Abstract

贝叶斯优化是一种优化目标函数的方法，需要很长时间（几分钟或几小时）才能评估。它最适合在少于 20 维的连续域上进行优化，并且可以容忍函数评估中的随机噪声。它为目标构建一个替代，并使用贝叶斯机器学习技术高斯过程回归量化该替代中的不确定性，然后使用从该替代定义的采集函数来决定在哪里采样。在本教程中，我们描述了贝叶斯优化的工作原理，包括高斯过程回归和三个常见的采集函数：预期改进、熵搜索和知识梯度。然后，我们讨论更高级的技术，包括并行运行多个函数评估、多保真度和多信息源优化、评估成本高昂的约束、随机环境条件、多任务贝叶斯优化以及衍生信息的纳入。最后，我们讨论了贝叶斯优化软件和该领域未来的研究方向。在我们的教程材料中，我们提供了对噪声评估的预期改进的概括，超出了更常应用的无噪声设置。这种概括是通过正式的决策理论论证来证明的，与以前的临时修改形成鲜明对比。

1 Introduction

贝叶斯优化（Bayesian optimization, BayesOpt）是一类专注于解决以下问题的优化方法：

$$\max_{x \in A} f(x),$$

其中可行集和目标函数通常具有以下特性：

- 输入 x 属于 $\mathbb{R}^d$，并且 d 的值不应过大。通常在大多数成功的 BayesOpt 应用中，$d \leq 20$。
- 可行集 A 是一个简单的集合，易于评估其成员资格。通常，A 是一个超矩形 $\{x \in \mathbb{R}^d : a_i \leq x_i \leq b_i\}$ 或 d-维单纯形 $\{x \in \mathbb{R}^d : \sum_i x_i = 1\}$。稍后（第 5 节）我们将放宽这一假设。
- 目标函数 f 是连续的。这通常是将 f 建模为高斯过程回归所必需的。
- f 的评估“代价高昂”，即可进行的评估次数有限，通常限制在几百次。这一限制通常是因为每次评估需要大量时间（通常为几个小时），但也可能是因为每次评估涉及经济成本（例如，购买云计算资源或实验材料），或者机会成本（例如，如果评估 f 需要询问人类受试者问题，而受试者只能容忍有限次数）。
- f 缺乏已知的特殊结构，如凹性或线性，这使得使用利用此类结构的技术提高效率变得容易。我们将此总结为 f 是一个“黑箱”。
- 当我们评估 f 时，仅观察到 $f(x)$，而没有一阶或二阶导数。这阻止了使用一阶和二阶方法（如梯度下降、牛顿法或拟牛顿法）。我们将具有此特性的问问题称为“无导数”。
- 在文章的大部分内容中，我们将假设 $f(x)$ 是无噪声观察到的。稍后（第 5 节）我们将允许 $f(x)$ 被随机噪声遮蔽。在几乎所有的贝叶斯优化工作中，噪声被假定在评估之间是独立的，并且是具有常量方差的高斯噪声。

我们的重点是寻找全*局*最优解而非*局*部最优解。

我们总结这些问题特征，称 BayesOpt 旨在进行黑箱无导数全局优化。

优化代价高昂的黑箱无导数函数使得 BayesOpt 极具灵活性。最近，它在机器学习算法的超参数调优方面变得非常流行，尤其是在深度神经网络中 (**?**)。在更长的时间跨度内，自 1960 年代以来，BayesOpt 在工程系统设计中得到了广泛应用 (**???**)。BayesOpt 还被用来选择材料和药物设计中的实验 (**???**)，用于环境模型的校准 (**?**)，以及强化学习中 (**???**)。

BayesOpt 起源于 Kushner 的研究 (**?**)，Zilinskas (**??**) 和 Mockus (**??**) 的工作，但在 Jones 等人 (**?**) 将其工作普及后，受到了更广泛的关注，并提出了有效全局优化（EGO）算法。跟随 Jones 等人的工作 (**?**)，该文献中发展的创新包括多保真度优化 (**??**)，多目标优化 (**???**)，以及收敛率的研究 (**????**)。Snoek 等人 (**?**) 观察到 BayesOpt 对

训练深度神经网络的有用性，引发了机器学习领域的关注，相关文献中的补充创新包括多任务优化 (**?**)，专门针对训练深度神经网络的多保真度优化 (**?**)，以及并行方法 (**???**)。高斯过程回归及其密切相关的克里金方法，近年来也在仿真文献中得到了研究 (**???**)，用于模拟和优化使用离散事件模拟的系统。

在 BayesOpt 之外，还有其他技术可用于优化代价高昂的无导数黑箱函数。虽然我们在此不详细回顾这方面文献中的方法，但其中许多方法与 BayesOpt 方法的风格相似：它们维护一个代理模型来建模目标函数，并使用该代理来选择评估的位置 (**????**)。这类更一般的方法通常被称为“代理方法”。贝叶斯优化通过使用基于贝叶斯统计的代理模型，并通过贝叶斯解释来决定何处评估目标，从而区别于其他代理方法。

我们首先在第 2 节介绍贝叶斯优化算法的典型形式。该形式涉及两个主要组件：统计推断方法，通常是高斯过程（GP）回归；以及用于决定下一个采样位置的采集函数，通常是期望改善。我们将在第 3 节和第 4.1 节详细描述这两个组件。接下来，我们将描述三种替代采集函数：知识梯度（第 4.2 节）、熵搜索和预测熵搜索（第 4.3 节）。这些替代采集函数在我们称之为“奇异”贝叶斯优化问题的情况下特别有用，这将在第 5 节中讨论。这些奇异贝叶斯优化问题包括具有并行评估、约束、多保真度评估、多信息源、随机环境条件、多任务目标和导数观测的问题。然后，我们将在第 6 节讨论贝叶斯优化和高斯过程回归软件，并在第 7 节总结未来的研究方向。

关于贝叶斯优化的其他教程和综述包括 Shahriari 等（2016）；Brochu 等（2009）；Sasena（2002）；Frazier 和 Wang（2016）。本教程与其他教程的不同之处在于其对非标准或“奇异”贝叶斯优化问题的覆盖。它还强调了采集函数的重要性，而较少强调 GP 回归。最后，它包括我们认为是对带噪声测量的期望改善的新颖分析，并认为 Scott 等（2011）提出的采集函数是应用期望改善采集函数时的最自然方法。

2 BayesOpt 概述

贝叶斯优化由两个主要组成部分构成：用于建模目标函数的贝叶斯统计模型，以及用于决定下一个采样位置的采集函数。在对初始空间填充实验设计（通常由在可行空间中均匀或随机选择的点组成）评估目标函数后，贝叶斯优化在两个步骤之间迭代。第一步拟合模型，该模型使用迄今观察到的函数值来估计目标函数 $f(x)$。第二步使用采集函数选择下一个要评估的点，预计在当前模型的不确定性下最大限度地提高目标。

Algorithm 1 贝叶斯优化的基本伪代码

在 f 上放置高斯过程先验
根据初始空间填充实验设计观察 f 在 n_0 个点的值。设置 $n = n_0$。
while $n \leq N$ **do**
    使用所有可用数据更新 f 的后验概率分布
    令 x_n 为在 x 上最大化采集函数的点，其中采集函数是根据当前的后验分布计算的。
    观察 $y_n = f(x_n)$。
    增加 n
end while
结束 while
返回解决方案：要么是评估过的最大 $f(x)$ 的点，要么是具有最大后验均值的点。

统计模型，通常是高斯过程，提供了一个贝叶斯后验概率分布，用于描述候选点 x 处的潜在值 $f(x)$。每次我们在新点上观察 f 时，这个后验分布会被更新。我们将在第 3 节详细讨论使用高斯过程的贝叶斯统计建模。采集函数衡量在新点 x 上评估目标函数的价值，该价值基于当前对 f 的后验分布。我们将在第 4.1 节讨论期望改善，这是最常用的采集函数，然后在第 4.2 节和第 4.3 节讨论其他采集函数。

算法 1 的一次 BayesOpt 迭代使用 GP 回归和期望改善的示意图如图 1 所示。上面面板显示了具有噪声的目标函数的观察，蓝色圆圈表示三个点。它还显示了 GP 回归的输出。我们将在第 3 节看到，GP 回归在每个 $f(x)$ 上产生的后验概率分布是正态分布，均值为 $\mu_n(x)$，方差为 $\sigma_n^2(x)$。图中用红色实线表示均值 $\mu_n(x)$，用红色虚线表示目标函数 $f(x)$ 的 95% 贝叶斯可信区间，即 $\mu_n(x) \pm 1.96 \times \sigma_n(x)$。均值可以解释为 $f(x)$ 的点估计。可信区间在频率统计中类似于置信区间，并且根据后验分布以 95% 的概率包含 $f(x)$。均值在先前评估的点之间插值。在这些点处，可信区间的宽度为 0，随着距离的增加而变宽。

