arXiv:2304.03540v1 [cs.DB] 7 Apr 2023

ChatPipe：通过优化人类与 ChatGPT 的交互来编排数据准备程序

Sibei Chen 中国人民大学 sibei@ruc.edu.cn

Hanbing Liu 中国人民大学 liuhanbing@ruc.edu.cn

Weiting Jin 中国人民大学 waiting_jin@ruc.edu.cn

Xiangyu Sun 中国人民大学 sunxiangyu0906@ruc.edu.cn

Xiaoyao Feng 中国人民大学 fengxiaoyao668@ruc.edu.cn

Ju Fan 中国人民大学 fanj@ruc.edu.cn

Xiaoyong Du 中国人民大学 duyong@ruc.edu.cn

Nan Tang QCRI ntang@hbku.edu.qa

摘要

编排高质量的数据准备程序对于机器学习（ML）的成功至关重要，但众所周知，这一过程耗时费力。尽管 ChatGPT 等大语言模型能够通过自然语言提示与用户交互并生成程序，展现出令人瞩目的能力，但仍存在一些局限。具体而言，用户必须提供具体的提示词，反复引导 ChatGPT 改进数据准备程序；这要求用户对编程、所用数据集以及机器学习任务具备一定程度的专业知识。此外，程序生成后，若不从头开始，想要恢复到之前的版本或修改程序并非易事。

本文介绍了 ChatPipe，这是一个旨在促进用户与 ChatGPT 无缝交互的新型系统。ChatPipe 会向用户有效推荐接下来的数据准备操作，并引导 ChatGPT 为这些操作生成程序。此外，ChatPipe 使用户能够轻松回滚到程序的先前版本，从而更高效地开展实验和测试。我们为 ChatPipe 开发了一个 Web 应用，并从 Kaggle 准备了若干真实世界的机器学习任务。这些任务能够展示 ChatPipe 的能力，使 VLDB 参会者可以轻松体验我们的新功能，快速编排高质量的数据准备程序。

1 引言

作为数据分析中不可或缺的一步，数据准备（或简称数据准备）将原始数据转换为可用于机器学习（ML）或数据科学任务的格式。典型的数据准备流程包含一系列操作，例如处理缺失或无效数据、对数据进行归一化以及特征工程，在此过程中，数据会从一个操作流向后续操作。在本文中，我们将这类操作序列称为数据预处理程序。

在实践中，领域专家（如医生）可能会发现，由于需要编程技能和数据预处理经验，编排数据预处理程序颇具挑战。近年来，大语言模型，尤其是ChatGPT，已成为协助领域专家编排数据预处理程序的一种有前景的工具。不同于现有的数据预处理工具，ChatGPT 只需与用户进行简单的“聊天”，即可直接生成数据预处理程序。

然而，用户很难提供具体的提示词来引导 ChatGPT 改进数据预处理程序。主要原因是，用户可能不知道针对特定数据集和特定任务，哪些操作最有帮助。此外， it is

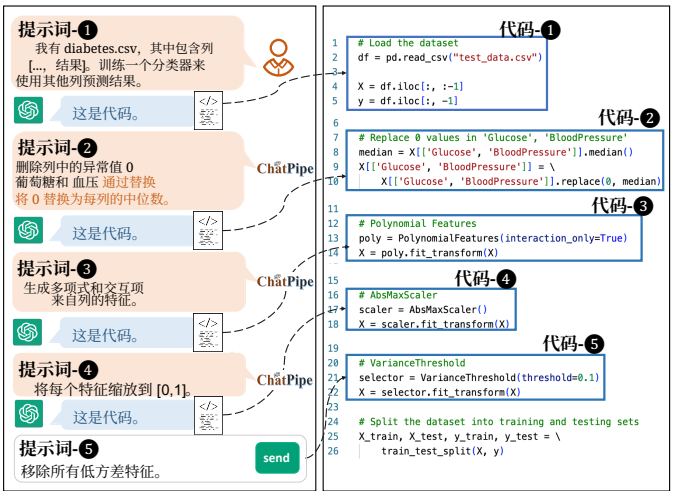

图 1：ChatPipe 概览，它通过建议操作来改进数据预处理程序编排中的人与ChatGPT的交互，从而提供指导。

此外，如果当前程序产生较差的性能，用户也很难重新查看之前的程序版本。因此，非常希望为用户配备一名助手，能够通过向 ChatGPT 建议改进程序的操作来提供有效指导，并轻松回滚到任何之前的版本，以促进实验和测试。

