arXiv:2303.01673v2 [cs.DS] 9 Mar 2023

Near 在线学习的最优内存—遗憾权衡

Binghui Peng* Aviad Rubinstein†哥伦比亚大学 斯坦
福大学 bp2601@columbia.edu aviad@cs.stanford.edu

摘要

在专家问题中，在每个 T 天里，代理都需要听从 n 位“专家”之一的建议。每天结束后，都会揭示与每位专家的建议相关的损失。学习理论中的一项基本结果表明，代理能够实现趋零遗憾，即其累积损失与事后最优专家的累积损失之差在 $o(T)$以内。

如果代理没有足够的空间来记住所有专家，还能表现良好吗？我们从两个方向拓展了这一新兴研究领域：

1. 针对非自适应对手，我们提出了一种新算法，在 [PZ23], 所获得的内存-遗憾权衡基础上有所改进，并且几乎达到了 [SWXZ22]的下界。2. 我们还考虑了一种能够观察代理过去选择的专家的自适应对手。在这种情况下，我们同时给出了一种新算法和一个新颖的下界，证明了大致为$\sqrt{n}$ 的内存对于获得 $o(T)$遗憾既是必要的，也是充分的。

*本研究获得美国国家科学基金会 CCF-1703925、IIS-1838154、CCF-2106429、CCF-2107187、CCF-1763970、AF2212233、COLL2134095、COLL2212745 项目资助
†获美国国家科学基金会 CCF-1954927 和 David and Lucile Packard Fellowship 资助

1 引言

考虑这样一个在线学习问题：代理必须在一个未知且动态变化的环境中做出决策。该问题的标准离散抽象考虑一个由 T 天组成的序列；每天早上， n 专家向代理提供“建议”，代理选择其中一位并遵循其建议；每天晚上，代理会观察其选择所造成的损失，以及她本可以做出的所有其他选择所造成的损失[1]。这一问题的基准是最小化遗憾，即

$$\text{(Loss of agent's actions)} - \text{(Loss of single best-in-hindsight expert)}.$$

令人惊讶的是，像乘法权重更新（MWU）这样的简单算法，即使面对对抗性选择的损失序列，其总遗憾相对于总可能损失也渐近趋于零。这一著名结果在理论和实践中有诸多应用，包括提升 [FS97], 博弈中的学习 [FS99、用于最大流的近似算法 [CKM+11] 等。

最近，Srinivas、Woodruff、Xu 和 Zhou [SWXZ22]开启了对在线学习内存界的研究。他们在每一天的损失都独立地从同一分布中抽取的情形下，对内存—遗憾权衡给出了近乎紧致的刻画（另一种说法是，存在一个固定的损失向量集合，它们以随机顺序到达）。特别是，他们给出了一种算法，只使用多重对数(nT)内存便能实现趋零遗憾。 [SWXZ22] 提出了一个开放问题：在不作独立同分布或随机顺序假设的情况下，可能实现怎样的内存—遗憾权衡？Peng 和 Zhang [PZ23] 首次在这个问题上取得进展，他们给出了一种具有趋零遗憾的低内存算法。我们的首项贡献是在这一设定下提出一种新的在线学习算法，改进了 [PZ23]的遗憾—内存权衡，并基本匹配 [SWXZ22]的下界。[2]

定理 1.1 (算法，非自适应对手)。 设 n、T 为足够大的整数，且多重对数$(nT) \le S \le n$。存在一种在线学习算法，使用至多 S 空间，并且针对非自适应对手，以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率达到$\widetilde{O}\left(\sqrt{\frac{nT}{S}}\right)$遗憾。

注意到，忽略多重对数因子后，这一内存—遗憾权衡是紧的，从而解决了 [SWXZ22] 提出的开放问题。另一个有趣之处在于， [SWXZ22] 的下界对损失独立同分布生成这一特殊情形也成立。因此，我们的算法表明，更一般的非自适应对手实际上并不会带来严格更大的困难。

到目前为止讨论的所有低内存算法，都适用于非自适应 对手，即对手会预先固定损失序列。但经典乘法权重更新算法还有一个更加令人惊叹的性质：即使面对自适应对手 ，它仍能实现趋零遗憾；这种对手可以根据代理过去的选择来更新当前损失向量。这一针对自适应对手的扩展，对于分析代理与环境交互 的情形至关重要，而这正是博弈论中最重要的应用所涉及的情形（例如参见 [Rou13，备注 3.2]）。

我们给出一种针对自适应对手、使用次线性内存的在线学习算法。~~具体而言，我们的算法仅使用 $O(\sqrt{n})$ 内存，即可实现 $o(T)$ 遗憾。

[1]代理能够获知自己未采取的动作事后损失的模型称为全信息反馈 。本文不研究另一种多臂老虎机 模型；在这种模型中，代理必须探索不同的动作，才能了解它们的损失分布。[2]我们的空间高效算法无需一次性访问整个损失向量——（单遍）流式访问即可。与此同时，即使面对更强的算法，我们的下界仍然成立：这类算法每天都可以使用任意大小的内部内存来处理损失向量，但在天与天之间存储的信息量受到限制。

定理 1.2 (算法，自适应对手)。设 n, T 为充分大的整数。存在一种在线学习算法，最多使用 S 空间，并且针对自适应对手实现 $\widetilde{O}\left(\text{最大}\left\{\sqrt{\frac{nT}{S}}, \frac{\sqrt{nT}}{S}\right\}\right)$ 遗憾，其成立概率至少为 $1-1/(nT)^{\varpi(1)}$。

最后，我们针对自适应对手情形给出了一个新颖的下界，表明要保证遗憾趋零，也需要 $\Omega(\sqrt{n})$ 的内存。

定理 1.3 （针对自适应对手的下界）。设 $\epsilon \in (\frac{1}{\sqrt{n}}, 1)$，任何能保证对抗自适应对手时遗憾为 $o(\epsilon^2 T)$ 的在线学习算法，都需要 $\widetilde{\Omega}(\frac{\sqrt{n}}{\epsilon})$ 空间。

备注 （运行时间）。本文关注的是内存。值得注意的是，这两种算法（定理 *1.1* 和定理 *1.2*）在计算上都是高效的——每轮的运行时间随内存线性增长。

1.1 相关工作

与先前工作的比较 [SWXZ22, PZ23] SWXZ22, PZ [的工作与我们密切相关，下面将进行详细讨论。Srinivas、Woodruff、Xu 和 Zhou [SWXZ22] 首先研究了在线学习的内存界，并给出了一个 $\Omega(\sqrt{\frac{nT}{S}})$ 的遗憾下界，即使对于独立同分布损失序列也成立。他们的下界通过抛硬币问题归约得到，并借助直和论证和强数据处理不等式加以证明。与我们针对自适应对手的下界相比，其构造和证明技巧都大不相同。 [SWXZ22] 还给出了一种算法，当损失序列以随机顺序到达时，其遗憾为 $O(\sqrt{\frac{nT}{S}})$。他们的分析严重依赖于损失序列的随机性，这并不令人意外。

Peng 和 Zhang [PZ23]为非自适应对手提供了首个次线性空间在线学习算法。该算法使用 n^δ 空间，获得的总遗憾为 $\widetilde{O}(n^2 T^{\frac{2}{2+\delta}})$（在同期工作中， [ACNS23] 将此遗憾界改进为 $\widetilde{O}(n^2 T^{\frac{1}{1+\delta}})$）。我们的基线算法与 [PZ23, ACNS23], 有些相似，但正如我们很快将要讨论的，其关键组件，即驱逐规则，大不相同。我们的鲁棒化过程、摊还技术以及低内存"单生"专家过程都是全新的组件，使我们能够针对非自适应对手实现近乎最优的权衡关系。

Peng 和 Zhang [PZ23] 还给出了一个空间下界：对于任意遗憾为 $O(\epsilon T)$ 的算法，面对强自适应对手时，其空间至少为 $\Omega(\min\{\epsilon^{-1}\log\ n, n\})$。正如 [PZ23], 中所指出的，强对手可以看到算法在当前轮的分布/混合策略，这比文献中通常考虑的自适应对手更强。我们的下界在定量上更强，而且即使面对较弱且更标准的自适应对手概念也成立。就技术而言， [PZ23] 通过计数论证以及与博弈中的学习之间的联系（这需要强对手假设）建立了他们的下界。相比之下，我们的下界实例基于标准的自适应对手概念构造，该对手可以攻击算法的近期行动；证明依赖于通信复杂性中的直积定理。事实上，利用我们的方法，可以很容易地推出：对于任何次线性遗憾算法 $o(T)$，面对强自适应对手时，其下界为 $\Omega(n)$（见第 6.1 节）。

与 [WZZ23] 针对自适应对手的工作[WZZ23]也针对自适应对手给出了下界和算法，但其问题设定和技术方法都与我们的工作大不相同。具体而言：（1）他们关注二元预测任务，而我们研究的是

一般模型，在该模型中，对手会观察算法在先前轮次中选出的具体专家。（2）他们还考虑了一个额外参数 M，即最佳专家做出错误二元预测的次数。在这一设定下，他们给出了确定性算法的空间-遗憾权衡紧确界，并给出了一种随机化算法，当$M = o(T/\log^2 n)$时，可在这一权衡基础上进一步改进。尽管设定不同，值得注意的是，我们针对自适应对手的算法无需对最佳专家的损失作任何假设，就能以次线性内存实现次线性遗憾；而且，我们的下界对随机化算法也成立。

在线学习 经典的乘法权重更新算法已被证明能够为专家建议在线学习实现最优遗憾 [LW89, OC98]。作为一种元方法，它在算法设计与分析中得到了广泛应用，包括提升 [FS97], 博弈中的学习 [FS99, CBL06],最大流近似算法 [CKM+11] 等 [GH16, KM17, HLZ20]。完整介绍请参阅综述[AHK12] 。

乘法权重更新算法属于更广泛的跟随随机化／扰动领导者框架 [KV05, Haz16], ，该框架涵盖了几乎所有现有的在线学习算法（ [PZ23]除外）。正如 [SWXZ22, PZ23], 所指出的，即使要找出一个（常数因子）近似最佳专家，也需要 $\Omega(n)$ 内存，因此需要新的思路来设计低内存在线学习算法。在线学习问题也在其他背景下得到研究，例如预言机高效算法 [HK16, DHL+20]、平滑对手 [HRS22, HHSY22, BDGR22]以及老虎机反馈 [ACBFS02, BCB+12]。研究者还考虑了其他扩展的遗憾概念，包括二阶遗憾 [KVE15], 、区间遗憾 [HS07], 、休眠专家 [FSSW97, BM07]等 [BGZ15, CP20, ADF+22, WL18, BLLW19]。

学习的内存界随机多臂老虎机和纯探索中的内存界已得到证明 [LSPY18, AW20, MPK21, AKP22]。但它们并非针对最坏情况或对抗性自然方的内存界。

过去十年间，学习中的内存作用得到了广泛研究，包括奇偶学习中的时间—空间权衡 [Raz17, Raz18, GRT18, GRT19, GKLR21], 、凸优化的内存界 [MSSV22, BZJ23], 、持续学习 [CPP22], 、泛化 [Fel20, BBF+21]以及许多其他方向 [SVW16, SSV19, GLM20, BBS22, DS18]。

通信复杂度一直是证明内存下界的有效方法。特别是，我们的下界证明关键地运用了直接乘积定理的工具 [Raz95, Hol07, BW15, BRWY13, Kla10]以及集合不相交的信息代价 [BYJKS04, She14]。

1.2 讨论

在这项工作中，我们研究专家问题中的遗憾—内存权衡关系；专家问题本身就是一个基础性问题。专家问题还被用作博弈论 [FS99], 机器学习 [KM17] 和优化 [PST95], 中众多应用的子程序；了解我们的结果是否能为其中任何应用带来子空间算法，是很有意义的。

在许多场景中，专家集合还具有其他结构（例如 Littlestone 维度有界），或者损失向量具有简洁的表示形式。这些结构性假设为实现更优的内存—遗憾权衡提供了机会，甚至可能优于我们的最坏情况界。理解专家问题中这些重要特殊情形下的内存—遗憾权衡，是一个令人振奋的未来研究方向。[3]

[3]请注意，即使存在允许以简洁方式存储任意一天收益的结构性假设，天真地记住这些收益达 T 天，对于较大的 T而言也无法实现高效的空间利用。

2 技术概述

我们将从宏观层面概述所采用的技术。

2.1 针对非自适应对手的算法

我们重点研究 $S = \text{polylog}(nT)$ 的情况，并以实现 $\widetilde{O}\left(\sqrt{nT}\right)$ 的遗憾界为目标。通过分组技巧，可以将结果推广到对 S 的一般依赖情形，这部分内容将在最后介绍。

2.1.1 基线构建模块

我们首先介绍一个基线子程序，其总遗憾为 $\widetilde{O}(nB + \frac{T}{\sqrt{B}})$。基线算法逐轮次运行，共有 $\frac{T}{B}$ 个轮次，每个轮次包含 B 天。算法维护一个专家池 $\mathcal{P}$，并在每个轮次中对池中的专家运行乘法权重更新；在该轮次内，专家池保持不变。专家池每轮次更新一次：在轮次开始时随机采样并添加 $O(1)$ 个专家，算法则在轮次结束时对专家池进行剪枝。到目前为止，这与 [PZ23] 的基本设置相同：算法现在相对于专家池中的最佳专家具有竞争性，任务是维护一个优质的专家集合。

驱逐规则——首次尝试 第一个关键步骤，也是与[PZ23] 以往工作的不同之处，是采用了一种新的驱逐规则。回顾一下，我们只有多对数空间，并且需要保留一小组核心专家。对于每个专家 $i \in \mathcal{P}$，我们将其在池中的生命周期内的累计损失与以下非常强大的基准进行比较：

> **覆盖基准：** 考虑一种假想算法：在每个轮次中，切换到池中（事后看来）在该轮次表现最佳的单个专家。我们的基准是这一假想算法的总累计损失[4]。

如果该专家的损失并不严格优于这一基准，我们就称它被池中的其他专家覆盖，因而会受到驱逐。直观地说，尽管这一基准很强，但只要某个专家被池覆盖，就足以保证在该池上运行乘法权重更新时，相对于该专家具有较低的遗憾。

- **内存。** 这一强大基准带来的直接优势是节省内存。对于一组在驱逐后存活的专家，很容易看出，由最后r 个专家覆盖的后缀至少具有 2^{r-1}的总体优势。这表明 $|\mathcal{P}| \le O(\log\ T)$，否则最大损失将超过 T。换句话说，剪枝步骤之后只有 $O(\log\ T)$个专家存活！最后需要指出的是，为避免记录每个轮次的损失，实际上只需记录成对损失，而内存需求会随池的大小呈二次增长。

- **遗憾分析（理想化设想）** 令 i^* 为最佳专家。为了进行遗憾分析，我们将整个时期划分为两种类型。如果在某个轮次中对 i^* 进行采样，而它最终会被覆盖并从池中驱逐，那么这个轮次就是“良好”的。考虑从当前良好轮次到 i^* 被踢出池为止的区间。我们希望论证，即使从未对 i^* 进行采样，该算法在整个区间上的表现也能与 i^* 相当。如果在某个轮次中对 i^* 进行采样，而它永远不会被踢出池，那么这个轮次就是“不良”的。我们无法控制不良时期中的遗憾，但直觉上，在对 i^* 进行采样并将其永久保留之前，我们只会经历少数几个不良时期。

[4]请将此基准与标准遗憾概念进行对比：后者的基准是在整个数据流中始终不变的单个专家。

将上述思路形式化时存在一个致命问题。我们希望通过考虑这样一种假设情形来分析遗憾：在给定轮次将 i^* 采样到池中，但驱逐时间可能取决于后来进入池中的专家。也就是说，在定义好/坏时期时，我们需要固定整个所选专家序列。然而，一旦真正将 i^* 采样到池中，它就会干扰池中的其余部分，因为将 i^* 加入池中可能会踢出其他专家，而被踢出的专家又可能使池保留其他专家（这些专家原本应该被移除），等等。这个问题看似微妙，但要解决它，技术上却极具挑战性！

一种鲁棒算法 我们展示了修复上述问题的高层次直觉。总体思路是让整个算法对于任一轮次中新加入的单个专家都不敏感，或称"鲁棒"。我们不再每轮都执行剪枝步骤，而是维护 $L+1=O(\log T)$ 个互不相交的子池 $\mathcal{P}=\mathcal{P}_0\cup\cdots\cup\mathcal{P}_L$，其中第 ℓ 个子池 $\mathcal{P}_\ell(\ell\in[0:L])$ 包含最近 2^ℓ 轮次中加入的专家，并且每隔 2^ℓ 轮次就将其合并到更高层的子池 $\mathcal{P}_{\ell+1}$。这项改动对于鲁棒算法至关重要，因为它确保一个专家最多可能参与驱逐过程 $O(\log T)$ 次（而不是每轮都参与）。

给定两个相邻的子池 $\mathcal{P}_\ell$，$\mathcal{P}_{\ell+1}$，合并步骤需要确保：(1) 输出子池的规模较小；(2) 结果对单个变化[5]不敏感。最终的合并和剪枝步骤是一个迭代过程。在每次迭代中，我们以一种不敏感的方式估计 $\mathcal{P}_\ell\cup\mathcal{P}_{\ell+1}$ 的大小；如果它过大，就采样 $1_{\text{多重对数}(nT)}$ 比例的专家，并用它们筛选该子池。重复此过程 $O(\log(nT))$ 次，我们证明每次迭代都能以高概率移除 $\Omega(1_{\log(nT)})$ 比例的专家。

更详细的基线算法描述 总而言之，基线算法维护 $L+1$ 子池 $\mathcal{P}$，并且对于 $\ell\in[0:L]$ 第 ℓ 个子池 $\mathcal{P}_\ell$ 每经过 $\mathcal{P}_{\ell+1}$ 2^ℓ 个轮次就会合并一次。合并和剪枝步骤最多进行 $K=O(\log(nT))$ 次迭代。在每次迭代中，算法首先通过子采样估计 $\mathcal{P}_\ell\cup\mathcal{P}_{\ell+1}$ 的大小，只有当其规模较大时才继续；随后从专家中采样 $1_{\text{poly}\log(nT)}$ 比例的专家作为 筛选 集合 $\mathcal{F}$。如果某个专家被 $\mathcal{F}$ 覆盖，则将其移除。

- **内存分析**。 现在，内存分析变得复杂起来，我们的目标是证明每次迭代都会移除 $\Omega(1_{\log(nT)})$-比例的专家。为此，我们递归地 识别（至多） $O(\log T)$ 个关键专家，使得在 $\mathcal{F}$ 中，最后 r 个关键专家的后缀至少比池中剩余专家中的很大一部分2^{r-1}更优；除非在递归过程中已经移除了 $\Omega(1_{\log(nT)})$-比例的专家。

- **遗憾分析**。 如果某个专家从未参与驱逐过程，即从未在筛选集 $\mathcal{F}$ 和大小估计中被采样，则称其为被动的。否则，它就是主动的。成为被动专家的概率很大，即至少为 $(1-1_{\text{多重对数}(nT)})^{2L\times 2K}\geq 1/2$。在分析中，我们固定每一轮次中被采样且主动的专家集合，并将所有轮次分成两类，定义方式与之前类似。被采样且被动的专家集合并未固定，但它们不会影响基线算法的执行（即估计大小、筛选集以及主动专家的存活集合）。良好轮次的遗憾得到了良好控制。而对于不良时期，如果专家 i^* 被采样，同时又是被动专家，那么它就永远不会被淘汰。由于专家 i^* 成为被采样且被动的专家的概率为 $\Omega(\frac{1}{n})$，

[5]在这个阶段，我们似乎需要一种"差分隐私合并/剪枝"过程。差分隐私技术确实给了我们启发，但不能直接应用。内存分析非常精细，即使是标准差分隐私机制造成的微小扰动，也会对其产生显著影响。

并且我们已经证明，池大小超过多重对数(nT)的情况很少发生，因此不良轮次的数量至多为 $O(n)$。

总之，基线的遗憾来自三个方面：(1) 每轮因乘法权重更新产生 $\widetilde{O}(\sqrt{B})$；(2) 对每个包含良好轮次的区间产生 1；(3) 对每个不良时期产生 B，因此遗憾的上界为 $\widetilde{O}(\sqrt{B}\cdot(T/B)+(T/B)+B\cdot n)=\widetilde{O}(T/\sqrt{B}+nB)$。

一种简单但非最优的自举方法 基线方法在优化时期大小后，遗憾为 $\widetilde{O}(n^{1/3}T^{2/3})$。一种简单的自举方法（与 [PZ23] 中对应的方法类似，甚至更简单）可用于降低遗憾，并实现对 T 的最优依赖关系。具体而言，我们对不良时期采用了较为粗略的遗憾界 B；通过针对每个时期分别运行基线（序列长度为 B 而非 T），并通过对逐时期基线（序列长度为 B）和完整基线（序列长度为 T）运行乘法权重更新来确定最终决策，可以降低该界。这样一来，不良时期的遗憾会降至 $\widetilde{O}(n^{1/3}B^{2/3})$，而非 B，良好时期的遗憾则保持在相同的阶。总遗憾变为 $\widetilde{O}(T/\sqrt{B}+n^{4/3}B^{2/3})$，平衡时期大小后，可得到改进的遗憾 $\widetilde{O}(n^{4/7}T^{4/7})$。自举方法可以重复多次，从而得到遗憾为 $\widetilde{O}(n\sqrt{T})$ 的算法。这已经实现了对 T 的最优依赖关系，但仍相差一个因子 $\sqrt{n}$。

2.1.2 针对非自适应对手的最优提升

最终的非自适应对手算法维护多个基线线程，每个线程使用不同的参数，并谨慎地摊销遗憾。新思路（与 [PZ23] 相比）包括：(1) 将池从高频基线继承到低频基线（而不仅仅是继承遗憾界）；(2) 将池维护与遗憾控制分开。

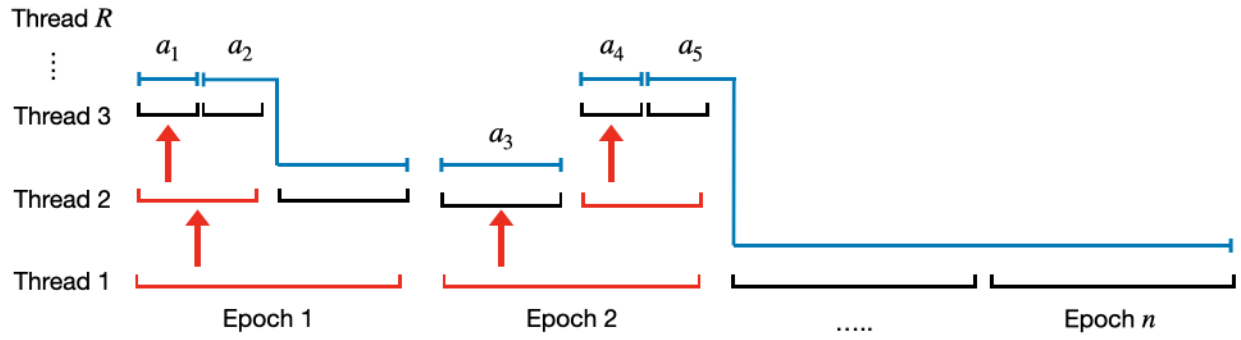

图 1：轮次分配。红色线段表示各线程的不良时期，蓝色线段表示由游走生成的区间。

图中游走的详细描述： 由于线程 1 和线程 2 中的第一个轮次都是坏轮次，游走转向线程 3 的第一个轮次，此处 i^* 可以被覆盖并驱逐（区间a_1）。在下一个轮次中，i^* 未被立即覆盖，因此游走向下转到线程 2，并在此处被覆盖（区间 a_2）。游走转向下一个轮次（线程 1 中的轮次 2），该轮次是坏轮次。游走向上转到线程 2，在那里找到一个能立即覆盖 i^* 的良好轮次（区间 a_3）。线程 2 的下一个轮次是坏轮次，因此游走向上转到线程 3；此处 i^* 被立即覆盖（区间 a_4）。之后，i^* 一直存活，直到最后一个轮次才被覆盖（区间 a_5）。

完整算法维护 $R = O(\log T)$ 个基线算法线程。最低的线程有 n 个轮次，每个轮次包含 $B_1 = \frac{T}{n}$ 天；第 r 个（$r \in [2:R]$）线程仅有 2 个轮次，每个轮次包含 $B_r = \frac{T}{2^{r-1}n}$ 天——它每隔 $T_r = B_{r-1}$ 天重启一次。

每次重启第 r个线程时，我们还会为较低层的线程启动一个新的轮次（轮次更长）；我们不再丢弃第 r个线程的池中的专家，而是将其移至开启新轮次的最低层线程。这是它与次优自举方法相比的一个关键区别。其直观上的合理性在于，人们可能希望最优专家在较高层线程中经历几轮采样/剪枝后，能有更大的机会被保留（尽管最终的遗憾分析结果大不相同）。内存界基本上沿用基线中的结论，下面我们将概述遗憾分析。

摊销遗憾 在遗憾分析中，与之前一样，我们固定每个线程中的活跃且已采样的 专家。为计算算法产生的全部遗憾，我们将整个序列划分为一组区间，并为这些区间提供不同的遗憾保证。该划分由对不同线程上的轮次进行的游走所定义。我们再次指出，这种游走仅用于分析。

