软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(6):2576−2603 [doi: 10.13328/j.cnki.jos.007304] [CSTR: 32375.14.jos.007304]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

高维贝叶斯优化研究综述*

陈泉霖[1], 陈奕宇[1], 霍 静[1], 曹宏业[1], 高 阳[1], 李 栋[2], 郝建业[2]

[1](计算机软件新技术国家重点实验室 (南京大学), 江苏 南京 210023)

[2](华为技术有限公司 诺亚方舟实验室, 广东 深圳 518129)

通信作者: 高阳, E-mail: gaoy@nju.edu.cn

摘 要: 贝叶斯优化是一种优化黑盒函数的技术, 高效的样本利用率使其在众多科学和工程领域中得到了广泛应用, 如深度模型调参、化合物设计、药物开发和材料设计等. 然而, 当输入空间维度较高时, 贝叶斯优化的性能会显著下降. 为了克服这一限制, 许多研究对贝叶斯优化方法进行了高维扩展. 为了深入剖析高维贝叶斯优化的研究方法, 根据不同工作的假设与特征将高维贝叶斯优化方法分为 3 类: 基于有效低维度假设的方法、基于加性假设的方法以及基于局部搜索的方法, 并对这些方法进行阐述和分析. 首先着重分析这 3 类方法的研究进展, 然后比较各类方法在贝叶斯优化应用中的优劣势, 最后总结当前阶段高维贝叶斯优化的主要研究趋势, 并对未来发展方向展开讨论.

关键词: 高维贝叶斯优化; 贝叶斯优化; 黑盒优化; 降维; 变量选择

中图法分类号: TP311

中文引用格式: 陈泉霖, 陈奕宇, 霍静, 曹宏业, 高阳, 李栋, 郝建业. 高维贝叶斯优化研究综述. 软件学报, 2025, 36(6): 2576–2603. http://www.jos.org.cn/1000-9825/7304.htm

英文引用格式: Chen QL, Chen YY, Huo J, Cao HY, Gao Y, Li D, Hao JY. Survey on High-dimensional Bayesian Optimization. Ruan Jian Xue Bao/Journal of Software, 2025, 36(6): 2576–2603 (in Chinese). http://www.jos.org.cn/1000-9825/7304.htm

Survey on High-dimensional Bayesian Optimization

CHEN Quan-Lin[1], CHEN Yi-Yu[1], HUO Jing[1], CAO Hong-Ye[1], GAO Yang[1], LI Dong[2], HAO Jian-Ye[2]

[1](State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023, China)

[2](Noah's Ark Laboratory, Huawei Technologies Co. Ltd., Shenzhen 518129, China)

Abstract: Bayesian optimization is a technique for optimizing black-box functions. Due to its high sample utilization efficiency, it is widely applied across various scientific and engineering fields, such as hyperparameters tuning of deep models, compound design, drug development, and material design. However, the performance of Bayesian optimization significantly deteriorates when the input space is of high dimensionality. To overcome this limitation, numerous studies carry out high-dimensional extensions on Bayesian optimization methods. To deeply analyze research methods of high-dimensional Bayesian optimization, this study categorizes these methods into three types based on assumptions and characteristics of different kinds of work: methods based on the effective low-dimensional hypothesis, methods based on additive assumptions, and methods based on local search. Then, this study elaborates on and analyzes these methods. This study first focuses on analyzing the research progress of these three types of methods. Then, the advantages and disadvantages of each method in the application of Bayesian optimization are compared. Finally, the main research trends in high-dimensional Bayesian optimization at the current stage are summarized, and future development directions are discussed.

Key words: high-dimensional Bayesian optimization; Bayesian optimization; black-box optimization; dimensionality reduction; variable selection

* 基金项目: 国家自然科学基金 (62192783, 62276128); 科技创新 2030—"新一代人工智能"重大项目 (2021ZD0113303); 中央高校基础研究基金 (14380128)
陈泉霖和陈奕宇为共同第一作者.
收稿时间: 2023-08-09; 修改时间: 2024-04-08, 2024-06-09, 2024-08-11; 采用时间: 2024-10-12; jos 在线出版时间: 2025-03-05
CNKI 网络首发时间: 2025-03-06

1 引 言

许多科学和工程问题可以抽象为黑盒优化问题, 而这些问题的目标函数评估成本通常是高昂的. 贝叶斯优化 (Bayesian optimization, BO) 是一种能够高效求解此类问题的方法, 已在许多领域得到广泛应用, 例如模型选择[1]、自动机器学习[2−4]、A/B 测试[5]、药物设计[6]、神经网络结构搜索 (neural architecture search, NAS)[7,8]、分子设计[9]、拓扑设计[10]、广告攻击[11]、机器人技术[12−14]、自然语言处理[15]、强化学习[16,17]等.

尽管贝叶斯优化在许多领域取得了成功, 但其应用通常局限于低维问题. 应用维度界限通常是 20 维[18], 当输入空间维度升高时, 贝叶斯优化的性能会显著下降[19]. 然而, 许多实际问题的输入空间是高维的, 例如混合整数规划算法 (mixed integer programming, MIP)[20]、深度学习模型[21]以及基因设计[22]的参数均可达上百个. 为了进一步扩展贝叶斯优化的应用范围, 许多工作致力于解决贝叶斯优化在高维空间中面临的挑战. 为了克服维度诅咒 (curse of dimensionality), 通常需要引入额外的假设: 一些方法引入了有效低维度假设[23]; 另一些方法引入了加性假设[19]. 还有一些方法并不引入额外假设, 而是基于局部搜索来避免过度探索[24].

贝叶斯优化作为一种成熟的黑盒优化技术, 已有许多文献对其进行回顾和分析[18,25], 然而, 高维贝叶斯优化作为贝叶斯优化的一个分支, 仍缺乏充分的回顾和分析, 通常仅作为贝叶斯优化综述的一个章节[25]. 近年来, 高维贝叶斯优化领域取得了一些重要进展, 有些工作提出了新方法, 有些则对现有方法进行了扩展、改进和深入分析. 此外, 神经网络的发展也推动高维贝叶斯优化应用于更广泛的场景.

基于上述认识, 本文聚焦于高维贝叶斯优化研究, 以各种工作的假设和特征为主要出发点, 将它们分为 3 类: 基于有效低维度的方法、基于加性假设的方法、基于局部搜索的方法, 并对这 3 类方法进行剖析与解读. 第 2 节介绍贝叶斯优化的基本概念和核心原理. 第 3 节分析贝叶斯优化在高维问题中面临的挑战. 第 4 节详细介绍基于有效低维度假设的方法, 分类总结随机降维、变量选择、基于学习的降维及 VAE 降维与贝叶斯优化结合的特点, 并分析其优劣势. 第 5 节详细介绍基于加性假设的方法, 总结各类分组方法与贝叶斯优化结合的特点, 并分析其优劣势. 第 6 节阐述基于局部搜索的方法并分析其优劣势. 第 7 节概述高维贝叶斯优化的应用领域. 第 8 节介绍高维贝叶斯优化的测试场景和代码库. 第 9 节分析并展望高维贝叶斯优化的未来方向. 最后总结全文.

2 贝叶斯优化

与所有优化问题类似, 贝叶斯优化的目标是找到目标函数 f 的最优解, 即:

$$\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}),$$

其中, $\mathcal{X}$ 是可行集. 与其他优化问题不同的是, 贝叶斯优化面向的优化问题具有一些特殊的限制, 具体而言, 可行集 $\mathcal{X}$ 和目标函数 f 通常具有以下性质[18].

(1) f 的具体形式和结构未知, 即 f 是一个黑盒 (black box), 且一般认为函数是多峰的.

(2) f 的评估开销很高, 可能是评估次数有限 (几十次或者几百次), 或者评估时间很久 (几小时).

(3) $\mathcal{X}$ 一般是一个简单的集合, 如凸集, 使得判断点 $\mathbf{x}$ 是否包含于 $\mathcal{X}$ 中是比较容易的.

贝叶斯优化由两个组件组成, 分别是代理模型 (surrogate model) 和采集函数 (acquisition function). 代理模型用于对目标函数进行建模, 它可以给出未知点所在函数的预测值和方差. 采集函数则通过权衡预测值和方差来度量未知点的采样价值. 最后, 采集函数的最优解即为下一个采样点. 其中, 代理模型的主要作用是提高算法的采样效率. 算法伪代码见算法 1.

算法 1. 贝叶斯优化过程.

输入: 优化函数 f, 初始采样点个数 n_0, 总采样次数 N;
输出: 函数 f 的最优解.

1. 确定 GPR 的先验信息, 选择采集函数 $\alpha(\mathbf{x};\mathcal{D}_n)$.
2. 初始时采集 n_0 个样本, 即 $\mathcal{D}_{n_0}=\{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_{n_0},y_{n_0})\}$, 其中 $y_i=f(\mathbf{x}_i)$.
3. $n\leftarrow n_0$.
4. while $n\leqslant N$ do
5. 使用数据集 $\mathcal{D}_n=\{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n)\}$, 构建 GPR 模型.
6. 通过最大化采集函数 $\alpha(\mathbf{x};\mathcal{D}_n)$ 来选取下一个点 $\mathbf{x}_{n+1}$, 即:

$$\mathbf{x}_{n+1}=\arg\max_{\mathbf{x}\in\mathcal{X}}\alpha(\mathbf{x};\mathcal{D}_n).$$

7. 评估 $\mathbf{x}_{n+1}$ 的函数值 $y_{n+1}\leftarrow f(\mathbf{x}_{n+1})$.
8. 更新数据集 $\mathcal{D}_{n+1}\leftarrow\mathcal{D}_n\cup\{(\mathbf{x}_{n+1},y_{n+1})\}$.
9. end while
10. 返回 $\mathcal{D}_N$ 中函数值最大的点

常用的代理模型包括高斯过程回归 (Gaussian process regression, GPR)[26]、随机森林[3]、核密度估计 (kernel density estimation)[27,28]、深度模型等[29−31]. 由于 GPR 高效的样本利用率和对不确定性良好的建模能力, 大多数贝叶斯优化算法使用 GPR 作为代理模型.

在后面章节中, 我们将在第 2.1 节对 GPR 进行简要介绍, 在第 2.2 节对采集函数进行简要介绍.

2.1 高斯过程回归

高斯过程回归是一种用于建模目标函数的回归模型. 它可视为线性回归或岭回归的扩展模型. 通常, 线性回归或岭回归分别表示为 $f(\mathbf{x})=\mathbf{x}^{\top}\mathbf{w}$, 或者 $f(\mathbf{x})=\phi(\mathbf{x})^{\top}\mathbf{w}$, 其中, 参数 $\mathbf{w}$ 是一个确定的向量. GPR 则扩展这一概念, 视参数 $\mathbf{w}$ 为随机变量 (如 $\mathbf{w}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})$), 使得 $f(\mathbf{x})$ 也成为随机变量, 增强了模型的灵活性和表达能力. 另一种更为普遍的表述形式是直接将函数值 $f(\mathbf{x})$ 视为随机变量, 并在函数空间中推理其形式. 接下来, 我们将介绍高斯过程的定义及其在回归问题中的应用.

定义 1. 见文献 [32] 的定义 3.1. 若对于任意固定的 $\mathbf{x}$, 其函数值 $f(\mathbf{x})$ 是一个随机变量, 则称 f 是一个随机函数. 若进一步满足 $\forall n,\forall\mathbf{x}_1,\ldots,\mathbf{x}_n$, n 元随机变量

$$\mathbf{f}=(f(\mathbf{x}_1),\ldots,f(\mathbf{x}_n))^{\top}$$

服从 n 元高斯分布, 则称 $f(\mathbf{x})$ 为一个高斯过程.

其次, GPR 基于两个基本假设: (1) 目标函数 f 在可行域上是一个高斯过程; (2) n 元高斯变量的均值和协方差分别由均值函数 $m(\mathbf{x})$ 和协方差函数 $k(\mathbf{x},\mathbf{x}')$ 来决定, 即:

$$\begin{cases}\mathbb{E}[f(\mathbf{x})]:=m(\mathbf{x})\\ \mathbb{E}[(f(\mathbf{x})-m(\mathbf{x}))(f(\mathbf{x}')-m(\mathbf{x}'))]:=k(\mathbf{x},\mathbf{x}')\end{cases},$$

其中, 均值函数 $m(\mathbf{x})$ 和协方差函数 $k(\mathbf{x},\mathbf{x}')$ 通常根据先验知识由研究者手动指定.

另外, 考虑到在现实场景中观测值一般带有噪音, 因此观测值 y 建模为 $y=f(\mathbf{x})+\epsilon$, 其中 ϵ 是服从高斯分布的噪音, 即 $\epsilon\sim\mathcal{N}(0,\sigma^2)$. 而建模噪音还能避免 GPR 过拟合. 类似地, 观测值的均值和协方差为:

$$\begin{cases}\mathbb{E}[y]=\mathbb{E}[f(\mathbf{x})]+\mathbb{E}[\epsilon]=m(\mathbf{x})\\ cov(y_p,y_q)=k(\mathbf{x}_p,\mathbf{x}_q)+\sigma^2\delta_{pq}\end{cases},$$

其中, δ_{pq} 是 Kronecker 记号, 当 $p=q$ 时, $\delta_{pq}=1$; 否则 $\delta_{pq}=0$.

基于上述假设, 给定训练集 $\mathcal{D}_n=\{\mathbf{x}_{1:n},\mathbf{y}_{1:n}\}$ 和测试点 $\mathbf{x}_\star$, 有 $n+1$ 元高斯分布:

$$\begin{bmatrix}\mathbf{y}_{1:n}\\ f(\mathbf{x}_\star)\end{bmatrix}\sim\mathcal{N}\left(\begin{matrix}\mathbf{K}(\mathbf{x}_{1:n},\mathbf{x}_{1:n})+\sigma_n^2\mathbf{I} & \mathbf{k}(\mathbf{x}_{1:n},\mathbf{x}_\star)\\ \mathbf{k}(\mathbf{x}_\star,\mathbf{x}_{1:n}) & k(\mathbf{x}_\star,\mathbf{x}_\star)\end{matrix}\right).$$

由 Sherman-Morrison-Woodbury 定理[26]得, $f(\mathbf{x}_\star)$ 的后验分布为:

$$f(\mathbf{x}_\star)|\mathcal{D}_n,\mathbf{x}_\star \sim \mathcal{N}\left(\mu_n(\mathbf{x}),\sigma_n^2(\mathbf{x})\right),$$

其中,

$$\mu_n(\mathbf{x}_\star) := m(\mathbf{x}_\star)+\mathbf{k}(\mathbf{x}_\star,\mathbf{x}_{1:n})\left(\mathbf{K}(\mathbf{x}_{1:n},\mathbf{x}_{1:n})+\sigma_n^2\mathbf{I}\right)^{-1}(\mathbf{y}_{1:n}-\mathbf{m}(\mathbf{x}_{1:n})) \tag{1}$$

$$\sigma_n^2(\mathbf{x}_\star) := k(\mathbf{x}_\star,\mathbf{x}_\star)-\mathbf{k}(\mathbf{x}_\star,\mathbf{x}_{1:n})\left(\mathbf{K}(\mathbf{x}_{1:n},\mathbf{x}_{1:n})+\sigma_n^2\mathbf{I}\right)^{-1}\mathbf{k}(\mathbf{x}_{1:n},\mathbf{x}_\star) \tag{2}$$

综上所述, GPR 可根据训练集 $\mathcal{D}_n$ 来预测 $\mathbf{x}_\star$ 上的目标函数值, 其中 $\mu_n(\mathbf{x}_\star)$ 表示预测值均值, 而 $\sigma_n^2(\mathbf{x}_\star)$ 表示预测误差.

GPR 目前难以扩展到大数据场景, 这是因为其时间复杂度为 $O(N^3)$, 空间复杂度为 $O(N^2)$. 因此, 有一些工作致力于提高 GPR 的扩展性, 如稀疏高斯过程 (sparse GP)[33−35]的时间复杂度为 $O(MN^2)$, 且 $M \ll N$. 高斯-马尔可夫过程 (Gauss-Markov processes)[36]的时间复杂度为 $O(N\log N)$, 空间复杂度为 $O(N)$.

GPR 的优势是能够充分利用先验信息提高样本利用率, 但若指定了错误的先验信息, 则其性能将大打折扣. 第 2.1.1 节我们简要介绍如何选取先验信息: 协方差函数 k 与均值函数 m.

2.1.1 均值函数和协方差函数

建模数据之间的相关性尤为重要, 一个基本的假设是, 距离相近的两点可能有相似的函数值. 而 GPR 使用协方差函数来建模数据的相关性, 其中协方差函数满足以下性质:

$$\|\mathbf{x}-\mathbf{x}'\| \leqslant \|\mathbf{x}-\mathbf{x}''\| \Rightarrow k(\mathbf{x},\mathbf{x}') \geqslant k(\mathbf{x},\mathbf{x}'').$$

此外, 协方差函数必须是半正定函数, 以保证协方差矩阵为半正定矩阵. 下面, 我们分别介绍 3 种协方差函数: 平稳协方差函数、非平稳协方差函数, 以及高维协方差函数.

● 平稳协方差函数. 平稳协方差函数是指协方差函数 $k(\mathbf{x},\mathbf{x}')$ 只由 $\mathbf{x}-\mathbf{x}'$ 决定, 即 $k(\mathbf{x},\mathbf{x}')$ 可以简写为 $k(\mathbf{x}-\mathbf{x}')$. 最常用的平稳协方差函数是高斯核函数, 其形式为:

$$k(r) := \alpha_0 \exp\left(-\frac{r^2}{2\ell^2}\right),$$

其中, $r=\|\mathbf{x}-\mathbf{x}'\|_2$. 从高斯核函数的形式可以看出, 当两个点的距离 r 增加时, $k(r)$ 呈指数级下降, 即它们的协方差 (或者相关性) 呈指数级下降. 此外, 超参数 ℓ 越大, $k(r)$ 随 r 增大下降得越缓慢. 故当 GPR 预测某个点 $\mathbf{x}_\star$ 的目标函数值 $\hat{f}_\star$ 时, 训练集中的点离 $\mathbf{x}_\star$ 越近对预测值 $\hat{f}_\star$ 的影响越大, 并且影响范围与超参数 ℓ 有关.

