arXiv:2304.14058v1 [cs.CC] 27 Apr 2023

PAC学习的参数化理论

Cornelius Brand[1], Robert Ganian[1], Kirill Simonov[2]

[1] 维也纳工业大学算法与复杂性组，奥地利[2]哈索·普拉特纳研究所算法工程讲席，德国{cbrand, rganian}@ac.tuwien.ac.at,kirill.simonov@hpi.de

摘要

大概正确（即 PAC）学习是样本复杂度理论的核心概念，而高效 PAC 可学习性通常被视为经典计算复杂性中 P 类的自然对应物。然而，尽管新兴的参数化复杂度理论使我们得以突破经典计算复杂性中的 P-NP“二分法”，并确定众多问题可处理性的精确边界，样本复杂度领域却没有类似的理论能够超越高效 PAC 可学习性。

作为核心贡献，我们通过建立参数化 PAC 学习理论填补了这一空白，使我们得以重新审视近期若干融入参数化复杂度要素的 PAC 学习结果。在该理论中，我们确定了两种而非一种固定参数可学习性概念，它们分别对应 FPT 类——参数化复杂性范式的核心概念；我们还发展了排除固定参数可学习性所需的方法。随后，我们展示了该理论的应用，借此确定合取范式和析取范式学习，以及一系列图上的学习问题，其可处理性的更精细边界。

1 引言

尽管文献中至今已经考虑了许多不同的样本复杂度模型，但大概正确（即 PAC）学习这一基本概念（Valiant 1984）仍然是一个核心支柱，它能够从理论上捕捉并解释学习算法在广泛情境中的成功。直观地说，在经典的适当PAC设定中，我们要研究的是：根据未知分布抽取一定数量的带标签样本[1] 后，是否能够从指定的假设空间中识别（或“学习”）出一个假设。作为最理想的情况，我们希望拥有一种算法，在使用至多多项式时间并且至多使用多项式数量的样本后，能够以任意高的精度和任意高的概率学习出一个假设。这类算法称为高效的 PAC 学习算法，而能够接受此类算法的学习问题所构成的类，则是经典计算问题复杂性理论中 P类的一个自然对应物。

[1]形式化定义将在预备知识部分给出。

尽管存在这种对应关系，我们对学习问题样本复杂度的理解，仍明显落后于目前已经积累的关于计算问题时间复杂度的大量知识。事实上，尽管理解计算问题中 NP- 困难实例类与多项式可处理实例类之间的区别仍然是一个重要的研究方向，但在过去二十年中，研究重点逐渐转向对这类问题可处理性边界进行更细粒度的分析。特别是，参数化复杂度（Cygan 等人（2015）；Downey和Fellows，2013）已经发展成为一种重要范式，相比经典复杂性理论，它能够更深入地理解可处理性的极限。如今，我们看到关于相关问题参数化复杂度的研究出现在广泛的学术场合，几乎涵盖计算机科学的所有领域。参数化范式背后的主要思想是：研究问题的复杂度时，不仅考虑输入规模 n，还要依据一个数值参数 k 进行分析，该参数刻画输入的某种性质；于是，核心概念便是固定参数可处理性（FPT），这意味着对于某个可计算函数 f，相关问题可以在 $f(k) \cdot n^{\mathcal{O}(1)}$ 时间内得到解决。

鉴于其无处不在的应用，探讨参数化复杂性分析的原则是否（以及如何）也能用于拓展我们对样本复杂度理论基础的理解，超越通过高效 PAC 可学习性视角所能实现的范围，是很自然的。事实上，已有少数论文（Li 和 Liang 2018；Arvind、K¨obler 和 Lindner 2009；Alekhnovich 等人 2008；G¨artner 和 Garriga 2007）通过识别出一些似乎可作为参数化复杂性中典型的固定参数算法之直观对应物的学习算法，暗示了这一新兴领域的潜力。然而，针对 PAC 学习框架的通用参数化扩展，其理论基础尚未得到形式化和发展，从而阻碍了这一连接两个研究群体的独特桥梁的进一步发展。

贡献。作为我们的首个概念性贡献，我们为 PAC 学习框架的一般参数化扩展奠定了基础。关键的是，在样本复杂度情境下定义固定参数可处理性的类比概念时，我们表明有两种自然的定义

需要加以考虑：参数化学习问题是

1. 如果可以使用固定参数算法，从多项式数量的样本中学习到合适的 PAC 假设，则称为可在固定参数可处理时间内学习；2. 如果可以使用固定参数算法学习到合适的 PAC 假设，则称为可用固定参数样本复杂度学习。

本质上，在后一种情况下，所使用的样本总数不必是多项式数量，但当然仍受算法的（固定参数）运行时间上界的限制。该框架的一个重要特征是，参数不仅可以依赖于所要寻找的概念（正如先前的研究（Arvind、K¨obler 和 Lindner 2009）所考虑的那样），还可以依赖于分布的性质。这可以类比于经典固定参数复杂度范式中参数的使用方式：参数既可以描述我们对输出所要求的性质（例如解的大小），也可以表示我们对输入施加的限制（例如输入图的结构参数）。

本文提出的参数化 PAC 学习框架还包含一种可用于获得下界的机制，从而排除 FPT 时间学习算法和可用固定参数样本复杂度学习的算法的存在；具体而言，这是通过在参数化复杂性理论中与经过充分研究的 W 层级建立联系来实现的。此外，我们还给出了复杂性类 XP 在样本复杂度方面的对应概念；与固定参数情形类似，有必要区分我们限制的究竟只是运行时间，还是样本数量。

在奠定基础之后，我们将注意力转向两个主要的问题设置，旨在说明这些概念如何拓展我们对样本复杂度的认识。首先，我们研究学习 DNF公式和 CNF公式的可学习性边界，这是 PAC 学习理论中最经典的问题之一。

一方面，对于每个固定的 k，已知 k-合取范式和 k-析取范式公式可以被高效地 PAC 可学习，尽管已知算法的运行时间形式为 $\mathcal{O}(n^k)$。作为我们的第一个结果，我们证明，尽管这两个学习问题都可以在 xp 时间内学习，但在公认的复杂性假设下，当以 k 为参数进行参数化时，它们既不能在固定参数可处理时间内学习，也不能在固定参数样本复杂度下学习。另一方面，对于每个 $p > 1$，众所周知，p-项析取范式和 p-子句合取范式公式不能被高效地 PAC 学习（Pitt 和 Valiant（1988）；Alekhnovich 等人 2008 年），除非 P $=$ NP。我们在此探讨，能否通过合适的参数化来突破这一可处理性障碍。我们证明，对于目标概念最自然的参数化方式，学习 p-项析取范式和p-子句合取范式公式仍然难以处理；但利用对分布的参数化，可以通过固定参数可处理时间算法学习这类公式。具体而言，不难看出，当每个样本中设为 True 的变量至多只有一个时，该学习问题是可处理的；我们还证明，以不受此限制的变量数量为参数时，可以将这一结果推广为一种非平凡的固定参数可处理时间学习算法，其中这些变量（即始终可以取任意值的变量）

虽然 CNF 和 DNF 学习是 PAC 学习的基础，但在研究的最后部分，我们转向一个在样本复杂度和参数化复杂性范式中都得到广泛研究的设置：图问题。事实上，顶点覆盖和子图检测等基础图问题曾作为参数化复杂性理论发展的试验平台，也在机器学习和样本复杂度领域得到深入研究（Nguyen and Maehara 2020；Abasi and Bshouty 2019；Abasi、Bshouty 和 Mazzawi 2018；Damaschke and Muhammad 2010；Alon and Asodi 2005；Choi and Kim 2010）。在这里，我们给出了一个元定理，表明一大类图问题是可在固定参数可处理时间内学习的：特别地，对于任意图类 $\mathcal{H}$，只要该图类可以由有限个禁用诱导子图刻画，我们就为寻找一个到 $\mathcal{H}$ 的顶点删除集提供了一个可在固定参数可处理时间内运行的学习算法。作为一个特例，这涵盖了隐藏顶点覆盖的学习问题（Damaschke and Muhammad 2010）。最后，我们排除了对由禁用子式刻画的类 $\mathcal{H}$ 得到类似结果的可能性。因此，例如，当以其大小作为参数时，隐藏反馈顶点集既不可在固定参数可处理时间内学习，也不可用固定参数样本复杂度学习。

2 预备知识

作为基本符号和术语，我们记 $\{0,1\}^* = \bigcup_{m\in\mathbb{N}}\{0,1\}^m$。$\{0,1\}^n$ 上的分布是一个映射 $\mathcal{D}$：$\{0,1\}^n \to [0,1]$，满足 $\sum_{x\in\{0,1\}^n}\mathcal{D}(x) = 1$，而 $\mathcal{D}$ 的支撑集是集合 $\operatorname{supp}\mathcal{D} = \{x \mid \mathcal{D}(x) > 0\}$。我们首先回顾 PAC 学习和参数化复杂性理论的基础知识。

