

异构网络环境下最大化谱间距的拓扑设计^①



缪一航¹, 徐跃东², 吴俊¹

¹(复旦大学 计算机科学技术学院, 上海 201203)

²(复旦大学 信息科学与工程学院, 上海 201203)

通信作者: 吴俊, E-mail: wujun@fudan.edu.cn

摘要: 分布式平均共识和去中心化机器学习是具有广泛应用的去中心化计算方法. 两种方法的收敛率主要由拓扑的谱间距所决定. 节点网络环境的异构性包括节点带宽和节点间连接可用性的不同. 异构网络环境对去中心化计算的效率提出了挑战. 本文研究异构网络环境下最大化谱间距的拓扑设计问题, 推导了谱间距针对拓扑任一条边的梯度, 并设计了基于该梯度的增删边算法来构建目标拓扑. 构建的拓扑具有更大谱间距, 且各节点的数据通信时间相近. 拓扑构建算法的性能在不同程度的异构网络环境下能够保持稳定, 且生成的拓扑在分布式共识中以更快的收敛率和更短的时间达到收敛. 基于该算法, 本文进一步验证了最新发现的谱间距与去中心化机器学习收敛率的弱相关性.

关键词: 去中心化机器学习; 分布式平均共识; 拓扑设计; 谱间距

引用格式: 缪一航, 徐跃东, 吴俊. 异构网络环境下最大化谱间距的拓扑设计. 计算机系统应用, 2023, 32(9): 248–256. <http://www.c-s-a.org.cn/1003-3254/9234.html>

Topological Design for Maximizing Spectral Gap in Heterogeneous Network Environment

MIAO Yi-Hang¹, XU Yue-Dong², WU Jun¹

¹(School of Computer Science, Fudan University, Shanghai 201203, China)

²(School of Information Science and Engineering, Fudan University, Shanghai 201203, China)

Abstract: Distributed average consensus and decentralized machine learning are widely employed decentralized computing methods. The convergence rates of the two methods are mainly determined by the spectral gap of the topology. The heterogeneity of the network environment among nodes includes the difference in node bandwidth and inter-node connection availability. The heterogeneous network environment poses a challenge to decentralized computation efficiency. This work studies the topology design of maximizing the spectral gap under a heterogeneous network environment. The gradient of the spectral gap for any edge of the topology is derived and an edge-addition and deletion algorithm is designed based on this gradient to construct the target topology. The generated topology has larger spectral gaps and similar data communication time of each node. The performance of this algorithm remains stable under different levels of heterogeneous network environments. The generated topology achieves convergence with a faster convergence rate and shorter time in distributed consensus. Based on this algorithm, this paper further verifies the recently discovered weak relationship between the spectral gap and convergence rate of decentralized machine learning.

Key words: decentralized machine learning; distributed average consensus; topological design; spectral gap

去中心化计算是一种重要的分布式计算范式. 相对于中心化的分布式计算, 去中心化的特性使得训练

集群没有一个特定的参数服务器. 在每个训练轮次, 计算节点与相邻节点交换数据并在本地进行模型更新. 本

① 基金项目: 国家重点研发计划 (2020YFA0711400); 国家自然科学基金 (61831018, U21A20452)

收稿时间: 2023-02-24; 修改时间: 2023-03-30; 采用时间: 2023-04-12; csa 在线出版时间: 2023-06-30

CNKI 网络首发时间: 2023-07-03

文研究以下两种去中心化计算方法: 分布式平均共识 (distributed average consensus, 以下简称平均共识) 和去中心化机器学习 (decentralized machine learning, DML). 平均共识通过节点间数据的交换和更新使得所有节点收敛于全局平均值. 它在分布式传感器网络和无人驾驶飞行器中具有广泛的应用^[1]. DML 允许计算节点与相邻节点协作训练机器学习模型, 是平均共识的重要扩展^[2].

去中心化的特性使得以上两种方法可以很大程度上降低通信的复杂性, 但代价是计算的收敛率低于中心化的分布式训练^[3]. 要高效地进行去中心化计算, 关键在于权衡好数据通信时间和收敛率的关系. 虽然收敛率降低使得训练需要更多轮次到达目标精度, 但更短的每轮通信时间使去中心化计算可以更快地完成训练任务^[4]. 正因如此, DML 逐渐被用于训练大规模机器学习模型中^[5,6].

节点的通信拓扑在权衡训练次数和通信时间的关系中起着至关重要的作用. 稀疏的拓扑意味着更少的数据传输, 降低每个轮次的通信时间. 另一方面, 具有高收敛率的拓扑意味着到达目标精度需要更少的训练轮次. 拓扑的谱间距, 定义为拓扑权重矩阵最大两个特征值的模之差, 可以准确刻画平均共识的收敛率^[7]. 当节点间数据独立同分布时, 它也用来刻画 DML 收敛率的上界^[4]. 设计一个通信高效且具有高谱间距的拓扑可以降低训练任务到达收敛的总时间.

然而现有的训练拓扑没有针对异构网络环境进行优化. 拓扑一般模式固定或基于节点带宽相同的假设设计获得^[6,8]. 但由于网络资源的竞争, 带宽异构在分布式计算中很常见. 一般来说, 享有更高上传带宽的节点在每个训练轮次应向更多节点发送数据, 因此应有更高的出度. 节点入度同理. Underlay 网络是由连接每个节点的物理网络组成的图. 由于节点间线路故障或隐私考虑, 该图不一定是全连接的. 在 Underlay 网络之外的连接不可使用. 一个设计良好的拓扑结构应在满足上述两个约束的同时最大化谱间距.

