采用多数投票的 Sparse-SignSGD：面向通信高效的分布式学习

Chanho Park，*IEEE* 学生会员，以及 Namyoon Lee，*IEEE* 高级会员

摘要—通过分布式优化，可以显著提高复杂深度学习模型的训练效率。然而，在迭代过程中，工作节点与参数服务器之间的大量通信成本往往会成为这一过程的瓶颈。为解决这一问题，本文提出一种新型通信高效算法，兼具稀疏化和符号量化的协同优势，称为S^3 GD-MV。S^3GD-MV 的工作节点选取局部梯度向量中幅值最大的前 K 个分量，并仅向服务器发送这些分量的符号。随后，服务器汇总这些符号，并通过多数投票规则返回结果。我们的分析表明，在某些温和条件下，如果根据工作节点数量和深度学习模型的规模适当选择稀疏化参数 K，S^3GD-MV 可以达到与 signSGD 相同的收敛速度，同时显著降低通信成本。基于独立同分布（IID）和非独立同分布数据集的实验结果表明，S^3GD-MV 的准确率高于 signSGD，同时显著降低了通信成本。这些发现凸显了S^3GD-MV 作为深度学习领域通信高效分布式优化方案的潜力。

索引词——分布式优化、梯度压缩、收敛速率分析

arXiv:2302.07475v1 [cs.LG] 15 Feb 2023

1 引言

DISTRIBUTED 随机梯度下降（SGD）是一种广泛用于解决大规模数据并行优化问题的方法 [2], [3]。理论上，同步分布式随机梯度下降可使高维模型的训练速度随工作节点数量线性提升 [4]。然而，实现此类分布式随机梯度下降面临诸多挑战，其中包括在中央服务器与工作节点之间交换梯度信息所需的高通信成本。通信成本会随工作节点数量线性增长。例如，最先进的图像分类模型 Coca [5],拥有 21 亿个参数，每次迭代都需要工作节点与中央服务器之间交换 8.4GB 信息，因此在通信网络受限时，分布式训练难以承担如此高昂的开销。为解决这一通信瓶颈，必须开发通信高效的分布式学习算法，在最大限度降低通信成本的同时保持较高的学习性能。

近年来，通信高效的分布式学习算法领域取得了重大进展。降低每次迭代通信成本的一种常见方法是梯度量化，例如 QSGD [6], ，其中每个工作节点都会将本地计算的随机梯度量化为有限数量的比特。另一个例子是 signSGD，这是一种简单而有效的算法，仅将本地计算的梯度向量的符号发送给工作节点 [7]。这种单比特量化信息出人意料地具有理论收敛保证可用于优化广泛的非凸损失函数。研究表明，signSGD 能够达到与具有完整梯度分辨率的传统随机梯度下降相同的收敛速度。

梯度稀疏化是实现通信高效的分布式学习的另一种方法，其中每个工作节点选择梯度中幅值最大的分量，并将其发送给服务器。已有研究表明，与 QSGD 相比，该方法可将通信成本降低一到两个数量级，同时仍能为优化非凸损失函数提供理论保证 [8]。近期的发展包括将梯度稀疏化与量化相结合，以进一步降低通信成本 [9], [10], [11], [12]。

在本文中，为了延续通过稀疏化与量化实现协同增益的理念，我们提出了一种名为 Sparse-Sign-随机梯度下降 with the majority vote 的分布式学习算法，即 S^3GD-MV。不同于先前工作 [7],，我们表明，在多数投票聚合下，只更新局部梯度向量中少量选定分量的符号，相较于更新梯度的完整符号信息更有利于提高测试准确率并节省通信成本，前提是根据工作节点数量和模型大小明智地选择稀疏度。这一观察结果有悖直觉，因为与 signSGD 相比，该方法能够以更低的通信成本实现更高的测试准确率。我们通过给出一些示例，对这一结果进行了清晰的解释。更重要的是，我们证明了这种采用多数投票聚合的激进梯度信息压缩更新方法，对于广泛的一类非凸损失函数具有理论收敛保证。

1.1 相关工作

理论文献中已有大量研究旨在探讨通信高效的分布式

• *C. Park* 就职于浦项科技大学电气工程系，韩国浦项，邮编 *37673*。电子邮箱：*chanho26@postech.ac.kr*。

• *N. Lee* 就职于高丽大学电气工程学院，韩国首尔，邮编 *02841*。电子邮箱：*namyoon@korea.ac.kr*。

学习。可以说，先前工作大致分为三大类：i）量化，ii）稀疏化，以及 iii）稀疏化与量化相结合。

量化： 梯度量化能够有效降低分布式学习中的通信成本。随机量化技术因具有无偏性而备受关注，即压缩后的随机梯度的均值近似等于真实均值 [6], [10]。为了大幅压缩梯度，[13], [14] 最初提出了 1-bitSGD，即采用一位量化并结合误差补偿机制的随机梯度下降。实验表明，与全精度对应方法相比，该方案几乎不会损失准确率，同时显著降低了成本。采用多数投票的 signSGD 在量化方法上不同于 1-bitSGD [7]，[15]。与 1-bitSGD 不同，signSGD 直接取随机梯度的符号，而不使用误差补偿机制。这会导致随机梯度的估计量有偏。尽管梯度存在偏差，研究表明该方法仍具有理论收敛保证，并支持工作节点与服务器之间双向通信的压缩。signSGD 还有一些变体，例如误差反馈方法 [16], [17]。此外还有许多其他量化方法：压缩梯度差 [18], [19],、向量量化 [20], [21],和自适应量化[22], [23]。这些一位量化方法的主要局限在于，通信成本最多只能降低 32 倍。例如，优化包含 21 亿个模型参数的 Coca 时，每个工作节点每次迭代都需要发送大小为 0.52 GB 的一位量化梯度向量。因此，使用低带宽通信网络训练超大规模模型可能难以满足需求。

稀疏化： 梯度稀疏化是另一种梯度压缩技术，可以大幅降低通信成本 [24], [25]。其思路是，每个工作节点在每次迭代中选取梯度中幅值最大的 K 个分量，并将它们发送给服务器。该技术结合误差反馈或累积机制实现，以确保所有坐标都能在需要时获得更新机会 [8], [26]。研究表明，通过采用适当的带内存误差累积方法，稀疏化随机梯度下降能够达到与原始 SGD 相同的收敛速率。Top-K 稀疏化的一个缺点是，除了所选 K 参数值的全精度信息外，还需要额外通信来编码稀疏模式。[27] 提出了一种基于梯度方差的随机稀疏化方法，从而增强了 Top-K 稀疏化的效果。在上述方法的基础上，人们还提出了许多改进方法：采用动量 [28],草图方法来寻找梯度中的重要坐标 [29], [30], ，以及自适应稀疏化 [31], [32]。与此同时，人们也提出了若干 Top-K 稀疏化方法的变体，以降低实现复杂度。例如，为避免不适用于图形处理器的排序操作，研究者提出了其他改进的 Top-K 稀疏化方法，以较低复杂度提升学习性能 [25], [33], [34],[35]。

稀疏化与量化相结合： 与本文最相关的先前工作是将稀疏化与量化结合起来。 [9] 提出了一种名为 Qsparse-local-SGD 的分布式学习算法。Qsparse-local-SGD 的思路是，将梯度稀疏化、量化与误差补偿以及本地计算结合起来。 [11], [12] 也提出了与 [9], 非常相似的学习方法，但它们考虑的是下行链路压缩，即压缩来自服务器的模型更新。这些算法与我们的算法相似，但其一位量化级别是根据非零值的经验均值计算的。此外，这些算法没有给出收敛速率的证明 [11], [12]。另一项相关工作见于 [10],，其中梯度被强力量化为三个级别 $\{-1, 0, 1\}$。这种名为 TernGrad 的梯度压缩技术采用概率稀疏化和量化机制，以保持无偏期望。这一点不同于我们用于梯度压缩的 Top-K 稀疏化与确定性符号量化方法。此外，我们的算法还采用多数投票聚合，以降低从服务器到工作节点的通信成本，而 Qsparse-local-SGD 和 TernGrad 均未考虑这一点。

