中国科学: 数学 2025 年 第 55 卷 第 2 期: 501 ∼ 516
SCIENTIA SINICA Mathematica
综 述

《中国科学》杂志社
SCIENCE CHINA PRESS
CrossMark click for updates

连续 DR 次模最大化问题的理论与算法综述

献给韩继业教授 90 寿辰

剧嘉琛[1], 杨瑞琪[1], 张真宁[1*], 堵丁柱[2]

1. 北京工业大学运筹学与信息工程研究所, 北京 100124;
2. Department of Computer Science, University of Texas at Dallas, Richardson, Texas 75080, USA
E-mail: jujc8201@emails.bjut.edu.cn, yangruiqi@bjut.edu.cn, zhangzhenning@bjut.edu.cn, dzdu@utdallas.edu

收稿日期: 2024-06-06; 接受日期: 2024-09-30; 网络出版日期: 2025-01-03; * 通信作者
国家自然科学基金 (批准号: 12131003 和 12101587) 资助项目

摘要 收益递减 (diminishing returns, DR) 次模最大化问题是研究具有边际收益递减性质的函数的优化问题. 在连续优化领域中, DR 次模函数不仅在理论上具有深远的研究价值, 而且在应用上也展现了广泛的实用意义, 特别是在资源分配、广告预算优化、传感器网络和能量管理等实际应用场景中发挥了关键作用. 本文主要阐述连续域上 DR 次模最大化问题的经典算法及其相关变形问题的算法, 涵盖 Frank-Wolfe 型算法、双贪婪算法、随机算法和在线算法等, 并回顾近年来有关连续 DR 次模最大化问题的主要研究进展, 为进一步探索该领域的前沿问题提供了有益参考.

关键词 DR 次模最大化 非凸优化 近似算法

MSC (2020) 主题分类 90C26, 68W25

1 引言

集合域上的次模函数 (submodular functions) 是指具有收益递减 (diminishing returns, DR) 性的一类函数, 即增加同一个元素对函数的增益随着集合中已有元素数量的增加而减少. 次模函数因其所具备的这一独特的非线性结构, 而在组合优化领域中受到了广泛的关注, 并得到了一系列的研究结果 (参见文献 [1,40,49,51]). 近年来, 次模函数在人工智能等领域也有重要应用, 如机器学习的特征选择 [16]、字典学习 [10]、稀疏正则化 [34]、文档摘要 [36] 和变分推理 [11] 等. 借助次模函数, 许多研究设计了高效的优化算法, 并显著地提高了问题解的质量.

随着研究的深入, 次模函数的概念被进一步推广到连续域上, 衍生出 DR 次模函数 [3,37]. 具体而言, 对于 n 维 Euclid 空间中任意两个向量 $\boldsymbol{x} \leqslant \boldsymbol{y}$、单位向量 $\boldsymbol{e}_i$ ($i \in \{1, \ldots, n\}$) 和正实数 a, 若函数 f 满足

$$f(\boldsymbol{x} + a\boldsymbol{e}_i) - f(\boldsymbol{x}) \geqslant f(\boldsymbol{y} + a\boldsymbol{e}_i) - f(\boldsymbol{y}),$$

英文引用格式: Ju J C, Yang R Q, Zhang Z N, et al. A survey on theory and algorithms for continuous DR-submodular maximization (in Chinese). Sci Sin Math, 2025, 55: 501–516, doi: 10.1360/SSM-2024-0195

© 2025《中国科学》杂志社

则称函数 f 为连续 DR 次模函数. 这种推广不仅保留了集合次模函数的许多特性, 还能够处理多样化的优化问题. 特别是在应对大规模数据和复杂决策时, 连续 DR 次模函数最大化问题在诸多领域中展现了应用潜力. 例如, 在影响力最大化问题 [29] 中, 连续 DR 次模函数可以用来模拟广告预算在不同渠道上的分配, 以最大化广告的覆盖范围和影响力. 在传感器网络 [35] 中, DR 次模函数被用于优化能量分配, 以延长网络的监测时间和覆盖范围. 此外, 在收入最大化 [24] 和设施选址 [33] 等问题中, DR 次模函数也显示出其强大的建模和优化能力. 相较于离散次模函数, 连续域上的 DR 次模函数往往能够更精确地模拟和优化这些问题, 因此应用更广, 灵活性也更高.

本文总结了连续 DR 次模函数最大化问题及其变形的相关算法. 本文余下内容的结构安排如下. 第 2 节介绍连续 DR 次模函数、弱连续次模函数的定义及相关记号. 第 3 节介绍经典连续 DR 次模函数最大化问题的研究现状, 阐述连续 DR 次模函数最大化问题的近似难度, 介绍 Frank-Wolfe 型算法、双贪婪算法及算法的性能保证. 第 4 节介绍连续 DR 次模最大化问题的变形问题, 即随机模型和在线模型. 第 5 节提出一些值得探讨的问题并总结本文的主要贡献.

2 记号

本节简要介绍连续 DR 次模函数的定义及相关符号.

对于任意的正整数 k, 标记 $[k]=\{1,2,\dots,k\}$. 对于任意的 n 维 Euclid 空间中的向量 $\boldsymbol{x}\in\mathbb{R}^n$, $\boldsymbol{x}_i$ 表示向量 $\boldsymbol{x}$ 的第 i 个分量的值. $\boldsymbol{x}\vee\boldsymbol{y}$ 表示两个向量的合取. $\boldsymbol{x}\vee\boldsymbol{y}$ 的每个分量为向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 的对应分量的最大值, 即 $(\boldsymbol{x}\vee\boldsymbol{y})_i=\max\{\boldsymbol{x}_i,\boldsymbol{y}_i\}$, $i\in[n]$. $\boldsymbol{x}\wedge\boldsymbol{y}$ 表示两个向量的析取. $\boldsymbol{x}\wedge\boldsymbol{y}$ 的每个分量为向量为 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 的对应分量的最小值, 即 $(\boldsymbol{x}\wedge\boldsymbol{y})_i=\min\{\boldsymbol{x}_i,\boldsymbol{y}_i\}$, $i\in[n]$. $\boldsymbol{x}\leqslant\boldsymbol{y}$ 表示两个向量对应的分量均有 $\boldsymbol{x}_i\leqslant\boldsymbol{y}_i$, $i\in[n]$. 引入 $\boldsymbol{e}_i$ 表示第 i 个分量为 1、其他分量为 0 的单位向量.

首先给出实数格 [22] 的定义.

定义 2.1 设 L 为 n 维 Euclid 空间 $\mathbb{R}^n$ 中的实向量集合, 若对于 L 中的任意向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$, 都有 $\boldsymbol{x}\vee\boldsymbol{y}\in L$ 和 $\boldsymbol{x}\wedge\boldsymbol{y}\in L$, 则称 $(L,\vee,\wedge)$ 是实数格.

现给出实数格上连续 DR 次模函数 [3, 14, 41] 的定义.

定义 2.2 设 $(L,\vee,\wedge)$ 为实数格. 函数 $f:L\to\mathbb{R}_+$, 若对任意满足 $\boldsymbol{x}\leqslant\boldsymbol{y}\in L$ 的向量, 在任意方向 $i\in[n]$ 上, 对任意正实数 $a\geqslant 0$ 有 $\boldsymbol{x}+a\boldsymbol{e}_i$ 和 $\boldsymbol{y}+a\boldsymbol{e}_i\in L$, 且不等式

$$f(\boldsymbol{x}+a\boldsymbol{e}_i)-f(\boldsymbol{x})\geqslant f(\boldsymbol{y}+a\boldsymbol{e}_i)-f(\boldsymbol{y})$$

均成立, 则称函数 f 为实数格 $(L,\vee,\wedge)$ 上的连续 DR 次模函数.

在具体的研究工作中, 研究者会根据所研究问题的特点, 将 L 取为整个非负实数域 $\mathbb{R}_+^n$, 或是 $\mathbb{R}_+^n$ 中的某个紧集. 若为后者, 为了简化问题, 研究者通常会将实数格 L 限制在 $[0,1]^n$ 上.

在一些问题中, 研究者发现某些函数并不满足连续 DR 次模性. 为更好地研究问题, 研究者们引入一个相对更弱的性质来刻画函数, 即 γ 弱次模性 [8]. 下面给出 γ 弱次模函数的定义.

定义 2.3 令函数 $f:\mathbb{R}_+^n\to\mathbb{R}_+$ 为连续可微的, 梯度函数为 $\nabla f=(\partial_1 f,\partial_2 f,\dots,\partial_n f)$, $[\nabla f]_i=\partial_i f$ 表示梯度 ∇f 的第 i 个分量, $i\in[n]$. 若对任意的 $\boldsymbol{x}\leqslant\boldsymbol{y}\in\mathbb{R}_+^n$ 和任意的 $i\in[n]$, 由

$$\gamma=\inf_{\boldsymbol{x}\leqslant\boldsymbol{y}\in\mathbb{R}_+^n}\inf_{i\in[n]}\frac{[\nabla f(\boldsymbol{x})]_i}{[\nabla f(\boldsymbol{y})]_i}$$

可以得到确切的参数 γ, 则称函数 f 为 γ 弱次模函数.

