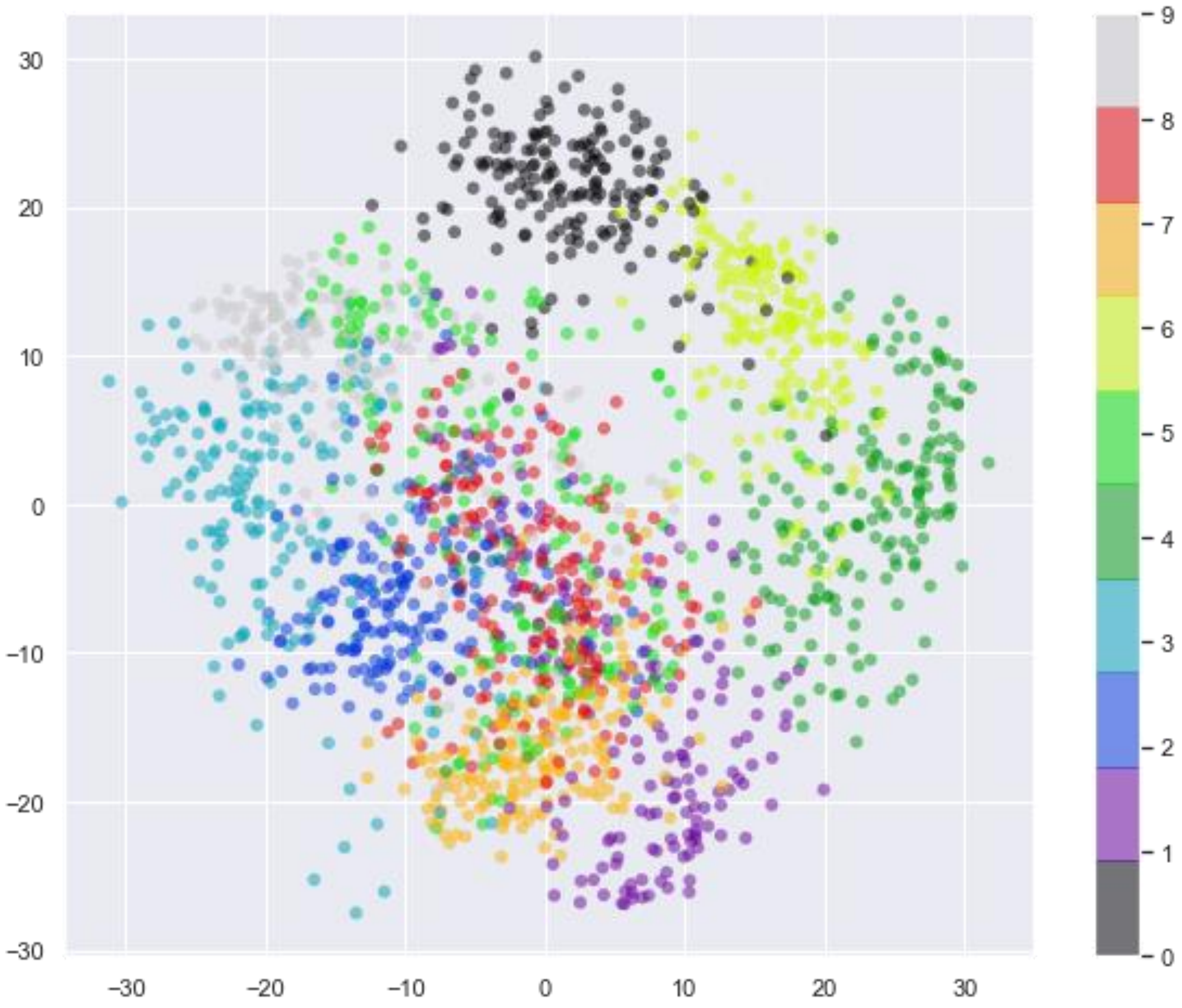
کاربردها

کاربردهای PCA

- به تصویر کشیدن داده‌ها.
  - انتخاب تعداد مؤلفه‌ها: $k = 2$ یا $k = 3$

- فشرده‌سازی داده‌ها.
  - کاهش حافظه مورد نیاز برای ذخیره‌سازی داده‌ها
  - افزایش سرعت اجرای الگوریتم یادگیری
  - انتخاب تعداد مؤلفه‌ها: بر اساس درصد واریانس حفظ شده