下方面板显示了与该后验相对应的期望改善采集函数。请注意，在之前评估的点上，其值为 0。这是合理的，因为在没有噪声的情况下评估这些点不会提供任何有用的信息来解决 (1)。还要注意，对于具有较大可信区间的点，它的值往往较大，因为在我们对目标函数的不确定性较大的情况下，评估这些点通常更有助于找到良好的近似全局最优解。还要注意，对于具有较大后验均值的点，它的值往往也较大，因为这些点通常接近良好的近似全局最优解。

接下来，我们将详细讨论 BayesOpt 的各个组成部分，首先在第 3 节讨论 GP 回归，然后在第 4 节讨论采集函数，首先讨论期望改善（第 4.1 节）。然后在第 4.2 节和第 4.3 节讨论更复杂的采集函数（知识梯度、熵搜索和预测熵搜索）。最后，我们将在第 5 节讨论第 1 节中描述的基本问题的扩展，讨论带测量噪声、并行函数评估、约束、多保真度观察和其他问题。

3 高斯过程回归 Gaussian Process Regression

高斯过程回归是一种用于建模函数的贝叶斯统计方法。我们在此提供简要介绍。更全面的处理可参考 Rasmussen 和 Williams（2006）。

我们首先描述高斯过程回归，重点关注在有限的一组点 $x_1, \ldots, x_k \in \mathbb{R}^d$ 上的 f。方便起见，我们将这些点上的函数值收集到一个向量中 $[f(x_1), \ldots, f(x_k)]$。每当我们有一个在贝叶斯统计中未知的量（如这个向量）时，我们假设它是由自然随机抽取的，来自某个先验概率分布。高斯过程回归采取这一先验分布，并通过观察 $f(x)$ 来更新它。

我们通过在每个 x_i 处评估一个 *均值函数* μ_0 来构建均值向量。通过在每对点 x_i, x_j 处评估 *协方差函数或核* Σ_0 来构建协方差矩阵。选择的核应使得在输入空间中距离较近的点 x_i, x_j 具有较大的正相关性，从而编码信念：它们的函数值应比远离的点更相似。核还应具备的性质是，无论选择哪些点，得到的协方差矩阵都是半正定的。下面在第 3.1 节讨论了一些示例均值函数和核。

在 $[f(x_1), \ldots, f(x_k)]$ 上得到的先验分布为

$$f(x_{1:k}) \sim \text{Normal}(\mu_0(x_{1:k}), \Sigma_0(x_{1:k}, x_{1:k})),$$

其中我们对应用于输入点集合的函数使用紧凑符号：$x_{1:k}$ 表示序列 $x_1, \ldots, x_k$，$f(x_{1:k}) = [f(x_1), \ldots, f(x_k)]$，$\mu_0(x_{1:k}) = [\mu_0(x_1), \ldots, \mu_0(x_k)]$，以及 $\Sigma_0(x_{1:k}, x_{1:k}) = [\Sigma_0(x_1, x_1), \ldots, \Sigma_0(x_{1:k}, x_{1:k})]$。

假设我们在某些 n 个点上观察 $f(x_{1:n})$ 没有噪声，我们希望推断某个新点 x 处的 $f(x)$ 的值。为此，我们令 $k = n + 1$ 并设置 $x_k = x$，因此在 $[f(x_{1:n}), f(x)]$ 上的先验为 (2)。我们可以使用贝叶斯法则计算给定这些观察的 $f(x)$ 的条件分布（见 Rasmussen 和 Williams（2006）第 2.1 章的详细信息）：

$$f(x) | f(x_{1:n}) \sim \text{Normal}(\mu_n(x), \sigma_n^2(x))$$

$$\mu_n(x) = \Sigma_0(x, x_{1:n}) \Sigma_0(x_{1:n}, x_{1:n})^{-1} (f(x_{1:n}) - \mu_0(x_{1:n})) + \mu_0(x)$$

$$\sigma_n^2(x) = \Sigma_0(x, x) - \Sigma_0(x, x_{1:n}) \Sigma_0(x_{1:n}, x_{1:n})^{-1} \Sigma_0(x_{1:n}, x).$$

这个条件分布在贝叶斯优化的术语中称为*后验概率分布*。

贝叶斯统计。后验均值 $\mu_n(x)$ 是先验 $\mu_0(x)$ 和基于数据 $f(x_{1:n})$ 的估计之间的加权平均，权重取决于核函数。后验方差 $\sigma_n^2(x)$ 等于先前的协方差 $\Sigma_0(x, x)$ 减去一个与观察 $f(x_{1:n})$ 相关的项。

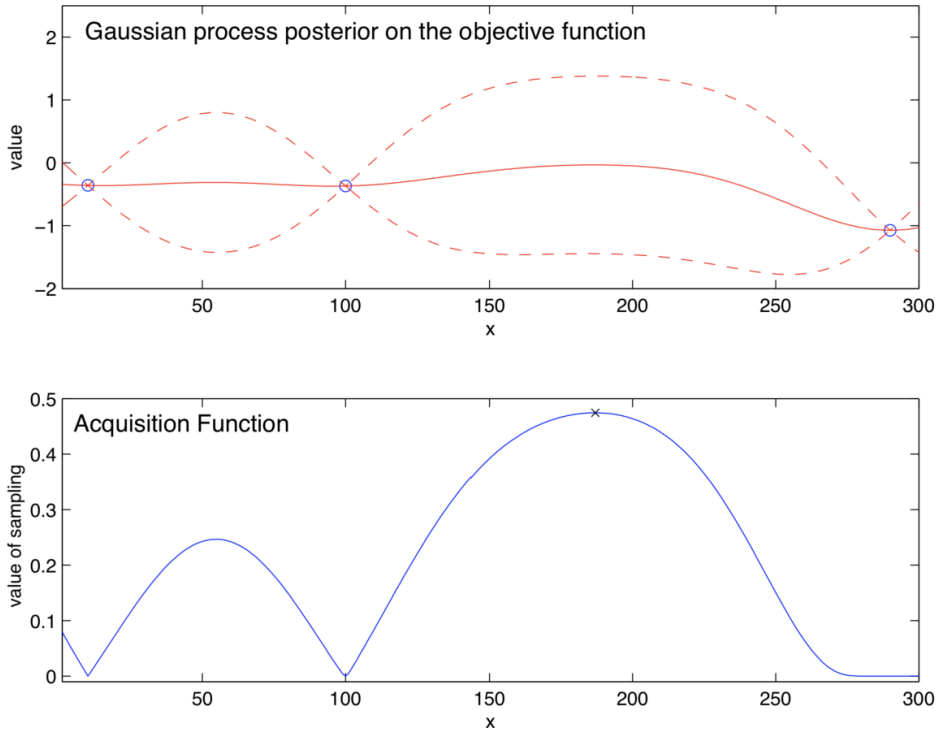

图 1: BayesOpt 的示意图，最大化目标函数 f 的 1 维连续输入。上面面板显示：在 3 个点上的无噪声目标函数 f 的观察（蓝色），对 $f(x)$ 的估计（红色实线），以及 $f(x)$ 的贝叶斯可信区间（类似于置信区间，红色虚线）。这些估计和可信区间是通过高斯过程回归获得的。下面的面板显示了采集函数。贝叶斯优化选择在最大化采集函数的点进行下一个采样，此点用“x”表示。

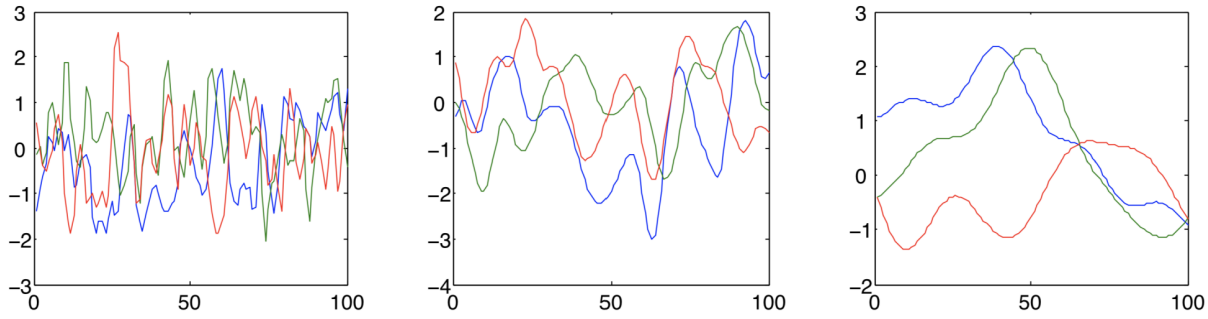

图 2: 从具有幂指数核的高斯过程先验中绘制的随机函数 f。每个图对应于参数 α_1 的不同值，α_1 从左到右逐渐减小。改变这个参数会导致对 $f(x)$ 随着 x 变化的速率的不同信念。

