面向数据科学的可解释符号回归：2022年竞赛分析

F. O. de Franca*, M. Virgolin†, M. Kommenda¶,M.S.Majumder‡,M.Cranmer‖,G.Espada††,L. Ingelse††, A. Fonseca††, M. Landajuela‡‡, B. Petersen‡‡,R.Glatt‡‡,N.Mundhenk‡‡,C.S.Lee‡‡,J.D.Hochhalter[x], D. L. Randall[x], P. Kamienny[xi], H. Zhang[xii],G.Dick[xiii],A.Simon[xiv],B.Burlacu¶,JaanKasak[xv], Meera Machado[xv], Casper Wilstrup[xv],W.G. LaCava‡

arXiv:2304.01117v3 [cs.LG] 3 Jul 2023

摘要——**符号回归旨在搜索能够准确描述所研究现象的解析表达式。这种方法的主要吸引力在于，它能返回可解释的模型，为用户提供有价值的见解。历史上，大多数符号回归算法都基于进化算法。然而，近期涌现出许多新方案，转而采用枚举算法、混合线性整数规划、神经网络和贝叶斯优化等方法。为了评估这些新方法在一系列真实世界数据中常见挑战上的表现，我们在2022年遗传与进化计算大会上举办了一场竞赛，其中包含多个合成数据集和真实世界数据集，参赛者事先无法看到这些数据集。在真实世界赛道中，我们通过让领域专家评判候选模型的可信度，以贴近实际的方式评估其可解释性。我们深入分析了本次竞赛的结果，讨论符号回归算法当前面临的挑战，并指出未来竞赛可能的改进方向。**

索引词——**符号回归、竞赛、可解释的机器学习**

I. 引言

目标是通过**符号回归**（SR） [1]找到一个参数化函数 $f(x,\ \theta)$，其形式为解析表达式，能够最佳拟合给定数据。这类表达式可以以与不透明模型（例如深度神经网络或树集成）相近的准确率描述非线性交互，同时还能像透明模型（例如线性模型或决策树）那样，让人们能够深入理解其行为。因此，符号回归已被用于从收集的数据中发现新现象，拓展我们对物理学 [2],、化学 [3], 及其他领域的认知。

尽管符号回归问题很有用，但它是 NP难的 [4], ，因此，不同领域提出了许多不同的算法来解决符号回归问题，涵盖进化计算、运筹学和深度学习 [5, 6, 7]。为弥合各领域之间的现有差距，并描绘符号回归领域的最先进水平，近期有人提出了*SRBench*[1] [8]。SRBench 是一个内容丰富的基准测试，涵盖不同的符号回归方法以及其他机器学习方法，并评估表达式准确度、简洁性，以及算法在物理问题上重新发现数据生成函数的能力。SRBench 发布的结果表明，基于遗传编程（GP）的符号回归算法目前在各种真实世界表格问题上往往表现最佳。此外，研究还表明，符号回归本身是对表格数据建模的一种可行方法，因为在许多病例中，发现的表达式展现出与最先进的机器学习方法相近的测试准确度，而通常认为这些方法最适合完成此项任务 [9]。

尽管这类结果十分重要，但截至目前，符号回归领域仍有许多值得探讨的问题尚未得到解答。例如，SRBench 并未研究各符号回归算法在一些常见的真实世界数据挑战中的表现，例如噪声、冗余特征，以及在训练分布之外的外推行为。

为深入探讨这些尚未解决的问题，我们在2022年遗传与进化计算大会（GECCO）上举办了名为面向数据科学的可解释符号回归 的竞赛。此次竞赛旨在推动对符号回归多个重要方面的研究。为此，我们采用了若干超越表达式准确度和长度的评估标准：特征选择、对局部最优的敏感性、外推区间的准确率、噪声敏感性、已知定律的重新发现，以及重要的可解释性 ，由该领域的专家根据现实应用场景进行评估，即预测以下关键指标：

*数学、计算与认知中心（CMCC）、启发式、分析与学习实验室（HAL）、ABC联邦大学，圣安德烈，巴西。†数学与信息学中心进化智能组，科学园123号，阿姆斯特丹，荷兰。‡波士顿儿童医院哈佛医学院计算健康信息学项目，美国波士顿。¶启发式与进化算法实验室（HEAL）、上奥地利应用科学大学，哈根贝格，奥地利。‖普林斯顿大学计算天体物理中心、Flatiron研究所及天体物理科学系，美国。†† LASIGE、里斯本大学科学学院，葡萄牙里斯本。‡‡劳伦斯利弗莫尔国家实验室计算工程部，美国利弗莫尔。x犹他大学机械工程系，美国犹他州。xiMeta，FAIR，法国。xii惠灵顿维多利亚大学工程与计算机科学学院，新西兰。xiii奥塔哥大学信息科学系，新西兰。xiv应用物理研究所、图宾根大学；马克斯·普朗克智能系统研究所，德国图宾根。xv Abzu AI, Orient Plads 1，北港2150，丹麦。审稿中预印本。通讯作者：W. G. La Cava（邮箱：william.lacava@childrens.harvard.edu）

[1]https://cavalab.org/srbench

新冠肺炎疫情。竞赛共收到11份不同的投稿，所采用的方法包括遗传编程、深度学习及其组合。

本文阐述了我们在竞赛设计中的选择依据，深入分析了竞赛，并根据所得结果总结了发现。此外，我们还报告了符号回归社区当前面临的挑战，并就如何改进本次竞赛提出了一些见解。本文结构如下：第二节简要介绍符号回归及其当前最先进水平。第三节介绍竞赛规则、任务、评估方式和参赛者。第四节展示并分析竞赛结果，重点关注获胜方案。最后，第五节讨论结果，总结从竞赛中获得的见解，并强调符号回归研究当前面临的挑战。第六节对本文作总结，并展望后续工作。

II. 背景

符号回归旨在通过组合用户自定义集合中的较简单函数（通常称为基本函数），搜索数据的闭式表达式。该搜索通常由损失函数 $\mathcal{L}(y,\ f(x,\ \theta))$ 引导，该函数用于衡量回归模型与测量目标之间的近似误差。

创建解析表达式的主要优势在于，实践者可以手动检查、调试和调整它。这种手动操作有助于更好地理解模型、提高模型的准确率，或让领域专家融入专家知识。除了手动操作之外，我们还可以使用常见的回归工具分析模型的行为 [10, 11]。值得注意的是，在许多数据集上，就准确率而言，较简单的模型（如解析表达式）可以与不透明模型相媲美 [8, 12]。具体到符号回归，有证据表明，与梯度提升和神经网络相比，解析表达式在训练数据区域之外的外推能力更出色 [13]。

尽管符号回归相较于其他回归方法具有一些优势，但直到最近，它在实践中的应用仍受到多种因素限制：

- 与传统回归相比，训练时间更长；
- 缺乏易于使用的工具箱实现
- 难以定制现有工具箱；
- 缺乏用于解释生成模型的辅助工具；
- 与其他机器学习领域相比，研究群体规模较小，也更为封闭；
- 不同方法和工具箱之间缺乏共享的依赖项／平台。

为解决这些局限性，相关尝试始于 [14] ，并促成了SRBench [8]的创建。SRBench 不仅揭示了使用符号回归进行预测任务的优势，还创建了一个环境，让从业者能够通过类似*scikit-learn*的 Python 接口轻松使用不同的符号回归方法。

III. 2022年符号回归竞赛

2022年的 SRBench 竞赛在美国马萨诸塞州波士顿举行的 GECCO 2022 会议上举办。本次竞赛的目标是评估符号回归算法在预测准确率之外的不同方面，了解符号回归当前面临的挑战，并进一步推动 SRBench 的发展。SRBench 旨在成为该领域用于追踪最先进水平的参考基准测试，并持续改进。

竞赛分为多个阶段，其中资格赛阶段用于筛除未达到最低性能水平的方法，该水平以简单线性回归器的准确率为标准。资格赛阶段结束后，竞赛设有两条赛道：合成赛道用于评估不同的关注属性，并与真实解进行比较；真实世界赛道则邀请一位专家，利用公开的健康数据评判用于预测新冠肺炎传播的各个竞争模型。通过资格赛的方法会在两条赛道上接受评估，并分别设立奖项。