如果函数 f 具有单调递增性质, 则 $\gamma \geqslant 0$. 函数 f 是连续 DR 次模函数当且仅当 $\gamma = 1$.

本文总结了连续 DR 次模函数最大化问题及相关变形问题的现有近似算法和性能结果. 具体的数学模型为

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

其中函数 $f(\boldsymbol{x})$ 是光滑 [1)] 的 DR 次模函数 (或光滑的 γ 弱次模函数), 集合 X 为可行域.

本文也用到如下符号. 向量 $\mathbf{0}$ 为 Euclid 空间的全零向量. B^n 为 n 维单位球, S^{n-1} 为 $n-1$ 维单位球面. 给定向量 $\boldsymbol{x}$ 和闭集 X, $\Pi_X(\boldsymbol{x})$ 表示向量 $\boldsymbol{x}$ 在集合 X 上的投影. $\boldsymbol{x}^*$ 表示优化问题的最优解, $f(\boldsymbol{x}^*)$ 或 OPT 表示最优值.

3 连续 DR 次模最大化问题的经典算法

本节介绍连续 DR 次模函数最大化问题的下界结果以及求解该问题的两类经典算法, 即 Frank-Wolfe 型算法和双贪婪型算法.

3.1 近似难度

在下闭凸集合 [2)] 上最大化单调连续 DR 次模函数是 NP (non-deterministic polynomial) 难的. 对于任意的 $\epsilon > 0$, 除非 RP = NP, 该问题无法在多项式时间内达到 $1 - 1/\mathrm{e} + \epsilon$ 的近似比 [3]. 箱型约束下非单调递增的 DR 次模函数最大化问题也被证明是 NP 难的 [3], 这意味着除非 RP = NP, 对于任意的 $\epsilon > 0$, 不存在多项式时间的$(1/2 + \epsilon)$- 近似算法. 从算法的收敛速率 [3)] 角度, Hassani 等 [26] 提出, 对于单调递增的连续 DR 次模最大化问题, 没有比 $O(1/\epsilon^2)$ 收敛速率更好的一阶算法. 对于在线模型下连续 DR 次模最大化问题, 当目标函数是一维函数时, 该问题可被视为在线凸优化问题. Hazan [30] 得到任何在线凸优化算法最坏情形下的遗憾界 [4)] 均为 $\Omega(\sqrt{T})$, 其中 T 为时间界限.

3.2 Frank-Wolfe 型算法

Bian 等 [3] 对连续 DR 次模最大化问题提出了 Frank-Wolfe 型近似算法. 该算法的设计思想来源于集合次模最大化问题的连续贪婪算法 [55] 和凸优化中经典的 Frank-Wolfe 型算法 [21, 31]. 假设目标函数 f 是单调递增的, 且可行域 X 为下闭凸集. 同时, 假设算法的总迭代次数为 K. 初始解取 $\boldsymbol{x}^1 = \mathbf{0}$. 在第 k ($k \in [K]$) 步迭代时, 算法在当前解 $\boldsymbol{x}^k$ 上寻找使得目标值上升的方向 $\boldsymbol{v}^k$, 即沿此方向既能保证解的可行性, 又能使目标函数 $f(\boldsymbol{x})$ 最大化. 为此, 算法对目标函数进行线性近似, 并通过如下线性优化问题来确定上升方向:

$$\boldsymbol{v}^k = \arg\max_{\boldsymbol{v} \in X} \langle \nabla f(\boldsymbol{x}^k), \boldsymbol{v} \rangle. \tag{3.1}$$

1) 光滑指的是连续优化领域中函数的光滑性, 函数 f 具有 L- 光滑性 (L-smoothness) 是指, 对任意向量 $\boldsymbol{x}, \boldsymbol{y} \in \mathrm{dom}(f)$, 函数 f 满足 $\|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x})\| \leqslant L\|\boldsymbol{y} - \boldsymbol{x}\|$.

2) 与凸集 X 和下界 $\boldsymbol{u} \in X$ 相关联的下闭凸集合要满足以下条件:
(i) 对于任意 $\boldsymbol{y} \in X$, 有 $\boldsymbol{u} \leqslant \boldsymbol{y}$;
(ii) 对于任意 $\boldsymbol{y} \in X$ 和 $\boldsymbol{x} \in \mathbb{R}^n$, 若 $\boldsymbol{u} \leqslant \boldsymbol{x} \leqslant \boldsymbol{y}$, 则 $\boldsymbol{x} \in X$.

3) 收敛速率指的是, 为了实现期望意义下的次最优性 $\mathbb{E}[\alpha f(\boldsymbol{x}^*) - f(\boldsymbol{x}^k)]$, 算法所需要访问的随机梯度的次数, 其中 α 为近似比, $\boldsymbol{x}^*$ 为最优解, $\boldsymbol{x}^k$ 为随机算法输出解.

4) 遗憾界的定义见第 4.2 小节.

进一步地, 通过如下迭代更新当前解:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \frac{1}{K}\boldsymbol{v}^k. \tag{3.2}$$

初始解 $\boldsymbol{x}^1 = \boldsymbol{0}$ 和步长 $1/K$ 的设定保证了最终解仍在凸集 X 上. Bian 等 [3] 证明了此算法可以达到 $1 - 1/\mathrm{e}$ 的近似比和次线性收敛速度.

在此算法的基础上, Bian 等 [2] 对非单调次模函数最大化问题提出了 "非单调 Frank-Wolfe 变形算法". 在此模型中仍然假设可行域 X 为下闭凸集. 但此时算法不能简单地沿上升方向以 $1/K$ 为步长更新当前解 (3.2), 这是因为目标函数值并不总是随解的增大而增加. 此时, 解的更新方式需要更为谨慎. 在搜索上升方向时, 算法会在比 (3.1) 更受限的范围内进行, 即由

$$\boldsymbol{v}^k = \arg \max_{\boldsymbol{v}\in X, \boldsymbol{v}\leqslant \overline{\boldsymbol{u}}-\boldsymbol{x}^k} \langle \boldsymbol{v}, \nabla f(\boldsymbol{x}^k)\rangle$$

得到上升方向, 其中向量 $\overline{\boldsymbol{u}}$ 为目标函数 f 定义域的上界. 此搜索中加入的额外约束 $\boldsymbol{v} \leqslant \overline{\boldsymbol{u}} - \boldsymbol{x}^k$ 旨在避免解的增长速度过快, 从而确保解的稳定性. 借助于这种更为保守的设置, Bian 等成功证明了非单调 Frank-Wolfe 变形算法能够达到 $1/\mathrm{e}$ 的近似比. 此外, Bian 等还设计了一个两阶段的非单调 Frank-Wolfe 型算法, 该算法需使用两次非凸 Frank-Wolfe 求解器来寻找稳定点, 并选择函数值较大的解. 这种方法提供了 $1/4$ 近似保证和 $1/\sqrt{k}$ 收敛率.

Dürr 等 [15] 研究了凸集上的非单调连续 DR 次模最大化问题. 这类模型具有更广泛的适用性, 其包含了上闭凸集、凸锥、覆盖和包装混合约束等. Dürr 等的算法也受到了 Frank-Wolfe 算法的启发. 算法的核心在于设计解的更新策略. 算法共执行 K 步, 在第 k ($k \in [K]$) 步时, 如 (3.1) 所述, 在确定上升方向 $\boldsymbol{v}^k$ 后, Dürr 等引入了一种全新的步长策略

$$\boldsymbol{x}^{k+1} = (1 - \eta_k)\boldsymbol{x}^k + \eta_k \boldsymbol{v}^k,$$

其中 $\eta_k = (\ln 3)/(2kH_K)$, H_K 为调和级数的前 K 项和. 这种特殊的步长设定, 使得算法可以达到 $(1 - \min_{\boldsymbol{x}\in X} \|\boldsymbol{x}\|_\infty)/(3\sqrt{3})$ 的近似比和 $O(n/(\ln^2 K))$ 的收敛率. 特别地, 当可行域包含原点时, 近似比可提升至 $1/(3\sqrt{3})$. 随后, Du 等 [13] 成功将该问题的近似比提升到了 $1/4$.

