arXiv:2302.05960v1 [cs.DS] 12 Feb 2023

树的截断度量维数计算

Paul Gutkovich，Zi Song Yeoh

2023年2月14日

摘要

设 $G=(V,E)$ 为一个简单、无权、连通图。设 $d(u,v)$ 表示顶点 u,v 之间的距离。G 的分辨集是 V 的一个子集 S，满足：已知顶点 v 到 S 中每个顶点的距离，就能唯一确定 v。G 的度量维数定义为 G 的最小分辨集的大小。我们类似地定义图的 k-截断分辨集和 k-截断度量维数，但将距离的概念替换为 $d_k(u,v):=\min(d(u,v)\ k+1)$。

本文证明，对于一般的 k，计算树的 k-截断维数是NP困难。随后，我们提出一种多项式时间算法，用于在 k 为固定常数时计算树的 k-截断维数。

1 引言

对于欧几里得平面中任意三个不共线的点，只要知道平面上某一点到这三个点的距离，就可以确定该点。一般而言，在 d 维空间中，任意一组 $d+1$ 线性无关点，只需知道某一点到这些点的距离，就可以确定该点。

如果将欧几里得空间换成图，问题就变得更加复杂。我们的任务是找出图中顶点的最小集合，使得仅根据某一点到集合中各顶点的距离，就能唯一确定该点。这个最小集合的大小称为图的度量维数。对于正整数 k，我们定义 k截断度量维数，这是一个类似的概念，区别在于它用k截断距离度量取代了通常的距离度量。对于图中的两个顶点 u,v，它们的 k截断距离记作 $d_k(u,v)$，定义为 $\min(d(u,v)\ k+1)$。

关于度量维数的分析已有大量研究（有关当前研究成果的综述，请参见 [10] ）。本文将重点讨论计算树的截断度量维数的算法。

已知计算树的度量维数很容易（可在线性时间内完成）；例如参见 [3, 5, 11, 12] 。相比之下，人们对 k 截断度量维数的算法知之甚少。另已知，对于一般图，计算k截断度量维数是 NP-完全问题 [6]。对于树，

已有一种基于动态规划的多项式时间算法，可计算树的 1-截断度量维数（也称为邻接维数）[1]。我们证明，对于一般的 k，除非 P = NP，否则不存在多项式时间算法来计算树的 k截断度量维数。相比之下，我们证明，对于任意常数 k，都存在一种多项式时间算法来计算树的 k截断度量维数。

在第2节中，我们给出全文将用到的基本定义和符号。在第3节中，我们证明，对于一般的 k 值，计算 k-截断度量维数是 NP困难的。在第 4 节中，我们表明，如果 k 是一个固定常数，则存在一种多项式时间算法，用于计算树的 k-截断维数。

2 预备定义与符号

本文中，我们令 $G = (V, E)$ 为一个简单无向连通图，并且 $n = |V|$。对于任意两个顶点 u、$v \in V$，定义图距离 $d(u, v)$ 为 u 与 v 之间最短路径所含边的数目。对于正整数 k，我们定义 k-截断距离 $d_k(u, v) = \min(d(u, v)\ k+1)$。下面给出全文将用到的一些预备定义。

定义 2.1（k-近、 k-远和 k-支配）。对于两个顶点 u、$v \in V$，若 $d(u, v) \leq k$，则称 u 与 v k-近；否则称 u 与 v k-远。若顶点 u 与某个顶点$v \in S$ k-近，则称集合 $S \subseteq V$ u k-支配。

定义 2.2（k-区分顶点）。我们称顶点 w k 能区分顶点 u 和 v，如果 $d_k(u, w) \neq d_k(v, w)$。我们称 w 为 (u, v) 的一个 k- 区分顶点。

定义 2.3（k-截断（支配）分辨集）。若子集 $S \subseteq V$ 满足：对于任意 $u, v \in V$，都存在某个顶点 $x \in S$，使其能区分 u 和 v，则称 $S \subseteq V$ 为 G 的一个 k-截断分辨集。我们还将满足以下条件的集合 S' 定义为 G 的一个 k-截断支配分辨集：

- S' 是 T 的一个 k-截断分辨集。

V 中的每个元素都被 S' k-支配。

定义 2.4（k-截断度量维数）。我们将 G 的 k-截断度量维数定义为其最小 k-截断分辨集的大小。图 G 的 k-截断度量维数记为 $\dim_k(G)$。我们将 G 的最小 k-截断支配分辨集定义为最小的 k-截断支配分辨集 G。

3 在树上计算截断度量维数的 NP困难性

鉴于计算一般图的度量维数很困难，计算一般图中的 k-截断度量维数同样困难也就不足为奇了。已有研究证明，计算一般图的 k-截断度量维数是 NP困难 [6]。

本节将证明，对于 k 的一般取值，计算一般树的 k-截断度量维数是 NP困难的。

定理 3.1。除非 P $=$ NP，否则对于任意常数 $c > 0$，不存在一个算法能够对所有 $k \leq n$ 在 $O(n^c)$ 时间内计算树上的 k-截断度量维数。

在本节中，令 $f_k(T)$ 表示 T 的最小 k-截断支配分辨集的大小。出于技术原因，处理 $f_k(T)$ 会更容易。两个量 $f_k(T)$ 和 $\dim_k(T)$（ T 的截断度量维数）至多相差 1。

引理 3.2。对于任意树 T 和正整数 k，我们有 $\dim_k(T) \leq f_k(T) \leq \dim_k(T) + 1$。

证明。由于 T 的每个 k-截断支配分辨集自动是 T 的一个 k-分辨集，因此我们得到 $\dim_k(T) \leq f_k(T)$。令 R 为 T 的一个最小 k-截断分辨集。我们分两种情形讨论。