另一个常用的协方差函数是马氏核函数[26]:

$$k_{\text{Matern}}(r) := \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}r}{\ell}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}r}{\ell}\right),$$

其中, ν, ℓ 是非负的超参数, K_ν 是修改的贝塞尔函数. 马氏核函数可以看作是高斯核函数的推广, 这是因为当 $\nu\to\infty$ 时, 马氏核函数退化为高斯核函数. 故马氏核函数可以拟合种类更丰富的代理模型.

● 非平稳协方差函数. 平稳协方差函数的一个局限是, 它使得代理模型的函数变化率在所有区域都是相同的. 实际上, 平稳协方差函数可以转化为非平稳协方差函数, 最常用的方法是作输入变换, 即:

$$k_{\text{warp}}(\mathbf{x},\mathbf{x}') := k_s(w(\mathbf{x}),w(\mathbf{x}')),$$

其中, 输入变换 w 为单调函数 (如贝塔分布的累积分布函数[37]), k_s 为平稳协方差函数.

● 高维协方差函数. 在高维空间中, 优化算法很难用有限的样本来充分探索整个空间. 因此, BOCK[38]设计圆柱形的高维协方差函数, 旨在鼓励优化算法更多地探索可行集的中心位置而不是边界 (BOCK 假设最优解在中心附近的概率更大).

首先, 将输入 $\mathbf{x}$ 从球内映射到圆柱中, 即:

$$T(\mathbf{x}) := \begin{cases} (\|\mathbf{x}\|_2, \mathbf{x}/\|\mathbf{x}\|_2), & \text{if } \|\mathbf{x}\|_2 > 0 \\ (0, \mathbf{a}_{\text{arbitrary}}), & \text{if } \|\mathbf{x}\|_2 = 0 \end{cases},$$

$$T^{-1}(r,\mathbf{a}) := r\mathbf{a},$$

其中, $\mathbf{a}_{\text{arbitrary}}$ 是单位球中任意的向量. 该输入变换可以扩展中心附近的区域, 收缩靠近边界的区域. 圆柱协方差函数的形式为:

$$k_{\text{cyl}}(\mathbf{x}_1,\mathbf{x}_2) := k_{\text{warp}}(r_1,r_2)\cdot k_{\text{poly}}(\mathbf{a}_1,\mathbf{a}_2),$$

其中, $k_{\text{warp}}(r_1,r_2)$ 衡量数据点的半径相似性. 因为非平稳核函数 k_{warp} 的扭曲函数 w 是单调非减的, 即输入越接近 0, k_{warp} 越大, 所以 $k_{\text{warp}}(r_1,r_2)$ 会鼓励算法探索原点附近的区域. $k_{\text{poly}}(\mathbf{a}_1,\mathbf{a}_2)$ 衡量数据点的角度相似性, 故 GPR 的预测方差主要取决于数据点的角度.

均值函数通常定义为常量, 即 $m(\mathbf{x})\equiv\mu$. 上述协方差函数的超参数以及均值函数的常量都属于 GPR 模型的超参数. 这些超参数通常可以基于训练数据确定, 方法包括最大似然估计 (maximum likelihood estimate, MLE)[18]、最大后验估计 (maximum a posteriori, MAP)[18]以及完全贝叶斯方法[25]. 然而, 基于数据的方法也往往使得优化方法陷入局部最优解[39], 故也有工作采用不基于学习的超参数调整方法[40].

2.2 采集函数

给定 GPR 的后验分布, 尽管可直接选择均值 $\mu_n(\mathbf{x})$ 最大的点作为候选解, 但这容易使算法陷入局部最优解. 为了权衡算法的局部搜索和全局搜索, 在采样时同时考虑均值和方差. 因此, 人们设计了不同的采集函数 $\alpha(\mathbf{x})$, 以度量点 $\mathbf{x}$ 的采样价值. 由此, 确定下一轮采样位置成为一个优化问题, 即:

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha(\mathbf{x}).$$

下面介绍 3 种最常用的采集函数, 分别为期望提升 (expected improvement, *EI*)[41]、概率提升 (probability improvement, *PI*)[42]和置信度上界 (upper confidence bound, *UCB*)[43].

EI 的思想是考虑采样 $\mathbf{x}$ 能带来多大的提升. 提升值定义为:

$$I(\mathbf{x}) := \max\{f(\mathbf{x})-y_{\max},0\},$$

其中, $y_{\max}=\max\mathbf{y}_{1:n}$ 表示当前训练集中最大的观测值. I 总是非负的, 这是因为若 $f(\mathbf{x})<y_{\max}$, 则数据集中最大的观测值仍为 $y_{\max}$. 值得注意的是, 因为 $f(\mathbf{x})$ 是随机变量, 所以提升值 $I(\mathbf{x})$ 也是随机变量, 故需要求其期望值, 即:

$$\alpha_{EI}(\mathbf{x}) := \mathbb{E}[I(\mathbf{x})] = (\mu_n(\mathbf{x})-y_{\max})\left[1-\Phi\left(\frac{y_{\max}-\mu_n(\mathbf{x})}{\sigma(\mathbf{x})}\right)\right]+\sigma(\mathbf{x})\phi\left(\frac{y_{\max}-\mu_n(\mathbf{x})}{\sigma(\mathbf{x})}\right),$$

其中, Φ 是标准正态分布, ϕ 是标准正态分布的密度函数.

PI 的思想与 *EI* 类似, 但它考虑的是 $f(\mathbf{x})>y_{\max}$ 的概率, 即:

$$\alpha_{PI}(\mathbf{x}) := \Pr(f(\mathbf{x})>y_{\max}) = 1-\Phi\left(\frac{y_{\max}-\mu_n(\mathbf{x})}{\sigma(\mathbf{x})}\right).$$

UCB 的思想是直接将误差 $\sigma_n(\mathbf{x})$ 加入均值 $\mu_n(\mathbf{x})$ 中, 从而增加采样的探索性, 避免搜索算法陷入局部最优解, 即:

$$\alpha_{UCB}(\mathbf{x}) := \mu_n(\mathbf{x})+\beta_{n+1}^{1/2}\sigma_n(\mathbf{x}),$$

其中, β_{n+1} 是一个常数, 用于控制置信水平, 若 β_{n+1} 越大, 则探索性越强.

3 高维挑战

贝叶斯优化的输入维度通常不超过 20 维; 当维度过高时, 算法的性能将急剧下降. 该问题主要是由两个因素导致[19]: 样本距离增大导致代理模型精度下降以及高维采集函数难以优化.

3.1 高维代理模型精度下降

高维空间中的点之间的距离存在下界, 即使样本容量非常大, 也不可能用有限的样本点密集地填充可行域, 使得代理模型仅在非常局部的空间有效. 事实上, 假设训练集 $\mathbf{X}=\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ 和测试点 $\mathbf{x}_*$ 独立同分布 (均匀分布) 地采样自 $[0,1]^D$, 则测试点 $\mathbf{x}_*$ 到训练集 $\mathbf{X}$ 的最近距离为:

$$d_\infty(\mathbf{x}_*,\mathbf{X}) := \mathbb{E}[\min_{1\leqslant i\leqslant n}\|\mathbf{x}_*-\mathbf{x}_i\|_\infty].$$

该距离存在下界 (证明见文献 [44]), 即:

$$d_\infty(\mathbf{x}_*,\mathbf{X}) \geqslant \frac{D}{2(D+1)}\cdot\frac{1}{n^{1/D}}.$$

如表 1 所示, 当维度为 10 以上时, 即使样本数量很大, 也很难让该下界逼近 0. 此外, 由公式 (2) 可得, 当测试点 $\mathbf{x}_*$ 远离训练集 $\mathbf{X}$ 时, 方差 $\sigma_n(\mathbf{x}_*)$ 增大, 故在高维下很难得到与在低维同样有效的代理模型.

表 1 $d_\infty(\mathbf{x}^\star,\mathbf{X})$ 的下界

D	$n=100$	$n=1000$	$n=10000$	$n=100000$
1	$d_\infty \geqslant 0.0025$	$d_\infty \geqslant 0.00025$	$d_\infty \geqslant 0.000025$	$d_\infty \geqslant 0.0000025$
10	$d_\infty \geqslant 0.28$	$d_\infty \geqslant 0.22$	$d_\infty \geqslant 0.18$	$d_\infty \geqslant 0.14$
20	$d_\infty \geqslant 0.37$	$d_\infty \geqslant 0.34$	$d_\infty \geqslant 0.30$	$d_\infty \geqslant 0.26$

3.2 高维采集函数难以优化

在最大化采集函数时, 计算开销随维度增加呈指数级增长. 最大化采集函数最常用的两类方法是全局优化和多起点的局部搜索.

全局优化以矩形切割法 (dividing rectangles, DIRECT)[45]为例, DIRECT 将搜索空间划分为多个超矩形, 利用 Lipschitz 连续性选择最有潜力的区域, 并继续将其划分为更小的超矩形, 重复此过程. DIRECT 对采集函数的查询次数为 $O(\zeta^{-D})$, 其中 ζ 为误差[46]. 可以看到 DIRECT 的计算开销随维度增加而指数增长, 故它在维度不超过 10 时才表现出较好性能.

多起点的局部搜索以基于梯度的局部搜索为例, 该方法在给定起点后, 每个起始点根据梯度迭代若干步直到收敛到局部最优解, 最终在多个局部最优解中选择最优者. 然而, 许多实际高维采集函数在大部分区域较为平坦, 只有少数区域存在尖锐的峰[47]. 只有当起点靠近这些尖峰时, 局部搜索才能收敛至峰顶, 否则局部最优解只能停留在平坦区域. 在高维空间中, 即使有大量起点, 也很难保证它们落在尖峰附近.

针对高维采集函数, EGP[47]提出了一种指定起点的方案, 以改善多起点局部搜索的性能. 具体而言, 当核函数的超参数 ℓ 较小时, 采集函数比较平坦; 反之, 采集函数会有明显的梯度. 另外, 采集函数在不同超参数 ℓ 下的最优解都是相近的. 因此, 设采集函数为 $\alpha(\mathbf{x}|\ell_\tau)$, 则 EGP 首先最大化一个 ℓ 足够大的采集函数, 即:

$$\mathbf{x}_\ell^\star = \max_{\mathbf{x}\in\mathcal{X}}\alpha(\mathbf{x}|\ell),$$

其中, $\ell > \ell_\tau$, 然后将 $\mathbf{x}_\ell^\star$ 作为起点来优化采集函数 $\alpha(\mathbf{x}|\ell_\tau)$.

3.3 分 类

为了克服维度诅咒, 通常需要引入额外的假设, 根据方法的假设及其特点, 我们将高维贝叶斯优化的工作分为以下 3 类.

(1) 基于有效低维度假设的方法. 这类方法假设输入空间中存在一个有效的低维子空间, 该子空间足以支持预测目标函数值. 故这类方法设法寻找该低维子空间, 然后在子空间上进行贝叶斯优化, 从而克服维度诅咒. 然而, 有效维度是未知的, 如何确定子空间的维度仍是开放的问题, 该限制了这类方法的应用范围.

(2) 基于加性假设的方法. 这类方法假设高维函数可分解为若干个低维函数之和. 所以, 这类方法设法将高维变量分为若干组, 每组变量对应一个低维目标函数, 然后用贝叶斯优化来求解低维的目标函数, 从而克服维度诅咒. 但是, 该方法的假设也限制了其应用范围. 此外, 因为加性结构是未知的, 如何确定加性结构仍是开放问题.

(3) 基于局部搜索的方法. 这类方法不需要额外的假设. 因为充分探索高维空间是不可能的, 所以这类方法认为应该着重探索更有潜力的局部区域. 相比于前两种方法, 这类方法的应用维度较低.

我们在表 2 中对上述高维贝叶斯优化方法的优劣势进行对比与总结.

表 2 高维贝叶斯优化方法的优劣势对比

方法	优势	劣势
基于有效低维度假设的方法	能有效地优化具有低维子空间的高维函数	(1) 应用范围限制于具有低维子空间的任务 (2) 难以确定子空间的维度
基于加性假设的方法	能有效地优化具有加性结构的高维函数	(1) 应用范围限制于具有加性结构的任务 (2) 难以确定真实的加性结构
基于局部搜索的方法	不需要额外的假设, 能广泛应用于各种高维函数	应用范围限制于较低维度的任务

4 基于有效低维度假设的高维贝叶斯优化

这类方法假设输入空间 $\mathcal{X}\subset\mathbb{R}^D$ 中存在一个有效的低维子空间, 该子空间足以预测数据的目标函数值. 基于这一假设, 我们可以在低维子空间中执行贝叶斯优化, 从而规避维度诅咒. 具体而言, 基于有效低维度假设的方法可概括为 3 个关键步骤.

(1) 建立输入空间 $\mathcal{X}$ 到低维子空间 $\mathcal{Z}$ 的映射, 将样本 $\mathbf{x}_1,\dots,\mathbf{x}_n$ 映射为隐变量 $\mathbf{z}_1,\dots,\mathbf{z}_n$;

(2) 利用数据集 $\{(\mathbf{z}_i,y_i)\}_{i=1}^n$ 在低维子空间 $\mathcal{Z}$ 中执行贝叶斯优化, 确定下一轮采样点 $\mathbf{z}^*$;

(3) 构建从低维子空间 $\mathcal{Z}$ 到输入空间 $\mathcal{X}$ 的逆映射, 将隐变量 $\mathbf{z}^*$ 映射为 $\mathbf{x}^*$, 并采样 $f(\mathbf{x}^*)$.

这一过程的示意图如图 1 所示. 其中, h 为从输入空间 $\mathcal{X}$ 到隐空间 $\mathcal{Z}$ 的变换函数, f 为隐空间 $\mathcal{Z}$ 到输出空间 $\mathcal{Y}$ 的回归模型, g 为从隐空间 $\mathcal{Z}$ 到输入空间 $\mathcal{X}$ 的重构函数.

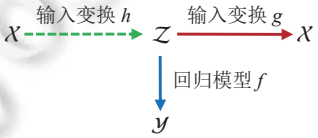


图 1 基于有效低维度方法的示意图

根据映射方法的不同, 基于有效低维度假设的方法可进一步细分为多个子类: 基于随机降维的方法、基于变量选择的方法、基于学习的方法以及基于变分自编码器 (variational auto-encoder, VAE) 降维的方法等.

4.1 基于随机降维的方法

这类方法基于一个核心假设: 有效的低维子空间是一个线性空间. 然而, 这种假设也带来了局限性, 即这类方法只能识别线性流形[48]. 为了更深入地理解这一概念, 我们首先介绍有效维度的定义.

定义 2. 有效维度. 若存在线性子空间 T ($\dim T=d_e$), 使得对于所有 $\mathbf{x}\in\mathbb{R}^D$ 都有 $f(\mathbf{x})=f(\mathbf{x}_T)$, 则称函数 $f:\mathbb{R}^D\to\mathbb{R}$ 具有有效维度 d_e ($d_e<D$). 其中 $\mathbf{x}_T$ 是 $\mathbf{x}$ 在 T 上的投影. 子空间 T 则称为有效低维子空间.

具体而言, 任一线性空间 T 可将输入空间 $\mathbb{R}^D$ 分解为 T 与其正交补空间 $T^\perp$ 的直和, 即 $\mathbb{R}^D=T\oplus T^\perp$. 于是任意向量 $\mathbf{x}\in\mathbb{R}^D$ 可唯一地分解为 $\mathbf{x}=\mathbf{x}_T+\mathbf{x}_\perp$, 其中 $\mathbf{x}_T\in T$, $\mathbf{x}_\perp\in T^\perp$. 而有效子空间意味着函数值只与 T 上的投影有关, 即 $f(\mathbf{x}_T+\mathbf{x}_\perp)=f(\mathbf{x}_T)$.

然而, 对于许多函数, 可能无法找到一个线性子空间 T 严格满足 $f(\mathbf{x})=f(\mathbf{x}_T)$. 为此, 有工作提出了一个更为灵活的概念: ϵ-有效维度. 这一定义允许 $f(\mathbf{x})$ 与 $f(\mathbf{x}_T)$ 存在一定误差, 具体如下.

定义 3. ϵ-有效维度[49]. 对于任意 $\epsilon>0$, 若存在线性子空间 $V_\epsilon\subset\mathbb{R}^D$, 使得对于所有 $\mathbf{x}\in\mathbb{R}^D$ 都有 $|f(\mathbf{x})-f(\mathbf{x}_\epsilon)|\leqslant\epsilon$, 则称函数 $f:\mathbb{R}^D\to\mathbb{R}$ 具有 ϵ-有效维度. 其中 $\mathbf{x}_\epsilon$ 是 $\mathbf{x}$ 在子空间 V_ϵ 的投影.

满足 ϵ-有效维度的子空间可能不唯一. 我们通常希望选择维度尽可能小的子空间, 故最优的 ϵ-有效维度定义为:

$$d_\epsilon:=\min_{V_\epsilon}\dim(V_\epsilon).$$

若能容忍的误差 ϵ 较小, 则 d_ϵ 往往较大; 反之, 若 ϵ 较大, 则 d_ϵ 较小. 这一观察表明, 尽管某些函数可能不存在严格意义上的有效维度, 但只要选取足够大的 ϵ, 总能找到一个满足 ϵ-有效维度的线性子空间 V_ϵ.

4.1.1 随机降维

在寻找上述线性子空间的过程中, 随机降维技术通常采用两种主要形式: 随机嵌入和 sketching. 这两种方法各有特点, 为高维贝叶斯优化问题提供了不同的解决思路.