我们的方案。 我们提出了 ChatPipe，这是一款通过建议操作来改进人与ChatGPT的交互，从而为数据预处理程序编排提供有效指导的工具。如图 1 所示，ChatPipe 支持用户与 ChatGPT 无缝沟通，并按如下方式生成数据预处理程序。

(1) 首先，用户上传一个数据集（例如，diabetes.csv），并向 ChatGPT 提供提示词（即，**提示词-❶**）。随后，ChatGPT 生成初始代码块，即 **代码-❶**，用于加载数据集并准备特征和标签。

(2) ChatPipe 分析数据集，并检测到列 Glucose 和 BloodPressure 中的异常值（例如 0），因此建议执行异常值移除操作。此外，ChatPipe 允许用户更新提示词以注入领域知识，例如，将所有 0 替换为各列的中位数。在提示词（**提示词-❷**）的指导下，ChatGPT 生成代码块**代码-❷**。

(3) 通过与用户交互，ChatPipe推荐一系列操作 多项式特征 **(提示词-❸)**、最大绝对值缩放器

(提示词-❹)，以及方差阈值（**提示词-❺**），它们专为特征工程量身定制。我们可以看到，这些建议的操作非常合理且连贯，即，先从列生成交互特征，然后缩放生成的特征，最后移除低方差特征。

通过这种方式，ChatPipe 可以引导 ChatGPT 通过人与ChatGPT的交互生成有效的数据预处理程序，从而加快耗时的“反复试错”数据预处理过程。

ChatPipe 的架构ChatPipe。为支持上述功能，ChatPipe 由三个关键模块组成。(1) 操作推荐：给定当前程序，ChatPipe 会推荐最有效的操作，以提升机器学习任务的整体性能。该模块采用了我们研究论文 [1] 中提出的技术。(2) 程序生成：根据推荐的操作，ChatPipe 会与 ChatGPT 交互，生成无错误的数据预处理程序，以有效实现这些操作。(3) 版本管理：人与ChatGPT的交互会生成多个程序版本，包括生成的数据集及相应的提示词。为使用户能够回滚到任意先前版本，ChatPipe 会将程序版本可视化，并开发加速程序执行的技术。

演示场景。我们将ChatPipe 构建为一个 Web 应用程序，并通过为来自最受欢迎的数据科学网站 Kaggle 的真实机器学习任务编排数据预处理程序来进行演示。我们准备了一组机器学习任务，每个任务都包含一个数据集，例如图 1 中的 diabetes.csv。我们允许 VLDB 与会者选择机器学习任务，然后利用 ChatPipe 与 ChatGPT 交互，以生成有效的数据预处理程序。在此过程中，我们展示ChatPipe 的创新功能，包括操作推荐、程序生成和版本管理。演示视频可在 YouTube[1] 上观看。

总而言之，我们做出了以下贡献。(1) 我们开发了 ChatPipe，这是一款新颖的工具，可优化用于数据预处理程序编排的人机 ChatGPT 交互。(2) ChatPipe 可通过建议用于生成程序的操作，为 ChatGPT 提供有效指导，并使用户能够跳转到不同的程序版本。(3) 我们部署了 ChatPipe
将其构建为具有用户友好界面的 Web 应用，并在真实的数据准备场景中展示其效用。