- 情况 1：存在一个顶点 $a \in V(T)$，使得对于所有 $x \in R$，$d(a, x) > k$。在这种情况下，这样的顶点 a 至多存在一个，因为如果存在两个这样的顶点 a和 b，那么对于所有 $x \in R$，$d_k(a, x) = d_k(b, x) = k + 1$，这与 R的定义矛盾。因此，若定义 $M = R \cup \{a\}$，那么M将是一个 k-截断支配分辨集。因此，$f_k(T) \leq |M| = \dim_k(T) + 1$。

- 情况 2：对于所有顶点 $a \in V(T)$，都存在某个 $x \in R$，使得 $d(a, x) \leq k$。在这种情况下，R 满足一个 k-截断支配分辨集的两个条件，因此 $f_k(T) \leq |R| = \dim_k(T)$。

在这两种情形下，我们得到 $f_k(T) \leq \dim_k(T) + 1$。 □

为了证明 NP困难性，我们将问题归约到三维匹配，而众所周知，三维匹配是 NP困难问题。

定义3.3（三维匹配）。设 m 为正整数，设X、Y、Z 为大小为 m的两两不相交的整数集合。设 U 为$X \times Y \times Z$的子集。 U中的每个元素都具有(x_i, y_i, z_i)的形式，其中 $x_i \in X$、 $y_i \in Y$、$z_i \in Z$。三维匹配问题是求最大的 $W \subseteq U$的问题，使得对于任意(x_i, y_i, z_i)、$(x_j, y_j, z_j) \in W$，都有 $x_i \neq x_j$、 $y_i \neq y_j$、 $z_i \neq z_j$。

我们需要以下 NP 困难性结果。

定理 3.4。 *[9]* 将三维匹配近似到乘法因子$_{95\ 94}$ 是NP困难的。

现在我们证明定理 3.1。

定理 3.1 的证明。我们将通过从三维匹配问题进行归约来证明这一点。固定一个三维匹配问题的实例，即令 m 为正整数，并令 X、Y、Z 为 $\{1,2,\ldots,m\}$ 的两两不相交子集，且大小均为 n。令 U 为 $X \times Y \times Z$ 的一个子集。令 M 为 U 中不相交元素的最大数量。

令 $r = |U|$，并令 $S_i = \{x_i, y_i, z_i\}$ 为针对 $i \in \{1,2,\ldots,r\}$ 的 U 的第 i^{th} 个元素。不失一般性，我们可以假设 $r > 2$ 和 $r - M \geq 2$ （后一假设可通过附加一个独立实例来实现，其答案比三元组的数量至少少 2）。令 $k = 2m + 2$。令 T 为一棵根为 u 的树，其子节点为 u_1、u_2、.....、u_r。令 T_i 为以 u_i 为根的子树。对于每个 $i \in \{1,2,\ldots,r\}$，按如下方式构造 T_i：

- 令 $w_{i,}$1, $w_{i,}$2,. . . $w_{i,2k}$ 为顶点，使得 $w_{i,}$1 是 T_i 的根（即 $w_{i,1} = u_i$），且对于所有 $1 \leq j \leq 2k-1$，$w_{i,j}$ 都与 $w_{i,j+1}$ 相连。令 $a_{i,}$1, $a_{i,}$2,$a_{i,}$3 为 S_i 的不同元素。创建顶点 $x_{i,}$1,1,$x_{i,}$1,2, . . . , $x_{i,}$1,$_{a_{i,1}+1}$，使得 $x_{i,}$1,1 是 $w_{i,a_{i,}1}$,1 的子节点，并且对于所有 $1 \leq j \leq a_{i,}$1，$x_{i,}$1,$_{j+1}$ 是 $x_{i,}$1,$_j$ 的子节点。对 $a_{i,}$2,$a_{i,}$3 重复此过程。为简洁起见，对于 $c \in \{1,2,3\}$，定义 $g_{i,c} := x_{i,c,a_{i,c}+1}$。

T 构造完成后，令 R 为 T 的一个最小 k-截断支配分辨集。对于 $i \in \{1,2,\ldots,r\}$，定义 $p_i := |T_i \cap R|$。注意，对所有 i，都有 $p_i \geq 1$，因为在 w_i 距离 k 以内必须存在 R 的一个元素, $_{2k}$，这意味着 T_i 中必须存在 R 的一个元素。

断言 3.5. 我们有 $|R| \geq 2r - M$。

证明。反设 $|R| < 2r - M$，即 $2r - |R| > M$。首先，注意到若 $u \notin R$，则 $\sum_i p_i = |R|$；若 $u \in R$，则 $\sum_i p_i = |R| - 1$。由于对所有 i，都有 $p_i \geq 1$，因此至少有 $2r - |R|$ 个 i 的取值满足 $p_i = 1$。根据我们的假设，满足 $p_i = 1$ 的 i 的取值多于 M 个。不失一般性，假设 $p_1 = p_2 = \cdots = p_{M+1} = 1$。

考虑某个子树 T_i，其中 $i \in \{1,2,\ldots,M+1\}$。它只有一个标记顶点 v_i（即位于 R 中的顶点），该顶点与 w_i 的距离必须在 k 以内，$_{2k}$。这意味着 $v_i \in \{w_{i,k}, w_{i,k+1}, \ldots, w_{i,2k}\}$。对于任意可能的 v_i，以及任意$c \in \{1,2,3\}$，我们得到 $d(v_i, g_{i,c}) \geq d(w_{i,k}, g_{i,c}) = d(w_{i,k}, w_{i,a_{i,c}}) + d(w_{i,a_{i,c}}, g_{i,c}) = (k - a_{i,c}) + (a_{i,c} + 1) = k + 1$。这意味着，对于任意 g_j，v_i 与它的距离都大于 k，对任意 $j \in \{1,2,\ldots,r\}$ 和任意 $c \in \{1,2,3\}$ 均如此，$_c$。这意味着，对于任意 $i \in \{1,2,\ldots,M+1\}$ 和任意 $c \in \{1,2,3\}$，必定存在某个顶点 $y \notin T_1 \cup T_2 \cup \cdots \cup T_{M+1}$，使得 y 与 g_i 的距离在 k 以内，$_c$。注意，要找到这个 y，我们只需考虑一个顶点，即在 $\{u\} \cup T_{M+2} \cup T_{M+3} \cup \cdots \cup T_r$中距离 u 最近的顶点。从现在起，y将表示这个顶点。因此，我们