游走从底层线程（轮次最长）中的第一个轮次开始。我们称游走中考察的某个轮次为不良轮次，如果假设专家 i^* 在该轮次开始时进入池中，那么直到算法结束，存活的活跃专家 集合（对所有线程取并集）都无法覆盖 i^*。

1. 即使当前轮次是不良轮次，我们也不希望在整个轮次内都承担最大损失。相反，游走会向上移动到轮次更短的较高线程，希望至少能在该轮次的部分时间内覆盖 i^*。（只要该轮次是不良轮次，且仍有轮次更短的较高线程，游走就会继续向上移动；如果游走到达最高线程，我们可以承受一个较短的不良轮次。）

2. 对于良好轮次，我们可以考察从该轮次到 i^* 将被踢出的轮次之间的区间；可以保证算法在此区间内（从当前轮次到被踢出的轮次）遭受的遗憾较小。根据我们紧致提升算法的定义，如果 i^* 停留足够长时间，它就会移动到较低的线程中（每次我们重启 i^* 当前所在的线程时，i^* 都会带着一个新的移动到最低线程）。我们的游走沿着 i^* 在不同线程的各个轮次中的轨迹前进，直到它被驱逐。

3. 一旦 i^* 被淘汰，我们就继续游走，移动到如果 i^* 没有被淘汰本应进入的下一个轮次。我们判断该轮次是好是坏，然后继续执行上面的步骤 1 或步骤 2。

在分析中，我们考虑由游走的步骤 2 定义的区间。在每个区间 $\mathcal{I}$上，遗憾的规模与$\sqrt{|\mathcal{I}|}$成正比，因此我们尤其希望界定花在短区间上的总时间。为此，我们证明，每个完全包含 $\mathcal{I}$ 的低层线程时期都必定是不良时期；正如我们之前所论证的，不良时期的数量不会太多。

单果型专家的区间遗憾保证有一个重要细节尚未解决——只有当我们能够保证在任意区间 $\mathcal{I}$ 上，获得 $O(\sqrt{|\mathcal{I}|})$ 遗憾（相较于存活的活跃专家按“轮次”计算的最小值）时，遗憾分析才成立。这并非易事，而且仅靠“在每个轮次上运行乘法权重更新”已无法保证这一点。尤其是，一个区间可能由来自不同线程的轮次构成；更重要的是，存活的活跃专家集合不一定来自

同一个线程（它们可能来自所有线程）。我们通过一个通用过程 MonocarpicExpert 解决这个问题，该过程能够针对任意存活专家获得区间遗憾。

我们研究休眠专家问题的一种变体 [FSSW97, HS07]，并将其称为单果型专家[6]。在单果型专家问题中，专家可以随时进入和退出池（但每位专家只会苏醒一次），我们希望针对任意存活专家获得区间遗憾保证。也就是说，在专家的生命周期内的任意时间区间 $\mathcal{I}$ 上，算法都能与该专家竞争，遗憾为 $O(\sqrt{|\mathcal{I}|})$。单果型专家问题与休眠专家问题密切相关，但两者的区别在于，休眠专家可以重新进入池（即其生命周期并非连续时段）。本文为单果型专家提出了一种具有区间遗憾保证的低内存算法，其内存占用与存活专家的最大数量成线性关系。该算法通过二分技巧实现，并利用了二阶遗憾保证方面的最新进展（即 Squint 算法 [KVE15]）。

分组技巧 最后，为了以 S 空间获得 $\widetilde{O}(\sqrt{\frac{nT}{S}})$ 遗憾，我们将专家划分为 $\widetilde{O}(S)$组，每组包含 $O(n/S)$ 名专家。我们为每组维护一个独立的算法元线程，并在这 $O(S)$ 个算法线程之上运行乘法权重更新。这将专家规模从 n 降至 $\widetilde{O}(n/S)$，并达到 $\widetilde{O}(\sqrt{\frac{nT}{S}})$ 遗憾。

2.2 自适应对手的下界

我们首先阐述针对低内存（在线学习）算法构造自适应对手的总体原则。该对手

1. 攻击算法选取的最近 $S_r = \Omega(S)$ 个动作；
2. 维持一组特殊专家 I ($|I| = \Omega(S) \gg S_r$)，它们大部分时间表现良好。

直观而言，第一个原理展示了自适应对手在内存方面的优势。低空间在线学习算法只会在内存中保留 S 个“好”专家；而自适应算法可以在算法释放这些专家后让它们变坏。第二个原理保证在 I 中存在一个好专家，这是因为在任意时刻，I 中的大多数专家都不会受到攻击（即在最近 S_r 轮中没有被选中），因而损失较小。

为简化表述，我们关注空间 $S = o(\sqrt{n})$ 的情形，并取特殊专家的数量为 $N = \sqrt{n}$。对手首先抽取 $(x_A, x_B) \in \{0,1\}^n \times \{0,1\}^n \sim \mu^N$，其中 μ 是集合不相交问题的某个困难分布，包含 $n = N^2 = n$ 个坐标。损失序列被划分为 $\frac{10T}{N}$ 个时期，每个时期包含 $\frac{N}{10}$ 天。每个时期的损失序列分别构造。在每个时期内，损失向量以相同概率 $1/2$ 设为 $1 - x_A$ 或 $1 - x_B$；但算法在当前时期内选过的坐标除外，这些坐标的损失固定为 $1/2$。容易看出，整个序列中最佳专家的损失至多为 $\frac{T}{20}$。要获得 $o(T)$ 遗憾，在线学习算法必须在某个时期至少确定 $\Omega(N)$ 个不同的相交坐标（来自 N 个集合不相交实例）；否则，算法受到的损失为 $\frac{T}{2} - o(T)$。

一个自然的三方通信问题可以表述如下：有一位先知 Charlie，知晓所有的 x_A 和 x_B，但在开始时只允许发送 S 比特的信息（这对应于一个轮次的初始内存状态）。Alice 和 Bob 分别接收输入 x_A和 x_B，并且最多可以通信 $\frac{N}{10} \cdot S$ 比特。目标是在最后输出 $\Omega(N)$ 个相交坐标。

[6]这个名称的灵感来自一次结果植物，也就是只开花一次的植物。

难点在于来自先知的 S 比特建议。我们的关键观察是，可以通过寻找直接乘积定理来消除这些建议。具体来说，一个好得令人难以置信的三方通信协议将意味着存在一个双方通信协议，能够解决 N份集合不相交问题，总通信量为 $S \cdot \frac{N}{10} = o(n)$ 比特，且成功概率至少为 $\exp(-O(S))$（无需建议！）。粗略地说，这是因为他们可以利用公共随机性来猜测这些建议，且平均而言，猜对的概率为 2^{-S}。集合不相交问题要求每份副本的内部信息成本为 $\Omega(\sqrt{n})$，而 [BW15] 的直接乘积定理 [表明，要以至少 $\exp(-O(N)) \ll \exp(-O(S))$ 的成功概率解决问题，必须进行 $\Omega(n)$ 比特的通信。

2.3 对抗自适应对手的算法

我们针对自适应对手的算法与针对非自适应对手的算法截然不同，需要一系列全新的思路。回顾一下，我们的目标是设计一种使用 $O(\sqrt{n}/\epsilon)$ 空间并获得 ϵT 遗憾的算法。一个自然的想法是，在真正采用某个专家的动作（或将其加入池中）之前，先观察该专家一段时间，但暂不采用其动作。然而，挑战在于，自适应对手可以轻易地反制这类算法：一旦算法选用某个专家，对手就让该专家表现糟糕。换言之，尽管找到这样一位优秀专家需要付出巨大的努力、面临很大困难，但它似乎一经揭晓就会变得毫无用处。

我们的解决方案在概念上很简单，由两个部分组成。我们先给出一个次优版本。

- RandomExpert。RandomExpert 过程会在每个轮次中对一组新的 $\sqrt{n}/\epsilon$ 专家进行子采样，每个轮次包含 $1/\epsilon^2$ 天。RandomExpert 的目标是在该轮次内，相对于排名前 $\sqrt{n}/\epsilon$ 的专家达到 ϵ-竞争性（即比该轮次中排名第 $\sqrt{n}/\epsilon$ 的专家至多多承受 ϵ 平均损失）。为应对前述对手在专家被选中后立即使其表现变差的挑战，我们采用一种常见技巧，使算法能够对抗自适应对手：RandomExperts 将集合划分为 $1/\epsilon^2$ 个组，每组包含 $\epsilon\sqrt{n}$ 位专家。它对每组专家分别运行乘法权重更新，并在第 b ($b \in [1/\epsilon^2]$) 天遵循第 b 组的决策。该算法保证相对于排名前 $\sqrt{n}/\epsilon$ 的专家达到 ϵ-竞争性，因为以高概率（w.h.p.），RandomExpert 会为每个组从排名前 $\sqrt{n}/\epsilon$ 位的专家中采样一位。

- LongExpert。LongExpert 过程维护一个由 $\mathcal{P}$ 个专家组成的池，并在每个轮次中对 $\mathcal{P}$ 执行乘法权重更新。每个轮次中，它会采样一组新的 $\sqrt{n}/\epsilon$ 个专家，并且只观察它们的损失。在轮次结束时，它会将这些专家的表现与 RandomExpert 进行比较；如果某个专家的表现比 RandomExpert 好 $\Omega(\epsilon)$，就会将其加入 $\mathcal{P}$。该专家会保留 $\sqrt{n}$ 个时期。

最终算法对 RandomExpert 和 LongExpert 过程执行乘法权重更新。其分析相当直观。

- **内存界**。LongExpert 每个轮次预留 $O(1/\epsilon^2)$ 个专家，因为只会保留排名靠前的 $\sqrt{n}/\epsilon$ 个专家（这是由 RandomExpert 的遗憾保证得出的），并且每个轮次会采样 $\sqrt{n}/\epsilon$ 个专家。一个专家会在 $\mathcal{P}$ 中保留 $O(\sqrt{n})$ 个时期，因此使用的总内存为 $O(\sqrt{n}/\epsilon^2)$。

- **遗憾分析**。对于任意专家 $i \in [n]$，如果它不属于排名前 $\sqrt{n}/\epsilon$ 的专家，或者它并未比 RandomExpert 差 ϵ，那么它在一个轮次中造成的遗憾至多为 ϵ。另一方面，

（也就是说，它是排名前 $\sqrt{n}/\epsilon$ 的专家，并且表现明显优于 RandomExpert），它以概率 $\frac{1}{\epsilon\sqrt{n}}$ 被采样，并保留 $\sqrt{n}$ 个时期（在这 $\sqrt{n}$ 个时期内造成 ϵ 的平均遗憾）。直观地说，这意味着，为了让我们的算法在某个轮次表现不佳（且每个轮次造成的平均遗憾至多为 1），对手应惩罚这位顶尖专家 $\sqrt{n}$ 个时期，使平均顶尖专家的损失在期望上增加 $\frac{1}{\epsilon\sqrt{n}} \cdot \sqrt{n} = 1/\epsilon$ 个时期，因此得到平均 ϵ 的遗憾。

优化遗憾。上述算法使用 $O(\sqrt{n}/\epsilon^2)$ 的空间，获得 $O(\epsilon T)$ 的遗憾。次优之处在于，对于任意 $\alpha \in (\epsilon, 1]$，与 RandomExpert 相比仅好 α 的专家都被一视同仁。为避免这种低效，我们维护 $R = \log_2(1/\epsilon)$ 个 RandomExpert 线程。时期大小仍为 $1/\epsilon^2$，并令 $\epsilon_r = 2^r\epsilon$ ($r \in [R]$)。第 r 个线程中的 RandomExpert 采样 $\epsilon_r^2\sqrt{n}/\epsilon$ 个专家，运行乘法权重更新 $1/\epsilon_r^2$ 天，并在每个时期内重复 ϵ_r^2/ϵ^2 次。可以保证它对排名第 $\epsilon\sqrt{n}/\epsilon_r^2$ 的专家具有 ϵ_r- 竞争性（遗憾保证较弱，但与表现顶尖的专家相比仍可相当）。LongExpert 维护 R 个子池 $\mathcal{P}_1$、.....、$\mathcal{P}_R$；第 r 个池采样 $\sqrt{n}/\epsilon$ 个专家，并观察它们 $1/\epsilon_r^2$ 天的表现（每个时期也会重启 ϵ_r^2/ϵ^2 次）。它只保留那些比第 r 个线程中的 RandomExpert 好 ϵ_r 的专家，并将专家保留 $\epsilon\sqrt{n}$ 个时期（总计 $\sqrt{n}/\epsilon$ 天）。在 $\mathcal{P}$ 上执行 MonocarpicExpert 过程。注意，每个时期平均有 $O(1/\epsilon_r^2) \cdot (\epsilon_r^2/\epsilon^2) = O(1/\epsilon^2)$ 个专家加入线程 r 池 $\times$，并在其中停留 $\epsilon\sqrt{n}$ 个时期，因此内存有界于 $O(\sqrt{n}/\epsilon)$。遗憾分析会变得稍微复杂一些，这里只概述直观思路。如果某个专家比排名第 $\epsilon\sqrt{n}/\epsilon_r^2$ 的专家好 $\Theta(\epsilon_r)$，那么它在 $1/\epsilon_r^2$ 天内平均会遭受 ϵ_r 的损失。同时，以 $1/\epsilon\sqrt{n}$ 的概率，它会被采样并保留在子池 $\mathcal{P}_r$ 中 $\sqrt{n}/\epsilon$ 天。因此，它平均具有 $(1/\epsilon_r) \cdot (\epsilon\sqrt{n})/(\sqrt{n}\epsilon) \leq \epsilon$ 的遗憾。复杂之处在于，该专家在最初 $1/\epsilon_r^2$ 天内可能并没有差 $\Omega(\epsilon_r)$，在这种情况下，RandomExpert 过程只能保证 $O(\epsilon_r)$（而不是 $O(\epsilon)$）的遗憾，因此我们需要在分析中谨慎地归属轮次。技术细节见第5节。

注释 2.1（通过差分隐私实现黑盒归约之外的方法）。人们可能会尝试在我们的非自适应算法之上，利用近期发展起来的差分隐私（差分隐私）*[HKM+20、BKM+22]* 技术，设计一个应对自适应对手的算法。但这种方法不适用于我们的主要设定，因为算法被要求输出实际动作，而不是单个可以通过差分隐私中位数机制隐藏的二元预测值。这似乎是差分隐私方法固有的局限；隐藏实际动作要困难得多，而这对于在线学习算法在更广泛应用中的使用至关重要。[7]

与此同时，如果我们缩小在线学习任务的范围，只要求输出一个二元预测（例如，任务是预测是否会下雨，而专家提供 *0/1* 建议），那么差分隐私技术确实适用，但只能得到次优的遗憾界 $\widetilde{O}(n^{1/4}T)$。下面我们对此作简要说明。

DP 方法的典型设置是维护 K 个非自适应算法的副本，每个副本使用 S' 空间。总空间为 $S = KS'$，我们需要在 K 和 S'之间进行权衡。典型的选择是取 $K = O(\sqrt{T})$，并在每一轮应用差分隐私中位数机制。然而，在线学习与典型的流式应用之间存在显著差异。在典型的流式应用中，每个副本都能以高概率保证正确，因此取中位数是一个好方法；而对于在线学习而言，并不存在这样的保证，而且很容易构造一个例子，说明中位数方法会遭受 $\Omega(1)$ 遗憾。相反，我们应当应用差分隐私均值机制

[7]例如，在博弈论的应用中，对手会观察实际动作；在机器学习的许多应用中，对手会观察算法输出的假设

并选择 $K = \sqrt{T/\epsilon}$，从而使平均遗憾为 ϵ。平衡这些项后，总遗憾的量级为 $\widetilde{\Theta}\left(\frac{n^{1/4}T}{\sqrt{S}}\right)$，这劣于我们的定理 *1.2*。

2.4 论文结构

第3节给出正式定义并介绍背景。我们在第4节给出针对非自适应对手的算法。第5节和第 6 节分别给出针对自适应对手的上界和下界。

3 预备知识

我们考虑带专家建议的在线学习标准模型。

定义 3.1 (带专家建议的在线学习)。 在在线学习问题中，有 n位专家，算法以初始内存状态 M_0 启动。在第 t 天（$t \in [T]$），

- 算法根据其内部状态 M_{t-1} 选择一位专家 $i_t \in [n]$；
- 自然方揭示损失向量 $\ell_t \in [0,1]^n$;
- 算法接收损失 $\ell_t(i_t)$，观察损失向量 ℓ_t，并将其内存状态从M_{t-1} 更新为 M_t。

损失向量由对手对抗性地选择:

- 非自适应对手。非自适应对手预先（随机地）选择损失序列 $\ell_1, \ldots, \ell_T$。
- 自适应对手。自适应对手在第 t 天开始时选择损失 ℓ_t，且该选择可以依赖于过去的动作 $i_1, \ldots, i_{t-1}$。

如果算法的内存使用量至多为 S 字，则称其使用至多 S 字的内存，若 $\max_{t\in[T]} |M_t| \le S$。

算法的表现以遗憾来衡量，其定义为

$$R(T) := \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(i_t) - \min_{i^*\in[n]} \sum_{t=1}^{T} \ell_t(i^*)\right]$$

其中，期望是对算法的随机性以及 ℓ_t 的选择取的。平均遗憾定义为 $R(T)/T$。

符号说明 在在线学习问题中，设 T 为天数总数，n 为专家数量。设 $[n] = \{1, 2, \ldots, n\}$ 和 $[n_1 : n_2] = \{n_1, \ldots, n_2\}$。我们有时使用 $\mathcal{N}$表示专家集合。对于任意整数 t，令 $\mathrm{pw}(t)$ 为满足 t 是 $2^{\mathrm{pw}(t)}$的倍数的最大整数。令 Δ_n 包含 $[n]$上的所有概率分布。

算法 1 乘法权重更新 [AHK12]

1: **对于** $t = 1, 2, \ldots\ldots, T$ **执行** 2: 计算 $p_t \in \Delta_n$ 在专家上的分布，使得 $p_t(i) \propto \exp\left(-\eta \sum_{\tau=1}^{t-1} \ell_\tau(i)\right)$ 对于 $i \in [n]$ 3: 采样一名专家 $i_t \sim p_t$ 并观察损失向量 $\ell_t \in [0,1]^n$ 4: **结束循环**

现有在线学习算法的遗憾保证 我们采用在线学习文献中的现有算法。乘法权重更新（见算法 1）是最基础的算法。

引理 3.2 ([AHK12])。设 n、$T \geq 1$、$\eta > 0$ 以及损失 $\ell_t \in [0,1]^n (t \in [T])$。乘法权重更新算法保证

$$\sum_{t=1}^{T} \langle p_t, \ell_t \rangle - \min_{i^* \in [n]} \sum_{t=1}^{T} \ell_t(i^*) \leq \frac{\log n}{\eta} + \eta T,$$

取 $\eta = \sqrt{\log(n/\delta)/T}$，以至少 $1-\delta$ 的概率，

$$\sum_{t=1}^{T} \ell_t(i_t) - \min_{i^* \in [n]} \sum_{t=1}^{T} \ell_t(i^*) \leq O\left(\sqrt{T \log(n/\delta)}\right).$$

定义 $v_t(i) = \langle p_t, l_t \rangle - \ell_t(i)$ 为专家 i 与算法之间的损失差，Squint 算法（见算法 2）提供二阶保证。

算法 2 SQUINT [KVE15]

1: **对于** $t = 1、2、\ldots\ldots、T$ **执行** 2: 计算 $p_t \in \Delta_n$，遍历专家，使得 $p_t(i) \sim \mathbb{E}_\eta \left[\eta \cdot \exp\left(\eta \sum_{\tau=1}^{t-1} v_\tau(i) - \eta^2 \sum_{\tau=1}^{t-1} v_\tau^2(i)\right)\right]$ 3: 采样一位专家 $i_t \sim p_t$ 并观察损失向量 $\ell_t \in [0,1]^n$ 4: **结束循环**

引理 3.3 (定理 3，出自 [KVE15])。假设学习率 η 是从 γ 在 $[0, 1/2]$ 上的先验分布中采样得到的，使得 $\gamma(\eta) \propto \eta^{\frac{1}{2\log(\eta)}}$。对于任意专家 $i \in [n]$，SQUINT 保证

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(i_t) - \sum_{t=1}^{T} \ell_t(i)\right] \leq O\left(\sqrt{V_T^i \log(nT)}\right).$$

其中 $V_T^i = \sum_{t=1}^{T} (\mathbb{E}_{i_t}[\ell_t(i_t)] - \ell_t(i))^2$ 是专家 i的损失方差。

信息论 我们采用信息论中的标准符号。令 X，Y 为随机变量，$H(X)$ 为熵，$I(X;Y)$ 为互信息。在双方通信问题中，内部信息成本定义如下。

定义 3.4 (内部信息)。通信协议Π 在从 μ 中抽取、定义于 $X \times Y$ 上的输入所对应的内部信息成本定义为

$$\mathsf{IC}_\mu(\Pi) := I(X; \Pi | Y) + I(Y; \Pi | X).$$

4 针对非自适应对手的近乎最优界

对于非自适应对手，我们有如下遗憾保证。

定理 1.1（算法，非自适应对手）。设 n、T 为充分大的整数，且多重对数$(nT) \le S \le n$。存在一种在线学习算法，使用至多 S 空间，并能针对非自适应对手实现 $\widetilde{O}\left(\sqrt{\frac{nT}{S}}\right)$ 遗憾，其概率至少为 $1 - 1/(nT)^{\omega(1)}$。

4.1 基线子程序构建区块

我们首先给出一个基线子程序，它能够实现以下遗憾保证。

命题 4.1。设 n、$T \ge 1$ 为充分大的整数，对于任意 $B \in [T]$，存在一种在线学习算法，使用至多 多重对数(nT) 空间，并能针对非自适应对手实现 $\widetilde{O}\left(nB + \frac{T}{\sqrt{B}}\right)$ 遗憾。 —=

基线子程序的伪代码见算法 3–7。基线（即算法 3）按轮次推进，每轮包含 B 天。它维护一个由专家组成的池 $\mathcal{P}$，并在每轮开始时向池中添加 $O(1)$ 个随机专家 $\mathcal{P}$（算法 3 第 3 行）。每轮中，它在 $\mathcal{P}$ 中的专家上运行乘法权重更新。池的大小增长很快，因此算法会在每轮结束时对池进行剪枝。

整个池 $\mathcal{P}$ 被划分为 $L + 1 = \log_2(T) + 1$ 个互不相交的子池 $\mathcal{P} = \mathcal{P}_0 \cup \cdots \cup \mathcal{P}_L$。粗略来说，$\mathcal{P}_\ell$ 仅包含最近 2^ℓ 个轮次中采样的专家，并且每隔 2^ℓ 个轮次就会合并到 $\mathcal{P}_{\ell+1}$ 中。

合并过程（算法4）将子池 $\mathcal{P}_\ell$ 合并到 $\mathcal{P}_{\ell+1}$，并执行后续的剪枝步骤以控制池大小。具体而言，该过程进行 $K = 16\log(nT)$ 次迭代；在第 k 次迭代中，只要子池大小超过（约为）$\log^9(nT)$ 的阈值，就会采样 $\frac{1}{\log^4(nT)}$ 比例的专家 $\mathcal{F}_k$，并用它们筛选 $\mathcal{P}_{\ell+1}$。接下来我们将证明，除非 $\mathcal{P}_{\ell+1}$ 已经很小，否则每次迭代都会移除 $\mathcal{P}_{\ell+1}$中至少 $\Omega(\frac{1}{\log(nT)})$ 比例的专家。我们注意到，大小估计和采样步骤（即算法 5 和 6）都以不敏感 的方式执行，这对内存分析至关重要。

核心思想在于筛选过程（算法 7）：它接收一组专家 $\mathcal{F}$ 和一个池$\mathcal{Q}$；如果专家 i 被 $\mathcal{F}$ 中的专家覆盖，则将其从 $\mathcal{Q}$ 中移除（参见定义 4.2）。

符号说明 为了正式介绍覆盖的概念，我们需要一些符号说明。对于任意轮次$t \in [T/B]$，令 $\mathcal{P}_\ell^{(t)}$ ($\ell \in [0 : L]$) 表示轮次 t 开始时的子池 $\mathcal{P}_\ell$（算法 3 的第 3 行之后），类似地，令 $\mathcal{P}^{(t)}$ 表示同一时刻的整个池。我们的算法采用有放回方式采样专家；即使它在不同轮次中两次采样到同一位专家，我们也将其视为两位独立的专家。对于专家 $i \in \mathcal{P}^{(t)}$，令 $E_i \in [T/B]$ 表示专家i 被采样并进入池中的时间。对于专家 i、$j \in \mathcal{P}^{(t)}$，如果 $E_i \le E_j$，我们记作 $i \preceq j$，即专家 i不晚于专家 j进入池中。令 $\Gamma_{i,\,j}$ 表示时间区间 $\{E_i, \ldots, E_j - 1\}$（如果 j 不晚于 i进入池中，则为 $\Gamma_{i,\,j} = \emptyset$），并稍微滥用符号，令 $\Gamma_{i,\,t}$ 表示时间区间 $\{E_i, \ldots, t\}$。对于任意区间 $\mathcal{I} \subseteq [T/B]$，专家 i 在 $\mathcal{I}$ 上的累积 损失定义为

$$\mathcal{L}_{\mathcal{I}}(i) := \sum_{t \in \mathcal{I}} \sum_{b=1}^{B} \ell_{(t-1)B+b}(i).$$