本文研究在异构带宽和 Underlay 网络约束下最大化有向图谱间距的问题. 有向图的特性给本文带来了以下挑战: 一方面, 不对称的有向图使其权重矩阵特征值变为复值, 并且有向图的谱间距不再具有凸性; 另一方面, 拓扑设计问题本质上是一个整数规划问题, 任一条边存在与否由一个 0-1 变量表示. 随着节点数量增加, 该问题的搜索空间呈指数增长.

本文通过推导并计算谱间距针对加边或删除操作的梯度来解决以上拓扑设计问题. 通过该梯度值进行边的增删从而构建拓扑. 本文的主要工作如下.

1) 本文对异构网络环境下最大化谱间距的问题进行建模, 并推导出有向图的谱间距针对任一条边增删的梯度.

2) 本文设计了基于该梯度的算法来构建具有较大谱间距的稀疏有向拓扑. 该拓扑同时考虑了带宽异构性和底层 Underlay 网络的约束.

3) 本文通过实验验证了生成的拓扑在平均共识中具有更快的收敛率和更短的收敛时间, 且算法在不同程度的异构网络环境下能保持性能稳定. 本文将生成的拓扑应用于 DML 训练, 进一步验证了 DML 的收敛率对谱间距的弱相关性.

1 背景知识

本节介绍了平均共识和 DML 的基本概念, 以及将两种机制应用在无向图和有向图上的不同算法.

1.1 平均共识

考虑一个包含 n 个节点的图 G , 每个节点分配一个随机初始值 $x_i^{(0)}$. 去中心化平均共识算法^[1]旨在通过相邻节点交换更新数据的方式使得所有节点收敛于全局平均值. 在无向图上, 每一轮任一节点的值都会更新为其邻居以及自身值的加权求和:

$$x_i^{(t+1)} = \sum_{j=1}^n P_{ji} x_j^{(t)} \quad (1)$$

变量 P_{ji} 表示从节点 j 到节点 i 的权重, 权重矩阵 $P = [P_{ij}]$ 描述了拓扑结构的信息. 如果权重矩阵 P 是双随机, 那么它可以确保随着轮次 t 的增加, 每个节点的值将收敛到全局平均 $\sum_{i=1}^n x_i^{(0)} / n$ ^[1]. 节点间权重的选择存在多种方案^[9]. 本文采用每个节点平均分配权重给邻居节点的方式, 使得 $P_{ij} = 1/d_i^{\text{out}}$, 其中 d_i^{out} 是节点 i 的出度.

在有向图上应用平均共识需要用到 push-sum 算法^[10]. 该算法仅要求权重矩阵 P 是列随机的, 应用上述的权重平均分配方法即可满足. 直接使用式 (1) 会因有向图的非对称性使得每个节点收敛的值偏离全局平均. push-sum 算法引入了一个额外的标量 y 来纠正由有向图产生的偏差:

$$\begin{cases} x^{(t+1)} = P^T \cdot x^{(t)} \\ y^{(t+1)} = P^T \cdot y^{(t)} \\ z^{(t+1)} = x^{(t+1)} / y^{(t+1)} \end{cases} \quad (2)$$

其中, x 和 z 是每个节点的有偏和无偏估计值. 标量 y 初始值为1, 即 $y(0) = 1$.

权重矩阵的谱间距是描述平均共识收敛率的紧凑上界^[9]. 谱间距表示为 $\rho = 1 - |\lambda_2|$, 其中 $|\lambda_2|$ 是 P 第2大特征值的模, 最大特征值由于行随机性质恒定为1. 具有更大谱间距的拓扑在应用式(1)或式(2)时, 消耗更少的轮次就能在目标精度内使各个节点收敛至全局平均.

1.2 去中心化机器学习

去中心化机器学习是平均共识的重要的扩展, 它允许节点仅与邻居通信来协作训练模型. 在无向图上应用 DML 的基本算法是 Decentralized SGD (D-SGD)^[9]. 在每个训练轮次, 任一节点在本地进行梯度更新后, 将其模型与邻居节点模型进行加权求和.

$$x_i^{(t+1)} = \sum_j P_{ji} \left(x_j^{(t)} - \eta \cdot \nabla f(x_j^{(t)}) \right) \quad (3)$$

变量 $x^{(t)}$ 和 $\nabla f(x)^{(t)}$ 是模型在第 t 轮的参数和梯度, η 是训练的学习率.

在有向图上进行 DML 训练需要使用 Stochastic Gradient Push 算法^[11]. 它的核心思想是结合 push-sum 算法和随机梯度下降, 利用消除误差后的梯度更新模型参数. 本文在有向图上训练 DML 使用该算法.

由于 DML 与平均共识在训练过程的紧密联系, 谱间距也很自然地多种文献用于推导并描述 DML 的收敛率上界^[4,6]. 然而谱间距对 DML 收敛率的影响在近期开始受到一些文献的质疑. 文献 [12] 发现在节点间数据独立同分布时, 谱间距对收敛率的影响虽然仍然存在但小于预期. 最新工作^[13] 则直接表明谱间距不能准确预测 DML 的性能. 本文工作可视为该文献的同时期工作. 本文提出了构建大谱间距拓扑的算法, 并在生成的有向图上进行 DML 实验, 进一步验证了谱间距和 DML 收敛率之间仅存在弱相关性.

2 问题动机

2.1 探索包含有向图的拓扑设计

以往的拓扑设计研究主要讨论无向图和一些特殊类型的图. 无向图的权重矩阵具有实数特征值以及凸的谱间距^[14]. 这些特性使得一些文献通过求凸优化模型的精确解来设计拓扑^[15]. 尽管这些工作可以确保求得最优解, 但求解的时间复杂度使它们无法解决更大规模的问题. 因为随着节点数增加, 表示边存在的 0-1 变量使优化模型的搜索空间呈指数增长. 文献 [14] 采

用松弛技术来求解更大规模的问题, 但所得拓扑仍然是无向的.

