arXiv:2603.14103v1 [cs.MS] 14 Mar 2026

Scorio.jl：用于对随机响应进行排序的 Julia软件包

Mohsen Hariri[1], 、Michael Hinczewski[2],和 Vipin Chaudhary[1]

[1]计算机与数据科学系，凯斯西储大学，美国俄亥俄州克利夫兰[2]物理系，凯斯西储大学，美国俄亥俄州克利夫兰

摘要

Scorio.jl是一个 Julia软件包，用于根据系统对共享任务的重复响应来评估和排序系统。它提供了一个通用的基于张量的接口，支持直接基于分数、成对比较、心理测量、投票、图以及列表式方法，因此可以在多种排序假设下分析同一基准测试。我们介绍了该软件包的设计，将其与现有的 Julia 工具进行比较，并报告了有关合成排名恢复、有限试验下的稳定性以及运行时间扩展的初步实验结果。

关键词

Julia、排序、随机评估、数值分析、科学计算、大语言模型

1. 概要

当基准测试涉及随机输出——例如针对每个提示词多次采样语言模型 [37, 16], 、随机化软件测试或重复模拟运行时， $R \in \{0, \ldots, C\}^{L \times M \times N}$ 自然的数据对象就是一个三维张量，用于 L 记录系统在 M 各项任务上经过 N 多次试验的表现。不同的排名方法会使用这个张量的不同视图：平均值、成对胜场计数、逐项任务排序或潜在特质估计。在这些视图之间切换，通常意味着要改用不同的工具。

Scorio.jl 将张量作为唯一输入，并在内部处理视图转换。对于二元响应基准测试，用户可以将同一个 R 传入 avg、bradley_terry、rasch、borda 或 pagerank，获得可比较的排序向量，而无需改动周边流程；诸如 bayes 这样的加权基于分数的方法也支持分级结果。因此，该软件包有两种用途：日常排行榜构建，以及研究排名规则变化时排名如何改变。Julia 的多重分派和稠密数组原语使这些异构方法能够通过一个 API 实现，且具有实用性 [4]。

2. 需求说明

在重复随机观测下进行排序，是多个领域的常见做法。在教育测验中，被试可能会多次回答相同题目，或按照自适应计划作答 [6, 23]。在软件基准测试中，硬件调度或随机初始化带来的非确定性意味着，单次运行很少能得出确凿结论。在语言模型评估中，Pass@k 等指标会汇总每项任务的多个样本，而有限的样本预算可能导致排序不稳定 [8, 17]。

在所有这些情况下，分析师面临的问题都超越了“哪个系统的平均值最高”：需要进行多少次试验排序趋于稳定之前？成对技能模型与直接分数平均值是否一致？采用不同方式处理平局时，排名最高的系统是否会发生变化？能否纳入先前运行中的先验信息？

如今，要回答这些问题，往往需要拼接使用多个工具，而每个工具都有自己的输入格式。Scorio.jl 消除了这一障碍。它将重复响应张量视为标准数据结构，并允许用户通过一次函数调用切换排序方法。只要排名规则本身是分析的一部分，而不是固定的后处理步骤，这种做法就很有用。

3. 领域现状

有几个 Julia 软件包能够处理这一问题的部分环节。StatsBase 提供通用的排序和并列排名工具，但无法处理重复的基准测试张量 [21]。RankAggregation.jl 将多个分数列汇总为一个共识排序 [20]。Ranking.jl 专注于二元比较记录，并包含 draftElo、Bradley–Terry 和 TrueSkill 的实现 [42]。TrueSkillThroughTime.jl 专门用于估计随时间演变的技能 [24]。这些软件包都很有用，但各自只涵盖了这一问题中较窄的一部分。

Scorio.jl 在两个方面有所不同。首先，它的基本输入是在随机采样下，跨系统、任务和试验重复采集的响应张量。其次，它通过统一接口整合了基于评估、成对、概率、心理测量、投票、图和列表式排名算法。这种组合是该软件包在软件方面的主要贡献：它将排名方法比较从需要定制集成的问题，转变为基准分析中的常规环节。