2 系统概述

ChatPipe 的架构如图 2 所示。给定一个机器学习任务，例如基于数据集训练用于糖尿病预测的分类器，ChatPipe 旨在通过优化人与 ChatGPT 之间的交互来编排数据预处理程序。为实现这一目标，ChatPipe 由三个关键模块组成。操作推荐根据当前程序和数据集，为用户提供下一步数据预处理操作的有效建议，例如处理缺失值和特征工程（第 2.1 节）。程序生成与 ChatGPT 交互，生成无错误的数据预处理程序，从而有效实现推荐的操作（第 2.2 节）。版本管理将人与 ChatGPT 交互生成的各种程序版本进行可视化，并支持用户快速回滚到先前版本，以提高实验和测试的效率（第 2.3 节）。

[1]https://youtu.be/bHwEXaA9y6Y

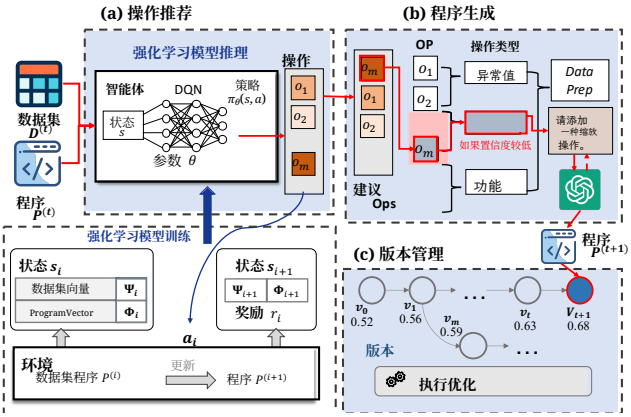

图 2：ChatPipe 的架构，由三个关键模块组成：(a) 操作推荐、(b) 程序生成和 (c) 版本管理。

2.1 操作推荐

操作推荐面临的主要挑战有两个方面。首先，数据预处理操作的类型多种多样（例如，缩放和离散化），即使对于同一种类型，也存在许多可能的操作（例如，MinMaxScaler 和 StandardScaler）。其次，推荐操作的方式可能有很多种，并且高度依赖于具体数据集和任务。为解决这一问题，我们提出了一种基于强化学习（*RL*）的方法，通过离线训练学习如何推荐操作。在推理阶段，给定数据集和当前程序，我们使用训练好的模型来推荐最合适的操作。有关我们方法的更多详细信息，请参阅我们的研究论文 [1]。

强化学习框架。我们将操作推荐问题表述为由智能体执行的序贯决策过程。智能体基于策略，将当前程序 $P(i)$和数据集 $D(i)$ 作为状态，并执行动作，即 从预定义的搜索空间中选择一个操作。随后，智能体从环境中获得奖励 ，并据此更新策略。

状态：我们根据当前程序 $P(i)$和生成的数据集 $D(i)$表示状态。具体而言，我们使用统计信息作为数据集特征，并利用公开的预训练代码表示模型 UniXcoder [4] 提取程序特征。然后，我们将这些特征拼接起来表示状态。

动作：我们将动作定义为从预定义的操作集合 O 中选择一个操作 o_i，将程序 $P(i)$ 转换为 $P(i+1)$。

奖励：奖励用于指示所执行的动作是否有益。我们运行程序 $P(i+1)$ 来评估其性能，例如，训练所得分类器的 F1 分数。

深度Q网络（深度Q网络）。我们采用深度Q网络（深度Q网络）框架来优化策略函数，如图2所示。具体而言，遵循深度Q网络中的典型策略，我们采用神经网络来近似深度Q网络中的价值函数，该网络以状态作为输入，并为每个动作（即操作）生成一个价值。该神经网络包含多个全连接层，以 LeakeyRelu作为激活函数，并包含一个输出tanh层。

深度Q网络训练与推理。图2说明了深度Q网络训练和推理的过程。我们的深度Q网络训练算法使用离策略策略，在多个回合中学习 ϵ-贪心策略。具体而言，在第 i个回合中，它首先计算当前状态 s_i，然后以概率 $1-\epsilon$选择使价值函数最大化的贪心动作，并以概率 ϵ选择随机动作。给定所选动作 a_i（即，操作），该算法将程序从 $P(i)$ 更新为 $P(i+1)$，并执行 $P(i+1)$ 以获得奖励 r_i。随后，可以使用随机梯度下降更新深度Q网络的参数。

