

Problem Chosen

C

2026
MCM/ICM
Summary Sheet

Team Control Number

1234

XXXXXXXX

Summary

Keywords: k222eyword1; keyword2

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 3 |
| 1.1 | Background | 3 |
| 1.2 | The Description of the Problem | 3 |
| 1.3 | Our work | 3 |
| 2 | Assumptions | 4 |
| 3 | Notations | 5 |
| 4 | 问题 1: 观众投票估计模型 | 6 |
| 4.1 | 问题剖析与建模策略 | 6 |
| 4.2 | 投票组合方法 | 6 |
| 4.3 | 基线模型: 幂律投票假设 | 6 |
| 4.4 | 精确反推模型: 约束优化框架 | 7 |
| 4.4.1 | 投票偏好因子的学习 | 7 |
| 4.4.2 | 优化问题的数学构建 | 7 |
| 4.4.3 | SLSQP 算法求解 | 8 |
| 4.5 | 模型评估与集成策略 | 8 |
| 4.5.1 | 一致性度量体系 | 8 |
| 4.5.2 | 集成策略的实现 | 9 |
| 4.5.3 | 不确定性的传播 | 9 |
| 4.5.4 | 双视角可解释性 | 9 |
| 5 | 问题 2: 投票方法对比与推荐 | 10 |
| 5.1 | 子问题 2.1: 两种方法的结果对比与观众投票偏向性 | 10 |
| 5.1.1 | 问题陈述 | 10 |
| 5.1.2 | 分析方法 | 10 |
| 5.1.3 | 结果 1: 方法一致性分析 | 10 |
| 5.1.4 | 结果 2: 观众投票偏向性分析 | 10 |

| | | |
|----------|--|-----------|
| 5.1.5 | 子问题 2.1 的明确回答 | 11 |
| 5.2 | 子问题 2.2: 争议性选手案例分析 | 11 |
| 5.2.1 | 问题陈述 | 11 |
| 5.2.2 | 四位争议选手的数据总结 | 11 |
| 5.2.3 | 案例 1: Bristol Palin - 方法影响最显著 | 11 |
| 5.2.4 | 案例 2、3、4: Jerry Rice、Billy Ray Cyrus、Bobby Bones | 12 |
| 5.2.5 | 评委裁决机制的影响分析 | 12 |
| 5.2.6 | 子问题 2.2 的明确回答 | 13 |
| 5.3 | 子问题 2.3: 方法推荐与理由 | 13 |
| 5.3.1 | 问题陈述 | 13 |
| 5.3.2 | 推荐方案与理由 | 13 |
| 5.3.3 | 子问题 2.3 的明确回答 | 14 |
| 6 | Sensitivity Analysis | 14 |
| 7 | Model Evaluation and Promotion | 14 |
| 7.1 | Model Evaluation | 14 |
| 7.1.1 | Advantages | 14 |
| 7.1.2 | Limitations | 14 |
| 7.2 | Future Work | 14 |
| 7.2.1 | Model extension | 14 |
| 7.2.2 | Model application | 14 |
| 8 | Conclusions | 14 |

1 Introduction

1.1 Background

1.2 The Description of the Problem

本问题聚焦于《与星共舞》(Dancing with the Stars, DWTS) 节目的投票机制分析。在该竞技真人秀中,名人选手 (celebrity contestants) 与专业舞者 (professional dancers) 配对表演,通过评委打分 (judge scores) 和观众投票 (fan votes) 的组合决定每周淘汰结果。然而,观众投票数据作为商业机密从未公开,且节目在 34 季中采用了不同的投票组合规则 (voting combination methods)。核心挑战在于:在观众投票未知的情况下,如何评估投票系统的公平性并提出优化方案。

具体任务包括:

- 观众投票估算:构建数学模型,基于评委得分、选手特征和淘汰结果反向推断每位选手的观众投票数。评估估算结果与实际淘汰情况的一致性 (consistency),并量化估算的不确定性 (uncertainty measures)。
- 投票方法对比:对比分析节目使用的两种组合方法——基于排名法 (rank-based method) 和基于百分比法 (percentage-based method) 在各赛季产生的结果差异。针对争议案例 (如 Jerry Rice、Bobby Bones 等评委低分但观众高票的选手),评估方法选择和评委复议机制 (judges choosing from bottom two) 对最终排名的影响,并推荐未来赛季的最优方法。
- 影响因素分析:构建模型量化专业舞者经验、名人年龄 (age)、行业背景 (industry) 等因素对比赛结果的影响。分析这些因素对评委得分和观众投票的作用机制是否存在差异。
- 新系统设计:提出一种更“公平”(fair) 或更具观赏性 (exciting) 的投票组合系统。通过历史数据模拟验证新系统的有效性,并为制作方提供可操作的实施建议。
- 报告撰写:提交不超过 25 页的分析报告及 1-2 页备忘录,为 DWTS 制作方提供关于投票组合方式影响的总结和未来赛季的决策建议。

1.3 Our work

2 Assumptions

Assumption 1: The light attenuation and scattering in the underwater environment adhere to the Jaffe-McGlamery imaging model, where the propagation of light follows an exponential decay law.

3 Notations

The core symbols and their definitions used in this study are summarized in Table 1, providing an overview of the key parameters and their related meanings.

Table 1: Notations used in this literature

| Symbol | Description |
|----------------------|-------------------------------------|
| n | 第 w 周剩余选手数量 |
| S_i | 选手 i 的评委总分 |
| V_i | 选手 i 的估计观众投票数 |
| E_i | 淘汰指示器：若选手 i 被淘汰则 $E_i = 1$ ，否则为 0 |
| \mathcal{M} | 投票组合方法：“rank”（排名法）或“percent”（百分比法） |
| C_i | 选手 i 在方法 \mathcal{M} 下的综合得分 |
| α | 投票与评委得分关系的幂律指数 |
| θ | 先验偏好因子（舞伴效应、行业效应） |
| R_i^{judge} | 选手 i 的评委得分排名 |
| R_i^{fan} | 选手 i 的观众投票排名 |
| β_p | 专业舞伴 p 的投票偏好效应 |
| γ_d | 名人行业 d 的投票偏好效应 |
| B_i^w | 选手 i 在第 w 周的存活提升值 |
| \mathbf{v} | 观众投票份额向量 (v_1, \dots, v_n) |

4 问题 1：观众投票估计模型

4.1 问题剖析与建模策略

问题 1 本质上是一个欠定逆问题（underdetermined inverse problem）：在观众投票这一隐变量缺失的情况下，从淘汰结果和评委得分的联合观测中进行统计推断。这类问题在计量经济学和信号处理领域广泛存在，其核心挑战在于解的非唯一性与模型的可识别性（identifiability）。

问题的数学结构。设第 w 周有 n 名选手，评委得分 $\mathbf{S} = (S_1, \dots, S_n)$ 可观测，观众投票 $\mathbf{V} = (V_1, \dots, V_n)$ 不可观测，淘汰结果 E 已知。问题可形式化为：给定映射 $f: (\mathbf{S}, \mathbf{V}) \rightarrow E$ 和观测 (\mathbf{S}, E) ，反推 \mathbf{V} 。由于 $|\mathbf{V}| = n$ 个未知数仅对应 1 个淘汰约束，系统严重欠定（ $n \gg 1$ ）。

可识别性困境。多个投票分布可产生相同淘汰结果，导致解不唯一。传统方法有二：（1）施加正则化约束（如最大熵、稀疏性），但缺乏投票行为的先验支持；（2）引入辅助假设（如投票与得分的参数关系），但可能过度简化现实。我们的策略是将两者结合——用行为模型提供软约束，用淘汰结果施加硬约束，在约束空间中搜索最符合经验规律的解。

方法演变的影响。第 28 季前后投票组合规则的变化（排名法 \leftrightarrow 百分比法）改变了 f 的函数形式，使得相同的 (\mathbf{S}, \mathbf{V}) 可能导致不同的 E 。这要求模型具备规则自适应性，能够在统一框架下处理两种机制。

基于上述分析，我们设计双层贝叶斯框架：基线层建立 $P(\mathbf{V}|\mathbf{S})$ 的先验分布，反推层通过 $P(\mathbf{V}|\mathbf{S}, E)$ 进行后验推断。这种分层设计既保证了理论严谨性（后验一致性），又具备实践可行性（先验可解释性）。