PAC 学习

我们首先将本文所研究的经典学习理论形式化（Valiant 1984；Mohri、Rostamizadeh 和 Talwalkar 2012）。

定义 2.1。概念是任意布尔函数c：$\{0,1\}^n \to \{0,1\}$。如果 $x \in \{0,1\}^n$ 满足 $c(x) = 1$，则称赋值 $x \in \{0,1\}^n$ 为 c 的正样本，否则称为负样本。概念类 $\mathcal{C}$ 是概念的集合。对于每个$m \in \mathbb{N}$，我们记 $\mathcal{C}_m = \mathcal{C} \cap \mathcal{B}_m$，其中 $\mathcal{B}_m$ 是所有m元布尔函数的集合。

定义 2.2. 设 $\mathcal{C}$ 是一个概念类。满射 ρ：$\{0,1\}^* \to \mathcal{C}$ 称为 $\mathcal{C}$ 的表示方案。我们称每个满足 $\rho(r)$ 的 $r = c$ 为概念 c 的一个表示。

简单来说，概念可以是任意函数，而表示则使它们变得“具体”，也是我们通常期望学习算法输出的内容。

定义 2.3. 学习问题是一个有序对 $(\mathcal{C}, \rho)$，其中 $\mathcal{C}$ 是一个概念类，ρ 是 $\mathcal{C}$ 的表示方案。

定义 2.4. 针对学习问题 $(\mathcal{C}, \rho)$ 的学习算法是一种随机化算法，满足：

1. 它获取值 n、ε、δ 作为输入，其中 n 是一个整数，而 $0 < \varepsilon$、$\delta \leq 1$ 是有理数。

2. 它可以通过一个返回带标签样本 $(x, c^*(x))$ 的预言机，访问某个概念 $c^* = \rho(r^*)$ 的隐藏表示 r^* 以及 $\{0,1\}^n$ 上的隐藏分布 $\mathcal{D}_n$，其中 $x \in \{0,1\}^n$ 是从 $\mathcal{D}_n$ 中随机抽取的。
3. 该算法的输出是某个概念的表示，称为其假设。

注释 2.5。显然，算法可以从预言机返回的样本中推断出 n 的值。不过，为了明确起见，输入中仍包含 n。我们用 $s = |r^*|$ 表示隐藏表示的大小。

定义 2.6。设 $\mathcal{A}$ 是一个学习算法。固定一个隐藏假设 c^* 以及其上的一个分布 $\{0,1\}^n$。设 h 是由 $\mathcal{A}$ 输出的假设，$c = \rho(h)$ 是 h 所表示的概念。我们定义

$$\mathrm{err}_h = \mathbb{P}_{x \sim \mathcal{D}_n}(c(x) \neq c^*(x))$$

为假设与隐藏概念在从 $\mathcal{D}_n$ 中抽取的样本上产生分歧的概率，即 h 在 $\mathcal{D}_n$ 下的所谓泛化误差。

如果算法 $\mathcal{A}$ 以至少 $1-\delta$. 的概率输出一个满足 err $_h \leq \varepsilon$ 的假设 h，则称该算法为概率近似正确（PAC）算法

通常，在这一框架中，如果学习问题能在多项式时间界内实现 PAC 可学习，就认为它是易处理的。更准确地说，如果存在一个针对 L 的 PAC 算法，其运行时间关于 n、s、$1/\varepsilon$ 和 $1/\delta$ 是多项式的，我们就称学习问题 L 是高效 PAC 可学习的。

参数化复杂度

为了将上述学习理论中的概念扩展到参数化情形，我们现在介绍参数化复杂性理论中一些后续将会用到的内容。我们遵循 Cygan 等人（2015）所著教材中的精彩阐述，并定义以下核心概念：

定义 2.7。参数化问题是一种语言 $L \subseteq \{0,1\}^* \times \mathbb{N}$。对于一对 $(x,k) \in \{0,1\}^* \times \mathbb{N}$, k 称为实例 (x,k) 的参数。(x,k) 的编码长度称为该实例的大小。

注释 2.8。参数化判定问题的定义可以很容易地扩展到搜索问题：在这种情况下，如果算法输出“是”，我们还需要输出一个见证。

定义 2.9。如果存在一个算法 $\mathcal{A}$、一个可计算的非递减函数 $f: \mathbb{N} \to \mathbb{N}$，以及一个多项式有界的非递减函数 $p(\cdot,\cdot): \mathbb{N}^2 \to \mathbb{N}$，使得对于输入 (x,k)，$\mathcal{A}$ 能够在 $f(k) \cdot p(n,k)$ 时间内正确判定 $(x,k) \in L$ 是否成立，则参数化问题 L 是固定参数可处理的。固定参数可处理问题的类记为 FPT。

注 2.10。虽然固定参数可处理性是参数化复杂性理论的核心，但下文定义的 XP 类刻画了一种较弱的可处理性概念；对于那些被认为不属于 FPT 的参数化问题，这种可处理性仍然值得追求。

定义 2.11. 若存在一个算法 $\mathcal{A}$，一个可计算的非递减函数 $f: \mathbb{N} \to \mathbb{N}$ 以及一个多项式有界的非递减函数 $p(\cdot,\cdot): \mathbb{N}^2 \to \mathbb{N}$，使得 $\mathcal{A}$ 能够正确判定对于输入 (x,k)，$(x,k) \in L$ 是否成立，且运行时间为 $p(n,k)^{f(k)}$，则称一个参数化问题 f 属于复杂度类 XP。

不太正式地说，XP 是这样一类问题：当参数取某个常数值时，可以在多项式时间内求解。XP 与 FPT 的区别在于，对 XP，我们允许该多项式（特别是它的次数）依赖于 k；而在 FPT 的定义中，对所有 k，多项式次数都固定不变。

为证明下界，我们将回顾参数化复杂性理论的基础知识，先从参数化归约的常见概念讲起。

定义 2.13。设 L、L' 是两个参数化问题。若算法 $\mathcal{A}$ 接收 L 的一个实例 (x,k) 并输出 L' 的一个实例 (x',k')，则称 $\mathcal{A}$ 是从 L 到 L' 的参数化归约，若

1. $(x,k) \in L$ 当且仅当 $(x',k') \in L'$，2. 对于某个函数 $g: \mathbb{N} \to \mathbb{N}$，$k' \leq g(k)$，以及 3. 对于某个可计算函数 $f: \mathbb{N} \to \mathbb{N}$，算法的运行时间上界为 $f(k) \cdot n^{\mathcal{O}(1)}$。

利用这个定义，我们可以给出参数化复杂度类 W[1] 和 W[2], 的临时定义，它们大致对应于通常的类 NP。虽然本文只会使用 W[2]，但为了完整起见，我们将给出 W[1] 的定义。

令 k-CLIQUE 表示这样一个问题：给定一个图 G 和一个参数 k，判断 G 是否具有大小至少为 k 的团。此外，我们令 k-HITTING SET 表示这样一个问题：给定一对 $(U,\mathcal{F})$ 和一个参数 k，其中 U 是形如 $U = \{1,\ldots,n\}$ 的全集，$\mathcal{F}$ 是关于 U 的集合族 $\{F_i\}_{i=1}^m$，判断是否存在一个大小至多为 k 的集合 H，使得对于所有 i 均有 $F_i \cap H \neq \emptyset$。

定义 2.14。类 W[1] 包含所有参数化问题 L，使得存在从 L 到 k-团的参数化归约。类 W[2] 包含所有参数化问题 L，使得存在从 L 到 k-命中集的参数化归约。此外，如果存在从 k-团（分别为 k-命中集）到 L' 的参数化归约，我们称问题 L' 为 W[1]-困难（分别为 W[2]-困难）。

在公认的复杂性假设下（Downey 和 Fellows，2013年），证明一个问题具有 W[1]-困难性或 W[2]-困难性，即可排除其具有固定参数可处理性。

3 参数化PAC学习

作为定义参数化PAC学习理论的第一步，我们需要考虑将使用的参数。在计算设置中，我们为每个实例关联一个参数，但在学习设置中并不存在这一概念——相反，学习算法需要处理隐藏的概念表示和隐藏的分布，并且我们允许二者都与参数化相关联。

定义 3.1（表示的参数化）。令 $\{\mathcal{R}_k\}_{k \in \mathbb{N}}$ 为一个映射，它为每个自然数 k 分配一个表示集合 $\mathcal{R}_k \subseteq \{0,1\}^*$，使得对于每个 $r \in \{0,1\}^*$，存在某个 k 满足 $r \in \mathcal{R}_k$，即：