4. 软件设计

4.1 响应张量与统一 API

评估单个系统时，Scorio.jl 使用结果矩阵$E \in \{0, \ldots, C\}^{M \times N}$，其中 M是任务数， N是试验数， C 是最大类别标签（总计有 $C + 1$ 个结果类别）。对多个系统进行排序时，它使用响应张量 $R \in \{0, \ldots, C\}^{L \times M \times N}$，其中 L 是系统数。二维输入会自动提升为单试验张量，因此一次性基准测试也使用相同的代码路径。对结果进行评分时，基于分数的方法可以通过评分标准权重直接处理这些结果。

不同的方法类别会使用 R 的不同视图。逐点方法使用试验均值 $\bar{r}_{\ell m} = \frac{1}{N} \sum_{n=1}^{N} R_{\ell m n}$ 或按评分标准加权的分数。成对比较方法使用导出的决定性胜利次数 $W_{\ell,\, \ell'} = |\{(m,\ n)\colon\ R_{\ell m n} > R_{\ell' m n}\}|$，必要时另行记录平局。投票和列表式方法使用由 $(R_{1mn},\ \ldots,\ R_{Lmn})$ 导出的逐任务排序。图方法则构建

加权比较图。项目反应理论方法根据二元结果估计潜在能力和任务难度。所有这些转换都在软件包内部完成；用户传入的张量采用相同的 $(L,\ M,\ N)$ 格式。

4.2 分数构建、平局处理与先验

分数构建与排名分配彼此分离。大多数方法会返回连续分数，再由共享的平局感知排序层将这些分数转换为排名。这样既避免了重复编写平局处理逻辑，也明确了排名语义（竞赛排名、密集排名、顺序排名、分数排名）。

该软件包还为最大后验估计方法提供了明确的先验，包括高斯分布、拉普拉斯分布、柯西分布、均匀分布、自定义先验和经验先验。当以往的基准测试运行或基线评估需要为当前排序提供参考时，这一功能很有用；它也鼓励用户明确说明自己的假设，而不是将其隐藏在自定义代码中。

4.3 Julia 工作流示例

典型的工作流程从一个整数张量 `R` 开始，其形状为 `(L, M, N)`。同一个对象无需重塑形状，即可传递给多个排名算法。代码清单 1 展示了共享 API。

代码 1：使用共享张量接口的 Scorio.jl 工作流程示例。

```
使用 Scorio
# R 的大小为 (L,M, N)，对应系统、任务和试验 r_bayes, s_bayes  =


针对单个系统在不同任务和试验中的分级评估E  = [0 1 2 2 1;


  # 将分数向量转换为多个考虑并列的排名视图排名_视图 = 排名_分
```

4.4 方法家族

表1 列出了 `Scorio.jl` 目前提供的排序方法族，以及各方法族使用的数据视图。基于评估的方法族包括 平均值、贝叶斯以及若干 Pass@k 变体 [8, 17]。序贯评分模型（Elo、Glicko、TrueSkill）处理由此生成的成对匹配 [12, 15, 18]。Bradley–Terry、Davidson 和 Rao–Kupper 等成对比较模型，根据汇总的胜局／平局计数估计实力参数 [7, 10, 34], ，此外还有汤普森采样和 Metropolis MCMC排序器 [38, 27]。项目反应理论模型（Rasch、二参数逻辑斯蒂模型、三参数逻辑斯蒂模型、边际最大似然及一种动态变体）联合估计潜在能力和任务难度 [35, 5, 28, 6, 40]。投票规则包括博尔达法、科普兰法、舒尔策法、排序对法、Kemeny–Young、Nanson法、Baldwin法和多数判断 [11, 9, 26, 36, 39, 22, 43, 29, 2, 3]。图与谱方法包括 PageRank、Rank Centrality、AlphaRank、纳什均衡元博弈求解器、SerialRank 和 HodgeRank [32, 41, 30, 31, 1, 14, 19]。最后，列表式选择模型包括 Plackett–Luce、Davidson–Luce 和 Bradley–Terry–Luce模型 [33, 13, 25]。

这一广度是该软件的主要贡献：用户无需在不兼容的工具之间切换，就能在同一数据上比较不同的排序假设。

5. 实验评估

我们检验三个问题：(i) 代表性方法能否恢复已知的合成排序；(ii) 仅有少量试验时，排序有多稳定；(iii) 实际运行时间开销如何。以下结果来自试点配置文件。图2通过平均绝对排名误差和 Top-1 稳定性提供了互补视角。

