arXiv:2303.12980v1 [physics.chem-ph] 23 Mar 2023

GMP-Featurizer：用于高效计算原子系统高斯多极特征的并行化 Python 软件包

Xiangyun Lei 丰田研究院 加利福尼亚州洛斯阿尔托斯 ray.lei@tri.global

Joseph Montoya 丰田研究院 加利福尼亚州洛斯阿尔托斯 joseph.montoya@tri.global

关键词 Python · C++ · 并行化 · 机器学习 · 化学 · 分子动力学

1 摘要

GMP-Featurizer 是一款轻量、准确、高效且可扩展的软件包，可计算多种原子系统的高斯多极（GMP）特征 [1]，涵盖元素周期表中的各种元素。该软件包以 AmpTorch [2], 中的 GMP 特征计算模块为基础，此后其能力得到了显著提升，包括准确性和效率，以及在不同核心乃至不同机器上进行并行化的能力。此外，这个 Python 软件包仅依赖少量标准 Python 库，以及用于 C++代码接口的 cffi 和用于并行化的 Ray [3]，因此轻巧而稳健。我们设计了一组单元测试，以确保输出可靠。软件包还包含大量示例和教程，以及两套赝势文件（用于指定 GMP 特征集），供用户使用。总而言之，该软件包旨在为有意开发基于 GMP 特征的模型的化学与材料科学家提供一种标准实现。该软件包的源代码依据 Apache 2.0 许可证免费向公众开放。

2 需求说明

以描述性强且高效的方式表示原子系统中的局部和全局环境，一直是化学、化学工程和材料科学领域的重要研究课题。事实证明，良好的化学环境表示（即特征）对于构建可靠的机器学习（ML）模型至关重要。这些模型能够准确预测原子系统的性质，在少数情况下甚至已用于发现或设计新的化学品和材料[4, 5]。迄今为止，科学家和研究人员设计了多种特征化方案，例如原子中心对称函数（ACSF） [6], 、原子位置平滑重叠（SOAP） [7] 和高斯动量 [8] 方案。近年来，基于图表示的机器学习模型（例如 MEGNet[9], 、CGCNN[10]）也取得了成功。高斯多极（Gaussian Multipole，简称 GMP） [1], 是一种近期开发的局部化学环境特征化方案，也就是对分子和晶体结构中各个原子附近空间的化学特征进行表征。GMP 使用多极展开来近似底层的局部电子环境（例如局部电子云的近似分布）；相关理论已在先前的出版物中阐述[1]。该特征化方案具有灵活性，只依赖于对原子身份和位置的先验假设，因此适用于已知原子排列的各种原子系统（分子、纳米颗粒、周期性晶体等）。特征计算速度快，且表示精度可系统地提升。此外，得益于 GMP 特征与物理学之间的深层联系，我们此前已证明，基于这些特征的机器学习模型具有可迁移性[1]。凭借这些特点，GMP 特征化有望服务于广泛的研究者，助力未来的化学与材料科学研究。因此，我们需要轻量、可靠且开放的软件，能够快速、准确地计算这些特征。

3 概述

GMP-Featurizer软件包主要使用 C++ 和 Python 编写。C++ 用于底层计算模块，以提高运行速度；Python 则提供直观易读的 API，供相关领域的科学家和研究人员使用。尽管该软件包并不显式依赖 ASE [11] 和 pymatgen [12]，（它们是化学与材料科学领域中最常用的两个 Python 库），但软件包提供了相应的 API，使其能够轻松读取由这些库定义的原子系统。此外，该软件包使用 Ray 并行执行特征计算，并行化效率接近 100%。总体而言，该软件包旨在实现轻量、易用、快速且准确。

工作流的主要输入包括一个 Python 字典，其中包含定义所需 GMP 特征集所需的超参数；以及一个待提取特征的原子系统列表。定义原子系统的原生方式很简单，即使用一个 Python 字典，其中包含诸如“晶格矢量”“原子位置”“原子类型”等信息。如前所述，该软件包还支持使用预定义的转换器处理 ASE Atoms 和 pymatgen Structure 对象。通过自定义转换器，还可以扩展支持其他格式。它还支持对无序原子结构进行特征化，而许多常用的特征化方法并不支持这一功能。详情请参阅第 4 节。输出则是一个字典列表，其中包含原子结构的特征结果，以及在需要时计算出的导数。

默认情况下，该软件包会在每个原子位置计算 GMP特征；但用户也可以为每个原子结构提供感兴趣位置的列表，从而在原子系统内任意一组参考点处计算特征。用户还可以指定用于并行计算的核心数。此外，计算结果可以缓存在本地，以便重新处理数据集，例如在进行增强或修改之后。软件包还提供两套标准赝势文件，这些文件是指定 GMP 特征集所必需的，但从商业或开源的密度泛函理论软件中收集可能并不容易。最后，代码库中提供了一系列教程，帮助用户快速上手并了解代码库的各项功能。

4 示例

这里提供一个快速示例，展示此软件包的接口。有关更详细的教程，请参阅代码库中包含的示例。

首先，必须导入特征化器模块，它是 GMP 特征计算程序的主要接口。还必须准备好原子系统列表，并在需要时配备适当的转换器。

```
1 from GMPFeaturizer import GMPFeaturizer , ASEAtomsConverter , PymatgenStructureConverter 2 import
```

接下来，必须定义一个用于指定 GMP 特征集的超参数字典，以便据此初始化特征化器。

```
1 GMP = { 2 "GMP" : { 3                    [ 1、 0、 1、 2]、"orders" :
```