有：如果 $g_{i_1,\ c_1}$ 且 $g_{i_2,\ c_2}$，其中 $i_1, i_2 \in \{1, 2, \ldots, M+1\}, c_1, c_2 \in \{1, 2, 3\}$ 被k区分（即，它们到该顶点的截断距离各不相同），而该顶点属于 R（即 k），那么它们也会被 y k 区分。我们证明，y 无法用 k 区分所有配对$g_{i_1,\ c_1}$ 以及 $g_{i_2,\ c_2}$。

注意，如果 y k 区分某些 $g_{i_1,\ c_1}$ 和 $g_{i_2,\ c_2}$，其中 $i_1, i_2 \in \{1, 2, \ldots, M+1\}, c_1, c_2 \in \{1, 2, 3\}$，那么 u k 也能区分它们。因此，我们可以假设$y = u$。

注意，对于某些 $i \in \{1, 2, \ldots, M+1\}, c \in \{1, 2, 3\}$，有 $d(u, g_{i,c}) = d(u, w_{i,a_{i,c}}) + d(w_{i,a_{i,c}}, x_{i,c,a_{i,c}+1}) = a_{i,c} + a_{i,c} + 1 = 2a_{i,c} + 1$。如果 u k 区分所有成对的 g_{i_1,c_1}, g_{i_2,c_2}，其中 $i_1, i_2 \in \{1, 2, \ldots, M+1\}, c_1, c_2 \in \{1, 2, 3\}$，那么 $\min(k+1, 2a_{i,c}+1)$ 对所有 $i \in \{1, 2, \ldots, M+1\}, c \in \{1, 2, 3\}$ 都互不相同。由于$k = 2m + 2 > 2a_{i,c}$，所有 $a_{i,c}$ 对所有 $i \in \{1, 2, \ldots, M+1\}, c \in \{1, 2, 3\}$ 都互不相同。然而，这等价于说 $S_1, S_2, \ldots, S_{M+1}$ 两两不相交，这与 M 的极大性矛盾。这导致矛盾，从而推出 $|R| \geq 2r - M$。 □

命题 3.6。存在一个大小为 $2r - M$ 的 T 的 k-截断支配分辨集 R。

证明。假设 U 的最大互不重叠子集为 S_1、S_2、.....、S_M。通过纳入 $b_i := w_{i,\ k}$（对所有 $i \in \{1, 2, \ldots, r\}$，称这些为底部顶点）以及 $t_i := u_i$（对所有 $i \in \{M+1, M+2, \ldots, r\}$，称这些为顶部顶点），构造集合 R。我们证明该集合 R 满足要求。

首先，我们证明所有顶点到 R 中某个顶点的距离都不超过 k。注意，u 与 t_r 距离很近。令 v 为 T_i 中任意一个顶点。只有在 $v = g_{i,\ c}$ 对某个 $c \in \{1, 2, 3\}$ 成立时，v 才不会靠近 b_i。然而，在这种情形下，我们有 $d(t_r, g_{i,\ c}) \leq 2a_{i,\ c} + 2 \leq 2m + 2 = k$，这意味着 v 靠近 t_r。

现在，我们将证明 T 的所有顶点都能被 R 中的某个顶点以 k 区分。为此，设 T 中有一个未知顶点 q。假设我们只知道对于所有 $y \in R$，$d_k(q, y)$ 的值，并且希望唯一确定 q。我们分几种情形讨论：

- 情况 1：q 与任何顶部顶点都不够 k。在这种情况下，q 到恰好一个底部顶点 b_i 的距离必须在 k 以内。q 唯一可能的值是 $w_{i,\ k+d_k(q,\ b_i)}$。

- 情况 2：q 与某个顶部顶点的距离在 k 以内。
  如果 q 仅与一个顶部顶点的距离在 k 以内，那么设该顶部顶点为 t_i。由于我们假设 $r - M \geq 2$，q 只有 $b_i = w_{i,\ k}$ 和 $w_{i,\ k+1}$ 两种可能，因此可以根据 q 到 b_i 的距离确定 q。
  现在假设 q 与至少两个顶部顶点的距离在 k 以内。在这种情况下，q 到所有顶部顶点的距离都必须在 k 以内。
  如果 q到所有顶部顶点的距离都相等，那么它必定在 $T_1 \cup T_2 \cup \cdots \cup T_M \cup \{u\}$中。如果 $d(q, t_r) = 1$，那么 $q = u$。假设 $d(q, t_r) > 1$。
  如果 q 也与某个底部顶点 b_i k-接近，那么它就属于 T_i。令 w' 为形如 $w_{i,j}$ 且距离 q最近的顶点。我们知道 $d(q, w') = d(q, t_r) +$

$d(q,b_i) - k - 1$，且 $d(u,w') = d(u,q) - d(q,w') = d(q,t_r) - 1 - d(q,w')$。根据这两个量，我们可以求出 q 是什么。

如果 q 与任何底部顶点都不相距 k，那么它必定具有 $g_{i,c}$ 的形式，其中 $i \in \{1,2,\ldots,M\}$，且 $d(t_r,q) = 2a_{i,c} + 2 \leq k$。由于对于 $i \in \{1,2,\ldots,M\}$，没有任何 S_i 重叠，因此仅凭它到 t_r 的距离就能确定 q。