对各类 DR 次模最大化问题, Du[12] 基于 Lyapunov 函数提出了普适的两阶段框架来设计和分析算法. 在该框架的第一阶段, 利用 Lyapunov 函数设计连续时间算法, 同时可得到该算法的近似比. Du[12] 首先定义了 Lyapunov 函数

$$E(\boldsymbol{x}(t)) := a_t f(\boldsymbol{x}(t)) - b_t f(\boldsymbol{x}^*),$$

其中 t ($t \in [0, T]$) 为时间变量, $\boldsymbol{x}(t)$ 为时刻 t 的算法输出解, a_t 和 b_t 为两组待定的参数. Du 希望通过设置参数 a_t 和 b_t 的形式, 使得函数 $E(\boldsymbol{x}(t))$ 关于时间变量 t 的导数恒大于等于 0, 即使得 $E(\boldsymbol{x}(t))$ 关于时间变量 t 是单调增函数. 进而由

$$E(\boldsymbol{x}(T)) = a_T f(\boldsymbol{x}(T)) - b_T f(\boldsymbol{x}^*) \geqslant E(0) = a_0 f(\boldsymbol{x}(0)) - b_0 f(\boldsymbol{x}^*)$$

可以推导出算法输出解 $\boldsymbol{x}(T)$ 的质量为

$$f(\boldsymbol{x}(T)) \geqslant \frac{b_T - b_0}{a_T} f(\boldsymbol{x}^*) + \frac{a_0}{a_T} f(\boldsymbol{x}(0)) \geqslant \frac{b_T - b_0}{a_T} f(\boldsymbol{x}^*).$$

由此, 便证明出算法解可以达到 $(b_T - b_0)/a_T$ 近似的效果. 而参数 a_t 和 b_t 的设置直接影响了近似比的大小. 在理论分析里, Du 尝试使用当前解 $\boldsymbol{x}(t)$ 的信息来估算最优值 $f(\boldsymbol{x}^*)$ 的上界, 并将该上界用于 Lyapunov 函数导数的计算. 通过确保导数始终非负, 推导得出参数 a_t 和 b_t 需要满足的条件. 该条件即为算法设计中得到近似比的充分条件, 是算法设计步骤的关键. 第二阶段里, 为了尽可能保持近似比的稳定性, Du 将连续时间算法离散化, 并对离散时间算法的近似比和时间复杂度进行了验证. 值得一提的是, 这套基于 Lyapunov 函数的算法设计和分析框架包含了文献 [2,3] 中的结果, 并且为新算法的设计和分析提供指导思想.

上述 Frank-Wolfe 型算法均依赖于目标函数的精确梯度. 这类算法利用梯度进行线性近似, 并通过求解该线性优化问题获得目标函数的上升方向. 然而, 在一些问题中, 梯度的计算缓慢且困难, 如图形模型推理 [56] 和结构预测 [48,52] 等领域. 甚至在一些特殊情形下, 如自然语言处理或图像分类的超参数转换 [47,54], 无法得到一阶信息. 针对这一现状, Chen 等 [9] 考虑能否仅使用目标函数的零阶信息, 同时避免使用投影算子来设计算法. 首先面临的问题是如何利用已有的信息近似出函数的一阶信息. 为此, Chen 等 [9] 首先采用光滑化技巧 [20], 通过利用函数的零阶信息来构建无偏梯度估计器. 具体而言, 函数 f 在某点 $\boldsymbol{x}$ 的光滑化取值, 是以点 $\boldsymbol{x}$ 为球心、δ 为半径的球内的点对应的函数值的平均. 确切地, 函数 f 在某点 $\boldsymbol{x}$ 处的光滑化函数定义为

$$f_\delta(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{v}\sim B^n}[f(\boldsymbol{x} + \delta\boldsymbol{v})], \tag{3.3}$$

其中 δ 为光滑化半径, $\boldsymbol{v} \sim B^n$ 表示向量 $\boldsymbol{v}$ 服从 n 维单位球上的均匀分布. 根据 Stoke 定理 [30], 可以计算出该光滑函数的梯度为

$$\nabla f_\delta(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{u}\sim S^{n-1}}\left[\frac{n}{\delta} f(\boldsymbol{x} + \delta\boldsymbol{u})\boldsymbol{u}\right], \tag{3.4}$$

其中 $\boldsymbol{u} \sim S^{n-1}$ 表示向量 $\boldsymbol{u}$ 服从 $n-1$ 维单位球面上的均匀分布. 然而, 如果直接使用光滑化函数 $f_\delta(\boldsymbol{x})$, 可能会产生 $O(n^2/\delta^2)$ 的方差, 从而影响 Frank-Wolfe 型算法的收敛性. 因此 Chen 等借鉴了 Mokhtari 等 [38] 提出的动量法, 结合无偏梯度估计器以降低方差. 对下闭凸集约束下的单调增连续 DR 次模函数优化问题, Chen 等设计了黑箱连续贪婪算法, 经过 $O(n/\epsilon^3)$ 次函数值估计后, 该算法能够实现 $(1-1/\mathrm{e})\mathrm{OPT} - \epsilon$ 的近似保证.

3.3 双贪婪型算法

双贪婪型算法主要适用于目标函数非单调且约束为箱型的优化模型. 双贪婪型算法最早用于求解非单调集合次模最大化问题 [4–6], 随后又应用于整数格上的非单调次模最大化问题 [23,42]. 受集合次模 [4] 和整数格次模 [23] 双贪婪算法的启发, Bian 等 [3] 对连续 DR 次模最大化问题设计了双贪婪型算法. 算法分别同时从箱约束的上界和下界开始向中间移动. 算法共执行 n 次迭代, 在第 i ($i \in [n]$) 次迭代里, 算法仅考虑解的第 i 个分量的变化情形. 此时所研究的问题可被视为一维优化问题. 用贪婪的思想, 算法从下界向上、上界向下两个方向分别得到更新解后函数值的增益, 再从中选择增益最大的解作为第 i 个分量的最终值. 经过 n 次迭代后, 沿两个方向得到的解重合, 此解即为双贪婪算法的输出解. Bian 等通过对比每一次迭代前后函数值的变化, 证明了该算法的近似比为 1/3, 误差界为 $(4n\delta)/3$, 其中 δ 为求解一维子问题的误差. Niazadeh 等 [41] 提出的二元搜索连续双贪婪算法将该问题的近似比提高到了 1/2.

4 连续 DR 次模最大化问题的重要变形

本节介绍连续 DR 次模最大化问题的两类重要变形, 即随机连续 DR 次模最大化问题和在线连续 DR 次模最大化问题, 并介绍它们的最新研究进展.

4.1 随机模型

在实际应用中, 优化问题的目标函数常常受外部不确定因素的影响. 为减弱不确定因素的影响, 这些问题的目标函数通常建模成期望的形式, 相应地称为这类问题为随机优化问题. 随机优化问题允许决策者考虑不确定性因素, 以确保最终解决方案更符合实际需求. 此类问题的主要挑战在于目标函数中涉及的随机变量 ξ 的概率分布通常未知, 无法直接计算随机函数的期望值, 从而无法精确获取目标函数 f 及其梯度 ∇f. 因此, 为随机优化问题设计出有次线性收敛速度的近似算法是极具挑战的. 本节介绍随机 DR 次模最大化问题的相关研究结果. 随机 DR 次模最大化问题研究的目标函数 f 同时具有连续 DR 次模性和不确定性. 进一步地, 根据随机变量 ξ 与决策变量 $\boldsymbol{x}$ 之间的依赖关系, 问题又可分为遗忘随机模型和非遗忘随机模型.

4.1.1 遗忘随机模型

遗忘随机模型是指随机变量 ξ 所服从的分布独立于决策变量 $\boldsymbol{x}$. 这类问题的数学模型为

$$\max_{\boldsymbol{x}\in X} f(\boldsymbol{x}) := \mathbb{E}_{\xi\in\Xi}[F(\boldsymbol{x},\xi)],$$

其中 $\xi\in\Xi$ 表示随机变量 ξ 服从样本空间 Ξ 中某一未知的概率分布.

Hassani 等 [27] 研究了带一般凸集约束的连续单调 DR 次模最大化问题, 并提出了随机投影梯度上升算法. 此算法与要求初始点位于原点的 Frank-Wolfe 型算法不同, 它允许任何一个可行解作为起始点. 该算法设定迭代总次数为 T, 在第 t ($t\in[T]$) 步迭代时, 一般通过

$$\boldsymbol{x}_{t+1} = \Pi_X(\boldsymbol{x}_t + \mu_t\nabla f(\boldsymbol{x}_t))$$

得到当前解, 其中 $\Pi_X(\boldsymbol{x})$ 表示向量 $\boldsymbol{x}$ 在集合 X 上的投影, μ_t 为学习率, $\nabla f(\boldsymbol{x}_t)$ 表示梯度 ∇f 在 $\boldsymbol{x}_t$ 处的值. 由于随机模型的特性, 无法直接得到梯度值 $\nabla f(\boldsymbol{x}_t)$, Hassani 等利用无偏随机梯度值代替精确的梯度值

$$\boldsymbol{x}_{t+1} = \Pi_X(\boldsymbol{x}_t + \mu_t\boldsymbol{g}_t),$$

设计了随机投影梯度上升算法, 该算法经过 $O((L_2/\epsilon)+\sigma^2/\epsilon^2)$ 次迭代, 达到了 $(\mathrm{OPT}/2)-\epsilon$ 的近似效果, 其中 L_2 为目标函数在 l_2 范数意义下的光滑化系数, σ 为随机梯度的方差上界. 更一般地, 当目标函数为 γ 弱次模函数时, Hassani 等证明了算法可以达到 $(\gamma^2)/(\gamma^2+1)$ 的近似比. 需要注意的是, 算法的迭代复杂度依赖于目标函数的光滑化系数 L_2, 但对于特定函数, 如多线性扩展, 光滑化系数 L_2 与问题的维数 n 相关, 这对于高维优化问题而言并不理想. 为了克服这一缺陷, Hassani 等 [28] 基于 1- 强凸镜像映射 Φ [5)] 提出了随机镜像梯度上升算法. 该算法共执行 T 次, 初始点选取 $\boldsymbol{x}_1 = \arg\min_{\boldsymbol{x}\in X}\Phi(\boldsymbol{x})$,

5) 设 $D\subset\mathbb{R}^n$ 为开凸集. 如果映射 $\Phi: D\to\mathbb{R}$ 满足以下性质:
(i) Φ 严格凸且可微;
(ii) Φ 的梯度可取尽所有可能的值, 即 $\nabla\Phi(D)=\mathbb{R}^n$;
(iii) Φ 的梯度在 D 的边界处发散, 即 $\lim_{x\to\partial D}\|\nabla\Phi(x)\|=+\infty$,
则称其为镜像映射.