随机嵌入: REMBO (random embeddings Bayesian optimization)[23]表明通过随机矩阵即可找到上述的子空间, 而不需要进行学习, 具体如下.

定理 1. 见 REMBO 的定理 2. 给定一个有效维度为 d_e 的函数 $f:\mathbb{R}^D\to\mathbb{R}$ 和一个随机矩阵 $\mathbf{A}\in\mathbb{R}^{D\times d}$ ($\mathbf{A}$ 中的元素独立地采样于标准正态分布且 $d>d_e$), 则对于任意 $\mathbf{x}\in\mathbb{R}^D$, 以 1 的概率存在一个 $\mathbf{z}\in\mathbb{R}^d$, 使得 $f(\mathbf{x})=f(\mathbf{Az})$.

这个定理的重要性在于, 它保证了对于最优值点 $\mathbf{x}^*\in\mathbb{R}^D$, 必然存在 $\mathbf{z}^*\in\mathbb{R}^d$ 使得 $f(\mathbf{x}^*)=f(\mathbf{Az}^*)$. 这使得我们可以直接优化一个低维的函数 $g(\mathbf{z}):=f(\mathbf{Az})$. 此外, 随机矩阵还具有 Johnson-Lindenstrauss transform 性质 [50], 保证了数据点之间的距离在高维空间上和在低维空间上是近似相等的. 对于 ϵ- 有效维度假设, 也有类似定理成立, 即定理 2.

定理 2. 见文献 [49] 的引理 1. 给定一个函数 $f:\mathbb{R}^D\to\mathbb{R}$ 和一个随机矩阵 $\mathbf{A}\in\mathbb{R}^{D\times d}$ $(d>d_e)$, 则以 1 的概率, 对于任意 $\mathbf{x}\in\mathbb{R}^D$, 都存在 $\mathbf{z}\in\mathbb{R}^d$, 使得 $|f(\mathbf{x})-f(\mathbf{Az})|\leqslant 2\epsilon$. 其中 $\mathbf{A}$ 中的元素独立地采样于标准正态分布.

在嵌入空间上的优化策略主要有两种: 一是像 REMBO 一样直接搜索最优解, 二是不断地缩小残差 (sequential random embeddings, SRE[49]). SRE 的核心思想是将优化目标定义为:

$$g(\mathbf{z}):=f(\mathbf{x}_i+\mathbf{A}^{(i)}\mathbf{z}),$$

其中, $\mathbf{A}^{(i)}$ 是当前步骤产生的随机矩阵. 若

$$\mathbf{z}_i=\arg\max_{\mathbf{z}} g(\mathbf{z}),$$

则当前解更新为 $\mathbf{x}_{i+1}:=\mathbf{x}_i+\mathbf{A}^{(i)}\mathbf{z}_i$. 此时残差则为 $\|\mathbf{x}^*-\mathbf{x}_{i+1}\|$, 下一轮的优化目标则为:

$$g(\mathbf{z}):=f(\mathbf{x}_{i+1}+\mathbf{A}^{(i+1)}\mathbf{z}).$$

这种方法的有效性可以通过性质 1 得到支持.

性质 1. 见 SRE 的性质 1. 令 $S_i=\{\mathbf{A}^{(i)}\mathbf{z}|\mathbf{z}\in\mathbb{R}^d\}$ 表示随机矩阵 $\mathbf{A}^{(i)}$ 定义的子空间, 记 $\mathbf{x}^*-\mathbf{x}_i$ 在 S_i 上的投影为 $\hat{\mathbf{x}}_i$. 若满足:

$$\frac{\|\hat{\mathbf{x}}_i-\mathbf{A}^{(i)}\mathbf{z}_i\|}{\|\hat{\mathbf{x}}_i\|}\leqslant\frac{1}{5}\frac{\|\hat{\mathbf{x}}_i\|}{\|\mathbf{x}^*-\mathbf{x}_i\|},$$

则 $\|\mathbf{x}^*-\mathbf{x}_i\|>\|\mathbf{x}^*-\mathbf{x}_{i+1}\|$.

然而, 值得注意的是, 虽然性质 1 保证了残差会不断减少, 但序列化地减少残差并不一定会比直接优化函数 $g(\mathbf{z}):=f(\mathbf{Az})$ 更有效.

● Sketching 实现降维. 除了随机嵌入外, sketching 技术也是实现降维的有效方法, 特别是它可以避免“边界问题 (hashing-enhanced subspace BO, HeSBO[51])”, 具体如下.

Sketching 技术的作用是在一个数据流中找到出现次数最多的元素. 在这里, 下标 $1,\ldots,D$ 视为数据流, 对应的数值 $x_1,\ldots,x_D$ 视为相应数据的频率. 现在考虑这样一个问题: 仅用 d 个存储单位, 找到数据流 $1,\ldots,D$ 中出现次数最多的数据. 为了实现这一点, 首先引入两个均匀哈希函数:

$$\text{uniform hash function } h:[D]\to[d],$$
$$\text{uniform hash function } \sigma:[D]\to\{-1,1\},$$

其中, $[D]:=\{1,\ldots,D\}$. 函数 h 将集合 $[D]$ 映射到集合 $[d]$ 中, 并且函数值在 $[d]$ 上服从均匀分布. 又因为 h 是函数, 故满足 $i=j\Rightarrow h(i)=h(j)$. 类似地, 函数 σ 将 $[D]$ 映射到集合 $\{-1,1\}$ 中. 现有 d 个计数器, 记为 $\mathbf{z}_{1:d}$. 若遇到哈希值等于 j 的数据, 则第 j 个计数器记录该数据, 即:

$$z_j+=\sigma(i)\cdot x_i,\ \text{if } h(i)=j.$$

故对于任意计数器有:

$$z_j = \sum_{i:h(i)=j} \sigma(i)\cdot x_i,\ \forall j \in [d].$$

注意到 $\mathbb{E}[\sigma(i)y_{h(i)}] = x_i$, 所以第 i 个元素频率的估计值为 $\sigma(i)y_{h(i)}$, 其方差为:

$$Var[\sigma(i)z_{h(i)}] = \frac{1}{d}\sum_{j=1,j\neq i}^{D} x_j^2.$$

这种用少量存储单元来统计大量数据的技术称为 sketching.

基于上述讨论, HeSBO 的优化过程可概括为两个步骤: (1) 使用 BO 优化低维变量 $\mathbf{z}_{1:d}$, 并得到最优解 $\mathbf{z}^*$; (2) 利用 sketching 技术将最优解 $\mathbf{z}^*$ 恢复为高维数据 $\mathbf{x}^*$. 此外, 若可行集 $\mathcal{X} = [-1,1]^D$, 则低维约束集为 $\mathcal{Z} = [-1,1]^d$, 故 HeSBO 不会产生“边界问题”.

然而, HeSBO 包含最优解的概率较低 (见文献 [52]), 其概率为:

$$p_H(Z^*;D,d,d_e) := \frac{d!}{(d-d_e)!d^{d_e}}.$$

为提高包含最优解的概率, BAXUS (BO with adaptively expanding subspaces)[53]使用稀疏矩阵 $\mathbf{S} \in \{0,\pm 1\}^{d\times D}$ 作为投影矩阵, 其中每列有且仅有一个非零元素, 每行有 D/d 个非零元素. 这时成功概率提升为:

$$p_B(Z^*;D,d,d_e) := \frac{\sum_{i=0}^{d_e}\begin{pmatrix} d(1+\beta_{\text{small}})-D \\ i \end{pmatrix}\begin{pmatrix} D-d\beta_{\text{small}} \\ d_e-i \end{pmatrix}\beta_{\text{small}}^i\beta_{\text{large}}^i}{\begin{pmatrix} D \\ d_e \end{pmatrix}},$$

其中, $\beta_{\text{small}} = \lfloor D/d \rfloor$, $\beta_{\text{large}} = \lceil D/d \rceil$. 此外, $\lim_{D\to\infty} p_B = p_H$, 故随着 D 逐渐增加, p_B 会越来越接近 p_H.

4.1.2 边界问题

另一个亟待解决的关键问题是: 如何选择低维空间的约束集 $\mathcal{Z} \subset \mathbb{R}^d$. 注意到可行集 $\mathcal{X} \subset \mathbb{R}^D$ 通常远小于 $\mathbb{R}^D$. 尽管定理 1 表明低维空间 $\mathbb{R}^d$ 必定包含最优解 $\mathbf{z}^*$, 但若约束集 $\mathcal{Z}$ 太大, 优化难度将显著增加; 反之, 若约束集太小, 则可能无法包含最优解 $\mathbf{z}^*$.

事实上, 若约束集 $\mathcal{Z}$ 过大, 采集点 $\mathbf{x}_0 := \mathbf{A}\mathbf{z}_0$ 可能会落在约束集 $\mathcal{X}$ 外. 此时, 需要将 $\mathbf{x}_0$ 投影到凸集 $\mathcal{X}$ 上, 即:

$$P_{\mathcal{X}}(\mathbf{x}_0) := \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmin}}\|\mathbf{x}-\mathbf{x}_0\|.$$

因此, 实际上的重构函数为 $\phi : \mathbf{z} \mapsto P_{\mathcal{X}}(\mathbf{A}\mathbf{z})$, 低维空间所能覆盖的高维区域至多为 $\mathcal{E} := \phi(\mathbb{R}^d)$. 在能覆盖 $\mathcal{E}$ 的情况下, 过大的约束集 $\mathcal{Z}$ 只会导致大量冗余点映射到边界 $\partial\mathcal{X}$ 上, 从而加大了优化难度. 这种现象称为“边界问题”, 如图 2 所示. 其中, 填充的水平线是定义在 $\mathcal{X}$ 上的函数, 仅依赖于第 2 个变量 x_2, x_2^* 是函数的最优解. $\mathcal{Y}$ 是低维空间的约束集, $Ran(\mathbf{A})$ 是矩阵 $\mathbf{A}$ 的列空间, $\mathbf{A}\mathcal{Y}$ 是低维约束集在高维空间的值域. $\mathbf{A}\mathcal{Y}_1$ 表示低维约束集选得过大, 而 $\mathbf{A}\mathcal{Y}_2$ 表示低维约束集选得过小. $P_A(\mathcal{X})$ 是可行集 $\mathcal{X}$ 在 $Ran(\mathbf{A})$ 上的投影. $\mathcal{E}$ 是低维空间在映射 ϕ 下的值域.

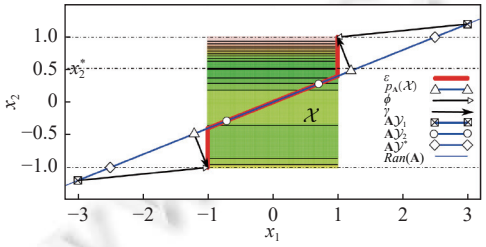


图 2 边界问题的示意图[54]

边界问题对核函数的影响可从以下几个方面观察: (1) 在边界 $\partial\mathcal{X}$ 上任取一点 $\mathbf{x}_1$, 其原像为 $\{\mathbf{z} : \phi(\mathbf{z}) = \mathbf{x}_1\}$. 尽管这些点在高维空间中都对应同一个点, 但低维核函数 $k_{\mathcal{Z}}(\mathbf{z},\mathbf{z}')$ 将它们视为不同的点, 这导致贝叶斯优化会浪费采样次数去采样这些冗余点. (2) 虽然用高维点来建模核函数更为准确, 但高维核函数 $k_{\mathcal{X}}(\phi(\mathbf{z}),\phi(\mathbf{z}'))$ 会受到维度诅咒.

为了在使用高维距离来建模核函数的同时保持空间是低维的, REMBO-ϕ[55]将边界 $\partial\mathcal{X}$ 上的点再次映射到 $Ran(\mathbf{A})$, 即 $\psi:\mathbf{z}\mapsto P_{\mathbf{A}}(\phi(\mathbf{z}))$. 这使得核函数 $k_{\mathcal{X}}(\psi(\mathbf{z}),\psi(\mathbf{z}'))$ 能够在使用高维距离的同时避免受到维度诅咒, 因为 $Ran(\mathbf{A})$ 是低维子空间. 然而, $k_{\mathcal{X}}(\psi(\mathbf{z}),\psi(\mathbf{z}'))$ 并不能有效地阻止采集函数去探索冗余区域.

基于有效维数来选择约束集: 定理 3 阐明, 仅基于有效为数 d_e 来选择约束集, 可以使约束集包含最优解的概率保持为一个常数.

定理 3. 见 REMBO 的定理 3. 给定一个有效维度为 d_e 的函数 $f:\mathbb{R}^D\to\mathbb{R}$ 和一个随机矩阵 $\mathbf{A}\in\mathbb{R}^{D\times d}$ $(d>d_e)$, 则以至少 $1-\epsilon$ 的概率存在一个 $\mathbf{z}^\star\in\mathbb{R}^d$, 使得:

$$f(\mathbf{x}_T^\star)=f(\mathbf{A}\mathbf{z}^\star)\wedge\ \|\mathbf{z}^\star\|_2\leqslant(\sqrt{d_e}/\epsilon)\|\mathbf{x}_T^\star\|_2.$$

例如, 若原问题的可行集为 $\mathcal{X}=[-1,1]^D$, 则根据定理 3, 至少 $1-\epsilon$ 的概率有:

$$\|\mathbf{z}^\star\|_2\leqslant(\sqrt{d_e}/\epsilon)\|\mathbf{x}_T^\star\|_2\leqslant(\sqrt{d_e}/\epsilon)\sqrt{d_e}.$$

因此, 所选择的 $\mathcal{Z}$ 应确保包含球 $(0,d_e/\epsilon)$. 然而, 这种方法存在一些局限性.

(1) 若要达到很高的成功率, 则 $\mathcal{Z}$ 需要变得相当大. 即便如此, 也无法确保成功概率为 1.

(2) 该定理给出的上界并不是一个紧的上界.

总之, 这种方法并不能有效地解决边界问题. 尽管它提供了一个理论基础, 但难以应用于实际问题.

求解最小的约束集以解决边界问题: 为解决边界问题, 理想的方法是寻找最小的约束集 $\mathcal{Z}$, 同时保证其能覆盖 $\mathcal{E}$, 即:

$$\begin{cases}\inf\limits_{\mathcal{Z}\subset\mathbb{R}^d} Vol(\mathcal{Z})\\ \text{s.t. }\phi(\mathcal{Z})=\mathcal{E}\end{cases}.$$

然而, 该问题的最优解 $\mathcal{U}$ 是一个星形集 (非凸集), 且求解过程计算成本高昂 (见文献 [54] 定理 1). 故 REMBO-γ[54]提出了一种替代方法, 不直接考虑映射 ϕ, 而是引入一个性质更优的映射 γ. 其核心思想包括以下几个步骤.

(1) 建立 $P_{\mathbf{A}}(\mathcal{E})\subset Ran(\mathbf{A})$ 与 $\mathcal{E}$ 之间的双射, 即 $P_{\mathcal{X}}:P_{\mathbf{A}}(\mathcal{E})\to\mathcal{E}$, $P_{\mathbf{A}}:\mathcal{E}\to P_{\mathbf{A}}(\mathcal{E})$.

(2) 构造 $\mathbf{B}P_{\mathbf{A}}(\mathcal{E})\subset\mathbb{R}^d$ 与 $P_{\mathbf{A}}(\mathcal{E})$ 之间的双射, 即 $\mathbf{B}^\top:\mathbf{B}P_{\mathbf{A}}(\mathcal{E})\to P_{\mathbf{A}}(\mathcal{E})$, $\mathbf{B}:P_{\mathbf{A}}(\mathcal{E})\to\mathbf{B}P_{\mathbf{A}}(\mathcal{E})$.

(3) 定义 $\mathbf{B}P_{\mathbf{A}}(\mathcal{E})$ 到 $\mathcal{E}$ 的双射 $\gamma:\mathbf{B}P_{\mathbf{A}}(\mathcal{E})\to\mathcal{E}$.

其中矩阵 $\mathbf{B}$ 的行向量由 $Ran(\mathbf{A})$ 的正交基组成. 综上, γ 的具体形式为:

$$\gamma(\mathbf{z}):=P_{\mathcal{X}\cap P_{\mathbf{A}}^{-1}(\mathbf{B}^\top\mathbf{z})}(\mathbf{B}^\top\mathbf{z}).$$

因为 $\mathcal{Z}^*:=\mathbf{B}P_{\mathbf{A}}(\mathcal{E})$ 和 $\mathcal{E}$ 能建立双射, 所以原优化问题可简化为:

$$\inf_{\mathcal{Z}\subset\mathbb{R}^d,\gamma(\mathcal{Z})=\mathcal{E}} Vol(\mathcal{Z}).$$

该问题的解即为 $\mathcal{Z}^*$, 它是 Zonotope (一种多面体). 相比星形集 $\mathcal{U}$, $\mathcal{Z}^*$ 结构更简单, 便于快速判断点是否在其中.

以带约束的采集函数解决边界问题: ALEBO (re-examining linear embeddings BO)[52]为采集函数添加约束以解决边界问题, 即:

$$\begin{cases}\max\limits_{\mathbf{z}\in\mathbb{R}^d}\alpha(\mathbf{z})\\ \text{s.t. }-\mathbf{1}\leqslant\mathbf{A}\mathbf{z}\leqslant\mathbf{1}\end{cases},$$

其中, $-\mathbf{1}\leqslant\mathbf{A}\mathbf{z}\leqslant\mathbf{1}$ 形成的环绕空间为多面体, 包含于可行集 $\mathcal{X}$ 中, 因此不会产生边界问题; 然而, 因为该环绕空间是可行集的真子集, 所以它包含最优解的概率小于 1.

4.2 基于变量选择的方法

这类方法隐式地假设了有效子空间是轴对齐的, 从而允许我们直接舍弃无关维度, 保留有效的维度, 整体流程如图 3 所示. 变量的选择方式主要包括两种策略: 随机选择变量和基于梯度选择变量. 其中, h 表示从 D 个变量中选择 d 个变量, $\mathcal{Z}$ 表示 d 个变量形成的空间, f 表示空间 $\mathcal{Z}$ 到输出空间 $\mathcal{Y}$ 的回归模型, g 表示填充 $D-d$ 个变量.

