为了更合理地计算运动员的历史成绩，需要综合考虑以下因素：

---

### \*\*1. 核心设计原则\*\*

1. \*\*时间衰减权重\*\*：近期成绩比早期成绩更重要。

2. \*\*职业生涯阶段划分\*\*：上升期、巅峰期、下降期的权重不同。

3. \*\*退役概率\*\*：长时间未参赛的运动员应降低权重。

4. \*\*奖牌质量\*\*：金牌、银牌、铜牌的权重不同。

5. \*\*标准化处理\*\*：避免参赛次数带来的偏差。

---

### \*\*2. 具体计算步骤\*\*

#### \*\*(1) 数据预处理\*\*

假设原始数据格式如下（示例）：

| Athlete\_ID | Year | Medal | Event |

|------------|------|-------------|--------------|

| 1 | 2012 | Gold | Swimming |

| 1 | 2016 | Silver | Swimming |

| 2 | 2020 | Bronze | Gymnastics |

\*\*代码实现：\*\*

```python

import pandas as pd

# 假设 medal\_df 是包含运动员历届奖牌的数据框

# 添加奖牌数值权重

medal\_points = {

'Gold': 3,

'Silver': 2,

'Bronze': 1,

'No medal': 0

}

medal\_df['Medal\_Points'] = medal\_df['Medal'].map(medal\_points)

```

#### \*\*(2) 时间衰减权重\*\*

使用指数衰减函数赋予近期比赛更高权重：

\[

\text{时间权重} = e^{-\lambda \cdot (T\_{\text{current}} - T\_{\text{year}})}

\]

其中：

- \( \lambda \) 是衰减率（例如 \( \lambda = 0.3 \)）。

- \( T\_{\text{current}} \) 是当前年份（如 2024）。

- \( T\_{\text{year}} \) 是比赛年份。

\*\*代码实现：\*\*

```python

current\_year = 2024

decay\_rate = 0.3

medal\_df['Time\_Weight'] = np.exp(-decay\_rate \* (current\_year - medal\_df['Year']))

```

#### \*\*(3) 职业生涯阶段划分\*\*

根据参赛频率和成绩动态调整权重：

- \*\*上升期\*\*（首次参赛后前2届）：权重 = 1.0

- \*\*巅峰期\*\*（奖牌数最多的连续2届）：权重 = 1.5

- \*\*下降期\*\*（巅峰期后）：权重 = 0.8

\*\*代码实现：\*\*

```python

# 按运动员分组计算职业生涯阶段

def calculate\_career\_phase(group):

group = group.sort\_values('Year')

group['Career\_Phase'] = 'Decline'

if len(group) >= 3:

# 找到奖牌数最多的连续2届

max\_points = 0

peak\_start = 0

for i in range(len(group)-1):

points = group['Medal\_Points'].iloc[i] + group['Medal\_Points'].iloc[i+1]

if points > max\_points:

max\_points = points

peak\_start = i

# 标记阶段

group.iloc[:2, group.columns.get\_loc('Career\_Phase')] = 'Rise'

group.iloc[peak\_start:peak\_start+2, group.columns.get\_loc('Career\_Phase')] = 'Peak'

return group

medal\_df = medal\_df.groupby('Athlete\_ID').apply(calculate\_career\_phase)

# 分配阶段权重

phase\_weights = {

'Rise': 1.0,

'Peak': 1.5,

'Decline': 0.8

}

medal\_df['Phase\_Weight'] = medal\_df['Career\_Phase'].map(phase\_weights)

```

#### \*\*(4) 退役概率调整\*\*

如果运动员超过2届未参赛，则标记为“可能退役”：

\[

\text{活跃度权重} = \begin{cases}

1.0 & \text{最近一届参赛在8年内} \\

0.5 & \text{否则}

\end{cases}

\]

\*\*代码实现：\*\*

```python

# 计算每个运动员最近参赛年份

recent\_year = medal\_df.groupby('Athlete\_ID')['Year'].max().reset\_index()

recent\_year.columns = ['Athlete\_ID', 'Last\_Year']

# 合并到原始数据

medal\_df = medal\_df.merge(recent\_year, on='Athlete\_ID')

# 计算活跃度权重

medal\_df['Activity\_Weight'] = np.where(

(current\_year - medal\_df['Last\_Year']) <= 8,

1.0,

0.5

)

```

#### \*\*(5) 综合历史成绩得分\*\*

将时间权重、阶段权重、活跃度权重和奖牌得分结合：

\[

\text{历史得分} = \sum (\text{Medal\\_Points} \times \text{Time\\_Weight} \times \text{Phase\\_Weight} \times \text{Activity\\_Weight})

\]

\*\*代码实现：\*\*

```python

medal\_df['Historical\_Score'] = (

medal\_df['Medal\_Points'] \*

medal\_df['Time\_Weight'] \*

medal\_df['Phase\_Weight'] \*

medal\_df['Activity\_Weight']

)

# 按运动员汇总得分

athlete\_history = medal\_df.groupby('Athlete\_ID')['Historical\_Score'].sum().reset\_index()

```

#### \*\*(6) 标准化处理\*\*

使用 Z-score 标准化：

\[

\text{标准化得分} = \frac{\text{历史得分} - \mu}{\sigma}

\]

\*\*代码实现：\*\*

```python

mean\_score = athlete\_history['Historical\_Score'].mean()

std\_score = athlete\_history['Historical\_Score'].std()

athlete\_history['Normalized\_Score'] = (athlete\_history['Historical\_Score'] - mean\_score) / std\_score

```

---

### \*\*3. 在贝叶斯模型中使用\*\*

将标准化后的得分作为固定效应的先验均值：

```python

with pm.Model() as model:

# 从 athlete\_history 获取标准化得分

normalized\_scores = athlete\_history.set\_index('Athlete\_ID')['Normalized\_Score']

# 运动员固定效应（历史成绩作为先验均值）

alpha\_athlete = pm.Normal(

'alpha\_athlete',

mu=normalized\_scores.loc[athelete\_df['Athlete\_ID']].values, # 对齐当前数据

sigma=0.5, # 根据实际情况调整

shape=len(athlete\_history)

)

# 后续模型定义...

```

---

### \*\*4. 动态更新机制\*\*

- \*\*每年更新\*\*：新增一届奥运会数据后，重新计算时间权重和活跃度。

- \*\*退役标记\*\*：如果某运动员超过3届未参赛，直接将其固定效应设为0。

---

### \*\*5. 总结\*\*

通过时间衰减、职业生涯阶段、活跃度调整和标准化，可以合理量化运动员的历史成绩。这种方法不仅反映了运动员的长期表现，还动态捕捉了其状态变化和退役可能性，适合作为贝叶斯模型中的固定效应。