通常，通过（3）和矩阵求逆直接计算后验均值和方差并不是最快或最具数值稳定性的方法。使用 Cholesky 分解，然后解决线性方程组的技术通常更快且更稳定。这种更复杂的技术在 Rasmussen 和 Williams（2006）第 2.2 节中的算法 2.1 中有讨论。此外，为了提高该方法或直接使用（3）时的数值稳定性，通常会在 $\Sigma_0(x_{1:n}, x_{1:n})$ 的对角线的每个元素中添加一个小的正数，例如 10^{-6}，尤其是当 $x_{1:n}$ 包含两个或更多相近的点时。这可以防止 $\Sigma_0(x_{1:n}, x_{1:n})$ 的特征值过于接近 0，并且只会对无限精度计算所做的预测产生微小的变化。

尽管我们仅在有限数量的点上对 f 建模，但在建模 f 的连续域 A 时也可以使用相同的方法。正式地，一个高斯过程具有均值函数 μ_0 和核 Σ_0，是一个函数 f 的概率分布，其特性是对于任何给定的点集合 $x_{1:k}$，$f(x_{1:k})$ 的边际概率分布由（2）给出。此外，当我们的先验概率分布是高斯过程时，用于证明（3）的论点仍然有效。

除了计算给定 $f(x_{1:n})$ 的 $f(x)$ 的条件分布之外，还可以计算在多个未评估点的 f 的条件分布。得到的分布是多元正态分布，均值向量和协方差核依赖于未评估点的位置、已测量点 $x_{1:n}$ 的位置以及它们的测量值 $f(x_{1:n})$。给出该均值向量和协方差矩阵的条目具有上述均值函数和核所需的形式，给定 $f(x_{1:n})$ 的 f 的条件分布是具有该均值函数和协方差核的高斯过程。

3.1 选择均值函数和核

现在我们讨论核的选择。核通常具有这样的特性，即在输入空间中距离较近的点的相关性更强，即如果 $\|x-x'\| < \|x-x''\|$ 对于某种范数 $\|\cdot\|$，那么 $\Sigma_0(x,x') > \Sigma_0(x,x'')$。此外，核被要求是半正定函数。这里我们描述两个示例核及其用法。

一个常用且简单的核是*幂指数*或*高斯核*，

$$\Sigma_0(x,x') = \alpha_0 \exp\left(-\|x-x'\|^2\right),$$

其中 $\|x-x'\|^2 = \sum_{i=1}^{d} \alpha_i (x_i - x_i')^2$，$\alpha_0$ 和 α_i 是核的参数。图 2 显示了从具有幂指数核的高斯过程先验中绘制的随机函数，这些函数具有不同的 α_1 值。改变这个参数会导致对 $f(x)$ 随着 x 变化速率的不同信念。

另一个常用的核是 *Matérn 核*，

$$\Sigma_0(x,x') = \alpha_0 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\|x-x'\|\right) K_\nu \left(\sqrt{2\nu}\|x-x'\|\right)$$

其中 K_ν 是修正的贝塞尔函数，此外我们还有一个参数 ν 除了参数 α_0, d。我们将在第 3.2 节中讨论如何选择这些参数。

或许最常见的均值函数选择是常数值，即 $\mu_0(x) = \mu$。当相信 f 具有某种趋势或特定应用的参数结构时，我们也可以将均值函数取为

$$\mu_0(x) = \mu + \sum_{i=1}^{p} \beta_i \Psi_i(x),$$

其中每个 Ψ_i 是一个参数函数，通常是 x 的低阶多项式。

3.2 选择超参数

均值函数和核包含参数。我们通常称这些先验的参数为*超参数*。我们用向量 η 来表示它们。例如，如果我们使用 Matérn 核和常数均值函数，则 $\eta = (\alpha_0, d, \nu, \mu)$。

要选择超参数，通常考虑三种方法。第一种是寻找*最大似然估计 (MLE)*。在这种方法中，当给定观察值 $f(x_{1:n})$ 时，我们计算这些观察值在先验下的似然 $P(f(x_{1:n})|\eta)$，其中我们修改我们的符号以表示其对 η 的依赖性。该似然是一个多元正态密度。然后，在最大似然估计中，我们将 η 设置为最大化此似然的值，

$$\hat{\eta} = \arg\max_{\eta} P(f(x_{1:n})|\eta).$$

第二种方法在第一种方法的基础上进行修正，假设超参数 η 也是从先验中选择的，即 $P(\eta)$。然后我们通过*后验最大化 (MAP)* 估计（Gelman 等人，2014）来估计 η，即最大化后验的 η 的值，

$$\hat{\eta} = \arg\max_{\eta} P(\eta|f(x_{1:n})) = \arg\max_{\eta} P(f(x_{1:n})|\eta)P(\eta).$$

在从第一个表达式转到第二个时，我们使用了贝叶斯规则，然后去掉了一个不依赖于要优化的量 η 的归一化常数 $\int P(f(x_{1:n})|\eta')P(\eta')\,d\eta'$。

如果我们将超参数的先验 $P(\eta)$ 取为在 η 的定义域内具有常量密度的（可能是退化的）概率分布，则 MLE 是 MAP 的特例。如果 MLE 有时会估计出不合理的超参数值，例如，对应于变化过快或过慢的函数（见图 2），那么 MAP 是有用的。通过选择一个先验，对特定问题更合理的超参数值给予更多权重，MAP 估计可以更好地对应于应用。先验的常见选择包括均匀分布（用于防止估计落在某些预设范围之外）、正态分布（用于建议估计接近某个标称值，而不设置硬截止）、以及对数正态和截断正态分布（用于对正参数提供类似的建议）。

第三种方法称为完全贝叶斯方法。在这种方法中，我们希望计算 $f(x)$ 的后验分布，边际化所有可能的超参数值，

$$P(f(x) = y|f(x_{1:n})) = \int P(f(x) = y|f(x_{1:n}), \eta)P(\eta|f(x_{1:n}))\,d\eta.$$

这个积分通常是不可解的，但我们可以通过抽样来近似它：

$$P(f(x) = y|f(x_{1:n})) \approx \frac{1}{J}\sum_{j=1}^{J} P(f(x) = y|f(x_{1:n}), \eta = \hat{\eta}_j),$$

其中 $\hat{\eta}_j; j = 1, \ldots, J$ 是通过 MCMC 方法（例如切片抽样，Neal, 2003）从 $P(\eta|f(x_{1:n}))$ 中抽样的。MAP 估计可以被看作是完全贝叶斯推断的近似：如果我们将后验 $P(\eta|f(x_{1:n}))$ 近似为在最大化后验密度的 η 处的点质量，那么用 MAP 进行推断可以恢复（5）。

4 获取函数 Acquisition Functions

在回顾了高斯过程之后，我们回到算法 1，讨论在该循环中使用的获取函数。我们首先关注在第 1 节中描述的没有噪声的评估设置，我们称之为“标准”问题，然后在第 5 节中讨论带噪声的评估、并行评估、导数观测和其他“特殊”扩展。

最常用的获取函数是期望改进（Expected Improvement，EI），我们首先讨论它（第 4.1 节）。期望改进性能良好且易于使用。然后我们讨论知识梯度（第 4.2 节）、熵搜索和预测熵搜索（第 4.3 节）获取函数。这些替代获取函数在特殊问题中最有用，其中期望改进的一个假设，即采样的主要好处发生在采样的点上，不再成立。

4.1 期望改进 Expected Improvement

期望改进获取函数通过一个思维实验推导而来。假设我们使用算法 1 来解决问题 (1)，其中 x_n 表示在第 n 次迭代中采样的点，y_n 表示观察值。假设我们只能返回一个已评估的解作为问题 (1) 的最终解。并且假设此刻我们没有剩余的评估可以进行，必须根据我们已经执行的评估返回一个解。由于我们无噪声地观察到 f，最佳选择是之前评估的点中观察值最大的点。设 $f_n^* = \max_{m\le n} f(x_m)$ 为此点的值，其中 n 是我们迄今为止评估 f 的次数。