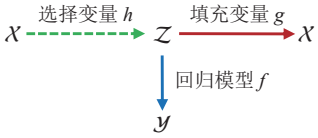


图 3 基于变量选择的示意图

随机选择变量: DropoutUCB[46]每一轮随机选择 d 个维度进行贝叶斯优化, 从而避免了维度诅咒. 在得到一个 d 维的最优解后, 还需要考虑如何填充其他 $D-d$ 个变量. 一种方法是直接使用当前最优的样本作为填充值, 保证填充值质量. 但是该方案很可能会导致算法陷入局部最优解, 为此需要加入一些扰动, 即以一个小概率从约束集中随机采样 $D-d$ 个值作为填充值.

这类算法的性能瓶颈在于填充算法, 这一点可从遗憾界观察到. 具体而言, 假设目标函数是 L-Lipschitz 连续的, 算法的遗憾界为:

$$\sqrt{C_1\beta_T^d\gamma_T T}+2TL(D-d)+2,$$

其中, C_1, β_T, γ_T, L 为常数. 遗憾界的主要部分 $2TL(D-d)$ 代表丢弃 $D-d$ 个变量带来的损失上界.

基于梯度选择变量: VS-BO (variable selection BO)[56]基于以下观察选择变量: 目标函数关于变量 x_i 的偏导 $\partial f/\partial x_i$ 越大, 该变量越重要. 由于无法直接获得黑盒函数的导数, VS-BO 对 GPR 的均值求偏导, 即:

$$\nabla_{\mathbf{x}} f \approx IS := \mathbb{E}_{\mathbf{x}\sim Unif(\mathcal{X})}\left[\frac{\nabla_{\mathbf{x}}\mu(\mathbf{x}|\mathcal{D})}{\sigma(\mathbf{x}|\mathcal{D})}\right].$$

根据重要度 IS 将变量从大到小排列, 选择前 d 个变量进行贝叶斯优化, 从而避免维度诅咒.

VS-BO 还提出一种新的填充算法: 假设样本点采样于一个多元高斯分布 $p(\mathbf{x}|\mathcal{D})$, 当贝叶斯优化得到 $\mathbf{x}^d$ 后, 其余的 $D-d$ 个变量 $\mathbf{x}^{D-d}$ 采样于条件分布 $p(\mathbf{x}^{D-d}|\mathbf{x}^d,\mathcal{D})$, 这类似于演化算法 CMA-ES[57]的思想.

4.3 基于学习的方法

上述方法均不基于机器学习, 本节将介绍基于学习的方法. 与随机降维类似, 这类方法也假设目标函数具有低维结构, 即函数 $f:\mathbb{R}^D\to\mathbb{R}$ 可由一个行满秩矩阵 $\mathbf{A}\in\mathbb{R}^{d\times D}$ 定义:

$$f(\mathbf{x})=g(\mathbf{A}\mathbf{x}) \tag{3}$$

其中, g 属于一类受限的函数.

将投影矩阵作为超参数: ActiveGP[58]将矩阵 $\mathbf{A}$ 视为 GPR 的超参数, 通过优化超参数求得矩阵 $\mathbf{A}$. 具体而言, GPR 旨在拟合函数 g, 假设使用高斯核函数:

$$k(\mathbf{u},\mathbf{u}'):=\gamma^2\exp\left[-\frac{1}{2}(\mathbf{u}-\mathbf{u}')^{\top}(\mathbf{u}-\mathbf{u}')\right],$$

则它对应高维核函数为:

$$k_{\mathcal{X}}(\mathbf{x},\mathbf{x}'):=\gamma^2\exp\left[(\mathbf{x}-\mathbf{x}')^{\top}\mathbf{A}^{\top}\mathbf{A}(\mathbf{x}-\mathbf{x}')\right].$$

可见矩阵 $\mathbf{A}$ 是核函数 $k_{\mathcal{X}}(\mathbf{x},\mathbf{x}')$ 的超参数. 假设 $\mathbf{A}$ 的先验分布服从多元高斯分布, 结合 GPR 的似然分布 $p(\mathbf{y}|\mathbf{A},\mathbf{X})$, 利用 Laplace 逼近可求得矩阵 $\mathbf{A}$ 的后验分布 $p(\mathbf{A}|\mathbf{y},\mathbf{X})$. 采样该后验分布可得 $\mathbf{A}$ 的估计值 $\hat{\mathbf{A}}$ 最后在函数 $g(\hat{\mathbf{A}}\mathbf{x})$ 上进行贝叶斯优化, 从而避免维度诅咒.

利用低秩矩阵恢复算法来求解 $\mathbf{A}$: SI-BO (subspace identification BO)[59]将求解 $\mathbf{A}$ 转化为低秩矩阵恢复问题 (low-rank matrix recovery). 具体而言, 由链式法则和公式 (3) 可得, $\nabla f(\mathbf{x})=\mathbf{A}^{\top}\nabla g(\mathbf{A}\mathbf{x})$. 函数 f 的泰勒展开为:

$$f(\mathbf{x}+\epsilon\phi)=f(\mathbf{x})+\epsilon\langle\phi,\nabla f(\mathbf{x})\rangle+\epsilon E(\mathbf{x},\epsilon,\varphi),$$

其中, $\epsilon E(\mathbf{x},\epsilon,\phi)$ 为泰勒余项. 将 $\nabla f(\mathbf{x})$ 替换为 $\mathbf{A}^{\top}\nabla g(\mathbf{A}\mathbf{x})$ 可得:

$$\langle\phi,\mathbf{A}^{\top}\nabla g(\mathbf{A}\mathbf{x})\rangle=(f(\mathbf{x}+\epsilon\phi)-f(\mathbf{x}))/\epsilon-E(\mathbf{x},\epsilon,\phi).$$

当有 m 个样本 $\{(\mathbf{x}_i,f_i)\}_{i=1}^m$ 时, 上式可向量化为:

$$\mathbf{y}=\Phi(\mathbf{X})+E(\mathbf{X},\epsilon,\Phi),y_i=(1/\epsilon)\sum_{j=1}^{m}[f(\mathbf{x}_j+\epsilon\phi_{ij})-f(\mathbf{x}_j)] \tag{4}$$

其中, $\mathbf{X}:=\mathbf{A}^{\top}\mathbf{G}$, $\mathbf{G}:=[\nabla g(\mathbf{A}\mathbf{x}_1),\ldots,\nabla g(\mathbf{A}\mathbf{x}_m)]$. 公式 (4) 是一个低秩矩阵恢复问题, 若其解为 $\hat{\mathbf{X}}$, 则 $\hat{\mathbf{A}}=SVD(\hat{\mathbf{X}})$ 即为所求. 最后在函数 $g(\hat{\mathbf{A}}\mathbf{x})$ 上进行贝叶斯优化, 从而避免维度诅咒.

KSIR-BO[60]使用逆向回归方法 (sliced inverse regression, SIR)[61]来求得投影矩阵 $\mathbf{A}$, 然后在函数 $g(\mathbf{A}\mathbf{x})$ 上进行优化, 从而避免维度诅咒.

MGPC-BO[62]则使用神经网络寻找低维子空间, 因为神经网络的逆映射是未知的, 所以其逆映射通过多输出 GPR[63]学习. 这样, 在低维子空间上搜索到的最优解通过多输出 GPR 重构.

4.4 基于 VAE 降维的方法

在优化现实中的对象 (如分子结构、拓扑结构或表达式) 时, 这些对象通常缺乏直接的数值表示. 因此, 需要首先对其进行编码, 然后才能进行优化. VAE-BO 是一类专门用于优化离散结构化对象的贝叶斯优化方法.

优化过程: 以分子设计为例, 其目标是发现具有更优化学属性的新分子, 即:

$$\max_{m\in\mathcal{M}} f(m),$$

其中, $f:\mathcal{M}\to\mathbb{R}$ 是形式未知的目标函数, $\mathcal{M}$ 表示所有分子组成的集合. 这类优化问题面临两个主要挑战: (1) 可行集 $\mathcal{M}$ 是离散的, 这使得我们很难产生一个候选解 $m\in\mathcal{M}$. (2) 目标函数是黑盒的且评估代价高昂, 这是因为评估分子的化学属性需要进行湿实验或者耗时一天的计算机模拟[64].

CVAE (character variational autoencoder)[9]结合 VAE 与 BO 以解决上述两个问题. 对于第 1 个问题, CVAE 将离散问题转换为连续问题. 具体而言, 训练一个 VAE, 包括编码器 $Enc:\mathcal{M}\to\mathbb{R}^D$ 和解码器 $Dec:\mathbb{R}^D\to\mathcal{M}$, 使得 $Dec(Enc(m))\approx m$, $\forall m\in\mathcal{M}$. 这样, 优化问题转换为:

$$\max_{\mathbf{z}\in\mathbb{R}^D} f(Dec(\mathbf{z})).$$

第 2 个问题则通过 BO 来解决, 因为 BO 是一种样本利用率高的黑盒优化技术. 具体而言, (1) 使用训练集 $\{Enc(m_i),y_i)\}_{i=1}^{n}$ 来构建 GPR 模型; (2) 通过最大化采集函数得到候选点 $\mathbf{z}_{n+1}$; (3) 生成新分子 $m_{n+1}=Dec(\mathbf{z}_{n+1})$; (4) 评估新分子的化学属性 $y_{n+1}=f(m_{n+1})$.

编码与解码过程: (1) 基于 SMILES 表示法, 将分子结构编码为字符串 $Enc_S:\mathcal{M}\to\Sigma^*$, 同时 SMILES 字符串也可以解码为相应的分子 $Dec_S:\Sigma^*\to\mathcal{M}$, (2) 字符自动编码器 (character variational autoencoder) 将 SMILES 字符串编码成低维的向量 $Enc_C:\Sigma^*\to\mathbb{R}^D$, 以及将低维向量解码为 SMILES 字符串 $Dec_C:\mathbb{R}^D\to\Sigma^*$. 整体结构如图 4 所示. 其中, (Enc_S,Dec_S) 分别为 SMILES 编码器与解码器, (Enc_C,Dec_C) 分别为 CVAE 的编码器和解码器. 回归模型 $g:\mathcal{Z}\to\mathcal{Y}$ 用于建模目标函数.

图 4 VAE-BO 的示意图

4.4.1 “死区”

尽管 CVAE 在优化结构化对象方面取得了显著进展, 但仍存在一些挑战. 其中最突出的问题是隐空间 $\mathcal{Z}$ 中存在“死区”——这些区域中的点解码后会产生无效的分子结构. 若 BO 的采集函数选择了“死区”中的点, 这些点解码后会产生无效解. 文献 [65] 总结了隐空间中出现“死区”的 3 种情况.

(1) VAE 会在隐空间 $\mathcal{Z}$ 上有一个先验分布 $p(\mathbf{z})$, 那些概率很低的点很可能是无效解.

(2) 隐空间维度较高时, 更容易产生“死区”.

(3) 当训练数据不均匀时, 原数据空间中的欠采样区域映射到隐空间后往往会变成“死区”.

带约束的采集函数: 为解决“死区”问题, 文献 [65] 额外训练一个二分类网络, 以隐空间的变量为输入, 输出隐变量成功解码的概率. 这将隐空间 $\mathcal{Z}$ 上的优化问题转化为一个带约束的优化问题:

$$\max_{\mathbf{z}\in\mathcal{Z}} f(\mathbf{z}),\ \text{s.t. } \Pr(C(\mathbf{z})) \geqslant 1-\delta,$$

其中, $f(\mathbf{z})$ 是目标函数, $\Pr(C(\mathbf{z}))$ 表示点 $\mathbf{z}$ 成功解码的概率 (由二分类网络给出), $1-\delta$ 表示可接受的成功概率下界. 在贝叶斯优化中, 通过带约束的采集函数来实现这种带约束优化:

$$EIC(\mathbf{z}) := \begin{cases} \Pr(C(\mathbf{z}))EI(\mathbf{z}), & \text{if } \exists \mathbf{z}, \Pr(C(\mathbf{z})) \geqslant 1-\delta \\ \Pr(C(\mathbf{z})), & \text{otherwise} \end{cases}.$$

若隐空间 $\mathcal{Z}$ 不存在满足约束的点, 则 EIC 函数的最优解为解码成功率最高的点; 若隐空间 $\mathcal{Z}$ 存在满足约束的点, 则 EIC 同时考虑提升值 $EI(\mathbf{z})$ 和解码成功率的大小.

尽管带约束的采集函数降低了采样“死区”的概率, 但二分类网络的训练仍然依赖于实际解码和测试结果, 这使得该方法难以与现实场景完全解耦.

使用编译文法来约束解码过程: GVAE (grammar VAE)[66]开发一种基于 SMILES 字符串的上下文无关文法 (context-free grammar, CFG), CFG 用于引导 VAE 始终生成有效的 SMILES 字符串. 具体而言, CFG 将 SMILES 字符串映射为解析序列 (或解析树), VAE 则对解析序列进行编码和解码, 其中解码器只生成合乎文法的解析序列. 整个模型的编码器和解码器分解为:

$$\begin{cases} Enc = Enc_S \circ Enc_N \\ Dec = Dec_N \circ Dec_S \end{cases},$$

其中, Enc_S 和 Dec_S 分别将分子编码为 SMILES 字符串以及将 SMILES 字符串解码为分子, Enc_N 和 Dec_N 分别将 SMILES 字符串编码为解析序列以及将解析序列解码为 SMILES 字符串.

类似地, SD-VAE (syntax-directed VAE)[67]开发一种基于 SMILES 字符串的属性文法. 属性文法在 CFG 的基础上增加了语义信息, 使得解码器能够考虑语义约束.

然而, 使用编译文法也带来了新的局限性. 这些方法假设 SMILES 字符串是合乎上下文无关文法的, 但实际上并非如此, 这限制了方法的适用范围.

使用图来约束解码过程: JT-VAE (junction tree, VAE)[68]使用连接树 (junction tree) 来表示分子, 其主要流程分为 3 步: (1) 将分子表示为图, 再将图分解为连接树, 每个树节点为图中的一个最大团. JT-VAE 分别将图和连接树编码为 $[\mathbf{z}_G, \mathbf{z}_T]$; (2) 解码时, $\mathbf{z}_T$ 会生成一棵连接树, 其中树节点解码为分子片段; (3) 最后枚举这些片段有效的组合方式, 从而保证生成有效的分子.

但 JT-VAE 存在一些局限性: (1) 网络结构复杂, 导致训练开销大; (2) 只能表征片段之间的连接, 无法表征原子间的连接.

MHG-VAE (molecular hypergraph grammar VAE)[64]针对 JT-VAE 的不足, 基于简单的 GVAE 开发一种图文法 (molecular hypergraph grammar, MHG) 来编码化学约束. 这使得 VAE 能够表征原子间的连接, 并引导 VAE 始终生成有效分子. 具体而言, (1) MHG-VAE 将分子建模为超图, 其中原子建模为超边, 共价键建模为节点; (2) 图文法的作用类似于编译文法, 将超图映射为解析树 (或解析序列); (3) GVAE 对解析序列进行编码和解码. 整个模型的编码器和解码器可分解为:

$$\begin{cases} Enc = Enc_H \circ Enc_G \circ Enc_N \\ Dec = Dec_N \circ Dec_G \circ Dec_H \end{cases},$$

其中, Enc_H 和 Dec_H 分别将分子编码为超图以及将超图解码为分子, Enc_G 和 Dec_G 基于图文法分别将超图编码为解析序列以及将解析序列解码为超图, Enc_N 和 Dec_N 构成 GVAE, 其中 Dec_N 始终生成有效的解析序列.

与场景解耦的方法: 上述缓解“死区”的方法往往都与场景强耦合, 为了实现更通用的解决方案, 一些研究提出了与场景解耦的方法, 如 COLD (constrained optimization with latent distribution)[69]和 BVAE (Bayesian VAE)[70]. 然

而, 这些方法的局限在于其解码的成功率不高.

COLD 在隐空间上采样与训练集相似的点, 这是因为与训练集相似的数据更可能是有效的数据. 具体而言, VAE 的隐空间可视为一个混合高斯模型 (GMM), 高概率密度区域的点更有可能产生与训练集相似的样本, 所以 COLD 设置一个阈值 η 来构建一个可行集 $\mathcal{Z}=\{\mathbf{z}: p(\mathbf{z})>\eta\}$, 并在该可行集上执行优化.

BVAE 将“死区”问题归因于“分布外 (out of distribution, OoD)”的数据. 假设训练数据服从某个分布 p^*, 若某个数据点 $\mathbf{x}^*$ 也服从分布 p^*, 则称 $\mathbf{x}^*$ 是“分布内 (in-distribution)”, 否则称 $\mathbf{x}^*$ 是 OoD. 类似地, 在 VAE 的隐空间上, 若候选点是 OoD 数据, 这时解码器大概率生成一个错误的结果, 且具有高置信度. 而 BVAE 认为, 使用 MLE 来确定超参数导致生成模型对预测结果过于自信 (尤其对 OoD 数据). 因此, BVAE 使用全贝叶斯法来确定超参数, 以提高模型的鲁棒性, 即:

$$p(\mathbf{x}|\mathcal{D})=\int p(\mathbf{x}|\mathcal{D},\theta)p(\theta|\mathcal{D})\mathrm{d}\theta.$$

4.4.2 利用标签信息来构建隐空间

除了上述的“死区”问题, CVAE 还面临另一个挑战: 作为无监督降维方法, CVAE 未能充分利用样本的标签信息. 以下简要介绍几种利用有标签数据构建更适合优化任务的隐空间的方法.