5.1 合成排序恢复

我们使用 Rasch 风格的潜在能力模型生成二元响应张量，其中包含 $L = 11$ $M = 500$ $N \in \{1, 2, 4, 8, 16, 32\}$ 个系统、任务、试验和四个随机种子。真实能力向量中包含一个有意设置的并列（两个系统的能力相同）。图1 （左）和表2总结了结果。

即使在 $N = 1$，大多数方法也能很好地恢复潜在顺序：Kendallτ_b 介于 0.963（Rasch）和 0.982（BT-Davidson、Plackett– Luce）之间。到 $N = 32$ 时，除 Elo 外的所有方法都达到 $\tau_b = 0.991$，平均绝对排名误差为 0.091。

Elo 明显是个例外（在 $N = 1$ 时为 $\tau_b = 0.624$，在 $N = 32$ 时为 0.661）。这符合预期：Elo 将任务—试验对作为比赛，按顺序更新评分；这些比赛的顺序会引入路径依赖噪声，即使增加试验次数，这种噪声也不会消失。值得注意的一个负面结果是，所有受测方法都未能恢复有意设置的真实平局，这表明在这种情形下，精确恢复平局比恢复粗略的排名顺序更困难。为此，软件包包含一个平局感知排序层，可应用于任意分数向量，方便用户尝试不同的平局阈值。

5.2 有限试验下的稳定性

对于十个合成数据集，每个数据集包含 $N_{\max} = 64$ 次试验，我们将 贝叶斯排序在 $N_{\max}$ 时的结果作为参考，并仅根据前 $n \in \{1, 2, 4, 8, 16, 32\}$ 次试验重新计算排序。图 1 （右）和表 3 展示了结果。

在 $n = 1$ 时，大多数方法的结果已经接近参考值。直接基于分数的方法（平均值、贝叶斯、g-Pass@k）、PageRank 和 BT-Davison 一开始就在 $\tau_b \approx 0.971$ 时达到完美的 top-1 一致性，并在 $n = 32$ 时达到约 0.989。Rasch 起步较低（0.956），但随后迎头赶上。标准 Pass@k 在最小的 n 规模下略不稳定，但在 $n \geq 8$ 时达到完全的 top-1 一致性。

特殊情况是 mG-Pass@k：在 $n = 1$ 时，它会退化为全平局（ τ_b 未定义，top-1 一致性为零）。到了 $n = 2$，它便与其他基于分数的方法表现一致。Elo 始终接近 $\tau_b \approx 0.59$，从未达到可靠的 top-1 一致性。这些结果表明，当试验次数有限时，方法的选择最为重要——而这恰恰是实践者可用于指导选择的数据最少的情形。

5.3 运行时间扩展

我们对以下配置的墙钟时间进行基准测试： $L \in \{4, 8, 16\}$、$M \in \{100, 500, 1000\}$和 $N \in \{1, 4\}$；每种配置均进行一次预热调用和两次计时重复实验。图3 展示了运行时间与 M的关系。由于其阶乘复杂度，Kemeny–Young 仅针对 $L \leq 8$ 进行测试。

成本差异很大。在所有测试配置中，贝叶斯、博尔达、PageRank 和平均值每次调用的耗时都低于 2×10^{-3} s；Alpha-Rank 接近这一阈值（最高 2.05×10^{-3} s），而 Plackett–Luce

表 1：Scorio.jl 中的排序方法族以及各自使用的数据视图。表中展示了具有代表性的方法；有几个类别还提供其他关键字变体。

类别	代表性方法	主要数据视图
基于评估	平均值，贝叶斯，pass_at_k，pass_hat_k，g_pass_at_k_tau，mg_pass_at_k	每个系统的重复结果和按评分标准加权的分数
逐点	逆向难度	每项任务的解题率与任务重新加权
序贯评级	Elo、Glicko、TrueSkill	诱导的成对比赛序列
成对比较	Bradley_Terry、Bradley_Terry_Davidson模型、Rao_Kupper，Thompson、贝叶斯_MCMC	跨任务和试验汇总的胜局与平局次数
投票	博尔达、科普兰法、舒尔策法、排名_对、凯梅尼_杨法，Nanson法、Baldwin法、多数_判断	系统之间由以下内容诱导的逐任务排序
IRT	Rasch、Rasch_2PL、Rasch_3PL、Rasch_边际最大似然、动态_项目反应理论	重复响应潜在系统能力和任务难度
图与谱方法	PageRank，rank_中心性，AlphaRank，纳什均衡，串行_排名，Hodge_排名	系统上的加权比较图
列表式和选择	plackett_luce, Davidson模型_luce, bradley_terry_luce	每个任务的胜者集和败者集和试验