一些特殊类型的有向图, 如环状图, 正则图和指数图也受到了不同研究的关注^[6,12]. 尽管这些图是有的, 但它们的连接具有一定的模式, 使得它们的权重矩阵是双随机的. 这些特性有助于推导这些谱间距的表达式. 但这类拓扑不一定能够满足异构网络环境的约束.

只有一般的有向图才能满足不同程度带宽异构和 Underlay 的约束, 因此需要新的算法在有向图上搜索和设计满足以上约束的高收敛率拓扑.

2.2 同时考虑带宽异构和 Underlay 网络约束

带宽的异构性表明只追求更稀疏的拓扑并不是最优的. 为了避免出现掉队者 (straggler) 和增加吞吐量, 不同节点数据传输所消耗的时间应该相似. 具有更高上传带宽的节点应该向更多的邻居发送数据, 因此具有更高的出度. 同理, 下载带宽较低的节点应该有较少的入度. 这种针对每个节点的带宽约束相比于针对总通信次数的约束^[16] 更细粒度一些.

由于网络可能出现的中断和节点间的隐私要求, Underlay 网络不能保证全连接, 某些连接可能不能使用. 文献 [8] 在设计拓扑时考虑了 Underlay 网络, 但假设不同节点带宽相同. 在 Underlay 网络约束下使用特定类型的图则必须对图进行删边, 此操作可能降低图的谱间距. 因此需要一种拓扑设计算法同时考虑这两个约束.

3 问题求解

本节首先对拓扑设计问题进行建模, 之后将谱间距针对加边或删边的梯度进行推导和计算. 基于以上结果, 本文提出了一个基于梯度的拓扑构建算法.

3.1 问题建模

给定一个包含 n 个节点的图 $G(V, E)$, 定义 0-1 变量 x_l 表示有向边 l 的存在. 若 $x_l = 1$, 则边 l 存在于图 G 中, $x_l = 0$ 则不存在. 将只包含边 l 的图的邻接矩阵定义为 A_l , 该矩阵只有一个元素为 1, 其余元素均为 0. 定义 M 为 n 个节点全连接图的边集, 则图 G 的邻接矩阵可以写成如下形式:

$$A = \sum_{l \in E} A_l = \sum_{l \in M} x_l A_l \quad (4)$$

定义 P 为图 G 的权重矩阵, 其中 P_{ij} 是从节点 i 到节点 j 的权重. 本文采用平均分配权重的方式, 每个节点的

权重平均分配给节点的外邻居 (out-neighbors), 表示为:

$$P_{ij} = \frac{1}{d_i^{\text{out}}}, \text{ if } l_{ij} \in E \quad (5)$$

变量 d_i^{out} 代表节点 i 的出度. 定义 D_{out} 为图的对角出度矩阵, 则权重矩阵可以用以上符号表示如下:

$$P = D_{\text{out}}^{-1} \cdot A = \text{diag} \left(\frac{1}{d_1^{\text{out}}}, \dots, \frac{1}{d_n^{\text{out}}} \right) \cdot \sum_l x_l A_l \quad (6)$$

本文的目标是通过对每条边的变量 x_l 进行赋值来最大化权重矩阵的谱间距. 拓扑设计问题建模如下:

$$\max_{x_l, \forall l \in M} 1 - |\lambda_2(P)| \quad (7)$$

$$\text{s.t. } d_i^{\text{in}} = \left[\begin{array}{c} d_m^{\text{in}} \\ b_m^{\text{in}} \end{array} \right], 1 \leq i \leq n \quad (8)$$

$$d_i^{\text{out}} = \left[\begin{array}{c} d_m^{\text{out}} \\ b_m^{\text{out}} \end{array} \right], 1 \leq i \leq n \quad (9)$$

$$d_m^{\text{out}} = \left[\begin{array}{c} d_m^{\text{out}} \\ b_m^{\text{out}} \end{array} \right] \quad (10)$$

$$x_l = 0, \text{ if } l \in H \quad (11)$$

$$\text{图}G\text{是强连通图} \quad (12)$$

约束(8)–(10)源于节点间的带宽异构性. 节点的出度和入度应与上传和下载带宽成正比, 使得数据传输消耗相似的时间. 这种方式可以避免 straggler 现象并增加吞吐量. 每个节点的带宽是一个已知的值, 可以通过节点主动探测或节点带宽分配获得. 常量 d_m^{out} 和 d_m^{in} 是图 G 预先定义的最大出度和入度. 具有最大上传带宽 b_m^{out} 的节点具有最大出度 d_m^{out} . 同理, 具有最大入度 d_m^{in} 的节点有最大的下载带宽. 其他节点的出入度由最大度数和节点带宽与最大带宽之比决定.

假设节点 i 的上传带宽为60%的 b_m^{in} , 下载带宽为40%的 b_m^{out} , 且 $d_m^{\text{out}} = d_m^{\text{in}} = 5$, 则节点 i 的出入度计算为 $d_i^{\text{in}} = 3$ 和 $d_i^{\text{out}} = 2$. 通过这种方式, 节点间的带宽异构约束转化为图的度数序列约束.

约束(11)源于给定 Underlay 网络的限制. 定义 H 为不可用连接的集合, 在这集合中的边不在拓扑设计考虑之中. 最后, 图的强连通性保证了训练的收敛性.