每次运行中，候选符号回归算法的预设时间预算为：对于样本数不超过 1000 的数据集，预算为 1 小时；对于样本数不超过 10000 的数据集，预算为 10 小时。参赛者须确保其算法的运行时间（包括超参数优化的选择）不会超出预算。为尽可能公平地进行比较，竞赛使用哪些数据集不会提前告知参赛者。他们只知道数据集会遵循宾夕法尼亚机器学习基准（PMLB） [15, 16], 的格式，且资格赛阶段将使用来自 PMLB 的数据集。

为了让外部同行能够参与竞赛并复现结果，我们在 SRBench 存储库[2] 中创建了一个新分支，其中说明了如何添加新方法，以及如何在本地计算机上运行竞赛。

A. 资格赛赛道

在资格赛赛道中，我们选用了 20 个 PMLB 数据集，以验证参赛者能否找到优于线性模型基线的模型。本赛道的主要目标是筛除未达到最低可接受预测准确率的参赛者。本赛道的评估标准是 10 次不同运行所得分数的中位数 R^2 。我们根据各数据集上的中位数对结果进行排名，并淘汰排名低于普通线性回归的方案。所选数据集见补充材料。

B. 合成赛道

对于合成赛道，我们的目标是评估不同的任务，这些任务模拟了真实世界数据中常见的挑战。通过评估此类挑战，我们希望更细致地了解当前算法在哪些特定任务上表现出色，或仍有改进空间。

在这些赛道中，我们采用了多项评估标准： R^2、简洁性，以及每项任务各自的任务特定分数。随后，我们计算了综合排名，即针对 n 个不同数据集，将各项评估标准的排名取调和平均数。调和平均数意味着，若要获得较高排名，就不能在任何一项评估标准中排名较低。这样可以避免出现以下情况：

[2]https://github.com/cavalab/srbench/tree/Competition202

表 I 精确表达式再发现任务的生成函数及其对应的难度等级。

函数	难度	生成函数
$f_1(x)$	较容易	$0.4x_1x_2 - 1.5x_1 + 2.5x_2 + 1$
$f_2(x)$	Easy	$f_1(x)$ + 对数 $(30x_3^2)$
$f_3(x)$	中等	$\frac{f_1(x)}{0.2(x_1^2+x_2^2)+1}$
$f_4(x)$	Hard	$\frac{f_1(x)+5.5\ 正弦\ (x_1+x_2)}{0.2(x_1^2+x_2^2)+1}$

符号回归算法返回一个非常简单的模型，但 R^2 较低，却仍在参赛者中排名靠前。

简洁性得分定义为 $\text{round}(-\log_5(s), 1)$，其中 s 是经 *sympy* [17], 简化后表达式树中的节点数量，而 $\text{round}(x, n)$ 会将数值 x 舍入到小数点后第 n 位。引入舍入是为了对大小相近的表达式提供一定的容差。

我们在该竞赛赛道中测试了 5 项不同的任务：

1) **精确表达式再发现：** 符号回归模型必须与数据集的精确生成函数相匹配。2) **相关特征选择：** 符号回归模型只能使用相关特征，舍弃任何噪声特征。3) **跳出局部最优：** 符号回归模型应由无噪声的低级特征构成，而非原始特征的含噪交互项。4) **外推准确度：** 给定一个变量定义域范围有限的数据集，符号回归模型应能在该范围之外按预期表现。5) **噪声敏感性：** 在目标值上施加不同程度的噪声后，符号回归算法应能恢复出接近原始函数的表达式。

在以下小节中，我们将详细说明这些任务的生成数据过程和评估标准。

1) 精确表达式再发现：在此任务中，符号回归算法应返回一个模型，其结果与生成函数相差一个常数因子（乘法因子或加法因子）。为评估这一点，我们依靠 *sympy* 以代数方式处理表达式并评估两个表达式是否等价的能力。

在此任务中，我们设置了三个难度级别，它们基于相同的基础函数构建，并逐步加入更复杂的项。表 I 展示了对应的生成函数。

2) 相关特征选择：对于此任务，我们创建了包含 20 个特征的数据集，其生成函数如下：

$$0.11x_1^3 + 0.91x_3x_5 + 0.68x_7x_9 + 0.26x_{11}^2x_{13} + 0.16x_{15}x_{17}x_{19}. \quad (1)$$

从公式 1 可以看出，生成数据集时不会使用偶数索引的变量（它们是与函数无关的随机变量）。主要目标是让符号回归模型识别并仅使用奇数索引的特征。为此，我们引入了此任务专用的分数，定义如下：

$$\text{feature-select}(f_{true}, f_{pred}) = \frac{\text{true-feats} - \text{false-sel-feats}}{\text{false-feats}}, \quad (2)$$

其中，伪特征是随机取值特征的集合，伪选择特征是符号回归模型所使用的噪声特征集合。此任务还设有三个难度级别——容易、中等、困难——通过向目标值添加不同程度的噪声，并将目标值替换为以下内容来构造：

$$y_{noise} = \mathcal{N}\left(y, \sigma_y\sqrt{\frac{\text{ratio}}{1-\text{ratio}}}\right) \quad (3)$$

其中，ratio 表示噪声水平。各难度设置对应的 ratio 分别为：容易级别 0.025，中等级别 0.05，困难级别 0.1。

3) 逃离局部最优：对于此任务，数据集包含原始特征以及一些元特征，这些特征是生成函数的更高层构建模块。但关键在于，关于元特征的数据包含噪声。因此，通过组合元特征而非原始特征，可以构建一个效果不错但并非最优的模型。我们将元特征（不含噪声）定义如下：

$$f(x) = \underbrace{0.77x_1x_2}_{g_1(x)} + \underbrace{1.52x_2x_3}_{g_2(x)} + \underbrace{1.2x_4^2}_{g_3(x)} + \underbrace{0.31x_1x_4x_5}_{g_4(x)} + \underbrace{0.23x_3x_4x_5}_{g_5(x)}. \quad (4)$$

具有 n 元特征的生成函数由$f_n(x) = \sum_{i=1}^{n} g_i(x)$给出。

相应的数据集由样本列表构成，格式为 $[x_1, x_2, x_3, x_4, x_5, g_1 + \epsilon, \ldots, g_n + \epsilon, f_n(x)]$, 其中 ϵ是遵循公式 3 的加性噪声，比例为 0.1。由于噪声被添加到元特征 g 的数据中，只有忽略元特征并使用原始特征 x，才能恢复对 f_n 的最优拟合。此外，还有 3 个难度级别，由 n的值控制：容易（ $n = 3$）、中等（ $n = 4$）和困难（ $n = 5$）。此处使用的任务特定分数与特征选择任务所用的分数相同。

4) 外推准确度：这项任务旨在评估符号回归模型在验证集位于训练数据域之外时的表现。这种情况颇具挑战性，因为在有限范围内拟合模型时，可能存在许多同样出色、但在训练边界之外表现各异的模型。因此，对于能够在描述训练数据的假设中选择最小复杂度假设的模型，我们会给予奖励。此任务使用的生成函数为：

$$f(x) = \text{erf}(0.22x) + 0.17\sin(5.5x), \quad (5)$$

其中，erf 是定义如下的误差函数

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt.$$

除了这个问题没有解析解之外，训练集和验证集也按如下方式划分： $-15 \le x \le$15 用于训练， $15 < x \le 40$ 用于验证。

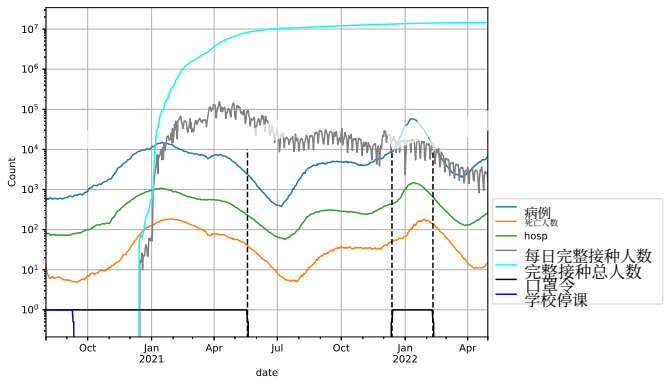


图1. 真实世界赛道所使用的新冠肺炎数据。数据包括2020年8月至2022年4月30日期间纽约州的新冠肺炎病例、死亡人数和住院人数，以及佩戴口罩和学校停课等政策指标。

此任务的具体衡量方式是在验证数据上评估 R^2分数。与其他任务一样，我们按照公式 3 添加噪声，以生成三个不同难度级别的数据：比例为 0.05（容易）、0.1（中等）和 0.2（困难）。