1.2 贡献

我们考虑一个通用的分布式学习问题，其中一组分布式的 M 工作节点独立执行计算，利用本地存储的数据集，将模型大小为 N 的经验风险损失函数最小化。

- 我们首先提出一种分布式随机梯度下降算法，将前K 稀疏化和符号量化与多数投票聚合相结合，称为S^{3}GD-MV。据我们所知，这是首个将前K 稀疏化、朴素符号量化与多数投票聚合共同用于分布式优化的算法。得益于激进的梯度量化，S^{3}GD-MV在每次迭代中与服务器交换的通信量最多仅为 $M\left[K + K\log_2\binom{N}{K} + N\right]$ 比特。由于signSGD所需的通信成本为 $2MN$，S^{3}GD-MV能够实现比signSGD更高的压缩增益，并且随着稀疏度 K的降低，这一增益会更加显著。

- 我们的主要贡献是，在一些温和条件下，为使用S^{3}GD-MV优化一大类非凸损失函数提供理论收敛保证。由于稀疏化、符号量化和多数投票聚合的综合影响，对S^{3}GD-MV的收敛速率进行严格的理论分析颇具挑战。通过推广 [7], 中提出的分析工具，我们根据关键算法参数确立了收敛速率，主要涉及稀疏度 K、工作节点数量 M和模型大小 N。我们的关键发现是，只要根据工作节点数量和模型大小适当选择稀疏度，即使通信成本大幅降低，收敛速率在阶数上仍可与标准signSGD相同。具体而言，我们推导出

使收敛速率上界最小的最优稀疏度水平为 $\frac{K}{N} = cM^{\frac{3}{2}}$，其中 c为某个正常数。我们用“代议制民主的力量”这一原则解释这一反直觉的结果。

- 我们还给出了采用随机-K 规格化的S^{3}GD-MV 的收敛速率。基于这一结果，我们从理论上解释了与随机-K 规格化方法相比，top-K 稀疏化如何提升学习性能。

- 通过仿真，我们验证了S^{3}GD-MV 相较于现有算法的性能提升。我们使用 MNIST 和 CIFAR-10 数据集训练卷积神经网络和 ResNet-56 模型。我们观察到，在某种设置下，与原始 SGD、top-K SGD 和 signSGD 相比，S^3 GD-MV 可将通信成本分别显著降低约 300 倍、200 倍和 10 倍，同时保持相同的测试准确率。

- 这是我们会议论文的扩展版本 [1]，，其中为所提出算法S^{3}GD-MV 的收敛速率提供了详细证明。第 5 节中的证明未包含在 [1]中。我们还指出， [1]并未提供采用随机-K 稀疏化的S^{3}GD-MV 的收敛速率证明。此外，借助丰富的仿真结果，我们说明了S^{3}GD-MV 的学习性能如何随数据分布、通信成本、稀疏化算子和超参数而变化；这些内容在 [1]中均未涉及。

算法1S^{3}GD-MV

输入： 初始模型 $\mathbf{x}^0$,学习率 δ^t，梯度稀疏度 K，工作节点数量 M，初始累积误差 $\mathbf{e}_m^0 = \mathbf{0}$，误差权重 η，总迭代次数 T

对于 $t = 0: \ T-1$ **执行**

  对于工作节点 $m = 1: \ M$ **执行**

    计算 $\tilde{\mathbf{g}}_m^t$ (局部梯度)

    $\mathbf{g}_m^t \leftarrow \tilde{\mathbf{g}}_m^t + \eta \mathbf{e}^t{}_m$

    $\mathbf{e}_m^{t+1} \leftarrow \mathbf{g}_m^t - \mathsf{TopK}\left(\mathbf{g}_m^t\right)$

    发送 $\mathsf{TopKSign}(\mathbf{g}_m^t)$ **至服务器**

    从服务器接收 $\mathsf{sgn}\left[\sum_{m=1}^M \mathsf{TopKSign}(\mathbf{g}_m^t)\right]$ 来自 **服务器**

    $\mathbf{x}^{t+1} \leftarrow \mathbf{x}^t - \delta^t \text{符号}\left[\sum_{m=1}^M \mathsf{TopKSign}\left(\mathbf{g}_m^t\right)\right]$

  结束循环

结束循环

2 S^{3}GD-MV 算法

本节简要介绍一个分布式学习问题，并在下文中通过一些示例说明所提出的 S^{3}GD-MV 算法。

2.1 预备知识

设 $\mathbf{x} \in \mathbb{R}^N$ 为大小为 N 的模型参数向量。机器学习的数学表述可定义为以下优化问题:

$$\min_{\mathbf{x}\in\mathbb{R}^N} f(\mathbf{x}) := \mathbb{E}_{\xi\sim\mathcal{D}}\left[F(\mathbf{x};\xi)\right], \tag{1}$$

其中，ξ 是从数据集 $\mathcal{D}$ 中抽取的随机样本，$F(\cdot)$ 是特定于问题的经验损失函数，可以是凸函数，也可以是非凸函数。在包含 M 个工作节点的分布式学习场景中，式（1）的优化问题归结为

$$\min_{\mathbf{x}\in\mathbb{R}^N}\left[f(\mathbf{x}) := \frac{1}{M}\sum_{m=1}^M f_m(\mathbf{x})\right], \tag{2}$$

其中，$f_m(\mathbf{x}) = \mathbb{E}_{\xi_m\sim\mathcal{D}_m}\left[F_m(\mathbf{x};\xi_m)\right]$ 是工作节点 $m \in [M]$ 上的局部损失函数，它根据分配给工作节点 m, $\mathcal{D}_m$ 的那部分数据集中的随机数据样本 ξ_m 得到。

我们引入两种用于梯度量化的操作。top-K 稀疏化算子选取向量 $\mathbf{u} \in \mathbb{R}^N$ 中绝对值最大的 K 个分量。该操作可通过阈值为 ρ 的指示函数定义如下:

$$\mathsf{TopK}(\mathbf{u}) = \mathbb{1}_{\{|\mathbf{u}|\geq\rho\}}, \tag{3}$$

其中，当 $\mathcal{A}$ 为真时，$\mathbb{1}_{\mathcal{A}} = 1$；否则，$\mathbb{1}_{\mathcal{A}} = 0$。阈值 ρ 可取为按绝对值排序后第 K 大分量与第 $(K+1)$ 大分量之间的任意值。将该 top-K 稀疏化算子与符号量化结合，我们定义算子 $\mathsf{TopKSign}(\cdot): \mathbb{R}^N \to \{-1,0,+1\}^N$，它将 $\mathbf{u} \in \mathbb{R}^N$ 映射为一个稀疏度参数为 $\gamma = \frac{K}{N} \in [0,1]$ 的 N 维三元向量，其定义如下:

$$\mathsf{TopKSign}\left(\mathbf{u}\right) = \mathsf{sgn}\left(\mathsf{TopK}(\mathbf{u})\right), \tag{4}$$

其中 $\mathsf{sgn}(\mathbf{u}) = \mathbb{1}_{\{\mathbf{u}>0\}} - \mathbb{1}_{\{\mathbf{u}<0\}}$。所有操作均按元素逐一进行。

2.2 算法

随机梯度计算： 对于第 t 次迭代的给定模型参数，记为 $\mathbf{x}^t$，工作节点 $m \in [M]$ 计算局部梯度 $\tilde{\mathbf{g}}_m^t := \nabla F_m(\mathbf{x}^t, \xi_m^t) \in \mathbb{R}^N$其中，$\xi_m^t$ 是从第 t 次迭代时的数据集 $\mathcal{D}_m$ 中随机抽取的小批量。请注意，由于对小批量数据进行采样，局部计算出的梯度是一个随机向量。

带稀疏化误差补偿的梯度更新： 工作节点 $m \in [M]$ 通过将先前迭代中累积的稀疏化误差加到随机梯度 $\tilde{\mathbf{g}}_m^t$ 上来更新它。$\mathbf{e}_m^t$，即

$$\mathbf{g}_m^t = \tilde{\mathbf{g}}_m^t + \eta \mathbf{e}_m^t, \tag{5}$$

其中，$\mathbf{e}_m^t$ 表示在迭代 t时添加的误差补偿，它包含先前迭代中 top-K 稀疏化未选中的随机梯度分量之和。每个工作节点在迭代 t 时使用$\mathbf{g}_m^{t-1}$更新此记忆项，方式如下:

$$\mathbf{e}_m^t = \mathbf{g}_m^{t-1} - \mathsf{TopK}\left(\mathbf{g}_m^{t-1}\right). \tag{6}$$