特别地，定义

$$\mathcal{L}_{\Gamma_{i,j}}(i) := \begin{cases} \sum_{t\in\Gamma_{i,j}} \sum_{b=1}^{B} \ell_{(t-1)B+b}(i) & \Gamma_{i,j} \neq \emptyset \\ +\infty & \Gamma_{i,j} = \emptyset \end{cases}$$

专家 i 在区间 Γ_{i}内的j累积损失。略微滥用符号说明，我们将 $\mathcal{L}_t(i) = \sum_{b=1}^{B} \ell_{(t-1)B+b}(i)$ 设为专家 i 在轮次 t中的总损失。

现在我们可以正式定义覆盖。

定义 4.2（覆盖）。给定一组专家 $\mathcal{F} = \{i_1, \ldots, i_{|\mathcal{F}|}\}$ 和一名专家 $j \in [n]$。令 $i_1 \preceq i_2 \cdots \preceq i_r \preceq j \preceq i_{r+1} \cdots \preceq i_{|\mathcal{F}|}$，其中 $r \in \{0, 1, 2, , \ldots, |\mathcal{F}|\}$。在轮次 t中，如果专家 j被 $\mathcal{F}$覆盖，则称

$$\mathcal{L}_{\Gamma_{j,t}}(j) \geq \min_{i_r^* \in \mathcal{F}} \mathcal{L}_{\Gamma_{j,i_{r+1}}}(i_r^*) + \left(\sum_{s=r+1}^{|\mathcal{F}|} \min_{i_s^* \in \mathcal{F}} \mathcal{L}_{\Gamma_{i_s,i_{s+1}}}(i_s^*) \right) - 1. \tag{1}$$

这里我们略微滥用符号说明，并记为 $\Gamma_{i_{|\mathcal{F}|},\ i_{|\mathcal{F}|+1}} = \Gamma_{i_{|\mathcal{F}|},\ t}$。

最后值得注意的是，基线只需记录 $\mathcal{L}_{\Gamma_{i,\ j}}(i)$ 对于 $i,\ j \in \mathcal{P}$ 的损失，其规模随池的大小呈二次增长。

算法 3基线

初始化池 $\mathcal{P} \leftarrow \emptyset$ 和子池 $\mathcal{P}_\ell \leftarrow \emptyset$ $(\ell \in [0 : L])$ ▷ $L = \log_2(T)$
对于 $t = 1, 2, \ldots, T/B$ **执行** ▷ 轮次 t
  $\mathcal{P}_0 \leftarrow$ 采样$(\mathcal{N}, 1/n)$ ▷ 将新专家采样到池中
  对 $\mathcal{P}$ 运行乘法权重更新，持续 B 天 ▷ $\mathcal{P} = \mathcal{P}_0 \cup \cdots \cup \mathcal{P}_L$
  对于 $\ell = 0, 1, \ldots, \mathsf{pw}(t)$ **执行** ▷ $\mathsf{pw}(t)$ 是满足 $(t/2^{\mathsf{pw}(t)}) \in \mathbb{N}$ 的最大整数
    $\mathcal{P}_{\ell+1} \leftarrow$ 合并$(\mathcal{P}_{\ell+1}, \mathcal{P}_\ell)$
    $\mathcal{P}_\ell \leftarrow \emptyset$
  结束循环
结束循环

算法4 合并$(\mathcal{Q}_A, \mathcal{Q}_B)$

$\mathcal{Q}_C \leftarrow \mathcal{Q}_A \cup \mathcal{Q}_B$ **对于** $k = 1$、2、.....、K **执行** ▷ $K = 16\log(nT)$
$s \leftarrow$ 估计大小$(\mathcal{Q}_C,\ {}^{1}\!/_{\log^4(nT)})$ ▷ 估计 $\mathcal{Q}_C$的大小
如果 $s \leq \log^5(nT)$ **则** 转到第11行
否则 $\mathcal{F}_k \leftarrow$ 采样$(\mathcal{Q}_C,\ {}^{1}\!/_{\log^4(nT)})$ ▷ 采样 $\frac{1}{\log^4(nT)}$-专家的比例
  $\mathcal{Q}_C \leftarrow$ 筛选$(\mathcal{F}_k, \mathcal{Q}_C) \cup \mathcal{F}_k$
结束如果
结束循环
返回 $\mathcal{Q}_C$

算法 5 估计大小$(\mathcal{Q}, p)$

$p_i \leftarrow$ 对每个 $i \in \mathcal{Q}$，伯努利分布(p)

返回 $\sum_{i\in\mathcal{Q}} p_i$

算法 6 采样$(\mathcal{Q}, p)$

$p_i \leftarrow$ 对每个 $i \in \mathcal{Q}$，伯努利分布(p)

返回 $\{i \in \mathcal{Q}\colon\ p_i = 1\}$

算法 7 筛选$(\mathcal{F},\ \mathcal{Q})$

对于每个专家 $i \in \mathcal{Q}$ **执行**

  如果 $\mathcal{F}$ 覆盖 i，**则** ▷ 定义 4.2

    $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{i\}$

  结束如果

结束循环

返回 $\mathcal{Q}$

4.1.1 内存界

我们首先给出基线的内存保证。

引理 4.3 (基线的内存界基线)。 以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率，算法 *3* 使用的总内存至多为 $O(\log^{20}(nT))$ 字。

关键观察来自合并过程的以下保证 .

引理 4.4。 以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率， $\mathcal{Q}_C$ 合并过程合并 过程的输出满足

$$|\mathcal{Q}_C| \le \max\left\{2\log^9(nT), \frac{1}{4}(|\mathcal{Q}_A| + |\mathcal{Q}_B|)\right\}.$$

证明。 假设筛选过程已被调用 $k_{\max} \in [0 :\ K]$ 次，并令$\mathcal{Q}_{C,k}$ 表示第 k 次循环开始时池 $\mathcal{Q}_C$ 的状态。注意到$|\mathcal{Q}_{C,1}| = |\mathcal{Q}_A| + |\mathcal{Q}_B|$。对于任意$k \in [k_{\max}]$，我们的目标是以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率证明，

1. 要么$|\mathcal{Q}_{C,k}| \le 2\log^9(nT)$; 要么
2. $|\mathcal{Q}_{C,k+1}| \le (1 - \frac{1}{6\log(nT)})|\mathcal{Q}_{C,k}|$。

假设我们有 p 证明了该结论后，再对 $k \in [k_{\max}]$ 使用并集界，即可保证

$$\begin{aligned}|\mathcal{Q}_{C,k_{\max}+1}| &\le \max\left\{2\log^9(nT), \left(1 - \frac{1}{6\log(nT)}\right)^{16\log(nT)} |\mathcal{Q}_{C,1}|\right\} \\ &\le \max\left\{2\log^9(nT), \frac{1}{4}(|\mathcal{Q}_A| + |\mathcal{Q}_B|)\right\}.\end{aligned}$$

为证明该结论，固定一次迭代 $k \in [k_{\max}]$ 并假设$|\mathcal{Q}_{C,k}| > 2\log^9(nT)$，只需证明$|\mathcal{Q}_{C,k+1}| \le (1 - \frac{1}{6\log(nT)})|\mathcal{Q}_{C,k}|$ 以高概率成立。首先，它以高概率通过阈值测试（合并的第 5 行），因为

$$\Pr[s \le \log^5(nT)] = \Pr\left[\sum_{i\in\mathcal{Q}_{C,k}} p_i \le \log^5(nT)\right] \le \exp(-\log^4(nT)/4) = 1/(nT)^{\omega(1)}.$$

第二步由切尔诺夫界以及以下事实得出：(1) $\{p_i\}_{i\in\mathcal{Q}_{C,k}}$ 是均值为 $\frac{1}{\log^4(nT)}$ 的相互独立的伯努利变量；(2) $|\mathcal{Q}_{C,k}| \geq 2\log^9(nT)$。

接下来，我们分析筛选过程，并证明以高概率（针对 $\mathcal{F}_k$ 的选择），至少有 $\frac{1}{6\log nT}$ 比例的专家会被移除。我们通过迭代证明这一组专家确实存在。

- 在分析的第 r 次迭代中，我们确定一个子集 R_r，其中的专家会以高概率（针对 $\mathcal{F}_k$ 的选择）被移除。如果 R_r 不太小（占子池的 $\Omega(1/\log(nT))$ 比例），我们就能保证子池规模缩小，因此问题得以解决。

- 否则，我们会从 $\mathcal{F}_k$ 中确定 r 个枢轴专家，并找到子集 $\mathcal{W}_{r+1}$ 中一个较大的部分（占 $r/\log(nT)$ 的 $1-O$），其损失相当大。具体而言，考虑由 $\mathcal{F}_k$ 中的 r 个专家所跨越的轮次后缀。在这个后缀期间，$\mathcal{W}_{r+1}$ 中的任何专家都由这些 r 个专家覆盖；此外，即使给 $\mathcal{W}_{r+1}$ 中的专家加上 2^{r-1} 的损失减少，这一点仍然成立。如果损失减少超过 T 的总可能损失，这种情况就不可能继续成立，因此对于某个 $r \leq \log(nT)$，前述 R_r 较大的情形必须成立。

为分析设定符号： 令 $M = |\mathcal{Q}_{C,k}|$。对于任意 $r \geq 1$，我们迭代定义

$$\mathcal{W}_{r+1} := \mathcal{W}_r \backslash (R_r \cup O_r \cup D_r(j_r)) \quad \text{and} \quad \mathcal{W}_1 = \mathcal{Q}_{C,k}.$$

j_r, R_r, O_r 和 $D_r(j_r)$ 的定义将在后文给出。

我们首先为每个专家 $i \in \mathcal{W}_r$ 定义集合 $D_r(i)$，令

$$D_r(i) := \{j \in \mathcal{W}_r \setminus \{i\}, \mathcal{L}_{\Gamma_{i,j_{r-1}}}(i) \geq \mathcal{L}_{\Gamma_{i,j_{r-1}}}(j) - 2^{r-1}\}, \forall i \in \mathcal{W}_r.$$

这里我们稍微滥用符号，令 $\Gamma_{i,j_0} = \Gamma_{i,t}$。换言之，集合 $D_r(i)$ 包含在区间 $\Gamma_{i,j_{r-1}}$ 上与专家 i 可比较的专家。

接下来我们定义集合 R_r，令

$$R_r = \{i : i \in \mathcal{W}_r, |D_r(i)| \geq \log^6(nT)\}.$$

集合 R_r 包含所有规模 $D_r(i)$ 较大的专家 $i \in \mathcal{W}_r$；也就是说，我们将论证，这些专家很容易被移除，因为它们被 $j_1, \ldots, j_{r-1}$ 与 D_r 中任意一个专家的组合所覆盖。

现在，将专家 j_r 定义为 $\mathcal{W}_r \backslash R_r$ 中最新且也被采样到 $\mathcal{F}_k$中的专家，即，

$$j_r := \text{argmax}_{j \in (\mathcal{W}_r \backslash R_r) \cap \mathcal{F}_k} E_j.$$

注意，如果不存在这样的专家，即 $(\mathcal{W}_r \backslash R_r) \cap \mathcal{F}_k = \emptyset$，则将其留空。

集合 O_r 包含比 j_r更晚进入池中的专家，即

$$O_r = \{i : i \in \mathcal{W}_r \backslash R_r, E_i \geq E_{j_r}\}.$$

归纳假设。 设 $r \geq 1$，我们的归纳假设以事件 $R_m \leq \frac{1}{4\log(nT)} M$ 对所有 $m \in [r]$均成立为条件。展望后续，这些都是简单情形，因为专家集合 R_m很容易被移除。我们通过归纳证明

1. $|\mathcal{W}_{r+1}| \geq (1 - \frac{r}{2\log(nT)})M$;
2. 专家 j_1、……、$j_r \in \mathcal{F}_k$并遵循 $j_r \preceq j_{r-1} \preceq \cdots \preceq j_1$ 的顺序；

3. 对于任意 $m \in [r]$ 和专家 $i \in \mathcal{W}_{r+1}$,

$$\mathcal{L}_{\Gamma_{j_m,j_{m-1}}}(i) \geq \mathcal{L}_{\Gamma_{j_m,j_{m-1}}}(j_m) + 2^{m-1}. \tag{2}$$

我们通过归纳法证明这些结论，并注意到 $r = 0$ 的基础情形显然成立。假设归纳法对 $r-1$ 之前的情形均成立。

对 r 次迭代进行分析。 现在我们考虑两种情况：第一种情况下，易移除专家 R_r 的集合很大，因此我们可以直接将它们移除！第二种情况下，虽然很难用单个专家覆盖典型专家，但我们可以继续进行归纳，并证明最终可以通过将多个专家汇集在一起，使典型专家得到覆盖。

情况 1。 如果 $|R_r| \geq \frac{1}{4\log(nT)} M$。我们证明，以高概率，$R_r$ 中的专家会被过滤过程移除。对于任意专家 $i \in R_r$，每个专家 $j \in D_r(i)$ 以（独立的）概率 $\frac{1}{\log^4(nT)}$ 被纳入 $\mathcal{F}_k$，因此，

$$\Pr\left[j \notin \mathcal{F}_k, \forall\, j \in D_r(i)\right] \leq \left(1 - \frac{1}{\log^4(nT)}\right)^{\log^6(nT)} \leq 1/(nT)^{\omega(1)}.$$

第一步源于对任意 $i \in R_r$，$|D_r(i)| \geq \log^6(nT)$ 均成立。

因此，以至少 $1-1/(nT)^{\omega(1)}$ 的概率，存在至少一个专家 $j \in D_r(i) \cap \mathcal{F}_k$。我们声称，专家 i 被 $j, j_{r-1}, j_{r-2}, \ldots, j_1$ 覆盖。特别地，我们有

$$\begin{aligned}
\mathcal{L}_{\Gamma_{i,t}}(i) &= \sum_{m=1}^{r-1} \mathcal{L}_{\Gamma_{j_m,j_{m-1}}}(i) + \mathcal{L}_{\Gamma_{i,j_{r-1}}}(i) \\
&\geq \sum_{m=1}^{r-1} (\mathcal{L}_{\Gamma_{j_m,j_{m-1}}}(j_m) + 2^{m-1}) + \mathcal{L}_{\Gamma_{i,j_{r-1}}}(j) - 2^{r-1} \\
&= \sum_{m=1}^{r-1} \mathcal{L}_{\Gamma_{j_m,j_{m-1}}}(j_m) + \mathcal{L}_{\Gamma_{i,j_{r-1}}}(j) - 1,
\end{aligned}$$

其中，我们在第一步中拆分了 i 的损失；第二步由归纳假设（见方程 (2)）和 $j \in \mathcal{D}_r(i)$ 得出。

因此，以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率，专家集合 $R_r \setminus \mathcal{F}_k$ 会被移除。注意，$|R_r| \geq \frac{1}{4\log(nT)} M$，且 $\mathcal{F}_k$ 的大小满足

$$\Pr\left[|\mathcal{F}_k| \geq \frac{2}{\log^4(nT)} M\right] \leq \exp(-M/4\log^4(nT)) \leq 1/(nT)^{\omega(1)},$$

第一步由切尔诺夫界得出。

应用并集界可得，$|R_r \setminus \mathcal{F}_k| \geq \frac{1}{6\log(nT)} M$ 以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率成立，并且我们已经确定了一个待移除的大集合。

情形 2. 如果 $|R_r| \leq \frac{1}{4\log(nT)} M$。回顾一下，$j_r$ 定义为被采样到 $\mathcal{F}_k$ 中的 $\mathcal{W}_r \setminus R_r$ 的最新专家。以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率，晚于 j_r 到达的专家至多有 $\log^6(nT)$ 个，即 $|O_r| \leq \log^6(nT)$。回顾一下，集合 $\mathcal{W}_{r+1}$ 递归定义为 $\mathcal{W}_{r+1} = \mathcal{W}_r \setminus (R_r \cup O_r \cup D_r(j_r))$。

我们证明归纳假设仍然成立。首先， $\mathcal{W}_{r+1}$ 的大小满足

$$\begin{aligned}|\mathcal{W}_{r+1}| &\geq |\mathcal{W}_r| - |R_r| - |O_r| - |D_r(j_r)| \\ &\geq \left(1 - \frac{r-1}{2\log(nT)}\right) M - \frac{1}{4\log(nT)} M - \log^6(nT) - \log^6(nT) \\ &\geq \left(1 - \frac{r}{2\log(nT)}\right) M.\end{aligned}$$

其次，根据定义， $j_r \in \mathcal{F}_k$ 显然成立，且它早于 j_{r-1}、……、 j_1。第三个结论成立，因为 $D_r(j_r) \cap \mathcal{W}_{r+1} = \emptyset$，我们已经完成了归纳法。

最后，如果事件 $R_r < \frac{1}{4\log(nT)} M$ 持续成立，直到 $r = \log(nT) + 1$，那么对于每个专家 $i \in \mathcal{W}_r$，都有 $D_r(i) = \mathcal{W}_r$，因为损失至多为 T。因此，有 $|R_r| = |\mathcal{W}_r| \geq \frac{1}{2} M$。证明到此结束。 □

现在我们可以界定池大小。

引理 4.5 (子池的内存界)。对于任意轮次 $t \in [T/B]$ 和子池 $\ell \in [0 : L]$，以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率，有 $|\mathcal{P}_\ell^{(t)}| \leq 2\log^9(nT)$。

证明。我们对 ℓ 进行归纳法证明。对于基础情形，在任意轮次 t 中，子池 $\mathcal{P}_0^{(t)}$ 由在轮次 t 开始时采样的专家组成，并且有

$$\Pr[|\mathcal{P}_0^{(t)}| \geq 2\log^9(nT)] \leq \exp(-\log^9(nT)/3) \leq 1/(nT)^{\omega(1)},$$

这来自 切尔诺夫界以及每位专家以 $1/n$ 的概率被采样这一事实 .

假设归纳法对 $\ell - 1$ ($\ell \geq 1$) 之前的情形均成立，子池 $\mathcal{P}_\ell$ 在开始时为空，并且每经过 $2^{\ell-1}$ 个时期便与 $\mathcal{P}_{\ell-1}$ 合并一次。也就是说，Merge 的输入子池（$\mathcal{P}_\ell$ 和 $\mathcal{P}_{\ell-1}$）的大小均不超过 $2\log^9(nT)$，因此根据引理 4.4，以至少$1 - 1/(nT)^{\omega(1)}$的概率，输出子池 $|\mathcal{P}_\ell^{(t)}| \leq 2\log^9(nT)$。 □

现在我们可以证明引理 4.3 了。

证明 引理 *4.13* 的证明。根据引理 4.5，在轮次 t，池的大小有如下界： :

$$|\mathcal{P}^{(t)}| \leq \sum_{\ell=0}^{L} |\mathcal{P}_\ell^{(t)}| \leq 2\log^9(nT) \cdot 2\log(nT) = O(\log^{10}(nT)).$$

为了执行过滤过程，只需记录专家 $i \in \mathcal{P}^{(t)}$ 在区间 $\Gamma_{i,\,j}$ 上的损失；对于任意 $i,\ j \in \mathcal{P}^{(t)}$，每位专家需要 $O(\log^{10}(nT))$ 个字，总共最多需要 $O(\log^{20}(nT))$ 个字的内存。至此，我们完成了基线的内存分析。 □

4.1.2 遗憾分析

令 i^* 表示最佳专家。从整体上看，遗憾分析考虑两种类型的轮次，可以非正式地描述如下：

- 如果我们在某个轮次中采样了 i^*，而它最终会被覆盖并从池中移除，那么该轮次就是“良好轮次”。考虑从当前良好轮次开始，直到 i^* 从池中移除为止的区间：我们认为，即使从未采样过 i^*，算法在整个区间上的表现仍与i^* 相当。

- “不良”轮次的集合记为 $\mathcal{H}$，这些轮次中，i^* 会一直存活到数据流结束。

我们的计划是论证不良时期的数量必须很少——否则池中就会塞满许多份 i^*；但我们已经在引理 4.12 中证明，池大小是多重对数级的。对于每个不良时期，我们无法控制遗憾，但不良时期带来的总遗憾很小，因为不良时期并不多。（正如我们上文所解释的，在良好轮次中，可以保证算法具有较低的遗憾。）

上述方法存在一个微妙的问题[8]：我们希望通过考虑这样一种假设情形来分析遗憾，即在某个轮次将 i^* 采样到池中。然而，采样 i^*也会影响此后池中的其余部分，因为 i^* 可能会逐出其他专家；逐出某些专家可能使池保留本应被逐出的其他专家，等等。这使得比较算法实际使用的池与包含 i^*的假设池变得非常困难。

为处理上述细微之处，我们精心设计了算法，使得即使在已知 i^*已被采样进入池的条件下，以高概率它也不会在采样和估计大小子过程中被考虑。因此，它不会干扰池中的其他部分。

随机位 首先，了解池选择所用的随机位很有帮助。对于每个轮次 $t \in [T/B]$，令 $\xi_t = (\xi_{t,1}, \ldots, \xi_{t,n})$，其中 $\xi_{t,i}$用于专家 i ($i \in [n]$)。随机位$\xi_{t,i} = (\xi_{t,i,1}, \xi_{t,i,2})$由两部分组成，其中 $\xi_{t,i,1} \in \{0,1\}$ 是均值为 $1/n$的伯努利变量，用于将专家 i 采样到池中（即算法 3 的第 3 行）。当且仅当 $\xi_{t,i,1} = 1$时，基线才会在轮次 t 添加专家 i。随机位的第二部分 $\xi_{t,i,2} \in \{0,1\}^{2L \times 2K}$用于调用合并过程时的估计大小和采样。$\xi_{t,i,2}$ 的每个坐标都是均值为 $\frac{1}{\log^4(nT)}$ 的伯努利随机变量；当对应的随机位等于 1 时，专家 i 就会被采样以用于采样和估计大小。———

我们引入被动/主动专家的概念。

定义 4.6（活跃/被动专家）。 在任意轮次 $t \in [T/B]$ 中，如果 ξ_t被动，则称专家 i 为，$\xi_{t,i,2} = \vec{0}$。否则称其为活跃专家。

轮次分配 固定损失序列 ℓ_1、.....、ℓ_T以及每个轮次 t 的采样且活跃的专家集合$Y_t \subseteq [n]$ （注意，采样且被动的专家集合尚未确定）。关键观察是，在 Baseline 的执行过程中，估计大小 s、筛选集 $\mathcal{F}$ 以及存活的活跃专家集合在任意时刻都是固定的。它们完全由损失序列 $\{\ell_t\}_{t\in[T]}$ 和处于活跃状态时被采样的专家 $\{Y_t\}_{t\in[T/B]}$决定。

我们将整个时期分为两类。考虑如下轮次分配机制。我们从 $\tau \leftarrow 1$、 $a_1 \leftarrow 1$ 和 $\mathcal{H} \leftarrow \emptyset$开始。这里，集合 $\mathcal{H} \subseteq [T/B]$ 收集了所有算法无法控制其遗憾的坏时期。令 $t(a_\tau) \in [a_\tau : T/B] \cup \{+\infty\}$为满足以下条件的首次时间：$i^*$ （进入时间为 a_τ）在轮次$t(a_\tau)$的存活的活跃专家集合所覆盖。注意，无论 i^* 是否被采样，覆盖的概念都有明确定义。我们分两种情况讨论。

- 专家 i^* 从未被覆盖（即 $t(a_\tau) = +\infty$）。然后，我们扩充不良时期集合$\mathcal{H} \leftarrow \mathcal{H} \cup \{a_\tau\}$，并更新 $a_{\tau+1} = a_\tau + 1, \tau \leftarrow \tau + 1$;
- 否则，更新 $a_{\tau+1} \leftarrow t(a_\tau) + 1$ 和 $\tau \leftarrow \tau + 1$。

[8]这个问题很微妙，但修正它或许是提出该算法并对其进行分析时最困难的部分。

当 a_τ 达到 $\frac{T}{B}$ 时，分配过程终止，并令 $\tau_{\max}$ 为步骤总数。我们根据上述分配过程分解基线的遗憾，并将其界定为不良时期 $\mathcal{H}$的规模。

引理 4.7。 固定损失序列以及每个轮次中采样且处于活跃状态的 专家集合后，对于 *MWU* 的选择，以至少 $1-1/(nT)^{\omega(1)}$ 的概率，有

$$\sum_{t=1}^{T/B}\sum_{b=1}^{B}\ell_{(t-1)B+b}(i_{(t-1)B+b})-\sum_{t=1}^{T/B}\mathcal{L}_t(i^*)\le O\left(\frac{T}{\sqrt{B}}\log(nT)\right)+B\cdot|\mathcal{H}|.$$

证明。 令 i_t^* 为轮次 t 中池 $\mathcal{P}^{(t)}$ 里的最佳专家，则以至少 $1-1/(nT)^{\omega(1)}$ 的概率，我们有

$$\begin{aligned}
&\sum_{t=1}^{T/B}\sum_{b=1}^{B}\ell_{(t-1)B+b}(i_{(t-1)B+b})-\sum_{t=1}^{T/B}\mathcal{L}_t(i^*)\\
=&\sum_{t=1}^{T/B}\sum_{b=1}^{B}\ell_{(t-1)B+b}(i_{(t-1)B+b})-\sum_{t=1}^{T/B}\mathcal{L}_t(i_t^*)+\sum_{t=1}^{T/B}\mathcal{L}_t(i_t^*)-\sum_{t=1}^{T/B}\mathcal{L}_t(i^*)\\
\le&\sum_{t=1}^{T/B}\mathcal{L}_t(i_t^*)-\sum_{t=1}^{T/B}\mathcal{L}_t(i^*)+\frac{T}{B}\cdot O\left(\sqrt{B}\log(nT)\right)\\
=&\sum_{t\in\mathcal{H}}\mathcal{L}_t(i_t^*)-\mathcal{L}_t(i^*)+\sum_{\tau=1}^{\tau_{\max}}\sum_{t\in[a_\tau:a_{\tau+1}-1]\setminus\mathcal{H}}\mathcal{L}_t(i_t^*)-\mathcal{L}_t(i^*)+O\left(\frac{T}{\sqrt{B}}\log(nT)\right)\\
\le&B\cdot|\mathcal{H}|+\tau_{\max}+O\left(\frac{T}{\sqrt{B}}\log(nT)\right)\\
\le&B\cdot|\mathcal{H}|+O\left(\frac{T}{\sqrt{B}}\log(nT)\right).
\end{aligned}$$

第二步由 MWU 的遗憾保证得出。我们根据第三步中的分配来分解遗憾。第四步由 $\mathcal{L}_t(i_t)\le B$ 的朴素界以及覆盖的定义得出。最后一步由 $\tau_{\max}\le\frac{T}{B}$ 得出。证明完毕。 □

接下来我们证明 $\mathcal{H}$ 以高概率较小。

引理 4.8。 以至少 $1-1/(nT)^{\omega(1)}$ 的概率， $|\mathcal{H}|\le O(n\,\text{对数}^{10}(nT))$.