5）噪声敏感性：在此任务中，使用带噪声的训练数据创建符号回归模型，并使用无噪声的验证数据进行评估。所使用的生成函数为：

$$f(x) = \frac{0.11x^4 - 1.4x^3}{0.68x^2 + 1}.$$

不同的噪声比例分别为0.05（容易）、0.1（中等）和0.15（困难）。

C. 真实世界赛道

对于真实世界赛道，目标是在真实世界场景中评估符号回归的可解释性，而不是依赖复杂度指标。在真实世界场景中，模型的可解释性具有主观性，并取决于应用领域领域专家的专业知识。为此，我们测试了参赛方法生成新冠肺炎病例、住院人数和死亡人数的可解释预测模型的能力。

该任务使用了纽约州2020年8月至2022年4月30日期间每日病例、住院人数和死亡人数的时间序列数据。我们采用了 GitHub 代码库 https://github.com/reichlab/covid19-forecast-hub/ 中现有的这些数据汇编。此外，我们还收集了可能与疾病暴发相关的协变量数据，包括政策措施（口罩令、学校停课）以及每日和累计疫苗接种数据。图1概述了数据趋势。

每种方法都需要利用历史数据预测病例、住院人数和死亡人数的未来计数。

三个时间序列（病例数、住院人数、死亡人数）按如下方式进行了预处理。首先，我们将任何异常值替换为一周窗口内的中位数。我们将与一周窗口平均值相差 4 个标准差的任何数据点定义为异常值。之后，我们通过指数加权移动平均生成新的时间序列，以平滑测量值。最后，对于时间序列中的每一天，我们提取了以下特征：

- $x_{t-i,\ j}$：向前回溯 i 个时间步的值，其中 j 表示病例数、住院人数或死亡人数。
- $x_{\Delta_i,\ j}$：向前回溯 i 个时间步的值与 $t = 0$ 时的值之差。
- $x_{tot,\ j}$：从 $t = 0$ 到当前时间步各值的累计和。

我们计算了 $i = \{1, 2\}$ 的这些特征，目的是预测 $t + 1$ 的值（即，不提供当天的预测）。训练集和测试集分别通过交替划分为 5 周和 3 周的数据块来创建，使算法能够获取覆盖整个数据收集时段的样本。

每种算法在每个数据集上运行 10 次，因此每种算法得到 10 个模型。我们根据测试集上的 R^2 分数，为每个候选算法选取最佳模型作为代表模型。传染病专家 Majumder博士使用 1（强烈不信任）至 5（强烈信任）的信任分数对最佳模型进行评估。候选模型的最终分数取准确率、简洁性和该专家信任分数的调和平均值。

D. 竞赛参赛作品

共有 13 支队伍参加了竞赛，其中 1 支队伍退出。为简洁起见，所有方法均汇总于表 II。补充材料中介绍了参赛者提交的每种方法。需要指出的是，GP_{ZGD} 是逾期提交的，仅在竞赛资格赛阶段接受了评估。

IV. 结果

本节将详细分析竞赛各赛道的结果。结果摘要可在竞赛网站[3]上查看。

A. 资格赛赛道

图2报告了资格赛赛道中受测符号回归方法的排名。从图中可以看出，大多数参赛者找到的模型都优于线性回归。在这一阶段，符号回归方法 TaylorGP 和 nsga-dcgp 被取消资格。前者仅返回线性模型，而后者返回了一些略优于线性回归的模型，也有一些略逊于线性回归的模型。尽管 E2ET 的中位分数优于线性模型，但它有时也会找到比线性模型更差的模型。原因之一是该算法最多只能处理包含 10 个变量的数据集，而其中一些数据集的变量数超过了这一限制。另一方面，在这批数据集中，Operon、PS-Tree 和 GP_{ZGD} 找到了最准确的模型。

关于运行时间，从图3中可以看到，一半的参赛者都用完了全部预算来寻找最佳解决方案。由于其中大多数采用基于种群的搜索算法，因此这是一种

[3]https://cavalab.org/srbench/competition-2022/

表 II 竞赛所提交方法的概要。对于方法类别，我们使用：EA=进化算法（例如，遗传编程），DL=深度学习，MIX=多个类别的组合。

Name	类别	Code	简要说明
Bingo [18]	EA	URL	通过线性表示演化通用无环图。包含系数微调、代数化简以及协同演化适应度预测器。
E2ET [7]	DL	URL	一种预训练Transformer，可直接根据数据预测符号回归模型。随后对预测出的模型进行微调，并返回其中最佳的模型。
PS-Tree [19]	MIX	URL	在进化过程中结合决策树、遗传编程和岭回归。它能够以分段方式处理符号回归问题。
QLattice [20]	EA	URL	使用在迭代过程中不断更新的概率分布，采样出越来越优的解决方案。它还包括微调和more.
TaylorGP [21]	EA	URL	将遗传编程与泰勒多项式近似相结合。它利用泰勒展开识别多项式特征并进行分解该问题。
EQL [22]	DL	URL	由一个完全可微的浅层神经网络构成，其中包含作为激活函数的符号回归算子。使用 L^0 损失来剪枝该网络。
GeneticEngine [23]	EA	URL	使用强类型和文法引导遗传编程。对于本次竞赛，文法不需要任何特定领域的信息。
Operon [24]	EA	URL	采用 C++ 编码的遗传编程，通过列文伯格-马夸尔特算法进行微调。它与 Optuna 配合，用于调优搜索超参数。
PySR [25]	EA	URL	使用基于树的表达式、锦标赛选择和局部叶节点搜索。搜索过程中还使用多个种群。
uDSR [26]	MIX	URL	结合递归式问题简化、神经引导搜索、大规模预训练和稀疏线性回归和遗传编程。
遗传编程$_{ZGD}$ [27]	EA	URL	Koza 风格的规范遗传编程，并加入随机梯度下降，在进化过程中调整系数。
NSGA-DCGP	EA	URL	将可微笛卡尔遗传编程 [28]与第二代非支配排序遗传算法（NSGA-II）相结合，以同时发现简短且准确的模型。

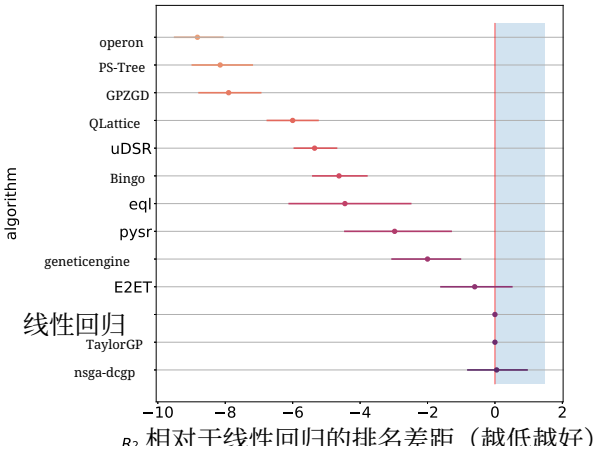


图2. 资格赛赛道中各符号回归方法的排名，以中位数 R^2 计算。在此图中，数值越低越好。阴影区域表示被取消资格的参赛者。

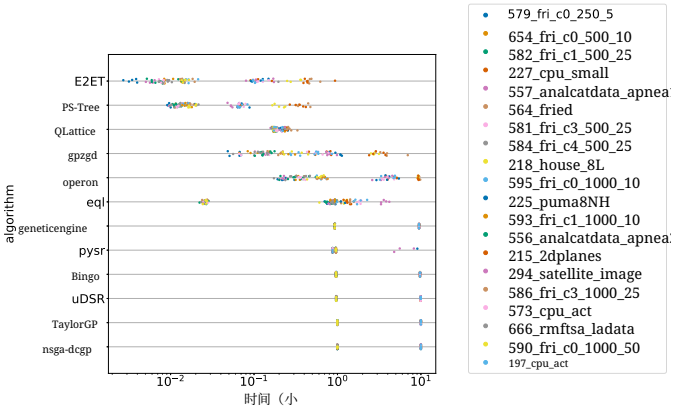


图3. 资格赛阶段各符号回归方法在每个测试数据集上的运行时间。请注意，对于样本少于 1000 个的数据集，方法的总运行时间限制为一小时；对于较大的数据集，限制为十小时。

合理的策略，旨在尽可能增加评估次数，以期找到一个较好的局部最优解。EQL、E2ET 和 PS-Tree 采用了不同的策略来搜索表达式，因此没有阻止过早收敛的机制。这样一来，一旦达到局部最优解，它们就可以停止搜索。