$\bigcup_k \mathcal{R}_k = \{0,1\}^*$。我们将 $\{\mathcal{R}_k\}_{k\in\mathbb{N}}$ 称为表示的参数化。给定一个表示的参数化$\{\mathcal{R}_k\}_{k\in\mathbb{N}}$，我们通过如下定义，为单个表示 $r \in \{0,1\}^*$ 关联一个值 $\kappa_{\mathcal{R}}(r)$：

$$\kappa_{\mathcal{R}}(r) = \min\{k : r \in \mathcal{R}_k\}.$$

注 3.2. 与通常的参数化概念一致，我们假设 $\kappa_{\mathcal{R}}$ 是一个可计算函数。

例。设 ρ 是一种表示方案，它将一个（k 项析取范式公式的）二进制表示映射为其底层布尔函数。如果我们关注于学习 k 项析取范式，那么我们可能也希望将 k' 项析取范式视为 k 关于 $k' \le k$ 的析取范式。因此，我们令 $\mathcal{R}_k$ 为所有关于 $k' \le k$ 的 k' 项析取范式的集合。相应的映射 $\kappa_{\mathcal{R}}$ 随后将每个 k 项析取范式映射为 k 的值。

定义 3.3（分布和样本的参数化）。设 λ 是一个映射，对于每个 n，它为 $\{0,1\}^n$ 上的每个分布分配一个自然数，并且满足：对于 $\{0,1\}^n$ 上的任意两个分布 $\mathcal{D}$、$\mathcal{D}'$，如果 supp $\mathcal{D} \subseteq$ supp $\mathcal{D}'$，那么 $\lambda(\mathcal{D}) \le \lambda(\mathcal{D}')$。在这种情况下，我们称 λ 为分布的参数化。

我们将每个分布的参数化 λ 扩展到子集 $X \subseteq \{0,1\}^n$，方法是通过以下方式定义其对应的样本集参数化：

$$\lambda(X) = \min_{\mathcal{D} \,:\, X = \operatorname{supp}\mathcal{D}} \lambda(\mathcal{D}). \tag{3.1}$$

备注 3.4。注意，该定义意味着，等价地，$\lambda(X) = \min_{\mathcal{D}:\ X \subseteq \operatorname{supp}\mathcal{D}} \lambda(\mathcal{D})$；换句话说，$\lambda(X)$ 是从以 X 为其支撑集的分布中可以获得的最低参数。

本文讨论的所有参数化都仅取决于支撑集；然而，我们确实也希望明确允许取决于分布的表达能力更强的参数化。为了构建允许此类参数化的理论，有必要施加一个额外的技术条件，以确保使 λ 的参数值最小化的分布具有“良好性质”。

定义 3.5。我们称分布 $\mathcal{D}^*$ 在 λ 下对于 X 是典型的，如果 $\mathcal{D}^*$ 在公式 (3.1) 中取得最小值，即 $\lambda(X) = \lambda(\mathcal{D}^*)$ 且 $\operatorname{supp}\mathcal{D}^* = X$。

定义 3.6（L-可采样参数化）。设 $\mathcal{A}$ 是一个随机化算法，它接收输入 $X \subseteq \{0,1\}^n$，并在时间 $L(n, |X|, \lambda(X))$ 内输出一个概念 c，其中 $L(\cdot,\cdot,\cdot)$ 是某个非递减函数。若对于随机变量 C（该变量对应于 $\mathcal{A}$ 在$\mathcal{A}$ 所使用的全部随机比特上运行并输出 c的结果），以下条件成立，则称 λ 是 L-可采样的：存在一个分布 $\mathcal{D}^*$，它对于 X 在 λ 下是典型的，并且 $C \sim \mathcal{D}^*$ 成立，即 C 与 $\mathcal{D}^*$ 具有相同的分布。

备注 3.7。每个仅依赖于支撑集 X 的参数化 λ 都是线性时间可采样的，因为 X 上的均匀分布对于 X 在 λ下是典型的。更一般地说，我们认为每个“合理的”参数化都是可多项式时间采样的。

示例。例如，λ 的一个有效选择是这样一种映射：选择最大的数 k，使得存在一个概念 x 位于 $\mathcal{D}$ 的支撑集中，且具有 k 个非零项。显然，这一参数可以在概念集合上计算，正如条件所要求的那样，并且可以在线性时间内采样（相对于输入的大小而言，$n \cdot t$）。

定义 3.8（参数化学习问题）。参数化学习问题是一个学习问题 $(\mathcal{C}, \rho)$，以及一对称为其参数的对象 $(\{\mathcal{R}_k\}_{k\in\mathbb{N}}, \lambda)$，其中 $\{\mathcal{R}_k\}_{k\in\mathbb{N}}$ 是表示的参数化，λ 是分布的参数化。

此时，有必要说明上述定义与 Arvind、K¨obler 和 Lindner（2009）所建立的形式体系之间的关系。后者虽然能够表达诸如 k-合取函数这样的学习问题，其中 k 是参数，但无法处理样本空间的性质。例如，人们可能会对从样本中学习布尔公式感兴趣，而这些样本中被设为真的 k 变量数量有限。我们的框架通过将分布纳入参数化来刻画这一点。从概念层面看，这可以类比于普通参数化复杂性理论中存在的差异：一种是依据解的性质进行参数化，另一种则是依据输入实例的性质进行参数化。

定义 3.9（参数化学习算法）。对于参数化学习问题（$\mathcal{C}$、ρ、$\{\mathcal{R}_k\}_{k\in\mathbb{N}}$、$\lambda$），参数化学习算法是指根据定义 2.4，针对（$\mathcal{C}$、$\rho$）的学习算法。除 n、ε、δ 外，参数化学习算法还接收两个输入 k 和ℓ，并承诺它们满足 $k = \kappa_{\mathcal{R}}(r^*)$ 以及$\ell = \lambda(\mathcal{D}_n)$，且算法必须始终输出一个假设 h，满足 $\kappa(h) \in \mathcal{R}_k$。

注 3.10。要求隐藏假设和输出假设遵循相同的表示方案，就将我们的讨论限制在适当学习的情形。原则上，没有理由不将我们的框架扩展到非适当学习的情形，正如 Arvind、K¨obler 和 Lindner（2009）在其形式化工作中所做的那样。由于我们所有的示例讨论的都是适当学习，因此我们将定义限定于这种情形。

注 3.11。熟悉参数化复杂性理论的读者或许会注意到，这里定义的参数化学习问题取决于两个参数，而非单一参数。对于参数化算法，通常会通过定义一个新参数（如 $k + \ell$ 或 $\max\{k, \ell\}$），将多个参数k、ℓ 合并为一个。这样确实能简化记号，同时不会损失所得理论的表达能力。

我们将在本节末尾看到，就本文的目的而言，保留两个独立参数更为方便，这样才能在普通参数化算法与参数化学习算法之间建立有意义的联系。

令 poly(·) 表示其函数值可由自变量中的非递减多项式函数所界定的函数集合。此外，$\mathrm{fpt}(x_1, \ldots, x_t;\ k_1, \ldots, k_t)$ 和 $\mathrm{xp}(x_1, \ldots, x_t;\ k_1, \ldots, k_t)$ 分别表示其函数值由 $f(k_1, \ldots, k_t) \cdot p(x_1, \ldots, x_t)$ 和 $p(x_1, \ldots, x_t)_{f(k_1,\ldots,k_t)}$ 所界定的函数，

其中 f 是 $k_1, \ldots, k_t$ 中的任意非递减可计算函数，且属于 $p \in \text{poly}(x_1, \ldots, x_t)$。

定义 3.12（(T, S)-PAC 可学习性）。令 $T(n, s, 1/\varepsilon, 1/\delta, k, \ell)$ $S(n, s, 1/\varepsilon, 1/\delta, k, \ell)$ 为任意两个取整数值且关于其所有参数均为非递减的函数。

如果存在一个针对 $\mathcal{L}$ 的 PAC 学习算法，其运行时间为 $\mathcal{O}(T(n, s, 1/\varepsilon, 1/\delta, k, \ell))$，且向预言机查询的次数至多为 $\mathcal{O}(S(n, s, 1/\varepsilon, 1/\delta, k, \ell))$，则参数化学习问题 $\mathcal{L} = (\mathcal{C}, \rho, \{\mathcal{R}_k\}_{k\in\mathbb{N}}, \lambda)$ 是 (T, S)-PAC 可学习的。