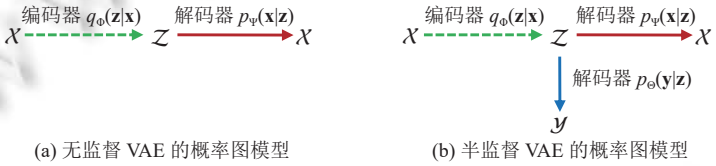
半监督 VAE-BO: VAE-guided-BO[71]提出一种半监督的 VAE, 通过增加一个解码器 $p_\Theta(\mathbf{y}|\mathbf{z})$ 来推断条件分布 $p(\mathbf{x},\mathbf{y}|\mathbf{z})$, 从而更容易构建回归模型 $g:\mathcal{Z}\to\mathcal{Y}$. 具体如下.

无监督的 VAE 图模型如图 5(a) 所示, 其中, 实线表示生成模型, 虚线表示后验分布的变分逼近. 其学习目标是最大化重建 $\mathcal{X}$ 的概率 $p_\Psi(\mathbf{x})$. 因为 $\log p_\Psi(\mathbf{x})$ 有下确界:

$$\log p_\Psi(\mathbf{x})\geqslant\mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}[\log p_\Psi(\mathbf{x}|\mathbf{z})]-D_{\mathrm{KL}}(q_\Phi(\mathbf{z}|\mathbf{x})\,\|\,p(\mathbf{z})).$$

故损失函数为:

$$\mathcal{L}_u:=-\mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}[\log p_\Psi(\mathbf{x}|\mathbf{z})]+D_{\mathrm{KL}}(q_\Phi(\mathbf{z}|\mathbf{x})\,\|\,p(\mathbf{z})).$$



(a) 无监督 VAE 的概率图模型 (b) 半监督 VAE 的概率图模型

图 5 无监督 VAE 和半监督 VAE 的概率图模型

半监督 VAE 图模型如图 5(b) 所示, 新增的实线 $\mathcal{Z}\to\mathcal{Y}$ 表示似然分布. 其学习目标更改为最大化同时重建 $\mathcal{X},\mathcal{Y}$ 的概率 $p(\mathbf{x},\mathbf{y})$. 类似地, 因为 $\log p(\mathbf{x},\mathbf{y})$ 有下确界:

$$\log p(\mathbf{x},\mathbf{y})\geqslant\mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}[\log p_\Psi(\mathbf{x}|\mathbf{z})+\log p_\Theta(\mathbf{y}|\mathbf{z})]-D_{\mathrm{KL}}(q_\Phi(\mathbf{z}|\mathbf{x})\,\|\,p(\mathbf{z})).$$

故有标签数据的损失函数为:

$$\mathcal{L}_\ell:=-\mathbb{E}_{\mathbf{z}\sim q_\Phi(\mathbf{z}|\mathbf{x})}[\log p_\Psi(\mathbf{x}|\mathbf{z})+\log p_\Theta(\mathbf{y}|\mathbf{z})]+D_{\mathrm{KL}}(q_\Phi(\mathbf{z}|\mathbf{x})\,\|\,p(\mathbf{z})).$$

因此, 半监督学习的损失函数为 $\mathcal{L}_u+\mathcal{L}_\ell$.

“加权再训练”[72]: 该方法认为原始 VAE-BO 有两个不足.

(1) 生成模型的学习目标与优化目标不匹配, 生成模型的学习目标是使隐空间上的先验分布尽可能接近原数据的分布, 而优化目标是在隐空间中找到最优值点. 若训练数据的目标函数值不高, 隐空间中可行域通常只包含次优解, 最优解可能不在可行域中.

(2) 生成模型未充分利用新采样点的信息, 而新采样点可能比其他点更靠近最优解.

针对这些不足, 该方法对训练数据进行加权, 目标函数值越高的数据拥有更高的权值, 故损失函数为:

$$\sum\nolimits_{\mathbf{x}_i\in\mathcal{D}}w_i\mathcal{L}(\mathbf{x}_i),$$

其中, 训练数据的权值定义为:

$$w(\mathbf{x};\mathcal{D},k)\propto 1/\left(kN+rank_{f,\mathcal{D}}(\mathbf{x})\right),\quad \text{where } rank_{f,\mathcal{D}}(\mathbf{x})=|\{\mathbf{x}_i:f(\mathbf{x}_i)>f(\mathbf{x}),\mathbf{x}_i\in\mathcal{D}\}|.$$

从而隐空间会向更有潜力的区域扩展. 此外, 每次采样新样本后, VAE 再训练一次, 以充分利用新样本信息.

“可判别”的隐空间: 在分类任务中, 所谓“可判别”隐空间要使得同类样本在隐空间上距离较小, 异类样本在隐空间上距离较大, 从而更适合下游的分类任务. 在优化场景下, 同类样本指函数值相近的样本, 而异类样本指函数值相差较大的样本, 即:

$$\mathcal{D}^+:=\{\mathbf{x}^+\in\mathcal{D}:\ |f(\mathbf{x})-f(\mathbf{x}^+)|<\eta\},\quad \mathcal{D}^-:=\{\mathbf{x}^-\in\mathcal{D}:\ |f(\mathbf{x})-f(\mathbf{x}^-)|\geqslant\eta\}.$$

T-LBO (triplet latent BO)[10]引入深度度量学习, 来构建“可判别”的隐空间, 使其更适合下游优化任务.

4.5 其 他

在直线上执行贝叶斯优化: 针对高维采集函数难以优化的问题, LineBO[73]在定义域中选择一条直线, 在该直线上搜索采集函数的最优解. 具体而言, 仿射子空间 (直线) 定义为:

$$\mathcal{L}(\mathbf{x},\mathbf{d}):=\{\mathbf{x}+\alpha\mathbf{d}:\alpha\in\mathbb{R}\}\cap\mathcal{X},$$

其中, $\mathbf{x}\in\mathcal{X}$ 为偏移量, $\mathbf{d}\in\mathbb{R}^D$ 为方向, 则最大化采集函数表示为:

$$\max_{\mathbf{x}\in\mathcal{L}(\mathbf{x}_t^\star,\mathbf{d}_t)}\alpha(\mathbf{x}|\mathcal{D}_t).$$

通常选择前 t 轮的最优解 $\mathbf{x}_t^\star$ 作为偏移量, 而方向的选择方式有 3 种策略: 随机地选择一个方向; 选择一个与坐标轴对齐的方向; 选择当前最优解的梯度方向 $\nabla_{\mathbf{x}}\mu(\mathbf{x}_t^\star)$.

稀疏化核函数以避免维度诅咒: SAASBO (sparse axis-aligned subspaces)[74]基于一个关键假设: 输入空间 $\mathcal{X}$ 中的各维度是相关的且具有层次结构, 比如某些维度是重要的特征, 另一些维度是中等重要特征, 其余则为不重要的特征. SAASBO 精心设计核函数超参数的先验分布, 使核函数各维度变得稀疏, 从而使大部分维度成为不重要特征, 具体如下.

SAASBO 的径向核函数的形式为:

$$k^\eta(\mathbf{x},\mathbf{y}):=\sigma_k^2\exp\left[-\frac{1}{2}\sum_{i=1}^{D}\rho_i(x_i-y_i)^2\right],$$

其中, 超参数的先验分布为 $\sigma_k^2\sim\mathcal{LN}(0,100)$, $\rho_i\sim\mathcal{HC}(\tau)$, $\tau\sim\mathcal{HC}(\alpha)$. 其中 $\mathcal{LN}$ 表示对数正态分布, $\mathcal{HC}$ 表示半柯西分布. 柯西分布和正态分布形状相似, 都是中间高, 两边低, 左右对称, 故 ρ_i 的取值集中在 0 附近, 这导致大部分维度对核函数贡献很小; 另一方面, 与正态分布的细尾不同, 柯西分布具有粗尾特性, 使 ρ_i 有不小概率逃离 0, 从而使对应维度成为重要特征. 最后结合样本信息, 通过最大化后验估计来确定超参数:

$$\hat{\eta}=\arg\max_{\eta}\Pr(\mathbf{y}|\mathbf{X},\eta)p(\eta).$$

那些 ρ_i 逃离 0 的维度构成一个轴对齐的子空间, 采集函数是关于 $k^\eta(\mathbf{x},\mathbf{y})$ 的函数, 因此会倾向于探索这样的轴对齐子空间, 从而避免维度诅咒.

根据以上讨论, 我们总结基于有效低维度假设方法的优缺点, 如表 3 所示.

表 3 基于有效低维度假设的方法小结

类别	代表工作	特点	不足
基于随机降维的方法	REMBO	假设目标函数存在有效维度, 使用随机矩阵将高维空间映射到低维子空间	低维结构仅限于线性流形; 存在边界问题
	SRE	放松了有效维度假设	仍然存在边界问题
	REMBO-ϕ	提出更健壮的核函数以缓解边界问题	未完全解决边界问题
	REMBO-γ	求解最小的低维约束集以解决边界问题	增加了计算开销
	ALEBO	使用带约束的采集函数解决边界问题	低维可行集不一定包含最优解
	HeSBO	使用count-sketch实现降维, 且不会产生边界问题	子空间包含最优解的概率低
	BAXUS	使用稀疏矩阵来实现降维, 以提高子空间包含最优解的概率	—

表 3 基于有效低维度假设的方法小结 (续)

类别	代表工作	特点	不足
基于变量选择的方法	DropoutUCB	随机地从D个变量中选择d个变量	性能受限于填充策略, 而最优填充测略尚不明确
	VS-BO	基于梯度大小选择d个变量, 并提出新的填充策略	未改善DropoutUCB的遗憾界
基于学习的方法	ActiveGP	将线性嵌入作为超参数的一部分, 使用Laplace逼近求解超参数	计算开销随维度增加而平方增大
	SI-BO	利用低秩矩阵恢复算法求解投影矩阵	—
	KSIR-BO	将KSIR与BO结合, 其中KSIR是有监督的非线性降维方法	BO数据分布不满足KSIR的假设
	MGPC-BO	利用神经网络实现非线性的输入映射, 通过多输出GPR构造逆映射	需要大量数据以确保学习结果准确
VAE-BO	CVAE	利用VAE实现非线性输入映射以及输出映射	未充分利用数据的标签信息; 候选点解码失败概率高
	GVAE和 SD-VAE	结合上下文无关文法提高候选点的解码成功概率	并非所有SMILES字符串都符合上下文无关文法
	JT-VAE 和MHG-VAE	结合图来约束解码过程, 以提高候选点的解码成功概率	增加了模型复杂度
	COLD和BVAE	缓解“死区”问题, 并与实际场景解耦	解码成功率低于场景耦合的方法
	VAE-guided-BO 和T-LBO	利用标签信息构建更适合优化的隐空间	未缓解“死区”问题
其他	LineBO	只在某直线上优化采集函数, 以降低采集函数的优化开销	收敛速度慢
	SAASBO	稀疏化核函数以避免维度诅咒	计算开销大

5 基于加性假设的高维贝叶斯优化

为避免维度诅咒, 一些方法引入了加性假设, 这一假设指目标函数 f 可分解为加性形式.

假设 1. 加性假设, 见 Add-GP-UCB[19]的定理 1.

$$f(\mathbf{x}) = f^{(1)}(\mathbf{x}^{(1)}) + \ldots + f^{(M)}(\mathbf{x}^{(M)}),$$

其中, $\mathbf{x}^{(i)}$ 是向量 $\mathbf{x}$ 某些维度的集合, 且每个变量至多出现在一个集合中, 即 $\mathbf{x}^{(i)} \cap \mathbf{x}^{(j)} = \varnothing, \forall i, j$.

基于加性假设, 我们可分别优化 M 个低维函数 $f^{(i)}(\mathbf{x}^{(i)})$, 从而避免维度诅咒. 最终, 拼接 M 个函数的最优解 $\mathbf{x}_\star^{(i)}$ 即可得高维函数的最优解 $\mathbf{x}_\star$.

以下简要介绍如何基于加性假设来构建 GPR, 以及如何对高维变量进行分组.

5.1 加性高斯过程回归

基于加性假设, Add-GP-UCB 用 M 个的 GPR 分别拟合 M 个低维的函数 $f^{(i)}$, 即:

$$f^{(i)} \sim \mathcal{GP}\left(m^{(i)}(\mathbf{x}^{(i)}), k^{(i)}(\mathbf{x}^{(i)}, \mathbf{x}^{(i')})\right).$$

根据文献 [19] 的观察 1, 这等价于一个高维 GPR 拟合高维函数 f :

$$f \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{5}$$

其中,

$$m(\mathbf{x}) = \sum\nolimits_i m^{(i)}(\mathbf{x}^{(i)}), \qquad k(\mathbf{x}, \mathbf{x}') = \sum\nolimits_i k^{(i)}(\mathbf{x}^{(i)}, \mathbf{x}^{(i)'}).$$

基于加性假设, 采集函数也是加性函数, 以 *UCB* 为例,

$$\alpha_{UCB}(\mathbf{x}) = \sum_i \mu_n^{(i)}(\mathbf{x}^{(i)}) + \beta_{n+1}^{1/2}\sigma_n^{(i)}(\mathbf{x}^{(i)}).$$

优化采集函数时, 可分别优化 M 个低维采集函数, 并拼接 M 个函数的最优解 $\mathbf{x}_{n+1}^{(i)}$ 即可得下一个采样点 $\mathbf{x}_{n+1}$. 关于如何选择 M 个均值函数和 M 个核函数以及如何确定超参数, 这些方法与原始 GPR 相同.

然而, 仍有一个问题需要解决: 如何将 $\mathbf{x}$ 中的 D 个变量分配进 M 个集合里面. 一个分配方案的优劣可通过似然概率来评估: 优秀的分配方案更利于 GPR 的拟合, 故 GPR 的似然概率更高. 尽管如此, 要在 M^d 种分配方案中选择最优者仍是一个挑战. 比如 Add-GP-UCB 仅在 M^d 个分配方案中随机选择若干个候选方案, 在候选方案中选择似然概率最大的方案. 显然, 这无法保证得到的分配方案是足够好的.

上述分配问题是一个组合优化问题 (属于 NP 难问题), 为了高效地优化该问题, 通常需要引入一些特殊结构, 将该问题转化为连续优化问题.

5.2 基于 MCMC 采样分组方案

Add-BO-MH[75]首先给每个分组方案赋予一个先验分布, 结合 GPR 的似然分布, 最后得到每个分组的后验分布, 所需的方案则从该后验分布中采样. 具体而言, 假设所有方案组成集合 G, 给定 G 一个先验分布 $p(G)$, 由贝叶斯定理, 每个分组的后验概率为:

$$p(g_i|\mathcal{D}) = \frac{p(\mathbf{y}|\mathbf{X}, g_i)p(g_i)}{\sum_j p(\mathbf{y}|\mathbf{X}, g_j)p(g_j)}.$$

该后验概率可量化集合中每个方案对数据的解释能力. 然而, 穷举所有方案的后验概率在计算上是不可行的, 故使用马氏链蒙特卡罗 (Markov chain Monte Carlo, MCMC) 方法来采样后验分布.

基于 MCMC 采样后验分布: 这里选用的 MCMC 方法是 Metropolis-Hastings (简称 MH 算法)[76]. MCMC 方法的核心是构造适当的马氏链, 使其平稳分布为待采样的后验分布. 而 MH 算法的主要任务是生成满足上述要求的马氏链 $\{g_0, g_1, \ldots\}$, 即给定方案 g_t 下, 转移到下一个方案 g_{t+1}. MH 算法的构造过程如下.

(1) 构造合适的建议分布 (proposal distribution) $q(\cdot|g_t)$.

(2) 从 $q(\cdot|g_t)$ 产生下一个方案 g'. 具体而言, 分别以 1/2 的概率选择“拆分”或“合并”操作. 若选择“拆分”, 则随机选择一组变量, 将其平分为两组. 若选择“合并”, 则随机选择两组变量, 将其合并为一组.

(3) 按一定的概率接受 g', 若接受 g', 则令 $g_{t+1} = g'$; 否则令 $g_{t+1} = g'$. 其中接受概率定义为:

$$A(g'|g_t) := \min\left(1, \frac{p(\mathbf{y}|\mathbf{X}, g')q(g_t|g')}{p(\mathbf{y}|\mathbf{X}, g_t)q(g'|g_t)}\right).$$

5.3 基于 Dirichlet 过程的分组方法

Add-BO-SKL[77]假设分组方案采样自多项分布, 则后验分布为 Dirichlet 分布, 而所需的方案从该 Dirichlet 分布中采样. 具体而言, D 个变量被分配到 M 个集合可视为 D 次独立重复试验, 每次试验中变量被分配到第 i 组的概率为 θ_i. 用随机变量 $z_j = i$ 来表示第 j 维变量被分配第 i 个集合, 则 $\Pr(z_j = i) = \theta_i$. 根据假设, M 个集合的元素数量 $(n_1, \ldots, n_M)$ 满足多项分布:

$$(n_1, \ldots, n_M) \sim Multi(\theta_1, \ldots, \theta_M),$$

其中, $n_1 + \ldots + n_M = D$. 由于 Dirichlet 分布是多项分布的共轭先验 (即当 θ 的先验分布是 Dirichlet 分布时, 其后验分布也是 Dirichlet 分布), 为简化计算, 令多元变量 θ 的先验分布为 Dirichlet 分布, 即:

$$(\theta_1, \ldots, \theta_M) \sim DIR(\alpha_1, \ldots, \alpha_M).$$

可得后验分布 $DIR(\alpha_1 + n_1, \ldots, \alpha_M + n_M) = D$. 最后对后验分布进行采样即可得所需的方案, 在实践中, 通常使用 MCMC 方法采样后验分布. 整体结构如图 6 所示. 其中, η 是核函数的超参数, z 控制输入空间的分解.