在（5）中，$\eta > 0$ 表示一个控制误差补偿项影响的权重参数。增大 η可以增强稀疏化误差项的作用，从而使梯度分量在各坐标上的选择更加均匀。当 $\eta = 0$时，不需要内存，并且$\mathbf{g}_m^t = \tilde{\mathbf{g}}_m^t$。

梯度压缩： 工作节点 $m \in [M]$ 使用式（4）中定义的 $\mathsf{TopKSign}(\mathbf{g}_m^t) \in \{-1,0,1\}^N$对$\mathbf{g}_m^t$进行压缩。为选出幅值最大的 K 个分量，阈值 $\rho_m^t(K)$ 取在第 K 个

最大分量与幅值第$(K+1)$大的分量之间。随后，通过通信网络将压缩后的向量 $\mathsf{TopKSign}(\mathbf{g}_m^t)$发送至服务器。由于$\|\mathsf{TopKSign}(\mathbf{g}_m^t)\|_0 = K$，每个工作节点到服务器的通信成本约为每个通信轮次 $K + K\log_2\binom{N}{K}$ 比特。

多数投票：服务器执行多数投票以进行梯度聚合。令 $\mathcal{N}_m^t = \{n \in [N] : \mathsf{TopKSign}(g_{m,n}^t) \neq 0\}$ 为工作节点 m 在第 t次迭代时选出的非零支撑集。然后，我们可以将 $\mathcal{N}_m^t$ 的并集定义为 $\mathcal{N}^t = \bigcup_{m=1}^M \mathcal{N}_m^t$，其补集为 $\mathcal{N}_c^t = [N]/\mathcal{N}^t$。接着，对非零支撑集的并集 $n \in \mathcal{N}^t$应用多数投票原则。而对于 $n \in \mathcal{N}_c^t$，服务器保留零值并将其发送回所有工作节点。

模型更新：每个工作节点以学习率 δ^t 更新模型参数，更新方式如下：

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \delta^t \cdot \mathsf{sgn}\left[\sum_{m=1}^M \mathsf{TopKSign}\left(\mathbf{g}_m^t\right)\right]. \tag{7}$$

从服务器到工作节点的通信成本每次迭代最多为 $|\mathcal{N}^t| + |\mathcal{N}^t|\log_2\binom{N}{|\mathcal{N}^t|} \leq N$ 位。

整个过程总结于算法1中。

2.3 稀疏化与符号量化的协同增益：代表性民主的力量

为什么更激进的压缩可能优于单纯的符号压缩？为了理解这一点，考察一些能够阐明在多数投票聚合下同时利用稀疏化和符号量化所带来协同优势的启发性示例是很有帮助的。

示例1：考虑 $M = 3$ 的情况。假设 $\mathbf{g}_m^t$ 的第 n 个分量在第 t 次迭代时分别生成为 $g_{1,n}^t = 3$、$g_{2,n}^t = -0.3$，以及 $g_{1,n}^t = -0.03$。如果三个工作节点以全精度将其发送至服务器，则聚合梯度的符号变为正，即 $\mathsf{sign}\left(\frac{1}{3}\sum_{m=1}^3 g_{m,n}^t\right) = +1$。应用 signSGD 后，三个工作节点分别发送 $\mathsf{sign}\left(g_{1,n}^t\right) = +1$，$\mathsf{sign}\left(g_{2,n}^t\right) = -1$，以及 $\mathsf{sign}\left(g_{3,n}^t\right) = -1$。随后，根据多数投票，服务器沿着错误方向聚合符号，因为 $\mathsf{sign}\left(\sum_{m=1}^3 \mathsf{sign}\left(g_{m,n}^t\right)\right) = -1$。只要每个工作节点仅在确信其数值时参与符号投票，就可以纠正这一符号错误。具体而言，我们规定工作节点 m 在 $|g_{m,n}^t| > 1$ 时发送符号信息。于是，工作节点1只能向服务器发送符号 $\mathsf{sign}(g_{1,n}^t) = +1$，而其余两个工作节点不参与投票。因此，聚合后的符号变为 $\mathsf{sign}\left(\mathsf{sign}(g_{1,n}^t)\right) = +1$，与聚合梯度的真实符号一致，从而纠正了 signSGD 的符号错误。该示例清楚地表明，在进行符号量化之前，保留幅值最大的 K 个分量进行稀疏化，有利于解码聚合梯度的真实符号。这一原则可以解释为“代表性民主的力量”。这与遵循“民主的力量”原则的 signSGD 明显不同。

示例2：从示例1可以明显看出，稀疏化和量化的协同增益。然而，过度的稀疏化可能会损失这一增益。我们通过一个著名的**“球与箱子问题”**来剖析这一关键问题。假设有大量工作节点，并且每个工作节点仅使用符号量化发送幅值最大的分量。假设由工作节点选定的分量在 $\{1, 2, \ldots, N\}$中独立且均匀分布，其中 N为模型大小。根据球与箱子原理，给定 M 个工作节点，工作节点m 参与更新第 n个分量的概率为

$$\mathbb{P}\left[m \text{ worker} \to n\text{th component}\right] = \frac{1}{N}. \tag{8}$$

因此，第 n 个梯度分量未被更新（即对应的箱子为空）的概率为

$$\mathbb{P}\left[n\text{th component is empty}\right] = \left(1 - \frac{1}{N}\right)^M \simeq e^{-\frac{M}{N}}. \tag{9}$$

这一现象解释了为什么过度稀疏化会导致学习性能显著下降，因为许多坐标在每次迭代中都不太可能得到更新。为解决这一问题，我们需要增大稀疏度 K，以确保在每个梯度坐标上都有一个或多个工作节点以较高概率参与。假设每个工作节点在 $\{1, 2, \ldots, N\}$ 上均匀选择 K 个坐标。那么，第 n 个梯度分量为空的概率为

$$\mathbb{P}\left[n\text{th component is empty}\right] = \left(1 - \frac{K}{N}\right)^M \simeq e^{-\frac{KM}{N}}. \tag{10}$$

因此，算法应根据工作节点数量 M 谨慎选择稀疏度 K，以确保每个训练轮次中所有坐标都得到充分调度，例如取 $M^\ell K = cN$，其中 $c > 0$ 和 $\ell \leq 1$ 为某些常数。

3 收敛速率分析

本节分析非凸损失设定下 S^{3}GD-MV 的收敛速率。我们首先说明推导分析结果所需的一些相关假设，然后介绍关于收敛速率的主要结果。

3.1 假设

假设 1（下界）。对于所有 $\mathbf{x} \in \mathbb{R}^N$ 以及某些局部极小点 $\mathbf{x}^\star$，我们有一个目标值

$$f(\mathbf{x}) \geq f\left(\mathbf{x}^\star\right) = f^\star. \tag{11}$$

该假设对于收敛到局部极小点是必要的。

假设 2（坐标平滑性）。对于所有 $\mathbf{x}$，$\mathbf{y} \in \mathbb{R}^N$，存在一个具有非负常数 $\mathbf{L} = [L_1, \cdots, L_N]$ 的向量，使其满足

$$|f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle| \leq \sum_{n=1}^N \frac{L_n}{2}\left(y_n - x_n\right)^2. \tag{12}$$

该假设表明，目标函数满足坐标逐分量利普希茨条件。

假设3（无偏随机梯度与有限方差）。 在迭代 t 时，经过误差补偿的随机局部梯度 $\mathbf{g}_m^t$ 是无偏的，并且 $\mathbf{g}_m^t$ 的每个分量都有一个非负常数 $\boldsymbol{\sigma} = [\sigma_1, \cdots, \sigma_N]$ 给出的有限方差上界，即 $\forall m \in [M]$, $\forall n \in [N]$,

$$\mathbb{E}\left[\mathbf{g}_m^t\right] = \bar{\mathbf{g}}_m^t, \quad \mathbb{E}\left[\left(g_{m,n}^t - \bar{g}_{m,n}^t\right)^2\right] \leq \sigma_n^2. \tag{13}$$