现在假设我们实际上还有一次额外的评估可以执行，并且可以在任何地方进行。如果我们在 x 处评估，我们将观察到 $f(x)$。在这次评估后，已观察到的最佳点的值将是 $f(x)$（如果 $f(x) \ge f_n^*$）或 f_n^*（如果 $f(x) \le f_n^*$）。然后，最佳观察值的改进为 $f(x) - f_n^*$ 如果该值为正，否则为 0。我们可以更简洁地将这个改进表示为 $[f(x) - f_n^*]^+$，其中 $a^+ = \max(a, 0)$ 表示正部分。

虽然我们希望选择 x 使得这个改进很大，但 $f(x)$ 在评估之前是未知的。然而，我们可以计算这个改进的期望值并选择 x 来最大化它。我们定义期望改进为：

$$\mathrm{EI}_n(x) := \mathbb{E}_n\left[[f(x) - f_n^*]^+\right]$$

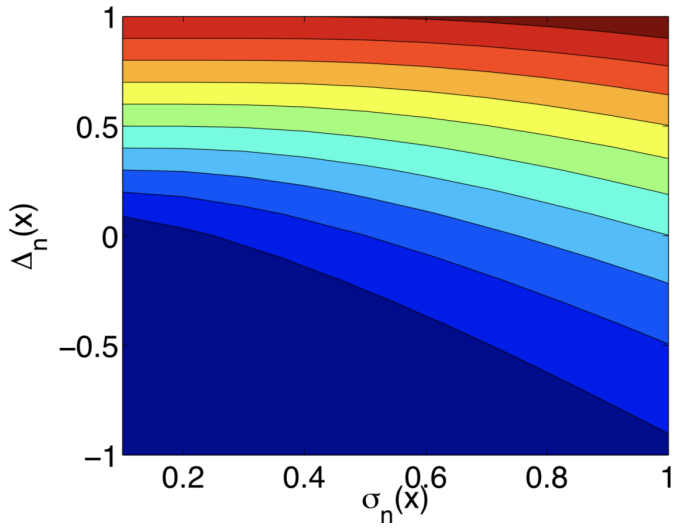

图 3: EI(x) 的等高线图，即预期改进 (8)，就 $\Delta_n(x)$（建议点与之前评估的最佳点之间的预期质量差异）和后验标准差 $\sigma_n(x)$ 而言。蓝色表示较小的值，红色表示较大的值。预期改进在两个数量上都在增加，具有相等 EI 的 $\Delta_n(x)$ 与 $\sigma_n(x)$ 的曲线定义了在具有高预期质量（高 $\Delta_n(x)$）的点处进行评估与在高不确定性（高 $\sigma_n(x)$）的点处进行评估之间的隐式权衡。

这里，$\mathbb{E}_n[\cdot] = \mathbb{E}[\cdot|x_{1:n}, y_{1:n}]$ 表示在给定 f 在 $x_1, \ldots, x_n$ 上的评估的后验分布下的期望。这个后验分布由（3）给出：$f(x)$ 在给定 $x_{1:n}, y_{1:n}$ 时呈正态分布，其均值为 $\mu_n(x)$ 和方差为 $\sigma_n^2(x)$。

期望改进可以通过分部积分以封闭形式计算，正如 Jones 等人（1998）或 Clark（1961）所描述的那样。得到的表达式为：

$$\mathrm{EI}_n(x) = [\Delta_n(x)]^+ + \sigma_n(x)\varphi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right) - |\Delta_n(x)|\Phi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right),$$

其中 $\Delta_n(x) := \mu_n(x) - f_n^*$ 是提议点 x 与先前最佳点的期望质量差异。

期望改进算法然后在具有最大期望改进的点进行评估，

$$x_{n+1} = \arg\max \mathrm{EI}_n(x),$$

任意打破平局。该算法最早由 Mockus 提出（Mockus，1975），但在 Jones 等人（1998）中得到了推广。后者的文章还使用了“有效全局优化”或 EGO 这个名称。

实现中使用了多种方法来解决（9）。与我们原始优化问题（1）中的目标 f 不同，$\mathrm{EI}_n(x)$ 的评估是便宜的，并且可以轻松评估一阶和二阶导数。因此，期望改进算法的实现可以使用连续的一阶或二阶优化方法来解决（9）。例如，作者发现的一个有效方法是计算一阶导数并使用拟牛顿方法 L-BFGS-B（Liu 和 Nocedal，1989）。

图 3 展示了 $\mathrm{EI}_n(x)$ 在 $\Delta_n(x)$ 和后验标准差 $\sigma_n(x)$ 方面的轮廓。$\mathrm{EI}_n(x)$ 在 $\Delta_n(x)$ 和 $\sigma_n(x)$ 上都是递增的。相同 EI 值的 $\Delta_n(x)$ 与 $\sigma_n(x)$ 的曲线显示了 EI 在评估高预期质量的点（高 $\Delta_n(x)$）和高不确定性的点（高 $\sigma_n(x)$）之间的平衡。在优化的背景下，相对于过往最佳点，评估高预期质量 (high expected quality) 的点很有价值, 因为好的近似全局最优解很可能位于此类点。另一方面，在高不确定性的点进行评估也是有价值的，因为这可以了解目标函数在我们知识较少且远离已测量点的位置的情况。一个远远好于我们之前见过的点，可能恰好位于那里。

图 1 展示了底部面板中的 EI(x)。我们可以看到这种权衡，期望改善最大的地方出现在后验标准差高（远离先前评估的点）和后验均值也高的地方。最小的期望改善为 0，出现在我们之前评估的点上。此时后验标准差为 0，后验均值必然不大于之前评估的最佳点。期望改善算法将选择在标有 x 的点进行下次评估，在该点 EI 达到最大值。

基于高期望性能与高不确定性之间的权衡来选择评估位置在其他领域中也出现，包括多臂赌博机（Mahajan 和 Teneketzis，2008）以及强化学习（Sutton 和 Barto，1998），通常被称为 “探索与开发权衡”（Kaelbling 等人，1996）。

4.2 知识梯度 Knowledge Gradient

知识梯度获取函数是通过重新审视在 EI 推导中假设我们仅愿意返回之前评估点作为最终解的假设而得出的。当评估没有噪声且我们非常规避风险时，这一假设是合理的，但如果决策者愿意容忍一定风险，那么她可能愿意报告一个带有某种不确定性的最终解。此外，如果评估存在噪声（在第 5 节中讨论），那么报告的最终解必然具有不确定值，因为我们几乎无法对其进行无限次评估。

我们通过允许决策者返回她喜欢的任何解来替换这个假设,即使它没有被之前评估。我们还假设风险中立(Berger，2013），即我们根据期望值来评估随机结果 X。如果我们在 n 个样本后停止采样，选择的解将是具有最大 $\mu_n(x)$ 值的解。这个解（称为 x^*，因为它近似全局最优解 x^*）的值为 $f(x^*)$。在后验下，$f(x^*)$ 是随机的，并且具有条件期望值 $\mu_n(x^*) = \max_{x'} \mu_n(x') = \mu_n^*$。

另一方面，如果我们在 x 处再进行一次采样，我们将获得一个新的后验分布，其后验均值为 $\mu_{n+1}(x')$。这个后验均值将通过（3）计算，但包括额外的观察值 x_{n+1}, y_{n+1}。如果我们在此采样后报告最终解，其期望值将是

$$\mu_{n+1}(x') = \mu_n(x') + \frac{\sigma_n^2(x')}{\sigma_n^2(x) + \sigma^2},$$

其中 σ^2 是由于在点 x 处进行新评估而引入的噪声方差。

因此，知识梯度（Knowledge Gradient, KG）策略会选择使得

$$\mathrm{KG}(x) = \mathbb{E}[f(x^*)|x_{1:n}, x] - \mu_n(x^*)$$

最大化的点, 即 $KG_n(x)$。

该算法最早由 Frazier 等人（2009）提出，适用于离散的 A 上的 GP 回归，基于早期的工作（Frazier et al., 2008）提出的相同算法，针对贝叶斯排名和选择（Chick 和 Inoue, 2001）采用独立先验的场景。（贝叶斯排名和选择与贝叶斯优化相似，只是 A 是离散的且有限的，观察结果必然有噪声，且先验通常在 x 之间是独立的。）

计算 KG 采集函数的最简单方法是通过模拟，如算法 2 所示。在一个循环中，该算法模拟可能的观察值 y_{n+1}，这可能是从在指定的 x 处进行评估 $n+1$ 得到的。然后它计算如果 y_{n+1} 的值是实际测量结果时，新后验均值 μ_{n+1}^* 的最大值。接着，它减去 μ_n^* 来获得相应的解质量提升。这构成了算法的一次循环。它将这个循环迭代多次（J 次），并对不同模拟值的 y_{n+1} 获得的差异 $\mu_{n+1}^* - \mu_n^*$ 进行平均，以估计 KG 采集函数 $\mathrm{KG}_n(x)$。随着 J 的增大，这个估计值收敛到 $\mathrm{KG}_n(x)$。