证明。 我们首先固定损失序列 ℓ_1、.....、ℓ_T以及每个轮次 t中采样且活跃的专家集合 Y_t。我们证明，在末尾池 $\mathcal{P}$ 的大小满足

$$\Pr\left[|\mathcal{P}|\ge\frac{|\mathcal{H}|}{4n}\mid Y_1,\dots,Y_{T/B},\ell_1,\dots,\ell_T\right]\ge\frac{1}{2},$$

只要 $|\mathcal{H}|\ge\Omega(n)$。这里的概率取自其余采样且不活跃的专家的随机性。

令 W_t 为轮次 t 中采样且被动的专家集合。对于任意轮次 $t\in\mathcal{H}$，根据轮次分配机制，专家 $i^*\notin Y_t$ （否则，它在轮次 t 会被自身覆盖）。以 $\{Y_t\}_{t\in[T/B]}$为条件， $i^*\in W_t$ 满足的概率为

$$\begin{aligned}
\Pr[i^*\in W_t|Y_1,\dots,Y_{T/B},\ell_1,\dots,\ell_T]=&\Pr[i^*\in Y_t|i^*\notin Y_t]\\
\ge&\Pr[i^*\in W_t]=\frac{1}{n}\cdot\left(1-\frac{1}{\log^4(n)}\right)^{2L\times2K}\ge\frac{1}{2n}.
\end{aligned}$$

关键的观察是，一旦 $i^* \in W_t$，也就是说， i^* 在轮次 t 被采样且处于被动状态，它就会一直存活到最后。这是因为筛选集 $\mathcal{F}$ 和估计大小 s 完全由采样且活跃的专家 $\{Y_t\}_{t\in[T/B]}$ 以及损失序列决定，而这些都已经固定。因此，如果被动专家 i^* 在轮次 t 加入，它就会一直存活到最后。

由于在给定损失序列和 $\{Y_t\}_{t\in[T/B]}$的条件下， $i^* \in W_t$ 的事件相互独立，因此根据切尔诺夫界，在轮次 T/B 结束时，池 $\mathcal{P}$ 满足

$$\Pr\left[|\mathcal{P}| \le \frac{|\mathcal{H}|}{4n} \mid Y_1,\dots,Y_{T/B},\ell_1,\dots,\ell_T\right] \le \Pr\left[\sum_{t\in\mathcal{H}} 1\{i^* \in W_t\} \le \frac{|\mathcal{H}|}{4n} \mid Y_1,\dots,Y_{T/B},\ell_1,\dots,\ell_T\right] \le \exp(-|\mathcal{H}|/16n). \tag{3}$$

注意，我们已在引理 4.3 中证明， $|\mathcal{P}| \ge \Omega(\log^{10}(nT))$ 发生的概率至多为 $1/(nT)^{\omega(1)}$。由方程 (3) 可知，这意味着

$$\Pr[|\mathcal{H}| \ge \Omega(n\log^{10}(nT))] \le 1/(nT)^{\omega(1)}.$$

证明到此结束。 □

结合引理 4.7 和引理 4.8，我们得到命题 4.1 的遗憾界。命题 4.1 的内存保证已在引理 4.3 中确立。

4.2 提升

通过维护多个基线线程并谨慎地进行遗憾摊销，我们实现了定理 1.1 的最优遗憾保证。新的算法要素包括：(1) 利用基线的高频运行中的池选择，供基线的低频运行使用；(2) 一种具有区间遗憾保证的低内存单生专家算法。

我们首先得到空间 S 较小时的最优速率。

命题 4.9。 当 n、$T \ge 1$ 足够大时，存在一种在线学习算法，使用至多 多重对数(nT) 空间，并能在面对非自适应对手时实现 $\widetilde{O}\left(\sqrt{nT}\right)$ 遗憾。

完整算法 完整算法如算法 8 所示。不失一般性，我们假设 T/n 是 2 的塔。设 $R = \log_2(T/n)$ 为线程总数。对于 $r \in [2\colon R]$，令 $B_r = \frac{T}{n2^{r-1}}$ $(r \in [R])$、 $T_1 = T$ 和 $T_r = B_{r-1}$。完整算法维护 Baseline+算法的 R 个不同线程，其中第 r 个线程（$r \in [R]$）Baseline+(r) 每隔 T_r 天重启一次，时期大小为B_r。Baseline+ 与 Baseline 略有不同：(1) 在每轮次开始时，除了采样 $O(1)$ 个专家外，还会继承来自更高线程池的专家；(2) 它只维护自己的池（运行合并过程），而不在该轮次内运行乘法权重更新。换言之，Baseline+(r)（$r \in [R]$）负责确定“存活”的专家，而遗憾则由一个单独的过程 MonocarpicExpert 控制。MonocarpicExpert 过程不提供按时期遗憾保证，而是保证在专家生命周期的任意 区间 $\mathcal{I}$ 上实现 $\widetilde{O}(\sqrt{|\mathcal{I}|})$ 遗憾。 —

4.2.1 MonocarpicExpert

我们完整算法的一个重要组成部分是 MonocarpicExpert 过程。在单果型专家问题中，不同于所有专家一开始就已出现，专家可以随时进入和退出（但每位专家只会醒来一次）。MonocarpicExpert

算法 8 完整算法

初始化基线+(r) ($\forall\, r \in [R]$) ▷ 基线+(r)维护池$\mathcal{P}_{r,\cdot}$
  $\mathcal{P} \leftarrow \mathcal{P}_{1,\cdot} \cup \cdots \cup \mathcal{P}_{R,\cdot}$ ▷ $\mathcal{P}$ 汇总所有线程的池
对于 $t = 1, 2, \ldots, T$ **do**
    运行基线+(r) ($r \in [R]$)
    在 $\mathcal{P}$上运行 MonocarpicExpert
结束循环

算法 9基线+(r) ▷ $r \in [R]$

对 $s = 1, 2, \ldots, T/T_r$ **执行** 第▷ s 次重启
  启动池 $\mathcal{P}_{r,\ \cdot} \leftarrow \emptyset$ 和子池 $\mathcal{P}_{r,\ \ell} \leftarrow \emptyset$ ($i \in [0 : L]$)
  对于 $t = 1, 2, \ldots, T_r/B_r$ **执行** ▷ 轮次 t
    // 在轮次 t 开始时
    如果 r 是开启新轮次的线程中编号最小的线程**那么**
      对于 $r' = R, \ldots, r+1$ **执行**
        $\mathcal{P}_{r,0} \leftarrow$ 合并($\mathcal{P}_{r,0} \cup \mathcal{P}_{r',\cdot}$) ▷ 从更高层的线程池继承
      结束循环
    结束条件判断
    $\mathcal{P}_{r,0} \leftarrow \mathcal{P}_{r,0} \cup$ 采样($\mathcal{N}, 1/n$) ▷ 将新专家采样到池中

    // 在第 t 轮次结束时
    对于 $\ell = 0, 1, \ldots,$ pw(t) **执行**
      $\mathcal{P}_{r,\ell+1} \leftarrow$ 合并($\mathcal{P}_{r,\ell+1}, \mathcal{P}_{r,\ell}$)
      $\mathcal{P}_{r,\ell} \leftarrow \emptyset$
    结束循环
  结束循环
结束循环

该过程在专家存活期间的每个区间内都能实现较低的遗憾，同时其内存使用量随存活专家的最大数量增长。

定理 4.10。 设 $T \geq 1$。对于在区间 $\mathcal{I} \subseteq [T]$内存活的任意专家 i，MONOCARPIC-EXPERT 过程（算法 *10*）保证以至少 $1 - 1/$（nT）$^{\omega(1)}$ 的概率，

$$\sum_{t \in \mathcal{I}'} \ell_t(i_t) - \sum_{t \in \mathcal{I}'} \ell_t(i) \leq O\left(\sqrt{|\mathcal{I}'| \log(nT)}\right)$$

对于每个区间 $\mathcal{I}' \subseteq \mathcal{I}$，在自适应对手下均成立。此外，假设存活专家的最大数量至多为 M，则 MONOCARPICEXPERT 过程最多使用 O（M 对数 ^{2}（nT）个字的内存。

值得注意的是，MonocarpicExpert 的内存和遗憾保证即使在自适应对手下也成立。MonocarpicExpert 的伪代码以算法 10 的形式给出。专家集合被划分为 L 个子集 $\mathcal{U}_1 \cup \cdots \cup \mathcal{U}_L$（$L =$对数$_2$（$T$））， 其中 $\mathcal{U}_\ell$包含存活时间至多为 2^ℓ 天的专家。子集 $\mathcal{U}_\ell$ 每 $2^{\ell-1}$ 天更新一次（从 $\mathcal{U}_{\ell-1}$ 接收专家或向 $\mathcal{U}_{\ell+1}$移交专家）， 并移除不活跃的专家。在两次更新之间， $\mathcal{U}_\ell$ 的成员关系保持不变。MonocarpicExpert 严重依赖区间遗憾子过程。它在 L 个专家 Exp$_\ell$ （$\ell \in [L]$）上运行区间遗憾，其中 Exp$_\ell$ 自身每 $2^{\ell-1}$ 天在区间遗憾上运行一次（两次更新之间，此时 $\mathcal{U}_\ell$ 的成员关系保持不变）。如果 $\mathcal{U}_\ell$中的某个专家在两次更新之间变得不活跃（即在开始时存活，但在中途退出），则我们会在下一次更新之前为其分配单位损失。

算法 10 MonocarpicExpert

初始化 $\mathcal{U}_\ell \leftarrow \emptyset$、Exp$_\ell$ ($\ell \in [L]$)　▷ $L =$ 对数$_2$(T)
Exp ←区间遗憾(Exp$_1$、.....、Exp$_L$， T)
对于 $t = 1, 2, \cdots, T$ **do**
　将新激活的专家添加到 $\mathcal{U}_1$
　遵循 Exp 的决策
　对于 $\ell = 1, 2, \ldots,$ pw(t) **执行**　▷ 更新成员关系
　　$\mathcal{U}_{\ell+1} \leftarrow \mathcal{U}_{\ell+1} \cup \mathcal{U}_\ell, \mathcal{U}_\ell \leftarrow \emptyset$
　　移除 $\mathcal{U}_{\ell+1}$ 中不活跃的专家
　结束循环
结束循环

过程EXP$_\ell$　▷ $\ell \in [L]$
　对于 $s = 1$、2、.....、$T/2^{\ell-1}$ **执行**　▷ s 次重启
　　区间遗憾($\mathcal{U}_\ell$， $2^{\ell-1}$)
　结束循环
结束过程

遗　　憾和内存保证主要来自区间遗憾子程序　　.

引理 4.11. 设 $T \geq 1$。对于任意专家 $i \in \mathcal{U}$ 和时间区间 $\mathcal{I} \subseteq [T]$，算法 区间-遗憾 保证以至少 $1 - 1/($ $nT)^{\omega(1)}$ 的概率，

$$\sum_{t \in \mathcal{I}} \ell_t(i_t) - \sum_{t \in \mathcal{I}} \ell_t(i) \leq O\left(\sqrt{|\mathcal{I}| \log(nT)}\right)$$

在面对自适应对手时仍然成立。此外，区间遗憾最多使用 $O(|\mathcal{U}| \log(nT))$ 个字的内存。

区间遗憾维护一组元专家 SingleInterval$_{a,b}$ ($a \in [L], b \in [T/2^a]$)。SingleInterval$_{a,b}$ 在时间区间 $[2^a(b-1)+1 : 2^a b]$内有效。它在 $[2^a(b-1)+1 : 2^a b]$ 期间对 $\mathcal{U}$ 运行乘法权重更新（以均匀权重初始化），并在区间之外不执行任何操作。令 $h(t,a,b)$ 表示 SingleInterval$_{a,b}$的有效性；也就是说，当 $t \in [2^a(b-1)+1 : 2^a b]$ 时，它为 $h(t,a,b) = 1$，否则为 $h(t,a,b) = 0$。我们用 $i_{t,a,b}$ 表示 SingleInterval$_{a,b}$ 在第 t 天的动作。区间遗憾维护一组权重 $\{w_{a,b}\}_{a\in[L],b\in[T/2^a]}$，对应于 SingleInterval$_{a,b}$。在第 t 天，有 L 个有效元专家，算法按与 $\{w_{a,b}\}_{h(t,a,b)=1}$成比例的概率采样其中一个，并遵循其建议。损失向量按如下方式构造：对于有效的元专家 SingleInterval$_{a,b}$（即 $h(t,a,b) = 1$），它的损失向量就是$\ell_t(i_{t,a,b})$；对于无效的专家（即 $h(t,a,b) = 0$），则将其设为 ℓ_t，其中 ℓ_t 是由公式（4）计算得到的"期望"损失。它等于区间遗憾在第 t 天获得的期望损失。权重 $w_{a,b}$ 由 Squint 算法 [KVE15]更新。

算法 11 区间遗憾($\mathcal{U}, T$)

在单区间$_{a,b}$ ($a \in [L], b \in [T/2^a]$)上初始化 $w_{a,b} \leftarrow 1$ **对于** $t = 1, 2, \ldots, T$ **执行** 从 $\{w_{a,b}\}_{h(t,a,b)=1}$ ▷ $h(t,a,b)$ 中采样动作 $i_{t,a,b} = 1$ 如果 $t \in [2^a(b-1)+1 : 2^a b]$ 计算期望损失

$$\bar{\ell}_t \leftarrow \sum_{a,b:h(t,a,b)=1} \frac{w_{a,b}}{\sum_{a,b:h(t,a,b)=1} w_{a,b}} \cdot \ell_t(i_{t,a,b}) \tag{4}$$

赋予损失

$$\widehat{\ell}_t(a,b) = \begin{cases} \ell_t(i_{t,a,b}) & h(t,a,b) = 1 \\ \bar{\ell}_t & h(t,a,b) = 0 \end{cases}$$

使用 Squint 更新权重分布

$$w_{a,b} \leftarrow \mathop{\mathbb{E}}_{\eta}\left[\eta \cdot \exp\left(\eta \sum_{\tau=1}^{t-1} v_\tau(a,b) - \eta^2 \sum_{\tau=1}^{t-1} v_\tau^2(a,b)\right)\right] \tag{5}$$

其中 $v_\tau(a,b) = \bar{\ell}_\tau - \widehat{\ell}_\tau(a,b)$ **结束循环** **过程**单区间$_{a,b}$ **对于** $t = 2^a(b-1)+1, \ldots, 2^a b$ **执行** 在 上运行乘法权重更新 $\mathcal{U}$。 **结束循环** **结束过程**

证明。对于任意 $a \in [L]$，$b \in [T/2^a]$ 以及专家 $i \in \mathcal{U}$，根据乘法权重更新的遗憾保证，以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率，有

$$\sum_{t=2^a(b-1)+1}^{2^a b} \ell_t(i_{t,a,b}) - \sum_{t=2^a(b-1)+1}^{2^a b} \ell_t(i) \le O\left(\sqrt{2^a \log(nT)}\right). \tag{6}$$

回想一下，ℓ_t 是第 t天公式（4）的解，它等于区间-

遗憾所获得的期望损失，因为

$$\bar{\ell}_t = \sum_{a,b:h(t,a,b)=1} \frac{w_{t,a,b}}{\sum_{a,b:h(t,a,b)=1} w_{t,a,b}} \ell(i_{t,a,b}) = \sum_{a,b:h(t,a,b)=1} p_{t,a,b}\ell(i_{t,a,b})$$

其中 $p_{t,\ a,\ b} = \frac{w_t}{\sum_{a\ b\ h\ t\ a\ b}}$ 是遵循 $i_{t,\ a,\ b}$ 的概率。

根据 Squint 的遗憾保证（见引理 3.3），可得

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t\in[2^a(b-1)+1:2^ab]} \ell_t(i_t) - \sum_{t\in[2^a(b-1)+1:2^ab]} \ell_t(i_{t,a,b})\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(i_t) - \sum_{t=1}^{T} \widehat{\ell}_t(a,b)\right] \\
&\le O\left(\sqrt{\left(\sum_{t\in[T]} (\mathbb{E}[\ell_t(i_t)] - \widehat{\ell}_t(a,b))^2\right) \cdot \log(nT)}\right) = O\left(\sqrt{2^a \log(nT)}\right). \qquad (7)
\end{aligned}$$

$\widehat{\ }$ $\widehat{\ }$第一步和第三步成立，是因为对于任意 $t \in [2^a(b-1)+1:\ 2^ab]$，$\ell_t(a,\ b) = \ell_t(i_{t,\ a,\ b})$，且损失 $\ell_t(a,\ b) = \ell_t = \mathbb{E}[\ell_t(i_t)]$ 对任意 $t \notin [2^a(b-1)+1:\ 2^ab]$ 均成立；第二步则根据 Squint 的遗憾保证得出。

应用 Azuma-Hoeffding 不等式于方程 (7)，并注意到 $|\mathbb{E}[\ell_t(i_t)] - \ell_t(i_t)| \le 2$，可得

$$\sum_{t\in[2^a(b-1)+1:2^ab]} \ell_t(i_t) - \sum_{t\in[2^a(b-1)+1:2^ab]} \ell_t(i_{t,a,b}) \le O\left(\sqrt{2^a \log(nT)}\right) \qquad (8)$$

以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率成立。

合并方程 (6) 和方程 (8) 合并可得，以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率，

$$\sum_{t\in[2^a(b-1)+1:2^ab]} \ell_t(i_t) - \sum_{t\in[2^a(b-1)+1:2^ab]} \ell_t(i) \le O\left(\sqrt{2^a \log(nT)}\right).$$

为结束遗憾分析，我们注意到，对于任意区间 $\mathcal{I} = [t_1:\ t_2] \subseteq T$，可以将 $\mathcal{I}$ 划分为 $X \le 2\log_2(|\mathcal{I}|)$ 个互不相交的区间 $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2 \cup \cdots \cup \mathcal{I}_X$，使得 (1) $\mathcal{I}_x (x \in [X])$ 恰好覆盖某个元专家单区间 $a_x,\ b_x$ 的生命周期；(2) 至多有两个长度为 2^x 的区间。于是我们得出

$$\begin{aligned}
\sum_{t\in\mathcal{I}} \ell_t(i_t) - \sum_{t\in\mathcal{I}} \ell_t(i) &= \sum_{x=1}^{X} \sum_{t\in\mathcal{I}_x} (\ell_t(i_t) - \ell_t(i)) \le \sum_{x=1}^{X} O\left(\sqrt{|\mathcal{I}_x| \log(nT)}\right) \\
&\le \sum_{x=1}^{\log(|\mathcal{I}|)} O\left(\sqrt{2^x \log(nT)}\right) = O\left(\sqrt{|\mathcal{I}| \log(nT)}\right).
\end{aligned}$$

IntervalRegret 的内存使用。朴素实现需要 $O(T|\mathcal{U}|\log(nT))$，因为需要维护 $O(T\log\ T)$ 个 SingleInterval$_{a,\ b}$ 过程。然而，请注意，在第 t 天，算法只需知道权重 $\{w_{a,b}\}_{h(t,a,b)=1}$，即可确定其动作并计算平均损失 ℓ_t。也就是说，只需知道有效专家的权重 $w_{a,\ b}$。任何时刻都有 $L = \log_2(T)$ 个有效元专家，而 SingleInterval$_{a,\ b}$ 从 $\mathbb{E}_\eta[\eta]$ 开始，在权重为 $w_{a,\ b}$ 的一段连续时期内有效。因此，IntervalRegret 总共需要 $O(|\mathcal{U}|\log(nT))$ 个字的内存。至此，我们结束证明。

□

现在我们可以结束对 MonocarpicExpert 的分析。

定理 *4.10* 的证明。我们首先给出遗憾分析。对于任意在区间 $\mathcal{I}$ 上存活的专家 i。令 $\mathcal{I}' = [t_1 : t_2] \subseteq \mathcal{I}$ 为任意子区间。可以将区间 $\mathcal{I}'$ 划分为 $X \le \log(|\mathcal{I}'|)$ 个连续区间 $\mathcal{I}' = \mathcal{I}'_1 \cup \mathcal{I}'_2 \cup \ldots \cup \mathcal{I}'_X$，使得专家 i 在 $\mathcal{U}_{\ell_x}$ 中存在，持续区间为$\mathcal{I}'_x$ ($\ell_1 < \ell_2 < \cdots < \ell_X$)。此外，$\mathcal{I}'_x$ 的大小呈指数增长，$\mathcal{I}'_1$ 和 $\mathcal{I}'_X$(i 可能不会在 $\mathcal{U}_{\ell_1}$ 和 $\mathcal{U}_{\ell_X}$ 处停留整个更新过程)除外。令 $i_{t,\,\ell}$ 为 Exp_ℓ 在第 t 天采取的动作。则由于 IntervalRegret 遗憾保证，以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率，有

$$\begin{aligned}
\sum_{t\in\mathcal{I}'} \ell_t(i_t) - \sum_{t\in\mathcal{I}'} \ell_t(i) &= \sum_{x=1}^{X} \sum_{t\in\mathcal{I}'_x} \ell_t(i_t) - \ell_t(i) \\
&= \sum_{x=1}^{X} \sum_{t\in\mathcal{I}'_x} \ell_t(i_t) - \ell_t(i_{t,\ell_x}) + \sum_{x=1}^{X} \sum_{t\in\mathcal{I}'_x} \ell_t(i_{t,\ell_x}) - \ell_t(i) \\
&\le \sum_{x=1}^{X} O(\sqrt{|\mathcal{I}'_x| \log(nT)}) \le O(\sqrt{|\mathcal{I}'| \log(nT)}).
\end{aligned}$$

这里第三步由 Exp^{ℓ_x} 的区间保证（见引理 4.11）、Exp^{ℓ_x} 在 $\mathcal{U}_{\ell_x}$ 上的区间遗憾保证，以及当 i 位于 $\mathcal{U}_{\ell_x}$ 中时 Exp^{ℓ_x} 至多重启两次这一事实得出。最后一步则由 $\mathcal{I}'_x$ 的大小递增得出。

最后，对于内存使用，共有 $L = \log_2(T)$ 个过程 Exp_ℓ ($\ell \in [L]$)，而 Exp_ℓ 在 $\mathcal{U}_\ell$ 上运行。只需界定 $\mathcal{U}_\ell$ 的大小。注意，$\mathcal{U}_\ell$ 每隔 $2^{\ell-1}$ 天更新一次，并且在更新时，$\mathcal{U}_\ell$ 中的所有专家都处于活动状态，且该集合在接下来的 $2^{\ell-1}$ 天内保持不变。因此，$|\mathcal{U}_\ell| \le M$，且 Exp_ℓ 的内存使用至多为 $O(M \log(nT))$。总内存至多为 $O(M \log^2(nT))$。 □

4.2.2 完整算法的分析

接下来，我们分析完整算法的内存和遗憾保证。

内存界

内存界主要继承自基线。

引理 4.12. 以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率，对于任意 $r \in [R]$，池 $\mathcal{P}_{r,\cdot}$ 始终不超过 $O(\log^{10}(nT))$。

证明。我们证明，以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率，在任意时刻，$\mathcal{P}_{r,\ell}$ 的大小至多为 $2\log^9(nT)$ ($r \in [R]$, $\ell \in [L]$)，且 $\mathcal{P}_{r,0}$ 至多为 $3\log^9(nT)$。$r = R$ 的情况显然成立，现假设归纳法对线程 $r+1$ 成立。