如果 q 到所有顶部顶点的距离不相等，那么存在某个 $i \in \{M+1, M+2, \ldots, r\}$，使得 $d(q,t_i)$ 最小。这意味着 q 位于 T_i 中。假设 q 与 b_i 相距 k。那么，令 w' 为距离 q 最近且具有 w_i 形式的顶点，其中 j，且 $_j$。那么 $d(w',q) = d(t_i,q) + d(b_i,q) - (k-1)$，并且 $d(u,w') = d(u,q) - d(q,w') = d(q,t_r) + 1 - d(q,w')$。根据这两个量，我们可以确定 q。

最后，如果 q 与 b_i不相距 k，那么对于某个 $c \in \{1,2,3\}$，$q = g_{i,c}$。因此，可以根据 $d(q,t_i)$确定 q。

这表明 R 是 T的一个 k-截断支配分辨集。因此，存在一个大小为 $2r - M$ 的 k-截断支配分辨集。 □

结合引理 3.2，我们可以看出，T的 k-截断度量维数处于范围 $[2r - M - 1, 2r - M]$ 内。

现在，假设我们可以在 $O(n^c)$时间内计算 T 的最小 k-分辨集。根据引理 3.2，这意味着我们可以将最小截断支配分辨集近似到加性误差 1 以内，也就是说，我们将 U 的三维匹配问题近似到了加性误差 1 以内。然而，这与定理 3.4矛盾。因此，在树上计算 k-截断度量维数是 NP困难的，证毕。

□

4 求解常数 k下截断度量维数的多项式时间算法

已有一种基于动态规划的多项式时间算法，可计算有根树的 1-截断度量维数（也称为邻接维数） [1]。我们将这一结果推广到任意常数 k。

固定一个正整数 k。本节介绍一种算法，用于计算树的 k-截断度量维数，其运行时间关于 n（即树的顶点数）为多项式时间。为简单起见，我们介绍该算法一个略欠优化的版本，其运行时间为 $O(n^2)$。我们可以将算法改进至 $O(n)$，但这并非本文的重点。

首先，我们给出一种算法，用于计算树 T 的最小 k-截断支配分辨集。随后，我们说明如何稍作修改这一算法，以计算 T 的 k-截断度量维数。

4.1 计算最小的 k-截断支配求解集

我们提出一种动态规划算法，用于计算任意固定常数 k 下最小 k-截断支配分辨集的大小。

设 T 是一棵具有 n 个顶点的有根树，且 k 为正整数。用整数 1、2、...、n 标记树中的顶点。对于 T 的任意顶点 u，令 ch(u) 表示 u 拥有的子节点数，并令 T_u 表示以 u 为根的 T 的子树。对于顶点 u 和正整数 $m \leq$ ch(u)，定义 $c_m(u)$ 为 u 中标签最小的第 m^{th} 个子节点。对于任意 $u \in V(T)$ 和正整数 $m \leq$ ch(u)，我们定义 T_u、$_m = \{u\} \cup T_{c_1(u)} \cup T_{c_2(u)} \cup \cdots \cup T_{c_m(u)}$。我们定义函数 $f(u, C, D, E, l, m)$ 为满足以下性质的最小集合 S 的大小：

- S 是 T' 的子集，其中 $T' := T_{u,\ m}$。
- 如果 v 是 T' 之外的任意顶点，且 $d(u, v) = l$，那么 $S \cup \{v\}$ 是 T'的 k-截断支配分辨集。如果 l 是 $null$，那么 S 是 T'的 k-截断支配分辨集。注意，如果 $l > k$，它无法k区分 T'中的元素，也无法与这些元素保持 k距离，因此我们只允许 l 取 1 到 k 之间的整数，或取 $null$。
- 集合 $\{d(u, z) \mid z \in S,\ d(u, z) \leq 2k\}$ 等于 C。
- 集合 $\{d(u, y) \mid y \in T',\ d(u, y) \leq 2k : \quad \forall z \in S,\ d(y, z) > k\}$ 等于D。
- 如果 $x \in V(T') \setminus S$ 满足以下条件，则称其为良好：
  - $d(x, u) \leq 2k$。– 存在 $z \in S$，使得 $d(x, z) \leq k$。– 对所有满足 $d(x, z) \leq k$的 $z \in S$，$d(x, z) - d(u, z)$ 均相等。

  我们从空集合 E 开始。对于每个良好的 x，我们将元组 $(d(x, u)\ d(x, z) - d(u, z)\ d(u, z'))$ 加入 E，其中 $z \in S$and $d(x, z) \leq k$，且$z' \in S$是在条件 $d(x, z') > k$ 下使 $d(u, z')$ 最小的顶点。如果这样的 z' 不存在，则令 $d(u, z') = null$。

我们定义 $g(u, C, D, E, l) := f(u, C, D, E, l,$ ch$(u))$。注意，T 的最小 k-截断支配分辨集的大小，正是对所有有效的 C、D、E，取 $g(root, C, D, E, null)$ 的最小值。

现在我们将介绍一种用于计算 f 的递归算法。首先，对于 u、C、D、E、l、m 的所有有效元组，将$f(u, C, D, E, l, m)$ 设为 $null$。证明的其余部分将通过大量分类讨论，描述所有必要的状态转移。

基本情形。若 u是叶节点，则 $f(u, C, D, E, l, m)$ 仅可能取值 0 或 1，分别对应集合 $\emptyset, \{u\}$。由于 u 没有子节点，我们只允许$m = 0$。使 $f(u, C, D, E, l, m) = 0$的 C, D, E, l 的有效组合为 $C = \emptyset, D = \{0\}, E = \emptyset, l \neq null$。使 $f(u, C, D, E, l, m) = 1$的 C, D, E, l的有效组合为 $C = \{0\}, D = \emptyset, E = \emptyset, l = null$。