我们将 (T, S)-PAC 可学习的参数化学习问题集合记为 $\text{PAC}[T, S]$。通过设置 $\text{PAC}[T, S] = \bigcup_{S\in\mathbf{S}} \bigcup_{T\in\mathbf{T}} \text{PAC}[T, S]$，可将其扩展到函数集合 S、T。

定义3.13。我们定义以下复杂性类：

$$\text{FPT-PAC}_{\text{time}} = \text{PAC}[\text{fpt}, \text{poly}], \tag{3.2}$$
$$\text{FPT-PAC} = \text{PAC}[\text{fpt}, \text{fpt}], \tag{3.3}$$
$$\text{XP-PAC}_{\text{time}} = \text{PAC}[\text{xp}, \text{poly}], \tag{3.4}$$
$$\text{XP-PAC} = \text{PAC}[\text{xp}, \text{xp}], \tag{3.5}$$

其中我们固定了

$$\text{poly} = \text{poly}(n, s, 1/\varepsilon, 1/\delta, k, \ell),$$
$$\text{fpt} = \text{fpt}(n, s, 1/\varepsilon, 1/\delta; k, \ell),$$
$$\text{xp} = \text{xp}(n, s, 1/\varepsilon, 1/\delta; k, \ell).$$

注3.14。除了刚刚定义的复杂性类之外，这里还可以考虑第五类：PAC[xp,fpt]. 不过，该类在本文所呈现的任何示例中都不起作用，因此我们将其留待未来研究。

图1概述了这些复杂性类及其相互关系。与XP一样，在学习设置中，XP-PAC类恰好包含这样的参数化学习问题：当参数固定为任意常数时，这些问题都变得可以高效地进行PAC学习。

例。 k-CNF和 k-DNF公式的学习（以 k为参数）属于XP-PAC类。另一方面，接下来的定理 5.6 和 6.2 给出了FPT-PAC$_{\text{time}}$类。Li 和 Liang（2018）中给出了一个在思想上属于FPT-PAC类的例子。最后，观察 6.5 提供了一个属于XP-PAC$_{\text{time}}$ 的学习问题示例。

一致性检查：理论之间的联系

在确立了基本的复杂性类之后，我们转而探讨如何建立新近发展起来的理论与如今已广为确立的、用于判定问题的参数化复杂性范式之间的联系。关键的是，这些联系将使我们能够在学习设置中也利用后者的算法上界和下界。

定义 3.15（参数化一致性检查）。对于每个参数化学习问题 $\mathcal{L} = (\mathcal{C}, \rho, \{\mathcal{R}_k\}_{k\in\mathbb{N}}, \lambda)$ 以及每个函数 $f(n, t, k, \ell)$，我们按如下方式关联一个参数化搜索问题 f-CONSISTENCY($\mathcal{L}$)：

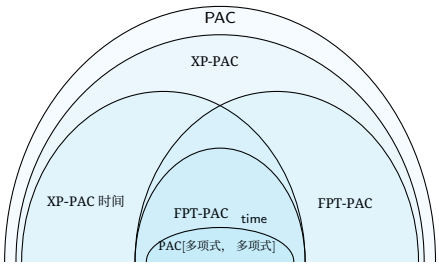


图1：定义3.13中所定义的参数化学习类示意图。

1. 其输入是一个带标签样本列表 $((x_1, a_1) \ldots, (x_t, a_t))$，其中对所有 i 都有 $x_i \in \{0,1\}^n$，且 x_i 两两不同，同时还有 $a_i \in \{0,1\}$。2. 其参数为 $k \in \mathbb{N}$，作为输入的一部分给出，以及 $\ell = \lambda(\{x_1, \ldots, x_t\})$。3. 任务是判断是否存在一个表示 r，使其具有 $\kappa(r) \in \mathcal{R}_k$，其中 $|r| \le f(n, t, k, \ell)$，并且对于概念 $c = \rho(r)$，对所有 i 都成立 $c(x_i) = a_i$。

我们将这个参数化问题称为与 $\mathcal{L}$ 和 f相关联的参数化一致性检查问题。

忽略定义3.15中的所有参数，就得到通常所说的一致性检查问题（Pitt 和 Valiant（1988））。函数 f 需要为所要寻找的表示提供一个明确的上界；事实上，虽然每个学习实例都带有一个大小确定的隐藏表示，但一致性检查问题并非如此，因此我们需要让运行时间界将表示的大小考虑在内。

众所周知，假设假设空间并非过大，则学习问题是 PAC 可学习的，当且仅当相应的判定问题可在随机多项式时间内求解。现在我们发现，在参数化情形下也存在类似的等价关系。

引理3.16. 设 $\mathcal{L} = (\mathcal{C}, \rho, \{\mathcal{R}_k\}_{k\in\mathbb{N}}, \lambda)$ 是一个参数化学习问题，且 $f(n, t, k, \ell)$ 是一个函数。如果 $\mathcal{L}$ 是 (T, S)-PAC 可学习的，且 λ 是 L- 可采样的，那么存在一个随机化算法：对于输入带标签的样本 $\{(x_1, a_1) \ldots, (x_t, a_t)\}$ 以及参数 k, ℓ，满足 $\lambda(\{x_1, \ldots, x_t\}) = \ell$ 时，以至少 $1 - \delta$ 的概率返回一个大小为 $|h| \le s = f(n, t, k, \ell)$ 的一致假设$h \in \mathcal{R}_k$（如果该假设存在），并在 $\mathcal{O}(T(n, s, t, \delta, k, \ell) + L(n, t, \ell) \cdot S(n, s, t, \delta, k, \ell))$ 时间内运行。

证明。设 $((x_1, a_1) \ldots, (x_t, a_t))$ k, ℓ 是与 $\mathcal{L}$ 相关的参数化一致性问题的某些输入，使得 $\lambda(\{x_1, \ldots, x_t\}) = \ell$。我们设定 $\varepsilon = \frac{1}{t+1}$，并取 δ 作为所需的成功概率。

我们使用输入为 $n = |x_1| = \ldots = |x_t|$、$\varepsilon$ 和 δ 的 (T, S)-PAC 算法。在该算法执行期间，我们通过调用由 λ 保证存在的采样算法，模拟针对隐藏分布的预言机。

根据定义，PAC 学习算法将返回一个来自 $\mathcal{R}_k$ 的假设，该假设与至多占样本 $1/(t+1)$- 比例的样本不一致，因此以至少 $1-\delta$ 的概率与所有样本一致（如果这样的假设存在），我们将输出该假设。否则，我们输出任意假设。这就得到了一个针对参数化一致性检查问题的随机化算法，并满足成功概率所要求的界。

关于运行时间，注意我们按照引理陈述中的参数调用学习算法，该算法需要 $T(n,s,t,\delta,k,\ell)$ 步；对于学习算法进行的每次 $S(n,s,t,\delta,k,\ell)$ 次预言机调用，我们都需要花费 $L(n,t,\ell)$ 时间，从分布中生成一个样本。注意，如果 PAC 算法耗费超过 $T(n,s,t,k,\ell)$ 步（或抽取超过 $S(n,s,t,k,\ell)$ 个样本），我们可以安全地终止其执行，并返回任意一个假设[2]。 □

定理 3.17. 设 $\mathcal{L}=(\mathcal{C},\rho,\{\mathcal{R}_k\}_{k\in\mathbb{N}},\lambda)$ 是一个参数化学习问题，且 $f(n,t,k,\ell)$ 是一个函数。

如果 $\mathcal{L}$ 属于 FPT-PAC，$f\in \mathsf{fpt}(n,t;\,k,\ell)$，并且对于某个 $S\in \mathsf{fpt}(n,t;\,\ell)$，$\lambda$ 是 S-可采样的，那么与 $\mathcal{L}$ 和 f 相关联的参数化一致性检查问题 f-Consistency($\mathcal{L}$) 属于 FPT。

类似地，如果 $\mathcal{L}$ 属于 XP-PAC，$f\in \mathsf{xp}(n,t;\,k,\ell)$，并且对于某个 $S\in \mathsf{xp}(n,t;\,\ell)$，$\lambda$ 是S-可采样的，那么与 $\mathcal{L}$ 和 f 相关联的参数化一致性检查问题 f-Consistency($\mathcal{L}$) 属于 XP。

证明。通过将适当的函数代入引理3.16中的T、S 和 L 即可得到。特别注意，根据 FPT-PAC 和 XP-PAC 的相应定义，T和 S中对 $s=f(n,t,k,\ell)$ 的依赖仅允许具有 $g(k,\ell)\cdot\mathsf{poly}(n,s,1/\varepsilon,1/\delta)$ 或 $\mathsf{poly}(n,s,1/\varepsilon,1/\delta)^{g(k,\ell)}$ 的形式。因此，如果 $f\in\mathsf{fpt}(n,t;\,k,\ell)$，那么 $\mathsf{poly}(n,f(n,t;\,k,\ell)\,1/\varepsilon,1/\delta)$ 仍由$g'(k,\ell)\cdot\mathsf{poly}(n,t,1/\varepsilon,1/\delta)$所界定；对于 $f\in\mathsf{xp}(n,t;\,k,\ell)$，同样如此。 □