B. 合成基准测试赛道

图4展示了合成基准测试赛道中所有任务的综合排名。从图中可以看出，QLattice 排名第一，其次是 pysr 和 uDSR，这一排名依据全部三项评估标准的调和平均数计算。请注意，排名数值越高越好；这是因为调和平均数对较小的数值惩罚更重，这符合我们的意图，即惩罚未能满足所有评估标准的模型。我们还可以从图中看到，竞争者之间的分布存在重叠，这表明结果可能不具有统计显著性。

就 R^2 分数而言，operon 返回的模型最佳，其次是 QLattice 和 PSTree。从简洁性得分来看，pysr、QLattice 和 Bingo 位列前三；而在特定属性得分方面，表现最好的算法是 QLattice、pysr 和 Operon。QLattice 在各项评估标准中均位列前三，这与其排名调和平均数居首的结果相符。

为验证结果在统计学上的显著性，图 5 展示了使用 Nemenyi 检验（$\alpha = 0.05$）绘制的临界差异图，以找出彼此之间存在显著差异的算法组。从图中可以看出，排名前五的算法在结果上不存在统计学显著差异；这意味着它们的平均表现可能相同，或者我们需要更多基准测试才能观察到显著差异。

1) 精确表达式再发现：从图 6 可以看到，精确表达式再发现任务的结果按不同难度级别分组展示。首先可以观察到，大多数算法都能够找到准确的

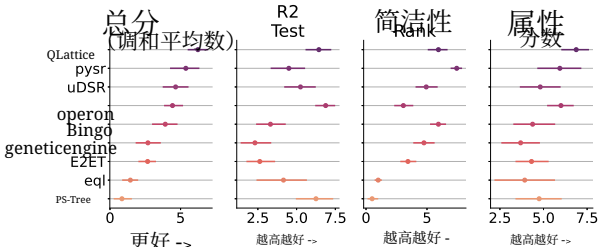


图 4. 合成基准测试赛道中所有任务的汇总结果。最左侧的图展示了全部三个评估标准的调和平均数分布，该图的中位数用于确定获胜者。接下来的三幅图分别展示了 R^2、简洁性和任务特定指标的分布。

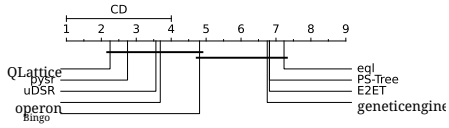


图 5. 使用 Nemenyi 检验并以 $\alpha = 0.05$ 作为事后检验，对评估标准调和平均数的平均值绘制的临界差异图。

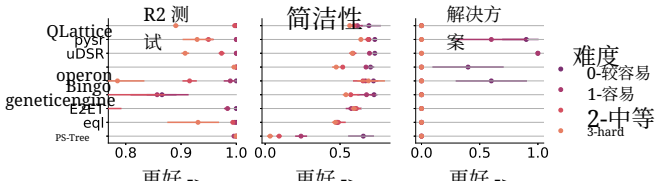


图 6. 精确表达式发现结果的排名分布。与其他图一样，排名值越高，结果越好。

模型（ R^2 高于 0.9），且适用于每个难度级别；唯一的例外是 Bingo、geneticengine 和 E2ET，它们在较高难度级别下表现不佳。它们也都能保持相近的简洁性，PS-Tree 除外，它生成的模型更复杂。

关于表达式的恢复情况，可以看到，这仅在较容易和容易难度级别的问题中实现了。此外，只有 pysr、uDSR、operon 和 Bingo 能够在这些级别中找到精确解。值得注意的是，uDSR 在这两个级别的每次运行中都找到了完美匹配。

通过在三个评估标准中选择排名调和平均数最高的模型，我们发现 Bingo 和 pySR 得到的模型与真实值非常接近。在较容易的难度级别下，Bingo 找到了以下表达式：

$$f(x) = 0.4 \times (x_1 + 6.25) \times (x_2 - 3.75) + 10.37$$

经过一些代数变换后，该表达式与生成函数完全一致。

另外两个难度级别对符号回归算法来说尤其困难。一个可能的原因是，符号回归通常难以处理有理函数，因为它会使用受保护除法，或将除法运算符替换为解析商，以避免非全定义性。按照上述相同的评估标准，operon 针对中等难度级别返回的最佳模型与真实表达式相似：

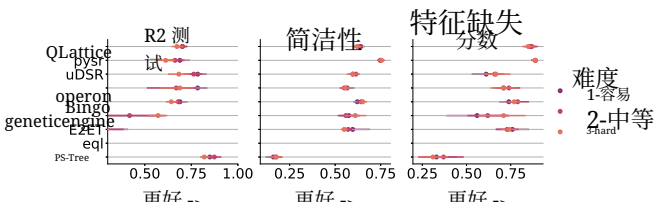


图7. 特征选择任务的排名分布。与其他图表一样，排名值越高越好。

$$f(x) = \frac{-1.66x_1 + (-0.02x_1 - 0.15)\,(-16.15x_2 - 6.46)}{(0.2x_1^2 + 1)\left(\frac{0.19x_2^2}{0.2x_1^2+1} + 1\right)^{0.5}\left(\frac{0.21x_2^2}{0.2x_1^2+1} + 1\right)^{0.5}} \tag{6}$$

如果我们将 $0.19x_2$，$0.21x_2$ 二者都改为 $0.2x_2$，就会得到一个表达式；经过一些代数变换后，它便成为真实表达式（表 I）。这揭示了两个问题：i）数值参数的内部优化可能存在精度不足，从而使表达式更难化简；ii）精确解的评估标准可能会因化简过程中的困难而漏掉正确表达式。

2）相关特征选择：图7展示了仅使用模型中的相关特征这一任务的排名分布。分析所有任务共有的评估标准可以看出，在每个难度级别下，PS-Tree 都是最准确、同时也最复杂的模型。巧合的是，它在特征缺失分数上的得分也最低。由于采用分段方法，它可能创建了额外的子表达式来处理加性噪声，从而生成了更复杂的表达式，并利用了几乎所有特征。

算法 pysr 在三个难度级别中都返回了调和平均数最佳的模型。巧合的是，这三个级别得到的都是同一个模型：

$$f(x) = 7.1x_1 + x_2x_3 + 0.27x_6^2x_7,$$

这与真实生成函数大相径庭。

从这个方程可以看出，pysr 模型使用了两个无关特征，并遗漏了一些真实特征。尽管如此，它仍远未达到理想的解决方案，因为它没有捕捉到相关特征之间正确的交互项。总体而言，这项任务对符号回归模型来说颇具挑战，也表明符号回归可能会受益于在拟合数据之前采用特征选择方法。

3）摆脱局部最优：对于局部最优任务（图 8），我们可以看到，在准确率和简洁性得分方面，其表现与特征选择任务类似。对于特征缺失分数，我们观察到结果的分布更加分散，这表明不同难度级别之间存在显著差异。在容易级别，pySR 和 Operon 在每次运行中都只使用相关特征。例如，以下模型：

$$f(x) = 3.97x_2\,(0.19x_1 + 0.38x_3) + 1.2x_4^2,$$

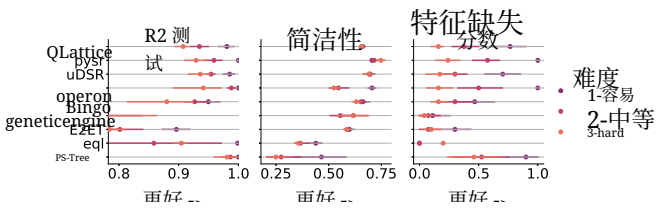


图8. 局部最优任务的排名分布。与其他图一样，排名值越高越好。

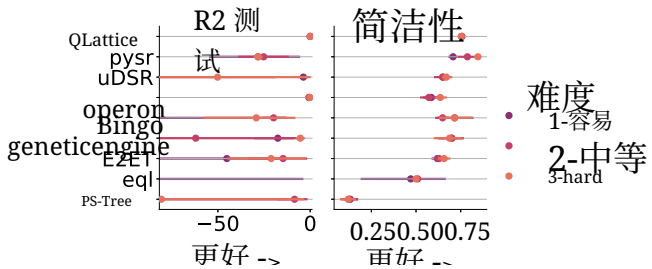


图9. 外推任务的排名分布。与其他图一样，排名值越高越好。

对应于精确的生成函数。对于其他难度级别，则没有完美的解决方案。按调和平均数排名最佳的模型如下：

$$f(x) = 0.77x_1x_2 + 0.31x_1x_4x_5 + 1.52x_2x_3 + 1.2x_4^2,$$

对于采用 operon 的中等难度级别，它也对应于真实模型，并且

$$f(x) = x_{10} + 1.53x_2x_3 + 1.23x_4^2 + x_6 + x_9,$$

对于采用 pySR 的困难级别，它利用一些局部最优特征来获得近乎完美的解决方案。正如我们从这些结果和图8中所看到的，噪声起着重要作用，因为它掩盖了真实特征的重要性。换句话说，尽管 pySR 和 Operon 有能力找到生成函数，但当目标函数中的噪声过多时，噪声变量可能会与真实变量混淆。这项任务仍然具有挑战性，还需要通过更多中间级别进行进一步研究。