کاربردها: به تصویر کشیدن داده‌ها



نگاشت داده‌ها از فضای ۶۴ بعدی به فضای ۲ بعدی به منظور به تصویر کشیدن و درک بهتر داده‌ها

کاربردها: فشرده‌سازی

k = 100, variance = 0.91

Original Data

کاربردها: فشرده‌سازی

k = 150, variance = 0.95

Original Data

کاربردها: فشرده‌سازی

k = 200, variance = 0.97

Original Data

کاربردها: فشرده‌سازی

- مجموعه آموزشی.
  - ۴۰۰ تصویر چهره (خاکستری)
- ابعاد تصاویر.
  - ۶۴ در ۶۴ پیکسل
- تعداد ویژگی‌ها.
  - ۴۰۹۶ ویژگی

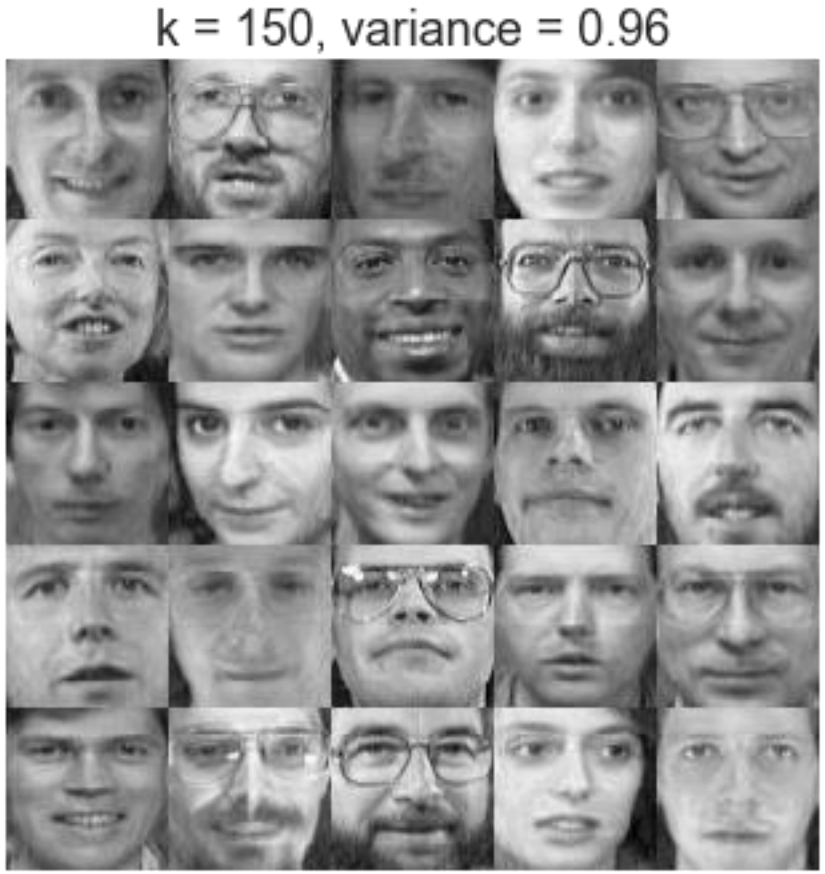
کاربردها: فشرده‌سازی

k = 50, variance = 0.87

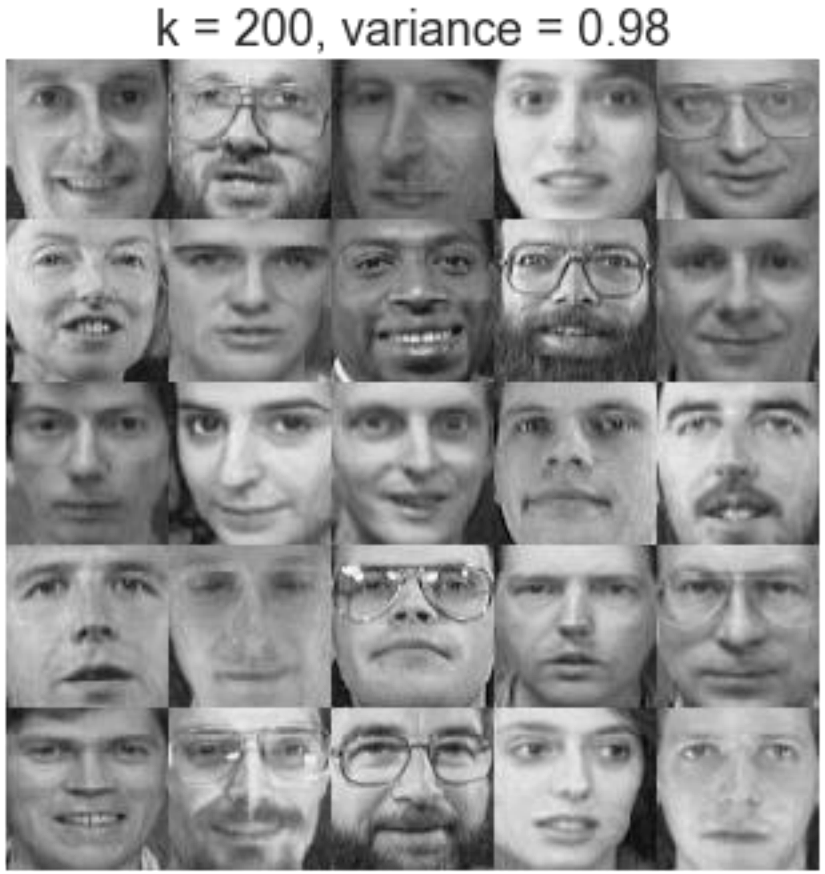
Original Faces

کاربردها: فشرده‌سازی

کاربردها: فشرده‌سازی



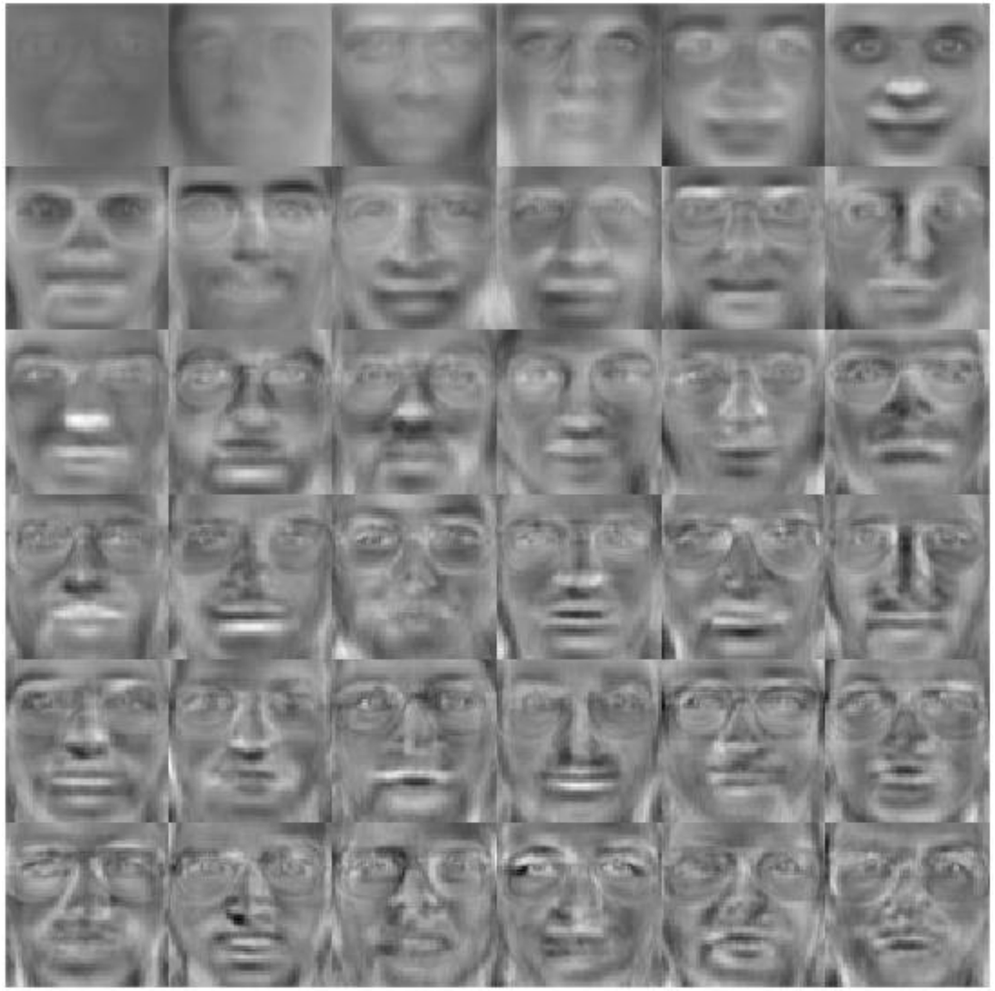
کاربردها: فشرده‌سازی



کاربردها: فشرده‌سازی (مؤلفه‌های اصلی)

k = 25, variance = 0.79

k = 36, variance = 0.84



کاربردها: فشرده‌سازی (بازسازی تصاویر)

Principal Components

k = 25, variance = 0.79

Original face

کاربردها: فشرده‌سازی (بازسازی تصاویر)

