

---

# 第二次作业

---

李子龙

上海交通大学

计算机科学与工程系

logcreative@outlook.com

## 1 PCA

### 1.1 特征值分解

---

**Algorithm 1:** 特征值分解 PCA

---

**Input:** 数据集  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}, \mathbf{x}_t \in \mathbb{R}^{n \times 1}$

**Output:** 主成分  $\mathbf{w}$

```
1 计算平均值  $\mu \leftarrow \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$ ;  
2 foreach  $i \leftarrow 1$  to  $N$  do  
3    $\mathbf{x}_i \leftarrow \mathbf{x}_i - \mu$ ;  
4 计算散度矩阵  $C \leftarrow \mathbf{X}\mathbf{X}^T$ ;  
5 特征值分解求  $C$  的特征值  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$  与对应的特征向量  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ ;  
6 选取最大的特征值对应的特征向量与数据的乘积即为主成分  $\mathbf{w} \leftarrow \mathbf{v}_1^T \mathbf{X}$ ;  
7 return  $\mathbf{w}$ ;
```

---

#### 优点

1. 简单易实现。
2. 解除线性相关。

#### 缺点

1. 需要的内存大，需要先计算散度矩阵，当样本数量很大时，这一步消耗的时间复杂度比较高。
2. 计算散度矩阵这一步在数据量较少时可能会丢失精度。
3. 只能压缩一个方向（行或列）。

## 1.2 奇异值分解

---

**Algorithm 2:** 奇异值分解

---

**Input:** 数据集  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}, \mathbf{x}_t \in \mathbb{R}^{n \times 1}$

**Output:** 主成分  $\mathbf{w}$

- 1 计算平均值  $\mu \leftarrow \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$ ;
  - 2 **foreach**  $i \leftarrow 1$  **to**  $N$  **do**
  - 3      $\mathbf{x}_i \leftarrow \mathbf{x}_i - \mu$ ;
  - 4 奇异值分解  $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ ;
  - 5 两边同乘  $\mathbf{U}^T$ ,  $\mathbf{U}^T\mathbf{X} = \mathbf{\Sigma}\mathbf{V}^T$  得到压缩数据;
  - 6 选取  $\mathbf{\Sigma}\mathbf{V}^T$  中最大的那一个奇异值（习惯上应为左上角的值）对应的向量（一般为第一行）即为主成分  $\mathbf{w}$ ;
  - 7 **return**  $\mathbf{w}$ ;
- 

### 优点

1. 可以直接对非方阵  $\mathbf{X}$  进行奇异值分解，而特征值分解需要分解方阵  $\mathbf{X}\mathbf{X}^T$ 。可免去计算  $\mathbf{X}\mathbf{X}^T$  的中间步骤。
2. 计算奇异值已经有快速地数值算法，在需要在时间空间与精度直接抉择时，可以选择后者直接取出较大的奇异值，精度上的折中是可以接受的。
3. 既能压缩行又能压缩列。

### 缺点

1. SVD 算法需要实现，算法实现难度比特征值分解大。
2. 分解后的矩阵缺少可解释性。

## 2 FA

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})} \quad (2.1)$$

$$= \frac{G(\mathbf{x}|\mathbf{A}\mathbf{y} + \mu, \mathbf{\Sigma}_e)G(\mathbf{y}|0, \mathbf{\Sigma}_y)}{p(\mathbf{A}\mathbf{y} + \mu + \mathbf{e})} \quad (2.2)$$

$$= \frac{G(\mathbf{x}|\mathbf{A}\mathbf{y} + \mu, \mathbf{\Sigma}_e)G(\mathbf{y}|0, \mathbf{\Sigma}_y)}{G(\mathbf{y}|\mu + \mu_e, \mathbf{A}\mathbf{\Sigma}_y\mathbf{A}^T + \mathbf{\Sigma}_e)} \quad (2.3)$$