4）外推准确度：如图9所示，外推任务对符号回归算法而言极具挑战性。在该图中可以看到，大多数模型获得的准确率都不尽如人意，即使在最容易的难度级别，测试集的 R^2 也常为负值。

为说明不同难度级别模型的行为，我们选取了每个难度级别中调和平均数最高的结果，如图10所示。从该图可以看出，operon 找到了与原始无噪声数据集非常接近的近似，而 QLattice 返回的模型则近似了不含正弦项的误差函数。算法 uDSR 返回了一个近似插值区域部分内容的函数，但在外推数据上表现得非常不同。

总而言之，我们可以看到，这些算法能够捕捉到目标函数的一部分，但缺少一个简单的组成部分。这些结果表明，有必要进一步探究

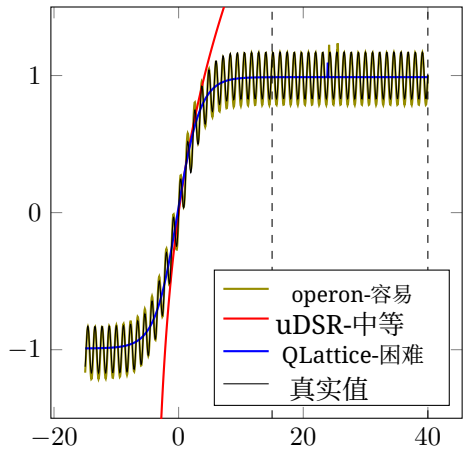


图10。不同难度级别下外推任务的选定解决方案。外推区域位于虚线之间。

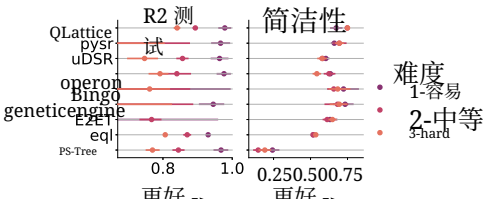


图11。噪声任务的排名分布。与其他图表一样，排名值越高越好。

为什么没有任何算法能够将正弦函数纳入最终表达式。

5）噪声敏感性：最后，在图11中，我们可以看到噪声敏感性任务的准确率和简洁性得分。从该图中可以清楚地看到，随着噪声水平的增加，难度也随之增加。尽管准确率有所下降，几乎所有算法都能够保持较高的简洁性得分。特别是在此任务中，QLattice、uDSR、operon、eql 和 PS-Tree 能够在所有难度级别下保持相对较高的准确率。

C. 真实世界赛道

向专家展示了一系列模型及评估这些模型所需的信息。评估界面包含任务信息（即预测病例、住院人数和死亡人数）、事件时间范围、算法名称、 R^2 分数、简洁性得分、回归模型表达式、真实时间序列与预测时间序列的图表，以及预测值与真实值的图表。专家通过回答“我信任这个模型”这一问题，为我们提供了整体专家评分，评分范围为1分（强烈不同意）至5分（强烈同意）。最终分数是模型准确性、简洁性得分和专家评分的调和平均数。

我们在表 III 中报告了该赛道的结果。在该赛道中，uDSR 获得最高分，其次是 QLattice 和 geneticengine。图 12 展示了一个使用 uDSR 预测死亡人数的示例结果，形式与提交给专家评分时相同。尽管有多种方法找到了

表 III 真实世界赛道的结果。

Rank	算法	分数	Rank	算法	分数
1	uDSR	5.75	5	Bingo	4.66
2	QLattice	5.21	6	pySR	4.17
3	geneticengine	4.99	7	PS-Tree	3.15
4	Operon	4.80	8	E2ET	2.72

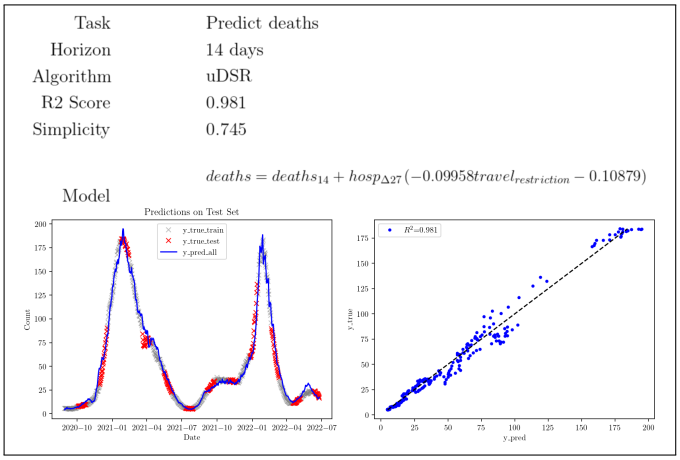


图 12. 用于专家在真实世界赛道中进行评估的 uDSR 模型示例。

在 R^2 分数方面适用于此任务的优秀模型，uDSR 仍成功给出了一个相对简单且合理的线性模型，其中包含一个交互项，并据此生成了良好的预测。

V. 讨论

总体而言，这场竞赛让我们深入了解了符号回归的当前状态。在本节余下部分，我们将重点讨论竞赛本身的不同方面。在这些基准测试中，我们注意到，各任务的难度层次仍存在一些缺口。弥补这些缺口对于凸显不同算法及其局限性之间的差异很重要。

A. 竞赛的局限性

关于这场竞赛的局限性，有一点值得注意：有些算法在一条赛道上表现较好，在另一条赛道上表现较差；甚至在一项评估标准上表现较好、在另一项上表现较差。尽管我们尝试通过使用调和平均数来平衡这一点，但汇总后的最终分数会掩盖结果中的细节。观察合成赛道可以发现，有些算法经过调整以提高准确率，例如 Operon 和 QLattice；另一些算法则试图在准确性与简洁性的权衡之间取得平衡，例如 pysr。由于其中一些方法采用了不同机制来寻求平衡，进一步优化超参数是否能让这些算法取得更好的结果？

简洁性度量只考虑表达式的大小，但有些结构会妨碍表达式的可解释性，却不会显著增加其大小。考虑由非线性函数构成的链，例如 $f(g(h(x))$，它包含 4 个节点；再看一个更简单的表达式，例如 $f(x)+h(x)*g(x)$，它包含 8 个节点。尽管第一个表达式更小，但第二个表达式的行为较容易推理。复杂度度量的示例见 [29, 30]。

另一个问题是计算预算。由于可用于运行实验的资源有限，我们必须限制最长运行时间和重复次数。虽然设定的运行时间可以说是合理的，但增加重复次数或许能够减少结果中的一些不确定性。

最后，我们应该强调，很难说有一个唯一的赢家，或某一种方法可以称为最先进水平。我们必须使用聚合得分来选出赢家；不过，查看结果可以发现，通常不同赛道和不同等级各有赢家。

B. 任务的总体难度

评估任务的难度可能颇具挑战，因为出于避免结果受到偏倚影响的考虑，竞赛开始前无法测试问题实例。事实证明，其中一些任务确实非常困难。

例如，在精确解任务中，只有少数正确表达式被找到，而且仅出现在较容易和容易难度等级中。这有两个主要原因：i) 任何表达式都可以写成不同的等价表达式，因此化简过程可能无法识别它们；ii) 数值参数优化可能导致精度不足，使表达式与真实值接近但并不相等（如式 6 所示）。

对于所有能够较好地近似误差函数、但未包含真实函数中正弦项的方法而言，外推任务同样很困难。一方面，鉴于同一数据集在插值区域内可能存在许多不同的解决方案，而它们在该区域之外的表现各异，因此在没有任何先验信息的情况下，我们不能指望算法作出正确选择。另一方面，即使仅针对训练数据，也没有任何算法能够找到良好的解决方案，这表明生成函数本身就颇具挑战性。