4.2 投票组合方法

不同赛季使用的两种投票组合方法为（相关符号定义见表 1）：

排名法（赛季 1-2、28-34）：

$$C_i^{\text{rank}} = R_i^{\text{judge}} + R_i^{\text{fan}} \quad (1)$$

其中 $R_i^{\text{judge}} = \text{rank}(-S_i)$ 和 $R_i^{\text{fan}} = \text{rank}(-V_i)$ 。综合排名最高（即 C_i^{rank} 最大，表现最差）的选手被淘汰。

百分比法（赛季 3-27）：

$$C_i^{\text{percent}} = \frac{S_i}{\sum_{j=1}^n S_j} + \frac{V_i}{\sum_{j=1}^n V_j} \quad (2)$$

综合百分比最低的选手被淘汰。

4.3 基线模型：幂律投票假设

观众投票与评委得分的关系既非完全独立，亦非严格线性。我们提出幂律假设捕捉这种非线性依赖：

$$V_i \propto S_i^\alpha, \quad \hat{V}_i = V_{\text{total}} \cdot \frac{S_i^\alpha}{\sum_{j=1}^n S_j^\alpha} \quad (3)$$

指数 α 的经济学解释：当 $\alpha > 1$ 时，高分选手获得超比例支持（“马太效应”）； $\alpha < 1$ 时体现“同情弱者”； $\alpha = 1$ 退化为比例模型。通过网格搜索最大化淘汰预测准确率，校准得 $\alpha^* \approx 1.2$ ，印证观众存在温和的“强者偏好”。

不确定性通过蒙特卡洛采样量化：向幂律预测注入 15% 高斯噪声，生成 100 个样本构建 95% 置信区间。这为后续分析提供了统计显著性检验的基础。

4.4 精确反推模型：约束优化框架

基线模型提供先验分布 $P(\mathbf{V}|\mathbf{S})$ ，但无法保证后验一致性。我们建立约束优化框架，从淘汰结果反推投票的后验分布 $P(\mathbf{V}|\mathbf{S}, E)$ 。

4.4.1 投票偏好因子的学习

在优化前，从全局数据学习系统性偏好。定义“存活提升” B_i^w 衡量评分排名与生存结果的错位程度：

$$B_i^w = \begin{cases} -(n - R_i^{\text{score}})/n & \text{若被淘汰（评分越高越意外）} \\ (R_i^{\text{score}} - 1)/n & \text{若存活（评分越低越意外）} \end{cases} \quad (4)$$

其中 R_i^{score} 为选手 i 的评分排名（1= 最高分）。

按专业舞伴 p 和名人行业 d 聚合提升值，得效应参数：

$$\beta_p = \frac{1}{|\mathcal{W}_p|} \sum_{w \in \mathcal{W}_p} B_i^w - \mu_{\text{overall}} \quad (5)$$

$$\gamma_d = \frac{1}{|\mathcal{W}_d|} \sum_{w \in \mathcal{W}_d} B_i^w - \mu_{\text{overall}} \quad (6)$$

结合标准化年龄，构建先验偏好因子：

$$\theta_i = \beta_{p(i)} + \gamma_{d(i)} + \delta_{\text{age}} \cdot \frac{\text{age}_i - \mu_{\text{age}}}{\sigma_{\text{age}}} \quad (7)$$

这些偏好因子识别了哪些舞伴、行业和年龄段的选手系统性地获得更多或更少的观众支持。正效应表示该因素带来额外投票支持，负效应则相反。

4.4.2 优化问题的数学构建

对投票份额 $\mathbf{v} = (v_1, \dots, v_n)$ （满足 $v_i \in [0, 1], \sum v_i = 1$ ），最小化目标函数：

$$\mathcal{J}(\mathbf{v}) = \|\mathbf{v} - \mathbf{v}^{\text{prior}}\|_2^2 + \lambda \cdot \mathcal{L}_{\text{elim}}(\mathbf{v}) \quad (8)$$

其中先验份额为：

$$\mathbf{v}^{\text{prior}} = \text{normalize} \left[\frac{\mathbf{S}}{\sum_j S_j} \odot (1 + \boldsymbol{\theta}) \right] \quad (9)$$

淘汰约束的软化处理。对于排名法，离散排名函数不可微，我们采用 **sigmoid** 软排名近似：

$$\text{SoftRank}_i = n - \sum_{j \neq i} \frac{1}{1 + \exp[(x_j - x_i)/T]} \quad (10)$$