在第 t ($t\in[T]$) 次迭代时, 借助于梯度 $\nabla f(\boldsymbol{x}_t)$ 的无偏估计 $\boldsymbol{g}_t$, 向量 $\boldsymbol{y}_{t+1}$ 满足

$$\nabla\Phi(\boldsymbol{y}_{t+1})=\nabla\Phi(\boldsymbol{x}_t)+\mu_t\boldsymbol{g}_t.$$

通过镜像映射, 将向量 $\boldsymbol{y}_{t+1}$ 投影到集合 X 上, 以获得下一个迭代点

$$\boldsymbol{x}_{t+1}=\arg\min_{\boldsymbol{x}\in X}(\Phi(\boldsymbol{x})-\Phi(\boldsymbol{y}_{t+1})-\langle\nabla\Phi(\boldsymbol{y}_{t+1}),\boldsymbol{x}-\boldsymbol{y}_{t+1}\rangle).$$

算法最终从 $\{\boldsymbol{x}_t\mid t\in[T]\}$ 中随机均匀地选出一个解作为最终输出. Hassani 等推导出了, 该算法经过 $O((L_*/\epsilon)+(\sigma^2/\epsilon^2))$ 次迭代后可以得到 $(\mathrm{OPT}/2)-\epsilon$ 质量的解, 其中 L_* 为目标函数在任何范数意义下的光滑化系数, 如可以取 L_1 范数意义下的光滑系数.

鉴于在高维问题中投影算子的计算代价过高, Mokhtari 等 [38,39] 提出了随机连续贪婪方法, 该方法是连续贪婪算法 [55] 的随机版本. 针对目标函数是否单调的不同情形, Mokhtari 等设计了不同的近似算法, 并进行了理论分析. 在目标函数具有单调性时, 传统的连续贪婪算法需利用目标函数的精确梯度 $\nabla_{\boldsymbol{x}}\mathbb{E}_{\xi\in\Xi}[F(\boldsymbol{x},\xi)]$, 但在随机模型中, 算法仅能访问随机梯度 $\nabla\tilde{F}(\boldsymbol{x},\xi)$. 此时, 随机梯度并非直接替代真实梯度, 而需要借助于梯度平均技巧进行处理. 在算法的第 t ($t\in[T]$) 步, 通过递归形式构造向量

$$\boldsymbol{d}_{t+1}=(1-\rho_t)\boldsymbol{d}_t+\rho_t\nabla\tilde{F}(\boldsymbol{x},\xi)\tag{4.1}$$

估计梯度, 其中初始值设为 $\boldsymbol{d}_0=\boldsymbol{0}$. 利用此向量近似真实梯度, 并据此构造线性优化问题以寻找目标函数的上升方向 $\boldsymbol{v}_t$:

$$\boldsymbol{v}_t=\arg\max_{\boldsymbol{v}\in X}\langle\boldsymbol{d}_t,\boldsymbol{v}\rangle.\tag{4.2}$$

获得上升方向 $\boldsymbol{v}_t$ 后, 再结合步长 $1/T$ 完成解的迭代

$$\boldsymbol{x}_{t+1}=\boldsymbol{x}_t+\frac{1}{T}\boldsymbol{v}_t.\tag{4.3}$$

在理论分析中, Mokhtari 等给出的梯度估计误差的期望上界为

$$\mathbb{E}[\|\nabla f(\boldsymbol{x}_t)-\boldsymbol{d}_t\|^2]\leqslant\frac{Q}{(t+9)^{2/3}},\tag{4.4}$$

其中 Q 为常数. 当目标函数为单调连续 DR 次模函数时, 算法可达到 $1-1/\mathrm{e}$ 的近似比与 $O(T^{-1/3})$ 的次线性收敛速度. 当目标函数为 γ 弱次模函数时, 在同样的收敛速度下, 近似比可推广至 $1-\mathrm{e}^{-\gamma}$. 对目标函数非单调情形, 算法的设计类同于离线模型, 上升方向

$$\boldsymbol{v}_t=\arg\max_{\boldsymbol{v}\in X,\boldsymbol{v}\leqslant\overline{\boldsymbol{u}}-\boldsymbol{x}_t}\langle\boldsymbol{d}_t,\boldsymbol{v}\rangle$$

需要满足额外约束 $\boldsymbol{v}\leqslant\overline{\boldsymbol{u}}-\boldsymbol{x}_t$, 其中 $\overline{\boldsymbol{u}}$ 为集合 X 的上界. 确定了上升方向后, 解的迭代仍然遵循迭代公式 (4.3). Mokhtari 等得到了非单调随机连续贪婪算法可以达到 $1/\mathrm{e}$ 的近似比和 $O(T^{-1/3})$ 的次线性收敛速度. 综上, 无论是单调还是非单调情形, 其近似比均与确定性模型的结果相匹配.

对于一般凸闭集上的单调 γ 弱次模函数最大化问题, Zhang 等 [59] 利用辅助函数设计算法, 并进行理论分析. 这些辅助函数旨在提供比目标函数 f 本身更优的近似保证. 受到文献 [18,19] 的启发, Zhang 等引入一系列辅助函数, 它们在点 $\boldsymbol{x}$ 的梯度记为 $\nabla f(z\cdot\boldsymbol{x})$, 其中 $z\in[0,1]$. 通过求解因子揭示

优化问题, Zhang 等成功地筛选出了最优辅助函数 F. 辅助函数 F 在其稳定点能够达到 $1-\mathrm{e}^{-\gamma}$ 的近似效果, 并且只需要 $O(1/\epsilon^2)$ 次迭代即可实现, 从而显著提高了算法的效率和适用性.

上述优化问题的约束条件通常假设为凸集. 在某些情形下, 可以将部分约束刻画为函数形式, 从而更好地利用函数的特性 (如可微性和凸性) 来处理这些约束. 在实际优化问题中, 为了描述目标函数和约束函数在平均意义上的表现, 常将其建模为期望形式的函数. 基于这一思路, Ju 等 [32] 研究了目标函数和约束函数均为期望形式的随机优化模型, 其中目标为连续 DR 次模函数, 约束为凸函数. 受经典 Lagrange 方法的启发, Ju 等设计了一种全新的双层循环结构算法. 该算法能够在迭代过程中保持 Lagrange 乘子的上界为常数. 当目标函数具有单调性时, 算法可以达到 $1-1/\mathrm{e}$ 的近似比, 具有 $O(T^{-1/3})$ 的复杂度和 $O(T^{-1/3})$ 的约束违反度, 其中 T 为算法的总迭代步数. 而在目标函数不具单调性时, 算法的近似比为 $1/4$.

4.1.2 非遗忘随机模型

若随机变量 ξ 的分布依赖于决策变量 $\boldsymbol{x}$, 并且在优化过程中其分布可能发生变化, 具有上述特点的问题被称为非遗忘随机问题, 其数学模型为

$$\max_{\boldsymbol{x}\in X} f(\boldsymbol{x}) := \mathbb{E}_{\xi\sim p(\xi;\boldsymbol{x})}[F(\boldsymbol{x},\xi)],$$

其中 $\xi\sim p(\xi;\boldsymbol{x})$ 表示随机变量 ξ 服从概率分布 $p(\xi;\boldsymbol{x})$.

Zhang 等 [58] 为非遗忘随机问题设计了无投影的算法. 此算法可视为 Frank-Wolfe 方法的随机变形, 其特色在于每次迭代仅需要调用一次随机样本. 在非遗忘模型的随机算法设计中, 最重要的是如何对梯度变化 $\Delta^t := \nabla f(\boldsymbol{x}^t) - \nabla f(\boldsymbol{x}^{t-1})$ 作出无偏估计. Mokhtari 等 [39] 使用动量方法 (4.1) 来减少方差, 可以达到 $O(t^{-2/3})$ 的收敛速率 (4.4), 但这种梯度近似方法存在偏差. 因此, Zhang 等对 (4.1) 稍作修改, 添加了 Δ^t 的无偏估计项 $\tilde{\Delta}_t$, 形成新的更新规则

$$\boldsymbol{d}_t = (1-\rho_t)(\boldsymbol{d}_{t-1} + \tilde{\Delta}_t) + \rho_t \nabla\tilde{F}(\boldsymbol{x}_t;\xi_t),$$

并得出梯度估计向量 $\boldsymbol{d}_t$ 为真实梯度向量 $\nabla f(\boldsymbol{x}_t)$ 的无偏估计. 对一般凸集上的单调连续 DR 次模函数优化问题, 上述算法实现了最优的 $1-1/\mathrm{e}-\epsilon$ 的近似比和 $O(1/\epsilon^2)$ 的收敛速度.