最后，局部最优任务和噪声敏感性任务在难度上形成了良好平衡，使我们得以验证：当前的符号回归算法能够在低噪声数据上找到良好的解决方案，但噪声会对模型质量产生重大影响。在局部最优任务中，我们可以看到，算法倾向于最大限度地追求简洁性，即使这会给表达式引入噪声。这是可以预料的，因为算法会基于信息作出选择，倾向于选择节点数较少的表达式，即使这会稍微降低准确率。

总的来说，我们应利用以往竞赛中获得的信息，持续调整基准问题，以改善难度范围，同时不对任何参赛者产生偏向。此外，持续调整也有助于避免符号回归方法逐渐演变成只为击败这些特定基准测试而优化；这在许多领域都是一个常见问题。

C. 人工检查仍然是一项挑战

关于真实世界赛道，仍有一些需要评估的挑战。首先，该领域的专家

必须通过查看解析表达式（只要其规模合理）、时间序列的插值和外推行为，以及准确度分数，来评估许多不同的模型。即使对于该领域的专家而言，比较两个或更多模型也可能颇具挑战，因为他们必须先确定什么样的模型对自己而言是可解释的。有时，这种临时评估可能会导致结果失真，因为在不使评估任务复杂度呈二次增长的情况下，无法对模型进行成对比较。

D. 我们已经走到这一步了吗？

总的来说，纵观结果，我们可以说，尽管见证了这一领域的诸多进展，但符号回归社区仍面临一些尚未解决的挑战。

其中一个挑战是为终端用户提供定制化体验，例如，当存在准确率较低的替代方案时，允许用户指定对更简单解决方案的重视程度。出于同样的目的，提供具有相近训练准确度的替代模型，可以帮助用户判断哪一个模型最有可能表现出理想的外推行为。此外，尽管解析解可能比不透明模型更具可解释性，符号回归工具仍应以不确定性量化、可视化行为以及偏导数和模型属性等其他数据的形式，提供辅助信息。另一个较小的问题是这些算法的运行时间：由于它们大多依赖进化算法，因此通常比常见的回归算法更慢。

具体而言，就所评估的任务来说，噪声处理和特征选择仍有改进空间。尽管大多数算法都能够妥善处理噪声，并移除部分无用特征，但含噪数据仍会影响准确率，使 R^2 最多降低 20%。此外，移除不必要的特征还可以得到更加简单、且对终端用户更具信息价值的模型。

六、结论

本文深入分析了在 GECCO 2022 举办的 2022 年数据科学可解释符号回归竞赛。竞赛的主要目的是了解现代符号回归算法如何应对数据科学中回归分析的常见挑战，并衡量当前最先进水平。

为此，我们设计了多个任务，并将其分为不同赛道：i）精确表达式再发现；ii）特征选择；iii）避免噪声局部最优；iv）外推准确率；v）噪声数据。这些任务不仅与回归分析相关，也与模型的可解释性相关。我们还增设了一个赛道，由相关领域的专家评估生成的模型在真实世界数据上的表现。

尽管竞赛为合成数据赛道和真实世界赛道分别选出了一位优胜者，但我们可以看到，没有哪种算法能在所有评估标准下都返回最佳模型。这些结果有助于理解每种方法的优缺点，并推动该领域向更优算法发展。

近期，我们将根据首届竞赛积累的经验，推出新一届竞赛，并改进任务和基准测试。我们还将采用不同的评估标准，以便更好地了解各算法应对不同挑战的表现。

致谢

作者谨向所有参赛者致谢，感谢他们为本次竞赛提供了有益的反馈。本次竞赛由波士顿儿童医院计算健康信息学项目和上奥地利应用科学大学启发式与进化算法实验室（HEAL）赞助。F.O. de Franca 获得圣保罗州研究基金会（FAPESP）资助，资助编号为 2021/12706-1。G. Espada、L. Inglese 和 A. Fonseca 获得科学技术基金会（FCT）UIDB/00408/2020、UIDP/00408/2020、SFRH/BD/137062/2018、UI/BD/151179/2021，以及 CMU—葡萄牙双学位博士项目（SFRH/BD/151469/2021）、CMU—葡萄牙 CAMELOT 项目（LISBOA-01-0247-FEDER-045915）和 RAP 项目（编号 EXPL/CCI-COM/1306/2021）的资助。M. Landajuela、B. Petersen、R. Glatt、N. Mundhenk 和 C.S. Lee 获得实验室定向研究与开发项目 19-DR-003 的资助。他们的研究由美国能源部支持，在劳伦斯利弗莫尔国家实验室开展，合同编号为 DE-AC52- 07NA27344。劳伦斯利弗莫尔国家安全有限责任公司。LLNL-JRNL-846810。W.G. La Cava 获得美国国立卫生研究院 R00-LM012926 资助。M.S. Majumder 部分获得美国国立卫生研究院下属美国国立普通医学科学研究所的 R35GM146974 资助。资助方未参与研究设计、数据收集与分析、发表决定或论文准备。

声明

F. O. de Franca、M. Virgolin、M. Kommenda 和 W. G. La Cava 设计并组织了这场竞赛，并分析了结果。M. S. Majumder 担任真实世界赛道的领域专家。其余作者参加了竞赛，并在结果公布后为本文作出贡献。作者声明，与本文内容相关的利益冲突均不存在。

参考文献

[1]J. R. Koza，《遗传编程：通过自然选择对计算机进行编程的方法》。MIT出版社，1992年。[2] M. Cranmer、A. Sanchez Gonzalez、P. Battaglia、R. Xu、K. Cranmer、D. Spergel 和 S. Ho，“通过带有归纳偏置的深度学习发现符号模型”，