理论上，该算法可以用于通过无导数的模拟优化方法来评估 $\mathrm{KG}_n(x)$，从而优化 KG 采集函数。然而，在没有导数的情况下优化噪声模拟函数是具有挑战性的。Frazier 等人（2009）提出了对 A 进行离散化，并利用正态分布的性质精确计算（10）。这一方法在低维问题中效果良好，但在高维时计算成本变得很高。

Algorithm 2 基于模拟的知识梯度因子的计算 $\mathrm{KG}_n(x)$。

设 $\mu_n^* = \max_{x'} \mu_n(x')$。
for 每个模拟的 μ_n 和 μ_{n+1} 下面，使用非线性优化方法如 L-BFGS。 **do**
  for $j = 1$ to J **do**
    生成 $y_{n+1} \sim \mathrm{Normal}(\mu_n(x), \sigma_n^2(x))$。(等效于 $Z \sim \mathrm{Normal}(0,1)$ 和 $y_{n+1} = \mu_n(x) + \sigma_n(x)Z$。)
    将 $\mu_{n+1}(x'; x, y_{n+1})$ 设置为通过 (3) 得到的后验均值在 x' 处，使用 (x, y_{n+1}) 作为最后的观察值。
    $\mu_{n+1}^* = \max_{x'} \mu_{n+1}(x'; x, y_{n+1})$。
    $\Delta^{(j)} = \mu_{n+1}^* - \mu_n^*$。
  end for
  通过 $\frac{1}{J}\sum_{j=1}^{J} \Delta^{(j)}$ 估计 $\mathrm{KG}_n(x)$。

针对维度问题，Wu 和 Frazier（2016）提出了一种更高效且可扩展的方法，基于多启动随机梯度上升。随机梯度上升（Robbins 和 Monro, 1951; Blum, 1954）是一种用于寻找函数局部最优解的算法，广泛用于机器学习中（Bottou, 2012）。多启动随机梯度上升（Martí 等人，2016）从不同的起始点运行多个随机梯度上升实例，并选择找到的最佳局部最优解作为近似的全局最优解。

我们在算法 3 中总结了最大化 KG 采集函数的这一方法。该算法遍历起始点，由 r 索引，并为每个起始点维护一个迭代序列 $x_t^{(r)}$，由 t 索引，收敛于 KG 采集函数的局部最优解。内循环 t 使用一个*随机梯度* G，它是一个随机变量，其期望值等于 KG 采集函数对我们采样位置的梯度，在当前迭代点 $x_{t-1}^{(r)}$ 处进行评估。我们通过在随机梯度 G 的方向上迈出一步来获得下一个迭代点。该步长的大小由 G 的幅度和递减步长 α_t 决定。一旦随机梯度上升在每个起始点上运行了 T 次迭代，算法 3 就会利用模拟（算法 2）来评估从每个起始点获得的最终点的 KG 采集函数，并选择最佳结果。

Algorithm 3 有效的方法，用于找到具有最大 $\mathrm{KG}_n(x)$ 的 x，基于多启动随机梯度上升。输入参数包括启动次数 R、每次随机梯度上升的迭代次数 T、用于定义步长序列的参数 a 以及复制次数 J。建议的输入参数：$R = 10$，$T = 10^2$，$a = 4$，$J = 10^3$。

for $r = 1$ to R **do**
  从 A 中均匀随机选择 $x_0^{(r)}$。
  for $t = 1$ to T **do**
    让 G 为算法 4 中 $\nabla \mathrm{KG}_n(x_{t-1}^{(r)})$ 的随机梯度估计。
    设定 $\alpha_t = a/(a+t)$。
    $x_t^{(r)} = x_{t-1}^{(r)} + \alpha_t G$。
  end for
  使用算法 2 和 J 次复制来估计 $\mathrm{KG}_n(x_T^{(r)})$。
end for
返回具有最大估计值 $\mathrm{KG}_n(x_T^{(r)})$ 的 $x_T^{(r)}$。=0

算法 3 的内循环中使用的随机梯度 G 是通过算法 4 计算的。该算法基于这样一个思想：我们可以在足够的规则性条件下交换梯度和期望，从而写成，

$$\nabla \mathrm{KG}_n(x) = \nabla \mathbb{E}_n\left[\mu_{n+1}^* - \mu_n^* \mid x_{n+1} = x\right] = \mathbb{E}_n\left[\nabla \mu_{n+1}^* \mid x_{n+1} = x\right],$$

这里我们注意到 μ_n^* 与 x 无关。这种方法称为*无穷小扰动分析* (infinitesimal perturbation analysis)（Ho 等, 1983）。因此，要构造随机梯度，仅需采样 $\nabla \mu_{n+1}^*$。换句话说，首先在算法 2 的内循环中采样 Z，然后在计算关于 x 的 μ_{n+1}^* 的

梯度时保持 Z 不变。为了计算这个梯度，考虑到 μ_{n+1} 是关于 x' 的 $\mu_{n+1}(x';x,y_{n+1}) = \mu^*_{n+1}(x';x,\mu_n(x)+\sigma_n(x)Z)$ 的最大值。这是一个关于 x 的函数集合的最大值。根据包络定理（Milgrom 和 Segal，2002），在充分规则性条件下，关于 x 的函数集合的最大值的梯度由首先找到该集合中的最大值，然后对这个单一函数关于 x 进行微分来给出。在我们的设置中，我们通过让 x^* 为使 $\mu_{n+1}(x';x,y_{n+1})$ 最大的 x'，然后计算 $\mu_{n+1}(x^*;x,\mu_n(x)+\sigma_n(x)Z)$ 关于 x 的梯度，同时保持 x^* 不变。换句话说，

$$\nabla \max_{x'} \mu_{n+1}(x';x,\mu_n(x)+\sigma_n(x)Z) = \nabla \mu_{n+1}(x^*;x,\mu_n(x)+\sigma_n(x)Z),$$

在这里，我们提醒读者，∇ 是指对 x 进行梯度运算，在这里和其他地方都如此。这个过程在算法 4 中进行了总结。

Algorithm 4 模拟无偏随机梯度 G，使得 $\mathbb{E}[G] = \nabla \mathrm{KG}_n(x)$。这个随机梯度可以用于随机梯度上升中，以优化 KG 采集函数。

for $j = 1$ to J **do**
  生成 $Z \sim \mathrm{Normal}(0,1)$
  $y_{n+1} = \mu_n(x) + \sigma_n(x)Z$。
  设定 $\mu_{n+1}(x';x,y_{n+1}) = \mu^*_{n+1}(x';x,\mu_n(x)+\sigma_n(x)Z)$ 为通过 (3) 计算的后验均值在 x' 处，使用 (x,y_{n+1}) 作为最后的观察值。
  求解 $\max_{x'} \mu_{n+1}(x';x,y_{n+1})$，例如使用 L-BFGS。令 x^* 为使其最大化的 x'。
  设 $G(j)$ 为 $\mu_{n+1}(x^*;x,\mu_n(x)+\sigma_n(x)Z)$ 关于 x 的梯度，保持 x^* 不变。
end for
通过 $G = \frac{1}{J}\sum_{j=1}^{J} G(j)$ 估计 $\nabla \mathrm{KG}_n(x)$。

与预期改进不同，KG 采集函数考虑了 f 在全域上的后验分布，以及该样本将如何改变该后验。KG 赋予了在最大化后验均值的情况下进行测量的正价值，即使该点的值并不比之前的最佳点好。这在无噪声评估的典型贝叶斯优化问题中提供了小的改进（Frazier et al., 2009）。

4.3 熵搜索和预测熵搜索 Entropy Search and Predictive Entropy Search

熵搜索 (ES) **?** 采集函数根据其微分熵评估我们对全局最大值位置的信息。ES 寻求评估的点，使得微分熵的降低最大。（回想一下，例如，Cover 和 Thomas (2012) 中提到的，连续概率分布 $p(x)$ 的微分熵为 $\int p(x)\log(p(x))\,dx$，而较小的微分熵表示较少的不确定性。）预测熵搜索 (PES) (Hernández-Lobato et al., 2014) 寻求相同的点，但使用基于互信息的熵减少目标的重构。对 PES 和 ES 的精确计算将给出等效的采集函数，但通常不可能进行精确计算，因此用于近似 PES 和 ES 采集函数的计算技术的差异在两个方法中产生了采样决策的实际差异。我们首先讨论 ES，然后讨论 PES。