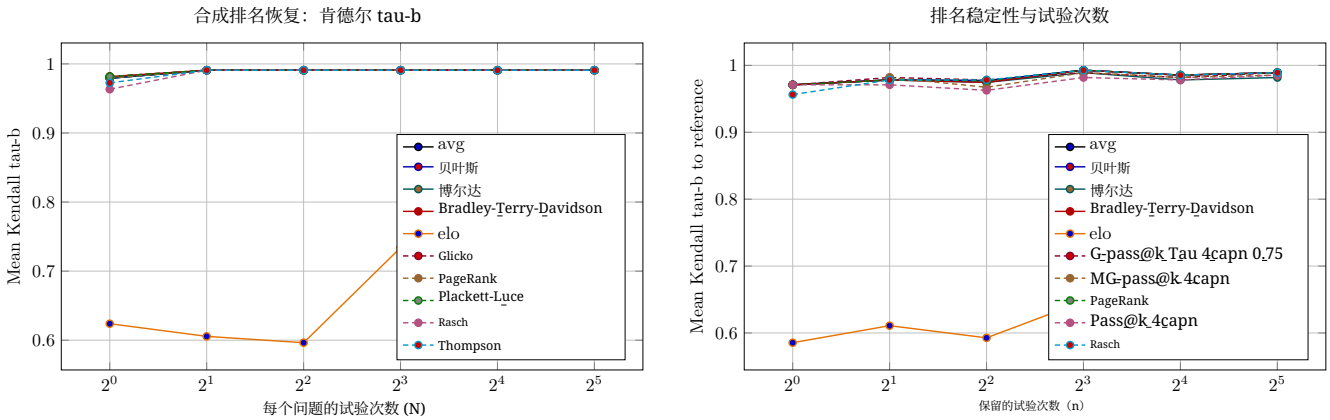

图1：**左：**随着试验次数增加，预测排序与真实排名之间的 Kendall τ_b。大多数方法很快聚集在 1。0 附近；Elo 则没有。**右：**有限试验排名相对于 $N_{\max} = 64$ 处贝叶斯参考的稳定性。方法间的差异在小-n 区间最为明显。

表2：合成排名恢复（4 个随机种子的均值）。除 Elo 外，所有方法的 Top-1 恢复均为 1。0。没有任何方法恢复出真实平局。

	$N = 1$		$N = 32$	
方法	τ_b	MAE	τ_b	MAE
avg	0.979	0.159	0.991	0.091
贝叶斯	0.979	0.159	0.991	0.091
BT-Davidson	0.982	0.136	0.991	0.091
Rasch	0.963	0.227	0.991	0.091
PageRank	0.979	0.159	0.991	0.091
Plackett–Luce	0.982	0.136	0.991	0.091
Elo	0.624	1.636	0.661	1.500

接近 4×10^{-3} s（最高 4.06×10^{-3} s）。Elo 和 Bradley–Terry 的耗时在几毫秒范围内，但会随 L 和 M显著增加。Kemeny–Young 慢了几个数量级（在 $L = 8$ 时为 0.34–0.42 秒），这反映了底层优化问题的 NP难性——

表3：相对于 $N_{\max} = 64$ 处贝叶斯参考的排名稳定性（对 10 个随机种子取均值）。对于 $n = 1$处的 mG-Pass@k，所有系统并列，因此 τ_b 未定义。