递归步骤：添加一个父节点。从现在起，我们假设 u 不是叶节点。假设 $m = 1$。令 $v = c_1(u)$，并令 C, D, E, l 满足： $g(v, C, D, l)$不是 $null$。令 S 为相应的标记顶点集合。令 l' 为从 u 到 $\{u\} \cup T_v$之外最近的标记顶点的距离。

我们有以下观察结果：

- 若 $l = 1$，则 u 会被标记，而 l' 可以取任意值。
- 若 $2 \leq l \leq k$，则 $l' = l - 1$。
- 若 $l = null$，则 l' 要么是 $null$，要么是 k。

我们只需检查 S 与对应于 l' 的顶点是否能定位并 k-支配 u。若 $l = 1$，则必定要标记 u，从而得到 $f(u, C', D', E', l', 1) = |S| + 1$。反之，假设 $1 < l' \leq k$。我们知道，u 被 S k-支配，因为 $l' \leq k$。此外，u 是距离对应于 l' 的顶点最近的顶点，因此它自动与 T_v 中的所有顶点 k-区分，从而得到 $f(u, C', D', E', l', 1) = |S|$，其中 C'、D'、E' 分别是对应于 $T_v \cup \{u\}$ 和集合 S 的 C、D、E 参数。现在假设 $l = l' = null$。对于任意的 $x \in V(T_v)$，我们要检查 S k 是否能区分 u 和 x。

如果存在两个不同的 $z_1, z_2 \in S$，使得 z_1, z_2 与 x 和 d 相距 k，且 $x, z_1 - d(v, z_1) \neq d(x, z_2) - d(v, z_2)$，那么我们声称 z_1, z_2 中至少有一个能区分 (x, u)。为此，注意到

$$\begin{aligned} d(x, z_1) = d(u, z_1) &\implies d(x, z_1) - d(v, z_1) = 1 \\ &\implies d(x, z_2) - d(v, z_2) \neq 1 \\ &\implies d(x, z_2) \neq d(u, z_2), \end{aligned}$$

这意味着，如果 z_1 不能区分 x, u，那么 z_2 就能区分。

现在，假设不存在这样的 z_1, z_2。注意，$l' = null$ 意味着 x 必须是良好的，即 $(d(x, v)\ d(x, z) - d(v, z)\ d(v, z')) \in E$，其中 $z, z' \in S$（z' 可能不存在）。如果某个与 x 相距 k 的 $z \in S$ 区分了 x, u，那么有 $d(x, z) \neq d(u, z) \iff d(x, z) - d(v, z) \neq 1$。如果 x, u 是由另一个 $z' \in S$ 区分的，那么 $d(v, z') \leq k - 1$ 是充要条件。因此，x, u 能被 S 区分，当且仅当 $d(x, z) - d(v, z) \neq 1$ 或 $d(v, z') \leq k - 1$。通过检查这对 E 中的所有元素是否成立，我们可以确定 S 是否为 $T_v \cup \{u\}$ 的 k-截断支配分辨集。如果所有 x, u 对都被区分，则 $f(u, C', D', E', l', 1) = |S|$。

递归步骤：合并一个子节点子树。现在假设 $m > 1$。令$v = c_m(u)$、$T' = T_u$、m、$T_1 = T_u$、$m-1$和 $T_2 = T_v$。选取有效的 C_1、D_1、E_1、l_1，使得 $f(u$、C_1、D_1、E_1、l_1、$m - 1)$ 不 $null$（将其对应的集合记为 S_1），以及有效的 C_2、D_2、E_2、l_2，使得 $g(v$、C_2、D_2、E_2、$l_2)$ 不 $null$（将其对应的集合记为 S_2）。令 u_1、u_2 分别为位于 T_1、T_2 之外的顶点，且满足 $d(u$、$u_1) = l_1$、$d(v$、$u_2) = l_2$。首先，我们要检查 S_1、S_2、l_1、l_2 是否相容。分几种情形讨论。

• 情况 1: $u_1 \in S_2$ 和 $u_2 \in S_1$。要使其成立，我们需要检查 $l_1 - 1 = \min(C_2)$ 和 $l_2 - 1 = \min(C_1)$。在这种情况下，我们可以令 l'（即 T' 之外距离 u 最近的标记顶点的距离）取 $\max(l_1, l_2 - 1)$ 与 $k+1$ 之间的任意整数，或者取 $null$。

• 情况 2: $u_1 \in S_2$ 和 $u_2 \notin T'$。对此，我们需要检查 $l_1 - 1 = \min(C_2)$、$l_2 \leq 1 + \min(C_1)$，以及 $l_1 \leq l_2 - 1$。在这种情况下，$l' = l_2 - 1$。

• 情况 3: $u_2 \in S_1$ 和 $u_1 \notin T'$。对此，我们需要检查 $l_2 - 1 = \min(C_1)$、$l_1 \leq 1 + \min(C_2)$，以及$l_2 \leq l_1 + 1$。在这种情况下，$l' = l_1$。

• 情况 4: u_1、$u_2 \notin T'$。此时，必须有 $u_1 = u_2$。对于这种情况，我们需要检查 $l_1 + 1 = l_2$、$l_1 \leq 1 + \min(C_2)$，以及 $l_2 \leq 1 + \min(C_1)$。在这种情况下，$l' = l_1$。

• 情况 5: $l_1 = null$ 和 $l_2 \neq null$。我们要求 $\min(C_2) + 1 > k$。两种情况之一是 $u_2 \in S_1$，此时要求 $l_2 = \min(C_1) + 1$，并允许 l' 取 $l_2 - 1$ 与 k 之间的任意整数；另一种情况是 $null$或 $u_2 \notin T$，此时要求 $l_2 \leq \min(C_1) + 1$，并得到 $l' = l_2 - 1$。