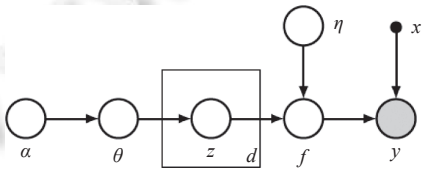


图 6 GPR 的图模型[77]

5.4 基于概率图模型的分组方法

以上分组方法都假设每一维度的变量至多出现在一个集合中, 即 $\mathbf{x}^{(i)} \cap \mathbf{x}^{(j)} = \varnothing, \forall i, j$. G-Add-GP-UCB (graph Add-GP-UCB)[78]放松了该假设, 允许一个变量可以同时出现在不同的集合中.

假设 2. 扩展的加性假设:

$$f(\mathbf{x}) = f^{(1)}(\mathbf{x}^{(1)}) + \ldots + f^{(M)}(\mathbf{x}^{(M)}),$$

其中, $\mathbf{x}^{(i)}$ 是向量 $\mathbf{x}$ 某些维度的集合, 一个变量可以同时出现在不同的集合中, 即 $\mathbf{x}^{(i)} \cap \mathbf{x}^{(j)} \neq \varnothing, \exists i, j$.

G-Add-GP-UCB 的核心思想是将相关变量放入同一个集合, 从而将变量分配问题转化为检测变量间相关性的问题. 为此, 该方法使用图模型来建模变量间的关系: 变量建模为顶点, 变量间的相关性用边来表示, 变量集合对应图中的最大团 (maximal clique).

使用邻接矩阵 $\mathbf{Z}$ 来表示图, $z_{ij} = 1$ 表示变量 x_i 和 x_j 相关, $z_{ij} = 0$ 则两变量无关. 选择伯努利分布作为 z_{ij} 的先验, 结合 GPR 的似然分布, 可得 z_{ij} 的后验分布 $p(z_{ij}|\mathcal{D}_n)$. 最终利用 MCMC 方法采样后验分布即可确定 z_{ij} 的取值.

此外, 采集函数是每个低维采集函数的总和:

$$\alpha(\mathbf{x}) = \sum_{i=1}^{M} \alpha^{(i)}(\mathbf{x}^{(i)}).$$

然而, 集合之间存在部分重叠变量, 因此不能独立地优化各个低维采集函数. 为此, G-Add-GP-UCB 使用消息传递算法来优化采集函数.

值得注意的是, 消息传递算法的计算开销随最大团增大而指数级增长. 为降低其计算开销, Tree-GP-UCB[79]使用树模型来建模变量间的关系, 以简化图模型. 具体而言, 每次从后验分布采样邻接矩阵 z_{ij} 时, 若 $z_{ij} = 1$, 则检查新增的边是否形成回路, 若形成回路则令 $z_{ij} = 0$, 以保证图模型始终是一棵树.

5.5 基于随机分组的方法

以上分组方法都基于学习, 它们容易为局部分解结构所误导, 而这种局部的分解结构往往难以推广到全局. 为此, RDUCB (random decompositions UCB)[80]提出一种数据无关的分组方法, 其核心是构建一棵随机树作为分解结构. 该随机分组方法不仅克服了前述缺陷, 还提供了更为可靠的理论保证, 即可分析的遗憾界. 具体而言, 分解结构所引入的错配误差定义为:

$$\epsilon_t := \min_{f_t \in \mathcal{H}_t} |f_t - f|_\infty,$$

其中, $\mathcal{H}_t$ 是 $k^{g_t}(\mathbf{x}, \mathbf{x}')$ 的再生希尔伯特空间, g_t 是第 t 轮的分解结构. 在随机分组方法中, 这种错配误差存在上界, 即:

$$\mathbb{E}\left[\sum_{t=1}^{T} \epsilon_t\right] < TM\left(1 - \frac{2E}{D(D-1)}\right),$$

其中, $M := \sum_c |f_c|_\infty$, E 为树的边数. 相比之下, 基于学习的分组方法难以确定错配误差的上界.

5.6 其他方法

投影加性假设: “投影加性假设”推广了“加性假设”, 即:

$$f(\mathbf{x}) = f^{(1)}(\mathbf{W}^{(1)}\mathbf{x}) + f^{(2)}(\mathbf{W}^{(2)}\mathbf{x}) + \ldots + f^{(M)}(\mathbf{W}^{(M)}\mathbf{x}),$$

其中, $\mathbf{W}^{(i)} \in \mathbb{R}^{d \times D}$, $\mathbf{W} = [\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(M)}] \in \mathbf{R}^{D \times D}$ 是投影矩阵. 当投影矩阵 $\mathbf{W} = \mathbf{I}$ 时, “投影加性假设”便退化为“加性假设”, 故“加性假设”可视为“投影加性假设”的特例.

在处理流程上, RPP-GP-UCB (restricted projection pursuit GP-UCB)[81]引入了输入变换 $\mathbf{Z} = \mathbf{WX}$. 除此之外, 其余步骤与 Add-GP-UCB 保持一致. 而投影矩阵 $\mathbf{W}$ 则通过以下方法求解: 将投影矩阵视为 GPR 模型的超参数, 由 EM 算法优化超参数即可得投影矩阵[82].

根据以上讨论, 我们对基于加性假设的方法的优缺点总结如表 4 所示.

表 4 基于加性假设方法的小结

代表工作	特点	不足
Add-GP-UCB	提出加性结构的GPR模型	未能充分优化分组方案
Add-BO-MH	令分组方案服从一个先验分布, 结合GPR模型的似然分布, 构建后验分布, 并用MH算法采样该后验分布	要预先定义方案子集; 有效性验证仅限于10维以下的问题空间
Add-BO-SKL	假设分组方案服从多项分布, 构建Dirichlet后验分布, 并用Gibss算法采样该后验分布	采用多项分布作为先验, 忽略了变量间的相关性
G-Add-GP-UCB	扩展了加性假设, 允许分组间存在重叠变量, 并引入概率图模型刻画变量间的关系	优化采集函数的计算开销大
Tree-GP-UCB	为降低优化采集函数的开销, 引入树结构建模变量间的关系	分组方法受局部分解结构所误导, 容易陷入局部次优解
RDUCB	为避免受局部分解结构误导, 提出随机分组方法, 并推导出可分析的遗憾界	—
RPP-GP-UCB	扩展了加性假设, 引入投影加性假设	引入过多假设, 难以应用于现实问题

6 基于局部搜索的高维贝叶斯优化

与以上方法不同, 基于局部搜索的方法无需要引入额外假设, 它们建立在两个关键观察之上: (1) GPR 中的平稳核函数使得代理模型的函数变化率在所有区域保持一致; (2) 采集函数过度探索整个高维空间. 基于这些观察, 此类方法专注于采样最有潜力的局部区域.

6.1 基于信任域的贝叶斯优化

基于信任域的贝叶斯优化 (trust-region BO, TuRBO)[24]将上述局部区域称为信任域 (trust region, TR), 并自适应地调整其大小. 具体而言, TR 呈超矩形, 其调整要平衡两个方面因素: (1) TR 应该足够大以包含优质解; (2) TR 也应适度小以确保代理模型的有效性. 故在优化过程中若发现更优解, 则扩充 TR, 反之则缩小 TR.

TuRBO 同时维持 m 个 TR, 每个 TR 构建独立的代理模型, 即 $f_\ell \sim \mathcal{GP}_\ell^{(t)}(\mu_\ell(\mathbf{x}), k_\ell(\mathbf{x},\mathbf{x}'))$, 其中下标 ℓ 表示第 ℓ 个 TR, 上标 t 表示第 t 次迭代. 这允许不同区域使用不同代理模型, 实现了异构的代理模型. 此外, 采集函数限制在各个 TR 中, 有效避免了过度探索. 具体而言, 首先将 TR_ℓ 离散化: 从 TR_ℓ 中随机选择若干个点, 记为 TR'_ℓ, 然后根据采集函数从 TR'_ℓ 中选择候选点, 即:

$$\mathbf{x}_i^{(t)} = \arg\max_{\ell} \arg\max_{\mathbf{x}\in \mathrm{TR}'_\ell} \alpha(\mathbf{x}).$$

重复此过程 q 次, 即可得 q 个候选点 $\mathbf{x}_i^{(t)}$, $i = 1,\ldots,q$.

基于信任域的方法不仅适用于连续输入空间, 也可扩展到混合的输入空间[83] (即某些维度的变量为离散型). 这是因为 TuRBO 在最大化采集函数时会离散化为 TR, 使其同样适用于离散空间.

6.2 基于划分搜索空间的贝叶斯优化

TuRBO 构建了多个局部模型, 而未分割整个搜索空间, 这导致局部区域之间存在未考虑的区域. 相比之下, 以下方法则考虑划分搜索空间 $\mathcal{X}$, 并集中采样最有潜力的区域. 例如, BaMSOO 将搜索空间划分为若干超矩形, VOOT 将搜索空间划分为 Voronoi 图, LA-MTCS 则更灵活地将搜索空间划分为不规则区域.

层次地将搜索空间划分为超矩形: DOO (deterministic optimistic optimization)[84]和 SOO (simultaneous optimistic optimization)[84]层次地分割可行集 $\mathcal{X}$, 若将局部区域视为节点, 则层次分割过程相当于构建一棵树. 具体而言, 每个节点有 k 个子节点, 比如第 h 层第 m 个节点的子节点为 $\{(h+1, km+i)\}_{0\leqslant i<k-1}$. 这意味着区域 $X_{h,m}$ 被等分为 k 个区域:

$$\{X_{h+1,km+i} : 0 \leqslant i < k-1\}.$$

DOO 假设存在一个半度量 ℓ 使得 $f(\mathbf{x}^*) - f(\mathbf{x}) \leqslant \ell(\mathbf{x},\mathbf{x}^*)$. 局部区域 $X_{h,m}$ 的半径定义为:

$$\delta(h,m) := \sup_{\mathbf{x}} \ell(\mathbf{x}_{h,m}, \mathbf{x}).$$

其中, $\mathbf{x}_{h,m}$ 是 $X_{h,m}$ 的中点. 在每次扩展叶节点时, 选择 $f(\mathbf{x}_{h,m})+\delta(h,m)$ 最大的叶节点:

$$\mathbf{x}_{h_\star,m_\star} = \arg\max_{(h,m)\in Leaf} f(\mathbf{x}_{h,m})+\delta(h,m),$$

其中, $\delta(h,m)$ 量化该区域的探索价值, 从而权衡“探索”与“利用”.

SOO 放松了 DOO 的假设, 它仅假设半度量 ℓ 存在而无需知道其具体形式, 也无需定义局部区域半径. 为权衡“探索”与“利用”, SOO 在每轮遍历树时, 一层至多扩展一个叶节点, 且该叶节点的值 $f(\mathbf{x}_{h,m})$ 要大于比同层和浅层的所有叶节点. 故一个叶节点被选中的情况有两种: (1) 当浅层和同层已没有其他叶节点时, 该叶节点将被选中, 这体现了“探索”; (2) 该叶节点的值 $f(\mathbf{x}_{h,m})$ 大于同层和浅层的所有叶节点, 这体现了“利用”.

BaMSOO (Bayesian multi-scale optimistic optimization)[85]通过在 SOO 的基础上引入 GPR 模型. 与 SOO 相比, 它降低了采样复杂度; 与 BO 相比, 它无需最大化采集函数. 尤其对于传统 BO, 其收敛性需要假设总能找到采集函数的最优解, 而 BaMSOO 的收敛则不需要该假设. 具体而言, 当扩展叶节点 (h,m) 时, 其子节点的值定义为:

$$g(\mathbf{x}_{h+1,km+i}) := \begin{cases} f(\mathbf{x}_{h+1,km+i}), & \text{if } UCB(\mathbf{x}_{h+1,km+i}) > f_{\max} \\ LCB(\mathbf{x}_{h+1,km+i}), & \text{otherwise} \end{cases}.$$

不像 SOO 会采样所有子节点, BaMSOO 只采样 *UCB* 大于当前最优值的节点, 其他子节点则用 *LCB* 作为标记, 从而降低了采样复杂度以及确保候选解的 *UCB* 单调上升.

与 BaMSOO 类似, IMGPO (infinite-metric GP optimization)[86]也在 SOO 上增加 GPR 模型, 但它利用 *UCB* 来估计半度量 ℓ, 以达到指数级遗憾界, 而 BaMSOO 的遗憾界为多项式级. 具体而言, IMGPO 扩展叶子节点要同时满足两个条件: (1) 该叶节点的值 $f(\mathbf{x}_{h,m})$ 大于同层和浅层的所有叶节点; (2) 该叶节点的 *UCB* 大于深层的叶节点.

将搜索空间划分为不规则区域: VOOT (Voronoi optimistic optimization)[87]利用已有样本构建 Voronoi 图, 然后集中采样当前最优解所在的区域, 从而得到新的候选解.

LA-MTCS[88]则使用 SVM 分类器构建不规则的局部区域, 并基于 *UCB* 选择局部区域, 以权衡“探索”与“利用”. 具体步骤如下.

(1) 初始化. 开始时, 根据“划分操作”和已有样本构建一棵蒙特卡洛搜索树. 然后根据“选择操作”选择一个叶节点, 并在该叶节点所代表的局部区域内进行贝叶斯优化.

(2) 划分操作. 当划分一个叶节点时, 首先在该叶节点代表的区域中聚类, 将样本划分为好坏两簇, 然后训练 SVM 分类器, 从而二分该区域, 并生成两个子节点分别代表两个更小的区域.

(3) 选择操作. 类似于多臂赌博机 (multi-arm bandits)[89]问题, 从根节点开始, 计算其子节点的 *UCB* 并选择较大的节点, 然后从该节点开始, 继续重复上述操作直到叶节点为止.

然而, VOOT 和 LA-MTCS 更灵活的划分方式也使其缺乏遗憾界, 即缺乏收敛性保证.

根据以上讨论, 我们总结了基于局部搜索方法的优缺点, 如表 5 所示.

表 5 基于局部搜索方法的小结

代表工作	特点	不足
EBO[90]	对输入空间进行划分, 在每个区域内拟合独立的加性GPR, 并进行贝叶斯优化	在局部区域中引入过强的加性假设
TuRBO	在多个局部区域建立代理模型, 在局部区域中采样候选点以避免过度探索	由于未划分搜索空间, 存在未被考虑的局部区域
BaMSOO	层次地将搜索空间划分为超矩形, 并在最具潜力的局部区域中采样候选点	仅达到多项式级的遗憾界
IMGPO	在BaMSOO基础上, 利用*UCB*估计半度量, 以达到指数级遗憾界	—
VOOT	采用Voronoi图实现更灵活的空间划分	算法缺乏收敛性保证
LA-MCTS	采用SVM分类器将搜索空间划分为不规则区域	算法缺乏收敛性保证

7 高维贝叶斯优化的应用领域

本节将概述高维贝叶斯优化在当前主要应用领域的发展.

7.1 强化学习

高维贝叶斯优化在强化学习的策略搜索中取得了显著成果. GIBO[91]和 BO-MPD[92]研究了线性策略的情况, 即 $\pi_\theta:\mathbb{R}^D\to\mathbb{R}^m$, $\pi_\theta(\mathbf{s})=\mathbf{A}\mathbf{s}+\mathbf{b}$, 参数空间为 $\theta=(\mathbf{A},\mathbf{b})\in\mathbb{R}^{p\times m+m}$. 它们在 CartPole、Swimmer 及 Hopper 等强化学习环境中使用局部 BO 搜索策略以最大化收益函数, 结果表明高维 BO 比随机搜索方法更快达到收益阈值.

若已知初始策略, 局部优化往往足以取得令人满意的性能. 比如, 文献 [91] 使用局部 BO 微调模仿学习的策略, 以快速优化机器人控制策略; CRBO[93]使用局部 BO 微调强化学习策略, 使得智能体快速适应新的收益信号.

7.2 机器人

高维贝叶斯优化在机器人领域也有成功应用. 比如, 文献 [52] 将高维 BO 用于控制六足机器人行走, 使其能够到达目标位置, 同时避免关节速度和高度偏差过大. 文献 [77] 利用高维 BO 配置三连杆平面双足机器人的参数, 以提高其行走速度. 文献 [90] 将高维 BO 用于配置两个机器手臂的参数, 使其能够将物体推至目标位置.

7.3 混合整数求解器

高维贝叶斯优化在配置混合整数规划 (mixed integer linear programming, MILP) 求解器的超参数方面表现出色. 不同的求解器具有不同的超参数空间, 比如 LPSolve (https://lpsolve.sourceforge.net/5.5/) 有 74 维超参数空间, 而 SCIP[94]则有 136 维超参数空间. 文献 [23] 使用随即嵌入 BO 优化 LPSolve 的超参数, 文献 [79] 则使用加性 BO 优化 LPSolve 的超参数.

7.4 工程系统

高维贝叶斯优化在配置工程系统参数方面也有广泛应用. 比如, 文献 [73] 使用高维 BO 配置电子激光器的参数, 以最大化激光能量; 文献 [74] 使用高维 BO 配置汽车设计的参数, 以最小化汽车质量; 文献 [95] 使用变量选择 BO 配置天线参数, 以降低天线传输损耗; 文献 [96] 使用变量选择 BO 优化焊接梁结构和燃气管道输送系统, 以及配置合金设计参数以最大化合金性能.

7.5 自动机器学习

在自动机器学习领域, 高维贝叶斯优化同样发挥重要作用. 比如, 文献 [80] 使用高维 BO 优化神经网络的 9 种超参数, 包括学习率、丢弃率、隐藏层单元数及激活函数种类等; 文献 [52] 使用高维 BO 搜索卷积神经网络的结构; 文献 [97] 使用高维 BO 优化线性回归模型的正则化超参数; 文献 [19] 使用加性 BO 优化级联分类器的超参数, 以提高人脸识别准确率; 文献 [75] 使用加性 BO 配置矩阵补全算法的参数, 以减小图像的重构误差; 文献 [74] 则使用高维 BO 优化核支持向量机 (kernel support vector machine, KSVM) 的超参数.