假设 4（非零支撑集的均匀采样）。 $\forall m \in [M]$, $\forall n \in [N]$，带有误差补偿的 top-K 算子在迭代 $t \in [T]$ 时均匀选择非零支撑集，即

$$\mathbb{P}\left[\mathsf{TopKSign}\left(g_{m,n}^t\right) \neq 0\right] = \gamma, \tag{14}$$

其中 $\gamma = \frac{K}{N}$。

这一假设表明，采用误差补偿技术的 top-K 选择能够以轮询方式更新每个坐标，从而确保均匀采样。特别是，适当调整记忆权重参数 η 可以使这一均匀采样假设更加准确。我们将在第 4 节通过仿真结果验证这一假设。我们强调，这一假设有助于使收敛性证明在数学上更易于处理。即使不满足这一均匀采样假设，也就是说，在非均匀采样的情况下，所提出的算法仍能以更低的通信成本实现比现有算法更高的学习性能；我们将在第 4 节对此进行验证。

3.2 SS^{3}GD-MV 的收敛速率

基于上述假设，我们给出理论结果。在呈现主要结果之前，我们先介绍一些用于证明收敛速率结果的重要引理。随后，我们将通过与 signSGD [7] 的收敛速率进行比较，对该收敛速率作一些解读。

引理 1. 对于 $m \in [M]$ 和 $n \in [N]$，*top-K* 阈值参数 $\rho_{m,n}^t(\gamma)$ 的下界为

$$\rho_{m,n}^t(\gamma) \geq \frac{\epsilon}{\sqrt{\gamma}} \left|\bar{g}_n^t\right|. \tag{15}$$

对于某个 $\epsilon \geq 0$。

证明。 参见第 5.1 节。 □

该引理表明，阈值参数 $\rho_{m,\,n}^t(K)$ 可以表示为稀疏度参数 γ 和真实梯度均值的幅值 $|\bar{g}_n^t|$ 的多项式函数。根据假设 4，梯度分量由 TopKSign 均匀选取，因此我们可以推导出采用逐元素阈值 $\rho_{m,\,n}^t(\gamma)$ 时的收敛速率。该阈值随真实梯度均值 $|\bar{g}_n^t|$ 增大而增大，同时与稀疏度 $\gamma = \frac{K}{N}$ 成反比。例如，若选择较大的 K，阈值会相应减小，这与我们的直觉一致。该引理为建立符号解码错误概率上界奠定了基础，该上界将在下一个引理中给出。

引理 2. 当 $M = 1$ 时，S^3*GD-MV* 引起的符号翻转错误上界为

$$\mathbb{P}\left[\mathsf{TopKSign}\left(g_{m,n}^t\right) \neq \mathsf{sign}\left(\bar{g}_n^t\right)\right] \leq \frac{1}{\sqrt{B^t}\left(1+\frac{\epsilon}{\sqrt{\gamma}}\right)} \frac{\sigma_n}{|\bar{g}_n^t|}, \tag{16}$$

其中，B^t 是迭代 t 时的小批量大小。

证明。 参见第 5.2 节。 □

引理 2 表明，可以通过增大小批量大小 B^t 或降低稀疏度 γ（即，较小的 K）来减少符号翻转误差。此外，提高信噪比（SNR） $\frac{|\bar{g}_n^t|^2}{\sigma_n^2}$ 可以减少解码误差。我们强调，我们的解码误差界是对 [7] 中结果的推广，因为我们在符号量化的基础上考虑了 top-K 选择操作。

引理 3. 令 $\mathcal{M}_n^t$ 为在迭代 t 时发送随机梯度第 n 个分量的工作节点集合，其定义为

$$\mathcal{M}_n^t = \left\{m \in [M] : \mathsf{TopKSign}\left(g_{m,n}^t\right) \neq 0\right\} \tag{17}$$

其基数为 $M_n^t \triangleq |\mathcal{M}_n^t|$。那么，$M_n^t$ 服从二项分布 $\mathsf{B}(M, \gamma)$，即

$$\mathbb{P}\left[M_n^t = u\right] = \binom{M}{u} \gamma^u (1-\gamma)^{M-u}. \tag{18}$$

for $u \in \{0, 1, \ldots, M\}$.

证明。 根据假设 4，每个工作节点都以概率 $\gamma = \frac{K}{N}$ 独立地采样每个坐标。因此，M_n^t 服从二项分布 $\mathsf{B}(M, \gamma)$。 □

引理 4. 令 $p_{m,n}^t = \mathbb{P}\left[\mathsf{TopKSign}(g_{m,n}^t) \neq \text{符号}\ (\bar{g}_n^t)\right]$。那么，在给定 $M_n^t = U$ 的条件下，S^3*GD-MV* 在迭代 t 时的符号解码错误上界为

$$\mathbb{P}\left[\mathsf{sgn}\left(\sum_{m \in \mathcal{M}_n^t} \mathsf{TopKSign}\left(g_{m,n}^t\right)\right) \neq \mathsf{sign}\left(\bar{g}_n^t\right) \middle| M_n^t = U\right] \leq \left[4(1-p_{m,n}^t)p_{m,n}^t\right]^{\frac{U}{2}}. \tag{19}$$

证明。 参见第 5.3 节。 □

该引理表明，只要选择相同坐标的工作节点数量呈指数增长，符号解码错误就会降低。增加工作节点数量 M_n^t 可以进一步提升这一效果。此外，结果表明，采用 top-K 稀疏化会降低解码误差，因为随着 γ 减小，符号翻转概率 $p_{m,\,n}^t$ 也会降低，正如引理 2 所证明的那样。因此，$p_{m,\,n}^t$ 与 M_n^t（$K\,\gamma$）之间存在一个有趣的权衡。具体而言，增大 K 可以增加获得较大 M_n^t 的机会，从而提升编码增益；而增大 K 也会使 $p_{m,\,n}^t$ 降低，因为 $\rho_{m,\,n}^t(K)$ 会变小。总之，我们必须谨慎选择 K，才能兼得两方面的优势。关于符号解码错误的这一分析结果，完全印证了我们对如何选择 K 以在示例 1 和 2 中获得协同增益的高层次解释。

借助上述引理，我们现在可以给出主要结果，该结果陈述如下。

定理 1（小批量 S^{3}GD-MV 的非凸收敛速率）。给定学习率 $\delta^t = \frac{1}{\sqrt{T\|\mathbf{L}\|_1}}$、批量大小 $B^t = T$ 和常数 $0 \leq \epsilon \leq 1$，S^3*GD-MV* 的收敛情况如下：

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|\bar{\mathbf{g}}^t\|_1\right] \leq \frac{1}{\sqrt{T}}\left[\sqrt{\|\mathbf{L}\|_1}\left(\frac{f^0-f^\star}{\alpha(M,\gamma)}+\frac{1}{2}\right)+\frac{\beta(M,\gamma)}{\alpha(M,\gamma)}\frac{2}{1+\frac{\epsilon}{\sqrt{\gamma}}}\|\boldsymbol{\sigma}\|_1\right], \tag{20}$$

其中 $\gamma = \frac{K}{N}$，$\alpha(M,\ \gamma) = 1 - (1-\gamma)^M$ 且

$$\beta(M,\gamma) = \sum_{u=1}^{M}\frac{1}{\sqrt{u}}\binom{M}{u}\gamma^u(1-\gamma)^{M-u}. \tag{21}$$

证明。参见第 5.4 节。 □

3.3 特殊情况

我们的收敛速率分析表明，随着迭代次数 T 的增加，S^{3}GD-MV 保证收敛到一个驻点，其收敛速率为 $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ 阶。为了更好地理解这一结果，我们通过对重要算法参数进行特化，进一步给出关于定理 1 的几点说明。

3.3.1 无稀疏化情形（$\gamma = 1$）

当 $\gamma = 1$ 时，即在压缩 $N = K$ 中不进行稀疏化，$\alpha(M,\ 1) = 1$ 以及 $\beta(M,\ 1) = \frac{1}{\sqrt{M}}$。在这种情况下，我们的算法等价于 signSGD，并且在 $\epsilon = 0$ 下推导出的收敛速率与 [7] 中的速率表达式一致。因此，我们的结果根据稀疏化参数 $0 < \gamma < 1$ 推广了先前工作 [7] 的收敛速率分析。