3.2 梯度计算

与无向图不同, 有向图权重矩阵的谱间距关于 x_l 是非单调且非凸的^[14]. 在有向图中添加新边并不代表可以提升谱间距, 反之亦然.

对以上模型进行直接求解并不可行. 首先, 0-1 变量 x_l 带来了随节点数指数增长的搜索空间. 另一方面, 基于凸优化的松弛技巧并不适用于该非凸模型. 用于图实现问题的 Havel-Hakimi 算法^[17]可根据特定的度数序列生成对应的有向图, 只要该度数序列是可实现的. 然而该算法并不感知谱间距和底层 Underlay 网络, 因而不能保证生成图的质量. 本文通过计算第2大特征值的模对 x_l 的梯度, 即 $\partial|\lambda_2|/\partial x_l$, 来求解这个问题. 该值代表着特定一条边的增删对谱间距的影响.

为了能够推导出梯度的表达式, 本文首先引入矩阵微扰理论的相关概念. 矩阵微扰理论的目标是估计矩阵受到微扰时其对应矩阵函数的变化^[18]. 在拓扑设计问题中, 微扰对应特定边的添加或删除, 特征值则是对应的矩阵函数. 通过微扰理论, 本文可以估计拓扑权重矩阵受到边的增删影响时特征值的变化.

将矩阵 P 的某一特征值写作 $\lambda(P)$, 则该矩阵受到微扰后的特征值可表示为 $\lambda(P + x\Delta P)$, 其中 x 是一个标量, ΔP 代表微扰. 当 $\lambda(P)$ 是简单特征值时, 扰动的特征值对于 x 是无穷可微的, 受到微扰后特征值的梯度可以表示为^[19]:

$$\frac{\partial \lambda}{\partial x} = \frac{v_0^H \frac{\partial P}{\partial x} u_0}{v_0^H u_0} \quad (13)$$

其中, v_0 和 u_0 分别代表 $\lambda(P)$ 的左右特征向量.

因为简单特征值对 x 是无穷可微的, 理论上可以继续推导受扰动特征值的二阶梯度. 但求解二阶梯度的计算量过于庞大且在实际应用中精度不高^[19], 因此本文仅使用一阶梯度.

获得式(13)之后, 本文的目标则是推导出 $\partial P/\partial x_l$ 的表达式. 根据式(6), 权重矩阵 P 针对某条边 x_l 的梯度可以分为两部分. 其中一部分是对邻接矩阵的梯度. 根据式(4), 邻接矩阵可视为图 G 所有有向边的求和. 因此邻接矩阵对特定边 l 的梯度只与这条边单独的邻接矩阵有关, 表示为:

$$\frac{\partial A}{\partial x_l} = \frac{\partial \sum_{l \in M} x_l A_l}{\partial x_l} = A_l \quad (14)$$

梯度的另一部分涉及对角出度矩阵 D_{out} 针对任意一条边的梯度. 注意到每个节点的出度可表示为从该点出发所有 0-1 变量 x_l 的和, 因此任一节点出度 d_i^{out} 对 x_l 的梯度可以表示为:

$$\frac{\partial d_i^{\text{out}}}{\partial x_l} = \begin{cases} 1, & \text{如果 } x_l \text{ 起点为 } i \\ 0, & \text{其他情况} \end{cases} \quad (15)$$

注意到任意一条边的存在只能影响该边起点的出度. 因此, 出度矩阵针对某一条边 x_l 的梯度只对该矩阵中起点相同的元素有影响. 结合式 (6), 式 (14) 和式 (15), 权重矩阵 P 对于 x_l 的梯度可以表示如下:

$$\frac{\partial P}{\partial x_l} = \text{diag}\left(0, \dots, \frac{-1}{d_i^2}, \dots, 0\right) \cdot A + D_{\text{out}}^{-1} \cdot A_l \quad (16)$$

本文的目标是计算特征值的模对特定边的梯度, 即 $\partial|\lambda|/\partial x_l$. 注意到特征值的模由实部和虚部组成, 对其所求梯度可以按实部和虚部分成两部分:

$$\begin{aligned} \frac{\partial |\lambda|}{\partial x_l} &= \frac{\partial \sqrt{\text{Re}(\lambda)^2 + \text{Im}(\lambda)^2}}{\partial x_l} \\ &= \frac{\text{Re}(\lambda)}{|\lambda|} \frac{\partial \text{Re}(\lambda)}{\partial x_l} + \frac{\text{Im}(\lambda)}{|\lambda|} \frac{\partial \text{Im}(\lambda)}{\partial x_l} \\ &= \frac{\text{Re}(\lambda)}{|\lambda|} \text{Re}\left(\frac{\partial \lambda}{\partial x_l}\right) + \frac{\text{Im}(\lambda)}{|\lambda|} \text{Im}\left(\frac{\partial \lambda}{\partial x_l}\right) \end{aligned} \quad (17)$$

将式 (17) 与式 (13), 式 (16) 结合, 并定义 $D_i^{-2} = \text{diag}\left(0, \dots, \frac{-1}{d_i^2}, \dots, 0\right)$, 最终的梯度表示如下:

$$\begin{aligned} \frac{\partial |\lambda|}{\partial x_l} &= \frac{\text{Re}(\lambda)}{|\lambda|} \text{Re}\left(\frac{v_0^H \frac{\partial P}{\partial x_l} u_0}{v_0^H u_0}\right) + \frac{\text{Im}(\lambda)}{|\lambda|} \text{Im}\left(\frac{v_0^H \frac{\partial P}{\partial x_l} u_0}{v_0^H u_0}\right) \\ &= \frac{\text{Re}(\lambda)}{|\lambda|} \text{Re}\left(\frac{v_0^H (D_i^{-2} \cdot A + D^{-1} \cdot A_l) u_0}{v_0^H u_0}\right) \\ &\quad + \frac{\text{Im}(\lambda)}{|\lambda|} \text{Im}\left(\frac{v_0^H (D_i^{-2} \cdot A + D^{-1} \cdot A_l) u_0}{v_0^H u_0}\right) \end{aligned} \quad (18)$$