设 x^* 为 f 的全局最优解。时间 n 时，f 的后验分布在 x^* 处引入一个概率分布。事实上，如果域 A 是有限的，那么我们可以通过向量 $(f(x) : x \in A)$ 表示 f 在其域上的值，x^* 将对应于该向量中的最大元素。此向量在时间 n 的后验分布将是多元正态分布，而该多元正态分布将暗示 x^* 的分布。当 A 是连续的时，同样的理念适用，其中 x^* 是一个随机变量，其分布由 f 的高斯过程后验暗示。

理解这一点后，我们用符号 $H(P_n(x^*))$ 表示时间 n 后验分布对 x^* 的熵。类似地，$H(P_n(x^*|x,f(x)))$ 表示如果我们在 x 处进行观察并看到 $f(x)$，则时间 $n+1$ 后验分布对 x^* 的熵。这个量依赖于观察到的 $f(x)$ 的值。然后，采样 x 带来的熵减少可以写成，

$$\mathrm{ES}_n(x) = H(P_n(x^*)) - \mathbb{E}_{f(x)}\left[H(P_n(x^*|f(x)))\right]. \tag{11}$$

在第二项中，外部期望的下标表示我们对 $f(x)$ 进行期望计算。等价地，这可以写为 $\int \varphi(y;\mu_n(x),\sigma_n^2(x))H(P_n(x^*|f(x)=y))\,dy$，其中 $\varphi(y;\mu_n(x),\sigma_n^2(x))$ 是均值为 $\mu_n(x)$ 和方差为 $\sigma_n^2(x)$ 的正态密度。

与 KG 一样，ES 和 PES 都受到测量如何改变全域后验分布的影响，而不仅仅是看它是否在采样的点上优于现有解决方案。这在决定在哪里采样时对于“异类”问题尤其有用，因此 ES 和 PES 相比于 EI 可以提供实质性的价值。

虽然 ES 可以近似计算和优化（Hennig 和 Schuler，2012），但这样做很具挑战性，因为 (a) 高斯过程最大值的熵没有封闭形式；(b) 我们必须为大量 y 计算此熵以近似 (11) 中的期望；以及 (c) 然后我们必须优化这个难以评估的函数。与 KG 不同，没有已知的方法可以计算随机梯度，从而简化此计算。

PES 提供了一种计算 (11) 的替代方法。该方法注意到，由于测量 $f(x)$ 而导致的 x^* 的熵减少等于 $f(x)$ 和 x^* 之间的互信息，这又等于由于测量 x^* 而导致的 $f(x)$ 的熵减少。这种等价性给出了表达式

$$\mathrm{PES}_n(x) = \mathrm{ES}_n(x) = H(P_n(f(x))) - \mathbb{E}_{x^*}\left[H(P_n(f(x)|x^*))\right]. \tag{12}$$

这里，第二项中的期望下标表示对 x^* 进行期望计算。

与 ES 不同，PES 采集函数中的第一项 $H(P_n(f(x)))$ 可以封闭计算。第二项仍然需要近似。Hernández-Lobato et al. (2014) 提供了一种从后验分布中采样 x^* 的方法，并使用期望传播 (Minka, 2001) 来近似 $H(P_n(f(x)|x^*))$。这种评估方法可以通过无导数优化的模拟方法进行优化。

4.4 多步骤最优采集函数 Multi-Step Optimal Acquisition Functions

我们可以将解决问题 (1) 的行为视为一个顺序决策问题 **??**，在这个问题中，我们依次选择 x_n，并观察 $y_n = f(x_n)$，而 x_n 的选择依赖于所有过去的观察。在这些观察结束时，我们会收到一个奖励，这个奖励可能等于所观察到的最佳点的值 $\max_{m\le N} f(x_m)$，如在 EI 分析中，或者可能等于基于所有观察选择的某个新点 x^* 的目标值 $f(x^*)$，如在 KG 分析中，或者可能是后验分布在 x^* 处的熵，如在 ES 或 PES 中。

根据构造，当 $N = n+1$ 时，EI、KG、ES 和 PES 采集函数是最优的，在后验下最大化预期奖励。然而，当 $N > n+1$ 时，它们不再明显最优。原则上，可以通过随机动态规划 **?** 计算一个多步骤最优采集函数，该函数可以最大化一般 N 的预期奖励，但所谓的维度诅咒 **?** 使得在实践中计算这个多步骤最优采集函数极具挑战性。

尽管如此，文献最近开始部署近似计算此解的方法，尝试包括 Lam et al. (2016)；Ginsbourger 和 Riche (2010)；González et al. (2016)。这些方法似乎尚未处于可以广泛应用于实际问题的状态，因为在近似解决随机动态规划问题时引入的误差和额外成本通常压倒了考虑多步骤所带来的好处。然而，考虑到强化学习和近似动态规划的并行进展，这代表了贝叶斯优化的一个有前景且令人兴奋的方向。

此外，还有其他问题设置与贝叶斯优化最常考虑的问题密切相关，在这些设置中可以计算多步骤最优算法。例如，Cashore et al. (2016) 和 Xie 和 Frazier (2013) 使用问题结构有效地计算某些类贝叶斯可行性判定问题的多步骤最优算法，在这些问题中，我们希望高效采样以确定每个 x 的 $f(x)$ 是否高于或低于某个阈值。类似地，Waeber et al. (2013) 基于 Jedyank et al. (2012) 计算一维随机根查找问题的多步骤最优算法，该问题具有熵目标。虽然这些最优多步骤方法仅直接适用于非常特定的设置，但它们提供了一个机会，以更普遍地研究从单步最优到多步最优所能实现的改进。令人惊讶的是，在这些设置中，现有的采集函数在性能上几乎与多步骤最优算法相当。例如，Cashore et al. (2016) 中进行的实验表明，KG 采集函数在计算中与最优相差仅 2%，而 Waeber et al. (2013) 显示在他们考虑的设置中，熵搜索采集函数是多步骤最优的。从这些结果中推广，可以认为单步采集函数足够接近最优，以至于进一步的改进在实际中并不具有意义，或者多步骤最优算法将在尚未识别的重要实践设置中提供显著更好的性能。

5 异类贝叶斯优化 Exotic Bayesian Optimization

在上面我们描述了解决第 1 节中描述的“标准”贝叶斯优化问题的方法。这个问题假设可行集合，其中成员资格易于评估，例如超矩形或单纯形；缺乏导数信息；以及无噪声评估。

虽然有很多应用问题满足标准问题的所有假设，但还有更多的问题会打破一个或多个这些假设。我们称这些为“异类”问题。在这里，我们描述一些突出例子，并提供更详细阅读的参考。(虽然我们在本节中讨论噪声评估，但它们在考虑时与其他问题相比，实质上并不那么异类，并且通常被视为标准问题的一部分。)

5.1 噪声评估 Noisy Evaluations

高斯过程回归可以自然扩展到具有已知方差的独立正态分布噪声的观察 **?**。这在 (3) 中的协方差矩阵添加了对角线项，条目等于噪声的方差。在实践中，这个方差是未知的，因此最常见的方法是假设噪声具有共同方差，并将此方差作为超参数包含在内。也可以通过使用第二个高斯过程来建模方差与领域的变化来进行推断 **?**。

KG、ES 和 PES 采集函数直接适用于带噪声的设置，并保持它们的一步最优性质。然而，直接使用 EI 采集函数存在概念上的挑战，因为来自函数值的“改进”不再容易定义，并且 (7) 中的 $f(x)$ 不再被观察到。作者们采用了各种启发式方法，替代 (7) 中的 $f(x)$ 的分布，通常使用在先前评估点的后验均值的最大值来替代 f_n^*。对于 $f(x)$ 的分布，流行的替代方法包括 $\mu_{n+1}(x)$ 的分布、y_{n+1} 的分布，以及继续使用 $f(x)$ 的分布，即使它没有被观察到。由于这些近似，KG 在存在较大噪声的问题中可以显著超过 EI **??**。

作为在测量有噪声时应用 EI 的替代方法，Scott et al. (2011) 考虑了在 EI 推导中所做的限制下的噪声评估：报告的解决方案需要是先前报告的点。然后它在这个假设下找到一个步骤最优的采样位置。它的分析类似于用于推导 KG 策略的分析，不同之处在于我们将 x^* 限制在那些已经被评估的点。