	$n = 1$		$n = 32$	
方法	τ_b	Top-1	τ_b	Top-1
avg	0.971	1.000	0.989	1.000
贝叶斯	0.971	1.000	0.989	1.000
Pass@k	0.971	1.000	0.985	1.000
g-Pass@k	0.971	1.000	0.989	1.000
mG-Pass@k	–	0.000	0.985	1.000
Rasch	0.956	1.000	0.989	1.000
PageRank	0.971	1.000	0.989	1.000
Elo	0.585	0.400	0.589	0.200

tion。Rasch 是测试过的最耗时方法，在 $(L,\ M,\ N) = (16,\ 1000,\ 1)$ 处耗时 8。95 秒，内存分配量约为 12。8 GB，这是因为当前的边际最大似然实现使用了稠密求积。

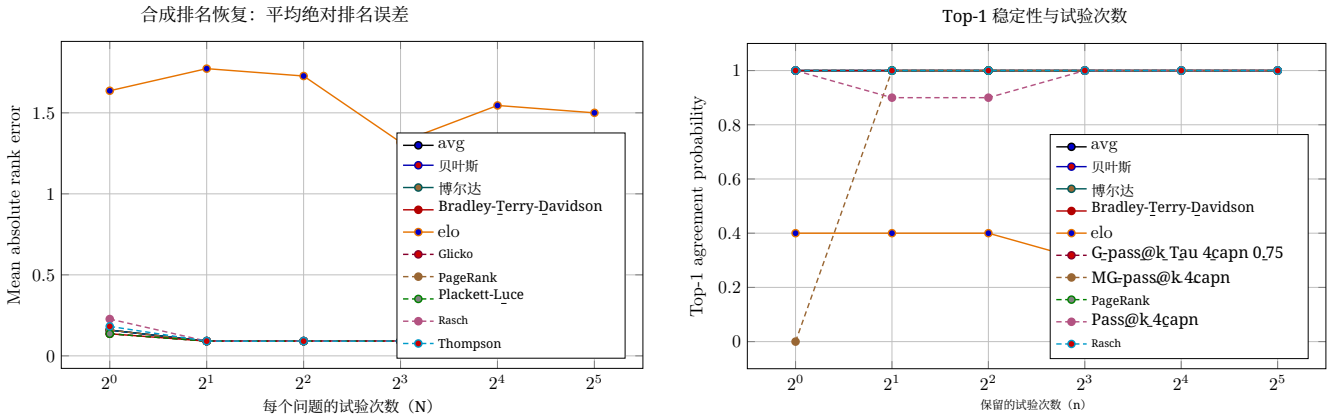

图 2：同一组实验的互补视角。**左：** 平均绝对排名误差与试验次数的关系。**右：** 有限试验次数下的 top-1 一致性。mG-Pass@k 在 $n = 1$ 时退化为全平局；Elo 在整个过程中仍不可靠。