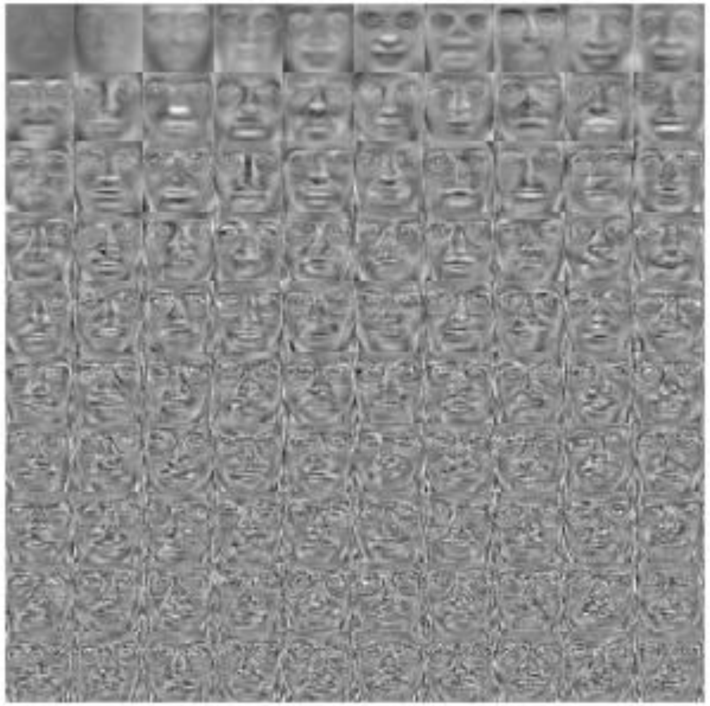
Principal Components

k = 36, variance = 0.84

Original face

کاربردها: فشرده‌سازی (بازسازی تصاویر)

Principal Components



k = 100, variance = 0.93

Original face

پیوست ۱: تجزیه مقادیر منفرد

تجزیه مقادیر منفرد

- انگیزه.
  - ساده‌سازی داده‌ها
  - حذف نویز و افزونگی
  - بهبود نتایج الگوریتم

- کاربردهای مثالی.
  - جستجو و بازیابی اطلاعات [شاخص گذاری معنایی نهان]
  - سیستم‌های توصیه‌گر

تجزیه مقادیر منفرد

- تجزیه مقادیر منفرد.

