

# 贝叶斯全局优化材料定向设计软件 (Bgolearn)

## 用户手册

## 关于本手册

本手册全面介绍贝叶斯全局优化材料定向设计软件的使用方法，让用户能够快速理解该软件的运算过程和操作逻辑，了解其中各个模块的使用功能，以使得用户能够在短时间内熟悉并利软件使用这一软件展开科研工作。

本手册主要包括：

- 第一章 贝叶斯全局优化材料定向设计软件（Bgolearn）概述。本章主要介绍 Bgolearn 的项目背景和功能介绍。
- 第二章 软件基本原理。本章从软件的组成模块入手，主要介绍了软件的使用原理、构建思想和软件的总体构架设计。
- 第三章 第三方库使用。本章介绍了如何将软件作为第三方库在基于 Python 的环境下通过 pip 进行包安装、更新和软件主要模块使用。

目录

关于本手册 .....2

第一章 贝叶斯全局优化材料定向设计软件（Bgolearn）概述 .....4

第二章 软件基本原理 .....5

    2.1 回归任务 .....6

    2.2 分类任务 .....6

    2.3 测评任务 .....7

    2.4 Bgolearn 总体架构设计 .....7

第三章 第三方库使用 .....7

    3.1 安装 .....7

    3.2 更新 .....8

    3.3 模块调用 .....8

## 第一章 贝叶斯全局优化材料定向设计软件（Bgolearn）概述

Bgolearn 用于材料成分定向设计以及性能定向优化过程。使用已有实验数据（样本）及其测试性能，Bgolearn 能够在给定的成分空间中搜索最优的材料成分设计，以将目标性能最大/最小化。推荐的成分通过实验合成后，又能成为新的数据并入数据集中，Bgolearn 将利用更多的数据信息对下一次设计做出更加可靠的推荐。通过迭代这个过程，Bgolearn 可以在给定的成分空间中高效地寻找到具有优秀性能的新材料。其中所有的实验过程也可以通过模拟过程代替，如下图所示：

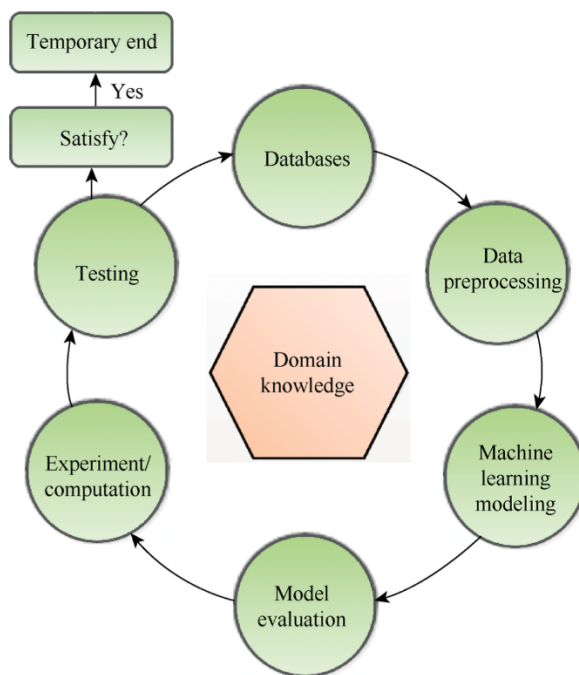


图 1-1 Bgolearn 寻找最优材料设计

Bgolearn 基于 Python 开发，可直接通过 pip 安装，其主要包括回归任务设计和分类任务设计两大功能模块。此外，本软件提供充足的案例参考，能够有效说明每个函数的使用范例，让用户能更方便快速地进行科研使用。

本软件可以根据已有的实验材料数据，高效地推荐可能具有优越性能的材料组成成分、加工工艺，从而给予实验有效的指导，加速材料的研发过程。本软件通过定义统一端口的克里金模型，对不同编程语言（python,R

等)的几种广泛使用的克里金函数的算法模块进行调用,极大地增加了软件的灵活性和应用场景。

## 第二章 软件基本原理

软件文件目录主要分为 Bgolearn (执行文件), Example (使用范例), Intro (软件介绍), Refs (参考资料)和 Template (模板文件)。本软件提供了公开在线阅读的文档地址为: [bgolearn.netlify.app](http://bgolearn.netlify.app)

### 1) 主程序文件 BGOsampling.py

此文件为用户提供调用端口,包含一个待实例化的类函数(.fit 函数),通过类函数(.fit 函数)传入模型参数。根据用户的任务请求,选择返回全局最大优化还是全局最小优化模块的执行端口。

若用户请求返回全局最大搜索端口:通过实例化 BGOmax.py 模块中的类函数实现。若用户请求返回全局最小搜索端口:通过实例化 BGOmin.py 模块中的类函数实现。其中,fit 函数是主程序的入口,会调用不同的类函数给出推荐结果,所包含的主要参数声明如下:

(.fit 函数)参数:参数 Mission:用户选择程序所执行的任务,可传入“Regression”或者“Classification”,默认为“Regression”。当传入的变量为“Regression”时,程序进行回归任务;当传入的变量为“Classification”时,程序进行分类任务。

(.fit 函数)参数:参数 Classifier:当参数 Mission 传入的值为“Classification”时,该参数被激活且值默认为“GaussianProcess”,Bgolearn 中内置了以下 5 种分类器:

- GaussianProcess : Gaussian Process Classifier (默认分类器)
- LogisticRegression: Logistic Regression
- NaiveBayes : Naive Bayes Classifier
- SVM : Support Vector Machine Classifier
- RandomForest : Random Forest Classifier

### 2) 执行程序文件 BGOmin.py 和 BGOmax.py

执行回归任务的具体模块：通过 `BGOmin.py` 和 `BGOmax.py` 两个文件调用九种贝叶斯采样函数的实现端口。

### 3) 执行程序文件 `BGO_clf.py`

执行分类任务的具体模块：提供了五个分类器进行分类任务，默认分类器为高斯过程分类器模型，通过 `BGOsampling.py` 中类函数 `(.fit 函数)` 的 `Classifier` 选择合适的分类器模型。

### 4) 执行程序文件 `BGO_eval.py`

采集函数的效率受多种因素的影响，例如优化函数的类型、训练数据的分布、Kriging 模型的适应度和数据噪声、计算预算等。该模块通过实例化类 `(.test 函数)` 实现。提供了四种方法来比较不同采集函数之间的效率。

## 2.1 回归任务

回归任务设计集成了以下九种回归效用函数的实现端口：

- Expected Improvement algorithm (期望提升函数)
- Expected improvement with “plugin” (有“plugin”的期望提升函数)
- Augmented Expected Improvement (增广期望提升函数)
- Expected Quantile Improvement (期望分位提升函数)
- Reinterpolation Expected Improvement (重插值期望提升函数)
- Upper confidence bound (高斯上确界函数)
- Probability of Improvement (概率提升函数)
- Predictive Entropy Search (预测熵搜索函数)
- Knowledge Gradient (知识梯度函数)

## 2.2 分类任务

分类任务集成了以下三种分类效用函数：

- Least Confidence (欠信度函数)
- Margin Sampling (边界函数)
- Entropy-based approach (熵索函数)

## 2.3 测评任务

测评任务集成了以下四种效用函数的测评函数：

- Trail path method
- Opportunity Cost method
- Probability density function method
- Count Strategy

## 2.4 Bgolearn 总体架构设计

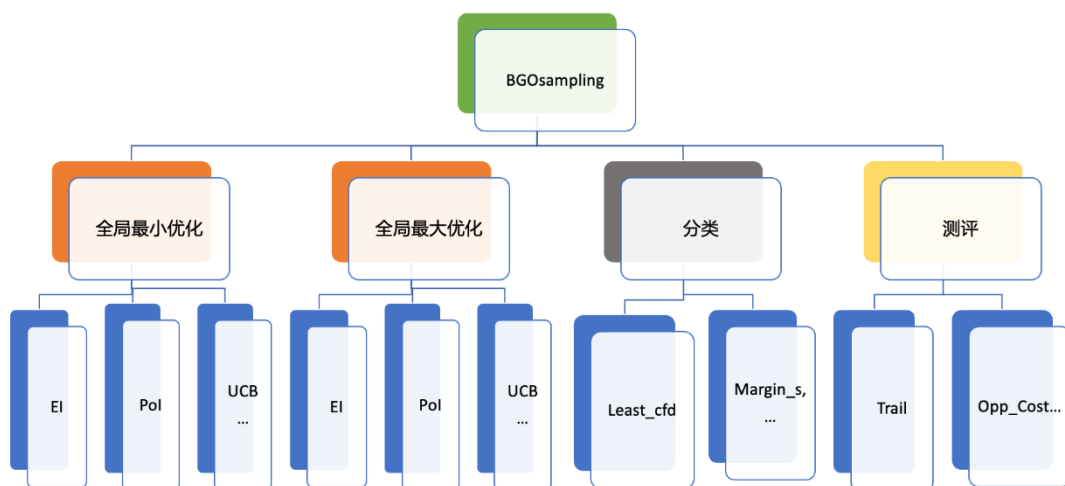


图 2-1 Bgolearn 总体框架设计

## 第三章 第三方库使用

### 3.1 安装

Bgolearn 支持作为 Python 环境下的第三方库通过 pip 进行安装使用，使用如下命令进行安装：

- `pip install Bgolearn - install package.`

## 3.2 更新

使用如下命令可对 Bgolearn 进行更新：

- `pip install --upgrade Bgolearn - update project.`

## 3.3 模块调用

- `Bgolearn().fit`
- `Bgolearn().test`