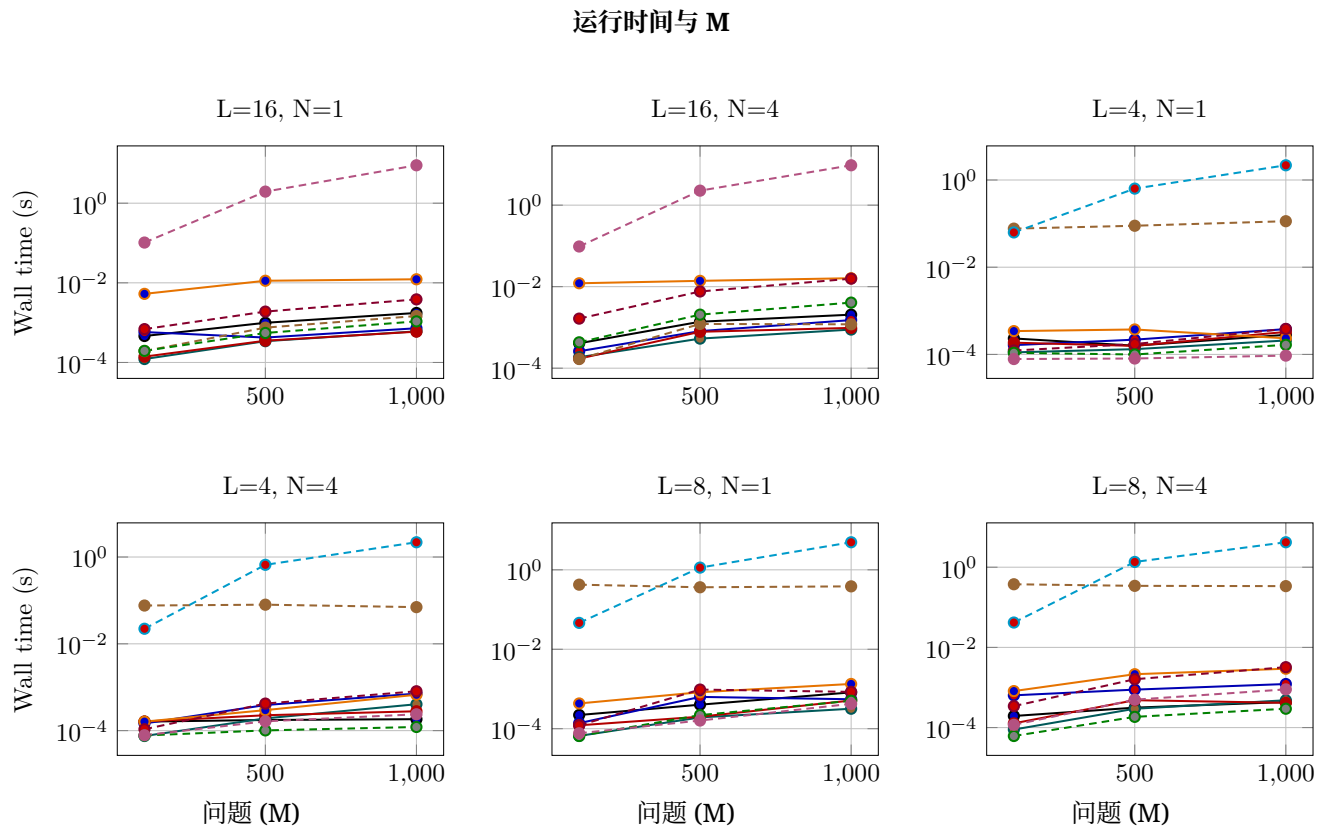

图 3：运行时间与任务数 n 的关系 种配置下的任务数 M。得分方法和图方法在整个网格范围内都保持较快速度 . Rasch 和 Kemeny–Young 方 要贵得多。

6. 研究影响声明

Scorio.jl 降低了研究基准测试结论如何取决于排名规则的成本。在随机软件基 成绩、重复模拟研究、教育评估，以及任何包含共享任务和重复观测的场景中，排序

方法都是一种可能改变结果的建模选择。该软件包为多种排序方法族提供了通用接口，使敏感性分析和假设比较变得简单——将原本需要定制集成的工作简化为一次函数调用。

7. 结论

Scorio.jl 为共享任务的重复响应排序提供了通用的 Julia 接口。其核心贡献在于软件基础设施：无需重建周边工作流程，即可对相同的张量值输入应用直接基于分数的方法、成对比较方法、心理测量学方法、投票方法、图方法及列表式方法。在此报告的试点实验中，大多数具有代表性的方法都能快速恢复并稳定所测试的合成排序；而运行时间测量则显示，不同方法类别的计算成本存在显著差异。这些结果表明，Scorio.jl 是一个实用的工作台，可用于在重复随机评估条件下开展比较排序分析。