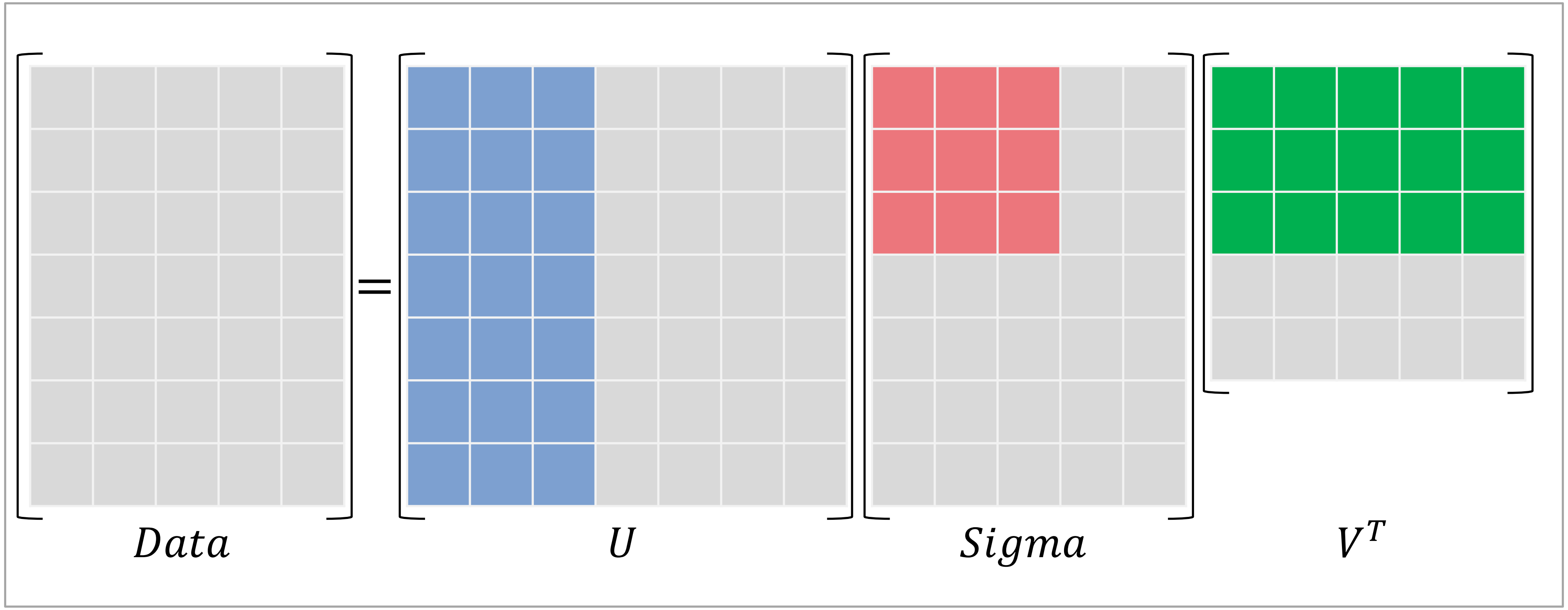
$$Data_{m\times n} = U_{m\times m}\Sigma_{m\times n}V_{n\times n}^{T}$$

ماتریس مقادیر منفرد

- ماتریس مقادیر منفرد.
  - یک ماتریس قطری که در آن مقادیر منفرد به صورت کاهشی مرتب هستند.
  - مقادیر منفرد از یک اندیس مانند r به بعد دارای مقدار صفر هستند.
  - مقادیر منفرد ریشه دوم مقادیر ویژه ماتریس $Data \times Data^T$ هستند.