在遗忘随机优化模型中, 算法通常仅对随机变量进行一次小批量采样, 来对两点梯度的差值作无偏估计. 然而当随机变量的分布依赖于决策变量时, 采取同样的处理就无法保证随机梯度向量估计的无偏性. 因此, 需要设法在不引入偏差的情形下, 来估计两点梯度的差值. 对非遗忘随机优化模型, Hassani 等 [25, 26] 提出了一种全新的方差缩减技术. 针对目标函数为单调、约束集合为一般凸集的情形, Hassani 等提出了一个二阶算法, 称为随机连续贪婪 ++ 算法. 算法迭代数为 T. 在第 1 次循环里, 算法基于初始解 $\boldsymbol{x}^1=\boldsymbol{0}$ 和随机分布 $p(\xi;\boldsymbol{x}^1)$ 对随机变量 ξ 进行小批量采样 $\mathcal{M}_1$, 获得该点处的随机梯度估计 $\boldsymbol{g}^1 = \nabla\tilde{F}(\boldsymbol{x}^1;\mathcal{M}_1)$, 利用该随机梯度来近似真实梯度 $\nabla f(\boldsymbol{x}^1)$, 进而构造线性优化问题以寻找上升方向

$$\boldsymbol{v}^1 = \arg\max_{\boldsymbol{v}\in X}\langle \boldsymbol{g}^1, \boldsymbol{v}\rangle.$$

再按照 Frank-Wolfe 型算法更新解的思想 (4.3) 获得当前解 $\boldsymbol{x}^2$. 在第 t ($t\in[T]$) 步迭代里, 算法随机均匀地选取 $\boldsymbol{x}^{t-1}$ 以及 $\boldsymbol{x}^t$ 连线上的某点

$$\boldsymbol{x}(a) := a\boldsymbol{x}^t + (1-a)\boldsymbol{x}^{t-1}.$$

设该点处的随机变量为 $\xi(a)$, 服从的概率分布为 $p(\xi(a);\boldsymbol{x}(a))$, 小批量采样为 $\mathcal{M}$, 构造积分

$$\int_0^1 \nabla^2 F(\boldsymbol{x}(a))da$$

的无偏估计为

$$\tilde{\nabla}_t^2 = \frac{1}{|\mathcal{M}|}\sum_{(a,\xi(a))\in\mathcal{M}} \tilde{\nabla}^2 F(\boldsymbol{x}(a);\xi(a)).$$

如此, 就可以获得梯度差 $\Delta^t := \nabla f(\boldsymbol{x}^t) - \nabla f(\boldsymbol{x}^{t-1})$ 的近似

$$\tilde{\Delta}^t := \tilde{\nabla}_t^2(\boldsymbol{x}^t - \boldsymbol{x}^{t-1}). \tag{4.5}$$

通过上一步的梯度估计获得当前步的梯度估计

$$\boldsymbol{g}_t = \boldsymbol{g}_{t-1} + \tilde{\Delta}^t.$$

之后, 便与遗忘随机模型中寻找当前解的上升方向 (4.2) 和更新当前解的方式 (4.3) 类似. Hassani 等证明了, 在使用 $O(1/\epsilon^2)$ 次随机梯度估计后, 随机连续贪婪 ++ 算法可以达到 $(1-1/\mathrm{e})\mathrm{OPT}-\epsilon$ 的近似保证. 这个结果提高了遗忘随机模型下 [39] 的收敛性结果 $O(1/\epsilon^3)$. 相比于文献 [39], 此处仅要求目标函数梯度具有 Lipschitz 光滑性, 文献 [26] 还额外要求目标函数的随机 Hessian 信息具有 Lipschitz 光滑性. 此外, Hassani 等还研究了当目标函数非单调、约束集合为下闭凸集时的情形, 并提出了度量随机连续贪婪算法. 该算法的设计思想与目标函数具有单调性时的设计基本相同, 区别主要在于无偏估计步骤 (4.5) 和寻找上升方向 $\boldsymbol{v}^t$ 的步骤. Hassani 等同样证明了, 在这种情形下, 算法经过 $O(1/\epsilon^2)$ 次迭代可以达到 $(1/\mathrm{e})\mathrm{OPT}-\epsilon$ 的近似保证.

4.2 在线模型

在线模型是指, 优化问题的目标函数一开始并不确定, 而是通过决策者 (即算法视角) 的一系列决策来逐步揭露的. 在线次模最大化可被视为决策者同对手进行的一场博弈. 设整数 T 是时间界限, 即迭代的最大次数; 集合 X 是决策空间, 每一轮迭代中决策者需要从 X 中选取决策 $\boldsymbol{x}$; 集合 $\mathcal{F}$ 是连续 DR 次模函数簇. 在第 t ($t\in[T]$) 轮迭代时, 决策者选取 $\boldsymbol{x}_t\in X$. 在决策者作出选择之后, 对手会从函数簇 $\mathcal{F}$ 中选出任意一个连续 DR 次模函数 f_t. 相应地, 决策者会收到奖励 $f_t(\boldsymbol{x}_t)$. 设计在线模型算法的核心在于如何巧妙地选择 $\boldsymbol{x}_t$. 得到 $\boldsymbol{x}_t$ 后如何评估这个算法性能的优劣? 这里引入 "遗憾界" (regret) 来评价算法的性能. α- 遗憾界的定义如下:

$$\mathrm{Regret}(\alpha,T) := \alpha\sum_{t=1}^{T} f_t(\boldsymbol{x}^*) - \sum_{t=1}^{T} f_t(\boldsymbol{x}_t),$$

其中 α 是近似比, $\boldsymbol{x}^*$ 是离线模型的最优解, 即

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in X}\sum_{t=1}^{T} f_t(\boldsymbol{x}).$$

遗憾界刻画了算法解的质量同最优解质量的差距, 且为近似比 α 和迭代步数 T 的函数. 由于 $\alpha\in(0,1)$, 所以遗憾界主要受限于 T. 若遗憾界是关于 T 的次线性函数, 则有 $\mathrm{Regret}(\alpha,T)=o(T)$. 在算法设计中, 期望尽可能地最大化近似比, 同时使得遗憾界至少是关于 T 的次线性函数.

根据在第 t ($t \in [T]$) 轮迭代中对手反馈的信息不同, 在线模型可进一步细分为全信息模型和强盗模型. 前者是对手能够提供整个函数 f_t 的信息, 后者是对手仅反馈在决策点处的值 $f_t(\boldsymbol{x}_t)$. 这两种模型下, 可供算法利用的信息量有所不同, 前者算法能够利用函数在不同点处的值, 后者算法只能基于决策点处的值 $f_t(\boldsymbol{x}_t)$. 下面详细介绍全信息模型和强盗模型下的成果.

4.2.1 全信息模型

Chen 等 [8] 系统地探索了在线次模最大化模型. 首先, 他们研究了一般凸集约束下的在线单调连续 DR 次模最大化问题. 当算法可以使用函数的精确梯度信息时, 受传统 Frank-Wolfe [21] 思想和 Meta 思想 [50] 的启发, Chen 等提出了 Meta-Frank-Wolfe 算法. 如第 3 节所言, 离线模型下的 Frank-Wolfe 型算法首先需要求解 K 个目标函数为线性函数, 约束集合为一般凸集的子问题, 这里 K 为迭代次数. 在在线模型下的第 t 轮中, 若算法尝试效仿离线模型下的策略 (3.1), 则同样需要求解 K 个子问题, 其中第 k ($k \in [K]$) 个子问题的目标函数的线性系数是当前目标函数在当前解处的梯度值 $\nabla f_t(\boldsymbol{x}_t^k)$. 然而在在线模型中, 由于决策 $\boldsymbol{x}_t$ 尚未作出, 故奖励函数 f_t 还处于未知状态, 此时需要寻找 $\nabla f_t(\boldsymbol{x}_t^k)$ 的有效替代品. 在此场景下, Chen 等选择调用已广泛使用的在线线性求解器 RFTL (regularized follow the leader) 来处理, 该求解器的思想是使用历史奖励函数在该点处的平均梯度值并结合正则项来获得上升方向 $\boldsymbol{v}_t^k$, 再效仿离线模型的更新规则