公式 (2.1) 采用了贝叶斯规则，公式 (2.2) 采用了已知条件，公式 (2.3) 由下面的方式推导：

$$\begin{aligned}
E(\mathbf{x}) &= E(\mathbf{A}\mathbf{y} + \mu + \mathbf{e}) \\
&= 0 + \mu + E(\mathbf{e}) \\
&= \mu + \mu_e \\
Cov(\mathbf{x}) &= Cov(\mathbf{A}\mathbf{y} + \mu + \mathbf{e}) \\
&= E((\mathbf{A}\mathbf{y} + \mu + \mathbf{e} - E(\mathbf{x}))(\mathbf{A}\mathbf{y} + \mu + \mathbf{e} - E(\mathbf{x}))^T) \\
&= E((\mathbf{A}\mathbf{y} + (\mathbf{e} - \mu_e))(\mathbf{A}\mathbf{y} + (\mathbf{e} - \mu_e))^T) \\
&= E(\mathbf{A}\mathbf{y}\mathbf{y}^T\mathbf{A}^T + \mathbf{A}\mathbf{y}(\mathbf{e} - \mu_e) + (\mathbf{e} - \mu_e)(\mathbf{A}\mathbf{y})^T + (\mathbf{e} - \mu_e)(\mathbf{e} - \mu_e)^T) \\
&= \mathbf{A}\Sigma_y\mathbf{A}^T + \Sigma_e
\end{aligned}$$

而公式 (2.3) 就是答案。

### 3 ICA

如果  $\mathbf{s}$  被分解为两个高斯分布  $N(0, \mathbf{I})$ ，那么  $\mathbf{s}$  也将满足高斯分布。接收信号（作为源信号的线性组合） $\mathbf{x} = \mathbf{A}\mathbf{s}$  也将是高斯分布， $E(\mathbf{s}_i^2) = 1$ 。这将导致

$$\begin{aligned}
E(\mathbf{x}) &= 0 \\
E(\mathbf{x}\mathbf{x}^T) &= E(\mathbf{A}\mathbf{s}\mathbf{s}^T\mathbf{A}^T) = \mathbf{A}\mathbf{A}^T
\end{aligned}$$

如果对  $\mathbf{A}$  右乘一个正交矩阵  $\mathbf{C}$  得到  $\mathbf{A}' = \mathbf{A}\mathbf{C}$ ，那么得到的新接收信号  $\mathbf{x}' = \mathbf{A}'\mathbf{s}$  也将是高斯分布， $\mathbf{C}\mathbf{C}^T = \mathbf{I}$ ，而且

$$\begin{aligned}
E(\mathbf{x}') &= 0 \\
E(\mathbf{x}'(\mathbf{x}')^T) &= E(\mathbf{A}\mathbf{C}\mathbf{s}\mathbf{s}^T(\mathbf{A}\mathbf{C})^T) = \mathbf{A}\mathbf{C}\mathbf{C}^T\mathbf{A}^T = \mathbf{A}\mathbf{A}^T
\end{aligned}$$

将无法通过不同的混淆矩阵来区分信号，也就无法预测  $\mathbf{A}$  的信息，也就无法通过接收信号分离出源信号的信息。

另一方面，中心极限定理告诉我们，多个独立的自由变量之和近似服从高斯分布。考虑最后的复原过程

$$\mathbf{y} = \mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{A}\mathbf{s} = \mathbf{z}^T\mathbf{s}$$

这样  $\mathbf{y}$  就会变成  $\mathbf{s}_i$  的线性组合，它将比任何一个分量都接近于高斯，这种复原结果将与之前的源分量不应当高斯相违背，所以要寻找  $\mathbf{w}$  最大化该式的非高斯性。

所以对于希望分解为独立分量的 ICA 来说，最大化非高斯性是其原则之一。

## 4 FA 降维

### 4.1 数据生成

数据生成代码见 [src/datagen.py](#)，这里我们固定  $\mu = 0$ ，数据使用下面的方法生成：

$$\begin{aligned}\mathbf{y}_t &\sim G(\mathbf{y}|0, \mathbf{I}) \\ \mathbf{e}_t &\sim G(\mathbf{e}|0, \sigma^2 \mathbf{I}) \\ \mathbf{x}_t &= \mathbf{A}\mathbf{y}_t + \mathbf{e}_t\end{aligned}$$