对于推理，给定一个新的程序 $P(t)$和数据集 $D(t)$，我们提取它们的特征来表示状态，然后利用训练好的模型计算候选集中的操作的值。最后，我们将这些操作维护为一个排序列表，并将其输入下文介绍的程序生成 模块。

2.2 程序生成

程序生成通过自然语言提示词与 ChatGPT 交互，以生成数据预处理程序。这里的挑战在于如何引导 ChatGPT 生成无错误的数据预处理程序，从而有效实现这些操作。为应对这一挑战，我们开发了三个组件：操作增强、程序优化和程序检查，具体如下。

操作增强。 一种直接的方法是直接“翻译”操作推荐返回的最佳操作。例如，对于标准化缩放器操作，可以使用如下提示词：“通过减去均值并缩放至单位方差来标准化特征”。然而，由于操作来自一组受限的候选操作（见第 2.1 节），在某些情况下，操作的改进可能并不显著。

幸运的是，我们观察到一个有趣的现象：ChatGPT 可能会对这些情况有所帮助。具体而言，当我们使用一个粗粒度的提示词来引导 ChatGPT 时，例如，“生成交互特征”，它可能会生成一个定制程序，例如 df[‘AB’] = df[‘A’] * df[‘B’],，而不是简单地使用预定义的操作。这启发我们将操作组织成一个两级层次结构，其中底层是预定义操作集合中的细粒度操作（例如，sklearn.preprocessing.MinMaxScaler），上层是操作符的类型（例如，缩放器）。基于这两级层次结构，我们生成提示词如下。当推荐操作的置信度较高时，例如，DQN 的输出分数大于某个阈值，我们直接选择细粒度操作。否则，我们选择最佳的操作类型，即其细粒度操作平均分数最高的类型，以生成粗粒度提示词。

程序优化。 该组件要求用户通过注入其领域知识或数据预处理经验来优化生成的程序。例如，用户可以利用领域知识来确定如何对 BMI进行离散化，例如，“BMI应被划分为 $[0, 18.5, 25, 30, 100]$，包含 [‘体重过轻’、‘正常’、‘超重’、‘肥胖’]”。此外，用户还可以自定义所推荐操作的参数，既可以优化提示词，也可以更新程序。

程序检查。 该组件检查生成的程序是否存在运行时错误，并在发现错误时修复这些错误。为此，在将程序发送给用户之前，ChatPipe 首先在本地运行时环境中运行生成的程序。如果发生某些错误，例如，“名称错误：名称 ’多项式特征’未定义。”，ChatPipe会将错误日志发送给 ChatGPT，而ChatGPT 可以修复错误并提供更新后的程序。再次考虑上述示例。ChatGPT 会在程序开头添加一行：*“fromsklearn.preprocessingimport多项式特征”*。它会迭代运行此过程，直到程序不再报错。

2.3 版本管理

人与ChatGPT的交互会生成程序的多个版本，包括生成的数据集和相应的提示词。为使用户能够回滚到任何先前的版本，版本管理 旨在将程序版本可视化，并支持快速版本切换。该模块包含两个组件：(1) 程序版本管理，用于维护程序之间的版本；以及 (2) 数据缓存，通过跨版本缓存中间数据变量来加速执行。

程序版本管理。我们使用关系数据库存储每个机器学习任务的版本相关信息，每个版本都包含相应的程序、执行结果和提示词。我们还维护版本之间的关系，以便将版本可视化为树结构（见图 2）。此外，用户可以通过与 ChatGPT 交互来添加新版本、删除无用版本，还可以选择任意两个版本来比较它们在程序、数据和提示词方面的差异。