对于第 r 个线程，$\mathcal{P}_{r,0}$ 初始化为 $\text{Sample}(\mathcal{N}, 1/n)$，或者初始化为 $\text{Sample}(\mathcal{N}, 1/n)$ 与合并 $\mathcal{P}_{r+1,\cdot}$、...、$\mathcal{P}_{R,\cdot}$ 的结果的并集（算法 9 的第 5–9 行）。根据归纳法，对于任意 $r' > r$，我们有 $|\mathcal{P}_{r',\cdot}| = |\mathcal{P}_{r',2}| \le 2\log^9(nT)$（每次重启中，基线 $+(r')$ 仅经历两个轮次）。因此，根据引理 4.4，合并 $\mathcal{P}_{r+1,\cdot}$、...、$\mathcal{P}_{R,\cdot}$ 所得到的池的大小至多为 $2\log^9(nT)$；根据切尔诺夫界，样本集 $\text{Sample}(\mathcal{N}, 1/n)$ 以高概率至多为 $\log^9(nT)$。

而对于 $\ell \geq 1$，根据引理 4.4，合并过程保证子池 $\mathcal{P}_{r,\ell}$ 满足

$$\begin{aligned}|\mathcal{P}_{r,\ell}| \leq & \max\left\{2\log^9(nT), \frac{1}{4}(|\mathcal{P}_{r,\ell}| + |\mathcal{P}_{r,\ell-1}|)\right\} \\ \leq & \max\left\{2\log^9(nT), \frac{1}{4}(2\log^9(nT) + 3\log^9(nT))\right\} = 2\log^9(nT).\end{aligned}$$

□

因此，完整算法的内存使用并不大。

引理 4.13. 以至少 $1-1/(nT)^{\omega(1)}$ 的概率，$\mathcal{P}$的池大小不会超过 $O(\log^{11}(nT))$。此外，算法 *8* 的内存使用上界为 $O(\log^{22}(nT))$ 字。

证明. 由引理 4.12 可知，池 $\mathcal{P}$的大小满足$|\mathcal{P}| = |\mathcal{P}_{1,\cdot} \cup \cdots \cup \mathcal{P}_{R,\cdot}| \leq O(\log^{11}(nT))$。合并过程需要知道 $i, j \in \mathcal{P}$的每个区间 $\Gamma_{i,j}$上的损失，这需要 $O(\log^{22}(nT))$ 字内存。根据定理 4.10，MonocarpicExpert 需要 $O(|\mathcal{P}|\log^2(nT)) = O(\log^{13}(nT))$ 内存。综上，总内存上界为 $O(\log^{22}(nT))$。证明至此结束。

□

遗憾分析

主要的不同之处（也是复杂之处）来自遗憾分析。

符号说明 令 $i^* \in [n]$ 为最佳专家，并令

$$K_1 = [0 : n-1], \quad K_2 = [0 : n-1] \times \{0,1\} \quad \ldots \quad K_R = [0 : n-1] \times \underbrace{\{0,1\} \times \cdots \times \{0,1\}}_{R-1}$$

且 K为 K_1、.....、K_R的并集，即

$$K = K_1 \cup \cdots \cup K_R.$$

给定任意时间步 $a = (a_1$、.....、$a_{r(a)}) \in K$（其中 $r(a)$的定义使得 $a \in K_{r(a)}$），时间步 a 唯一确定基线的一个轮次$+(r(a))$，即它指的是第 $\sum_{r=1}^{r(a)-1} a_r 2^{r(a)-r-1}$ 次重启的第 $a_{r(a)}$ 轮。

定义 4.14（算子 $\oplus$）。对于任意时间步 $a \in K$，我们记

$$a' = (a'_1, a'_2, \ldots a'_{r(a')}) = a \oplus 1 \in K$$

作为满足该条件的唯一时间步

$$\sum_{i=1}^{r(a')} a'_i B_i = \sum_{i=1}^{r(a)} a_i B_i + B_{r(a)} \quad and \quad a'_{r(a')} \neq 0.$$

也就是说，$a' = a \oplus 1$ 是以2为底、末尾截去0 后的下一个数（第一个坐标除外，它属于 $[0 : n-1]$）。

直观地说，如果存在多个不同线程的时期，$a\oplus 1$ 是 a之后的下一个完整轮次，指的是线程编号最低的那个。

令 $K(a)$包含在 $\oplus$ 操作下继 a 之后的所有时间步，即

$$K(a) := \{a\} \cup \{a \oplus 1\} \cup \{(a \oplus 1) \oplus 1\} \cup \cdots \subseteq K$$

粗略地说 ng，$K(a)$ 包含专家在 a 进入后所处的所有时间步/时期。

随机位 同样，在定义活跃/被动专家之前，理解用于池选择的随机位是有帮助的。对于任意时间步 $a \in K$，令随机位为 $\xi_a = (\xi_{a,1}, \ldots, \xi_{a,n})$，其中 $\xi_{a,i} = (\xi_{a,i,1}, \xi_{a,i,2})$ 用于专家 $i (i \in [n])$。第一个坐标 $\xi_{a,i,1} \in \{0,1\}$ 是一个均值为 $1/n$ 的伯努利变量。它用于在时期 a 开始时将新专家采样入池（即算法 9 的第 10 行）。随机位的第二部分 $\xi_{a,i,2} \in \{0,1\}^{R \times 2L \times 2K}$ 用于估计大小以及各子池的合并过程中的筛选，其均值为 $1 \log^4(nT)$。

回顾被动/主动专家的概念。

定义 4.15 (活跃/被动专家)。 在任意时间步 $a \in K$，如果专家 $i \in [n]$ 是被动的，则 $\xi_{a,\,i,\,2} = \vec{0}$。否则，它是一个主动专家。

轮次分配 我们的核算论证将把整个轮次序列划分为一组不相交子序列，并将每个子序列分配给池中不同的一组专家。

固定损失序列 $\ell_1, \ldots, \ell_T$ 以及每个时间步 $a \in K$ 对应的采样且活跃的专家集合 $Y_a \subseteq [n]$。我们的关键观察是

观察 4.16。 假设损失序列 $\{\ell_t\}_{t \in [T]}$ 和采样且活跃的专家集合 $\{Y_a\}_{a \in K}$ 均已固定，那么在算法 8 执行期间的任意时刻，估计大小 s、筛选集 $\mathcal{F}$ 以及存活的活跃专家集合也都是固定的，与采样且被动的专家集合无关。

定义 4.17 (驱逐时间，完整算法)。 对于任意时间步 $a \in K$，考虑进入时间为 a 的专家 i^*，其驱逐时间 $t(a) \in K(a) \cup \{+\infty\}$ 定义为 $K(a)$ 的第一个时间步，使得在 $t(a)$ 结束时，存活的活跃专家集合覆盖 i^*。如果无法覆盖 i^*，则设定 $t(a) = +\infty$。

算法 12 轮次分配（注意：仅用于分析）

初始化 $\mathcal{H}_r \leftarrow \emptyset$ $(r \in [R])$、 $\tau \leftarrow 1$、 $a_1 \leftarrow 0$
当 $\cup_{\tau' \le \tau} \mathcal{I}_{\tau'} \neq [T]$ **时执行**
  如果 $t(a_\tau) = +\infty$ **，则** ▷ i^* 存活至结束
    $\mathcal{H}_{r(a_\tau)} \leftarrow \mathcal{H}_{r(a_\tau)} \cup a_\tau$ ▷ a_τ 是线程 r（a_τ）中的一个不良时期
    如果 $r(a_\tau) = R$ **那么** ▷ 在顶层线程处停止
      $\mathcal{I}_\tau \leftarrow a_\tau$, $a_{\tau+1} \leftarrow a_\tau \oplus 1$, $\tau \leftarrow \tau + 1$,
    else
      $a_\tau \leftarrow (a_\tau,\ 0)$ ▷ 移至下一个线程
    结束如果
  else
    $a_{\tau+1} \leftarrow t(a_\tau) \oplus 1$, $I_\tau \leftarrow [a_\tau :\ t(a_\tau)]$, $\tau \leftarrow \tau + 1$
  结束如果
结束循环

我们分析中的关键要素是轮次分配机制，其伪代码见算法 12（请再次注意，此算法仅用于分析）。算法 12 将整个序列划分开，并通过自底向上的游走确定每个线程 r （$r \in [R]$）的不良时期 $\mathcal{H}_r$。算法从底部线程 $r = 1$ 开始。在每一步 τ，如果专家 i^* 能够存活到末尾，且进入时间为 a_τ，那么它就是线程 r（a_τ）中的一个不良时期，算法会将其加入

线程 r（a_τ）的不良时期 $\mathcal{H}_{r(a_\tau)}$ 集合（第 4 行）。算法 12 不会移至同一线程的下一个轮次，而是转到下一个线程，除非当前已位于最顶层线程 R。如果专家 i^* 能够存活并被覆盖至时间步 t（a_τ）$\in K$（a），那么算法 12 会将第 τ 个区间设为 $\mathcal{I}_\tau = [a_\tau\colon\ t$（$a_\tau$）$]$，并转到下一个轮次 $a_{\tau+1} = t$（a_τ）$\oplus 1$。这里，a_τ 和 t（a_τ）不必位于同一线程，我们用 $\mathcal{I}_\tau = [a_\tau\colon\ t$（$a_\tau$）$]$ 表示从 a_τ 开始到 t（a_τ）结束的时间区间。图 1 展示了一个示例。

令 $\tau_{\max}$ 为算法 12 生成的划分中区间的总数。我们先对这些区间 $\{\mathcal{I}_\tau\}_{\tau\in[\tau_{\max}]}$作几点简单的观察。

引理 4.18. 我们有

- 区间 $\{\mathcal{I}_\tau\}_{\tau\in[\tau_{\max}]}$ 两两不相交，且 $\bigcup_{\tau\in[\tau_{\max}]}\mathcal{I}_\tau = [T]$。
- 设 $L_1 = \{\mathcal{I}_\tau\}_{\tau\in[\tau_{\max}]}$，并设 $L_r := \{\mathcal{I}_\tau\colon |\mathcal{I}_\tau| < B_{r-1}\}(r \in [2\colon R])$ 包含长度小于 B_{r-1}的区间，则

$$\sum_{\mathcal{I}\in L_r}|\mathcal{I}| \le |\mathcal{H}_{r-1}|\cdot B_{r-1}.$$

证明。 </style}第一个结论直接由分配过程得出，下面我们着重证明第二个结论。对于任意线程 $r \in [2:R]$ 以及任意区间 $\mathcal{I}_\tau \in L_r$，我们证明 $\mathcal{I}_\tau$ 包含在线程 $r-1$的某个不良时期中。

回忆一下，$\mathcal{I}_\tau$ 从轮次 a_τ 的开始处起始，并在轮次 $t(a_\tau)$ 的末尾终止（$a_\tau, t(a_\tau)$ 不一定位于同一线程）。首先观察到，$t(a_\tau)$ 所在线程至少为 r；否则，$\mathcal{I}$ 会跨越线程 $r-1$的至少一个轮次。假设时间步 $t(a_\tau)$ 包含在线程 $r-1$ 的轮次 a' 中（即，$r(a') = r-1$），只需证明 (1) $\mathcal{I}$ 在 a' 内开始，且 (2) a'是线程 $r-1$ 的一个不良时期（即，$a' \in \mathcal{H}_{r-1}$）。第一个结论由纪元分配过程得出。对于第二个结论，假设 a' 不是线程 $r-1$的一个不良时期，那么令 a'_1 为 a' 之前最近的线程 $r-1$ 不良时期，令 a'_2 为 a'之后最近的线程 $(r-1)$ 不良时期。算法 12 定义的“游走”在 a'_1 与 a'_2之间不会上升到线程 $r-1$ 以上，这与我们对 $\mathcal{I} \in L_r$的假设矛盾。至此证明完毕。

□

接下来我们证明不良时期的数量以高概率至多为 $\mathcal{H}_r$ $O(n\log^{11}(nT))$.

引理 4.19。 以至少 $1-1/(nT)^{\omega(1)}$ 的概率，$|\mathcal{H}_r| \le O(n\log^{11}(nT))$ 对任意 $r \in [R]$ 均成立。

证明。 回顾一下，我们首先固定了损失序列 $\ell_1,\ldots,\ell_T$，以及采样且活跃的专家集合 $\{Y_a\}_{a\in K}$。对于任意 $r \in [R]$，我们证明池 $\mathcal{P}$ 在末尾的大小满足

$$\Pr\left[|\mathcal{P}| \ge \frac{|\mathcal{H}_r|}{4n} \mid \{Y_a\}_{a\in K}, \{\ell_t\}_{t\in[T]}\right] \ge \frac{1}{2},$$

只要 $|\mathcal{H}_r| \ge \Omega(n)$。这里的概率是针对剩余采样且不活跃的专家的随机性而言的。

证明与引理 4.8 类似。令 W_a 为时间步 a 的采样且被动的专家集合。对于任意时间步 $a \in \mathcal{H}_r$，根据我们的分配机制（算法 12），专家$i^* \notin Y_a$（否则 i^* 会被其自身覆盖。）$i^* \in W_a$ 遵循的概率为

$$\begin{aligned}\Pr\left[i^* \in W_a | \{Y_a\}_{a\in K}, \{\ell_t\}_{t\in[T]}\right] &= \Pr[i^* \in W_a | i^* \notin Y_a] \\ &\ge \Pr[i^* \in W_a] = \frac{1}{n}\cdot\left(1-\frac{1}{\log^4(n)}\right)^{R\times 2L\times 2K} \ge \frac{1}{2n}.\end{aligned}$$

关键观察是，一旦 $i^* \in W_a$，即 i^* 在时间步 t 被采样且处于被动状态，它就会一直存活到最后。这是因为筛选集 $\mathcal{F}$、估计大小 s 以及存活的活跃专家集合，都完全由采样且活跃的专家 $\{Y_a\}_{a\in K}$ 和损失序列$\{\ell_t\}_{t\in[T]}$决定，而后者在定义驱逐时间 t（a）时就已经固定。因此，如果被动专家 i^* 在时间步 a 进入，它就会一直存活到最后。

$i^* \in W_a$ 发生的事件彼此独立，对于 $a \in \mathcal{H}_r$（以损失序列和 $\{Y_a\}_{a\in[K]}$为条件），因此由切尔诺夫界可得

$$\begin{aligned}\Pr\left[|\mathcal{P}| \le \frac{|\mathcal{H}_r|}{4n} \mid \{Y_a\}_{a\in K}, \{\ell_t\}_{t\in[T]}\right] &\le \Pr\left[\sum_{a\in\mathcal{H}_r} 1\{i^* \in W_a\} \le \frac{|\mathcal{H}_r|}{4n} \mid \{Y_a\}_{a\in K}, \{\ell_t\}_{t\in[T]}\right] \\ &\le \exp(-|\mathcal{H}_r|/16n).\end{aligned}$$

注意，我们已在引理 4.13 中证明，$|\mathcal{P}| \ge \Omega(\log^{11}(nT))$发生的概率至多为 $1/(nT)^{\omega(1)}$，这意味着

$$\Pr[|\mathcal{H}_r| \ge \Omega(n\log^{11}(nT))] \le 1/(nT)^{\omega(1)}.$$

至此，我们完成证明。 □

现在我们应用 MonocarpicExpert 的遗憾保证，并证明算法 8 在每个区间 $\mathcal{I}_\tau$上都能实现较低的遗憾。

引理 4.20. 以至少 $1-1/(nT)^{\omega(1)}$的概率，对于任意 $\tau \in [\tau_{\max}]$ 和 $a_\tau \notin \mathcal{H}_R$，

$$\sum_{t\in\mathcal{I}_\tau} \ell_t(i_t) - \sum_{t\in\mathcal{I}_\tau} \ell_t(i^*) \le O\left(\sqrt{|\mathcal{I}_\tau|}\cdot\log^6(nT)\right).$$

证明. 我们以引理 4.12 中的事件为条件。给定任意一个以 a_τ开始、以 $t(a_\tau)$结束且满足 $a_\tau \notin \mathcal{H}_R$ 的区间 I_τ，进入时间为 a_τ的专家 i^*在 $t(a_\tau)$结束时被 $\mathcal{P}$覆盖。令 $i^*_1,\dots,i^*_s$为覆盖 i^*的专家集合。根据定义 4.2，我们可以将区间$\mathcal{I}_\tau = \mathcal{I}_{\tau,1}\cup\cdots\cup\mathcal{I}_{\tau,s}$划分为若干部分，并得到

$$\sum_{t\in\mathcal{I}_\tau} \ell_t(i^*) \ge \sum_{j=1}^{s}\sum_{t\in\mathcal{I}_{\tau,j}} \ell_t(i^*_j) - 1 \tag{9}$$

根据遗憾保证 MonocarpicExpert，以至少 $1-1/(nT)^{\omega(1)}$ 的概率，可以得出 s

$$\begin{aligned}\sum_{t\in\mathcal{I}_\tau} \ell_t(i_t) - \sum_{t\in\mathcal{I}_\tau} \ell_t(i^*) &= \sum_{t\in\mathcal{I}_\tau} \ell_t(i_t) - \sum_{j=1}^{s}\sum_{t\in\mathcal{I}_{\tau,j}} \ell_t(i^*_j) + \sum_{j=1}^{s}\sum_{t\in\mathcal{I}_{\tau,j}} \ell_t(i^*_j) - \sum_{t\in\mathcal{I}_\tau} \ell_t(i^*) \\ &\le \sum_{j=1}^{s} O\left(\sqrt{|\mathcal{I}_{\tau,j}|\log(nT)}\right) + \sum_{j=1}^{s}\sum_{t\in\mathcal{I}_{\tau,j}} \ell_t(i^*_j) - \sum_{t\in\mathcal{I}_\tau} \ell_t(i^*) \\ &\le \sum_{j=1}^{s} O\left(\sqrt{|\mathcal{I}_{\tau,j}|\log(nT)}\right) + 1 \\ &\le \sqrt{|\mathcal{I}_\tau|}\cdot\log^6(nT).\end{aligned}$$

第二步由定理 4.10 得出，第三步由方程 (9) 得出，最后一步由 $s \le \log^{11}(nT)$（引理 4.12）、$\mathcal{I}_\tau = \mathcal{I}_{\tau,1}\cup\cdots\cup\mathcal{I}_{\tau,s}$ 以及柯西-施瓦茨不等式得出。至此，我们完成证明。

□

现在我们可以对完整算法的遗憾进行界定。

引理 4.21。 以至少 $1-1/(nT)^{\omega(1)}$ 的概率，有

$$\sum_{t\in[T]}\ell_t(i_t)-\sum_{t\in[T]}\ell_t(i^*)\le\widetilde{O}\left(\sqrt{nT}\right).$$

证明。 在证明中，我们首先固定损失序列 $\{\ell_t\}_{t\in[T]}$ 和采样且活跃的专家集合$\{Y_a\}_{a\in K}$。现在，我们可以使用轮次分配机制（算法 12），将整个序列 $[T]$ 划分为一组区间 $\{I_\tau\}_{\tau\in[\tau_{\max}]}$。

我们首先根据区间长度对其分类。以至少 $1-1/(nT)^{\omega(1)}$ 的概率，有

$$\begin{aligned}\sum_{t\in[T]}\ell_t(i_t)-\sum_{t\in[T]}\ell_t(i^*)&=\sum_{r=1}^{R-1}\sum_{\mathcal{I}\in L_r\setminus L_{r+1}}\sum_{t\in\mathcal{I}}(\ell_t(i_t)-\ell_t(i^*))+\sum_{\mathcal{I}\in L_R}\sum_{t\in\mathcal{I}}(\ell_t(i_t)-\ell_t(i^*))\\&\le\sum_{r=1}^{R-1}\sum_{\mathcal{I}\in L_r\setminus L_{r+1}}\sum_{t\in\mathcal{I}}(\ell_t(i_t)-\ell_t(i^*))+O(|\mathcal{H}_{R-1}|)\\&\le\sum_{r=1}^{R-1}\sum_{\mathcal{I}\in L_r\setminus L_{r+1}}O\left(\sqrt{|\mathcal{I}|}\log^6(nT)\right)+O(|\mathcal{H}_{R-1}|).\end{aligned}\tag{10}$$

第一步由 $[T]=\cup_{\mathcal{I}\in L_1}\mathcal{I}$ 得出，第二步使用了以下朴素界：

$$\sum_{\mathcal{I}\in L_R}\sum_{t\in\mathcal{I}}(\ell_t(i_t)-\ell_t(i^*))\le\sum_{\mathcal{I}\in L_R}|\mathcal{I}|\le|\mathcal{H}_{R-1}|\cdot B_{R-1}=O(|\mathcal{H}_{R-1}|),$$

最后一步由引理 4.20 的区间遗憾界得出。

We 分别界定每一组的遗憾。注意，对于任意 $r\in[R-1]$，有

$$\begin{aligned}\sum_{\mathcal{I}\in L_r\setminus L_{r+1}}\sqrt{|\mathcal{I}|}&\le\sqrt{|L_r\setminus L_{r+1}|\cdot\sum_{\mathcal{I}\in L_r\setminus L_{r+1}}|\mathcal{I}|}\\&\le|\mathcal{H}_{r-1}|B_{r-1}\cdot\sqrt{\frac{1}{B_r}}\\&\le\begin{cases}|\mathcal{H}_{r-1}|\cdot\sqrt{2T/n} & r\ge2\\ \sqrt{nT} & r=1\end{cases}.\end{aligned}\tag{11}$$

第一步由柯西-施瓦茨不等式得出，第二步由以下两点得出：(1) 区间$\mathcal{I}\in L_r\setminus L_{r+1}$ 的长度满足 $|\mathcal{I}|\in[B_r:\ B_{r-1})$；(2) $\sum_{\mathcal{I}\in L_r\setminus L_{r+1}}|\mathcal{I}|\le|\mathcal{H}_{r-1}|\cdot B_{r-1}$ （见引理 4.18）。最后一步由 $\{B_r\}_{r\in[R]}$ 的选取得出。注意，我们稍微滥用了符号说明，并设 $|\mathcal{H}_0|=1$和 $B_0=T$。

利用方程 (11)，我们可以对方程 (10) 的右端项进行界定。以至少 $1-1/(nT)^{\omega(1)}$ 的概率，有

$$\begin{aligned}&\sum_{r=1}^{R-1}\sum_{\mathcal{I}\in L_r\setminus L_{r+1}}O\left(\sqrt{|\mathcal{I}|}\log^6(nT)\right)+O(|\mathcal{H}_{R-1}|)\\&\le O\left(\sqrt{nT}\log^6(nT)+\sum_{r=2}^{R-1}|\mathcal{H}_{r-1}|\cdot\sqrt{T/n}\log^6(nT)+|\mathcal{H}_{R-1}|\right)\\&\le\widetilde{O}\left(\sqrt{T/n}\right),\end{aligned}\tag{12}$$

~其中第二步由引理 4.19 得出，即对于任意 $r \in [R]$, $|\mathcal{H}_r| \le O(n)$ 以高概率成立。

结合方程 (10) 和方程 (12)，我们得出

$$\sum_{t\in[T]} \ell_t(i_t) - \sum_{t\in[T]} \ell_t(i^*) \le \widetilde{O}\left(\sqrt{nT}\right)$$

以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率成立。至此，证明完成。 □

结合遗憾分析（引理 4.21）和内存界（引理 4.13），我们完成命题 4.9 的证明。通过分组技巧，我们完成定理 1.1 的证明。

~~定理 *1.1* 的证明。我们将专家 $[n]$ 划分为 $G = O(S)$ 组，每组有 $n' = O(n/S)$ 个专家。我们在每组 $j \in [G]$ 上实例化算法 8 的一个元线程 [9]，并将每个元线程视为一个独立的元专家。最终决策由对元专家集合 $[G]$运行乘法权重更新得出。设最优专家 $i^* \in [n]$ 属于第 j^* 组（$j^* \in [G]$），则遗憾保证由

$$\begin{aligned}\sum_{t=1}^{T} \ell_t(i_t) - \sum_{t=1}^{T} \ell_t(i^*) &= \sum_{t=1}^{T} \ell_t(i_t) - \sum_{t=1}^{T} \ell_t(j^*) + \sum_{t=1}^{T} \ell_t(j^*) - \sum_{t=1}^{T} \ell_t(i^*) \\ &\le \widetilde{O}(\sqrt{T}) + \widetilde{O}(\sqrt{n'T}) = \widetilde{O}\left(\sqrt{\frac{nT}{S}}\right),\end{aligned}$$

其中第二步由乘法权重更新的遗憾保证和命题 4.9 得出。为了界定内存，注意该算法维护着 G 个算法 8 的线程，每个线程占用多重对数(nT)空间，因此使用的总空间至多为 $G \cdot$ 多重对数$(nT) \le S$。至此，证明完毕。

□

5 对抗自适应对手时的趋零遗憾

我们给出了针对自适应对手的遗憾保证。

定理 1.2 (算法，自适应对手)。 设 n, T 为足够大的整数。存在一种在线学习算法，使用的空间至多为 S，并且以至少 $1 - 1/(nT)^{\omega(1)}$的概率，对自适应对手实现 $\widetilde{O}\left(\text{最大}\left\{\sqrt{\frac{nT}{S}}, \frac{\sqrt{nT}}{S}\right\}\right)$ 遗憾。

我们聚焦于 $T \ge S$ 的情形，并给出一种算法，当 $T \ge S$ 时可实现总遗憾 $\widetilde{O}\left(\frac{\sqrt{nT}}{S}\right)$。这意味着，在~~$T$ 的整个范围内，该算法都满足定理 1.2 的遗憾保证；其证明采用与定理 1.1 相同的分组技巧。具体而言，我们证明