تجزیه مقادیر منفرد: مثال

$$Data_{m\times n} \approx U_{m\times 3}\Sigma_{3\times 3}V_{3\times n}$$



تجزیه مقادیر منفرد: مثال


```python
X = np.array([[1, 1, 1, 0, 0],
              [2, 2, 2, 0, 0],
              [1, 1, 1, 0, 0],
              [5, 5, 5, 0, 0],
              [1, 1, 0, 2, 2],
              [0, 0, 0, 3, 3],
              [0, 0, 0, 1, 1]])

# Singular Value Decomposition
U, Sigma, VT = svd(X)
print(Sigma)
```

```
[9.72e+00 5.29e+00 6.84e-01 4.12e-16 1.36e-16]
```

داده‌های اولیه

1	1	1	0	0
2	2	2	0	0
1	1	1	0	0
5	5	5	0	0
1	1	0	2	2
0	0	0	3	3
0	0	0	1	1

تجزیه مقادیر منفرد: مثال


```python
X_approx = U[:, :1] @ np.diag(Sigma)[:1, :1] @ VT[:1, :]

print("SSE = {:.2f}".format(np.linalg.norm(X - X_approx) ** 2))
```

$SSE \approx 28.5$

```python
X_approx = U[:, :2] @ np.diag(Sigma)[:2, :2] @ VT[:2, :]

print("SSE = {:.2f}".format(np.linalg.norm(X - X_approx) ** 2))
```

$SSE \approx 0.47$

```python
X_approx = U[:, :3] @ np.diag(Sigma)[:3, :3] @ VT[:3, :]

print("SSE = {:.2f}".format(np.linalg.norm(X - X_approx) ** 2))
```

$SSE \approx 0.00$

تعیین تعداد مقادیر منفرد

- تعیین یک تعداد مناسب برای مقادیر منفرد.
  - مشابه با تعیین تعداد مؤلفه‌های اصلی

- یک روش تجربی. انتخاب کوچک‌ترین k به طوری که:

$$\frac{\sum_{i=1}^{k} s_{ii}^2}{\sum_{i=1}^{n} s_{ii}^2} \geq 0.90$$

9.72	0	0	0	0
0	5.29	0	0	0
0	0	0.68	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

$k = 1.\ energy = 0.768$

$k = 2.\ energy = 0.996$

$k = 3.\ energy = 1.000$

پیوست ۲: مقادیر ویژه و بردارهای ویژه

مقادیر ویژه و بردارهای ویژه

- حاصل‌ضرب ماتریس مربعی A را در بردار x در نظر بگیرید:

$$Ax = y$$

- ماتریس A همانند یک تابع بردار x را به بردار جدید y تبدیل می‌کند.

- بردار ویژه. بردار x یک بردار ویژه است اگر با بردار Ax موازی باشد:

$$\boxed{Ax = \lambda x}$$

مقدار ویژه

مقدار ویژه

- مقدار ویژه. در رابطه بالا، λ مقدار ویژه متناظر با بردار ویژه x است.

مقادیر ویژه و بردارهای ویژه

- مثال. اگر A یک ماتریس جایگشت به صورت زیر باشد:

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- در این صورت:

$$A\begin{bmatrix} 1 \\ 1 \end{bmatrix} = 1 \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow \lambda = 1$$

متعامد

$$A\begin{bmatrix} -1 \\ 1 \end{bmatrix} = (-1) \times \begin{bmatrix} -1 \\ 1 \end{bmatrix} \Rightarrow \lambda = -1$$

$$trace(A) = 0 + 0 = 1 + (-1)$$

- توجه. مجموع مقادیر ویژه با مجموع عناصر قطر اصلی (اثر) برابر است.