引理 3.18. 设 $\mathcal{L}=(\mathcal{C},\rho,\{\mathcal{R}_k\}_{k\in\mathbb{N}},\lambda)$ 是一个参数化学习问题，并设 $\mathcal{H}_{n,k}=\mathcal{R}_k\cap\rho^{-1}(\mathcal{C}_n)$ 是 $\mathcal{C}_n$ 中概念在 $\mathcal{R}_k$ 下、属于 ρ 的所有表示构成的集合。假设对于一致性检查问题 $\log|\mathcal{H}_{n,k}|$-Consistency($\mathcal{L}$)，存在一个运行时间为 $T(n,t,\delta,k,\ell)$ 的确定性算法。那么，$\mathcal{L}$ 是 (T',S)-PAC 可学习的，其中[3]

$$S(n,s,1/\varepsilon,1/\delta,k,\ell)=\frac{1}{\varepsilon}(\log|\mathcal{H}_{n,k}|+\frac{1}{\delta})),$$
$$T'(n,s,1/\varepsilon,1/\delta,k,\ell)=T(n,s,\frac{1}{\varepsilon}(\log|\mathcal{H}_{n,k}|+\frac{1}{\delta}),k,\ell).$$

证明。学习算法从该分布中抽取 $t=\frac{1}{\varepsilon}(\log|\mathcal{H}_{n,k}|+\frac{1}{\delta})$ 个样本 $(x_1,a_1)\ldots,(x_t,a_t)$：

[2]更精确地说，这些函数必须满足与某个时间可构造函数渐近等价的标准。由于在任何自然情形下这完全无关紧要，我们选择忽略这些细节。[3]对 s 的依赖消失了，因为 s 本身受 n 和 k 的某个函数所界定。

未知分布 $\mathcal{D}$，并使用这些t 样本和参数 k 调用算法。它输出算法返回的假设 h，该假设保证属于 $\mathcal{R}_k$，并满足 $|h|\leq\log|H_{n,k}|$。通过并集界的标准论证，所得算法确实是一个参数化 PAC 算法。

对于这次算法调用，请注意，根据定义，$\lambda(\{x_1,\ldots,x_t\})\leq\ell$。回想定义 3.12，我们假设 T 和 S 都是非递减的。因此，根据假设，这至多需要 $T(n,\frac{1}{\varepsilon}(\log|\mathcal{H}_{n,k}|+\frac{1}{\delta})\;k,\ell)$ 的时间，并使用 $t=\frac{1}{\varepsilon}(\log|\mathcal{H}_{n,k}|+\frac{1}{\delta})$ 个样本。 □

定理 3.19。设 $\mathcal{L}=(\mathcal{C},\rho,\{\mathcal{R}_k\}_{k\in\mathbb{N}},\lambda)$，并将 $\mathcal{R}_k$ 中 $\mathcal{C}_n$ 在 ρ 下的表示集合记为 $\mathcal{H}_{n,\,k}$。

如果参数化一致性检查问题 $\log|\mathcal{H}_{n,\,k}|$-Consistency($\mathcal{L}$) 属于 FPT，且 $\log|\mathcal{H}_{n,\,k}|\in\mathsf{fpt}(n;k)$，那么 $\mathcal{L}$ 属于 FPT-PAC$_{\text{time}}$。

类似地，如果参数化一致性检查问题 $\log|\mathcal{H}_{n,\,k}|$-Consistency($\mathcal{L}$) 属于 XP，且 $\log|\mathcal{H}_{n,\,k}|\in\mathsf{xp}(n;k)$，那么 $\mathcal{L}$ 属于 XP-PAC$_{\text{time}}$。

证明。由引理 3.18 得出，随后代入相应的运行时间函数以及关于 $\mathcal{H}_{n,\,k}$ 的界；这些界特别限制了任何可能返回的假设的长度。 □

前述定理的重要性在于，它们分别将一致性检查的参数化算法上界和下界转化为参数化学习问题的上界和下界。

也就是说，定理3.17表明，如果一个参数化一致性检查问题可以由参数化算法高效求解，那么它所属的参数化学习问题也可以高效求解。这一点将在定理5.6、6.2中加以利用。

另一方面，定理 3.19指出，若一个参数化学习问题存在高效算法，那么相应的参数化一致性检查问题也存在高效算法。反过来看，一致性检查问题的下界意味着学习问题也存在下界。这一点将在定理5.1、6.4中加以利用。

4 参数化学习的相关研究

介绍这一新框架后，我们将专门用一节讨论它与少数探讨学习理论参数化方法的论文之间的联系，以及它们之间的差异。

按时间顺序来看，参数化复杂性理论与学习理论之间最早的联系，早在参数化复杂性的先驱者们（Downey、Evans 和 Fellows，1993 年）那里就已建立。他们不仅建立了参数化学习模型，还将学习复杂性与研究充分的图问题联系起来；但与我们的工作不同，他们通过扩展等价查询研究精确学习。G¨artner 和 Garriga（2007年）研究了包括 PAC学习在内的各种学习模型中学习有向割的复杂性。虽然他们的相应结果属于标准多项式时间PAC学习，但值得注意的是，他们给出的改进的学习界是关于图的

结构参数的。关于学习 k-项析取范式公式，已有研究给出了优于平凡界的指数运行时间界，例如，$n^{\tilde{O}(\sqrt{n}\text{对数 }k)}$（Alekhnovich 等人 2008 年）。在上一节中，我们已经将我们的框架与 Arvind、K¨obler 和 Lindner（2009 年）的早期工作进行了比较；该工作只识别出了 FPT-PAC 类。

Li 和 Liang（2018）关于学习线性回归混合模型的参数化算法的较新工作，严格来说并不属于我们的参数化PAC学习框架，因为他们的误差参数是到隐藏概念的距离，而不是在任意分布下的错误标记概率。不过，他们的样本复杂度界是 FPT 的；当参数不受限时，该界会变为无穷大。这也是 PAC学习结果的一个重要例子：该结果仅对某个特定的分布族成立（这一方法也包含在我们的框架中；参见定义 3.6 背后的直觉）。最后，van Bergerem、Grohe 和 Ritzert（2022）研究了一阶逻辑学习的参数化复杂度；他们通过参数化归约证明了困难性结果，同时还通过一致性检查实现了 FPT 时间的学习，这与我们的框架十分契合。

5 参数化析取范式和合取范式学习

本节的大部分内容都考虑在各种参数化方式下学习隐藏 DNF 公式的问题，这些参数化方式将在本节中形式化定义。我们首先注意到，在任何一种在公式否定下保持不变的参数化方式下，学习 CNF公式的问题都等价于学习 DNF 公式：事实上，只需将输入取否定后交给 DNF学习算法，再将输出公式取否定，即可得到 CNF 学习算法，反之亦然。因此，我们的结果同样适用于学习 CNF公式，下面不再对此作明确说明。

令 LEARNING DNF 为学习问题 $(\mathcal{C}, \rho)$，其中 $\mathcal{C}$ 是与析取范式公式对应的布尔函数集合，ρ 是公式项的直接表示。为了定义该问题的一个基本变体，其中目标公式的项数受到限制，令 LEARNING k-TERM DNF 为参数化学习问题 $(\mathcal{C}, \rho, \{\mathcal{R}_k\}_{k\in\mathbb{N}}, \lambda)$，其中 $\mathcal{C}$ 和 ρ 如上所述，κ 将公式的表示映射到其中的项数，而 λ 是任意常数参数化。然而，众所周知，即使是学习 2 项析取范式的一致性检查问题也是 NP-困难的（Alekhnovich 等人 2008 年），因此，根据定理 3.17 的一个非参数化变体，除非 P =NP，否则 LEARNING k-TERM DNF 不属于 XP-PAC$_{\text{time}}$。因此，对于这种参数化，不应期待任何可处理性结果。