实际上，如果我们在 n 次测量后报告最终解决方案，那么它将在 $x_{1:n}$ 中是具有最大 $\mu_n(x)$ 值的点，并且它的条件期望值为 $\mu_n^{**} = \max_{i=1,\dots,n} \mu_n(x_i)$。如果我们在 $x_{n+1} = x$ 处再进行一次采样，它将在新后验下的条件期望值为 $\mu_{n+1}^{**} = \max_{i=1,\dots,n+1} \mu_{n+1}(x_i)$。取差值的期望，采样在 x 处的价值为

$$E_n[\mu_{n+1}^{**} - \mu_n^{**} \mid x_{n+1} = x].$$

与无噪声评估的情况不同，这个样本可能导致 $\mu_{n+1}(x_i)$ 与 $\mu_n(x_i)$ 对于 $i \le n$ 不同，从而需要比无噪声设置中更复杂的计算（但比 KG 策略简单）。计算这个量及其导数的过程在 Scott et al. (2011) 中给出。虽然我们可以将这个采集函数视为对 KG 采集函数的近似，正如 Scott et al. (2011) 所做的（他们称之为 KGCP 采集函数），但我们在这里争辩说，它是 EI 假设最自然的推广到有噪声测量的情况。

5.2 并行评估 Parallel Evaluations

使用多个计算资源进行并行评估可以在获得通常仅需一次序列评估所需的时间内获得多个函数评估。因此，进行并行函数评估是一种在更短时间内解决优化问题的概念上具有吸引力的方法。EI、KG、ES 和 PES 都可以以简单的方式扩展以允许并行函数评估。例如，EI 变为

$$EI_n(x^{(1:q)}) = E_n\left[\max_{i=1,\dots,q} f(x^{(i)}) - f_n^*\right]^+,$$

其中 $x^{(1:q)} = (x^{(1)}, \dots, x^{(q)})$ 是我们提出评估的一组点 **?**。并行 EI（也称为 Ginsbourger 等（2007）提出的多点 EI）则提议评估共同最大化该标准的点集。这种方法还可以异步使用，我们固定当前正在评估的 $x^{(j)}$，并通过优化它们对应的 $x^{(j)}$ 来分配空闲的计算资源。

并行 EI (14) 和其他并行采集函数比第 4 节中的原始序列版本更难优化。一个创新是对并行 EI 采集函数的常量骗子近似（Ginsbourger et al., 2010），它假设对于 $j < i$，$f(x^{(j)})$ 已经被观察到，并且它们的值等于后验下的常量（通常是 $f(x^{(j)})$ 的期望值），然后顺序选择 $x^{(j)}$。这大大加快了计算。Wang et al. (2016a) 在此基础上展示了无穷小扰动分析可以产生随机的随机梯度，这些梯度是 $\nabla EI_n(x^{(1:q)})$ 的无偏估计，然后可以在多起始随机梯度上升中使用以优化 (14)。该方法已用于实现多达 $q = 128$ 的并行评估的并行 EI 过程。并行 KG 的计算方法由 Wu 和 Frazier (2016) 开发，并在第 6 节中讨论的康奈尔 MOE 软件包中实现。那篇文章遵循第 4.2 节中描述的随机梯度上升方法，这种方法在并行设置中良好地推广。

与多信息源优化密切相关的问题是一对问题

$$\max_x \int f(x,w)p(w)\,dw,$$

$$\max_x \sum_w f(x,w)p(w),$$

其中 f 的评估代价昂贵。这些问题在文献中以多种名称出现：在统计学中称为带随机环境条件的优化（Chang et al., 2001），在机器学习中称为多任务贝叶斯优化（Swersky et al., 2013），以及集成响应函数的优化（Williams et al., 2000）和具有昂贵被积函数的优化（Toscano-Palmerin 和 Frazier, 2018）。

在这里，w 可以表示影响目标函数 f 的环境因素，且通常是随机变量。目标是选择 x 来优化期望表现。由于 f 可能很复杂（例如，具有多种局部最优解），贝叶斯优化在解决此类问题中表现出很大潜力。

在这些问题中，EI、KG、ES 和 PES 等采集函数也可以直接应用于随机条件下的优化。特别是，EI 采集函数可以修改为考虑 $f(x,w)$ 的期望值，即

$$EI_n(x) = E_w\left[\max\left(f(x,w) - f_n^*, 0\right)\right],$$

其中 $f_n^* = \max_{i=1,\ldots,n} f(x_i,w)$ 的最大值可以计算为多个环境条件下的观察值。

KG、ES 和 PES 采集函数同样能够以更自然的方式整合环境的不确定性。在这类问题中，选择采样点 x 可能涉及到评估在不同的 w 下的性能，从而更全面地了解目标函数的行为。

这类方法的研究可参见文献中对这类优化问题的讨论，包括使用多任务学习的策略和处理随机环境条件的最优方法。

5.3 约束条件 Constraints

在第 1 节中提出的问题中，我们假设可行集是一个简单的集合，在其中评估成员资格很容易。文献中也考虑了更一般的问题，

$$\max_x f(x)$$

满足 $g_i(x) \geq 0$，其中 $i = 1,\ldots,I$，这里的 g_i 与 f 的评估成本相同。当 f 和 g_i 在没有噪声的情况下可以评估时，EI 自然地推广到这种设置：当评估的 x 是可行的（即对于所有 x 满足 $g_i(x) \geq 0$）且 $f(x)$ 优于先前评估的最佳可行点时，会产生改进。这一点在 Schonlau et al. (1998) 的第 4 节中提出，并由 Gardner et al. (2014) 独立研究。PES 也被研究用于此设置 (Hernández-Lobato et al., 2015)。

5.4 多保真度和多信息源评估 Multi-Fidelity and Multi-Information Source Evaluations

在多保真度优化中，我们有一个信息源集合 $f(x,s)$，而不是单一目标 f。这里，s 控制 “保真度”，较低的 s 表示更高的保真度，而 $f(x,0)$ 对应于原始目标。提高保真度（减小 s）可以更准确地估计 $f(x,0)$，但代价更高，记作

$c(s)$。例如，x 可能描述工程系统的设计，s 则表示用于求解建模该系统的偏微分方程的网格大小。或者，s 可能描述在稳态模拟中使用的时间范围。作者们最近还考虑了神经网络的优化，其中 s 索引在训练机器学习算法中使用的迭代次数或数据量 (Swersky et al., 2014; Klein et al., 2016)。通过假设 $f(x,s)$ 等于 $f(x,0)$ 并且以其方差 $\Lambda(s)$ 随 s 增加的噪声进行观察，或者假设 $f(x,s)$ 提供确定性的评估，其中 $f(x,s+1)-f(x,s)$ 由一个在 x 上变化的均值 $G(x)$ 模型化的高斯过程来描述。这两种设置都可以通过在 f 上建模高斯过程来建模，包括 x 和 s 作为建模域的一部分。

总体目标是通过在总成本小于某个预算 B 的一系列点和保真度 (x_n,s_n) 处观察 $f(x,s)$ 来解决 $\max_x f(x,0)$。关于多保真度优化的工作包括 Huang et al. (2006); Sóbester et al. (2004); Forrester et al. (2007); McLeod et al. (2017); Kandasamy et al. (2016)。

在更一般的多信息源优化问题中，我们放宽了对 $f(x,s)$ 按准确性和成本排序的假设。相反，我们简单地有一个函数 f，接受设计输入 x 和信息源输入 s，其中 $f(x,0)$ 是目标，而对于 $s\neq 0$，$f(x,s)$ 观察到时相对于目标有不同的偏差、不同数量的噪声和不同的成本。

例如，x 可能代表飞机机翼的设计，$f(x,0)$ 是在准确但缓慢的模拟器下对机翼的性能预测，而 $f(x,s)$ 在 $s=1,2$ 时代表在两个便宜的近似模拟器下的预测性能，这两个模拟器做出不同的假设。可能在搜索空间的某些区域中 $f(x,1)$ 是准确的，而在其他区域中则有明显的偏差，而 $f(x,2)$ 在其他区域中是准确的。在这种情况下，$f(x,1)$ 与 $f(x,2)$ 的相对准确性取决于 x。关于多信息源优化的工作包括 Lam et al. (2015); Poloczek et al. (2017)。

在这些问题中直接应用 EI 是困难的，因为评估 $f(x,s)$ 对于 $s\neq 0$ 从未提供观察到的最佳目标函数值的改进 $\max\{f(x_n,0):s_n=0\}$。因此，EI 在这个设置下的直接转换导致 $EI=0$ 对于 $s\neq 0$，从而只测量最高的保真度。出于这个原因，来自 Lam et al. (2015) 的基于 EI 的方法在假设将观察到 $f(x,0)$ 时使用 EI 来选择 x_n（即使它不会被观察到），并使用一个单独的过程来选择 s。KG、ES 和 PES 可以直接应用于这些问题，如 Poloczek et al. (2017) 所示。

5.5 随机环境条件和多任务贝叶斯优化 Random Environmental Conditions and Multi-Task Bayesian Optimization