3.3.2 单工作节点情况（$M = 1$）

当只有一个工作节点（$M = 1$）时，我们观察到 $\|\mathbf{L}\|_1$ 和 $\|\boldsymbol{\sigma}\|_1$ 的常数项表现不同。$\|\mathbf{L}\|_1$ 项以 $\mathcal{O}\left(\frac{N}{K}\right)$ 的速率增加，而 $\|\boldsymbol{\sigma}\|_1$ 项以 $\mathcal{O}\left(\sqrt{\frac{K}{N}}\right)$ 的速率变化。这一观察表明，可能存在一个能够实现最高收敛速率的最优稀疏度。

3.3.3 M 较大但 γ 较小

最重要的情形是选择较小的 γ 和足够大的 M。在这种情况下，参数渐近地变为 $\alpha(M,\ \gamma) \approx M\gamma = \frac{MK}{N}$ 和 $\beta(M,\ \gamma) \approx M\gamma = \frac{MK}{N}$。这些近似意味着 $\|\mathbf{L}\|_1$ 和 $\|\boldsymbol{\sigma}\|_1$ 的常数项分别为 $\mathcal{O}\left(\frac{N}{MK}\right)$ 和 $\mathcal{O}\left(\sqrt{\frac{K}{N}}\right)$ 阶。

在此情况下，稀疏度参数 $\gamma = \frac{K}{N}$ 在改变收敛速率方面发挥着重要作用。为了加快收敛速率，我们刻画了最优稀疏度参数 γ，该参数使定理 1 中收敛速率上界达到最小值，具体表述如下 corollary。

表1 signSGD 和S^{3}GD-MV 的收敛速率（$\gamma \ll 1$）

工作节点	SIGNSGD		S^{3}GD-MV	
	$\|\mathbf{L}\|_1$	$\|\boldsymbol{\sigma}\|_1$	$\|\mathbf{L}\|_1$	$\|\boldsymbol{\sigma}\|_1$
$M = 1$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}\left(\frac{N}{K}\right)$	$\mathcal{O}\left(\sqrt{\frac{K}{N}}\right)$
$M \geq 2$	$\mathcal{O}(1)$	$\mathcal{O}\left(\frac{1}{\sqrt{M}}\right)$	$\mathcal{O}\left(\frac{N}{MK}\right)$	$\mathcal{O}\left(\sqrt{\frac{K}{N}}\right)$
速率：		$\times \frac{1}{\sqrt{T}}$		

推论 1（S^{3}GD-MV 的最优稀疏度）。对于足够小的 $\gamma \ll 1$，使收敛速率上界最小的最优稀疏度水平为

$$\gamma^\star = \left(\frac{\epsilon\left(f^0-f^\star\right)}{M}\frac{\sqrt{\|\mathbf{L}\|_1}}{\|\boldsymbol{\sigma}\|_1}\right)^{\frac{2}{3}}. \tag{22}$$

证明。见第5.5节。 □

这意味着，为了提高收敛速率，我们需要在保持 $M^{\frac{2}{3}}K \simeq N$ 的同时选择 K 的最小可能值。这与我们在示例2中的解释一致：算法必须选择足够大的 K 值，以便以高概率在每次迭代中更新梯度的所有分量，从而获得协同增益。表1总结了与 signSGD 的收敛速率比较。

3.4 随机-K 稀疏化的影响

此外，比较使用由一位量化构成的随机-K 稀疏化技术与使用 TopKSign 时的收敛速率，也很有启发性。我们将这一操作定义为 RandKSign，其形式如下：

$$\mathsf{RandKSign}\left(\mathbf{g}_m^t\right) = \mathsf{sgn}\left(\mathsf{RandK}\left(\mathbf{g}_m^t\right)\right). \tag{23}$$

将算法1中的 TopKSign 算子替换为 RandKSign，即可修改 S^{3}GD-MV 算法。关键区别在于，该修改后的算法在选择随机梯度的 K 个分量时，不需要执行误差累积过程。完整步骤总结于算法2中。

以下定理给出了该修改后算法的收敛速率。

定理2. 给定学习率 $\delta^t = \frac{1}{\sqrt{T\|\mathbf{L}\|_1}}$ 和批量大小 $B^t = T$，S^3*GD-MV* 使用 RandKSign 算子时的收敛形式为

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|\bar{\mathbf{g}}^t\|_1\right] \leq \frac{1}{\sqrt{T}}\left[\sqrt{\|\mathbf{L}\|_1}\left(\frac{f^0-f^\star}{\alpha(M,\gamma)}+\frac{1}{2}\right)+2\frac{\beta(M,\gamma)}{\alpha(M,\gamma)}\|\boldsymbol{\sigma}\|_1\right]. \tag{24}$$

算法2S^{3}GD-MV（随机-K 版）

输入： 初始模型 $\mathbf{x}^0$,学习率 δ^t，梯度稀疏度 K, 工作者数量 M , 总迭代次数 T

对于 $t = 0, \ldots, T-1$ **执行**

  对于工作节点 $m = 1, \ldots, M$ **执行**

    计算 $\mathbf{g}_m^t$ (局部梯度)

    发送 $\text{RandKSign}(\mathbf{g}_m^t)$ **至服务器**

    从服务器接收 $\text{sgn}\left[\sum_{m=1}^M \text{RandKSign}(\mathbf{g}_m^t)\right]$

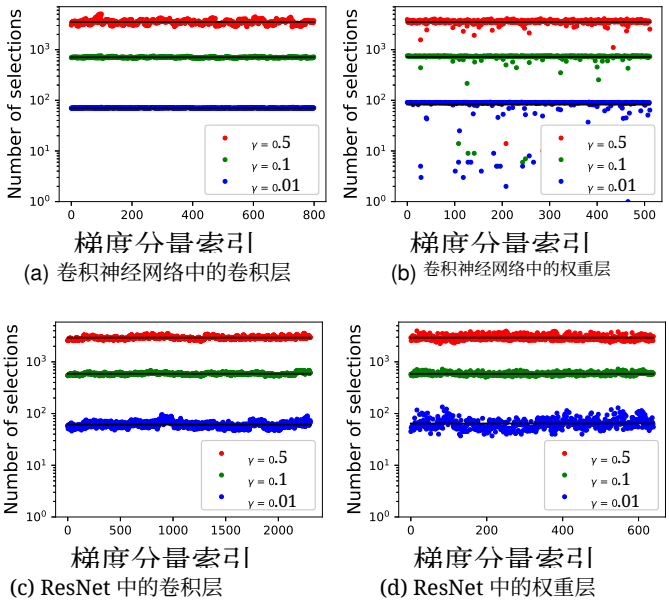
    $\mathbf{x}^{t+1} \leftarrow \mathbf{x}^t - \delta^t \text{符号}\left[\sum_{m=1}^M \text{RandKSign}\,(\mathbf{g}_m^t)\right]$

  结束循环

结束循环

表2 分布式学习算法

算法	总通信成本
原始 SGD [2]	$[32MN + 32MN] \times T$
Top-K 随机梯度下降	$\left(M\left[32K + K\,\text{对数}_2\binom{N}{K}\right] + 32MN\right) \times T$
内存 [8]	
signSGD-MV [7]	$(MN + MN) \times T$
稀^{3}GD-MV	$\left(M\left[K + K\log_2\binom{N}{K}\right] + MN\right) \times T$



(a) 卷积神经网络中的卷积层 (b) 卷积神经网络中的权重层

(c) ResNet 中的卷积层 (d) ResNet 中的权重层

图 1. 在用于 MNIST 的卷积神经网络（顶部）和用于 CIFAR10 的 ResNet（底部）的某一层中，各梯度分量被选择的次数。

证明。 参见第 5 节.6。 □

与定理 1 中的收敛速率相比，关键区别在于缺少了 $\|\boldsymbol{\sigma}\|_1$ 中的项 $\sqrt{1+\frac{2}{\epsilon}}$。因此，学习性能会下降，因为 $\|\boldsymbol{\sigma}\|_1$ 不会随 K变化。该结果表明，为了获得稀疏化与符号量化的协同增益，在梯度稀疏化过程中选择可靠的 K 分量至关重要。我们将在第 4 节通过仿真结果验证这种学习性能损失。