对 LEARNING DNF 进行参数化的另一种自然方式，是限定一个项的最大长度。具体而言，令 LEARNING k-DNF 表示参数化学习问题（$\mathcal{C}$，ρ，$\{\mathcal{R}_k\}_{k\in\mathbb{N}}$，$\lambda$），其中 $\mathcal{C}$ 和 ρ 与 LEARNING DNF 中的定义相同，$\kappa = \kappa_{\mathcal{R}}$ 将公式的表示映射为其中一个项的最大长度，而 λ 是平凡的。众所周知，对于每个固定的 k，LEARNING k-DNF 可通过穷举论证高效地进行 PAC 学习，因此该问题立即属于 XP-PAC$_{\text{time}}$，其参数为 k（Mohri、Rostamizadeh 和 Talwalkar，2012 年，例 2.9）。另一方面，我们证明，在标准参数化复杂性假设下，这一学习结果是紧的，将 Arvind 等人（Arvind、K¨obler 和 Lindner，2009 年）针对 k-合取函数和 – 单项式的困难性归约进行了扩展。

定理 5.1. 假设 W[2] $\neq$ FPT，LEARNING k-析取范式和 LEARNING k-合取范式不属于 FPT-PAC$_{\text{time}}$。

证明。我们对 LEARNING k-合取范式证明该结果。假设结论不成立，则根据定理 3.17，存在一个用于一致性检查问题的 FPT 算法。现在我们证明：即使 f 关于 n 和 k 线性，仍存在一个从 k-命中集到 LEARNING k-合取范式一致性检查问题的参数化归约，由此得到矛盾。

考虑 k-命中集的一个实例 $(U, \mathcal{F})$。对于每个 $i \in [t-1]$，设置 $n = |U|$、$t = |\mathcal{F}| + 1$，当且仅当全集的第 j 个元素属于 $\mathcal{F}$ 的第 i 个集合，并且 $x_i[j] = 1$。最后，将 x_t 设置为零向量，并且 $a_t = 0$。

首先，考虑一个 k-命中集 $S \in U$。考虑一个单子句 k-合取范式公式 φ，其中包含与集合 S相对应的正文字。对于每个 $i \in [t-1]$，$\varphi(x_i) = 1 = a_i$，因为 S 与 $\mathcal{F}$的第 i个集合相交，而对应的 x_i 元素被设为 1。此外，$\varphi(x_t) = 0 = a_t$，因为 φ不包含负文字。

反过来，考虑一个能正确标记所有样本的 k-合取范式公式 φ。由于 $\varphi(x_t) = a_t = 0$，φ 中存在一个只包含正文字的子句 C。对于每个 $i \in [t-1]$，$\varphi(x_i) = a_i = 1$，因此 $C(x_i) = 1$，于是 C中存在一个正文字，其变量对应于 $\mathcal{F}$的第 i个集合中的某个元素。因此，由与 C 中的变量相对应的元素组成的集合 S 是 $\mathcal{F}$的一个 k-命中集。□

上述内容涵盖了假设类最自然的两种参数化方式——项数以及目标公式中项的最大大小——然而，我们尚未涉及样本空间的参数化。作为一个简单的步骤，请注意，当每个赋值至多将一个变量赋为“真”时，一致性问题很容易解决；在这种情况下，总是可以构造出一个单项析取范式。另一方面，先前已知的着色归约（Pitt 和 Valiant（1988））表明，即使每个赋值至多将两个变量赋为“真”，该问题也会变为 NP-困难问题。在下文中，我们分析介于这两种情况“之间”的学习 k-项析取范式的可处理性，并展示以下基于一种“平凡性后门”概念进行参数化的积极结果（Gaspers 和 Szeider（2012）；Semenov 等（2018））。

引理5.2。LEARNING k-TERM DNF 关于参数 $k+s$ 是 FPT-PAC$_{\text{time}}$ pa-化的，其中 s 是满足以下条件的变量集合 S 的最小大小：

- 每个赋值在 S 外最多将一个变量赋为“真”，并且
- 对于其余每个变量，至多有一个赋值将其设为“真”。

证明。根据定理 3.19，只需针对 $f = \text{fpt}(n; k+s)$ 的一致性检查问题提供一个 FPT 算法；特别地，只需论证我们可以用这样的 f 来界定所有可能假设的长度。

将给定的带标签样本集合 $(x_1, a_1), \ldots, (x_t, a_t)$ 划分为 b 个类，使同一类中的所有赋值在 S 上一致：$\{(x_i, a_i)\}_{i\in[t]} = \{(x_i^1, a_i^1)\}_{i\in[t_1]} \cup \cdots \cup \{(x_i^b, a_i^b)\}_{i\in[t_b]}$，其中对每个 i，$i' \in [t_j]$，$x_i\big|_S = x_{i'}^j\big|_S$。现在我们证明，除非每个类的大小至多为 k，否则总能将该实例约简为一个更小的实例。接下来的两个断言给出了确切的约简规则；第一个断言表明，当同一类中的是实例过多时，可以安全地删除其中一个。

断言 5.3. 设 $x_1, .., x_t$ 是 $[n]$ 的赋值，$a_1, \ldots, a_t$ 是相应的标签。设 $t \geq k+2, S \subset [n]\backslash[k+2]$，并且对每个 $i \in [k+2]$，$a_i = 1, x_i$ 将 i 赋为“真”，并将 $[n] \setminus S$ 中的所有其他变量赋为“假”。设除 x_1 之外，$x_1, ..., x_t$ 中没有任何赋值将 1 赋为“真”。如果存在一个 k 项析取范式公式 φ 与 $(x_i, a_i)_{i=2}^t$ 一致，那么也存在一个 k 项析取范式公式 ψ 与 $(x_i, a_i)_{i=1}^t$ 一致。

证明。一般而言，设 $\varphi = T_1 \vee \cdots \vee T_k$。如果 T_1、.....、T_k 各自都包含一个正文字，根据抽屉原理，存在一个来自 x_2、.....、x_{k+2} 的赋值 x_i，使其不满足 φ。因此，存在一个仅包含负文字的合取项 T_j。如果 T_j 不包含变量 1 的否定，$\varphi(x_1) = 1 = \lambda_1$，因此令 $\psi = \varphi$ 即可完成证明。否则，取 ψ 为从项 T_j 中移除变量 1 的否定后得到的 φ。容易看出，$\psi(a_1) = 1 = \lambda_1$。此外，$a_2$、.....、$a_t$ 均不会改变赋值的值，因为缩小项 T_j 不会使是实例变得不满足，而没有任何否实例会因此变得满足，因为只有 x_1 会将变量赋为“真”。■

下一个命题使我们能够简化负样本。

命题 5.4. 设 x_1，...，x_t 是 $[n]$ 的赋值，a_1，...，a_t 是相应的标签。设 $t \geq 2$、$S \subset [n]\backslash\{1\}$，并设 $a_i = 0$、x_1 将变量 1 赋值为“真”，将 $[n]\backslash S$ 中的所有其他变量赋值为“假”。除 x_1 外，设 x_1，...，x_t 中没有任何赋值将 1 赋值为“真”。如果存在一个与 $(x_i, a_i)_{i=2}^t$ 一致的 k-项析取范式公式 φ，那么也存在一个与 $(x_i, a_i)_{i=1}^t$ 一致的 k-项析取范式公式 ψ。

证明。设 ψ 是由 φ 得到的公式，具体做法是在每个项中附加变量 1 的否定。对于每个 $i \in [t] \setminus \{1\}$，赋值 x_i 保持其值不变，因为它将 1 赋值为“假”。另一方面，赋值 a_{t+1} 显然不满足所构造的公式。■

最后，前两个断言减少了需要考虑的样本数量，而下一个断言则通过移除那些被赋予相同值的变量来减少变量数量。

命题 5.5. 设 x_1、.....、x_t 是 $[n]$ 的赋值，a_1、.....、a_t 是相应的标签。假设变量 1 和 2 在赋值 x_1、.....、x_t 中取值相同。存在一个定义在变量 2、.....、n 上的 k 项析取范式公式 φ，与 $(x_i, a_i)_{i=1}^t$ 一致，当且仅当存在一个定义在变量 1、.....、n 上的 k 项析取范式公式 ψ，与 $(x_i, a_i)_{i=1}^t$ 一致。

证明。一方面，令 $\psi = \varphi$，显然可满足该断言。另一方面，令 φ 为公式 ψ，其中每一项中的变量 1 的文字都替换为变量 2 的相同文字。由于在 x_1、...、x_t 中，没有赋值能区分 1 和 2，因此对于每个 $i \in [t]$，$\psi(x_i) = \varphi(x_i)$。■

现在，我们使用断言 5.3、命题 5.4 和断言 5.5，构造一个与原实例等价的实例，其中至多有 $2^s \cdot (k+2)$ 个赋值-标签对，且涉及至多 $s + 2^s \cdot (k+2)$ 个 $+1$ 变量。算法只需反复应用断言 5.3、命题 5.4 和断言 5.5，直到没有断言可以应用为止；过程中可能需要重新编号赋值和变量。我们首先断言，处理后每个类中至多有 $(k+2)$ 个赋值。否则，该类中要么存在一个负标签赋值，它将 S 之外的某个变量赋为“真”，此时可应用命题 5.4；要么该类中至少有 $(k+2)$ 个正赋值，此时可应用断言 5.3，这就产生了矛盾。因此，构造出的实例中至多有 $2^s \cdot (k+2)$ 个赋值，正如所需。