命题 5.1。 设 $n, T \ge 1, \epsilon \in (1/\sqrt{n}, 1)$ 和 $S = \sqrt{n}/\epsilon$。假设 $T \ge S$，则存在一种在线学习算法，使用至多 $S \cdot$ 多重对数(nT) 空间，并能在自适应对手面前实现 $\epsilon T \cdot$ 多重对数(nT) 遗憾。

[9]这里我们使用元线程 来与算法 8 内部的 R 线程 加以区分。

主算法的伪代码见算法 13。它维护两个组成部分：RandomExpert 和 LongExpert，并在它们之上运行区间遗憾。

令 $B = 1/\epsilon^2$ 为时期大小。对于每个轮次，RandomExpert（算法 14）会运行 IntervalRegret，调用 $R = \lceil \log_4(1/\epsilon^2) \rceil = \lceil \log_2(1/\epsilon) \rceil$ 个过程 Rexp$_r$ ($r \in [R]$)。对于任意 $r \in [R]$，Rexp$_r$ 每隔 $1/\epsilon_r^2$ 天重启一次，并在每次重启期间，运行乘法权重更新，对 $N_r = \epsilon_r^2 \sqrt{n} \log^2(nT)/\epsilon$ 个随机选出的专家进行处理。我们采用一个常见技巧，使其能够应对自适应对手：Rexp$_r$ 维护 $1/\epsilon_r^2$ 个不同的专家池，而第 b 天的决策来自第 b 个池（$b \in [1/\epsilon_r^2]$）。这将自适应对手转化为非自适应对手。从高层次来看，Rexp$_r$ 能够保证在每次重启期间，相对于除排名前 $\epsilon\sqrt{n}/\epsilon_r^2$ 名以外的所有专家，具有 ϵ_r 的竞争性。

LongExpert（算法 14）旨在对顶尖专家进行对冲。它维护一个池 $\mathcal{P}$，其中包含在过去某些轮次中表现良好的专家，并在 $\mathcal{P}$ 上运行 MonocarpicExpert。每个轮次中，LongExpert 都会调用 MaintainPool$_r$ ($r \in [R]$)。对于每个 $r \in [R]$，MaintainPool$_r$ 每隔 $1/\epsilon_r^2$ 天重启一次。每次重启时，它会采样一组专家 O_r ($|O_r| = O(\sqrt{n}/\epsilon)$)，并观察他们在 $1/\epsilon_r^2$ 天内的损失。如果专家 $i \in O_r$ 的损失显著小于 Rexp$_r$，就称其存活，并将其加入 $\mathcal{P}_r$。每个轮次结束时，池 $\mathcal{P}_r$ 会移除存活了 $\epsilon\sqrt{n}$ 个轮次的专家。

算法 13 针对自适应对手的算法

在 RandomExpert 和 LongExpert 上运行 IntervalRegret

算法 14 RandomExpert

初始化 Rexp$_r$, $\epsilon_r \leftarrow 2^{r-1}\epsilon$, $N_r \leftarrow \epsilon_r^2\sqrt{n}\log^2(nT)/\epsilon$ ($\forall\, r \in [R]$) ▷ $R = \lceil$对数$_2(1/\epsilon)\rceil$

针对 $t = 1, 2, \ldots, \epsilon^2 T$ **执行** ▷ 轮次 t

　　每隔 $1/\epsilon^2$ 天在 Rexp$_r$ ($r \in [R]$) 上运行 IntervalRegret。

结束循环

过程 Rexp$r$ ▷ $r \in [R]$

　　遍历 $s = 1$、2、.....、ϵ_r^2/ϵ^2 **执行** ▷ 每隔 $1/\epsilon_r^2$ 天重启

　　　　采样 $N_r \times 1/\epsilon_r^2$ 位专家 $\{i_{\alpha,\beta}\}_{\alpha\in[N_r],\ \beta\in[1/\epsilon_r^2]}$

　　　　遍历 $b = 1$、2、.....、$1/\epsilon_r^2$ **do**

　　　　　　使用乘法权重更新，在 $\{i_{\alpha,\beta}\}_{\alpha\in[N_r]}$ ($\forall \beta \in [1/\epsilon_r^2]$) 上维护权重

　　　　　　选择 $i_{\alpha(b),\ b}$ ▷ 第 b 个池的决策

　　　　结束循环

　　结束循环

结束过程

符号说明 对于任意轮次 $t \in [\epsilon^2 T]$、线程 $r \in [R]$、重启 $s \in [\epsilon_r^2/\epsilon^2]$ 和天 $\beta \in [1/\epsilon_r^2]$，令 $i_{t,r,s,\beta}$ 为 Rexp$_r$ 所采取的动作，令 $\ell_{t,r,s,\beta}$ 为线程 r 的第 t 轮次中第 s 次重启的第 β 天的损失向量。对于任意 $k \in [n]$，令 $i^*_{t,r,s,k} \in [n]$ 为线程 r 的第 t 轮次中第 s 次重启期间排名第 k 的专家；若出现并列，则任意且一致地打破并列。

RandomExpert 的遗憾保证

我们首先给出 RandomExpert 的遗憾保证，并旨在获得一个高概率界。

算法 15 LONGEXPERT

运行 MaintainPool$_r$ ($r \in [R]$)
在 $\mathcal{P}$上运行 MonocarpicExpert ▷ $\mathcal{P} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_R$
过程MaintainPool$_r$ ▷ $r \in [R]$
  $\mathcal{P}_r \leftarrow \emptyset$
  对于 $t = 1, 2, \ldots, \epsilon^2 T$ **执行** ▷ 轮次 t
    对于 $s = 1$、2、.....、ϵ_r^2/ϵ^2 **do** ▷ 重启 s
      $O_r \leftarrow$ 采样($\mathcal{N}$, $\frac{1}{\epsilon\sqrt{n}}$) ▷ 采样 $\Theta(\sqrt{n}/\epsilon)$ 专家
      观察 O_r for $1/\epsilon_r^2$ 天
      for 每位专家 $i \in O_r$ **执行**
        $\mathcal{P}_r \leftarrow \mathcal{P}_r \cup \{i\}$ 如果 $\mathcal{L}_s(i) < \mathcal{L}_s(\text{Rexp}_r) - 2\log(nT)/\epsilon_r$.
      结束 for
    结束 for
    在 $\mathcal{P}_r$ 中移除过期专家 ▷ 一位专家在 $\mathcal{P}$ 中存活 $\epsilon\sqrt{n}$ 轮次
  结束循环
结束过程

引理 5.2。 以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率，对于每个线程 $r \in [R]$、每个轮次 $t \in [\epsilon^2 T]$ 和重启 $s \in [\epsilon_r^2/\epsilon^2]$，均有

$$\sum_{\beta=1}^{1/\epsilon_r^2} \ell_{t,r,s,\beta}(i_{t,r,s,\beta}) - \sum_{\beta=1}^{1/\epsilon_r^2} \ell_{t,r,s,\beta}(i^*_{t,r,s,k}) \le \epsilon_r^{-1} \log(nT) \tag{13}$$

对于任意 $k \ge \epsilon\sqrt{n/\epsilon_r^2}$，面对自适应对手时均成立。

证明。 对于一个轮次 $t \in [\epsilon^2 T]$、一个线程 $r \in [R]$ 和一次重启 $s \in [\epsilon_r^2/\epsilon^2]$，令 $B_r = 1/\epsilon_r^2$。在不会造成混淆的情况下，简化符号说明并省略 t, r, s 上的下标会更方便。

这一高层思路已出现在先前的工作中，例如 [CBL06、GHM19]。通过在不同天从不同副本中选取决策，该决策满足

$$\Pr[i_\beta = i | i_1, \ldots, i_{\beta-1}, \ell_1, \ldots, \ell_{\beta-1}] = \Pr[i_\beta = i | \ell_1, \ldots, \ell_{\beta-1}]. \tag{14}$$

也就是说， i_β 对先前行动的依赖，可以通过基于$\ell_1, \ldots, \ell_{\beta-1}$的依赖来解释。这直观地意味着，自适应性（基于先前行动自适应选择的损失序列）并无助益。令方程 (14) 的分布为 $P(\ell_1, \ldots, \ell_{\beta-1})$，在不引起混淆时将其简写为$p_\beta$ 。

步骤 1。归约为期望遗憾。 令

$$X_\beta = \ell_\beta(i_\beta) - \mathop{\mathbb{E}}_{i_\beta \sim p_\beta}[\ell_\beta(i_\beta)] \quad \text{and} \quad Y_\beta = X_1 + \cdots + X_\beta, \forall \beta \in [B]$$

$\{Y_\beta\}_{\beta \in [B]}$ 相对于 $\{\ell_\beta\}_{\beta \in [B]}$ 和 $\{i_\beta\}_{\beta \in [B]}$的随机性构成一个鞅，且其有界差分至多为 2（即 $|X_\beta| \le 2$）。根据阿祖马-霍夫丁界，可得

$$\Pr\left[|Y_B| \ge \frac{1}{2}\sqrt{B}\log(nT)\right] \le 2\exp(-B\log^2(nT)/32B) \le 1/(nT)^{\omega(1)}.$$

因此，以至少 $1-1/(nT)^{\omega(1)}$ 的概率，在选择 i_t时，有

$$\sum_{\beta=1}^{B}\left(\ell_\beta(i_\beta)-\mathop{\mathbb{E}}_{i_\beta\sim p_\beta}[\ell_t(i_\beta)]\right)\le\frac{1}{2}\sqrt{B}\log(nT). \tag{15}$$

步骤 2. 归约到非自适应对手。 接下来证明，只需考虑相对于非自适应对手的遗憾。回顾一下，i_k^* 是排名第 k 的专家。在步骤 1 的高概率事件成立的条件下，有

$$\begin{aligned}
&\sum_{\beta=1}^{B}(\ell_\beta(i_\beta)-\ell_\beta(i_k^*))\\
\le&\sum_{\beta=1}^{B}\left(\mathop{\mathbb{E}}_{i_\beta\sim p_\beta}[\ell_\beta(i_\beta)]-\ell_\beta(i_k^*)+\frac{1}{2}\sqrt{B}\log(nT)\right)\\
\le&\sup_{\ell_1}\mathop{\mathbb{E}}_{i_1\sim p_1}\cdots\sup_{\ell_{B-1}}\mathop{\mathbb{E}}_{i_{B-1}\sim p_{B-1}}\sup_{\ell_B}\mathop{\mathbb{E}}_{i_B\sim p_B}\left[\sum_{\beta=1}^{B}\ell_\beta(i_\beta)-\sum_{\beta=1}^{B}\ell_\beta(i_k^*)\right]+\frac{1}{2}\sqrt{B}\log(nT)\\
=&\sup_{\ell_1}\mathop{\mathbb{E}}_{i_1\sim p_1}\cdots\sup_{\ell_{B-1}}\sup_{\ell_B}\mathop{\mathbb{E}}_{i_{B-1}\sim p_{B-1}}\mathop{\mathbb{E}}_{i_B\sim p_B}\left[\sum_{\beta=1}^{B}\ell_\beta(i_\beta)-\sum_{\beta=1}^{B}\ell_\beta(i_k^*)\right]+\frac{1}{2}\sqrt{B}\log(nT)\\
&\vdots\\
=&\sup_{\ell_1}\cdots\sup_{\ell_B}\mathop{\mathbb{E}}_{i_1\sim p_1}\cdots\mathop{\mathbb{E}}_{i_B\sim p_B}\left[\sum_{\beta=1}^{B}\ell_\beta(i_\beta)-\sum_{\beta=1}^{B}\ell_\beta(i_k^*)\right]+\frac{1}{2}\sqrt{B}\log(nT).
\end{aligned}\tag{16}$$

第一步由方程 (15) 得出。在第三步中，我们可以交换期望与上确界，因为在以 $\ell_1,\ldots,\ell_{\beta-1}$ 为条件时，i_β不依赖于 $i_1,\ldots,i_{\beta-1}$。

步骤 3. 归约到单个副本 现在固定一个损失序列 $\ell_1,\ldots,\ell_B$，我们来界定期望遗憾。考虑一种只维护一份专家副本的算法：它在开始时采样 $N_r=\epsilon_r^2\sqrt{n}\log^2(nT)/\epsilon$ 个专家，并在其上运行乘法权重更新 $B_r=1/\epsilon_r^2$ 天。令 $\widehat{i}_\beta$ 表示该算法（单个副本）在第 β 天采取的动作，不难看出

$$\Pr\left[\widehat{i}_\beta=i|\ell_1,\ldots\ell_{\beta-1}\right]=\Pr[i_\beta=i|\ell_1,\ldots\ell_{\beta-1}].$$

因此，固定损失序列后，只需对其进行界定

$$\mathsf{reg}=\mathop{\mathbb{E}}_{i_t\sim p_1}\cdots\mathop{\mathbb{E}}_{i_B\sim p_B}\left[\sum_{\beta=1}^{B}\ell_t(i_t)-\sum_{\beta=1}^{B}\ell_\beta(i_k^*)\right]=\mathbb{E}\left[\sum_{\beta=1}^{B}\ell_\beta(\widehat{i}_\beta)-\sum_{\beta=1}^{B}\ell_\beta(i_k^*)\right]. \tag{17}$$

步骤 4. 界定遗憾。最后，我们界定遗憾（即方程 (17) 的右端项）。注意，损失序列是固定的，因此专家 i_1^*、.....、i_n^* 也是固定的。单副本算法在开始时采样 N_r 个专家，因此，以至少.....的概率

$$1-\left(1-\frac{\epsilon}{\epsilon_r^2\sqrt{n}}\right)^{N_r}=1-\left(1-\frac{\epsilon}{\epsilon_r^2\sqrt{n}}\right)^{\epsilon_r^2\sqrt{n}\log^2(nT)/\epsilon}\ge 1-1/(nT)^{\omega(1)},$$

该算法会采样一个专家 $i \in \{i_1^*, \ldots, i^*_{\epsilon\sqrt{n}/\epsilon_r^2}\}$。以此为条件，对于任意 $k \geq \epsilon\sqrt{n}/\epsilon_r^2$，有

$$\mathbb{E}\left[\sum_{\beta=1}^{B}\ell_\beta(\widehat{i}_\beta) - \sum_{\beta=1}^{B}\ell_\beta(i_k^*)\right] = \mathbb{E}\left[\sum_{\beta=1}^{1/\epsilon_r^2}\ell_\beta(\widehat{i}_\beta) - \sum_{\beta=1}^{1/\epsilon_r^2}\ell_\beta(i_k^*)\right]$$
$$\leq \mathbb{E}\left[\sum_{\beta=1}^{1/\epsilon_r^2}\ell_\beta(\widehat{i}_\beta) - \sum_{\beta=1}^{1/\epsilon_r^2}\ell_\beta(i)\right] \leq O(\epsilon_r^{-1}\sqrt{\log(nT)}). \tag{18}$$

这里我们在第一步中代入 $B = B_r = 1/\epsilon_r^2$。由于损失至多为 $1/\epsilon_r^2$，可以证明期望遗憾至多为 $O(\epsilon_r^{-1}\sqrt{\log(nT)})$（无需以高概率事件为条件）。

Co结合上述四个步骤，即方程 (15)(16)(17)(18)，我们得出证明 f.

□

内存界

我们首先给出主算法内存使用量的一个上界。

引理 5.3。 以至少 $1 - 1/(nT)^{\omega(1)}$ 的概率，算法 *13* 使用的内存至多为$O(\sqrt{n}\log^2(nT)\log(1/\epsilon)/\epsilon)$ 字。

证明。 我们分别界定 RandomExpert 和 LongExert 的内存使用量。RandomExpert 在 $R = O(\log(1/\epsilon))$ 个线程上运行区间遗憾 Rexp$_r$，其中每个线程 Rexp$_r$ 在 $[n]$ 中采样 $N_r \cdot 1/\epsilon_r^2 = \epsilon_r^2\sqrt{n}\log^2(nT)/\epsilon \cdot 1/\epsilon_r^2 = \sqrt{n}\log^2(nT)/\epsilon$ 个专家，并在每次重启期间维护其权重。因此，RandomExpert 的内存使用量有上界$O(\sqrt{n}\log^2(nT)\log(1/\epsilon)/\epsilon)$。

关键在于界定 LongExpert 过程的内存。对于任意轮次 $t \in [\epsilon^2 T]$、线程 $r \in [R]$ 和重启 $s \in [\epsilon_r^2/\epsilon^2]$，令 $O_{t,r,s}$ 为样本集（算法 15 的第 7 行）。我们以引理 5.2 的高概率事件为条件，并界定第 10 行之后存活的专家数量。根据引理 5.2，任意专家 $i \in [n] \setminus \{i^*_{t,r,s,1}, \ldots, i^*_{t,r,s,\epsilon\sqrt{n}/\epsilon_r^2}\}$ 都满足

$$\sum_{\beta=1}^{1/\epsilon_r^2}\ell_{t,r,s,\beta}(i_{t,r,s,\beta}) - \sum_{\beta=1}^{1/\epsilon_r^2}\ell_{t,s,\beta,r}(i) \leq \epsilon_r^{-1}\log(nT),$$

因此，根据 LongExpert 的驱逐规则，该专家不会存活（见算法 15 的第 10 行）。接下来需要界定交集$|O_{t,r,s} \cap \{i^*_{t,r,s,1}, \ldots, i^*_{t,r,s,\epsilon\sqrt{n}/\epsilon_r^2}\}|$的大小。注意，每个专家都以概率 $\frac{1}{\epsilon\sqrt{n}}$独立采样，即

$$\mathbb{E}\left[\left|O_{t,r,s} \cap \{i^*_{t,r,s,1}, \ldots, i^*_{t,r,s,\epsilon\sqrt{n}/\epsilon_r^2}\}\right|\right] = \epsilon\sqrt{n}/\epsilon_r^2 \cdot \frac{1}{\epsilon\sqrt{n}} = 1/\epsilon_r^2,$$

因此，根据切尔诺夫界，有

$$\Pr\left[|O_{t,r,s} \cap \{i^*_{t,r,s,1}, \ldots, i^*_{t,r,s,\epsilon\sqrt{n}/\epsilon_r^2}\}| \geq \log^2(nT)/\epsilon_r^2\right] \leq \exp(-\log^2(nT)/3\epsilon_r^2) \leq 1/(nT)^{\omega(1)}.$$

对 $s \in [\epsilon_r^2/\epsilon^2]$取并集界后，某一轮次中存活的专家数量至多为 $\log^2(nT)/\epsilon^2$。对 $t \in [\epsilon^2 T]$ 取并集界，并注意到一个专家会在 $\mathcal{P}_r$ 处存活 $\epsilon\sqrt{n}$ 个轮次， $\mathcal{P}_r$ 的大小满足 $|\mathcal{P}_r| \leq \log^2(nT)/\epsilon^2 \cdot \epsilon\sqrt{n} = \sqrt{n}\log^2(nT)/\epsilon$。 $\mathcal{P}$ 的大小满足

$$|\mathcal{P}| = |\mathcal{P}_1 \cup \cdots \mathcal{P}_R| \leq \sqrt{n}\log^2(nT)/\epsilon \cdot R = \sqrt{n}\log^2(nT)\log(1/\epsilon)/\epsilon.$$

□

遗憾分析

接下来，我们分别界定相对于每个专家的遗憾。对于任意一天 $t \in [T]$，令 $i_t^{(L)}$ 表示 LongExpert 在第 t 天采取的动作，令 $i_t^{(R)}$ 表示 RandomExpert 采取的动作。令 $i_t^{(r)}$ 表示 Rexp$_r$ 在第 t天采取的动作。

引理 5.4。 以至少 $1 - 1/(nT)^\omega$ 的概率，对于任意专家 $i \in [n]$，遗憾有如下界

$$\sum_{t=1}^{T} (\ell_t(i_t) - \ell_t(i)) \le O(\epsilon T \log^2(nT)).$$

证明。 我们固定 RandomExpert 的随机性。证明通过将每天 $[T]$ 分配给 RandomExpert 或 LongExpert 来进行。具体而言，我们将整个序列 $[T]$ 划分为如下区间：第一个区间持续到专家 i 首次被加入 $\mathcal{P}$ 时结束；第二个区间在 i 过期并从 $\mathcal{P}$ 中移除时结束；我们重复上述过程，直到序列结束。我们用 E_R 表示奇数编号区间的集合（在奇数编号区间中，$\mathcal{P}$ 不包含 i），用 E_L 表示偶数编号区间的集合（在偶数编号区间中，$\mathcal{P}$ 包含 i）。为方便证明，如果区间 $\mathcal{I} \in E_R$ 跨越多个轮次，我们就按照轮次边界将其拆分（因此 $\mathcal{I}$ 的长度不会超过一个轮次）。对于区间 $\mathcal{I} \in E_R$，假设它包含在轮次 $t(\mathcal{I}) \in [\epsilon^2 T]$ 中，请注意，$\mathcal{I}$ 不一定是一个完整的轮次，因为专家 i 可能在轮次 $t(\mathcal{I})$ 的中途加入 $\mathcal{P}$。

符号说明 我们首先介绍一些定义。

$$K_1 = [\epsilon^2 T], K_2 = [\epsilon^2 T] \times [4], \ldots, K_R = [\epsilon^2 T] \times \underbrace{[4] \times \cdots \times [4]}_{R-1} \quad \text{and} \quad K = K_1 \cup \cdots \cup K_R.$$

对于时间步 $a \in K$，令 $r(a)$ 定义为 $a \in K_{r(a)}$。时间步 a 唯一标识 Rexp$^{r(a)}$ 的一次重启，即第 a_1 轮次中的第 $\sum_{r=2}^{r(a)} a_r 4^{r-2}$ 次重启。令 $\mathcal{L}_a(i) = \sum_{t \in a} \ell_t(i)$ 表示时间步 a 的损失；这里我们稍微滥用了符号，也用 $a \in K$ 表示时间步/重启 a 的整个时段。对于任意区间 $\mathcal{I} \in E_R$，它从轮次 $t(\mathcal{I}) \in [\epsilon^2 T]$的开始处起始，并在某个时间步 $a(\mathcal{I}) \in K$结束。我们进一步根据$a(\mathcal{I})$ 的四进制表示将 $\mathcal{I}$ 划分为： $\mathcal{I} = a_1 \cup \cdots \cup a_{\tau(\mathcal{I})}$。

区间划分 关键步骤是根据专家i和 Rexp$_r$ 的表现来划分区间 $\mathcal{I} \in E_R$。对于任意满足 $\mathcal{I} = a_1 \cup \cdots \cup a_{\tau(\mathcal{I})}$的 $\mathcal{I} \in E_R$，我们对每个子区间 $a_\tau (\tau \in [\tau(\mathcal{I})])$ 调用 AssignInterval(a_τ, i)。请注意，AssignInterval 程序仅用于分析。它返回一组时间步 $M(a_\tau) \subseteq K$，其并集恰好覆盖 a_τ。令 $M(I) = \cup_{\tau \in [\tau(\mathcal{I})]} M(a_\tau)$ 表示这些集合的并集，则有

$$\bigcup_{a \in M(\mathcal{I})} a = \mathcal{I}. \tag{19}$$

AssignInterval 程序（算法 16 中的伪代码）工作如下。它接收一个时间步 $a \in K$ 和一个目标专家 i，并输出一个互不相交的时间步集合 $M(a)$，该集合恰好覆盖 a。AssignInterval 是一个递归过程。它首先通过检查 Rexp$^{r(a)}$ 是否获得良好性能（算法 16 的第 1 行），尝试返回整个区间 a。如果没有，则将时间步 a 分成四个部分 $\{(a, j) \in K_{r(a)+1} : j \in [4]$，并向 Rexp$^{r(a)+1}$寻求帮助，

算法16 AssignInterval(a, i)（注：仅用于分析） ▷ 时间步 a，专家 i

1: **如果** $\mathcal{L}_a(i) \geq \mathcal{L}_a(i^{(r(a))}) - 2\log(nT)/\epsilon_{r(a)}$ **则**2: 返回 a3: **否则**4: 调用 AssignInterval 并返回以下各项的并集：
AssignInterval（(a, j) i），对于 $j \in [4]$5: **结束 if**

通过递归调用 AssignInterval$((a, j)$ $i)$（$j \in [4]$）。注意，该过程保证会终止，因为对于任意 $a \in K_R$，$\mathcal{L}_a(i) \geq 0 > \mathcal{L}_a(i^{(R)}) - 2\log(nT)/\epsilon_R$。

现在我们可以界定遗憾了。首先，我们根据 E_L 和 E_R 将整个序列划分开。以主算法的 IntervalRegret 的随机性为条件，以至少 $1-1/(nT)^{\omega(1)}$ 的概率，有

$$\begin{aligned}
&\sum_{t=1}^{T}(\ell_t(i_t) - \ell_t(i)) \\
=& \sum_{\mathcal{I}\in E_R}\sum_{t\in\mathcal{I}}(\ell_t(i_t) - \ell_t(i)) + \sum_{\mathcal{I}\in E_L}\sum_{t\in\mathcal{I}}(\ell_t(i_t) - \ell_t(i)) \\
\leq& \sum_{\mathcal{I}\in E_R}\sum_{t\in\mathcal{I}}(\ell_t(i_t^{(R)}) - \ell_t(i)) + \sum_{\mathcal{I}\in E_L}\sum_{t\in\mathcal{I}}(\ell_t(i_t^{(L)}) - \ell_t(i)) + \sum_{\mathcal{I}\in E_R\cup E_L} O(\sqrt{|\mathcal{I}|}\log(nT)) \\
\leq& \underbrace{\sum_{\mathcal{I}\in E_R}\sum_{t\in\mathcal{I}}(\ell_t(i_t^{(R)}) - \ell_t(i))}_{(R\text{-regret})} + \underbrace{\sum_{\mathcal{I}\in E_L}\sum_{t\in\mathcal{I}}(\ell_t(i_t^{(L)}) - \ell_t(i))}_{(L\text{-regret})} + O(\epsilon T\log(nT)).
\end{aligned} \tag{20}$$