4 仿真结果

本节中，我们提供仿真结果，以验证S^{3}GD-MV在广泛使用的基准数据集上的性能。我们从测试准确率和通信成本方面考察其性能，并将其与现有的通信高效的分布式学习算法进行比较。接下来，我们将介绍实验设置和结果。

4.1 仿真环境

数据集与分布： 我们使用了两个常用的基准数据集 MNIST 和 CIFAR10 [36], [37], , 它们广泛用于分类实验。MNIST 和 CIFAR10 数据集分别包含 60,000 张和 50,000 张训练图像，且两个数据集均有 10,000 个测试图像样本，分为十个类别。虽然对 CIFAR10 使用了随机裁剪和水平随机翻转等数据增强方法，但实验中未对 MNIST 使用数据增强方法。

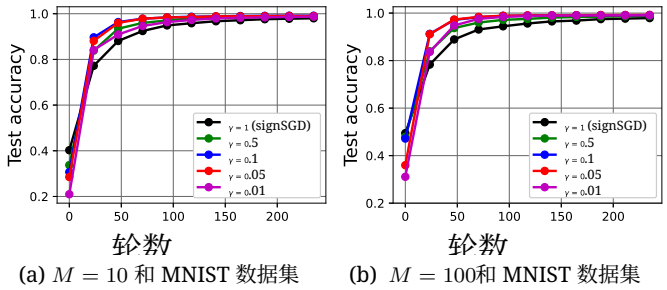
我们考虑两种不同的数据分布。对于独立同分布（IID）数据分布，我们随机将数据样本分配给各个工作节点。相比之下，对于非 IID 数据集，我们将单个类别的数据分配给特定工作节点，因此各工作节点之间的数据分布存在异质性。

模型： 为验证所提出算法的性能，我们考虑了两种常用的神经网络模型：卷积神经网络（CNN） [36] 和 ResNet-56[38]。CNN 是一种浅层网络，由五个卷积层和三个全连接层组成。CNN 的模型大小设为 $N = 5 \times 10^5$。与此同时，模型大小为 $N = 8.5 \times 10^5$ 的 ResNet-56 是一种用于评估大规模数据集性能的典型网络模型。

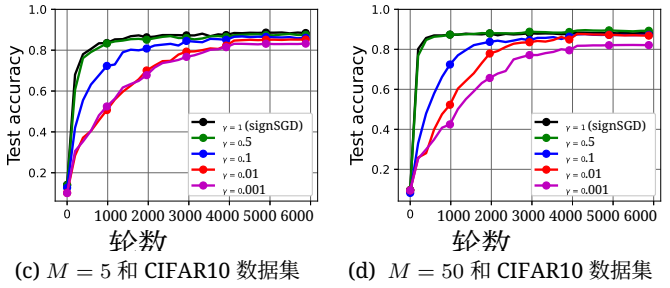
基准： 我们考虑以下四种基准分布式学习算法：

- 原始 SGD [2]: 这是一种基线分布式学习算法。在每次迭代中，每个工作节点与服务器以 32 位分辨率交换随机梯度。因此，原始 SGD 每轮的通信成本为 $2 \times (32NM)$ 位。
- 带内存的 Top-K SGD [8]: 该算法的通信成本最多为 $M\left[32K + K\log_2\binom{N}{K}\right] + 32NM$ 比特。与从工作节点到服务器的梯度更新所需的 SGD 通信量相比，该算法实现了压缩增益。
- 带多数投票的符号随机梯度下降 [7]: 该算法所需的通信成本为 $2MN$ 比特，与原始 SGD 相比减少了 32 倍。
- S^{3}GD-MV: 我们的算法每个训练轮次最多需要 $M\left[K + K\log_2\binom{N}{K}\right] + MN$。它通过稀疏化和一位量化实现了梯度压缩增益。

为了优化学习性能，我们分别为前两种和最后两种算法仔细选择了学习率 $\delta^t \in \{10^{-1}, 10^{-3}\}$。我们将研究误差权重和动量的影响。



(a) $M = 10$ 和 MNIST 数据集　(b) $M = 100$和 MNIST 数据集



(c) $M = 5$ 和 CIFAR10 数据集　(d) $M = 50$ 和 CIFAR10 数据集

图 2. 不同工作节点数量 M 下，独立同分布数据分布的测试准确率与训练轮次的关系。

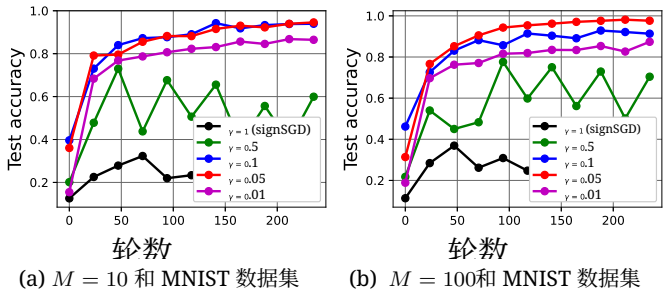
4.2 均匀采样假设的验证

为了验证我们关于在 top-K 稀疏化中均匀采样非零支撑集的假设（假设 4），我们给出数值结果。如图 1 所示，我们统计了每个梯度分量索引在 30 个训练周期内被 top-K 稀疏化选中的次数，并将其作为经验概率质量函数。数值结果表明，经过仔细选择误差权重参数 η，top-K 稀疏化能够保持非零支撑集的均匀采样特性。这一结果支持假设 4。

4.3 γ 和 M 对独立同分布数据集的影响

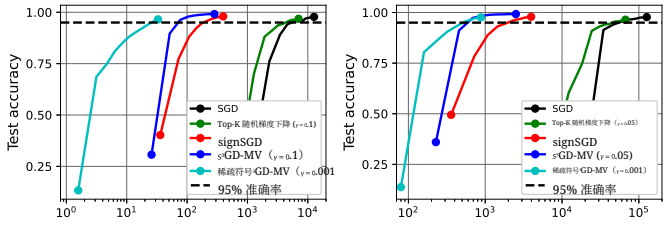
稀疏度和工作节点数量对测试准确率的影响如图 2 所示。顶部的图展示了使用独立同分布的 MNIST 数据集优化卷积神经网络时，S^{3}GD-MV 在 $M \in \{10, 100\}$ 条件下的测试准确率。当 $\gamma = 1$ 时，使用 signSGD 作为基准。对于 MNIST 数据集，我们的算法在所有 γ和 M情况下都能优于 signSGD，同时显著降低成本。有趣的是，通过谨慎选择 $\gamma = \frac{K}{N}$，使其满足 $M^{\frac{2}{3}}K \simeq N$（如推论 1 所推导），我们可以加快收敛速率。例如，随着工作节点数量增加，S^{3}GD-MV 的最优稀疏度应当降低。结果表明，对于 $M = 10$ 和 100，$\gamma = 0.1$ 和 0.05 的稀疏度是最优选择。这与推论 1 中的发现一致。

底部的图展示了使用 CIFAR10 数据集优化 ResNet-56 模型时，所提算法在工作节点数量为 $M \in \{5, 50\}$ 时的测试准确率。由于 CIFAR10 数据集比 MNIST 数据集复杂得多，因此运行 S^{3}GD-MV 算法时，对于给定的 M，我们需要为 ResNet-56 使用比卷积神经网络更高的 γ，以加快学习性能。正如推论 1 所预期的，当 $\gamma = 0.5$ 和 $M = 50$ 时，可以获得更高的测试准确率，甚至优于 signSGD。这