7.6 生物、化学

高维贝叶斯优化也成功应用于生物和化学领域. 比如, 文献 [98] 使用变量选择 BO 配置微藻动态代谢模型的参数, 以提高模型的预测精度; 文献 [9] 使用 VAE-BO 生成新分子, 以提高分子的水-辛醇分配系数.

7.7 线性二次型调节器

高维贝叶斯优化也成功应用于线性二次型调节器 (linear quadratic regulator, LQR). LQR 是控制理论中的一个基本问题, 旨在控制动力系统并最小化二次代价. 令动力系统为 $\mathbf{x}(t+1)=\mathbf{A}\mathbf{x}(t)+\mathbf{B}\mathbf{u}(t)+\mathbf{w}(t)$, 代价函数为:

$$J:=\lim_{T\to\infty}\frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1}\mathbf{x}^{\top}(t)\mathbf{Q}\mathbf{x}(t)+\mathbf{u}^{\top}(t)\mathbf{R}\mathbf{u}(t)\right],$$

其中, $\mathbf{x}(t)$, $\mathbf{u}(t)$, $\mathbf{w}(t)$ 分别为 t 时刻的系统状态、控制输入和高斯噪声, 加权矩阵 $\mathbf{Q}$ 和 $\mathbf{R}$ 为正定矩阵.

当动力系统已知时, LQR 问题存在可高效求解的最优解. 然而, 当动力系统未知时 (即矩阵 $\mathbf{A}$ 和 $\mathbf{B}$ 未知), LQR

成为极具挑战性的黑盒优化问题. GIBO 使用局部 BO 求解该问题, 其样本复杂度与 LSPI[99]相当, 但小于 ARS[100].

此外, 若代价函数的加权矩阵 $\mathbf{Q}$ 和 $\mathbf{R}$ 未知, LQR 则成为另一种黑盒优化问题. 文献 [101] 使用基于熵搜索的 BO 求解该问题.

8 软件实现

高维贝叶斯优化领域存在多种测试场景和软件库, 本节将介绍几个代表性的测试场景和常用软件库.

8.1 测试场景

8.1.1 基于数学函数的测试场景

这类实验通常采用数学函数作为测试场景, 不同类型的贝叶斯优化方法会选择与其基本假设相契合的数学函数. 例如, 基于加性假设的方法倾向于选择加性数学函数, 基于有效低维度的方法则倾向于选择稀疏维度的数学函数, 具体如下.

加性数学函数: 加性高维函数由若干个子函数相加构成. 例如, 首先构造低维子函数:

$$f_d(\mathbf{x}) = \log\left(0.1\frac{1}{h_d^d}\exp\left(\frac{\|\mathbf{x}-\mathbf{v}_1\|^2}{2h_d^2}\right)+0.1\frac{1}{h_d^d}\exp\left(\frac{\|\mathbf{x}-\mathbf{v}_2\|^2}{2h_d^2}\right)+0.8\frac{1}{h_d^d}\exp\left(\frac{\|\mathbf{x}-\mathbf{v}_3\|^2}{2h_d^2}\right)\right),$$

其中, $\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3$ 是固定的 d 维向量, $h_d=0.01d^{0.1}$, 可行集限制在 $\mathbf{x}\in[-1,1]^d$, 最优解为 $\mathbf{x}^*=\mathbf{v}_3$. 若创建 M 组变量, 每组变量 $\mathbf{x}^{(i)}$ 独立使用一个低维子函数 $f_d(\mathbf{x}^{(i)})$, 则高维的函数可表示为:

$$f(\mathbf{x}) = f_d(\mathbf{x}^{(1)})+\ldots+f_d(\mathbf{x}^{(M)}).$$

值得注意的是, 函数 $f_d(\mathbf{x})$ 有 3 个峰, 故 $f(\mathbf{x})$ 总共有 3^M 个峰.

一些测试场景则直接使用加性 GPR (见公式 (5)) 作为测试函数[77], 当评估某点 $\mathbf{x}$ 的目标函数值时, 只需采样该 GPR 的后验分布. G-Add-GP-UCB 进一步扩展了该测试函数, 其加性 GPR 允许不同组间存在重叠变量.

稀疏维度的数学函数: 基于有效低维度假设的贝叶斯优化则通常使用稀疏维度的数学函数, 例如, 为 2 维的 Branin 函数加入 $D-2$ 个无关维度, 即可将其扩展为 D 维的 Branin 函数. 类似地, 6 维的 Hartmann 函数或者 d ($d\leqslant 10$) 维的 Ackley 函数都可以加入无关维度来扩展为高维的函数.

8.1.2 仿真实验

贝叶斯优化在实际应用中通常用于机器学习算法或工程系统的参数调优. 例如:

(1) SVM 调参任务: SVM 的核函数的超参数数量约等于数据的维度, 若分类或回归任务的数据维度较高, SVM 调参任务则成为高维黑盒优化问题[74].

(2) 神经网络结构搜索任务 (neural architecture search, NAS): NAS 任务可转化为具有 36 个设计参数的拓扑优化问题[52].

(3) 车辆设计任务 MOPTA08: 该任务具有 124 个设计参数, 用于描述材料、仪表和车辆形状, 且不具有明显的低维结构[74].

(4) 六足机器人的控制任务: 该任务有 72 个参数, 旨在学习最佳策略参数, 使机器人能到达目标位置, 同时避免关节速度过快和各关节高度偏差过大[52].

除了以上具有数值含义的优化对象, 还有一些结构化的输入对象, 如分子、表达式、拓扑等, 这些数据往往用于 VAE-BO 的仿真实验[10].

8.2 软件库

本节将介绍几个常用软件库.

(1) BoTorch[102]是基于 PyTorch 的贝叶斯优化编程框架, 集成了常用的采集函数和经典的 BO 算法. 此外, BoTorch 提供了一些高维贝叶斯优化算法 (如 TuRBO、SAASBO、BAXUS) 和多目标贝叶斯优化算法的实现.

(2) Ax-platform (https://github.com/facebook/Ax) 是关于贝叶斯优化以及多臂赌博机的代码库. 其贝叶斯优化的

框架基于 BoTorch 开发, 包含了一些高维贝叶斯优化算法 (如 ALEBO、SAASBO) 和多目标贝叶斯优化算法的实现.

(3) HEBO (https://github.com/huawei-noah/HEBO) 是关于贝叶斯优化的代码库, 实现了多种高维贝叶斯优化算法 (如 RDUCB、T-LBO). 此外, 它还包含了前沿的采集函数优化方法 (如 HEBO[103]、CompBO[104]).

9 未来研究方向

随着高维贝叶斯优化的持续发展, 它已广泛应用于各类科学和工程领域, 为多种高维黑盒优化问题提供了一种样本高效的求解技术. 高维贝叶斯优化的研究工作主要围绕 3 种高维扩展思路展开: 基于有效低维度假设的方法主要适用于存在低维结构的目标函数, 可处理上千乃至上万维的输入空间; 基于加性假设的方法主要适用于存在加性结构的目标函数, 可处理几十维的输入空间; 基于局部搜索的方法广泛适用于各种目标函数, 可处理几十维的输入空间.

尽管高维贝叶斯优化研究已取得显著进展, 但仍有许多问题亟待解决, 主要表现为以下几个方面.

(1) 假设的局限性. 许多方法基于较强的假设, 如要求目标函数存在低维结构或加性结构, 然而这在现实场景中并不总是成立. 因此, 如何放宽这些假设并降低违反假设带来的误差, 是未来研究的重点问题.

(2) 弱假设方法的扩展性不足. 虽然基于局部搜索的方法无需额外假设, 但其维度处理能力通常限于几十维. 因此, 如何提升这类方法的维度处理能力, 也是未来研究的一个关键问题.

(3) 许多方法缺乏收敛性保证. 高维贝叶斯优化的收敛性分析集中于基于变量选择和加性假设的方法, 然而基于降维和局部搜索的方法普遍缺乏收敛性保证. 这导致许多方法的可行性、正确性和鲁棒性都受到质疑. 因此, 如何进一步这些方法的收敛性保证, 是未来研究的另一个关键问题.

与此同时, 贝叶斯优化其他领域的迅速发展, 也为高维贝叶斯优化带来了新的机遇, 主要表现在以下几个方面.

(1) 与多目标贝叶斯优化融合. 许多科学和工程问题需同时优化多个相互竞争的黑盒函数. 当目标函数评估成本高昂时, 多目标贝叶斯优化因其卓越的样本利用率而备受青睐. 然而, 类似于传统 BO, 多目标 BO 同样面临高维挑战. 目前已有一些工作取得初步进展, 如文献 [105]、文献 [106] 和文献 [107] 分别将 TuRBO、SAASBO 和 LA-MCTS 扩展到多目标优化中. 进一步探索多目标优化与高维 BO 的融合, 是拓宽 BO 应用范畴的关键.

(2) 应用于混合搜索空间. 高维 BO 通常假设搜索空间是连续的, 但许多实际问题的输入空间由连续和离散变量混合而成, 如 MILP 的调参任务[20]. 目前已有一些工作取得初步进展, 如文献 [83] 将 TuRBO 扩展到混合搜索空间中. 深入探索高维 BO 在混合搜索空间的扩展同样是拓宽 BO 应用范畴的重要途径.

(3) 与多成本优化融合. 多成本优化假设当评估目标函数时, 可花费高代价得到噪音小的观测结果, 也可花费低代价得到噪音大的观测结果. 比如, 在机器学习调参任务中, 减少迭代次数或缩小训练集和验证集的规模可降低时间成本, 但会增加观测结果的噪声. 目前已有一些工作将随机降维 BO 应用于多成本优化[52], 但很少有工作将加性 BO 或局部 BO 应用于多成本优化. 进一步融合高维 BO 与多成本优化, 是拓宽 BO 应用范畴的重要途径.

(4) 与多任务优化融合. 多任务优化旨在将先前优化任务中获得的知识迁移到新任务中, 以加快优化过程. 文献 [63] 提出了多任务 BO, 类似于传统 BO, 多任务 BO 同样面临高维挑战. 为此, 文献 [52] 融合随机降维 BO 与多任务 BO, 文献 [93] 融合局部 BO 与多任务 BO. 进一步探索高维 BO 与多任务优化的融合, 是拓宽 BO 应用范畴的关键.

10 总 结

贝叶斯优化作为黑盒优化领域的关键技术, 近年来受到广泛关注与研究. 高维贝叶斯优化作为贝叶斯优化的关键扩展技术, 相当大地拓展了贝叶斯优化的应用场景. 本文根据方法的假设和特点, 将高维贝叶斯优化的研究分为 3 类: 基于有效低维度假设的方法、基于加性假设的方法以及基于局部搜索的方法, 并详细阐述和分析了高维贝叶斯优化的研究进展. 在此基础上总结并展望了高维贝叶斯优化的未来研究方向. 总之, 基于有效低维度假设的方法主要适用于存在低维结构的目标函数, 支持上千乃至上万维的输入空间; 基于加性假设的方法主要适用于具有加性结构的目标函数, 支持几十维的输入空间; 基于局部搜索的方法能够普遍适用于各种目标函数, 支持几十维

的输入空间. 另外高维贝叶斯优化与其他领域的融合, 进一步拓宽了贝叶斯优化的应用范畴, 为解决更复杂的优化问题提供了新的可能性.

References:

[1] Močkus J. On Bayesian methods for seeking the extremum. In: Proc. of the 1975 IFIP Technical Conf. on Optimization Techniques. Novosibirsk: Springer, 1975. 400–404. [doi: 10.1007/3-540-07165-2_55]

[2] Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. In: Proc. of the 25th Int'l Conf. on Neural Information Processing Systems. Lake Tahoe: ACM, 2012. 2951–2959.

[3] Hutter F, Hoos HH, Leyton-Brown K. Sequential model-based optimization for general algorithm configuration. In: Proc. of the 5th Int'l Conf. on Learning and Intelligent Optimization. Rome: Springer, 2011. 507–523. [doi: 10.1007/978-3-642-25566-3_40]

[4] Klein A, Falkner S, Bartels S, Hennig P, Hutter F. Fast Bayesian optimization of machine learning hyperparameters on large datasets. In: Proc. of the 20th Int'l Conf. on Artificial Intelligence and Statistics. Fort Lauderdale: PMLR, 2017. 528–536.

[5] Letham B, Karrer B, Ottoni G, Bakshy E. Constrained Bayesian optimization with noisy experiments. Bayesian Analysis, 2019, 14(2): 495–519. [doi: 10.1214/18-BA1110]

[6] Negoescu DM, Frazier PI, Powell WB. The knowledge-gradient algorithm for sequencing experiments in drug discovery. INFORMS Journal on Computing, 2011, 23(3): 346–363. [doi: 10.1287/ijoc.1100.0417]

[7] Kandasamy K, Neiswanger W, Schneider J, Póczos B, Xing EP. Neural architecture search with Bayesian optimisation and optimal transport. In: Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems. Montréal: ACM, 2018. 2020–2029.

[8] Zhou HP, Yang MH, Wang J, Pan W. BayesNAS: A Bayesian approach for neural architecture search. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 7603–7613.

[9] Gómez-Bombarelli R, Wei JN, Duvenaud D, Hernández-Lobato JM, Sánchez-Lengeling B, Sheberla D, Aguilera-Iparraguirre J, Hirzel TD, Adams RP, Aspuru-Guzik A. Automatic chemical design using a data-driven continuous representation of molecules. ACS Central Science, 2018, 4(2): 268–276. [doi: 10.1021/acscentsci.7b00572]

[10] Grosnit A, Tutunov R, Maraval AM, Griffiths RR, Cowen-Rivers AI, Yang L, Zhu L, Lyu WL, Chen ZT, Wang J, Peters J, Bou-Ammar H. High-dimensional Bayesian optimisation with variational autoencoders and deep metric learning. arXiv:2106.03609, 2021.

[11] Ru BX, Cobb AD, Blaas A, Gal Y. BayesOpt adversarial attack. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: OpenReview.net, 2020.

[12] Lizotte D, Wang T, Bowling M, Schuurmans D. Automatic gait optimization with Gaussian process regression. In: Proc. of the 20th Int'l Joint Conf. on Artifical Intelligence. Hyderabad: ACM, 2007. 944–949.

[13] Calandra R, Seyfarth A, Peters J, Deisenroth MP. Bayesian optimization for learning gaits under uncertainty: An experimental comparison on a dynamic bipedal walker. Annals of Mathematics and Artificial Intelligence, 2016, 76(1): 5–23. [doi: 10.1007/s10472-015-9463-9]

[14] Jaquier N, Rozo LD, Calinon S, Bürger M. Bayesian optimization meets riemannian manifolds in robot learning. In: Proc. of the 3rd Annual Conf. on Robot Learning. Osaka: PMLR, 2019. 233–246.

[15] Yogatama D, Kong LP, Smith NA. Bayesian optimization of text representations. In: Proc. of the 2015 Conf. on Empirical Methods in Natural Language Processing. Lisbon: Association for Computational Linguistics, 2015. 2100–2105. [doi: 10.18653/v1/D15-1251]

[16] Wilson A, Fern A, Tadepalli P. Using trajectory data to improve Bayesian optimization for reinforcement learning. The Journal of Machine Learning Research, 2014, 15(1): 253–282.

[17] Marco A, Berkenkamp F, Hennig P, Schoellig AP, Krause A, Schaal S, Trimpe S. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization. In: Proc. of the 2017 IEEE Int'l Conf. on Robotics and Automation. Singapore: IEEE, 2017. 1557–1563. [doi: 10.1109/ICRA.2017.7989186]

[18] Frazier PI. A tutorial on Bayesian optimization. arXiv:1807.02811, 2018.

[19] Kandasamy K, Schneider J, Póczos B. High dimensional Bayesian optimisation and bandits via additive models. In: Proc. of the 32nd Int'l Conf. on Machine Learning. Lille: ACM, 2015. 295–304.

[20] Hutter F, Hoos HH, Leyton-Brown K. Automated configuration of mixed integer programming solvers. In: Proc. of the 7th Int'l Conf. on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Bologna: Springer, 2010. 186–202. [doi: 10.1007/978-3-642-13520-0_23]

[21] Bergstra J, Yamins D, Cox DD. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: Proc. of the 30th Int'l Conf. on Machine Learning. Atlanta: ACM, 2013. 115–123.