其中 x 可以是分数或投票份额， T 为温度参数（代码中设为 0.1）。该函数近似计算“有多少人比 i 差”，温度越小越接近真实排名。

综合软排名为 $C_i^{\text{soft}} = \text{SoftRank}_i(\mathbf{S}) + \text{SoftRank}_i(\mathbf{v})$ 。淘汰约束惩罚项为：

$$\mathcal{L}_{\text{rank}}(\mathbf{v}) = \sum_{\substack{i:E_i=1 \\ j:E_j=0}} \max(0, C_j^{\text{soft}} - C_i^{\text{soft}} + \delta)^2 \quad (11)$$

其中边际 $\delta = 0.5$ 确保被淘汰者的软排名显著高于存活者。

对于百分比法，约束为：

$$\mathcal{L}_{\text{percent}}(\mathbf{v}) = \sum_{\substack{i:E_i=1 \\ j:E_j=0}} \max(0, C_i - C_j + \epsilon)^2 \quad (12)$$

其中 $C_i = S_i / \sum S_j + v_i$ ，边际 $\epsilon = 0.01$ 。

正则化项的作用。 $\|\mathbf{v} - \mathbf{v}^{\text{prior}}\|_2^2$ 防止过拟合，在众多满足约束的解中选择最接近先验的那个。权重 $\lambda = 1000$ 通过实验确定，平衡正则化与约束满足。

4.4.3 SLSQP 算法求解

采用序列最小二乘规划（SLSQP）求解约束优化。该算法适合处理等式约束 $\sum v_i = 1$ 和盒约束 $v_i \in [0.001, 0.999]$ ，基于梯度信息快速收敛。

初始值设为先验 $\mathbf{v}^{\text{prior}}$ ，优化参数为：最大迭代 500 次，收敛容差 10^{-8} 。成功收敛的判断是目标函数值低于阈值且淘汰约束得到满足。

失败案例分析。对于约束无法满足的周次（优化失败），模型回退到备用估计：

$$\mathbf{v}^{\text{fallback}} = \text{normalize} \left[\frac{\mathbf{S}}{\sum S_j} \odot (1 + 0.3 \cdot \boldsymbol{\theta}) \right] \quad (13)$$

这些失败案例往往对应评分与淘汰结果存在强矛盾（如高分选手被淘汰），或数据记录异常（多人同时淘汰/无人淘汰）。

4.5 模型评估与集成策略

4.5.1 一致性度量体系

评估投票估计需要多个互补指标。代码中实现了以下度量：

精确匹配准确率定义为：

$$\text{ACC}_{\text{exact}} = \frac{1}{W} \sum_{w=1}^W \mathbb{1}_{\{\text{predicted}(w) = \text{actual}(w)\}} \quad (14)$$

衡量能否精准预测每周被淘汰者，这是最严格的标准。

底部 N 准确率放宽判定条件，检验实际淘汰者是否落入预测的”危险区”（综合得分最低的若干名）。这更符合实际应用——关心”谁有风险”而非”谁一定淘汰”。

约束满足边际量化淘汰的”确定性”：

$$\text{Margin}_k = \min_{i:E_i=0} |C_k - C_i| \quad (15)$$

边际大表示淘汰显著，边际小则属”惊险淘汰”，投票的微小波动即可改变结果。

4.5.2 集成策略的实现

代码采用场景感知的模型选择机制：

淘汰周次：优先使用精确反推模型。运行 SLSQP 优化，若收敛则采用优化结果。

非淘汰周次：使用基线模型结合偏好因子。具体为：

$$\mathbf{v}^{\text{non-elim}} = 0.7 \cdot \mathbf{v}^{\text{baseline}} + 0.3 \cdot \mathbf{v}^{\text{prior}} \quad (16)$$

其中 $\mathbf{v}^{\text{baseline}}$ 为基线模型的幂律估计， $\mathbf{v}^{\text{prior}}$ 为带偏好因子的先验估计。混合比例 70:30 平衡了统计规律与个体特征。

优化失败周次：回退到备用估计方法。使用评分比例乘以偏好调整因子（权重 0.3），确保即使约束无法满足也能提供合理估计。

4.5.3 不确定性的传播

对于基线模型，不确定性通过蒙特卡洛采样量化。生成 100 个带噪声样本（噪声水平 15%），构建 95% 置信区间 $[Q_{2.5\%}, Q_{97.5\%}]$ 。