• 情况 6: $l_1 \neq null$ 和 $l_2 = null$。我们要求 $\min(C_1) + 1 > k$。两种情况之一是 $u_1 \in S_2$，此时要求 $l_1 = \min(C_2) + 1$，并允许 l' 取 l_1 与 k 之间的任意整数；另一种情况是 $null$或 $u_1 \notin T'$，此时要求 $l_1 \leq \min(C_2) + 1$，并得到 $l' = l_1$。

• 情况 7: $l_1 = l_2 = null$。在这种情况下，我们要求 $\min(C_1) + 1$、$\min(C_2) + 1 > k$。必须有 $l' = null$。

仅当上述情形之一成立时，我们才能继续处理由该情形 l' 确定的内容。现在，我们要检查 $S' := S_1 \cup S_2 \cup \{u_1, u_2\}$ 是否构成$T_{u\text{中的}m}$的 k-截断支配分辨集。只需检查对于所有$x \in V(T_1)$ $y \in V(T_2)$，在 S' 中存在某个 z，能够以 k区分 x 和 y。我们分几种情形讨论。

• 情况 1: 某个 $z_1 \in S_1$ 与 x 的距离为 k，且某个 $z_2 \in S_2$ 与 y 的距离为 k。在这种情况下，我们断言 x 和 y 可由 z_1、z_2 之一以 k 区分。为导出矛盾，假设此断言不成立。那么 $d(x, z_1) = d(y, z_1)$ $d(x, z_2) =$

$d(y, z_2)$。令 $a = \text{lca}(x, z_1)$ $b = \text{lca}(y, z_2)$。那么

$$\begin{aligned}
0 &= d(x, z_1) - d(y, z_1) \\
&= d(x, a) - d(y, a) \\
&= (d(x, z_2) - d(a, z_2)) - d(y, a) \\
&= d(y, z_2) - d(a, z_2) - d(y, a) \\
&= d(y, b) - d(a, b) - d(y, a) \\
&= d(y, b) - d(a, b) - (d(y, b) + d(a, b)) \\
&= -2d(a, b) < 0,
\end{aligned}$$

这就得到了所需的矛盾。

- 情况 2： x 与 S_1 中的任何元素都不 k-近，且 y 与 S_2中的任何元素都不 k-近

  首先，我们需要检查 x 是否与 S_2 中的某个元素 k-近，或者 y是否与 S_1 中的某个元素 k-近。因此，我们必须检查 $d(x, u)$ $+\min(C_2)$ $+ 1$ 或 $d(y, v)$ $+ \min(C_1)$ $+ 1$ 是否小于 $k + 1$。注意，$d(x, u) \in D_1$， $d(y, v) \in D_2$。如果这两个条件都不成立，那么我们需要检查 x 和 y 是否由 T'之外的一个顶点区分；也就是说，它们由对应于 l'的顶点区分。为此，我们需要检查 $\min(k + 1, d(x, u) + l') \neq \min(k + 1, d(y, v) + l' + 1)$。如果不满足该条件，那么 S' 不会 k-区分(x, y)。否则，S' 确实会区分(x, y)。

- 情况 3： x 与 S_1 中的某个元素相距 k，而 y 与 S_2 中的任何元素都不相距 k

  注意， $d(v, y) \in D_2$。如果存在两个 z_1、 $z_2 \in S_1$，它们与 x相距 k，并且 $d(x, z_1) - d(u, z_1) \neq d(x, z_2) - d(u, z_2)$，那么我们声称，$z_1$ 或 z_2 k-区分了 x 和 y。为证明这一点，假设 z_1 不k-区分 x 和 y。那么

$$\begin{aligned}
d(x, z_1) = d(y, z_1) &\implies d(x, z_1) - d(u, z_1) = d(u, y) \\
&\implies d(x, z_2) - d(u, z_2) \neq d(u, y) \\
&\implies d(x, z_2) \neq d(z_2, y),
\end{aligned}$$

  这意味着 z_2 区分了 x、 y。
  从现在起，假设不存在这样的 z_1、 z_2 。这意味着，对于所有与 x相距 k 的 $z \in S_1$，$d(x, z) - d(u, z)$ 都是常数。这意味着 E_1 包含元组（$d(u, x)$ $d(x, z) - d(u, z)$ $d(u, z')$），其中， z是 S_1 中与 x相距 k的任意元素，而 z' 是 S_1 中与 x不相距 k的所有元素中距 u 最近的顶点。现在我们将检验 x、 y 是否被区分。我们有几种情形：

  - 子情况 3.1： x、 y 被 S_2中的一个元素区分
    我们只需要 d 我们只需判断 $d(x, y)$ 靠近一个 C_2元素 $1 \leq k$。

– 子情况 3.2：x、y 由 S_1 的一个元素区分。对于任意与 x k-接近的 $z \in S_1$，有 $d(x,z) \neq d(y,z) \iff d(x,z) - d(u,z) \neq d(u,y)$，因此只需检查 $d(x,z) - d(u,z) \neq d(u,y) = 1 + d(v,y)$。对于任意不与 x k-接近的 $z' \in S_1$，我们需要检查与 u 最近的此类 z' 是否与 y k-接近，即需要确保 $d(u,z') + d(v,y) + 1 \leq k$。– 子情况 3.3：$x$、$y$ 由 T' 外部的一个标记顶点区分，即由对应于 l' 的顶点区分。要使 x、y 得以区分，我们要求 $d(x,u) + l' \neq d(y,v) + l' + 1$，这等价于 $d(x,u) \neq d(y,v) + 1$，且 $\min(d(x,u) + l', d(y,v) + l' + 1) \leq k$。

顶点 x 和 y 可区分，当且仅当以下情形中至少有一种成立。

- 情况 4：y 与 k 接近某个 S_2 中的元素，而 x 与 k 不接近 S_1 中的任何元素

  这种情形与情况 3 类似。
  我们必须确保，对于任意 $x \in V(T_1)$ $y \in V(T_2)$，上述情形中至少有一种成立。现在我们将给出一个算法，在 $O(A(k))$ 时间内完成所有这些检查，其中 A 是一个仅以 k 为变量的函数（即与 n 无关）。该算法包含以下步骤：