第二步成立是因为算法 13 在 LongExpert 和 RandomExpert 上运行区间遗憾；最后一步由柯西-施瓦茨不等式得出，并且 $E_R \cup E_L$ 中区间的总数至多为 $\epsilon^2 T$。

我们分别界定（20）右端项中的两个和。（L-遗憾）容易处理，因为在任意区间 $\mathcal{I} \in E_L$ 内，$i \in \mathcal{P}$（根据定义成立），因此，以至少 $1-1/(nT)^{\omega(1)}$ 的概率，

$$\sum_{\mathcal{I}\in E_L}\sum_{t\in\mathcal{I}}(\ell_t(i_t^{(L)}) - \ell_t(i)) \leq O\left(\sum_{\mathcal{I}\in E_L}\sqrt{|\mathcal{I}|}\log(nT)\right) \leq O(\epsilon T\log(nT)). \tag{21}$$

第一步由 MonocarpicExpert 的遗憾保证推出，第二步由柯西-施瓦茨不等式推出。

为了界定 (L-遗憾)，我们根据 $M(\mathcal{I})$ 对其进行拆分。对于每个线程 $r \in [R]$，以及满足存在一个区间 $\mathcal{I} \in E_R$ 使得 $t(I) = t$ 的轮次 $t \in [\epsilon^2 T]$，定义 $A_{t,\ r}$ 为 $M(\mathcal{I})$ 中第 r 层的时间步/重启集合，即，

$$A_{t,r} = \{a : a \in M(\mathcal{I}), a \in K_r\},$$

且 $N_{t,\ r} = |A_{t,\ r}|$。我们设置 $A_{t,\ r} = \emptyset$，并且当轮次 t 包含于 E_L中时，设置 $N_{t,\ r} = 0$。

以至少 $1-1/(nT)^{\omega(1)}$ 的概率，我们有

$$\begin{aligned}\sum_{\mathcal{I}\in E_R}\sum_{t\in\mathcal{I}}(\ell_t(i_t^{(R)})-\ell_t(i)) &= \sum_{\mathcal{I}\in E_R}\sum_{a\in M(\mathcal{I})}\sum_{t\in a}(\ell_a(i_t^{(R)})-\ell_t(i))\\ &= \sum_{\mathcal{I}\in E_R}\sum_{a\in M(\mathcal{I})}\sum_{t\in a}(\ell_t(i_t^{(R)})-\ell_t(i_t^{(r(a))})+\ell_t(i_t^{(r(a))})-\ell_t(i))\\ &\le \sum_{\mathcal{I}\in E_R}\sum_{a\in M(\mathcal{I})}O(\log(nT)/\epsilon_{r(a)})\\ &= \sum_{t=1}^{\epsilon^2T}\sum_{r=1}^{R}\frac{N_{t,r}}{\epsilon_r}\cdot O(\log(nT)) \qquad (22)\end{aligned}$$

第一步成立，因为 $M(\mathcal{I})$ 覆盖 $\mathcal{I}$ （见式（19））。第三步由 RandomExpert 的区间遗憾保证（算法 14 第 3 行）以及 AssignInterval 程序（算法16 第 1 行）得出。

接下来需要界定式（22）的右端项，我们分别界定 $\sum_{t=1}^{\epsilon^2T} N_{t,r}\frac{r}{\epsilon_r}(r\in[2:R])$。

对于任意轮次 $t\in[\epsilon^2T]$ 和线程 $r\in[R]$，定义

$$B_{t,r}=\{a\in K_r : a_1=t, i\notin\mathcal{P}_{r,a} \text{ and } \mathcal{L}_a(i)<\mathcal{L}_a(i^{(r)})-2\log(nT)/\epsilon_r\} \qquad (23)$$

其中 $\mathcal{P}_{r,a}$ 是线程 r 在时间步 a 开始时维护的池。换句话说，$B_{t,r}$ 包含轮次 t 中所有满足以下条件的重启 a：专家 i 的表现远优于 Rexp$r$，且专家 i 尚未进入池 $\mathcal{P}_r$。令 $M_{t,r}=|B_{t,r}|$。随机变量 $M_{t,r}$ 比 $N_{t,r}$ 更容易处理。首先，我们观察到

命题5.5。对于任意 $t\in[\epsilon^2T], r\ge 2$, $M_{t,r-1}\ge \frac{N_{t,r-3}}{4}$.

证明。对于任意时间步 $a=a_1\dots a_r\in A_{t,r}$，根据算法16的第1行，有 $a'=a_1\dots a_{r-1}\in B_{t,r-1}$。唯一的例外是，AssignInterval 以 a 启动；这在每个轮次中只发生 3 次。因此，我们有 $M_{t,r-1}\ge \frac{N_{t,r-3}}{4}$。 □

接下来我们对项 $\sum_{t=1}^{\epsilon^2T}M_{t,r}$ 进行界定。

命题5.6。对于任意 $t\in[\epsilon^2T], r\in[R]$，以至少 $1-1/(nT)^{\omega}$ 的概率，有 $\sum_{t=1}^{\epsilon^2T}M_{t,r}\le O(\epsilon^2T\log(nT))$。

证明。对于任意时间步 $a\in B_{t,r}$，LongExert 以概率 $\frac{1}{\epsilon\sqrt{n}}$ （LongExpert 第 7 行）将专家 i 采样到 O_r 中（对手并不知道这一点）。根据方程 (23) 和 LongExpert 第 10 行，专家 i 随后会被加入 $\mathcal{P}$，并存活 $\epsilon\sqrt{n}$ 个轮次。因此，以至少 $1-1/(nT)^{\omega(1)}$ 的概率，在 $\bigcup_{t\in[\epsilon^2T]}B_{t,r}$ 中，至多经过 $O(\epsilon\sqrt{n}\log(nT))$ 个时间步，就能将专家 i 加入池中并维持 $\epsilon\sqrt{n}$ 个轮次。后一种情况最多发生 $(\epsilon^2T+\epsilon\sqrt{n})/\epsilon\sqrt{n}=O(\frac{\epsilon T}{\sqrt{n}})$ 次（注意这里要求 $\epsilon^2T\ge\epsilon\sqrt{n}$），因此，

$$\sum_{t=1}^{\epsilon^2T}M_{t,r}=\sum_{t=1}^{\epsilon^2T}|B_{t,r}|\le O(\epsilon\sqrt{n}\log(nT))\cdot O(\frac{\epsilon T}{\sqrt{n}})=O(\epsilon^2T\log(nT)).$$ □

结合命题5.5和命题5.6，我们得出

$$\sum_{t=1}^{\epsilon^2 T} \frac{N_{t,r}}{\epsilon_r} \le \sum_{t=1}^{\epsilon^2 T} \frac{N_{t,r}}{\epsilon} = 3\epsilon T + \sum_{t=1}^{\epsilon^2 T} \frac{4M_{t,r}}{\epsilon} \le O(\epsilon T \log(nT)). \tag{24}$$

对任意 $r \ge 2$ 均成立，且对 $r = 1$、$N_{t,1} \le 1$ 也成立。结合方程（20）（21）（22）（24），可得

$$\sum_{t=1}^{T} (\ell_t(i_t) - \ell_t(i)) \le O(\epsilon T \log^2(nT)).$$

至此，我们完成证明。 □

结合遗憾分析（引理 5.4）和内存界（引理 5.3），我们证明了命题 5.1。

定理 *1.2* 的证明。命题 5.1 保证了当 $T \ge S$ 时的 $\widetilde{O}(\sqrt{\frac{nT}{S}})$ 遗憾。对于 corner case of $T \in [\frac{n}{S}, S]$（注意，由于我们的下界，只需考虑 $S \ge \Omega(\sqrt{n})$），我们的目标是获得总遗憾 $\widetilde{O}(\sqrt{\frac{nT}{S}})$。这可以通过分组技巧实现。具体而言，我们将专家 $[n]$ 划分为 $G = O(\frac{S}{T})$ 组，每组包含 $n' = O(\frac{n}{G})$ 个 $= O(\frac{nT}{S}) \in [1, n]$ 专家。我们对每组运行算法 13，$j \in [G]$ 使用空间 $S' = S/G = O(T)$。将每组视为一个元专家后，我们在元专家集合 $[G]$ 上运行乘法权重更新。对于任意专家 $i \in [n]$，假设它属于第 $j(i)$ 组（$j(i) \in [G]$）。经过 T 天后，可得

$$\begin{aligned}\sum_{t=1}^{T} \ell_t(i_t) - \sum_{t=1}^{T} \ell_t(i) &= \sum_{t=1}^{T} \ell_t(i_t) - \sum_{t=1}^{T} \ell_t(j(i)) + \sum_{t=1}^{T} \ell_t(j(i)) - \sum_{t=1}^{T} \ell_t(i) \\ &\le \widetilde{O}\left(\sqrt{T}\right) + \widetilde{O}\left(\frac{\sqrt{n'T}}{S'}\right) = \widetilde{O}\left(\sqrt{T}\right) + \widetilde{O}\left(\frac{\sqrt{nT/S} \cdot T}{T}\right) = \widetilde{O}\left(\sqrt{\frac{nT}{S}}\right).\end{aligned}$$

这里第二步依据乘法权重更新的遗憾保证和命题 5.1 得出。至此，证明完成。

□

6 自适应对手的下界

定理 6.1 (下界，定理 1.3 的正式版本)。设 $n, T \ge 1$ 和 $\epsilon \in (\log_{\sqrt{n}}^{4} n, 12)$，则不存在空间为 $o(\epsilon^{\sqrt{n}\log^3 n})$ 的在线学习算法，能够抵御自适应对手并达到 $\frac{\epsilon^2}{20} T$ 遗憾。

困难实例 设 $M = \epsilon\sqrt{n}$，将专家 $[n]$ 划分为 $N = n/M = \sqrt{n}/\epsilon$ 个区块，每个区块包含 M 位专家。开始时，对手对 SetDis-jointness 实例采样 N 个独立副本（每个实例的大小为 M），其中每个实例均从以下分布$\mu = (\mu_A, \mu_B)$中抽取

- 随机均匀地采样一个索引 $i^* \in [M]$，并设置 $(x_{A,i^*}, x_{B,i^*}) = (1, 1)$
- 对于其余坐标，$\frac{M-1}{3}$ 个坐标分别取值为 (0，0)、(1，0)、(0，1)。

换句话说，对手从 μ^N 中抽取 $x_A = (x_{A,1}, \ldots, x_{A,N}) \in \{0,1\}^n$，$x_B = (x_{B,1}, \ldots, x_{B,N}) \in \{0,1\}^n$，其中对于每个 $\alpha \in [N]$，$(x_{A,\alpha}, x_{B,\alpha}) \in \{0,1\}^{2M}$ 均从 μ 中抽取。令 i^*_α 为第 α 个副本的相交索引，$I^* = \{i^*_\alpha\}_{\alpha \in [N]}$ 为相交索引的集合。

自适应损失序列 整个损失序列被划分为 $\frac{10T}{N}$ 轮次，每个轮次包含 $\frac{N}{10}$ 天。每个轮次的损失序列分别构造。对于任意轮次 $t \in [\frac{10T}{N}]$ 和天 $b \in [\frac{N}{10}]$，令 $N_{t,\,b} \subseteq [N]$ 为在线算法在前 $(b-1)$ 天内执行过动作的区块集合。随后，损失向量 $\ell_{t,\,b} = (\ell_{t,\,b,\,1}, \ldots, \ell_{t,\,b,\,N}) \in [0,1]^n$ 被构造为

- 以概率 $1-\epsilon^2$, $\ell_{t,\,b,\,\alpha} = \vec{0}$ 对于任意区块 $\alpha \in [M]$;
- 以概率 ϵ^2,
  - For any played block $\alpha \in N_{t,b}$, $\ell_{t,b,\alpha} = \frac{1}{2} \cdot \vec{1}$;
  - For the rest of blocks,
    - With probability $1/2$, $\ell_{t,b,\alpha} = \vec{1} - x_{A,\alpha}$ for all $\alpha \in [N] \backslash N_{t,b}$,
    - With probability $1/2$, $\ell_{t,b,\alpha} = \vec{1} - x_{B,\alpha}$ for all $\alpha \in [N] \backslash N_{t,b}$,

换言之，自然方以概率 $1-\epsilon^2$ 将 $\ell_{t,\,b}$ 设为零损失 $\vec{0}$，并以概率 ϵ^2 将 $\ell_{t,\,b}$ 设为 $\vec{1} - x_A$ 或 $\vec{1} - x_B$，概率为 $1/2$；但对于已被在线学习算法"探索"过的区块，损失设为 $1/2$。

这相当 容易验证，最优专家的期望损失至多为 $\frac{\epsilon^2}{20}$ —.

引理 6.2（最优专家）。 最优专家的期望平均损失至多为 $\frac{\epsilon^2}{20}$。 —

证明。 考虑集合 $I^* = \{i_1^*, \ldots, i_N^*\}$。对于每个轮次 $t \in [\frac{10T}{N}]$，由于该轮次有 $\frac{N}{10}$ 天，因此我们知道， I^* 中的指标有 $9\,10$ 的比例在整个轮次中获得 0 损失，其余 $1\,10$ 的指标获得的（平均）损失至多为 $1/2 \cdot \epsilon^2$。因此， I^* 中至少存在一位专家，在整个 T 天内获得的平均损失至多为 $\frac{\epsilon^2}{20}$。 □

主要工作致力于界定内存为 $S = o(\epsilon^{\sqrt{n}\,\text{对数}^3\,n})$ 的在线学习算法的期望损失。

引理 6.3。 对于任意轮次 $t \in [T/B]$，内存为 $S = o(\epsilon^{\sqrt{n}\,\text{对数}^3\,n})$ 的在线学习算法获得的期望（平均）损失至少为 $\frac{\epsilon^2}{10}$。 —

证明分为三个步骤。我们首先构造一个三方通信问题 SetDisjointnessAd（带建议的集合不相交问题），并证明低内存的在线学习算法意味着存在一个通信成本较低的 SetDisjointnessAd 协议。然后，我们去除"建议"，证明 SetDisjointnessAd 的低通信成本协议意味着存在一个用于多个集合不相交问题副本的低通信成本协议，且其成功概率不可忽略。最后，我们利用 [BW15] 的直接乘积定理以及内部信息成本下界，证明成功概率的上界。

步骤 1 We 首先介绍 SetDisjointnessA 的三方通信问题 D.

定义 6.4 (SetDisjointnessAd)。 在SetDisjointnessAd通信问题中，有三方——*Alice*、*Bob* 和 *Charlie*。*Alice* 和 *Bob* 的输入分别为 x_A 和 x_B，取自 μ^N。*Charlie* 同时收到 x_A 和 x_B。通信过程如下：*Charlie* 首先向 *Alice* 和 *Bob* 各发送一份通信记录 Π_C （大小至多为 S），此后不得再发言。随后 *Alice* 和 *Bob* 进行多轮通信，并输出一个尽可能大的索引集合$I_{\text{out}} \subseteq I^*$。

令 Π_A 和 Π_B 分别为 Alice 和 Bob 的通信记录，$\Pi = (\Pi_A, \Pi_B)$ 为双方之间的完整通信记录。下面的归约基于这样一个事实：Alice、Bob 和 Charlie 可以模拟一个空间占用较低的在线学习算法。

引理 6.5. 假设存在一种在线学习算法，使用的空间至多为 S，且在第 t 个轮次中的期望损失至多为 $\frac{\epsilon^2}{10}$。那么，存在一个针对 SetDisjointnessAd 的随机公共币通信协议，使得

1. $|\Pi_C| \le S$；*2.* $|\Pi| \le \epsilon\sqrt{n}\log n \cdot (S + \log n) + 2\epsilon^{-1}\sqrt{n}\log n = o(n/\log^2 n)$；以及 *3.* $\mathbb{E}[|I_{\text{out}}|] \ge \frac{1}{25}N$。

证明。给定一个集合不相交问题实例，Charlie 在其输入 (x_A, x_B) 上模拟在线学习算法，共进行 $(t-1)$ 个轮次。其通信记录 Π_C 设为第 t 个轮次开始时的内存状态。随后，Alice 和 Bob 模拟在线学习算法。他们进行 $\frac{N}{10}$ 轮，在第 $b \in [\frac{N}{10}]$ 轮，他们使用公共随机性来确定损失向量。也就是说，

- **情况 1。** 以概率 $1-\epsilon^2$，损失向量为 $\vec{0}$。随后，Alice 和 Bob 在本地模拟在线学习算法（双方都拥有截至第 $(b-1)$ 轮的内存状态）。这不会产生任何通信成本。

- **情况 2。** 以概率 ϵ^2，损失向量取决于 Alice 或 Bob 的输入。具体而言，以概率 $1/2$，对于尚未操作过的区块，损失向量设为 $\vec{1} - x_A$；对于已操作过的区块，则设为 $\frac{1}{2}\cdot\vec{1}$。Alice 模拟在线学习算法，并将其内存状态以及动作 $i_{t,\,b}$ 加入通信记录。这会产生 $S + \log n$ 的通信成本。以剩余的 $1/2$ 概率，Bob 执行相同的操作。

如果情况 2 的总次数超过 $\epsilon\sqrt{n}\log n = 10\log n \cdot \frac{\epsilon^2 N}{10}$，Alice 和 Bob 就停止模拟在线学习算法。注意，根据切尔诺夫界，这种情况发生的概率至多为 $1/n^{\omega(1)}$。设 $I = \{i_{t,1}, \ldots, i_{t,\frac{N}{10}}\}$ 为这 $\frac{N}{10}$ 轮中执行的动作集合，最终输出集合 $I_{\text{out}} = I \cap I^*$ 通过额外两轮通信确定。

总通信成本可界定为

$$|\Pi_C| \le S \quad \text{and} \quad |\Pi| \le \epsilon\sqrt{n}\log n \cdot (S + \log n) + 2\epsilon^{-1}\sqrt{n}\log n.$$

为了界定 I_{out} 的大小，注意 Alice、Bob 和 Charlie 模拟在线学习算法，该算法的期望损失至多为 $\frac{\epsilon^2}{10}$（平均）。具体而言，在第 $b \in [N/10]$ 天，如果算法选择动作 $i_{t,\,b} \in I^* \backslash N_{t,\,b}$，那么它收到的期望损失为 0；否则，它收到的期望损失至少为 $\frac{1}{2}\epsilon^2$。后一种情况最多会发生 $\frac{(\epsilon^2/10)\cdot(N/10)}{\epsilon^2/2} = \frac{N}{50}$ 天。这意味着在线学习算法期望至少选择 I^* 中 $\frac{N}{10} - \frac{N}{50} = \frac{2}{25}N$ 个相交索引。注意，Alice 和 Bob 未能模拟在线学习协议的概率不超过概率 $1/n^{\omega(1)}$，因此他们的输出满足 $\mathbb{E}[|I_{\text{out}}|] \ge \frac{N}{25}$。至此，证明完毕。 □

步骤 2 接下来，我们定义多副本集合不相交问题这一通信问题。

定义 6.6 (多副本集合不相交问题)。在多副本集合不相交问题的通信问题中，*Alice* 和 *Bob* 分别接收输入 x_A, x_B，其取自 $(x_A, x_B) \sim \mu^N$。他们进行多轮通信，目标是输出索引集 I^*。

令 suc(Π, μ^N) 表示通信协议 Π 在分布μ^N上的成功概率。关键观察是，集合不相交问题Ad 的低通信成本协议蕴含一个用于多副本集合不相交问题的低通信成本协议，且具有不可忽略的成功概率。形式化地，

引理 6.7。 假设存在一个用于 SetDisjointnessAd 的通信协议 (Π_C, Π)，其$|\Pi_C| \le S$和 $|\Pi| \le S_2$，且其输出 $I_{\text{out}} \subseteq I^*$满足 $\mathbb{E}[|I_{\text{out}}|] \ge \frac{N}{25}$。那么存在一个随机公币通信协议 Π_N，能够解决多副本 集合不相交问题，其通信成本为 $O(S_2 \log\, n)$，且成功概率至少为 $2^{-O(S_{\log n})}$。

证明。 不失一般性地，可以假设 SetDisjointnessAd 的协议是确定性的，因为它是一个分布式通信问题。令 f_C: $\{0,1\}^{2n} \to \{0,1\}^S$ 表示 Charlie 的（确定性）策略，该策略将输入（x_A, x_B）$\in \{0,1\}^{2n}$映射为通信记录 $\Pi_C \in \{0,1\}^S$。对于任意可行的通信记录 $\Pi_C \in \{0,1\}^S$，定义

$$f_C^{-1}(\Pi_C) = \{(x_A, x_B) : f_C(x_A, x_B) = \Pi_C\}$$

也就是说， $f_C^{-1}(\Pi_C)$ 是 Π_C 的原像。令 μ_C 为 Π_C 的分布。

步骤 2-1 考 虑以下用于多副本集合不相交问题的通信协议 $\Pi_{N,1}$.

- Alice 和 Bob 首先使用公共随机性采样一个可能的建议 $\Pi_C \sim \mu_C$；
- Alice 和 Bob 随后根据 Π_C 模拟 SetDisjointnessAd 的协议 Π 。

通信成本为$|\Pi_{N,1}| = |\Pi| = S_2$。我们证明，以至少 $\frac{1}{2500} \cdot 2^{-S}$ 的概率，输出索引集$I^{\Pi,\Pi_C}_{\text{out}} \subseteq I^*$ 满足$|I_{\text{out}}| \ge \frac{N}{50}$。特别地，有

$$\begin{aligned}
&\Pr\left[|I^{\Pi,\widehat{\Pi}_C}_{\text{out}}| \ge \frac{N}{50}\right] \\
\ge &\sum_{\Pi_C \in \{0,1\}^S} \Pr[(x_A, x_B) \in f_C^{-1}(\Pi_C)] \cdot \Pr[\widehat{\Pi}_C = \Pi_C] \cdot \Pr\left[|^{\Pi,\widehat{\Pi}_C}_{\text{out}}| \ge \frac{N}{50} \mid (x_A, x_B) \in f_C^{-1}(\widehat{\Pi}_C)\right] \\
= &\sum_{\Pi_C \in \{0,1\}^S} \Pr[(x_A, x_B) \in f_C^{-1}(\Pi_C)]^2 \cdot \Pr\left[|I^{\Pi,\Pi_C}_{\text{out}}| \ge \frac{N}{50} \mid (x_A, x_B) \in f_C^{-1}(\Pi_C)\right] \ge \frac{1}{2500} \cdot 2^{-S}.
\end{aligned}$$

我们将成功事件限制为第一步中 $\Pi_C = \Pi_C$ 的情形。第二步成立，是因为 Alice 和 Bob 从 μ_C 中抽取 $\widehat{\Pi}_C$，因此 $\Pr[\widehat{\Pi}_C = \Pi_C] = \Pr[(x_A, x_B) \in f_C^{-1}(\Pi_C)]$。最后一步由柯西-施瓦茨不等式以及以下事实得出：

$$\begin{aligned}
&\sum_{\Pi_C \in \{0,1\}^S} \Pr[(x_A, x_B) \in f_C^{-1}(\Pi_C)] \cdot \Pr\left[|I^{\Pi,\Pi_C}_{\text{out}}| \ge \frac{N}{50} \mid (x_A, x_B) \in f_C^{-1}(\Pi_C)\right] \\
= &\Pr\left[|I^{\Pi,\Pi_C}_{\text{out}}| \ge \frac{N}{50}\right] > \frac{1}{50},
\end{aligned}$$

我们在最后一步用到了这样一个事实： $\mathbb{E}[|I^{\Pi,\Pi_C}{}_{\text{输出}}|] \ge \frac{N}{25}$

步骤 2-2 上述协议 $\Pi_{N,\ 1}$ 保证 Alice 和 Bob 找到一个大小至少为 $\frac{N}{50}$ 的相交索引集合，且其概率至少为 $2^{-O(S)}$。接下来，我们通过重复执行 $\Pi_{N,\ 1}$，共执行 $R = 100\log\ n$ 次，来增大集合的大小。我们的最终协议 Π_N如下所示。

- 对于 $r = 1, 2, \ldots, R$: – 利用公共随机性，Alice 和 Bob 采样置换函数 γ_r: $[N] \to [N]$和 $\theta_{r,\alpha}$: $[M] \to [M]$ $(\alpha \in [M])$ – Alice 和 Bob 将其输入 $x_A^{(r)}, x_B^{(r)}$ 重置为 $x_{A\ \alpha',\beta}^{(r)} = x_{A,\gamma_r(\alpha)\theta_{r,\alpha}(\beta)}$ 以及 $x_{B\ ,\alpha,\beta}^{(r)} = x_{B,\gamma_r(\alpha)\theta_{r,\alpha}(\beta)}$ – Alice 和 Bob 在输入 $x_A^{(r)}$ 和 $x_B^{(r)}$ 上运行通信协议 $\Pi_{N,1}$，并对输出集合执行相同的置换，即 $I^{(r)}{}_{\text{out}} \leftarrow \{(\gamma_r(\alpha)\ \theta_{r,\alpha}(\beta)) : (\alpha, \beta) \in I^{(r)}{}_{\text{out}}\}$