3.3 算法设计

式 (18) 表示了添加或删除某一条边对权重矩阵谱间距的影响. 可以基于该等式, 向一个基底图 G_{base} 依次添加新边来求解问题 (7). 每个轮次更新可用的边集 E , 计算当下权重矩阵的特征值和特征向量, 并使用式 (18) 来计算所有可用边添加到现有图时的梯度. 因为优化目标是最大化谱间距 $\rho = 1 - |\lambda_2|$, 所以每个轮次具有最小梯度值的边会添加到现有图中. 算法 1 描述了构建拓扑的流程. 建模时列出的约束体现在可用边集 E 中, 它排除了会使节点超过指定度数或处于 Underlay 网络之外的边.

式 (18) 的结构虽然看起来复杂, 但它的实际计算

复杂度并不高. 其中计算特征值和特征向量的时间复杂度为 $O(n^3)$, n 为图 G 的节点数, 但在每个添删边轮次只需将它们计算一次即可. 计算该等式中的二次型 $v_0^H (D_i^{-2} \cdot A + D^{-1} \cdot A_l) u_0$ 只需要 $O(n)$ 的时间复杂度. 因为 D_i 和 A_l 都是只有一个非零元素的稀疏矩阵.

算法 1. 基于梯度的拓扑构建算法

输入: 图 $G=G_{\text{base}}$, 约束集合 C , 可用边集 E .
输出: 构建后的图 G .

```

1. while  $E \neq \emptyset$  do
2.   计算图  $G$  权重矩阵的特征值以及特征向量
3.   for edge  $x_l \in E$  do
4.     基于式 (18) 计算梯度
5.   end for
6.   选择梯度值最小的边  $\hat{l}$ 
7.    $G = G \cup \hat{l}$ 
8.   基于新边  $\hat{l}$  和约束  $C$  更新可用边集  $E$ 
9. end while

```

算法 1 还可加入进行删边的模块, 如经过固定轮次周期性的删边, 来帮助算法进一步避免局部最优值.

基础图 G_{base} 可选择为一个强连通且尽量稀疏的有向图. 本文选择环形拓扑作为基础图. 当给定的 Underlay 网络不是全连接图时, 基础图可能存在一些不可用的边. 在这种情况下, 算法可以在加边过程完成后删除这些不可用边, 然后继续加边步骤, 直到可用边集变空为止. 另一方面, 模型给出的度数序列可能没有对应有向图. 在这种情况下, 算法依然把可用边集 E 为空作为停止条件, 最终获得一个尽量接近该度数序列的拓扑.

4 实验分析

本节通过实验验证基于梯度的拓扑设计算法性能. 首先通过实验测试该算法在最大化谱间距上的效果. 然后将该算法生成的拓扑部署在平均共识和 DML 两种场景上, 来验证这些拓扑能否加快收敛率和训练时间. 为了更好地验证算法 1 的有效性, 本文选择了 3 种基线对比方法.

- 指数图^[6]: 其中每个节点仅与距离 2^n 个节点的邻居相连. 该图中每个节点都有 $O(\log(n))$ 个邻居.

- Havel-Hakim 算法: 该算法是基于度数序列的图实现算法^[17]. 只要该度数序列是可实现的, 该算法就可以给出一个有向图作为可行解.

- 基于贪心的启发式方法: 在每个轮次, 该算法向现有图中添加使谱间距最大化的边来构造拓扑.

4.1 具有对数度数的拓扑

指数图在具有 $O(\log(n))$ 度数的情况下有较大的谱间距,是常用的训练拓扑^[6].本文首先在度数与指数图相同的情况下,测试基于梯度的算法1性能.因此每个节点的入度和出度为 $d = \lfloor \log_2(n-1) \rfloor$.该度数设置可以视作不同节点具有相同的带宽,各节点在每一轮向 d 个邻居发送数据.对于 n 个节点,所有可能的连接构成了

一个全连接图.参数 q 控制着该全连接图中有多少比例的连接不可用,体现了 Underlay 网络的约束.指数图或 Havel-Hakimi 算法生成的图中不可用的边将被移除.

不同算法生成拓扑的谱间距随节点数的变化如图1所示.算法1在绝大部分情况下具有更大的谱间距.在节点数增大的情况下这种优势更为明显.且该算法生成拓扑的谱间距在节点数量增加时可以保持稳定.