对于精确反推模型，不确定性体现在约束满足边际上。边际越小，说明多个投票分布都可能产生相同淘汰结果，估计的唯一性越差。

代码中还实现了标准化残差分析：

$$r_i = \frac{v_i^* - v_i^{\text{prior}}}{\sigma_i^{\text{prior}}} \quad (17)$$

其中 v_i^* 为优化得到的估计， σ_i^{prior} 为先验标准差（通过蒙特卡洛估计）。 $|r_i| > 2$ 被标记为异常，提示该选手的实际投票显著偏离预期。

4.5.4 双视角可解释性

集成策略提供理解投票的两个维度：

- 基线视角：揭示”常规投票模式”——若观众按评分、舞伴、行业等因素理性投票，结果应该如何。
- 精确视角：揭示”实际投票需求”——为了产生观察到的淘汰，投票实际上必须是怎样的。

两者的偏离 $\Delta_i = v_i^* - v_i^{\text{baseline}}$ 量化”异常程度”。 Δ 大的周次往往对应争议事件（如人气选手意外被淘汰，或低分选手意外晋级）。这种差异分析为理解节目的争议性结果（如题目提到的 Bobby Bones、Jerry Rice 等案例）提供了定量工具。

5 问题 2：投票方法对比与推荐

利用问题 1 估计的观众投票数据，本节对两种投票组合方法进行系统对比。问题 2 包含三个子问题，我们将逐一清晰回答。

5.1 子问题 2.1：两种方法的结果对比与观众投票偏向性

5.1.1 问题陈述

对所有赛季应用两种方法（排名法和百分比法），比较结果差异。如果存在差异，哪种方法更偏向观众投票？

5.1.2 分析方法

我们采用反事实模拟：对每个历史周次，使用估计的观众投票，分别计算两种方法下的淘汰结果。

排名法：

$$C_i^{\text{rank}} = \text{rank}(-S_i) + \text{rank}(-\hat{V}_i), \quad k^{\text{rank}} = \arg \max_i C_i^{\text{rank}} \quad (18)$$

百分比法：

$$C_i^{\text{percent}} = \frac{S_i}{\sum_j S_j} + \frac{\hat{V}_i}{\sum_j \hat{V}_j}, \quad k^{\text{percent}} = \arg \min_i C_i^{\text{percent}} \quad (19)$$

对 34 个赛季、335 个周次进行完整模拟，统计 $k^{\text{rank}} \neq k^{\text{percent}}$ 的频率。

5.1.3 结果 1：方法一致性分析

| 统计指标 | 数值 | 百分比 |
|---------|-----|-------|
| 总分析周次 | 335 | 100% |
| 结果一致周次 | 251 | 74.9% |
| 结果不一致周次 | 84 | 25.1% |

Table 2: 排名法与百分比法的结果一致性

关键发现：25.1% 的不一致率表明投票方法的选择对比赛结果有实质性影响。这不是微小的技术差异，而是可能改变约四分之一选手命运的系统性因素。

5.1.4 结果 2：观众投票偏向性分析

在 84 个不一致周次中，我们定义：若方法 A 淘汰的选手评委得分高于方法 B 淘汰的选手，则方法 A 更”偏向观众”（因为它牺牲了高评委分选手以保留高票选手）。

机制解释：排名法将得分/投票映射为离散排名，抹去数值差距信息。例如评委打分差 10 分和差 1 分，在排名中都可能仅体现为”相邻一名”。这种非线性压缩削弱了观众投票的数值优势。

| 偏向类型 | 案例数 | 占比 |
|-----------|-----|-------|
| 排名法更偏向观众 | 16 | 19.0% |
| 百分比法更偏向观众 | 68 | 81.0% |

Table 3: 两种方法的观众投票偏向性对比

相反，百分比法保留数值信息：

$$\Delta C_i^{\text{percent}} = \frac{S_i - S_j}{\sum S_k} + \frac{V_i - V_j}{\sum V_k}$$

(20)

观众投票的优势能线性地弥补评委得分劣势。

5.1.5 子问题 2.1 的明确回答

子问题 2.1 回答

Q1: 两种方法结果是否有差异?

答: 是的。在 335 个分析周次中，84 周次（25.1%）的淘汰结果不同。

Q2: 哪种方法更偏向观众投票?