$$\boldsymbol{x}_t^{k+1} = \boldsymbol{x}_t^k + \frac{1}{K}\boldsymbol{v}_t^k,$$

即可获得第 t 轮的决策

$$\boldsymbol{x}_t = \boldsymbol{x}_t^{K+1}.$$

Chen 等证明了 Meta-Frank-Wolfe 算法达到 $(1-1/\mathrm{e})$- 近似的遗憾界为 $O(\sqrt{T})$. 但在某些场景下, 决策者只能获取目标函数的随机梯度而非精确梯度. 针对这种设定, Chen 等提出了在线梯度上升算法, 该算法更多地利用前一轮函数 f_{t-1} 的信息来获得第 t 轮的解 $\boldsymbol{x}_t$. 具体地, 在第 t ($t \in [T]$) 轮迭代里, 决策者先执行上一轮迭代中选出的 $\boldsymbol{x}_t$, 并获得这一轮迭代的奖励函数值 $f_t(\boldsymbol{x}_t)$ 和梯度的随机无偏估计 $\boldsymbol{g}_t$, 然后使用自适应步长 η_t 进行梯度投影, 得到下一轮的决策

$$\boldsymbol{x}_{t+1} = \Pi_X(\boldsymbol{x}_t + \eta_t \boldsymbol{g}_t).$$

当目标函数为更一般的 γ 弱次模函数时, Chen 等证明了 Meta-Frank-Wolfe 算法达到 $\gamma^2/(\gamma^2+1)$- 近似的遗憾界为 $O(\sqrt{T})$. 特别地, 当 $\gamma = 1$ 时, 即当目标函数是连续 DR 次模函数时, 近似比可以达到 $1/2$. 注意到, 上述两种方法中, 前者在使用在线线性求解器时需要调用 $K = O(\sqrt{T})$ 次梯度信息, 后者在每轮迭代里需要计算投影算子. 对于大规模的应用场景, 这两者的成本都过于昂贵. 基于此, Chen 等 [7] 提出了 Meta-Frank-Wolfe 变形算法. 与求解 K 个子规划问题的过程不同, 他们使用之前步骤里记录的梯度估计的平均值和上一轮观测到的随机梯度的综合信息来代替真实梯度值, 即使在这种较弱的设定下, 算法同样达到了 $1-1/\mathrm{e}$ 的近似比、$O(\sqrt{T})$ 的遗憾界. 前面介绍的 3 个算法都需要调用在线线性求解器, Chen 等 [7] 进一步提出不需要调用在线线性求解器的 One-Shot Frank-Wolfe 算法. 在每轮迭代里, 该方法仅需要对梯度进行一次随机估计. 由于算法使用了更少的信息, 在保持相同近似比的前提下, 遗憾界增加到 $O(T^{2/3})$.

当目标函数非单调时, Thắng 和 Srivastav [53] 针对 3 种不同类型的约束集合 (即超立方体约束)、下闭凸集和一般凸集进行了研究, 分别得到了 $1/2$ 的近似比和 $O(\sqrt{T})$ 的遗憾界、$1/\mathrm{e}$ 的近似比和

$O(T^{1/2})$ 的遗憾界、$(1-\min_{\boldsymbol{x}\in X}\|\boldsymbol{x}\|_\infty)/(3\sqrt{3})$ 的近似比和 $O(T/(\ln T))$ 的遗憾界. 针对约束集合为下闭凸集的情形, 他们研究了在线非凸优化问题, 即在线合取问题. 在第 t 轮迭代时, 算法利用已知信息 $\boldsymbol{c}^t$ 来选取 $\boldsymbol{x}^t\in X$; 随后, 对手返回 $\boldsymbol{a}^t\in\mathbb{R}^n$, 算法获取奖励 $\langle \boldsymbol{a}^t, \boldsymbol{c}^t\vee\boldsymbol{x}^t\rangle$. 针对约束集合 X, 他们定义了 $[0,1]^n$ 上一个足够稠密的格结构 $\mathcal{L}$, 确保集合 X 中的点都可以在 $\mathcal{L}$ 上作近似. 接着, 他们将集合 X 上的所有格点都提升到高维空间, 使其在新定义的空间中形成单位超立方体的顶点的子集. 而算法获得的奖励函数 $\langle \boldsymbol{a}^t, \boldsymbol{c}^t\vee\boldsymbol{x}^t\rangle$ 恰巧可以转换为新空间的线性函数. 然后, 他们设计的在线合取问题的算法在每轮迭代中, 先在高维空间中更新解, 再将它投影回原始空间. 利用连续 DR 次模的性质和离线模型下的技巧 [15], 获得算法的近似比和遗憾界. 针对约束集合为超立方体约束, 他们也采取类似的思想, 将 $[0,1]^n$ 上的 DR 次模最大化问题转化为高维空间中超立方体顶点上的离散次模函数的最大化问题. 他们将 $[0,1]^n$ 上的格点对应于 $\{0,1\}^{n\times\log T}$ 中的顶点, 证明了转化后的函数为集合次模函数. 利用集合次模的在线算法 [44] 和舍入技巧, 便可得到最终的近似比. 针对一般凸集上的约束类型, 他们设计了 Meta-Frank-Wolfe 算法, 算法使用自适应步长, 得到了跟离线模型统一的近似比结果. Dürr 等 [14] 也研究了一般凸集上的 DR 次模最大化问题, 并使用在线梯度上升算法实现了 1/4 近似和 $O(\sqrt{T})$ 的遗憾界. 考虑到投影算子的昂贵, 基于文献 [12] 的 Lyapunov 函数分析框架, Feng 等 [17] 设计了 Frank-Wolfe 型算法, 也达到 1/4 近似和 $O(\sqrt{T})$ 的遗憾界.

可行域 X 除了被建模为集合约束的形式, 也可能含有额外的约束, 如凸集合约束和额外的函数型约束. Sadeghi 和 Fazel [45] 研究了凸集合约束 C 及长期预算限制的约束类型. 在第 t ($t\in[T]$) 轮时, 算法执行决策 $\boldsymbol{x}_t\in C$、目标函数 $f_t(\boldsymbol{x}_t)$ 和预算 $\langle \boldsymbol{p}_t, \boldsymbol{x}_t\rangle$ 会被揭露. 设 B_T 为总预算, 每轮循环中揭露的预算限制函数为 $\boldsymbol{g}_t(\boldsymbol{x}) := \langle \boldsymbol{p}_t, \boldsymbol{x}\rangle - B_T/T$. 先前的工作已经表明, 如果目标和约束函数的揭露具有敌对性, 则无法同时实现次线性遗憾和总约束违反. 因此, Sadeghi 和 Fazel 修改了遗憾的概念. 确切地, 窗口长度为 W 的遗憾界为

$$\mathrm{Regret}_W(\alpha,T) := \alpha\sum_{t=1}^{T} f_t(\boldsymbol{x}^*_W) - \sum_{t=1}^{T} f_t(\boldsymbol{x}_t),$$

其中

$$\boldsymbol{x}^*_W = \arg\max_{\boldsymbol{x}\in\mathcal{X}_W}\sum_{t=1}^{T} f_t(\boldsymbol{x}),$$
$$\mathcal{X}_W = \left\{\boldsymbol{x}\in C:\ \sum_{\tau=t}^{t+W-1} g_\tau(\boldsymbol{x})\leqslant 0,\ 1\leqslant t\leqslant T-W+1\right\}.$$

Sadeghi 和 Fazel 提出了在线鞍点混合梯度上升算法, 设步长为 μ, 调节参数为 δ, 构造 Lagrangian 函数为

$$\mathcal{L}_t(\boldsymbol{x}_t,\boldsymbol{\lambda}_t) = f_t(\boldsymbol{x}_t) - \boldsymbol{\lambda}_t\boldsymbol{g}_t(\boldsymbol{x}_t) + \frac{\delta\mu}{2}\boldsymbol{\lambda}_t^2.$$

在第 t ($t\in[T]$) 轮迭代里, 算法将进行 K 次内循环, 每次循环均执行一次在线梯度上升算法. 在第 k ($k\in[K]$) 次内循环中, 算法先执行动作 $\boldsymbol{v}_t^k$, 并由在线梯度上升算法计算得到奖励 $\langle \boldsymbol{v}_t^k, \nabla_{\boldsymbol{x}}\mathcal{L}_t(\boldsymbol{x}_t^k,\boldsymbol{\lambda}_t)\rangle$. 然后, 算法再沿方向 $\boldsymbol{v}_t^k$ 更新当前解, 步长为 $1/K$,

$$\boldsymbol{x}_t^{k+1} = \boldsymbol{x}_t^k + \frac{1}{K}\boldsymbol{v}_t^k.$$

完成所有内循环后, 输出第 t 轮的最终解 $\boldsymbol{x}_t = \boldsymbol{x}_t^{K+1}$. 同时, 更新乘子

$$\boldsymbol{\lambda}_{t+1} = (\boldsymbol{\lambda}_t - \mu \nabla_{\boldsymbol{\lambda}} \mathcal{L}_t(\boldsymbol{x}_t, \boldsymbol{\lambda}_t)) \vee \mathbf{0},$$

以便下轮循环时使用. 此外, 更新上升方向 $\boldsymbol{v}_{t+1}^k$ 为

$$\boldsymbol{v}_{t+1}^k = \Pi_X(\boldsymbol{v}_t^k + \mu \nabla_{\boldsymbol{x}} \mathcal{L}_t(\boldsymbol{x}_t^k, \boldsymbol{\lambda}_t)).$$