  – 检查对于每个 $d_1 \in D_1$、$d_2 \in D_2$，是否有 $\min(k+1, d_1 + l') \neq \min(k+1, d_2 + l' + 1)$。这将检查情况 2 的所有情形。
  – 对于每个元组 $(d(u,x)\ d(x,z) - d(u,z)\ d(u,z')) \in E_1$ 以及每个 $d_2 \in D_2$，检查以下条件是否至少有一个成立：
    * 检查 $d(x,u) + \min(C_2) + 1 \leq k$ 是否成立。这对应情况 3 子情况 1。
    * 检查 $d(x,z) - d(u,z) \neq 1 + d_2$ 或 $d(u,z') + d_2 + 1 \leq k$ 是否成立。这对应情况 3 子情况 2。
    * 检查 $d(x,u) \neq d_2 + 1$ 且 $\min(d(x,u) + l', d_2 + l' + 1) \leq k$ 是否成立。这对应情况 3 子情况 3。

    这会检查情况 3 的所有情形。
  – 对于每个元组 $(d(v,y)\ d(y,z) - d(v,z)\ d(v,z')) \in E_2$ 以及每个 $d_1 \in D_1$，检查以下条件是否至少有一个成立：
    * 检查 $d(y,v) + \min(C_1) + 1 \leq k$。这会检查情况 4 子情形 1。
    * 检查 $d(y,z) - d(v,z) \neq 1 + d_1$ 或 $d(v,z') + d_1 + 1 \leq k$。这会检查情况 4 子情形 2。
    * 检查 $d(y,v) \neq d_1 - 1$ 以及 $\min(d(y,v) + l' + 1, d_1 + l') \leq k$。这会检查情况 4 子情形 3。

    这会检查情况 3 的所有情形。

注意，每个步骤的完成时间仅取决于 k，而不取决于 n；通过完成这些步骤，我们检查了上述情形中的所有情况。如果

在算法执行过程中，某种情况不满足条件，我们就知道 S' 与对应于 l' 的顶点构成的集合不是 T' 的有效 k-截断支配分辨集。如果所有情况都满足各自的条件，那么我们就得到 T' 的一个有效 k-截断支配分辨集。

现在，我们将说明如何根据参数$C_1, D_1, E_1, C_2, D_2, E_2$，求出 T', S' 的参数 C, D, E。我们有

$$C = C_1 \cup \{c+1 \mid c \in C_2, c \leq 2k-1\}$$

$$D = \{d \mid d \in D_1, d + \min(C_2) + 1 > k\}$$
$$\cup \{d+1 \mid d \in D_2, d \leq 2k-1, d + \min(C_1) + 1 > k\}.$$

考虑满足$(d(x,u)\ d(x,z_1) - d(u,z_1)\ d(u,z')) \in E_1$的 x, z_1, z'。如果 x 与 S_2 中的任何元素都不相距 k以内（即 $d(u,x) + \min(C_2) + 1 > k$），那么 E应包含$(d(x,u)\ d(x,z_1) - d(u,z_1), \min(d(u,z'), \min(C_2) + 1))$。反之，假设 x 与 $z_2 \in S_2$相距 k 以内。令 c 为 C_2中满足 $d(u,x) + 1 + c > k$的最小元素。我们有 $d(x,z_2) - d(u, z_2) = d(u,x)$。因此，当且仅当$d(u,x) \neq d(x,z_1) - d(u,z_1)$时，$E$必须包含$(d(x,u)\ d(x,z_1) - d(u,z_1), \min(d(u,z')\ c+1))$。

对于 E_2 中的元组，同样的推理也成立。令 $(d(y,v)\ d(y,z_1) - d(v,z_1)\ d(v,z'))$ 为 E_2 的任意元素。如果 y 与 C_1 中的任何元素都不 k-接近（即 $d(y,v) + 1 + \min(C_1) > k$），那么 E 应包含
$(d(y,v)\ d(y,z_1) - d(v,z_1), \min(d(v,z'), \min(C_1)))$。反之，假设 y 与某个 $z_2 \in S_1$ k- 接近。令 c 为 C_1 中满足 $d(u,x) + 1 + c > k$ 的最小元素。我们有 $d(y,z_2) - d(v,z_2) = d(y,v)$。因此，当且仅当 $d(y,z_1) - d(v,z_1) = d(y,v)$ 时，E 包含 $(d(y,v)\ d(y,z_1) - d(v,z_1), \min(d(y,z')\ c))$。这样，我们可以在仅取决于 k 的时间内，从 $C_1, D_1, E_1, C_2, D_2, E_2$ 得到 C, D, E。

如果算法中的所有情形都通过了检查，并且我们有$f(u,C,D,E,l,m)$大于或等于 $|S'|$，那么就将其设为 $|S'|$。遍历 $C_1, D_1, E_1, l_1, C_2, D_2, E_2, l_2$的所有有效组合，以及所有有效的 l'（其数量仅取决于 k），即可找到 $f(u,C,D,E,l,m)$的最小可能值，正如所需。

注意，由于每个递归步骤的运行时间仅取决于 n，因此，对于固定的 k，计算 T的最小 k-截断支配分辨集的大小（即在所有有效的 C, D, E中取 $g(root, C, D, E, null)$的最小值）的运行时间关于 n 是线性的。

4.2 计算 k-截断度量维数

现在我们将说明如何修改上一节中的算法，以计算 $\dim_k(T)$，即 T的 k-截断度量维数。注意，唯一的区别在于，现在允许一个顶点与分辨集中的所有顶点都相距 k。