关于变量，请注意，除 S 外，每个赋值至多将一个变量赋为“真”。因此，至多有 $s + 2^s \cdot (k+2)$ 个变量至少被赋为一次“真”，这意味着所有其他变量都只被赋为“假”。由于命题 5.5 不适用，至多只能有一个这样的变量。

现在，算法尝试对剩余变量的所有可能的项，并检查每个项是否不满足任何负标记赋值。然后，算法尝试所有由此类项组成的 k-项析取范式公式，并查看其中是否有公式满足所有正标记赋值。

关于运行时间，命题 5.3、5.4 和 5.5 的应用显然都是多项式时间的。在约简实例中，至多有 $s + 2^s \cdot (k+2)$ 个 $+1$ 变量，因此至多有 $3^{s+2^s\cdot(k+2)+1}$ 个可能的项，以及至多有 $3^{k\cdot(s+2^s\cdot(k+2)}$ 个不同的 k-项析取范式公式。这将假设空间的大小限制为 $\text{fpt}(n; k+s)$。□

通过应用上述从析取范式到合取范式的标准变换，我们可以得到一个类似的表述，但其中每次出现的“真”都被反转为“假”。另一种标准变换则是将每个样本中的每个赋值以及隐藏假设中的每个文字都取反，由此可以为问题和琐凡性概念的四种可能组合都得到相同的结果。

定理 5.6。$\text{L}_{\text{学习}}$ k-$\text{C}_{\text{子句}}$ 合取范式和 $\text{L}_{\text{学习}}$- k-$\text{T}_{\text{项}}$ 析取范式都是 $\text{FPT-PAC}_{\text{time}}$，其参数为 $k+s$，其中 s 是变量集合 S 的最小大小，使得分布的支集满足以下条件：

- 每个赋值至多将 S 之外的一个变量赋为“假”（分别为“真”），并且
- 对于每个剩余变量，至多有一个赋值将其赋为“假”（分别为“真”）。

6 图上的学习

设 $H_1, \ldots, H_p$ 是一个固定的图族。设 $\mathcal{H}$ 是一个不包含任何 $H_1, \ldots, H_p$ 作为诱导子图的图类。在经典的 $\mathcal{H}$-顶点删除问题中，给定一个图 G 和一个参数 k，任务是判断是否存在一个顶点子集 $S \subset V$（G），其大小至多为 k，使得 $G - S$ 属于 $\mathcal{H}$。按照标准方式构造后，该问题的一致性版本接收一列定义在同一顶点集 V 上的图 $G_1, \ldots, G_t$，以及相应的标签序列 $\lambda_1, \ldots, \lambda_t$ 作为输入；任务是找出一个大小至多为k 的子集 $S \subset V$，使得 $G_i - S \in \mathcal{H}$ 当且仅当 $\lambda_i = 1$。我们将相应的学习问题称为学习 $\mathcal{H}$-删除集。

特别地，当禁用族由单个图组成时，K_2，$\mathcal{H}$ 是空图类，而 $\mathcal{H}$-顶点删除等价于顶点覆盖。如果该族为P_3，问题就变为簇图顶点删除——寻找一个要删除的顶点子集，使图转变为簇图，即团的不相交并。学习$\mathcal{H}$-删除集因此同时推广了学习顶点覆盖和学习簇图删除集：前者的任务是学习一个隐藏的顶点覆盖，后者则是学习一个使图变为簇图的删除集。

现在让我们将其明确联系到上文建立的参数化学习形式化框架。在此语境下，我们将元素 $x \in \{0,1\}^n$ 解释为一个具有 N 个顶点的图的邻接矩阵，其中 $n = N^2$。概念 $c : \{0,1\}^n \to \{0,1\}$ 由 N个顶点中的一个子集 S 表示。值 $c(x)$表示 S 是否是邻接矩阵为 x的图的 $\mathcal{H}$-删除集。如前所述，c 的表示方案将子集 S 映射为 c，正如上文所描述的。因此，$\mathcal{C}_n$ 对应于所有N顶点图上的 $\mathcal{H}$-删除集。我们通过令 $\mathcal{R}_k$ 为所有大小至多为 k的顶点子集构成的集合来进行参数化，并令分布携带平凡的常参数化 $\lambda(\mathcal{D}) = 1$。

请注意，对于图问题，假设的长度总是自然地在线性范围内受 n 限制，因此我们将省略在一致性检查中对 f 的明确引用。

接下来，我们证明，用于 $\mathcal{H}$-顶点删除的著名固定参数可处理时间算法（Cygan 等人（2015））可以扩展到相应的一致性问题，这意味着由有限个禁用诱导子图族刻画的广泛图类是可在固定参数可处理时间内学习的。

引理 6.1。设 $\mathcal{H}$ 是一个禁止诱导 H_1、.....、 H_p 的图类。那么，与 Learning $\mathcal{H}$-Deletion Set 相关联的参数化一致性检查问题是固定参数可处理的。

证明。给定 $(x_1, a_1) \ldots, (x_t, a_t)$ 和 k，并令 $G_1, \ldots, G_t$ 为相应的图，每个图有 N 个顶点。记 $\mathcal{Y}$ 为满足 $a_i = 1$ 的 G_i 集合，记 $\mathcal{N}$ 为满足 $a_i = 0$ 的那些集合。我们需要找到一个至多包含 k 个顶点的子集 S，使得对于所有 $G \in \mathcal{Y}$，$G - S \in \mathcal{H}$ 成立；并且对于所有 $G \in \mathcal{N}$，$G - S \notin \mathcal{H}$ 成立。首先注意到，由于 $\mathcal{H}$ 是通过禁用诱导子图定义的，用仍满足第一个条件的 S 子集替换 S，也总能继续满足第二个条件。因此，只需寻找包含意义下极小的子集，使其满足对于所有 $G \in \mathcal{Y}$，$G - S \in \mathcal{H}$ 成立，然后检查对于所有 $G \in \mathcal{N}$，$G - S \notin \mathcal{H}$ 是否确实为真。

为了生成这样一个大小至多为 k 的极小集 S，我们实际上将在固定参数时间内枚举出所有这类集合，方法是扩展枚举极小顶点覆盖的常规有界搜索树方法。从一个空解 $S = \emptyset$ 开始，我们遍历 $\mathcal{Y}$ 中的图。

通过暴力枚举，我们检查其中是否存在任意一个 H_i 的诱导副本。回想一下，图 H_i 的数量为常数，且每个图的大小均为常数，例如 $q = \max_i |V(H_i)|$，因此可以在 $O(pN^q)$时间内完成，这关于 N是多项式时间。

如果该过程发现了 H_i的诱导副本，那么任何 $\mathcal{H}$-删除集都必须删除该诱导副本在 G 中占据的至多 q 个顶点之一（如果此时 $|S| = k$，则丢弃 S）。我们的枚举算法针对这些 q 种选择分别进行分支，并将相应的顶点加入 S

另一方面，如果不存在任何H_i的诱导副本，那么 S 就是所有 $G \in \mathcal{Y}$ 的一个最小 $\mathcal{H}$-删除集，因此我们检查 $G - S \notin \mathcal{H}$ 是否为真；若是，则输出 S。

如前所述，该方法是正确的。至于其运行时间，注意我们构建了一棵深度有界为 k、分支因子至多为 q的搜索树，并在递归的每一步花费至多 $O(pN^q)$时间，因此总运行时间至多为 $O(q^k pN^q)$，关于 k是固定参数可处理的。 □

定理 6.2. 设 $\mathcal{H}$ 是一个禁止诱导 H_1、.....、 H_p 的图类。Learning $\mathcal{H}$-Deletion Set 是 FPT 时间 PAC 可学习的，其参数为删除集的大小。

证明。由引理 6.1 和定理 3.19 可得。 □

相比之下，我们现在说明，另一个经过充分研究的删除问题族很可能不具备与上述结果类似的正面结论。具体而言，对于一个图 H，令 $\mathcal{H}$ 为一个不含 H minor 的图类。特别地，当 $H = K_3$ 时，$\mathcal{H}$-VERTEX DELETION 等价于反馈顶点集。众所周知，当以待删除顶点集的大小为参数时，该问题允许使用 FPT 时间算法（Cygan 等人（2015））；然而，我们将证明，与学习反馈顶点集相关的一致性检查问题是 W[2]-难的。也就是说，除非参数化复杂性类 FPT 与 W[2] 重合，否则一致性问题不属于 FPT。结论 FPT = W[2] 是 P = NP 的参数化对应物，被认为极不可能成立。