8. 参考文献

[1] David Balduzzi、Karl Tuyls、Julien Pérolat 和 Thore Graepel。重新评估评估。在 神经信息处理系统进展，第 31 卷，第 3272–3283 页。Curran Associates, Inc.，2018 年。[2] J. M. Baldwin。Nanson偏好多数选举制度的技术。维多利亚皇家学会会刊（新系列），39(1)：42–52，1926 年。[3] Michel Balinski 和 Rida Laraki。多数判决法：测量、排序与选举。麻省理工学院出版社，2011 年。doi:10.7551/mitpress/9780262015134.001.0001。[4] Jeff Bezanson、Alan Edelman、Stefan Karpinski 和 Viral B. Shah。Julia：一种数值计算的新方法。《*SIAM*评论》，59(1)：65–98，2017 年。doi:10.1137/141000671。[5] Allan Birnbaum。一些潜在特质模型及其在推断受试者能力中的应用。载于 Frederic M. Lord 和 Melvin R. Novick 编，心理测验分数的统计理论，第 396–479 页。Addison-Wesley，马萨诸塞州雷丁，1968 年。[6] R. Darrell Bock 和 Murray Aitkin。项目参数的边际极大似然估计：EM 算法的应用。《心理测量学》，46(4)：443–459，1981 年。doi:10.1007/BF02293801。[7] Ralph Allan Bradley 和 Milton E. Terry。不完全区组设计的秩分析：成对比较法。《生物统计学》，39(3-4)：324–345，1952 年。doi:10.1093/biomet/39.3- 4.324。[8] Mark Chen、Jerry Tworek、Heewoo Jun、Qiming Yuan、Henrique Ponde de Oliveira Pinto、Jared Kaplan、Harri Edwards、Yuri Burda、Nicholas Joseph、Greg Brockman、Alex Ray、Raul Puri、Gretchen Krueger、Michael Petrov、Heidy Khlaaf、Girish Sastry、Pamela Mishkin、Brooke Chan、Scott Gray、Nick Ryder、Mikhail Pavlov、Alethea Power、Lukasz Kaiser、Mohammad Bavarian、Clemens Winter、Philippe Tillet、Felipe Petroski Such、Dave Cummings、Matthias Plappert、Fotios Chantzis、Elizabeth Barnes、Ariel Herbert-Voss、William Hebgen Guss、Alex Nichol、Alex Paino、Nikolas Tezak、Jie Tang、Igor Babuschkin、Suchir Balaji、Shantanu Jain、William Saunders、Christopher Hesse、Andrew N. Carr、Jan Leike、Josh Achiam、Vedant Misra、Evan Morikawa、Alec Radford、Matthew Knight、Miles Brundage、Mira Murati、Katie Mayer、Peter Welinder、Bob McGrew、Dario Amodei、Sam McCandlish、Ilya Sutskever 和 Wojciech Zaremba。评估基于代码训练的大语言模型，2021。doi:10.48550/arXiv.2107.03374。2107.03374。[9] Arthur H. Copeland。合理的社会福利函数。数学在社会科学中的应用研讨会，1951。密歇根大学，安娜堡。油印讲义。[10] Roger R. Davidson。将Bradley–Terry模型扩展至适用于包含平局的成对比较实验。《美国统计学会杂志》，65 328，1970 329 317 10 1080 01621459 1970 10481082 ( ): – 。doi: . / . . .[11] Jean-Charles de Borda。关于投票选举的回忆录。皇家科学院史，巴黎，1781。通常引述为刊载于《皇家科学院史/回忆录》1781年卷（1784年发行）。[12] Arpad E. Elo。《国际象棋棋手评级：过去与现在》。Arco Publishing，纽约，1978。[13] David Firth、Ioannis Kosmidis 和 Heather Turner。适用于带平局多项目选择的Davidson–Luce模型，2019。doi:10.48550/arXiv.1909.07123。1909.07123。[14] Fajwel Fogel、Alexandre d’Aspremont 和 Milan Vojnovic。使用序列化进行谱排名。《机器学习研究杂志》，17:88:1–88:45，2016。[15] Mark E. Glickman。大型动态成对比较实验中的参数估计。《英国皇家统计学会杂志：C辑（应用统计）》，48(3):377–394，1999。doi:10.1111/1467-9876.00159。[16] Mohsen Hariri、Michael Hinczewski、Jing Ma 和 Vipin Chaudhary。测试时扩展下排序推理大语言模型。预印本，2026。[17] Mohsen Hariri、Amirhossein Samandar、Michael Hinczewski 和 Vipin Chaudhary。不要Pass@k：一种用于大语言模型评估的贝叶斯框架。arXiv预印本 arXiv:2510.04265，2025。已被国际学习表征会议2026录用。[18] Ralf Herbrich、Tom Minka 和 Thore Graepel。TrueSkillTM：贝叶斯技能评级系统。载于《神经信息处理系统进展19》，第569–576页。麻省理工学院出版社，2007。[19] Xiaoye Jiang、Lek-Heng Lim、Yuan Yao 和 Yinyu Ye。统计排序与组合霍奇理论。《数学规划》，127(1):203–244，2011。doi:10.1007/s10107-010-0419-x。[20] JuliaML贡献者。RankAggregation.jl，2026。GitHub代码库。访问日期：2026-03-07。[21] JuliaStats贡献者。StatsBase.jl，2026。GitHub代码库。访问日期：2026-03-07。[22] John G. Kemeny。《没有数字的数学》。《代达罗斯》，88(4):577–591，1959。[23] Nan M. Laird 和 James H. Ware。纵向数据的随机效应模型。《生物统计学》，38(4):963–974，1982。doi:10.2307/2529876。[24] Gustavo Landfried。TrueSkillThroughTime.jl，2026。GitHub代码库。访问日期：2026-03-07。[25] R. Duncan Luce。《个体选择行为：理论分析》。约翰·威立父子出版公司，1959。[26] Marie Jean Antoine Nicolas Caritat，孔多塞侯爵。《关于将分析应用于多数票裁决概率的论文》。皇家印刷局，巴黎，1785。