设 T 是一棵树，k 是一个正整数。设 S 是 T 的最小 k-分辨集。我们知道，要么 $V(T)$ 中的每个元素都与某个元素相距 k

于 S 不超过该距离，要么 $V(T)$ 中恰有一个元素与 S 中的任何元素都不相距 k 以内。在第一种情况下， S 是 T 的最小 k-截断支配分辨集，这意味着可以使用上一节中的算法找到它。设 T 的最小 k-截断支配分辨集的大小为 s_{mdr}。现在令 r 为 $V(T)$ 中的任意元素，并假设 r 与 S 中的每个元素都相距 k 以外。以 r 为根对树 T 进行定根。下面我们将说明如何修改第 4 节中的算法，从而得到能够分辨 T 的最小集合 S，同时 S 中的每个元素都与 r 相距 k 以外。

令 $v_1, v_2, \ldots, v_p$ 为 u 的子节点。我们使用上一节所述的算法，对所有 $i \in \{1, 2, \ldots, p\}$ 以及所有有效的 C_i, D_i, E_i，计算 $g(v_i, C_i, D_i, E_i, null)$。我们希望确保 u 与所有标记顶点的距离至少为 k，因此要求 $l = null$，这意味着 D_i 必须为 $\emptyset$。这也意味着我们要求 $\min(C_i) \geq k$。

现在，我们希望计算 T 的最小分辨集 S，使得在 T 中，唯一未被 S 截断支配的顶点是 r。令 S_i 为 $i \in \{1, 2, \ldots, p\}$ 的 $S \cap T_i(r)$。我们知道，对于所有 i，S_i 必须是某个满足 $\min(C_i) > k$ 的有效 C_i, E_i 所对应的 $g(v_i, C_i, D_i = \emptyset, E_i, l_i = null)$ 集合。注意，由于 $l_i = null$，仅凭其自身就必须是 $T_i(r)$ 的一个 k-截断支配分辨集。令 q_i 为所有满足 $\min(C_i) \geq k$ 的有效 C_i, E_i 所对应的 $g(v_i, C_i, D_i = \emptyset, E_i, l_i = null)$ 中的最小值。那么必有 $|S| = q_1 + q_2 + \cdots + q_p$。

令 $s_{\min}$ 为所有 $r \in V(T)$ 中 $|S|$ 的最小值。我们必有 $\dim_k(T) = \min(s_{\mathrm{mdr}}, s_{\min})$。至此，算法的描述完毕。

5 结论与未来工作

本文重点研究了树的截断度量维数计算。我们证明，对于一般 k，计算树的 k-截断度量维数是 NP困难的；但对于任意常数 k，都可以在多项式时间内求解。

关于 k-截断度量维数的计算，仍有许多尚未解决的问题。

- 对于树 T 的 k-截断度量维数 $\dim_k(T)$，算法对 k 的依赖程度能达到多优?

- 已知，对于一般图，计算度量维数的最佳近似比为 $\Theta(\log n)$ [4]。另一方面，可以在线性时间内计算树的度量维数 [3]。对于树的 k-截断度量维数，我们能获得的最佳近似比是多少?

- 众所周知，我们无法在一般图中高效计算 k-截断度量维数（即使 k 是一个较小的常数） [4]。不过，对于其他图类，给定任意常数 k，我们能否高效计算截断度量维数?

6 致谢

我们感谢 Jesse Geneson 提供这个项目，并与我们讨论了与截断度量维数相关的问题。我们也感谢 Felix Gotti 和 Tanya Khovanova 审阅本文，并提出许多有助于改进文章表达的意见。最后，我们感谢 MIT PRIMES 项目给予我们开展这项研究的机会。

参考文献

[1] R.C. Tillquist、R.M. Frongillo 和 M.E. Lladser, 《有限图的截断度量维数》（2021）https://arxiv.org/abs/2106.14314[2]J. Geneson、E. Yi, 《图的距离-k 维数》（2021）https://arxiv.org/abs/2106.08303[3] G. Chartrand、L. Eroh、M.A. Johnson 和 O.R. Oellermann, 《图中的可分辨性与图的度量维数》。《离散应用数学》105（2000）99-113[4] M. Hauptmann、R. Schmied 和 C. Viehmann, 《度量维数问题的近似复杂性》。《离散算法杂志》14（2012）214-222[5] A. Rosenfeld、B. Raghavachari、S. Khuller, 《图中的地标》。《离散应用数学》70（3）（1996）217–229[6]A. Estrada-Moreno、I.G. Yero 和 J.A. Rodriguez-Velazquez, 《图的k-度量维数：一种通用方法》（2016）https://arxiv.org/pdf/1605.06709.pdf[7] B. Piotr、D. Bhaskar、K. Ming-Yang, 《计算机与系统科学杂志》。《生物信息学中测试集问题的紧致可近似性结果》71（2）（2005）145-162[8] M. Chleb´ík、J. Chleb´íkov´a, 《有界度图中支配集问题的近似困难性》。《信息与计算》206（11）（2008）1264-1275[9] M. Chleb´ík、J. Chleb´íkov´a, 《优化问题有界变体的近似复杂性》。《理论计算机科学》354（3）（2006）320-338[10] R.C. Tillquist、R.M. Frongillo 和 M.E. Lladser, 《离散空间中的地形概览：度量维数及其应用综述》https://arxiv.org/abs/2104.07201[11] S. Peter J., 《树的叶节点》。《第六届东南组合学、图论与计算会议论文集》（佛罗里达大西洋大学，佛罗里达州博卡拉顿，1975），Congressus Numerantium。14（1975）549–559

[12] H. Frank、M. Robert A.，《图的度量维数》。《Ars Combinatoria》2（1976）191–195[13] Garey, M. R.; Johnson, D. S.（1979），《计算机与不可解性：NP 完全性理论指南》，W.H. Freeman，ISBN 0-7167-1045-5 A1.5：GT61，第204页。