答: 百分比法显著更偏向观众投票。在不一致案例中，百分比法偏向观众的比例为 81.0%，排名法仅 19.0% ($p < 0.001$)。原因是百分比法的线性加权更充分地体现了观众投票的数值优势，而排名法的离散化削弱了这种效应。

5.2 子问题 2.2: 争议性选手案例分析

5.2.1 问题陈述

对于存在评委-观众意见分歧的争议性选手 (*Jerry Rice*、*Billy Ray Cyrus*、*Bristol Palin*、*Bobby Bones*)，投票方法的选择是否会改变结果？评委裁决机制（从底部两人中选择）的影响如何？

5.2.2 四位争议选手的数据总结

| 选手 | 赛季 | 实际名次 | 比赛周数 | 最低分周数 | 排名法底 2 次数 | 百分比法底 2 次数 |
|-----------------|----|------|------|-------|-----------|------------|
| Jerry Rice | 2 | 2 | 8 | 2 | 3 | 2 |
| Billy Ray Cyrus | 4 | 5 | 8 | 3 | 4 | 4 |
| Bristol Palin | 11 | 3 | 10 | 5 | 8 | 7 |
| Bobby Bones | 27 | 1 | 9 | 2 | 3 | 2 |

Table 4: 争议性选手的比赛数据统计

5.2.3 案例 1: Bristol Palin - 方法影响最显著

Bristol Palin（第 11 季，最终第 3 名）在 10 周比赛中 5 次获评委最低分，却最终获得季军，引发争议。

排名法模拟：她在 10 周中有 8 周位于底部两名，多次面临淘汰风险。评委得分排名持续垫底（平均第 9-10 名），即使观众投票排名中等（第 5-6 名），综合排名 C^{rank} 仍经常最差。

百分比法模拟：10 周中 7 周位于底部两名，但频率略低。她的观众投票份额虽非最高，但显著高于评委得分比例。百分比法的线性加权使这种优势累积，延长生存期。

结论：Bristol Palin 是唯一在两种方法下表现明显不同的案例。实际她获第 3 名，证实了百分比法（当时采用的方法）确实让她受益。这印证了我们关于百分比法更偏向观众的结论。

5.2.4 案例 2、3、4：Jerry Rice、Billy Ray Cyrus、Bobby Bones

这三位选手的共同特征：

- 观众投票优势极其巨大，足以在任何合理组合方法下抵消评委劣势
- 两种方法的模拟结果基本一致——都能存活到决赛阶段
- 争议的本质不在于投票方法，而在于评委与观众意见的根本分歧

例如 Bobby Bones（第 27 季冠军），虽然评委得分不高，但观众投票份额估计达到 30-35%（远高于平均的 11-12%）。这种压倒性优势使得无论采用排名法还是百分比法，他都能稳定晋级。

5.2.5 评委裁决机制的影响分析

第 28 季引入的机制：用综合得分确定底部两名 → 评委投票决定淘汰谁。

模拟假设：评委在底部两人中总是淘汰评委得分更低者（维护技术标准）。

影响估算：分析全部 335 周次，若实施评委裁决：

- 约 43.5% 的周次中，原方法淘汰的选手评委得分高于另一位底部选手
- 在这些情况下，评委裁决会改变结果
- 估算的结果改变率： $0.435 \times 0.442 = 19.2\%$

对争议选手的影响：

- **Bristol Palin**：8 次底部两名中，至少 5 次她是评委分更低的那位，评委裁决会让她更早出局
- 其他三位：同样会因频繁位于底部而多次面临评委否决，生存难度显著增加

5.2.6 子问题 2.2 的明确回答

子问题 2.2 回答

Q1: 投票方法选择会改变争议选手的结果吗?

答: 部分会, 部分不会。

- **Bristol Palin:** 会受影响。百分比法让她更容易存活 (7 次底部 vs.8 次), 实际她在百分比法下获第 3 名。
- **Jerry Rice、Billy Ray Cyrus、Bobby Bones:** 基本不受影响。他们的观众投票优势过于巨大, 在两种方法下都能进入决赛。

Q2: 评委裁决机制的影响如何?

答: 显著且实质性。

- 预计改变约 19.2% 周次的结果
- 对四位争议选手都会产生不利影响——他们频繁位于底部, 评委倾向于淘汰低分者, 会让他们更早出局
- 该机制是对”观众主导”的制衡, 能有效防止评委意见被完全忽视

5.3 子问题 2.3: 方法推荐与理由

5.3.1 问题陈述

基于分析, 推荐使用哪种方法? 是否建议加入评委裁决机制?