这里，阶数列表对应 MCSH 角向探针函数的阶数，sigma 列表对应高斯径向探针的 sigma 值（详情参见文献[1] ）。生成的特征集由阶数列表和 sigma 列表的笛卡尔积定义。例外的是阶数 -1，它对应局域电子密度，因此 sigma 值无关紧要。所以，阶数为 -1 时只有一个特征。

因此，在上述设置下，特征集如下所示：

[(−1, 0), (0, 0.1), (0, 0.2), (0, 0.3), (1, 0.1), (1, 0.2), (1, 0.3), (2, 0.1), (2, 0.2), (2, 0.3)]

这里，第一个数字仍然表示 MCSH 角向探针的阶数，第二个数字表示高斯径向探针的 sigma。请注意，将 calc_derivatives=设置为 True 后，我们就是在要求特征化器同时计算相对于原子位置的 GMP特征导数。默认情况下，用户无需指定局部截断距离。该距离将根据 overlap_threshold 设置推断得出。

此时，只需通过以下方式，即可计算并访问所需的原子系统 GMP特征集：

```
1 结果 = 特征化器.prepare_features(systems, cores=5, converter=converter) 2 3 gmp_
```

这里，cores 指定了用于并行计算的核心数。

因此，计算给定原子系统列表的 GMP 特征的完整脚本如下所示：

```
来自GMPFeaturizer 导入GMPFeaturizer, ASEAtomsConverter, PymatgenStructureConverter导入pickle


# 假设示例数据是一个 ASE 原子对象列表
使用 打开("./example.p", "rb") as f:
    系统 = pickle.load(f)
转换器 = ASEAtomsConverter()

GMP = {
    "GMP": {
        "顺序": [-1, 0, 1, 2],
        “西格玛值”: [0.1, 0.2, 0.3]
    },
    “赝势_路径”: “<路径>/NC-SR.gpsp”,
    “重叠_阈值”: 1e-16,
}
特征化器 = GMPFeaturizer(GMPs=GMPs, 计算_导数=True)
结果 = 特征化器.准备_特征(系统, 核心=5, 转换器=转换器)

gmp_特征 = [条目 [“特征”] 针对条目 在 结果]
gmp_特征_导数 = [条目 [“特征_素数”] 针对条目 在 结果]
```

致谢

本研究得到丰田研究院能源与材料部的支持。作者感谢 Jens Hummelshøj 就单元测试框架提供了有益的讨论。

参考文献

[1] Xiangyun Lei 和 Andrew J. Medford。《原子体系特征化的通用框架》。《物理化学快报》，13(34):7911–7919，2022。PMID: 35980312。[2] AmpTorch。https://github.com/ulissigroup/amptorch，2020。[3] Ray：简化 Python 机器学习工作负载的生产化与扩展。https://www.ray.io/。访问日期：2023-02-23。[4] Yunxing Zuo、Mingde Qin、Chi Chen、Weike Ye、Xiangguo Li、Jian Luo 和 Shyue Ping Ong。利用贝叶斯优化和图深度学习加速材料发现。《今日材料》，51:126–135，2021年12月。[5] C. Collins、M. S. Dyer、M. J. Pitcher、G. F. S. Whitehead、M. Zanella、P. Mandal、J. B. Claridge、G. R. Darling 和 M. J. Rosseinsky。在复杂无机相区中加速发现两种晶体结构类型。《自然》，546(7657):280–284，2017年6月。期号：7657。出版商：Nature Publishing Group。[6] Jörg Behler 和 Michele Parrinello。高维势能面的广义神经网络表示。《物理评论快报》，98:146401，2007年4月。[7] Albert P Bartók、Sandip De、Carl Poelking、Noam Bernstein、James R Kermode、Gábor Csányi 和 Michele Ceriotti。机器学习统一了材料和分子的建模。《科学进展》，3(12):e1701816–e1701816，2017。

[8] V. Zaverkin 和 J. Kästner。高斯矩：受物理启发的分子描述符，可用于构建准确且可扩展的机器学习势。《Journal of Chemical Theory and Computation》，16(8):5410–5421，2020。PMID: 32672968。[9] Chi Chen、Weike Ye、Yunxing Zuo、Chen Zheng 和 Shyue Ping Ong。图网络：分子和晶体的通用机器学习框架。《Chemistry of Materials》，31(9):3564–3572，2019。[10] Tian Xie 和 Jeffrey C Grossman。用于准确且可解释地预测材料性质的晶体图卷积神经网络。《Physical Review Letters》，120(14):145301–145301，2018。[11] Ask Hjorth Larsen、Jens Jørgen Mortensen、Jakob Blomqvist、Ivano E Castelli、Rune Christensen、Marcin Dułak、Jesper Friis、Michael N Groves、Bjørk Hammer、Cory Hargus、Eric D Hermes、Paul C Jennings、Peter Bjerre Jensen、James Kermode、John R Kitchin、Esben Leonhard Kolsbjerg、Joseph Kubal、Kristen Kaasbjerg、Steen Lysgaard、Jón Bergmann Maronsson、Tristan Maxson、Thomas Olsen、Lars Pastewka、Andrew Peterson、Carsten Rostgaard、Jakob Schiøtz、Ole Schütt、Mikkel Strange、Kristian S Thygesen、Tejs Vegge、Lasse Vilhelmsen、Michael Walter、Zhenhua Zeng 和 Karsten W Jacobsen。原子模拟环境：用于处理原子的 Python 库。《Journal of Physics: Condensed Matter》，29(27):273002，2017。[12] Shyue Ping Ong、William Davidson Richards、Anubhav Jain、Geoffroy Hautier、Michael Kocher、Shreyas Cholia、Dan Gunter、Vincent L. Chevrier、Kristin A. Persson 和 Gerbrand Ceder。Python 材料基因组学（pymatgen）：用于材料分析的稳健开源 Python 库。《Computational Materials Science》，68:314–319，2013。