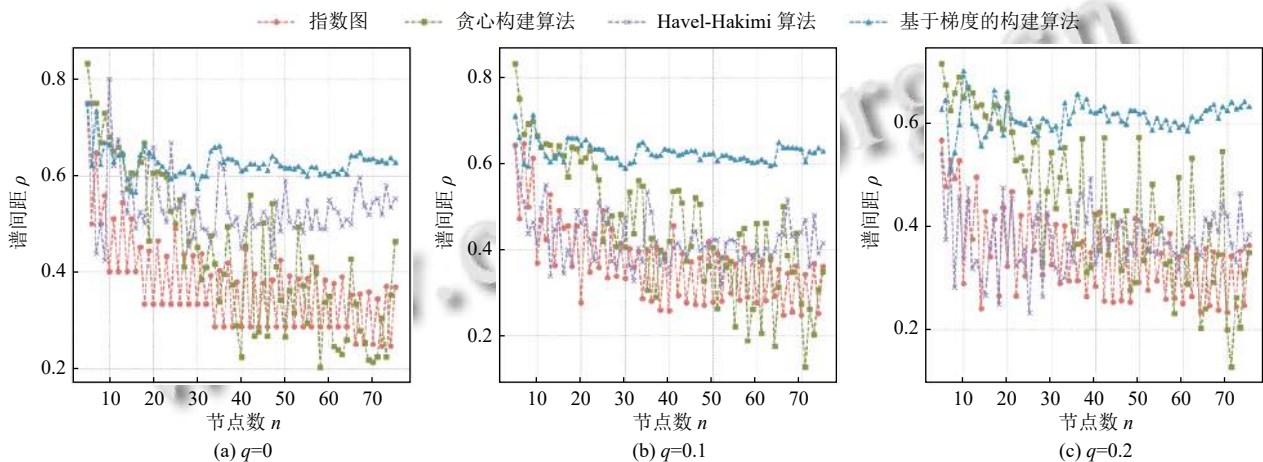


图1 不同算法的谱间距随节点数变化图

Underlay 约束产生的效果可在图1中明显观察到.随着 q 值的增加,可用的连接数逐渐减少,确定性方法如指数图或 Havel-Hakimi 算法面临更多删边的可能.而删边操作使图的谱间距变得不稳定,有较大的概率使谱间距变小.由图可见,贪心方法和 Havel-Hakimi 算法受到 q 的影响后性能恶化较多,而基于梯度的算法在不同 q 值的影响下性能依然保持稳定.

本文接下来测试以上拓扑在平均共识上的性能,采用设定如下:各节点具有一个100维向量,向量中每个元素的初始值从 $U(1,100000)$ 均匀分布中采样获得.为了使最后结果保持稳定,所有实验数据是100次独立实验的平均值.

在 $q = 0.2$, $d = \lfloor \log_2(n-1) \rfloor$ 的条件下,不同算法生成的30节点拓扑在平均共识上的收敛情况如图2(a)所示.由于该实验假设节点具有相同带宽,本文认为各节点通信时间相同,而计算时间仅包括求平均步骤可忽略不计.因此定义每一轮的通信时间为1单位时间.由图2可见共识过程的误差与迭代次数大致呈对数线性关系,且权重矩阵的谱间距准确刻画了不同拓扑的收敛率.基于梯度的算法生成了谱间距最大的拓扑,它

的收敛速度最快.在均方误差为 10^{-2} 的目标精度下,该拓扑的收敛速度比其他算法快至少24%.

4.2 同时考虑带宽和 Underlay 约束

本文接下来对基于梯度的算法在带宽异构的情景下进行性能测试,并同时考虑 Underlay 网络对算法性能的影响.实验中带宽异构的定义如下:设上传和下载带宽相同, $d_m^{\text{out}} = d_m^{\text{in}} = 5$. 1/3 节点具有60%的最大带宽,另1/3节点具有80%的最大带宽,其余节点则拥有最大带宽.根据式(7)~式(11),这3种节点的入度和出度应分别为3、4和5,每种节点占总节点数的比例相同.训练集群中不可用连接的比例用参数 q 控制.

在上述异构网络环境的约束下,不同算法的性能如图3所示.对于确定性的对比算法,度数序列的约束则转化为节点的随机删边,最后的结果是100次随机删边的最优结果.由于有向图的谱间距对于增删边并不单调,删边对于拓扑的影响是不确定的.在异构带宽和 Underlay 网络的共同作用下,对比方法的性能由于删边而出现了不稳定的波动.随着 q 值的增大,对比方法的谱间距波动情况愈发严重. Havel-Hakimi 算法由于对底层 Underlay 并不感知,对其结果删除不可用的

边可能会使该算法无法生成强连通图. 图中 $\rho = 0$ 的数据点体现了该现象. 随着 q 值增加, 该算法还产生了很多 ρ 接近 0 的数据点. 而基于梯度的算法不仅能产生具有更大谱间距的拓扑, 且其性能随着节点数和 q 值的增加可以保持稳定.