跨版本数据缓存。程序版本之间可能只有细微差异，因为这些版本中可能有很大比例的代码块是相同的。因此，缓存一些可能会在不同版本间重复使用的中间变量是很自然的做法，这样可以快速切换版本。

我们研究了跨版本数据缓存中的两个技术挑战。首先，一些研究表明，存储所有中间变量并不现实 [3]。因此，有必要选择最“有益”的变量进行缓存。为应对这一挑战，我们将缓存决策建模为一个二元分类问题：给定一个数据变量，估计该变量适合缓存的概率。我们设计了一种深度神经网络来预测缓存概率，其特征包括变量的存储大小、计算时间、先前的操作以及下游机器学习任务的准确率。第二个挑战是确定何时可以重用缓存变量。例如，如果计算该变量的代码块发生变化，那么缓存的数据就会过时，因此不能重用。为此，我们使用抽象语法树（AST）分析每个变量的计算轨迹，并使用一种优化的最大流算法 [3] 来判断变量应当重用还是重新计算，

3 演示场景

在即将举行的演示中，我们从 Kaggle 精选了 62 个机器学习任务，涵盖教育、医学和金融等 10 多个不同领域。欢迎 VLDB 参会者选择自己最熟悉的数据集或领域进行体验。

接下来，我们将通过一个具体示例介绍如何使用 ChatPipe，并重点展示两项关键特征。首先，ChatPipe 可以通过建议下一步操作，协助用户与 ChatGPT 互动。