- 返回 $I_{\text{out}} \leftarrow I_{(1)\ \text{out}} \cup \cdots \cup I_{(r)\ \text{out}}$。

关键观察是，R次迭代中的输入集合（$x_A^{(1)}$，$x_B^{(1)}\ldots$，$x_A^{(R)}$，$x_B^{(R)}$）是从乘积分布（μ^N）^{R}中抽取的，这一点关键地利用了输入分布 μ的对称性。

令 $\mathcal{E}_r$ 为第 r次迭代的成功事件，即 $|I^{(r)}{}_{\text{输出}}| \geq \frac{N}{50}$，则有 —

$$\Pr[\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_R] = \Pr[\mathcal{E}_1] \cdots \Pr[\mathcal{E}_R] = 2^{-SR} \tag{25}$$

对于任意 $\alpha \in [n]$

$$\Pr\left[i_\alpha^* \notin \bigcup_{r\in[R]} I_{\text{out}}^{(r)} \mid \bigcup_{r\in[R]} \mathcal{E}_r\right] = \prod_{r\in[R]} \Pr\left[i_\alpha^* \notin I_{\text{out}}^{(r)} | \mathcal{E}_r\right] \leq \left(1 - \frac{1}{50}\right)^R \leq \frac{1}{n^2}. \tag{26}$$

这两个方程都利用了（$x_A^{(1)}$，$x_B^{(1)}\ldots$，$x_A^{(R)}$，$x_B^{(R)}$）的独立性。方程（26）的第二步成立，是因为 I^* 中的每个索引以相同的概率包含在 $I^{(r)}$ 输出 中（这是由于洗牌步骤），最后一步则由 $R = 100\log\ n$的取值选择得出。

对 $\alpha \in [N]$应用并集界，可得

$$\Pr\left[I^* \neq \bigcup_{r\in[R]} I_{\text{out}}^{(r)} \mid \bigcup_{r\in[R]} \mathcal{E}_r\right] \leq \frac{1}{n}$$

因此

$$\Pr[I_{\text{out}} = I^*] \geq \Pr[\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_R] \cdot \Pr\left[I^* = \bigcup_{r\in[R]} I_{\text{out}}^{(r)} \mid \bigcup_{r\in[R]} \mathcal{E}_r\right] \geq 2^{-O(S\log n)}$$

其中，第二步来自方程 (25)(26)。最后，我们注意到通信成本等于$O(S_2R)\ = O(S_2 \log n)$。

□

步骤 3 最后，我们证明多副本 SetDisjoint-ness 的成功概率下界。特别地，我们证明

引理 6.8. 假设一个针对多份集合不相交问题 的通信协议 Π以至少$\exp(-O(N))$的概率成功，则 $|\Pi| \geq \Omega(n/\log\ n)$。

我们将用到以下由 [BW15] 提出的直积定理

定理 *6.9*（直接乘积，[*BW*15] 的定理 *2*）。存在一个全局常数 $a > 0$，使得对于定义在分布 ν 上的任意双方通信问题 f，若 $\mathit{suc}(\nu, f, I) \leq \frac{2}{3}$，且 $K\,\mathit{log}(K) \leq \alpha N \cdot I$，则有 $\mathit{suc}(\mu^N, f^N, K) \leq \mathit{exp}(-\Omega(N))$。这里，$\mathit{suc}(\nu, f, I)$ 表示在 μ下计算 f 且内部信息成本为 I 的协议的最大成功概率。

我们证明，为了以至少 $2/3$ 的概率成功，集合不相交问题的通信协议在 μ 下必须具有 $\Omega(M) = \Omega(\epsilon\sqrt{n})$ 的内部信息成本。

引理 6.10。在分布 μ 下，任何用于不相交性问题（成功概率至少为 $2/3$）的通信协议，其内部信息成本至少为 $\Omega(M) = \Omega(\epsilon\sqrt{n})$。

毋庸赘述引理 6.10 的（标准）证明（见附录 B），结合定理 6.9，我们已经证明了引理 6.8。结合引理 6.8、引理 6.7 和引理 6.5，我们已经证明了引理 6.3，即该在线学习算法的平均遗憾至少为 $\frac{\epsilon^2}{10}$。由于最优专家的期望平均损失至多为 $\frac{\epsilon^2}{20}$，这就完成了定理 6.1 的证明。

6.1 强自适应对手下的 $\Omega(n)$ 内存下界

最后，我们采用定理 1.3 的方法，证明针对强自适应对手的线性内存下界 $\Omega(n)$，这改进了 [PZ23] 的结果。强自适应对手可以根据算法在当前轮次的策略（对 $[n]$ 上的概率分布）来选择损失向量（而不只是依据过去的动作）。有关这一概念的详细讨论，请参阅 [PZ23] 。

定理 6.11。面对强自适应对手时，任何能够实现 $o(T)$ 遗憾的在线学习算法，都需要 $\Omega(n)$ 空间。

证明。困难实例的构造沿用了定理 1.3 的高层方法，尽管证明只用到了简单的计数论证。我们用反证法，假设 $S < \frac{n}{2000}$。令 I 为一组特殊专家，且 $|I| = 10S$。给定策略 $p \in \Delta_n$，损失向量 $\ell_{I,p}$ 按如下方式构造。

- 对于任意非特殊专家 $i \in [n] \backslash I$，$\ell_{I,p}(i) = 1$;
- 对于任意特殊专家 $i \in [I]$，如果 i 不在I中排名前 $2S$ 的高权重专家之列，即 $|\{i' \in I : p(i') \geq p(i)\}| \geq 2S$，则令 $\ell_{I,p}(i) = 0$；否则（即 i 属于 I中排名前 $2S$的高权重专家），令 $\ell_{I,p}(i) = 1$。

我们立即得到

引理 6.12。对于任意固定策略 $p \in \Delta_n$，满足 $\langle p, \ell_{I,p} \rangle < \frac{2}{5}$ 的集合 $I \subseteq [n]\,(|I| = 100S)$ 至多有 $2^{10S} \cdot \binom{n}{8S}$ 个。

证明。我们假设 $p_1 \geq \cdots \geq p_n$，这不失一般性。若 $\sum_{i=10S+1}^{n} p_i \geq \frac{2}{5}$，则易于验证，对于任意 I，都有 $\langle p, \ell_{I,p} \rangle \geq \frac{2}{5}$。考虑 $\sum_{i=10S+1}^{n} p_i < \frac{2}{5}$ 的情况，若 $|I \cap [10S]| \leq 2S$，则 $\ell_{I,\,p}(i) = 1$对于任意 $i \in [10S]$，因此，

$$\langle p, \ell_{I,p} \rangle \geq \sum_{i=1}^{10S} p_i \ell_{I,p}(i) = 3/5.$$

因此， I 的总数至多为

$$\sum_{k=2S+1} \binom{10S}{k}\binom{n-10S}{10S-k} \le 2^{10S}\cdot\binom{n}{8S}.$$

□

回到定理 6.11 的证明。损失序列生成如下。

- 对手从 $[n]$ 中均匀随机选择特殊专家集合 I ($|I| = 100S$)。该集合在开始时选定，此后保持不变。

- 在第 t 天，给出 在算法的策略 p_t 下，对手构造损失向量 $\ell_{I,\ p}$ $_t$.

每天，至多有 $2S$ 个特殊专家获得损失 1，其余 $8S$ 个特殊专家获得损失 0。因此，最佳专家获得的损失至多为 $_1$ $_5$。我们的目标是证明，对于每一天，Alice 的期望损失至少为 $_3$ $_{10}$ ，并且对 I 的选择以高概率成立。

该算法总共有 2^S 个可能的内存状态 X。对于每个内存状态 $x \in X$，假设算法输出一个策略 $p_x \in \Delta_n$ （注意， p_x 可能是一个随机变量）。定义

$$T_x := \left\{ I : \Pr[\langle p_x, \ell_{I,p_x}\rangle < \frac{2}{5}] \ge 0.1, I \subseteq [n], |I| = 10S \right\}.$$

根据引理 6.12，我们知道 $|T_x| \le 10\cdot 2^{10S}\binom{n}{8S}$。对所有内存状态$x \in X$取并集，可得

$$\left|\bigcup_{x\in X} T_x\right| \le 2^S\cdot 10\cdot 2^{10S}\cdot\binom{n}{8S} \le \frac{1}{10}\binom{n}{10S}.$$

最后一步由参数的选择得出。

因此，我们得出结论：以至少 0.9 的概率，对手会抽取一组特殊专家I，使得 $I \notin \bigcup_{x\in X} T_x$。这意味着，在每一天，算法收到的期望损失至少为 $0.9\cdot\frac{2}{5} \ge \frac{3}{10}$。证明到此结束。

□

参考文献

[ACBFS02] Peter Auer、Nicolo Cesa-Bianchi、Yoav Freund 和 Robert E Schapire。非随机多臂老虎机问题。《*SIAM* 计算杂志》，32(1):48–77，2002年。[ACK19] Sepehr Assadi、Yu Chen 和 Sanjeev Khanna。图流算法的多项式遍数下界。载于第*51*届*ACM SIGACT*计算理论年会论文集，第265–276页，2019年。[ACNS23] Anders Aamand、Justin Chen、Huy Le Nguyen 和 Sandeep Silwal。专家学习的改进空间界。*https://arxiv.org/abs/2303.01453*，2023年。[ADF+22] Ioannis Anagnostides、Constantinos Daskalakis、Gabriele Farina、Maxwell Fishelson、Noah Golowich 和 Tuomas Sandholm。多人一般和博弈中相关均衡的近最优无悔学习。载于第*54*届*ACM SIGACT*计算理论年会论文集，第736–749页，2022年。[AHK12] Sanjeev Arora、Elad Hazan 和 Satyen Kale。乘法权重更新方法：一种元算法及其应用。计算理论，8(1):121–164，2012年。

[AKP22] Arpit Agarwal、Sanjeev Khanna 和 Prathamesh Patil。多遍流式老虎机问题中尖锐的内存—遗憾权衡。学习理论会议（COLT），2022年。[AW20] Sepehr Assadi 和 Chen Wang。有限内存下的探索：用于抛硬币、噪声比较和多臂老虎机的流式算法。载于第52届ACM SIGACT计算理论年会论文集（STOC），2020年。[BBF+21] Gavin Brown、Mark Bun、Vitaly Feldman、Adam Smith 和 Kunal Talwar。何时必须记忆无关训练数据才能实现高精度学习？载于第53届ACM SIGACT计算理论年会论文集，第123–132页，2021年。[BBS22] Gavin Brown、Mark Bun 和 Adam Smith。自然模型学习的强内存下界。学习理论会议（COLT），2022年。[BCB+12] Sébastien Bubeck、Nicolo Cesa-Bianchi 等。随机和非随机多臂老虎机问题的遗憾分析。《机器学习基础与趋势》，5(1):1–122，2012年。[BDGR22] Adam Block、Yuval Dagan、Noah Golowich 和 Alexander Rakhlin。平滑在线学习与统计学习一样简单。载于学习理论会议（COLT），2022年。[BGZ15] Omar Besbes、Yonatan Gur 和 Assaf Zeevi。非平稳随机优化。《运筹学》，63(5):1227–1244，2015年。[BKM+22] Amos Beimel、Haim Kaplan、Yishay Mansour、Kobbi Nissim、Thatchaphol Saranurak 和 Uri Stemmer。针对自适应对手的动态算法：通用构造与下界。载于第54届ACM SIGACT计算理论年会论文集，第1671–1684页，2022年。[BLLW19] Sébastien Bubeck、Yuanzhi Li、Haipeng Luo 和 Chen-Yu Wei。老虎机问题中改进的路径长度遗憾界。载于学习理论会议，第508–528页。PMLR，2019年。[BM07] Avrim Blum 和 Yishay Mansour。从外部遗憾到内部遗憾。《机器学习研究杂志》，8(6)，2007年。[BRWY13] Mark Braverman、Anup Rao、Omri Weinstein 和 Amir Yehudayoff。通信复杂度中的直积。载于2013年IEEE第54届计算机科学基础年会论文集，第746–755页。IEEE，2013年。[BW15] Mark Braverman 和 Omri Weinstein。交互式信息里程表及其应用。载于第47届ACM计算理论年度研讨会论文集，第341–350页，2015年。[BYJKS04] Ziv Bar-Yossef、Thathachar S Jayram、Ravi Kumar 和 D Sivakumar。数据流与通信复杂度的信息统计方法。《计算机与系统科学杂志》，68(4):702–732，2004年。[BZJ23] Moïse Blanchard、Junhui Zhang 和 Patrick Jaillet。凸优化中实现最优查询复杂度需要二次内存：质心法是帕累托最优的。arXiv预印本 arXiv:2302.04963，2023年。

[CBL06] Nicolo Cesa-Bianchi 和 Gábor Lugosi。《预测、学习与博弈》。剑桥大学出版社，2006年。[CKM+11] Paul Christiano、Jonathan A Kelner、Aleksander Madry、Daniel A Spielman 和 Shang- Hua Teng。无向图中的电流、拉普拉斯系统以及更快的最大流近似算法。载于《第四十三届 ACM 计算理论研讨会》（STOC）论文集，第 273–282 页，2011年。[CP20] Xi Chen 和 Binghui Peng。博弈中的对冲：更快地收敛于外部遗憾和交换遗憾。《神经信息处理系统进展》，33：18990–18999，2020年。[CPP22] Xi Chen、Christos Papadimitriou 和 Binghui Peng。持续学习的内存界。载于《2022 年 IEEE 第 63 届计算机科学基础年会》（FOCS），2022年。[DHL+20] Miroslav Dudík、Nika Haghtalab、Haipeng Luo、Robert E Schapire、Vasilis Syrgkanis 和 Jennifer Wortman Vaughan。预言机高效在线学习与拍卖设计。《ACM 学报》（JACM），67(5)：1–57，2020年。[DS18] Yuval Dagan 和 Ohad Shamir。以较少的内存和通信检测相关性。载于《学习理论会议》（COLT），2018年。[Fel20] Vitaly Feldman。学习是否需要记忆化？一个关于长尾的短篇故事。载于《第52届ACM SIGACT计算理论年会论文集》，第 954–959 页，2020年。[FS97] Yoav Freund 和 Robert E Schapire。在线学习的决策理论泛化及其在提升中的应用。《计算机与系统科学杂志》，55(1)：119–139，1997年。[FS99] Yoav Freund 和 Robert E Schapire。利用乘法权重进行自适应博弈。《博弈与经济行为》，29(1-2)：79–103，1999年。[FSSW97] Yoav Freund、Robert E Schapire、Yoram Singer 和 Manfred K Warmuth。使用并组合专门化的预测器。载于《第二十九届 ACM 计算理论研讨会论文集》，第 334–343 页，1997年。[GH16] Dan Garber 和 Elad Hazan。半定规划近似算法的次线性时间算法。《数学规划》，158(1)：329–361，2016年。[GHM19] Alon Gonen、Elad Hazan 和 Shay Moran。隐私学习蕴含在线学习：一种高效归约。《神经信息处理系统进展》，32，2019年。[GKLR21] Sumegha Garg、Pravesh Kumar Kothari、Pengda Liu 和 Ran Raz。带噪声学习奇偶的记忆-样本下界。载于第24届组合优化问题近似算法国际会议（APPROX 2021）和第25届随机化与计算国际会议（RANDOM 2021），2021年。[GLM20] Alon Gonen、Shachar Lovett 和 Michal Moshkovitz。迈向有界记忆学习的组合刻画。《神经信息处理系统进展》（NeurIPS），2020年。

[GRT18] Sumegha Garg、Ran Raz 和 Avishay Tal。基于提取器的学习时间-空间下界。载于《第50届ACM SIGACT计算理论研讨会》（STOC）论文集，2018年。[GRT19] Sumegha Garg、Ran Raz 和 Avishay Tal。两遍学习的时间-空间下界。载于第34届计算复杂性会议（CCC），2019年。[Haz16] Elad Hazan。《在线凸优化导论》。《优化基础与趋势》，2(3-4)：157–325，2016年。[HHSY22] Nika Haghtalab、Yanjun Han、Abhishek Shetty 和 Kunhe Yang。面向超越最坏情况对手的预言机高效在线学习。《神经信息处理系统进展》（NeurIPS），2022年。[HK16] Elad Hazan 和 Tomer Koren。在线学习中优化的计算能力。载于《第48届ACM计算理论年度研讨会》（STOC）论文集，第 128–141 页，2016年。[HKM+20] Avinatan Hasidim、Haim Kaplan、Yishay Mansour、Yossi Matias 和 Uri Stemmer。通过差分隐私实现对抗鲁棒流式算法。《神经信息处理系统进展》，33：147–158，2020年。[HLZ20] Sam Hopkins、Jerry Li 和 Fred Zhang。通过遗憾最小化，简单实现鲁棒重尾均值估计。《神经信息处理系统进展》（NeurIPS），2020年。[Hol07] Thomas Holenstein。平行重复：简化与无信号情形。载于《第三十九届 ACM 计算理论研讨会论文集》，第 411–419 页，2007年。[HRS22] Nika Haghtalab、Tim Roughgarden 和 Abhishek Shetty。针对自适应对手的平滑分析。载于《2021年 IEEE 第62届计算机科学基础年会》（FOCS），2022年。[HS07] Elad Hazan 和 Comandur Seshadhri。在线决策问题的自适应算法。载于《计算复杂性电子讨论会》（ECCC），2007年。[Kla10] Hartmut Klauck。不相交性的强直接积定理。载于《第四十二届 ACM 计算理论研讨会论文集》，第 77–86 页，2010年。[KM17] Adam Klivans 和 Raghu Meka。利用乘法权重学习图模型。载于《2017年 IEEE 第58届计算机科学基础年会》（FOCS），第 343–354 页。IEEE，2017年。[KV05] Adam Kalai 和 Santosh Vempala。在线决策问题的高效算法。《计算机与系统科学杂志》，71(3)：291–307，2005年。[KVE15] Wouter M Koolen 和 Tim Van Erven。面向专家和组合博弈的二阶分位数方法。载于《学习理论会议》，第 1155–1175 页。PMLR，2015年。[LSPY18] David Liau、Zhao Song、Eric Price 和 Ger Yang。常数空间中的随机多臂老虎机。载于《国际人工智能与统计会议》（AISTATS），2018年。

[LW89] N Littlestone 和 MK Warmuth。加权多数算法。载于第30届计算机科学基础年度研讨会（FOCS），1989年。[MPK21] Arnab Maiti、Vishakha Patil 和 Arindam Khan。有界臂记忆的多臂老虎机：最优臂识别和遗憾最小化的近乎最优保证。神经信息处理系统进展（NeurIPS），2021年。[MSSV22] Annie Marsden、Vatsal Sharan、Aaron Sidford 和 Gregory Valiant。高效凸优化需要超线性内存。载于学习理论会议（COLT），2022年。[OC98] Erik Ordentlich 和 Thomas M Cover。实现事后最优投资组合的代价。《运筹学数学》，23(4):960–982，1998年。[PST95] Serge A Plotkin、David B Shmoys 和 Éva Tardos。分数装填与覆盖问题的快速近似算法。《运筹学数学》，20(2):257–301，1995年。[PZ23] Binghui Peng 和 Fred Zhang。次线性空间中的在线预测。载于第三十四届 ACM-SIAM 离散算法研讨会论文集，2023年。[Raz95] Ran Raz。一个并行重复定理。载于第二十七届 ACM 计算理论研讨会论文集，第447–456页，1995年。[Raz17] Ran Raz。一大类学习问题的时间-空间下界。载于2017年 IEEE 第58届计算机科学基础年度研讨会（FOCS），2017年。[Raz18] Ran Raz。快速学习需要良好的内存：奇偶学习的时间-空间下界。《ACM 学报》（JACM），66(1):1–18，2018年。[Rou13] Tim Roughgarden。Cs364a：算法博弈论。第17讲：无遗憾动态，2013年11月。[She14] Alexander A Sherstov。通信复杂度理论：集合不相交的三十五年。载于计算机科学数学基础国际研讨会（MFCS），第24–43页。施普林格，2014年。[SSV19] Vatsal Sharan、Aaron Sidford 和 Gregory Valiant。小误差线性回归的内存-采样权衡。载于第51届 ACM SIGACT 计算理论研讨会（STOC）论文集，2019年。[SVW16] Jacob Steinhardt、Gregory Valiant 和 Stefan Wager。内存、通信与统计查询。载于学习理论会议（COLT），2016年。[SWXZ22] Vaidehi Srinivas、David P. Woodruff、Ziyu Xu 和 Samson Zhou。专家问题的内存界。载于第54届ACM SIGACT计算理论年会论文集（STOC），2022年。[WL18] Chen-Yu Wei 和 Haipeng Luo。针对对抗性老虎机的更多自适应算法。载于学习理论会议，第1263–1291页。PMLR，2018年。[WZZ23] David Woodruff、Fred Zhang 和 Samson Zhou。专家学习的流式算法：确定性与鲁棒性。2023年。

概率工具

我们记 本文中使用的切尔诺夫界和阿祖马-霍夫丁界的版本 .

引理 A.1 (切尔诺夫界)。 设 $X = \sum_{i=1}^{n} X_i$，其中 $X_i = 1$ 以概率 p_i 发生， $X_i = 0$以概率 $1 - p_i$ 发生，且所有 X_i 彼此独立。设 $\mu = \mathbb{E}[X] = \sum_{i=1}^{n} p_i$。则 *1.* 概率$[X \geq (1+\delta)\mu] \leq \exp(-\delta\mu/3)$，$\forall \delta > 1$； *2.* 概率$[X \leq (1-\delta)\mu] \leq \exp(-\delta^2\mu/2)$， $\forall 0 < \delta < 1$。

引理 A.2 (阿祖马-霍夫丁界)。 设 $X_0, \ldots, X_n$ 是关于筛选 $F_0 \subseteq F_1 \cdots \subseteq F_n$ 的鞅序列，使得对于 $Y_i = X_i - X_{i-1}$, $i \in [n]$，有 $|Y_i| = |X_i - X_{i-1}| \leq c_i$。那么

$$\Pr[|X_t - X_0| \geq t] \leq 2\exp\left(-\frac{t^2}{2\sum_{i=1}^{n} c_i^2}\right).$$

B 第 6 节中缺失的证明

*Pr*引理 *6.10* 的证明。 考虑定义在 $\{0,1\}^{2M}$ 上的以下分布 μ'

- 均匀随机采样一个索引 $i^* \in [M]$，并令 $(x_{A,\,i},\ x_{B,\,i}) = (1,\ 1)$
- 对于其余坐标，令 $(x_{A,\,i},\ x_{B,\,i})$ 以相同的概率取值为 (0，1)，(1，0)，(0，0)。

μ 与 μ' 之间唯一（细微）的区别在于，在 μ 下，$(x_A,\ x_B)$ 恰好包含 $\binom{M-1}{3}$ 对 (0，0)，(0，1)，(1，1)。

已知 μ' 的内部信息成本下界，特别地，我们有 e

引理 B.1 （[ACK19]的定理 3）。 对于分布 μ' 下的不相交性，任何成功概率至少为 $2/3$ 的通信协议，其内部信息成本至少为 $\Omega(M)$。

给定分布 μ' 下集合不相交问题的一个实例（x'_A，x'_B），我们通过增加几个坐标，将其转换为来自 μ 的输入。Alice 和 Bob 将各自的输入扩充为 x_A，x_B （x_A，$x_B \in \{0,1\}^{3M+1}$，使得恰好有 M 对（0，1），（1，0），（0，1）。这可以通过以下方式实现：（1） Alice 添加 $M - |x'_A| + 1$ 个值为 1 的坐标；（2）Bob 添加 $M - |x'_B| + 1$ 个（不同的）值为 1 的坐标。然后，他们使用公共随机性打乱坐标，再在 x_A, x_B 上运行协议 Π （该协议是为来自 μ 的输入设计的）。令 γ 为用于打乱的公共随机性。

显然，Alice 和 Bob 成功的概率至少为 $2/3$。为界定内部信息成本，我们注意到

$$\begin{aligned}
\mathsf{IC}_{\mu'}(\Pi') &= I(X'_A; \Pi' | X'_B) + I(X'_B; \Pi' | X'_A) \\
&= I(X'_A; \Pi, \gamma | X'_B) + I(X'_B; \Pi, \gamma | X'_A) \\
&= I(X'_A; \Pi | X'_B, \gamma) + I(X'_B; \Pi | X'_A, \gamma) \\
&= I(X_A; \Pi | X_B, \gamma) + I(X_B; \Pi | X_A, \gamma) \\
&\leq I(X_A; \Pi | X_B) + I(X_B; \Pi | X_A) \\
&= \mathsf{IC}_{\mu}(\Pi).
\end{aligned}$$

第三步由互信息链式法则和 $I(X'_A; \gamma | X'_B) = 0$ 得出。第四步成立是因为（X_B，γ）与（X'_B，γ）一一对应，且（X_A，γ）与（X'_A，γ）一一对应。第五步成立是因为 $I(\gamma; \Pi | X_A,\ X_B) = 0$，也就是说，给定 X_A 和 X_B 后，通信记录 Π 与置换函数相互独立。证明至此完成。 □