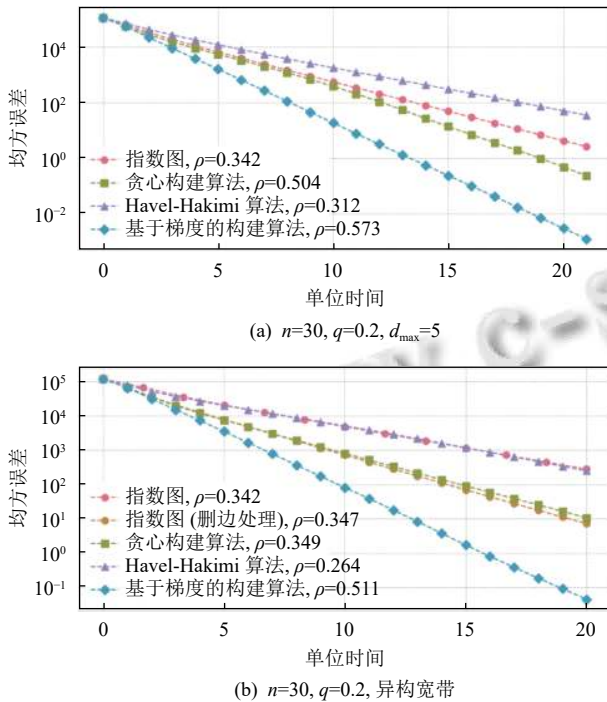


图2 不同拓扑的平均共识收敛情况

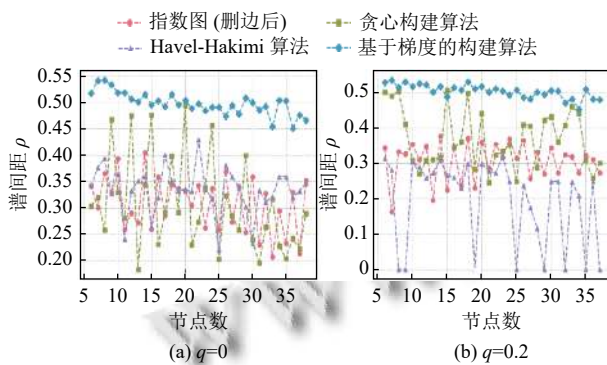


图3 异构网络环境下不同算法性能比较

在以上异构网络环境约束下, 本文测试 30 个节点的不同拓扑在平均共识下的收敛情况. 定义各节点在给定度数下的每轮通信时间为 1 单位时间. 由于 30 个节点的指数图度数均为 5, 因此其带宽最少的一组节点传输数据时间为 $5/3$ 单位时间. 由于该组节点占总结点数的 $1/3$, 节点整体的传输时间会立即受到它们的限制. 因此指数图在该种情况下完成一次数据传输需要

$5/3$ 单位时间.

在上述约束下, 不同拓扑的平均共识过程如图 2(b) 所示. 虽然指数图具有较小的谱间距, 但因其未考虑节点带宽异构性, 它随时间的收敛速度只能和谱间距小于它的图实现算法相似. 指数图同样可以通过随机删边的方法来应对异构网络环境的约束. 但经过删边的指数图在谱间距上没有太大变化. 而基于梯度的加边算法一方面考虑到带宽的异构性, 在传输时间上优于指数图这种固定模式的拓扑. 另一方面, 与其他可以适应度数序列的方法相比, 算法 1 可产生具有更大谱间距的拓扑, 因此在随训练轮次的收敛率上也占优. 基于梯度的算法无论在所需传输时间还是传输轮次上都少于其他对比方法.

4.3 谱间距对 DML 的影响

该部分研究谱间距对 DML 收敛率的影响. 机器学习实验的具体配置如下: 本文使用 CIFAR-10 数据集^[20], ResNet-18 残差神经网络^[21], 并使用 Stochastic Gradient Push 算法进行训练^[11]. 拓扑节点数为 17, 节点间的数据是独立同分布的, batch size 为 128, 学习率为 0.1, 使用不带动量的随机梯度下降算法. 本文通过程序仿真的方法来模拟集群 DML 训练的过程.

实验带宽异构配置如下: 17 个节点的集群中各有 6 的节点具有 4 和 5 的出入度, 其余 5 个节点具有 3 的出入度, 参数 $q = 0$. 在该实验环境下, 节点的模型训练时间依然远小于模型传输时间. 定义具有 b_{max} 的节点所消耗的模型传输时间为 1 个单位时间. 在该配置下使用不同算法生成的拓扑训练模型, 其损失函数随时间变化的曲线如图 4 所示. 由于指数图出入度均为 5, 在以上带宽异构约束下每轮消耗 $5/3$ 单位时间, 其随时间的收敛速度明显慢于其他方法. 虽然剩下的拓扑在谱间距上差距较大且每轮消耗时间相同, 但它们的收敛曲线却有明显重合的现象. 这与之前谱间距调节 DML 收敛率的结论相矛盾.

为了更加深入地探讨谱间距对 DML 收敛率的影响, 本文利用所有节点出入度 $d = 5$ 和 $d = 4$ 两种场景生成了更多的拓扑, 并使用这些拓扑进行 DML 训练. 通过多种谱间距不同的拓扑来验证 DML 收敛率和谱间距的关系. 在各种拓扑上训练 25 个 epoch 后, 不同拓扑的平均损失函数值如图 5 所示. 该损失函数值由前后 5 个 batch 的损失值取平均所得. 相比于其他图, 环状图的谱间距远小于其他拓扑, 且有着最高的损失值,

符合理论预期. 但是其他拓扑的数据点之间会出现谱间距增大, 损失函数值反而增大的现象. 由此可见谱间距并不能准确预测 DML 的收敛性能.

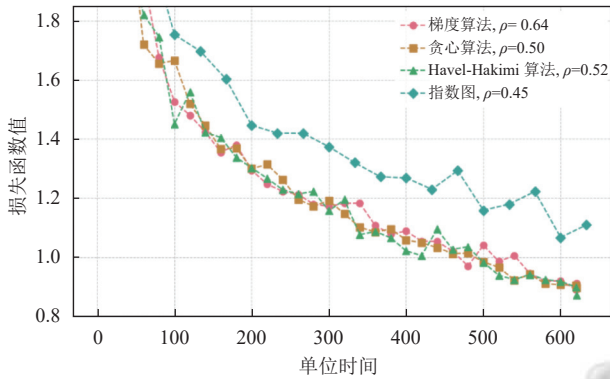


图4 不同拓扑的 DML 训练损失随时间变化图

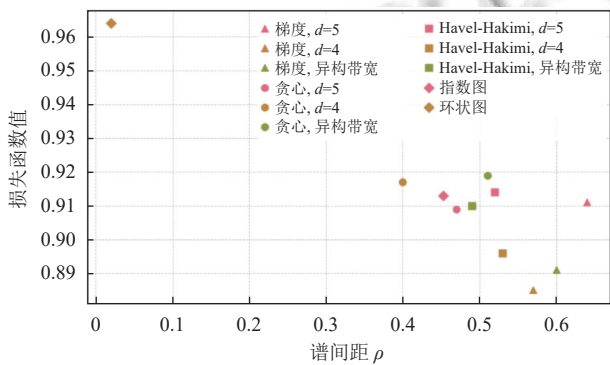


图5 不同拓扑的谱间距与训练损失散点图

关于以上现象, 本文推测谱间距仅渐进地约束了 DML 的收敛率. 谱间距作为 DML 收敛率的上界并不是时刻紧凑的. 该上界的微小变化并不能准确反映真实收敛率的改变. 从 DML 的更新过程也可以理解该现象. 在 DML 的每个迭代轮次, 每个节点都会收到其他节点更新后的模型, 该更新版本的模型比前面轮次的模型对 DML 整体性能贡献更大. 但这个更新后的模型只经历了一步平均共识, 这种一次性平均共识的效果与谱间距不具有十分紧密的联系.