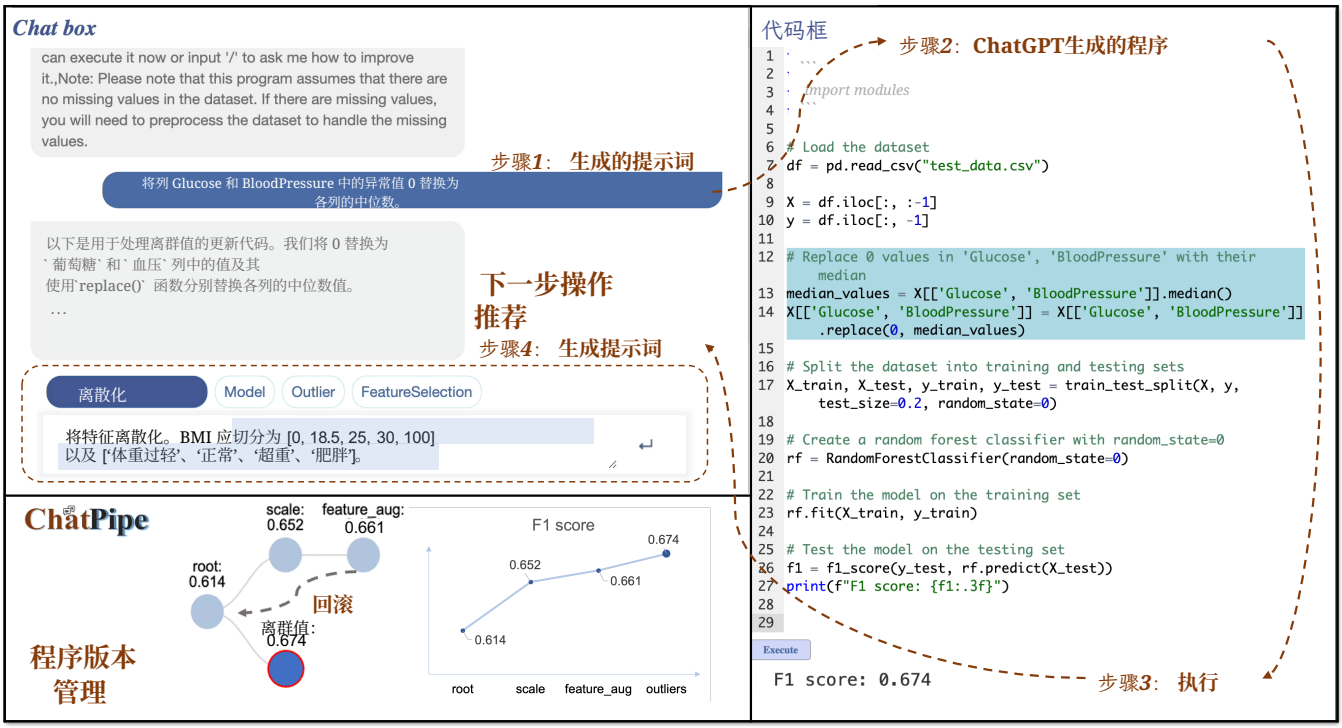

图 3：ChatPipe 的演示场景。

其次，ChatPipe 可以管理用户历史记录，让用户回退到之前的版本并比较差异。

机器学习任务。 我们的示例机器学习任务是预测患者是否患有糖尿病 [2]。该数据集包含 769 个元组，以及八个作为特征的数值型诊断测量值。我们使用 sklearn。model_selection。train_test_split 将训练集和测试集进行划分，默认使用随机_状态 = 0 。

设置。 用户将糖尿病数据集上传到 ChatPipe，并指定用于分类的列名或索引。随后，她可以通过在代码框中手动编写代码或上传现有代码来生成代码的初始版本。或者，她也可以使用 ChatPipe 为其生成代码。代码生成后，即可执行并保存。之后，用户通过与 ChatPipe 进行迭代式交互，不断提升分类任务的 F1 分数。

图 3 展示了一个示例：用户已与 ChatPipe 交互四次，正准备发起第五次交互，具体如下所述。

（*1*）下一步操作推荐。 在图 3 中，用户确认了 ChatPipe 推荐的操作（或提示词）：“移除列 *Glucose* 中值为 *0* 的异常值 。。。”，该操作将发送给 ChatGPT 以更新代码，如图 3 的代码框所示。浅蓝色部分突出显示了新增的离群值处理操作。

之后，用户执行新代码，并观察到 F1 提升（0.674 vs. 0.614）。随后，用户输入“/”符号，触发 ChatPipe 显示几项推荐操作（例如，输入框上方的文本气泡）。这些操作会由ChatPipe排序并显示。选择一个操作（在此例中为“离散化”）后，相应的提示词（“离散化特征”）会自动填入输入框。

用户还可以运用领域知识并添加额外的提示词，例如“*BMI* 应划分为 [0, 18.5, 25, 30, 100]，类别包括 [‘体重过轻’、‘正常’、‘超重’、‘肥胖’]”。

（*2*）程序版本管理。ChatPipe 会保存所有更新并执行过的程序，可在代码编辑器中查看。不同程序的执行结果（例如，F1 分数）可以通过统计折线图查看。ChatPipe 将这些信息作为版本进行管理，并将它们组织成版本树。用户可以点击相应节点来切换当前版本。在图 3 中，用户从 ChatPipe 自动生成的代码根节点初始化版本开始。她随后通过提示词添加缩放操作，接着添加特征增强操作。在第三次交互中，她决定探索不同选项，于是点击根节点返回原始版本。然后，她选择离群值处理操作，进一步进行探索。

参考文献

[1] Sibei Chen、Nan Tang、Ju Fan、Xuemi Yan、Chengliang Chai、Guoliang Li 和 Xiaoyong Du。2023（即将发表）。HybridPipe：结合人工生成和机器生成的数据准备流水线。发表于*SIGMOD*。[2] Kaggle 糖尿病数据集。 [无日期]。https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset。[3] Behrouz Derakhshan、Alireza Rezaei Mahdiraji、Ziawasch Abedjan、Tilmann Rabl 和 Volker Markl。2020。优化协作环境中的机器学习工作负载。发表于*SIGMOD*。1701–1716。[4] Daya Guo、Shuai Lu、Nan Duan、Yanlin Wang、MingZhou 和 Jian Yin。2022。Unix-coder：用于代码表示的统一跨模态预训练。*arXiv:2203.03850*（2022）。