[27] Nicholas Metropolis、Arianna W. Rosenbluth、Marshall N. Rosenbluth、Augusta H. Teller 和 Edward Teller。使用快速计算机进行状态方程计算。《化学物理学杂志》，21(6):1087–1092，1953。doi:10.1063/1.1699114。[28]Robert J. Mislevy。项目反应模型中的贝叶斯众数估计。《心理测量学》，51(2):177–195，1986。doi:10.1007/BF02293979。[29] E. J. Nanson。选举方法。《维多利亚皇家学会会刊与会议录》，19:197–240，1883。二手资料中常引作1882年。[30] Sahand Negahban、Sewoong Oh 和 Devavrat Shah。Rank Centrality：基于两两比较的排序。《运筹学》，65(1):266–287，2017。doi:10.1287/opre.2016.1534。[31] Shayegan Omidshafiei、Christos Papadimitriou、Georgios Piliouras、Karl Tuyls、Mark Rowland、Jean-Baptiste Lespiau、Wojciech M. Czarnecki、Julien Pérolat 和 Rémi Munos。 α-rank：通过进化进行多智能体评估。《科学报告》，9(1):9937，2019。doi:10.1038/s41598-019-45619-9。[32] Lawrence Page、Sergey Brin、Rajeev Motwani 和 Terry Winograd。PageRank引文排名：为万维网建立秩序。技术报告1999-66，斯坦福信息实验室，1999年11月。前一编号：SIDL-WP-1999-0120。[33] R. L. Plackett。排列分析。《应用统计》，24(2):193–202，1975。doi:10.2307/2346567。[34] P. V. Rao 和 L. L. Kupper。成对比较实验中的平局：Bradley–Terry模型的推广。《美国统计学会杂志》，62 204，1967 317 194 10 1080 01621459 1967 10482901 ( ): – 。doi: . / . . .

[35] Georg Rasch。《若干智力与成就测验的概率模型》。丹麦教育研究所，哥本哈根，1960年。[36] Markus Schulze。一种新的单一胜者选举方法，具有单调性、克隆独立性、反转对称性，并符合孔多塞规则。《社会选择与福利》，36(2)：267–303，2011年。doi:10.1007/s00355-010-0475-4。[37] Charlie Victor Snell、Jaehoon Lee、Kelvin Xu 和 Aviral Kumar。对于推理而言，最优扩展大语言模型测试时计算可能比扩展参数更有效。载于《国际学习表征会议论文集》，2025年。口头报告。[38]William R. Thompson。根据两组样本的证据，一个未知概率大于另一个的似然。《生物统计学》，25(3-4)：285–294，1933年。doi:10.1093/biomet/25.3-4.285。[39] T. N. Tideman。克隆独立性作为投票规则的判据。《社会选择与福利》，4(3)：185–206，1987年。doi:10.1007/BF00433944。[40] Norman D. Verhelst 和 Cees A. W. Glas。Rasch模型的动态推广。《心理测量学》，58 415，1993 3 395 10 1007 02294648 ( ): – . doi: . /BF 。[41] Sebastiano Vigna。谱排名。《网络科学》，4(4)：433–445，2016年。doi:10.1017/nws.2016.21。[42] John Myles White 和贡献者。Ranking.jl，2026年。GitHub代码库。访问日期：2026-03-07。[43] H. P. Young。扩展孔多塞规则。《经济理论杂志》，16(2)：335–353，1977年。doi:10.1016/0022-0531(77)90012-6。