[22] González J, Longworth J, James DC, Lawrence ND. Bayesian optimization for synthetic gene design. arXiv:1505.01627, 2015.
[23] Wang ZY, Hutter F, Zoghi M, Matheson D, De Feitas N. Bayesian optimization in a billion dimensions via random embeddings. Journal of Artificial Intelligence Research, 2016, 55: 361–387. [doi: 10.1613/jair.4806]
[24] Eriksson D, Pearce M, Gardner JR, Turner R, Poloczek M. Scalable global optimization via local Bayesian optimization. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: ACM, 2019. 493.
[25] Shahriari B, Swersky K, Wang ZY, Adams RP, De Freitas N. Taking the human out of the loop: A review of Bayesian optimization. Proc. of the IEEE, 2016, 104(1): 148–175. [doi: 10.1109/JPROC.2015.2494218]
[26] Rasmussen CE, Williams CKI. Gaussian Processes for Machine Learning. Cambridge: MIT Press, 2006.
[27] Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. In: Proc. of the 24th Int'l Conf. on Neural Information Processing Systems. Granada: ACM, 2011. 2546–2554.
[28] Watanabe S, Hutter F. c-TPE: Tree-structured Parzen estimator with inequality constraints for expensive hyperparameter optimization. In: Proc. of the 32nd Int'l Joint Conf. on Artificial Intelligence. Macao: ACM, 2023. 486. [doi: 10.24963/ijcai.2023/486]
[29] Snoek J, Rippel O, Swersky K, Kiros R, Satish N, Sundaram N, Patwary MMA, Prabhat P, Adams RP. Scalable Bayesian optimization using deep neural networks. In: Proc. of the 32nd Int'l Conf. on Machine Learning. Lille: ACM, 2015. 2171–2180.
[30] Springenberg JT, Klein A, Falkner S, Hutter F. Bayesian optimization with robust Bayesian neural networks. In: Proc. of the 30th Int'l Conf. on Neural Information Processing Systems. Barcelona: ACM, 2016. 4141–4149.
[31] Wilson AG, Hu ZT, Salakhutdinov R, Xing EP. Deep kernel learning. In: Proc. of the 19th Int'l Conf. on Artificial Intelligence and Statistics. Cadiz: JMLR, 2016. 370–378.
[32] Lu DJ, Zhang H. Random Process and Its Application. 2nd ed., Beijing: Tsinghua University Press, 2012 (in Chinese).
[33] Snelson EL, Ghahramani Z. Sparse Gaussian processes using pseudo-inputs. In: Proc. of the 18th Int'l Conf. on Neural Information Processing Systems. Vancouver: ACM, 2005. 1257–1264.
[34] Titsias MK. Variational learning of inducing variables in sparse Gaussian processes. In: Proc. of the 12th Int'l Conf. on Artificial Intelligence and Statistics. Clearwater Beach: JMLR, 2009. 567–574.
[35] Burt DR, Rasmussen CE, van der Wilk M. Convergence of sparse variational inference in Gaussian processes regression. The Journal of Machine Learning Research, 2020, 21(1): 131.
[36] Saatçi Y. Scalable Inference for Structured Gaussian Process Models. Cambridge: University of Cambridge, 2011.
[37] Snoek J, Swersky K, Zemel RS, Adams RP. Input warping for Bayesian optimization of non-stationary functions. In: Proc. of the 31st Int'l Conf. on Machine Learning. Beijing: ACM, 2014. 1674–1682.
[38] Oh C, Gavves E, Welling M. BOCK: Bayesian optimization with cylindrical kernels. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 3865–3874.
[39] Bull AD. Convergence rates of efficient global optimization algorithms. The Journal of Machine Learning Research, 2011, 12: 2879–2904.
[40] Berkenkamp F, Schoellig AP, Krause A. No-regret Bayesian optimization with unknown hyperparameters. The Journal of Machine Learning Research, 2019, 20(1): 1868–1891.
[41] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. Journal of Global Optimization, 1998, 13(4): 455–492. [doi: 10.1023/A:1008306431147]
[42] Viana FAC, Haftka RT. Surrogate-based optimization with parallel simulations using the probability of improvement. In: Proc. of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conf. Fort Worth: AIAA, 2010. 9392. [doi: 10.2514/6.2010-9392]
[43] Srinivas N, Krause A, Kakade SM, Seeger MW. Gaussian process optimization in the bandit setting: No regret and experimental design. In: Proc. of the 27th Int'l Conf. on Machine Learning. Haifa: ACM, 2010. 1015–1022.
[44] Györfi L, Kohler M, Krzyzak A, Walk H. A Distribution-free Theory of Nonparametric Regression. New York: Springer, 2002. [doi: 10.1007/b97848]
[45] Jones DR, Perttunen CD, Stuckman BE. Lipschitzian optimization without the Lipschitz constant. Journal of Optimization Theory and Applications, 1993, 79(1): 157–181. [doi: 10.1007/BF00941892]
[46] Li C, Gupta S, Rana S, Nguyen V, Venkatesh S, Shilton A. High dimensional Bayesian optimization using dropout. In: Proc. of the 26th Int'l Joint Conf. on Artificial Intelligence. Melbourne: ACM, 2017. 2096–2102.
[47] Rana S, Li C, Gupta S, Nguyen V, Venkatesh S. High dimensional Bayesian optimization with elastic Gaussian process. In: Proc. of the 34th Int'l Conf. on Machine Learning. Sydney: ACM, 2017. 2883–2891.
[48] Siivola E, Paleyes A, González J, Vehtari A. Good practices for Bayesian optimization of high dimensional structured spaces. Applied AI Letters, 2021, 2(2): e24. [doi: 10.1002/ail2.24]

[49] Qian H, Hu YQ, Yu Y. Derivative-free optimization of high-dimensional non-convex functions by sequential random embeddings. In: Proc. of the 25th Int'l Joint Conf. on Artificial Intelligence. New York: ACM, 2016. 1946–1952.

[50] Woodruff DP. Sketching as a tool for numerical linear algebra. Foundations and Trends® in Theoretical Computer Science, 2014, 10(1–2): 1–157. [doi: 10.1561/0400000060]

[51] Nayebi A, Munteanu A, Poloczek M. A framework for Bayesian optimization in embedded subspaces. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 4752–4761.

[52] Letham B, Calandra R, Rai A, Bakshy E. Re-examining linear embeddings for high-dimensional Bayesian optimization. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: ACM, 2020. 131.

[53] Papenmeier L, Nardi L, Poloczek M. Increasing the scope as you learn: Adaptive Bayesian optimization in nested subspaces. In: Proc. of the 36th Int'l Conf. on Neural Information Processing Systems. New Orleans: ACM, 2022. 842.

[54] Binois M, Ginsbourger D, Roustant O. On the choice of the low-dimensional domain for global optimization via random embeddings. Journal of Global Optimization, 2020, 76(1): 69–90. [doi: 10.1007/s10898-019-00839-1]

[55] Binois M, Ginsbourger D, Roustant O. A warped kernel improving robustness in Bayesian optimization via random embeddings. In: Proc. of the 9th Int'l Conf. on Learning and Intelligent Optimization. Lille: Springer, 2015. 281–286. [doi: 10.1007/978-3-319-19084-6_28]

[56] Shen YH, Kingsford C. Computationally efficient high-dimensional Bayesian optimization via variable selection. In: Proc. of the 2nd Int'l Conf. on Automated Machine Learning. Potsdam: PMLR, 2023. 15.

[57] Hansen N. The CMA evolution strategy: A tutorial. arXiv:1604.00772, 2016.

[58] Garnett R, Osborne MA, Hennig P. Active learning of linear embeddings for Gaussian processes. In: Proc. of the 30th Conf. on Uncertainty in Artificial Intelligence. Quebec City: ACM, 2014. 230–239.

[59] Djolonga J, Krause A, Cevher V. High-dimensional Gaussian process bandits. In: Proc. of the 26th Int'l Conf. on Neural Information Processing Systems. Lake Tahoe: ACM, 2013. 1025–1033.

[60] Zhang M, Li HQ, Su SW. High dimensional Bayesian optimization via supervised dimension reduction. In: Proc. of the 18th Int'l Joint Conf. on Artificial Intelligence. Macao: IJCAI, 2019. 4292–4298.

[61] Li KC. Sliced inverse regression for dimension reduction. Journal of the American Statistical Association, 1991, 86(414): 316–327. [doi: 10.1080/01621459.1991.10475035]

[62] Moriconi R, Deisenroth MP, Kumar KSS. High-dimensional Bayesian optimization using low-dimensional feature spaces. Machine Learning, 2020, 109(9–10): 1925–1943. [doi: 10.1007/s10994-020-05899-z]

[63] Swersky K, Snoek J, Adams RP. Multi-task Bayesian optimization. In: Proc. of the 26th Int'l Conf. on Neural Information Processing Systems. Lake Tahoe: ACM, 2013. 2004–2012.

[64] Kajino H. Molecular hypergraph grammar with its application to molecular optimization. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 3183–3191.

[65] Griffiths RR, Hernández-Lobato JM. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. Chemical Science, 2020, 11(2): 577–586. [doi: 10.1039/c9sc04026a]

[66] Kusner MJ, Paige B, Hernández-Lobato JM. Grammar variational autoencoder. In: Proc. of the 34th Int'l Conf. on Machine Learning. Sydney: ACM, 2017. 1945–1954.

[67] Dai HJ, Tian YT, Dai B, Skiena S, Song L. Syntax-directed variational autoencoder for structured data. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: ICLR, 2018.

[68] Jin WG, Barzilay R, Jaakkola TS. Junction tree variational autoencoder for molecular graph generation. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 2328–2337.

[69] Mahmood O, Hernández-Lobato JM. A COLD approach to generating optimal samples. arXiv:1905.09885, 2019.

[70] Daxberger E, Hernández-Lobato JM. Bayesian variational autoencoders for unsupervised out-of-distribution detection. arXiv: 1912.05651, 2019.

[71] Eismann S, Levy D, Shu R, Bartzsch S, Ermon S. Bayesian optimization and attribute adjustment. In: Proc. of the 34th Conf. on Uncertainty in Artificial Intelligence. Monterey: AUAI Press, 2018. 1042–1052.

[72] Tripp A, Daxberger E, Hernández-Lobato JM. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: ACM, 2020. 945.

[73] Kirschner J, Mutny M, Hiller N, Ischebeck R, Krause A. Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 3429–3438.

[74] Eriksson D, Jankowiak M. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In: Proc. of the 37th Conf. on

Uncertainty in Artificial Intelligence. AUAI Press, 2021. 493–503.

[75] Gardner JR, Guo C, Weinberger KQ, Garnett R, Grosse RB. Discovering and exploiting additive structure for Bayesian optimization. In: Proc. of the 20th Int'l Conf. on Artificial Intelligence and Statistics. Fort Lauderdale: PMLR, 2017. 1311–1319.

[76] Mao SS, Tang YC. Bayesian Statistics. 2nd ed., Beijing: China Statistics Press, 2012 (in Chinese).

[77] Wang Z, Li CT, Jegelka S, Kohli P. Batched high-dimensional Bayesian optimization via structural kernel learning. In: Proc. of the 34th Int'l Conf. on Machine Learning. Sydney: ACM, 2017. 3656–3664.

[78] Rolland P, Scarlett J, Bogunovic I, Cevher V. High-dimensional Bayesian optimization via additive models with overlapping groups. In: Proc. of the 21st Int'l Conf. on Artificial Intelligence and Statistics. Playa Blanca: PMLR, 2018. 298–307.

[79] Han E, Arora I, Scarlett J. High-dimensional Bayesian optimization via tree-structured additive models. In: Proc. of the 35th AAAI Conf. on Artificial Intelligence. AAAI Press, 2021. 7630–7638. [doi: 10.1609/aaai.v35i9.16933]

[80] Ziomek JK, Bou-Ammar H. Are random decompositions all we need in high dimensional Bayesian optimisation? In: Proc. of the 40th Int'l Conf. on Machine Learning. Honolulu: ACM, 2023. 1825.

[81] Li CL, Kandasamy K, Póczos B, Schneider JG. High dimensional Bayesian optimization via restricted projection pursuit models. In: Proc. of the 19th Int'l Conf. on Artificial Intelligence and Statistics. Cadiz: JMLR, 2016. 884–892.

[82] Gilboa E, Saatçi Y, Cunningham JP. Scaling multidimensional Gaussian processes using projected additive approximations. In: Proc. of the 30th Int'l Conf. on Machine Learning. Atlanta: ACM, 2013. I-454–I-461.

[83] Wan XC, Nguyen V, Ha H, Ru BX, Lu C, Osborne MA. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. In: Proc. of the 38th Int'l Conf. on Machine Learning. PMLR, 2021. 10663–10674.

[84] Munos R. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In: Proc. of the 24th Int'l Conf. on Neural Information Processing Systems. Granada: ACM, 2011. 783–791.

[85] Wang ZY, Shakibi B, Jin L, De Freitas N. Bayesian multi-scale optimistic optimization. In: Proc. of the 17th Int'l Conf. on Artificial Intelligence and Statistics. Reykjavik: JMLR, 2014. 1005–1014.

[86] Kawaguchi K, Kaelbling LP, Lozano-Pérez T. Bayesian optimization with exponential convergence. In: Proc. of the 28th Int'l Conf. on Neural Information Processing Systems. Montreal: ACM, 2015. 2809–2817.

[87] Kim B, Lee K, Lim S, Kaelbling L, Lozano-Pérez T. Monte Carlo tree search in continuous spaces using Voronoi optimistic optimization with regret bounds. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence. New York: AAAI Press, 2020. 9916–9924. [doi: 10.1609/aaai.v34i06.6546]

[88] Wang LN, Fonseca R, Tian YD. Learning search space partition for black-box optimization using Monte Carlo tree search. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: ACM, 2020. 1637.

[89] Slivkins A. Introduction to multi-armed bandits. Foundations and Trends® in Machine Learning, 2019, 12(1–2): 1–286. [doi: 10.1561/2200000068]

[90] Wang Z, Gehring C, Kohli P, Jegelka S. Batched large-scale Bayesian optimization in high-dimensional spaces. In: Proc. of the 21st Int'l Conf. on Artificial Intelligence and Statistics. Playa Blanca: PMLR, 2018. 745–754.

[91] Müller S, Von Rohr A, Trimpe S. Local policy search with Bayesian optimization. In: Proc. of the 35th Int'l Conf. on Neural Information Processing Systems. ACM, 2021. 1584.

[92] Nguyen Q, Wu KW, Gardner JR, Garnett R. Local Bayesian optimization via maximizing probability of descent. In: Proc. of the 36th Int'l Conf. on Neural Information Processing Systems. New Orleans: ACM, 2022. 958.

[93] Fröhlich LP, Zeilinger MN, Klenske ED. Cautious Bayesian optimization for efficient and scalable policy search. In: Proc. of the 3rd Annual Conf. on Learning for Dynamics and Control. PMLR, 2021. 227–240.

[94] Maher S, Miltenberger M, Pedroso JP, Rehfeldt D, Schwarz R, Serrano F. PYSCIPOPT: Mathematical programming in Python with the SCIP optimization suite. In: Proc. of the 5th Int'l Conf. on Mathematical Software. Berlin: Springer, 2016. 301–307. [doi: 10.1007/978-3-319-42432-3_37]

[95] Salem MB, Bachoc F, Roustant O, Gamboa F, Tomaso L. Gaussian process-based dimension reduction for goal-oriented sequential design. SIAM/ASA Journal on Uncertainty Quantification, 2019, 7(4): 1369–1397. [doi: 10.1137/18M1167930]

[96] Spagnol A, Riche RL, Veiga SD. Global sensitivity analysis for optimization with variable selection. SIAM/ASA Journal on Uncertainty Quantification, 2019, 7(2): 417–443. [doi: 10.1137/18M1167978]

[97] Sehic K, Gramfort A, Salmon J, Nardi L. LassoBench: A high-dimensional hyperparameter optimization benchmark suite for Lasso. In: Proc. of the 2022 Int'l Conf. on Automated Machine Learning. Baltimore: PMLR, 2022. 2/1–24.

[98] Ulmasov D, Baroukh C, Chachuat B, Deisenroth MP, Misener R. Bayesian optimization with dimension scheduling: Application to biological systems. Computer Aided Chemical Engineering, 2016, 38: 1051–1056. [doi: 10.1016/B978-0-444-63428-3.50180-6]

[99] Tu S, Recht B. Least-squares temporal difference learning for the linear quadratic regulator. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 5012–5021.

[100] Mania H, Guy A, Recht B. Simple random search of static linear policies is competitive for reinforcement learning. In: Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems. Montréal: ACM, 2018. 1805–1814.

[101] Marco A, Hennig P, Bohg J, Schaal S, Trimpe S. Automatic LQR tuning based on Gaussian process global optimization. In: Proc. of the 2016 IEEE Int'l Conf. on Robotics and Automation. Stockholm: IEEE, 2016. 270–277. [doi: 10.1109/ICRA.2016.7487144]

[102] Balandat M, Karrer B, Jiang DR, Daulton S, Letham B, Wilson AG, Bakshy E. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: ACM, 2020. 1807.

[103] Cowen-Rivers AI, Lyu WL, Tutunov R, Wang Z, Grosnit A, Griffiths RR, Maraval AM, Hao JY, Wang J, Peters J, Bou-Ammar H. HEBO: Pushing the limits of sample-efficient hyper-parameter optimisation. Journal of Artificial Intelligence Research, 2022, 74: 1269–1349. [doi: 10.1613/jair.1.13643]

[104] Grosnit A, Cowen-Rivers AI, Tutunov R, Griffiths RR, Wang J, Bou-Ammar H. Are we forgetting about compositional optimisers in Bayesian optimisation? The Journal of Machine Learning Research, 2021, 22(1): 160.

[105] Daulton S, Eriksson D, Balandat M, Bakshy E. Multi-objective Bayesian optimization over high-dimensional search spaces. In: Proc. of the 38th Conf. on Uncertainty in Artificial Intelligence. Eindhoven: PMLR, 2022. 507–517.

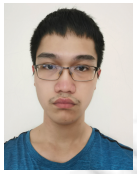
[106] Eriksson D, Chuang PIJ, Daulton S, Xia P, Shrivastava A, Babu A, Zhao SC, Aly A, Venkatesh G, Balandat M. Latency-aware neural architecture search with multi-objective Bayesian optimization. arXiv:2106.11890, 2021.

[107] Zhao YY, Wang LN, Yang K, Zhang TJ, Guo T, Tian YD. Multi-objective optimization by learning space partitions. arXiv:2110.03173, 2021.

附中文参考文献:

[32] 陆大絟, 张颢. 随机过程及其应用. 第 2 版, 北京: 清华大学出版社, 2012.

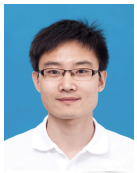
[76] 茆诗松, 汤银才. 贝叶斯统计. 第 2 版, 北京: 中国统计出版社, 2012.



陈泉霖(1998－), 男, 博士生, 主要研究领域为贝叶斯优化.

高阳(1972－), 男, 博士, 教授, CCF 会士, 主要研究领域为人工智能, 机器学习, 智能系统.

陈奕宇(1998－), 男, 博士生, CCF 学生会员, 主要研究领域为元强化学习, 机器人控制.



李栋(1992－), 男, 博士, 主要研究领域为强化学习, 贝叶斯优化.

霍静(1989－), 女, 博士, 副教授, CCF 专业会员, 主要研究领域为机器学习, 计算机视觉, 具身智能.

郝建业(1986－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为深度强化学习, 多智能体系统.

曹宏业(1998－), 男, 博士生, 主要研究领域为强化学习, 机器学习.