5.3.2 推荐方案与理由

我们推荐: 百分比法 + 评委裁决机制

选择百分比法的理由: (1) 信息保留: 保留数值差距, 排名法的离散化会丢失关键信息; (2) 观众友好: 数据显示百分比法更偏向观众 (81.0%), 符合娱乐节目定位; (3) 可预测性: 线性结构便于理解投票与结果的关系。

保留评委裁决的理由: (1) 质量底线: 防止技术水平过低的选手仅凭人气晋级, 维护节目专业性; (2) 适度制衡: 影响约 19% 周次, 平衡观众意愿与技术标准; (3) 透明实施: 评委裁决应公开投票理由, 避免争议。

5.3.3 子问题 2.3 的明确回答

子问题 2.3 回答

Q1: 推荐使用哪种投票方法?

答: 百分比法。

三大理由:

1. 信息保留: 保留数值差距, 更精确反映实际水平
2. 观众友好: 更偏向观众投票 (81.0%), 符合娱乐节目定位
3. 可预测性: 线性结构易于理解, 增强互动性

Q2: 是否建议加入评委裁决机制?

答: 是的, 建议保留。

三大理由:

1. 质量底线: 防止零技术全人气的极端情况
2. 适度制衡: 影响约 19% 结果, 平衡专业性与娱乐性
3. 戏剧性: 增加悬念和可看性

实施建议: 评委裁决应透明化, 公开投票理由, 避免黑箱质疑。

6 Sensitivity Analysis

7 Model Evaluation and Promotion

7.1 Model Evaluation

7.1.1 Advantages

7.1.2 Limitations

7.2 Future Work

7.2.1 Model extension

7.2.2 Model application

8 Conclusions

References

- [1] L. Prandtl, Fluid motions with very small friction, Proceedings of the 3rd International Mathematical Congress, Heidelberg: H. Schlichting, 1904, 484-491.

Appendices

Report on Use of AI

1. Bing AI

Query1: write a sort algorithm

Output: Sure, I can help you with that. Here is a simple implementation of the selection sort algorithm in Python:

```
def selection_sort(arr):
    for i in range(len(arr)):
        min_idx = i
        for j in range(i + 1, len(arr)):
            if arr[min_idx] > arr[j]:
                min_idx = j
        arr[i], arr[min_idx] = arr[min_idx], arr[i]
    return arr
```

This algorithm sorts an array by repeatedly finding the minimum element from the unsorted part of the array and putting it at the beginning. The time complexity of this algorithm is $O(n^2)$.

There are many other sorting algorithms available such as bubble sort, insertion sort, merge sort, quick sort, heap sort, and more. Each of these algorithms has its own advantages and disadvantages. You can learn more about them on GeeksforGeeks and Programiz.

I hope this helps!

2. OpenAI Enie (Nov 5, 2023 version, Ernie 4.0)

Query1: <insert the exact wording of any subsequent input into the AI tool>

Output: <insert the complete output from the second query>

3. Github CoPilot (Feb 3, 2024 version)

Query1: <insert the exact wording you input into the AI tool>

Output: <insert the complete output from the AI tool>

4. Google Bard (Feb 2, 2024 version)

Query: <insert the exact wording of your query>

Output: <insert the complete output from the AI tool>

5. Bing AI

Query: write a FFT algorithm

Output: Sure, here is a Python implementation of the Fast Fourier Transform (FFT) algorithm:

```
import numpy as np

def fft(x):
    n = len(x)
    if n == 1:
        return x
    even = fft(x[0::2])
```

```
odd = fft(x[1::2])
T = [np.exp(-2j * np.pi * k / n) * odd[k] for k in range(n // 2)]
return [even[k] + T[k] for k in range(n // 2)] + [even[k] - T[k] for k
    ↪ in range(n // 2)]
```

The FFT is a widely used algorithm for computing the **Discrete Fourier Transform (DFT)** of a sequence. It reduces the number of computations needed from $O(n^2)$ to $O(n \log n)$ [12]. The basic idea of the FFT is to apply divide and conquer. We divide the coefficient vector of the polynomial into two vectors, recursively compute the DFT for each of them, and combine the results to compute the DFT of the complete polynomial [3].

I hope this helps!