其中  $\mathbf{A}$  随机选取于  $G(0, \mathbf{I})$ ，由数据生成开始时随机生成并固定。

### 4.2 模型选择

采用 `sklearn.decomposition` 中的 `FactorAnalysis` 进行分析，采用“二步法”：

$$\begin{aligned}m^* &= \arg \max_{m=1, \dots, M} J(m) \\ J_{\text{AIC}}(m) &= \ln [p(X|\hat{\Theta}_m)] - d_m \\ J_{\text{BIC}}(m) &= \ln [p(X|\hat{\Theta}_m)] - \frac{\ln N}{2} d_m\end{aligned}$$

其中对于 FA 的自由参数个数

$$d_m = nm + 1 - (1 + 2 + \dots + m) = nm + 1 - \frac{m(m-1)}{2}$$

### 4.3 测试结果

测试将对每一对指标取 10 个随机种子计算预测出的  $\hat{m}$  与真实  $m$  的差别，使用公式 4.1 计算得分，这个公式将会显示出平均而言是预测高了还是预测低了，这个值越接近于 0 越好。

$$\text{Score} = \frac{\hat{m} - m}{m} \quad (4.1)$$

如图 1 所示，对于不同的  $(n, m)$  二元组，在  $m$  比较大  $n$  比较小时（左下角），AIC 和 BIC 预测表现都比较差，但是 AIC 略胜一筹；而  $m$  比较小和  $n$  比较大的时候（右上

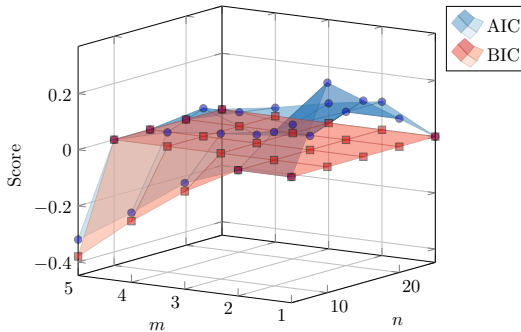


图 1:  $n$ - $m$  平均得分统计图

表 1: 参数范围

类型	范围
$n$	[5,10,15,20,25]
$m$	[1,2,3,4,5]
随机种子个数	10
样本数量 $N$	200
噪声方差 $\sigma^2$	0.1

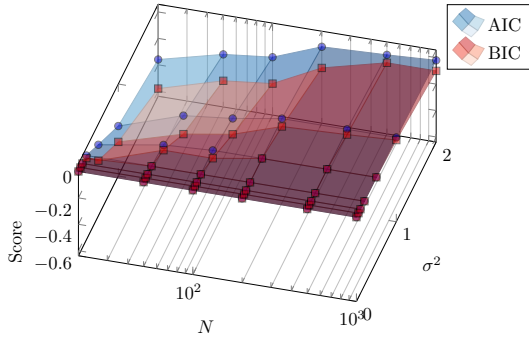


图 2:  $N$ - $\sigma^2$  平均得分统计图

表 2: 参数范围

类型	范围
样本数量 $N$	[20, 50, 100, 200, 500, 1000]
噪声方差 $\sigma^2$	[0, 0.05, 0.1, 0.2, 0.5, 1, 2]
随机种子个数	10
$n$	10
$m$	5

角), BIC 表现就比较好, AIC 往往会预测得略高一些。可见 AIC 更适合因素较多的时候, 而 BIC 更适合因素单一的情况。

如图 2 所示, 对于不同的  $(N, \sigma^2)$  二元组, 在这个范围内, AIC 都要比 BIC 要好一些 (主要是  $m$  比较大, 在上一点已论证), 而样本数量越少, 噪声越大 (左上角), 两者表现就会越差, 会预测出较少的因素, BIC 的表现会下降得更快一些。

#### 4.4 模型选择小结

单一元素情况, BIC 更好一些; 多元情况, AIC 更好一些。样本数量越多, 噪声越小, 模型选择的表现就越好。