Sadeghi 和 Fazel 证明了, 当窗口长度 $W = T^{1-\epsilon}$, 算法可达到 $1-1/\mathrm{e}$ 近似、$O(T^{1-\epsilon/2})$ 的遗憾界和 $O(T^{1-\epsilon/4})$ 的约束违反度. Raut 等 [43] 也研究了随机累积约束的在线连续 DR 次模最大化问题, 并提出了在线 Lagrangian-Frank-Wolfe 算法. 该算法可达到 $(1-1/\mathrm{e})$- 近似、$O(T^{3/4})$ 遗憾界和 $O(T^{1/2})$ 的约束违反度. Sadeghi 等 [46] 主要研究了在线凸函数约束类型的优化问题. 他们设计了同时适用于对抗性机制、动态机制和随机约束机制这 3 种模型的算法. 对于对抗性机制, 当窗口长度 $W = T^{1-\epsilon}$, 算法可实现 $1-1/\mathrm{e}$ 近似、$O(T^{1-\epsilon/2})$ 的遗憾界和 $O(T^{1-\epsilon/2})$ 的约束违反度. 对于动态机制, 算法可实现 $1-1/\mathrm{e}$ 近似、$O(\sqrt{TP_T^*})$ 的遗憾界和 $O(\sqrt{TP_T^*})$ 的约束违反度, 其中

$$P_T^* := \sum_{t=1}^{T-1} \|\boldsymbol{x}_{t-1}^* - \boldsymbol{x}_t^*\|.$$

对于随机约束机制, 算法可实现 $1-1/\mathrm{e}$ 近似、$O(\sqrt{T})$ 的遗憾界和 $O(\sqrt{T})$ 的约束违反度.

4.2.2 强盗模型

在全信息模型下, 算法设计允许在每轮迭代中多次获取目标函数 f_t 在不同点的函数值. 然而, 在强盗模型的设定下, 每轮迭代中算法仅能获取一个点处的函数值, 即该轮的决策点. 为了降低访问目标函数 f_t 及其梯度的调用频率, 算法采用分组策略, 使用这组目标函数的平均函数值作为该组的代表. 鉴于在强盗模型下点的选择要非常精细, 不仅要确保其函数值足够大, 又要确保该点的函数值经过变换后 (如步骤 (3.3) 和 (3.4) 所示) 能够有效近似目标函数的梯度值, 因此, 在选择点时需要在这两者间进行平衡. 算法将每组函数再分成两部分, 分别进行开发和探索两种处理. 具体地, 对于每组里的大部分函数, 优先考虑选择使得函数值可以达到近似保证的点; 对于剩余部分的函数, 则选择使得随机梯度近似表现效果良好的点. 为确保算法近似比能达到离线模型下的标准, 算法仍采用 Frank-Wolfe 思想进行当前解的迭代更新.

Zhang 等 [57] 基于上述方法, 研究了强盗模型下的在线 DR 次模最大化问题. 他们结合文献 [20] 中的光滑化技术 (3.3)、(3.4) 以及文献 [38,39] 中的方差减小技术 (4.1), 为下闭凸集上的单调连续 DR 次模最大化问题设计了 Bandit-Frank-Wolfe 算法, 实现了 $1-1/\mathrm{e}$ 的近似比和 $O(T^{8/9})$ 的遗憾界. 此外, 他们还研究了一般凸集上的单调 DR 次模最大化问题, 实现了 $1-1/\mathrm{e}$ 近似比和 $O(T^{\frac{5\min\{1,\gamma\}+3}{6\min\{1,\gamma\}+3}})$ 的遗憾界, 其中参数 γ 是与约束集合 X 相关的参数. Zhang 等 [60] 进一步拓展了该优化模型, 研究了下闭凸集上的非单调连续 DR 次模最大化问题. 借鉴文献 [53] 中对全信息模型下算法设计的思想, 他们设计了 Bandit-Measured Frank-Wolfe 算法, 达到了 $1/\mathrm{e}$ 的近似比和 $O(T^{8/9})$ 的遗憾界.

5 结论

连续 DR 次模最大化问题在优化领域具有广泛的应用和理论价值, 尽管已有许多研究成果, 但仍有不少值得深入探索的研究方向. 以下是一些值得进一步研究的问题.

(1) 设计加速梯度型算法. 在连续 DR 次模最大化问题的研究中, 如何借助如 Nesterov 加速方法这样经典的加速技巧, 来有效减少迭代次数和加快收敛, 是值得思考的问题. 除了 Nesterov 加速, 还可以考虑通过二阶信息 (如 Newton 法或拟 Newton 法) 加速梯度型算法和 Anderson 加速等, 提高在复杂约束下的收敛速率.

(2) 设计非凸约束下的次模最大化算法. 目前, 连续 DR 次模最大化问题的研究主要集中在凸约束条件下. 然而, 在实际应用中, 许多问题的约束条件可能是非凸的. 如何设计适用于非凸约束的近似算法, 同时确保算法的收敛性, 是一个富有挑战性的研究方向.

(3) 设计连续 DR 次模结构的双层规划算法. 双层规划问题本质上是一个嵌套的优化问题, 上层决策的输出往往依赖于下层决策, 而下层问题的约束或目标则受到上层问题的影响. 在双层结构中如何将连续 DR 次模函数嵌入上下层次的目标函数中, 使问题既符合双层次逻辑又能保持次模函数的性质, 是值得探讨的基础问题. 同时, 设计能够处理凸或非凸约束的双层次模最大化算法, 并确保算法在这些约束下的收敛速度, 是另一个重要的研究挑战.

连续 DR 次模最大化问题作为一类重要的优化问题, 其挑战主要源于潜在的非凸结构和目标函数的边际收益递减性. 近年来, 研究者们开始关注如何利用连续 DR 次模函数的这一结构特点来设计和分析有效的优化算法. 目前, 解决连续 DR 次模最大化问题的主流算法主要包括 Frank-Wolfe 变形算法、双贪婪算法、随机算法和在线算法等. 本文对这些算法的研究进展进行梳理和总结, 简要介绍相关算法的设计思想和基本理论性质, 如近似比和复杂度. 通过本文的综述, 读者将对连续 DR 次模函数最大化的理论基础、算法设计及其在实际中的应用有较为全面的了解, 从而为未来的研究和应用提供有价值的参考.

参考文献

1 Badanidiyuru A, Mirzasoleiman B, Karbasi A, et al. Streaming submodular maximization: Massive data summarization on the fly. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery, 2014, 671–680

2 Bian A, Levy K Y, Krause A, et al. Continuous DR-submodular maximization: Structure and algorithms. In: Proceedings of the 31st Neural Information Processing Systems. Red Hook: Curran Associates, 2017, 486–496

3 Bian A A, Mirzasoleiman B, Buhmann J, et al. Guaranteed non-convex optimization: submodular maximization over continuous domains. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Fort Lauderdale: JMLR, 2017, 111–120

4 Bilmes J, Bai W. Deep submodular functions. arXiv:1701.08939, 2017

5 Buchbinder N, Feldman M, Naor J S, et al. Submodular maximization with cardinality constraints. In: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia: SIAM, 2014, 1433–1452

6 Buchbinder N, Feldman M, Seffi J, et al. A tight linear time (1/2)-approximation for unconstrained submodular maximization. SIAM J Comput, 2015, 44: 1384–1402

7 Chen L, Harshaw C, Hassani H, et al. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In: Proceedings of the 35th International Conference on Machine Learning. Stockholm: PMLR, 2018, 814–823

8 Chen L, Hassani H, Karbasi A. Online continuous submodular maximization. In: Proceedings of the 21st International Conference on Artificial Intelligence and Statistics. Lanzarote: PMLR, 2018, 1896–1905

9 Chen L, Zhang M, Hassani H, et al. Black box submodular maximization: Discrete and continuous settings. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics. Boston: Addison-Wesley Publ, 2020, 1058–1070