(a) $M = 10$ 和 MNIST 数据集　(b) $M = 100$和 MNIST 数据集

图 3. 不同工作节点数量下，非独立同分布数据分布中的测试准确率与训练轮数的关系 M。



(a) $M = 10$ 和 MNIST 数据集　(b) $M = 100$和 MNIST 数据集

图 4. 测试准确率与总通信成本的关系。我们比较了独立同分布数据设置下的四种分布式学习算法。

一结果表明，我们的算法提高了测试准确率，并显著降低了通信成本。这与定理 1 中的分析一致。

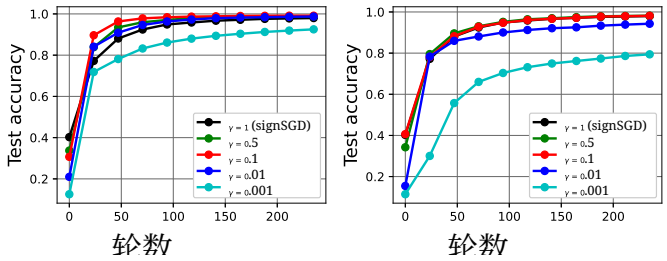
4.4 γ 和 M 对非独立同分布数据集的影响

图 3 展示了稀疏度和工作节点数量对非独立同分布数据集测试准确率的影响。随着轮数增加，图中绘制了在不同稀疏度参数 γ 和工作节点数量下，S^{3}GD-MV 算法的测试准确率。基准方法为 signSGD；当 $\gamma = 1$时，它与我们的算法等价。

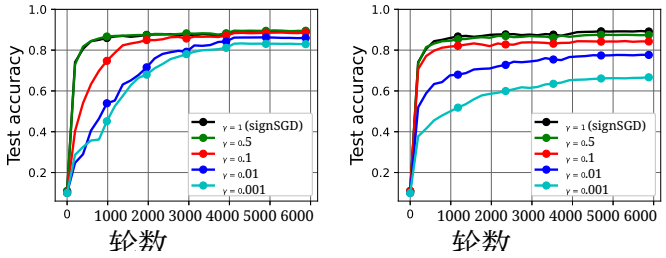
从图 3-(a) 和 3-(b) 可以明显看出，对于非独立同分布的 MNIST 数据集，传统 signSGD 在 $M = 10$和 $M = 100$ 两种情况下都无法训练卷积神经网络模型。我们的算法在 $\gamma = 0.5$ 以及 M的两种取值下也呈现出类似结果。然而，根据推论 1 的结果，为 $M = 10$ 选择 $\gamma = 0.1$，并为 $M = 100$选择 $\gamma = 0.05$，我们在尽量降低通信成本的同时实现了最高的学习率。这一结果凸显了我们的算法在处理非独立同分布数据分布时的效率和鲁棒性：根据模型大小和工作节点数量，谨慎选择 γ。

4.5 通信成本降低

对于分布式学习而言，比较各算法在测试准确率与总通信成本方面的表现非常重要。总通信成本通过将通信轮次乘以每次迭代成本计算得出。基于总通信成本对不同算法进行的测试准确率比较如图 4 所示。我们使用参数为 (M, γ) =(10, 0.001)、(M, γ) =(10, 0.001)、(M, γ) =(100, 0.001) 和 (M, γ) =(100, 0.05) 的算法，与包括原始 SGD、Top-K SGD

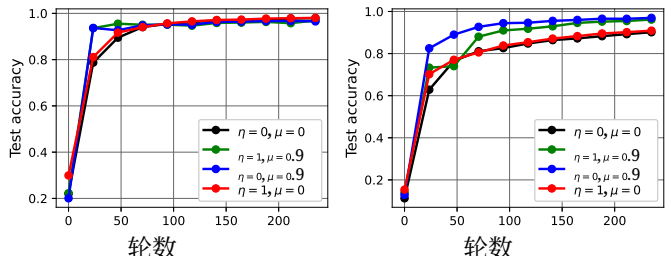


(a) TopKSign 和 MNIST 数据集 (b) RandKSign和 MNIST 数据集

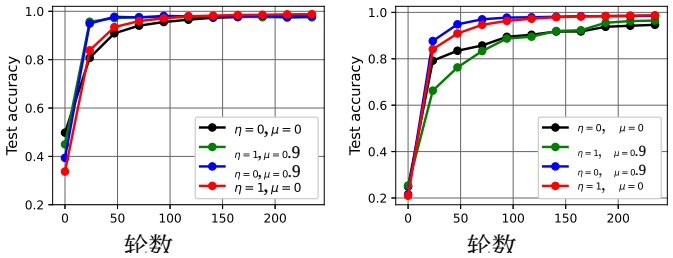


(c) TopKSign 和 CIFAR10 数据集(d) RandKSign 和 CIFAR10 数据集

图 5. $M = 10$ 与独立同分布数据分布下，不同梯度压缩方案的测试准确率随训练轮数的变化。



(a) $M = 1$ 和 $\gamma = 0.5$ (b) $M = 1$ 和 $\gamma = 0.01$



(c) $M = 10$ 和 $\gamma = 0.5$ (d) $M = 10$ 和 $\gamma = 0.01$

图 6. MNIST 数据集上不同超参数的测试准确率对比：误差权重 η 和动量 μ。

with memory，以及 signSGD 在内的基准算法进行比较。每种算法的总通信成本汇总于表 2 中。

图 4-(a) 所示结果表明，S^{3}GD-MV 在测试准确率和通信成本降低方面均表现出优异性能。当$(M, \gamma) = (10, 0.1)$时，所提出的算法取得了最高的测试准确率，同时与原始 SGD、top-K SGD 和 signSGD 相比，将总通信成本分别降低了约 100 倍、80 倍和 3 倍。通过减小 γ，我们可以进一步降低通信成本，但会略微牺牲测试准确率。例如，当$(M, \gamma) = (10, 0.001)$时，为达到 95% 的测试准确率，与原始 SGD、top-K SGD 和 signSGD 相比，我们的算法将通信成本分别降低了约 300 倍、200 倍和 10 倍。

即使在$M = 100$ 时，这一显著提升仍然能够实现，如图 4-(b)所示。例如，当$(M, \gamma) = (100, 0.05)$时，为达到 95% 的测试准确率，我们的算法所需的通信开销约比原始 SGD、top-K SGD 和 signSGD 算法分别低 200 倍、100 倍和 5 倍。这些结果清楚地证明了S^{3}GD-MV所采用的稀疏化和符号量化技术的有效性。

4.6 随机选择的影响

我们比较了两种算法的测试准确率：算法1使用TopKSign，算法2使用RandKSign，测试数据集包括独立同分布的MNIST和CIFAR10数据集。如图所示，两种算法的通信成本相同，但测试准确率表现会随 γ的取值而变化。对于使用TopKSign的算法1，γ 存在一个最优取值，能带来最佳学习表现；该取值取决于工作节点数量 M和模型大小 N。然而，对于使用RandKSign的算法2，在两个数据集上，测试准确率都会随着γ 增加而持续提高。这一结果表明，可以通过增加通信来提高测试准确率成本；这与TopKSign的情况相反，后者在 γ 取特定值时可获得最高学习表现，同时降低通信成本。

4.7 优化超参数的影响

优化深度学习模型时，研究超参数的影响也很重要，包括学习率和动量项，因为它们可能加快收敛速率并提高准确率。为此，我们将所提算法在式 (7) 中的模型更新规则修改为

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \delta^t \mathbf{v}^t, \tag{25}$$

其中，$\mathbf{v}^t$ 表示在迭代 t时加入的动量，其定义为

$$\mathbf{v}^t = \mu \mathbf{v}^{t-1} + \operatorname{sgn}\left[\sum_{m=1}^{M} \operatorname{TopKSign}\left(\mathbf{g}_m^t\right)\right]. \tag{26}$$

图 6 绘制了不同超参数 η和 μ 下的测试准确率结果。加入动量后，收敛速率更快，但最终准确率低于不使用动量时的结果。结果显然表明，选择合适的超参数可以提升学习性能。尽管超参数会影响学习性能，我们的算法在通信成本降低方面仍优于所有基准算法。生成前述图表时，使用了超参数 $\eta = 1$和 $\mu = 0$。