محاسبه مقادیر ویژه و بردارهای ویژه

- محاسبه مقادیر ویژه.

$$Ax = \lambda x \Rightarrow (A - \lambda I)x = 0$$

- بنابراین ماتریس $A - \lambda x$ یک ماتریس منفرد است. [زیرا بردار پوچ دارد]

- در نتیجه:

$$\boxed{det(A - \lambda I) = 0}$$

- مثال. محاسبه مقادیر ویژه

$$A = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$$

$$\det(A - \lambda I) = \det\left(\begin{bmatrix} 3-\lambda & 1 \\ 1 & 3-\lambda \end{bmatrix}\right) = \lambda^2 - \overset{\text{trace}(A)}{6}\lambda + \overset{\det(A)}{8} = 0 \Rightarrow \lambda = 4,2$$

محاسبه مقادیر ویژه و بردارهای ویژه

- محاسبه بردارهای ویژه.

- مثال.

$$A = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \Rightarrow \lambda_1 = 4, \lambda_2 = 2$$

$$(A - 4I)x_1 = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} x_1 = 0 \Rightarrow x_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$(A - 2I)x_2 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} x_2 = 0 \Rightarrow x_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

- مشاهده.

$$Ax = \lambda x \Rightarrow (A + 3I)x = Ax + 3x = \lambda x + 3x = (\lambda + 3)x$$

تجزیه ماتریس A: قطری سازی

- فرض کنید S ماتریسی باشد که ستون‌های آن بردارهای ویژه ماتریس A هستند.

$$
\begin{aligned}
AS &= A\begin{bmatrix} | & | & \cdots & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & \cdots & | \end{bmatrix} \\
&= \begin{bmatrix} | & | & \cdots & | \\ \lambda_1 x_1 & \lambda_2 x_2 & \cdots & \lambda_n x_n \\ | & | & \cdots & | \end{bmatrix} \\
&= \begin{bmatrix} | & | & \cdots & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & \cdots & | \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \\
&= S\Lambda
\end{aligned}
$$

$$AS = S\Lambda \Rightarrow S^{-1}AS = \Lambda$$

$$AS = S\Lambda \Rightarrow A = S\Lambda S^{-1}$$

تجزیه ماتریس A: قطری سازی

- مشاهده. اگر ماتریس A را به توان دو برسانیم، مقادیر ویژه به توان دو می‌رسند و بردارهای ویژه تغییر نمی‌کنند.

$$A = S\Lambda S^{-1} \Rightarrow A^2 = S\Lambda S^{-1}\, S\Lambda S^{-1} = S\Lambda^2 S^{-1}$$

- به طور کلی.

$$A = S\Lambda S^{-1} \Rightarrow A^k = S\Lambda^k S^{-1}$$

تجزیه ماتریس متقارن

- در یک ماتریس متقارن حقیقی:
  - مقادیر ویژه حقیقی هستند.
  - بردارهای ویژه متعامد نرمال هستند.

$$A = Q\Lambda Q^{-1} = Q\Lambda Q^T$$

$$A = \begin{bmatrix} | & | & \cdots & | \\ q_1 & q_2 & \cdots & q_n \\ | & | & \cdots & | \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} - & q_1^T & - \\ - & q_2^T & - \\ \vdots & \vdots & \vdots \\ - & q_n^T & - \end{bmatrix}$$
$$= \lambda_1 q_1 q_1^T + \lambda_2 q_2 q_2^T + \cdots + \lambda_n q_n q_n^T$$

- مشاهده. هر ماتریس متقارن ترکیب خطی یک مجموعه از ماتریس‌های پروجکشن متعامد است.

ماتریس‌های مثبت معین متقارن

- ماتریس مثبت معین. ماتریس A مثبت معین است اگر به ازای هر بردار غیر صفر مانند x:

$$x^T A x > 0$$

- در یک ماتریس مثبت معین متقارن:
  - مقادیر ویژه همگی مثبت هستند.
  - محورها همگی مثبت هستند.
  - دترمینان‌ها همگی (دترمینان زیرماتریس‌های مقدم) مثبت هستند.

- مثال.

$$A = \begin{bmatrix} 5 & 2 \\ 2 & 3 \end{bmatrix} \Rightarrow \lambda = 4 \pm \sqrt{5}, p_1 = 5, p_2 = \frac{11}{5}, \det(A) = 11$$