5 结语

本文推导并计算了谱间距针对增删边的梯度, 基于该梯度提出拓扑构建算法解决了异构网络环境下最大化谱间距的拓扑设计问题. 该算法具有较低的时间复杂度, 能够生成谱间距更大的拓扑. 生成的拓扑在平均共识过程中的误差收敛速度显著快于其他拓扑. 该算法在不同程度的异构网络环境下性能保持稳定. 该

算法在增删边数量极少的场景下可能不优于直接遍历, 由于搜索空间较小.

应用该算法, 本文生成大量拓扑来验证谱间距对 DML 收敛率的影响: 谱间距并不能准确预测 DML 收敛率的快慢. 虽然谱间距的大幅变化对收敛率的影响符合预期, 但小幅增加谱间距不能保证收敛率提升. 本文推测这是由于 DML 与平均共识的具体训练过程差异造成的.

参考文献

- Olfati-Saber R, Murray RM. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 2004, 49(9): 1520–1533. [doi: 10.1109/TAC.2004.834113]
- Nedic A, Ozdaglar A. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 2009, 54(1): 48–61. [doi: 10.1109/TAC.2008.2009515]
- Li M, Andersen DG, Smola A, *et al.* Communication efficient distributed machine learning with the parameter server. *Proceedings of the 27th International Conference on Neural Information Processing Systems*. Montreal: MIT Press, 2014. 19–27.
- Lian XR, Zhang C, Zhang H, *et al.* Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach: Curran Associates Inc., 2017. 5336–5346.
- Yuan K, Chen YM, Huang XM, *et al.* DecentLaM: Decentralized momentum SGD for large-batch deep training. *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal: IEEE, 2021. 3009–3019.
- Ying BC, Yuan K, Chen YM, *et al.* Exponential graph is provably efficient for decentralized deep training. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS, 2021. 13975–13987.
- Gerencsér B, Gerencsér L. Tight bounds on the convergence rate of generalized ratio consensus algorithms. *IEEE Transactions on Automatic Control*, 2022, 67(4): 1669–1684. [doi: 10.1109/TAC.2021.3067629]
- Marfoq O, Xu C, Neglia G, *et al.* Throughput-optimal topology design for cross-silo federated learning. *Proceedings of the 34th International Conference on Neural*

- Information Processing Systems. NIPS, 2020. 19478–19487.
- 9 Koloskova A, Loizou N, Boreiri S, *et al.* A unified theory of decentralized sgd with changing topology and local updates. Proceedings of the 37th International Conference on Machine Learning. PMLR, 2020. 5381–5393.
- 10 Tsianos KI, Lawlor S, Rabbat MG. Push-sum distributed dual averaging for convex optimization. Proceedings of the 51st IEEE Conference on Decision and Control (CDC). Maui: IEEE, 2012. 5453–5458.
- 11 Assran M, Loizou N, Ballas N, *et al.* Stochastic gradient push for distributed deep learning. Proceedings of the 36th International Conference on Machine Learning. Long Beach: PMLR, 2019. 344–353.
- 12 Neglia G, Xu C, Towsley D, *et al.* Decentralized gradient methods: Does topology matter? Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics. PMLR, 2020. 2348–2358.
- 13 Vogels T, Hendrikx H, Jaggi M. Beyond spectral gap: The role of the topology in decentralized learning. Proceedings of the 36th International Conference on Neural Information Processing Systems. New Orleans: NIPS, 2022.
- 14 Ghosh A, Boyd S. Growing well-connected graphs. Proceedings of the 45th IEEE Conference on Decision and Control. San Diego: IEEE, 2006. 6605–6611.
- 15 Dai R, Mesbahi M. Optimal topology design for dynamic networks. Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference. Orlando: IEEE, 2011. 1280–1285.
- 16 Wang JY, Sahu AK, Yang ZY, *et al.* MATCHA: Speeding up decentralized SGD via matching decomposition sampling. Proceedings of the 6th Indian Control Conference (ICC). Hyderabad: IEEE, 2019. 299–300.
- 17 Havel V. A remark on the existence of finite graphs. Casopis Pest Mathematics, 1955, 80: 477–480.
- 18 Stewart GW, Sun JG. Matrix Perturbation Theory. Boston: Academic Press, 1990.
- 19 Magnus JR. On differentiating eigenvalues and eigenvectors. Econometric Theory, 1985, 1(2): 179–191. [doi: [10.1017/S0266466600011129](https://doi.org/10.1017/S0266466600011129)]
- 20 Krizhevsky A, Nair V, Hinton G. The CIFAR-10 dataset. <http://www.cs.toronto.edu/~kriz/cifar.html>. (2009-09-23) [2023-02-26].
- 21 He KM, Zhang XY, Ren SQ, *et al.* Deep residual learning for image recognition. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778.

(校对责编: 孙君艳)