与多信息源优化密切相关的问题是以下一对问题

$$\max_x \int f(x,w)p(w)\,dw,$$

$$\max_x \sum_w f(x,w)p(w),$$

其中 f 的评估成本很高。这些问题在文献中有多种名称：随机环境条件下的优化 (Chang et al., 2001)、多任务贝叶斯优化 (Swersky et al., 2013)、集成响应函数的优化 (Williams et al., 2000) 和具有昂贵积分的优化 (Toscano-Palmerin 和 Frazier, 2018)。

并非将目标函数 $\int f(x,w)p(w)\,dw$ 作为评估单元，一个自然的方法是在感兴趣的 x 处对少量的 w 进行评估 $f(x,w)$。这为 x 处的目标函数提供了部分信息。基于这些信息，可以探索不同的 x，或者以更高的精度解决当前的 x。此外，通过利用在附近 x 处的 w 的观察值，可能已经获得了关于特定 $f(x,w)$ 的大量信息，从而减少了对其进行评估的需要。基于这种直觉的算法在效率上往往远超那些通过数值积分或完全求和的方式在每次评估中简单地评估整个目标的方法（Toscano-Palmerin 和 Frazier，2018）。

这一对问题出现在工程系统的设计和生物医学问题中，例如关节置换（Chang 等，2001）和心血管旁路移植（Xie 等，2012），其中 $f(x,w)$ 是某种昂贵评估的计算模型下设计 x 在环境条件 w 下的性能，$p(w)$ 是描述条件 w 发生频率的简单函数（例如，正态密度），我们的目标是优化平均性能。它还出现在机器学习中，优化交叉验证性能。在这里，我们将数据分成多个块或“折”，由 w 索引，而 $f(x,w)$ 是在没有来自此折数据的情况下训练的机器学习模型在折 w 上的测试性能。

该领域的方法包括针对无噪声问题的 KG（Xie 等，2012）和一般问题（Toscano-Palmerin 和 Frazier，2018）、PES（Swersky 等，2013）以及 EI 的修改（Groot 等，2010；Williams 等，2000）。如在多信息源优化中一样，未修改的 EI 获取函数在这里并不合适，因为观察 $f(x, w)$ 并不提供目标函数的观察（除非所有 $w' \neq w$ 已在该 x 处被观察），也不一定是严格正的改善。因此，Groot 等（2010）和 Williams 等（2000）使用 EI 来选择 x，仿佛我们观察了目标，然后使用一个单独的策略来选择 w。

5.6 导数观察 Derivative Observations

最后，我们讨论具有导数的优化。对 $\nabla f(x)$ 的观察（可选地带有正态分布噪声）可以直接纳入 GP 回归（Rasmussen 和 Williams, 2006，第 9.4 节）。Lizotte（2008）提出以这种方式在贝叶斯优化中使用梯度信息，并结合 EI 采集函数，显示出相对于 BFGS（Liu 和 Nocedal, 1989）的改进。EI 在提出对 $\nabla f(x)$ 的观察时未改变其值，与仅观察 $f(x)$ 时的值相比。（然而，如果之前的导数观察对时间 n 后验有贡献，则该时间 n 后验将与仅观察 $f(x)$ 时的情况不同。）因此，EI 没有利用导数信息的可用性，例如，在远离先前评估的点进行评估，而导数信息在这些点会特别有用。解决此问题的 KG 方法由 Wu et al. (2017) 提出。在该领域的其他相关工作中，Osborne et al. (2009) 提出了利用梯度信息来改善 GP 回归中协方差矩阵的条件性，而 Ahmed et al. (2016) 提出了在观察梯度时选择单个方向导数以提高 GP 推理的计算可行性的方法。

6 Software

有多种贝叶斯优化和高斯过程回归的代码。这些高斯过程回归和贝叶斯优化包有几个是一起开发的，其中贝叶斯优化包利用了高斯过程回归包。其他包是独立的，只提供高斯过程回归支持或贝叶斯优化支持。我们在此列出一些最重要的包，以及截至 2018 年 6 月的当前网址。

- DiceKriging 和 DiceOptim 是分别用于高斯过程回归和贝叶斯优化的 R 包。它们在 Roustant et al. (2012) 中进行了详细描述，并可通过 CRAN 获取，网址为 https://cran.r-project.org/web/packages/DiceOptim/index.html。
- GPyOpt (https://github.com/SheffieldML/GPyOpt) 是一个基于高斯过程回归库 GPy (https://sheffieldml.github.io/GPy) 的 Python 贝叶斯优化库，由谢菲尔德大学的机器学习小组编写和维护。
- Metrics Optimization Engine (MOE, https://github.com/Yelp/MOE) 是一个用 C++ 编写的贝叶斯优化库，带有 Python 包装器，支持基于 GPU 的计算以提高速度。它是由 Yelp 的创始人开发的，后者是贝叶斯优化初创公司 SigOpt 的创始人（http://sigopt.com）。康奈尔 MOE (https://github.com/wujian16/Cornell-MOE) 是基于 MOE 构建的，进行了更易于安装的更改，并支持并行和导数启用的知识梯度算法。
- Spearmint（https://github.com/HIPS/Spearmint），早期版本在不同许可证下可在 https://github.com/JasperSnoek/spearmint 获取，是一个 Python 贝叶斯优化库。Spearmint 由贝叶斯优化初创公司 Whetlab 的创始人编写，该公司于 2015 年被 Twitter 收购（Perez, 2015）。
- DACE（计算实验的设计与分析）是一个用 MATLAB 编写的高斯过程回归库，可在 http://www2.imm.dtu.dk/projects/dace/ 获取。尽管最后更新于 2002 年，但它仍然被广泛使用。
- GPflow (https://github.com/GPflow/GPflow) 和 GPyTorch (https://github.com/cornellius-gp/gpytorch) 是分别基于 Tensorflow (https://www.tensorflow.org/) 和 PyTorch (https://pytorch.org/) 的 Python 高斯过程回归库。

- laGP (https://cran.r-project.org/web/packages/laGP/index.html) 是一个用于高斯过程回归和贝叶斯优化的 R 包，支持不等式约束。

7 结论与研究方向

我们介绍了贝叶斯优化，首先讨论了 GP 回归，然后是期望改进、知识梯度、熵搜索和预测熵搜索采集函数。接着，我们讨论了一系列异域贝叶斯优化问题：带噪声的测量；并行评估；约束；多保真度和多信息源；随机环境条件和多任务贝叶斯优化；以及导数观察。

在这个令人兴奋的领域中，许多研究方向涌现出来。首先，在贝叶斯优化的理论理解方面有很大的发展空间。如第 4.4 节所述，能够计算多步最优算法的环境设置极为有限。此外，尽管我们目前在实践中使用的采集函数似乎与我们能够计算的最优多步算法的表现几乎相当，但我们目前没有有限时间界限来解释它们的近似最优经验表现，也不知道多步最优算法在尚未理解的环境中是否能提供显著的实际利益。即使在渐近情况下，关于贝叶斯优化算法收敛速率的知识相对较少：尽管 Bull (2011) 为结合周期性均匀采样的期望改进建立了收敛速率，但尚不清楚去除均匀采样是否会导致相同或不同的速率。

其次，构建利用新颖统计方法的贝叶斯优化方法的空间。尽管高斯过程（或其变体，如 Snoek et al. (2014) 和 Kersting et al. (2007)）在贝叶斯优化的大多数研究中被使用，但似乎存在问题类别，可以通过其他方法更好地建模目标。开发广泛适用的新统计模型和为特定应用设计的模型都是很有意义的。

第三，开发在高维中表现良好的贝叶斯优化方法具有重要的实际和理论意义。研究方向包括开发统计方法，以识别和利用实际中出现的高维目标中存在的结构，这一研究已被 Wang et al. (2013, 2016b) 和 Kandasamy et al. (2015) 的最近研究所追求。参见 Shan 和 Wang (2010)。新的采集函数可能会在高维问题中提供显著的价值。

第四，开发利用今天的方法未考虑的异域问题结构的方法也是一个有趣的方向，符合第 5 节的精神。将这种方法论开发与贝叶斯优化应用于重要的现实问题相结合可能特别有效，因为在现实世界中使用方法往往会揭示出意想不到的困难并激发创造力。

第五，通过应用贝叶斯优化在多个领域实现显著影响似乎是可能的。其中一组应用领域，贝叶斯优化似乎特别适合提供影响的是化学、化工、材料设计和药物发现，这些领域的从业者进行涉及反复物理实验的设计工作，耗费数年的努力和大量的金钱成本。尽管在这些领域已有一些早期工作（Ueno et al., 2016；Frazier 和 Wang, 2016；Negoescu et al., 2011；Seko et al., 2015；Ju et al., 2017），但在这些领域的研究人员仍然相对较少，尚未意识到贝叶斯优化的强大能力和适用性。