引理 6.3。L学习-反馈 V顶点 S集合 的一致性检查问题是 W[2]-困难的，即使输入中只有是实例。

证明。我们从 k-命中集归约。再次说明，输入是一个二元组（U，$\mathcal{F}$），其中 $\mathcal{F}$ 是由子集 $\{F_i\}_{i=1}^m$组成的族 $\mathcal{F}$，满足对所有 i都有 $F_i \subseteq U$，且 $U = \{1, \ldots, n\}$。任务是生成一个大小至多为 k 的集合 $H \subseteq \{1, \ldots, n\}$，使得对所有 i都有$F_i \cap H \neq \emptyset$，其中 k 是参数。已知该问题是 W[2]-困难的（Downey 和 Fellows，1999 年）。

我们按如下方式构造 FEEDBACK VERTEX SET 的一致性检查问题实例：对于满足 $i = 1$的实例（x_i，a_i），其中 m 为 $a_i = 1$，对所有 i均成立；每个 x_i 都是一个图 G_i，其顶点为 $\{1, \ldots, n\}$，并且其边在顶点集 F_i上构成一个环。$\{1, \ldots, n\}$ 中大小为 k 的每个顶点子集，若是所有 G_i 的同时反馈顶点集，就必须至少命中每个 F_i 一次，因此是一个大小至多为 k的命中集。反过来，所有 G_i 的每个同时反馈顶点集都必须破坏每个环，因此对于所有 i，都至少包含F_i 中的一个元素。

因此，存在一个大小至多为 k、与所有样本一致的反馈顶点集，当且仅当原实例中存在一个大小至多为 k 的命中集，因此，一致性检查 FEEDBACK VERTEX SET 是 W[2]-困难的。□

定理 6.4. 除非 FPT $=$ W[2], , 否则学习问题 LEARNING FEEDBACK VERTEX SET 不属于 FPT-PAC。

证明。由引理 6.3 和定理 3.17 得出。□

另一方面，LEARNING FEEDBACK VERTEX SET 的一致性检查问题显然属于 XP。由于 $\log |\mathcal{H}_{n,\ k}|$ 显然以上界 $\mathcal{O}(n)$ 为界，由定理 3.19 可立即得到：

观察 6.5. LEARNING FEEDBACK VERTEX SET 属于 XP-PAC time。

同样的论证也适用于学习许多其他图结构，例如支配集和独立集。

7 结论

在过去二十年中，参数化复杂性范式可以说彻底改变了我们对整个计算机科学领域中计算问题的理解。我们坚信，以更加细粒度的参数化视角审视学习问题，同样能够揭示大量迄今仍隐藏不见的研究问题与潜在突破。本文为研究人员提供了开始严谨探索这一新颖研究方向所需的工具和概念，同时也为搭建一座桥梁奠定了理论基础，这座桥梁将连接学习理论与时间复杂性领域的研究群体。或许值得注意的是，迄今为止，这些研究群体仍相当彼此隔绝；它们之间为数不多的互动，都发生在致力于人工智能研究的会议上。

不同于典型的复杂性理论论文，我们认为本文最重要的贡献在于理论构建部分，即第 3 节。的确，尽管我们用来展示该理论的具体算法和下界本身并非无足轻重，而且颇具意义，但要确保理论基础彼此协调、足够宽泛以涵盖未来可能出现的参数化，同时还要支撑与参数化一致性检查问题之间的关键联系，是一项极具挑战性的任务。

本文提出的参数化 PAC 学习框架为未来研究开辟了许多方向。例如，PAC[xp,fpt] 类中存在一些有趣且非平凡的参数化学习问题。

致谢

作者感谢奥地利科学基金会（FWF，项目 Y 1329 START 计划）的支持。

参考文献

Abasi, H.; 和 Bshouty, N. H. 2019. 使用边检测查询学习图。载于 Garivier, A.; 和 Kale, S. 编，算法学习理论，ALT 2019，2019年3月22日至24日，芝加哥，伊利诺伊州，美国，机器学习研究论文集第98卷，3–30。PMLR。Abasi, H.; Bshouty, N. H.; 和 Mazzawi, H. 2018. 隐藏超图的非自适应学习。《理论计算机科学》，716：15–27。

Alekhnovich, M.; Braverman, M.; Feldman, V.; Klivans, A. R.; 和 Pitassi, T. 2008. 正确学习简单概念类的复杂度。《计算机与系统科学杂志》，74(1)：16–34。

Alon, N.; 和 Asodi, V. 2005. 学习隐藏子图。SIAM J. Discret. Math., 18(4): 697–712. Arvind, V.; K¨obler, J.; 和 Lindner, W. 2009. 合取函数的参数化可学习性。Theor. Comput. Sci., 410(47-49): 4928–4936. Choi, S.; 和 Kim, J. H. 2010. 寻找图的最优查询复杂度界。Artif. Intell., 174(9-10): 551–569. Cygan, M.; Fomin, F. V.; Kowalik, L.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; 和 Saurabh, S. 2015. 参数化算法。Springer. ISBN 978-3-319-21274-6. Damaschke, P.; 和 Muhammad, A. S. 2010. 竞争性群体测试与以最小自适应性学习隐藏顶点覆盖。Discret. Math. Algorithms Appl., 2(3): 291–312. Downey, R. G.; Evans, P. A.; 和 Fellows, M. R. 1993. 参数化学习复杂度。载于 Pitt, L. 编，《第六届 ACM 计算学习理论年会论文集》，COLT 1993，美国加利福尼亚州圣克鲁斯，1993年7月26日至28日，51–57。ACM。Downey, R. G.; 和 Fellows, M. R. 1999. 参数化复杂度。计算机科学专著丛书。Springer. ISBN 978-1-4612-6798-0。

Downey, R. G.；以及 Fellows, M. R. 2013年。参数化复杂性基础。计算机科学文本。Springer。ISBN978-1-4471-5558-4。

G¨artner, T.；以及 Garriga, G. C. 2007年。学习有向割的代价。载于 Kok, J. N.；Koronacki, J.；de M´antaras, R. L.；Matwin, S.；Mladenic, D.；以及 Skowron, A.（编），机器学习：ECML 2007，第18届欧洲机器学习会议，波兰华沙，2007年9月17日至21日，会议论文集，计算机科学讲义第4701卷，第152–163页。Springer。

Gaspers, S.；和 Szeider, S. 2012。《通向可满足性的后门》。载于 Bodlaender, H. L.；Downey, R.；Fomin, F. V.；和 Marx, D. 编，《多变量算法革命及其后续：献给 Michael R. Fellows 60 岁生日的论文集》，计算机科学讲义第7370卷，287–317页。Springer。

Li, Y.；和 Liang, Y. 2018。以近乎最优复杂度学习线性回归混合模型。载于 Bubeck, S.；Perchet, V.；和 Rigollet, P. 编，《学习理论会议（COLT 2018）》，瑞典斯德哥尔摩，2018年7月6日至9日，机器学习研究论文集第75卷，1125–1144页。PMLR。Mohri, M.；Rostamizadeh, A.；和 Talwalkar, A. 2012。《机器学习基础》。自适应计算与机器学习。麻省理工学院出版社。ISBN 978-0-262-01825-8。Nguyen, H.；和 Maehara, T. 2020。图同态卷积。载于《第37届国际机器学习会议论文集（ICML 2020）》，2020年7月13日至18日，线上活动，机器学习研究论文集第119卷，7306–7316页。PMLR。Pitt, L.；和 Valiant, L. G. 1988。从示例中学习的计算局限。《ACM杂志》，35(4)：965–984。Semenov, A. A.；Zaikin, O.；Otpuschennikov, I. V.；Kochemazov, S.；和 Ignatiev, A. 2018。利用 SAT 后门的密码攻击。载于 McIlraith, S. A.；和 Weinberger, K. Q. 编，《第三十二届 AAAI 人工智能会议论文集》（AAAI-18）、《第30届人工智能创新应用会议》（IAAI-18）和《第8届 AAAI 人工智能教育进展研讨会》（EAAI-18），美国路易斯安那州新奥尔良，2018年2月2日至7日，6641–6648页。AAAI出版社。Valiant, L. G. 1984。《可学习性理论》。《ACM通讯》，27(11)：1134–1142。van Bergerem, S.；Grohe, M.；和 Ritzert, M. 2022。学习一阶逻辑的参数化复杂度。载于《第41届 ACM SIGMOD-SIGACT-SIGAI 数据库系统原理研讨会论文集》（PODS ’22），337–346页。美国纽约：计算机协会。ISBN 9781450392600。