10 Das A, Kempe D. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. arXiv:1102.3975, 2011

11 Djolonga J, Krause A. From map to marginals: Variational inference in Bayesian submodular models. In: Proceedings

of the 27th Neural Information Processing Systems. Nevada: Curran Associates, 2014, 244–252
12 Du D. Lyapunov function approach for approximation algorithm design and analysis: With applications in submodular maximization. arXiv:2205.12442, 2022
13 Du D, Liu Z, Wu C, et al. An improved approximation algorithm for maximizing a DR-submodular function over a convex set. arXiv:2203.14740, 2022
14 Dürr C, Thắng N K, Srivastav A, et al. Non-monotone DR-submodular maximization: Approximation and regret guarantees. arXiv:1905.09595, 2019
15 Dürr C, Thắng N K, Srivastav A, et al. Non-monotone DR-submodular maximization over general convex sets. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence. Yokohama: IJCAI, 2020, 2148–2154
16 Elenberg E R, Khanna R, Dimakis A G, et al. Restricted strong convexity implies weak submodularity. Ann Statist, 2018, 46: 3539–3568
17 Feng J, Yang R, Zhang H, et al. Online non-monotone DR-submodular maximization: 1/4 approximation ratio and sublinear regret. In: Proceedings of the 28th International Computing and Combinatorics Conference. Shenzhen: Springer, 2022, 118–125
18 Filmus Y, Ward J. The power of local search: Maximum coverage over a matroid. In: Proceedings of the 29th Symposium on Theoretical Aspects of Computer Science. Paris: Leibniz-Zentrum für Informatik, 2012, 601–612
19 Filmus Y, Ward J. Monotone submodular maximization over a matroid via non-oblivious local search. SIAM J Comput, 2014, 43: 514–542
20 Flaxman A D, Kalai A T, McMahan H B. Online convex optimization in the bandit setting: Gradient descent without a gradient. In: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms. Vancouver: SIAM, 2005, 385–394
21 Frank M, Wolfe P. An algorithm for quadratic programming. Naval Res Logist, 1956, 3: 95–110
22 Fujishige S. Submodular Functions and Optimization. London: Elsevier, 2005
23 Gottschalk C, Peis B. Submodular function maximization on the bounded integer lattice. In: Proceedings of the 13th International Workshop on Approximation and Online Algorithms. Patras: Springer, 2015, 133–144
24 Hartline J, Mirrokni V, Sundararajan M. Optimal marketing strategies over social networks. In: Proceedings of the 17th International Conference on World Wide Web. Beijing: ACM, 2008, 189–198
25 Hassani H, Karbasi A, Mokhtari A, et al. Stochastic continuous greedy++: When upper and lower bounds match. In: Proceedings of the 33rd Neural Information Processing Systems. Vancouver: Curran Associates, 2019, 13087–13097
26 Hassani H, Karbasi A, Mokhtari A, et al. Stochastic conditional gradient++: (Non-)convex minimization and continuous submodular maximization. SIAM J Optim, 2020, 30: 3315–3344
27 Hassani H, Soltanolkotabi M, Karbasi A. Gradient methods for submodular maximization. In: Proceedings of the 31st Neural Information Processing Systems. Long Beach: Curran Associates, 2017, 5843–5853
28 Hassani H, Soltanolkotabi M, Karbasi A. Gradient methods for submodular maximization. arXiv:2205.12442, 2017
29 Hatano D, Fukunaga T, Maehara T, et al. Lagrangian decomposition algorithm for allocating marketing channels. In: Proceedings of the 29th of the AAAI Conference on Artificial Intelligence. Austin: AAAI, 2015, 1144–1150
30 Hazan E. Introduction to online convex optimization. Found Trends Optim, 2016, 2: 157–325
31 Jaggi M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In: Proceedings of the 30th International Conference on Machine Learning. Atlanta: PMLR, 2013, 427–435
32 Ju J, Wang X, Xu D. Stochastic approximation algorithms for DR-submodular maximization with convex functional constraints. Https://optimization-online.org/wp-content/uploads/2024/07/SA_CDR-1.pdf, Optimization Online, 2024
33 Krause A. Capacitated facility location: Separation algorithms and computational experience. Math Program, 1998, 81: 149–175
34 Krause A, Cevher V. Submodular dictionary selection for sparse representation. In: Proceedings of the 27th International Conference on Machine Learning. Haifa: Omnipress, 2010, 567–574
35 Leskovec J, Krause A, Guestrin C, et al. Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Jose: ACM, 2007, 420–429
36 Lin H, Bilmes J. A class of submodular functions for document summarization. In: Proceedings 49th Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Portland: ACL, 2011, 510–520
37 Marinacci M, Montrucchio L. Ultramodular functions. Math Oper Res, 2005, 30: 311–332
38 Mokhtari A, Hassani H, Karbasi A. Conditional gradient method for stochastic submodular maximization: Closing the gap. In: Proceedings of the 21st International Conference on Artificial Intelligence and Statistics. Lanzarote: PMLR,

2018, 1886–1895
39 Mokhtari A, Hassani H, Karbasi A. Stochastic conditional gradient methods: From convex minimization to submodular maximization. J Mach Learn Res, 2020, 21: 4232–4280
40 Nemhauser G L, Wolsey L A, Fisher M L. An analysis of approximations for maximizing submodular set functions — I. Math Program, 1978, 14: 265–294
41 Niazadeh R, Roughgarden T, Wang J R. Optimal algorithms for continuous non-monotone submodular and DR-submodular maximization. J Mach Learn Res, 2020, 21: 1–31
42 Nong Q, Fang J, Gong S, et al. A 1/2-approximation algorithm for maximizing a non-monotone weak-submodular function on a bounded integer lattice. J Comb Optim, 2020, 39: 1208–1220
43 Raut P, Sadeghi O, Fazel M. Online DR-submodular maximization: Minimizing regret and constraint violation. In: Proceedings of the 35th of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI, 2021, 9395–9402
44 Roughgarden T, Wang J R. An optimal learning algorithm for online unconstrained submodular maximization. In: Proceedings of the 31st Annual Conference on Learning Theory. Stockholm: PMLR, 2018, 1307–1325
45 Sadeghi O, Fazel M. Online continuous DR-submodular maximization with long-term budget constraints. In: Proceedings of the 23th International Conference on Artificial Intelligence and Statistics. Sicily: Curran Associates, 2020, 4410–4419
46 Sadeghi O, Raut P, Fazel M. A single recipe for online submodular maximization with adversarial or stochastic constraints. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. Red Hook: Curran Associates, 2020, 14712–14723
47 Snoek J, Larochelle H, Adams R P. Practical bayesian optimization of machine learning algorithms. In: Proceedings of the 25th Neural Information Processing Systems. Nevada: Curran Associates, 2012, 2951–2959
48 Sokolov A, Kreutzer J, Lo C, et al. Stochastic structured prediction under bandit feedback. In: Proceedings of the 30th Neural Information Processing Systems. Barcelona: Curran Associates, 2016, 1497–1505
49 Soma T, Yoshida Y. Maximizing monotone submodular functions over the integer lattice. Math Program, 2018, 172: 539–563
50 Streeter M, Golovin D. An online algorithm for maximizing submodular functions. In: Proceedings of the 21st Neural Information Processing Systems. Whistler: Curran Associates, 2008, 1577–1584
51 Sviridenko M. A note on maximizing a submodular set function subject to a knapsack constraint. Oper Res Lett, 2004, 32: 41–43
52 Taskar B, Chatalbashev V, Koller D, et al. Learning structured prediction models: A large margin approach. In: Proceedings of the 22nd International Conference on Machine Learning. Bonn: Omnipress, 2005, 896–903
53 Thắng N K, Srivastav A. Online non-monotone DR-submodular maximization. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI, 2021, 9868–9876
54 Thornton C, Hutter F, Hoos H H, et al. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Chicago: ACM, 2013, 847–855
55 Vondrák J. Optimal approximation for the submodular welfare problem in the value oracle model. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing. Victoria: ACM, 2008, 67–74
56 Wainwright M J, Jordan M I. Graphical Models, Exponential Families, and Variational Inference. Norwell: Now Foundations and Trends, 2008
57 Zhang M, Chen L, Hassani H, et al. Online continuous submodular maximization: From full-information to bandit feedback. In: Proceedings of the 32nd Neural Information Processing Systems. Vancouver: Curran Associates, 2019, 9210–9221
58 Zhang M, Shen Z, Mokhtari A, et al. One sample stochastic Frank-Wolfe. In: Proceedings of the 23th International Conference on Artificial Intelligence and Statistics. Boston: Addison-Wesley Publ, 2020, 4012–4023
59 Zhang Q, Deng Z, Chen Z, et al. Stochastic continuous submodular maximization: Boosting via non-oblivious function. In: Proceeding of the 39th International Conference on Machine Learning. Baltimore: PMLR, 2022, 26116–26134
60 Zhang Q, Deng Z, Chen Z, et al. Online learning for non-monotone submodular maximization: From full information to bandit feedback. In: Proceedings of the 26th International Conference on Artificial Intelligence and Statistics. Valencia: PMLR, 2023, 3515–3537

A survey on theory and algorithms for continuous DR-submodular maximization

Jiachen Ju, Ruiqi Yang, Zhenning Zhang & Ding-Zhu Du

Abstract The diminishing returns property defines an optimization challenge known as the DR-submodular maximization problem. In the continuous domain, the DR-submodular function holds significant theoretical research value and demonstrates extensive practical relevance across various applications. This type of problem is commonly abstracted from practical scenarios including resource allocation, advertising budget optimization, sensor network, and energy management. In this paper, we discuss classic algorithms for continuous DR-submodular maximization and their variants, covering Frank-Wolfe algorithms, double greedy algorithms, stochastic algorithms, and online algorithms. We also review recent research progress on continuous DR-submodular maximization, providing useful references for further exploration of frontier issues in this field.

Keywords **DR-submodular maximization, non-convex optimization, approximation algorithms**

MSC(2020) **90C26, 68W25**

doi: 10.1360/SSM-2024-0195