《神经信息处理系统进展》，第33卷，第17 429–17442页，2020年。[3] A. Hernandez、A. Balasubramanian、F. Yuan、S. A. M. Mason 和 T. Mueller，“通过符号回归实现快速、准确且可迁移的多体原子间势”，*npj*《计算材料》，第5卷，第1期，第112页，2019年11月。[4]M. Virgolin 和 S. P. Pissis，“符号回归是NP难问题”，《机器学习研究汇刊》，2022年。[在线]。可访问：https://openreview.net/forum? id= LTiaPxqe2e[5] B. K. Petersen、M. L. Larma、T. N. Mundhenk、C. P. Santiago、S. K. Kim 和 J. T. Kim，“深度符号回归：通过风险偏好策略梯度从数据中恢复数学表达式”，*arXiv*预印本 *arXiv:1912.04871*，2019年。[6]L. Biggio、T. Bendinelli、A. Neitz、A. Lucchi 和 G. Parascandolo，“可扩展的神经符号回归”，载于国际机器学习会议。PMLR，2021年，第936–945页。[7] P.-A. Kamienny、S. d’Ascoli、G. Lample 和 F. Charton，“使用Transformer实现端到端符号回归”，*arXiv*预印本 *arXiv:2204.10532*，2022年。[8] W. La Cava、P. Orzechowski、B. Burlacu、F. O. de França、M. Virgolin、Y. Jin、M. Kommenda 和 J. H. Moore，“当代符号回归方法及其相对性能”，*arXiv*预印本 *arXiv:2107.14351*，2021年。[9]R. Shwartz-Ziv 和 A. Armon，“表格数据：深度学习并非你所需的一切”，《信息融合》，第81卷，第84–90页，2022年。[10] J. H. Friedman，“贪心函数逼近：梯度提升机”，《统计年鉴》，第1189–1232页，2001年。[11]G. F. Bomarito、P. E. Leser、N. Strauss、K. Gar- brecht 和 J. Hochhalter，“利用量化不确定性自动学习可解释模型”，*arXiv*预印本 *arXiv:2205.01626*，2022年。[12] C. Rudin，“停止为高风险决策解释黑箱机器学习模型，改用可解释模型”，《自然·机器智能》，第1卷，第5期，第206–215页，2019年。[13] W. Roland、C. Marschik、M. Kommenda、A. Haghofer、S. Dorl 和 S. Winkler，“预测单螺杆挤出中的非线性输送行为：基于计算流体力学模拟比较各种基于数据的建模方法”，《国际聚合物加工》，第36卷，第5期，第529–544页，2021年。 [在线]。可访问：https://doi.org/10.1515/ipp-2020-4094[14] P. Orzechowski、W. La Cava 和 J. H. Moore，“我们现在处于什么阶段？近期符号回归方法的大规模基准测试研究”，载于*2018*年遗传与进化计算会议论文集，丛书 GECCO ’18，2018年4月。[15] J. D. Romano、T. T. Le、W. La Cava、J. T. Gregg、D. J. Goldberg、P. Chakraborty、N. L. Ray、D. Himmelstein、W. Fu 和 J. H. Moore，“PMLB v1.0：用于机器学习基准测试的开源数据集集合方法》，生物信息学，2022年。[16]R. S. Olson、W. La Cava、P. Orzechowski、R. J. Urbanowicz 和 J. H. Moore，《PMLB：用于机器学习评估与比较的大型基准测试套件》，生物数据挖掘，2017年。ˇ[17] A. Meurer、C. P. Smith、M. Paprocki、O. Certík、S. B. Kirpichev、M. Rocklin、A. Kumar、S. Ivanov、J. K. Moore、S. Singh 等，《Sympy：Python 中的符号计算》，*PeerJ* 计算机科学，第 3 卷，第 e103 页，2017年。[18] D. L. Randall、T. S. Townsend、J. D. Hochhalter 和 G. F. Bomarito，《Bingo：一种用于遗传编程符号回归的可定制框架》，载于遗传与进化计算会议伴会论文集，2022年，第2282–2288 页。[19] H. Zhang 、 A. Zhou 、 H. Qian 和 H. Zhang，《PS-Tree：一种分段符号回归树》，群体与进化计算，第 71 卷，第 101061 页，2022年。[20]K. R. Broløs、M. V. Machado、C. Cave、J. Kasak、V. Stentoft-Hansen、V. G. Batanero、T. Jelen 和 C. Wilstrup，《一种使用 Feyn 进行符号回归的方法》，*arXiv*预印本 *arXiv:2104.05417*，2021年。[21] B. He、Q. Lu、Q. Yang、J. Luo 和 Z. Wang，《用于符号回归的泰勒遗传编程》，*arXiv*预印本 *arXiv:2205.09751*，2022年。[22] S. Sahoo、C. Lampert 和 G. Martius，《学习用于外推和控制的方程》，载于国际机器学习会议。PMLR，2018年，第 4442–4450 页。[23]G. Espada、L. Ingelse、P. Canelas、P. Barbosa 和 A. Fonseca，《将数据类型作为语法引导遗传编程中更符合人体工程学的前端》，载于第 *21* 届 *ACM SIGPLAN* 生成式编程国际会议：概念与经验（*GPCE 2022*），奥克兰，新西兰，*2022*年*12*月*6*日至*7*日论文集，B. Scholz 和 Y. Kameyama 编。ACM，2022年，第 86–94 页。 [在线]。可访问：https://doi.org/10.1145/3564719.3568697[24] B. Burlacu、G. Kronberger 和 M. Kommenda，《Operon：一种高效的符号回归遗传编程框架》，载于*2020*年遗传与进化计算会议伴会论文集，2020年，第 1562–1570 页。[25] M. Cranmer，《PySR：Python/Julia 中快速且并行化的符号回归》，2020年。[26]M. Landajuela、C. Lee、J. Yang、R. Glatt、C. P. Santiago、I. Aravena、T. N. Mundhenk、G. Mulcahy 和 B. K. Petersen，《深度符号回归的统一框架》，载于《神经信息处理系统进展》，A. H. Oh、A. Agarwal、D. Belgrave 和 K. Cho 编，2022年。 [在线]。可访问：https: //openreview.net/forum?id=2FNnBhwJsHK[27]G. Dick、C. A. Owen 和 P. A. Whigham，《特征标准化与系数优化：实现有效符号回归》，载于*2020*年遗传与进化计算会议论文集，GECCO ’20 系列。美国纽约：美国计算机协会，2020年，第 306–314 页。 [在线]。可访问：https://doi.org/10.1145/3377930.3390237

[28]D. Izzo、F. Biscani 和 A. Mereta，《可微遗传编程》，载于遗传编程：第 *20* 届欧洲会议 *EuroGP 2017*，阿姆斯特丹，荷兰，*2017*年*4*月*19*日至*21*日，论文集 *20*。施普林格，2017年，第 35–51 页。[29] M. Kommenda、A. Beham、M. Affenzeller 和 G. Kronberger，《多目标符号回归的复杂度度量》，载于计算机辅助系统理论国际会议。施普林格，2015年，第 409–416 页。[30] M. Virgolin、A. De Lorenzo、E. Medvet 和 F. Randone，《学习可解释性公式以学习可解释公式》，载于国际自然问题并行求解会议。施普林格，2020年，第 79–93 页。[31]Anaconda，《Anaconda 软件发行版》。 [在线]。可访问：https://anaconda.com/[32]F. Pedregosa、G. Varoquaux、A. Gramfort、V. Michel、B. Thirion、O. Grisel、M. Blondel、P. Prettenhofer、R. Weiss、V. Dubourg、J. VanderPlas、A. Passos、D. Cournapeau、M. Brucher、M. Perrot 和 E. Duchesnay，《scikit-learn：Python 中的机器学习》，机器学习研究期刊，第 12 卷，第 2825–2830 页，2011年。 [在线]。可访问：https: // dl.acm.org/doi/10.5555/1953048.2078195[33] M. D. Schmidt 和 H. Lipson，《适应度预测器的协同进化》，*IEEE*《进化计算汇刊》，第 12 卷，第 6 期，第 736–749 页，2008年。[34] T.-W. Huang、D.-L. Lin、C.-X. Lin 和 Y. Lin，《Taskflow：一种轻量级并行异构任务图计算系统》，*IEEE*《并行与分布式系统汇刊》，第 33 卷，第 6 期，第 1303–1320 页，2022年。[35] T. Akiba、S. Sano、T. Yanase、T. Ohta 和 M. Koyama，《Optuna：新一代超参数优化框架》，载于第 *25* 届 *ACM SIGKDD* 国际知识发现与数据挖掘会议论文集，2019年。[36] S.-M. Udrescu、A. Tan、J. Feng、O. Neto、T. Wu 和 M. Tegmark，《AI Feynman 2.0：利用图模块性的帕累托最优符号回归》，《神经信息处理系统进展》，第 33 卷，第 4860–4871 页，2020年。[37] S. L. Brunton、J. L. Proctor 和 J. N. Kutz，《通过对非线性动力系统进行稀疏辨识，从数据中发现控制方程》，《美国国家科学院院刊》，第 113 卷，第 15 期，第 3932–3937 页，2016年。[38] T. Mundhenk、M. Landajuela、R. Glatt、C. P. Santiago、B. K. Petersen等，《通过深度强化学习增强的遗传编程种子生成实现符号回归》，《神经信息处理系统进展》，第 34 卷，第 24 912–24923 页，2021年。[39]G. Dick，《重新审视遗传编程、标准化和随机梯度下降：SRBench 的初步发现》，载于遗传与进化计算会议伴会论文集，GECCO '22 系列。美国纽约：美国计算机协会，2022年，第 2265–2273 页。 [在线]。可获取：https://doi.org/10.1145/3520304.3534040

附录

为了参赛，候选者必须提供一个安装脚本，使其能够在 Linux 系统上的 *conda* [4] 环境 [31],中无需管理员权限即可运行；还需提供一份 *metadata.yml* 文档来描述其提交内容，以及一个名为 *regression.py* 的 Python 脚本，其中包含一个 *scikit-learn* [5]回归器对象 [32], 和一个函数，用于返回与 *sympy*[6] [17]兼容的符号回归模型字符串。每次运行中，候选者的符号回归算法都有预先规定的运行时间预算：对于不超过 1000 个样本的数据集，预算为 1 小时；对于不超过 10000 个样本的数据集，预算为 10 小时。候选者有责任确保其算法的运行时间（包括超参数优化的选择）不超过预算。为尽可能公平地进行比较，参赛者不会提前获知竞赛将使用哪些数据集。他们只知道数据集将遵循宾夕法尼亚机器学习基准（PMLB） [15, 16], 的格式，并且资格赛阶段将使用来自 PMLB 的数据集。

为了让外部同行能够参与并复现这场竞赛，我们在 SRBench 仓库中创建了一个新的分支[7]，其中说明了如何添加新方法，以及如何在本地计算机上运行竞赛。只要遵循这些要求，公众便可以通过克隆竞赛分支并按照安装和运行说明，复现竞赛并试用这些符号回归算法。

表 IV 展示了资格赛阶段使用的精选数据集。这些数据集是从 PMLB 数据集中选出的一个子集，在这些数据集上，srbench 排名前十的算法表现相当。

由于我们针对每条赛道采用了多项评估标准：R^2简洁性，以及每项任务的任务特定分数，因此我们计算了综合排名，即针对 n不同数据集，将各项标准的排名取调和平均数。

$$\text{agg_rank} = \frac{n}{\sum_{i=1}^{n} \frac{1}{\text{rank}_i}} \tag{7}$$

调和平均数意味着，若要获得较高排名，就不能在任何一项评估标准上排名过低。这避免了符号回归算法返回非常简单但 R^2 较低的模型，却仍在参赛者中名列前茅的情况。

简洁性得分定义为 round(–log$_5$(s)，1)，其中 s 表示经 *sympy* [17]，简化后表达式树中的节点数，而 round(x，n) 会将值 x 舍入到小数点后第 n 位。引入舍入是为了给大小相近的表达式提供一定的容差。图 13 展示了简洁性得分与树中节点数之间的关系。

[4]https://www.anaconda.com/ [5]https://scikit-learn.org/ [6] https://www.sympy.org/en/index.html [7] https://github.com/cavalab/srbench/tree/Competition2022

表 IV 资格赛阶段使用的数据集。

数据集	样本数	变量数
197_cpu_act	8192	22
215_2dplanes	40768	11
227_cpu_小型	8192	13
556_analcatdata_呼吸暂停2	475	4
557_analcatdata_呼吸暂停1	475	4
564 fri_	40768	11
573_cpu_活动	8192	22
218_房屋 _8L	22784	9
225_puma8NH	8192	9
294_卫星_图像	6435	37
666_rmftsa _ladata	508	11
579_fri_c0_250_5	250	6
586 fri c3 1000 25_ _ _	1000	26
590 fri c0 1000 50_ _ _ _	1000	51
593_fri_c1_1000_10	1000	11
595_fri_c0_1000_10	1000	11
654 fri c0 500 10_ _ _ _	500	11
581 fri c3 500 25_ _ _ _	500	26
582_fri_c1_500_25	500	26
584_fri_c4_500_25	500	26

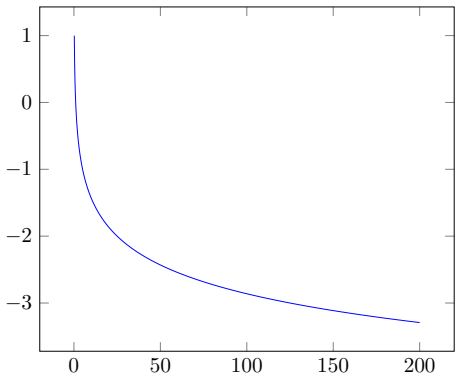


图 13. 简洁性得分图。

Bingo [18]采用了一种用于符号回归的遗传编程策略，与典型的基于通用编程的符号回归有所不同。首先，表达式以无环图而非树的形式表示。表达式的适应度计算分为两个步骤：先对无环图进行代数化简，再对数值常数执行基于梯度的局部优化。最后，对于大型数据集（ $>$ 1200 个数据点），采用适应度预测器的协同进化 [33] 。该方法的若干超参数在提交前经过调优（详情见 [18]），以着重获得低复杂度模型。

E2ET [7] 通过生成模型，根据给定的观测值生成解析表达式。具体而言，基于 Transformer 的神经网络以观测值为条件，生成表达式的分布。从该分布中采样时，会对表达式符号进行自回归解码，其中包括变量、运算符词元和词元化浮点数。为了进一步改进生成的表达式，使用 BFGS 算法优化常数。这个搜索过程效率很高，对单次前向传播而言仅需约一秒。

PS-Tree [19], ，即分段符号回归树，是一种分段回归方法它结合决策树、遗传编程和岭回归，将特征空间划分为多个子空间，并在各子空间中创建一组非线性模型进行预测。所提出的 PS-Tree 旨在实现较高的预测性能，同时保持合理的可解释性。具体而言，PS-Tree 在多目标优化框架下演化一组遗传编程树，同时为所有子空间中的岭回归模型演化一组通用的非线性特征。随后，它在进化过程中动态构建决策树，根据运行时信息自动确定最优划分方案。通过这种方式，PS-Tree 能够以合理的可解释性，高准确率地解决分段回归任务。

QLattice [20] 是一个结合遗传编程和数值优化的符号回归框架。用户可通过 Python 库 feyn 使用它。QLattice 管理一个模型池，并通过对池中表现最佳的模型进行编辑来采样生成新模型。只有在使用内部调优的梯度下降算法，根据训练数据优化每个模型的自由参数后，才会评估模型性能。模型可用的编辑操作通过加权采样选取，其权重由 Abzu 优化。该框架也原生支持包含类别特征的训练数据：每个可能的类别都会被赋予一个权重，该权重通过拟合训练数据得出。在本次竞赛中，最终模型集合中由贝叶斯信息准则选出的最佳模型被返回。

TaylorGP [21] 将符号回归与泰勒多项式近似相结合。该方法利用泰勒展开来识别多项式特征，并将问题分解为更简单的任务。

EQL [22] 是一种完全可微的符号回归神经网络，因此可用于需要可微端到端计算的场景。它通过使用带有非线性激活函数的浅层网络来近似方程，并可根据具体任务进行调整。尽管可以通过不同方式实现稀疏性，但在本次竞赛中，我们优化了展开参数的随机 L^0 损失，并根据拟合质量和表达式的简洁性，按照帕累托最优性选出最佳表达式。表达式确定后，最后使用 BFGS 优化其中出现的常数。

GeneticEngine [23]是一个支持多种强类型遗传编程和文法引导遗传编程的 Python 框架。该参赛方案采用树形表示，因为与基于数组的方法相比，它能够产生质量更高的交叉操作。文法通过标准 Python 类和继承来定义，从而能够轻松地将领域知识融入搜索过程。尽管这是该方法的主要优势，但本次竞赛中，文法并未使用任何领域特定信息。

Operon [24] 是一个用于符号回归（SR）的 C++ 框架，注重性能与可扩展性。它支持单目标和多目标的基于树的遗传编程（GP），并可通过列文伯格-马夸尔特算法对模型系数进行局部调优。并行性通过一种

灵活的基于图的任务模型 [34]在单个重组事件层面实现。此外，树求值过程中还采用了数据级并行。Python 模块 PyOperon 将此功能封装为 scikit-learn 估计器。在本次竞赛中，Operon 与 Optuna [35] 配合进行超参数调优，返回最后一代帕累托前沿中的最佳模型。

PySR [25] 及其配套后端 SymbolicRegression.jl，是一种基于锦标赛选择和局部叶节点搜索、用于树结构表达式的进化优化算法。PySR 注重性能，且具有很强的可配置性——它不会对表达式空间施加特定的先验，也不会定义任何标准算子或函数形式。PySR 通过将算子即时编译为融合的 SIMD 内核，并在种群级进行多线程处理，实现高效评估。搜索过程中会记录完整的表达式帕累托前沿；但在本次竞赛中，最终返回的是沿帕累托前沿准确率提升幅度最大的方程。

uDSR [26]，即统一深度符号回归，是一种模块化、统一的符号回归框架，结合了五种解决策略：递归式问题简化 [36]，神经引导搜索[5],用于问题泛化的大规模预训练 [6],使用非线性基函数进行稀疏线性回归 [37],以及将进化搜索与学习相结合 [38]。通过将这些模块视为算法框架中彼此关联但互不重叠的组成部分，uDSR 得以发挥各自的独特优势并克服相应的局限性。这种模块化设置甚至适用于仅采用部分所提模块的情况，并允许在有新方法可用时轻松替换单个模块。uDSR 基于开源的 Deep Symbolic Optimization 框架 [8]实现。

遗传编程$_{ZGD}$ [27] 是针对符号回归（SR）直接实现的 Koza 风格的规范遗传编程 [1]，并在进化过程中对模型系数应用随机梯度下降（SGD）。为提升 SGD 的效果，所有回归变量均采用 Z 分数标准化。我们对 Koza 的规范设置进行了超参数调优，以优化其在符号回归中的表现，具体方法见 [39]。遗传编程$_{ZGD}$ 是临近截止时提交的竞赛作品，因此仅在竞赛的资格赛阶段接受评估。

NSGA-DCGP [28] 是一种将可微笛卡尔遗传编程 [28] 与 NSGA-II 相结合的方法，其优化目标是最小化误差和复杂度。

竞赛的完整结果文件可通过以下统一资源定位符获取：https://zenodo.org/record/6842176。运行分析所需的代码可通过以下网址获取：https://cavalab.org/srbench/competition-guide/。

[8]https://github.com/brendenpetersen/deep-symbolic-optimization