5 定理和引理的证明

本节给出前文定理和引理中所述主要分析结果的证明。

5.1 引理1的证明

证明。 随机梯度 $g_{m,n}^t$ 的二阶矩下界为

$$\begin{aligned}\mathbb{E}\left[\left|g_{m,n}^t\right|^2\right] &= \int_0^{\infty} \mathbb{P}\left[\left|g_{m,n}^t\right|^2 \geq x\right] dx \\ &\overset{(a)}{=} \int_0^{\left(\rho_{m,n}^t(\gamma)\right)^2} \mathbb{P}\left[\left|g_{m,n}^t\right|^2 \geq x\right] dx \\ &\quad + \int_{\left(\rho_{m,n}^t(\gamma)\right)^2}^{\infty} \mathbb{P}\left[\left|g_{m,n}^t\right|^2 \geq x\right] dx \\ &\geq \int_0^{\left(\rho_{m,n}^t(\gamma)\right)^2} \mathbb{P}\left[\left|g_{m,n}^t\right|^2 \geq x\right] dx \\ &\geq \left(\rho_{m,n}^t(\gamma)\right)^2 \mathbb{P}\left[\left|g_{m,n}^t\right| \geq \rho_{m,n}^t(\gamma)\right] \\ &= \left(\rho_{m,n}^t(\gamma)\right)^2 \mathbb{P}\left[\mathsf{TopKSign}\left(g_{m,n}^t\right) \neq 0\right] \\ &\overset{(b)}{=} \gamma \left(\rho_{m,n}^t(\gamma)\right)^2, \end{aligned} \tag{27}$$

其中，(a)成立是因为$\left|g_{m,\ n}^t\right|^2$ 是一个正随机变量，而(b)则由假设 4得出。由于可以选择顶部-K 阈值$\rho_{m,\ n}^t(\gamma)$ 使其与稀疏度参数 γ成反比，因此我们可以用 γ 的多项式函数表示$\rho_{m,\ n}^t(\gamma)$ 的一个下界：

$$\rho_{m,n}^t(\gamma) \geq \epsilon' \frac{1}{\gamma^{\ell}}, \tag{28}$$

其中，$\epsilon' \geq 0$ 是某个常数，$\ell \in \mathbb{R}^+$是正值。将(28)代入(27)，可得

$$(\epsilon')^2 \frac{1}{\gamma^{2\ell-1}} \leq \mathbb{E}\left[\left|g_{m,n}^t\right|^2\right]. \tag{29}$$

此外，根据假设3，$\mathbb{E}\left[\left|g_{m,n}^t\right|^2\right] \leq \left|\bar{g}_n^t\right|^2 + \sigma_n^2$，我们可以得到$(\epsilon')^2\ {}_{\gamma^{2\ell-1}}^{1}$的上界：

$$(\epsilon')^2 \frac{1}{\gamma^{2\ell-1}} \leq \left|\bar{g}_n^t\right|^2 + \sigma_n^2. \tag{30}$$

通过设置 $\ell = 1/2$，我们有

$$\epsilon' \leq \left|\bar{g}_n^t\right| \sqrt{1 + \frac{\sigma_n^2}{\left|\bar{g}_n^t\right|^2}}. \tag{31}$$

由于对于所有 $n \in [N]$，$\sqrt{1 + \sigma_n^2/\left|\bar{g}_n^t\right|^2} \geq 1$ 均成立，只要我们选择小于 $\left|\bar{g}_n^t\right|$的 ϵ'，该不等式始终成立。因此，如果设置 $\epsilon = \epsilon'/\left|\bar{g}_n^t\right|$ 和 $\ell = 1/2$，$\rho_{m,\ n}^t(\gamma)$ 的下界为$\rho_{m,\ n}^t(\gamma) \geq \frac{\epsilon}{\sqrt{\gamma}} \left|\bar{g}_n^t\right|$，从而完成证明。 □

5.2 引理 2 的证明

证明。 我们关注的是计算真实梯度的符号$\mathsf{sign}(\bar{g}_n^t)$ 与 TopKSign 算子$\mathsf{TopKSign}(g_{m,n}^t)$选出的随机梯度符号不同的概率。如图7所示，当随机梯度与真实梯度之间的差值$\left|g_{m,\ n}^t - \bar{g}_n^t\right|$大于真实梯度的幅值与阈值之和时，就会发生符号翻转错误。采用式（3）中 top-K 稀疏化操作的阈值$\rho_{m,n}^t(\gamma)$，我们可以将其上界估计为

$$\begin{aligned}&\mathbb{P}\left[\mathsf{TopKSign}\left(g_{m,n}^t\right) \neq \mathsf{sign}\left(\bar{g}_n^t\right)\right] \\ &\quad \leq \mathbb{P}\left[\left|g_{m,n}^t - \bar{g}_n^t\right| \geq \left|\bar{g}_n^t\right| + \rho_{m,n}^t(\gamma)\right]. \end{aligned} \tag{32}$$

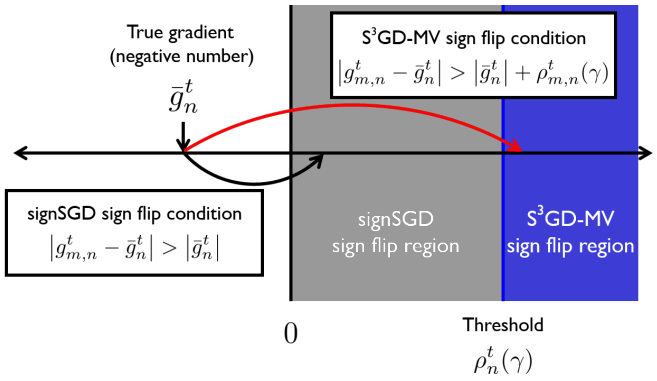


图7。SS³GD-MV 和 signSGD 符号翻转事件示意图。

然后，应用马尔可夫不等式和詹森不等式，可进一步将该概率上界估计为

$$\begin{aligned}\mathbb{P}\left[\left|g_{m,n}^t - \bar{g}_n^t\right| \geq \left|\bar{g}_n^t\right| + \rho_{m,n}^t(\gamma)\right] &\leq \frac{\mathbb{E}\left[\left|g_{m,n}^t - \bar{g}_n^t\right|\right]}{\left|\bar{g}_n^t\right| + \rho_{m,n}^t(\gamma)} \\ &\leq \frac{\sqrt{\mathbb{E}\left[\left|g_{m,n}^t - \bar{g}_n^t\right|^2\right]}}{\left|\bar{g}_n^t\right| + \rho_{m,n}^t(\gamma)} \\ &\leq \frac{\sigma_n^t}{\left|\bar{g}_n^t\right| + \rho_{m,n}^t(\gamma)}, \end{aligned} \tag{33}$$

其中，最后一个不等式由假设3推出。为了得到上界，我们需要$\rho_{m,\ n}^t(\gamma)$的下界，该下界在引理 1 中给出。将（15）代入（33），我们得到通过前K 个稀疏化得到的符号翻转错误概率的上界如下：

$$\begin{aligned}\mathbb{P}\left[\mathsf{TopKSign}\left(g_{m,n}^t\right) \neq \mathsf{sign}\left(\bar{g}_n^t\right)\right] &\leq \frac{1}{1 + \frac{\epsilon}{\sqrt{\gamma}}} \cdot \frac{\sigma_n^t}{\left|\bar{g}_n^t\right|} \\ &= \frac{1}{\sqrt{B^t}\left(1 + \frac{\epsilon}{\sqrt{\gamma}}\right)} \cdot \frac{\sigma_n}{\left|\bar{g}_n^t\right|}, \end{aligned} \tag{34}$$

其中，最后一个等式由大小为 B^t的小批量的样本方差得出，即 $\sigma_n^t = \sigma_n/\sqrt{B^t}$。 □

5.3 引理4的证明

证明。 令 $\mathcal{Z}_n^t$ 为一个工作节点集合，这些工作节点的随机梯度符号在迭代 t时与真实梯度的符号不一致，即

$$\mathcal{Z}_n^t = \left\{m \in [M] : \mathsf{TopKSign}\left(g_{m,n}^t\right) \neq \mathsf{sign}\left(\bar{g}_n^t\right)\right\} \subset \mathcal{M}_n^t \tag{35}$$

基数为 $Z_n^t \triangleq |\mathcal{Z}_n|$。注意，$Z_n^t$ 是 M_n^t 次独立伯努利试验之和，其符号翻转错误概率为 $p_{m,\ n}^t = \mathbb{P}\left[\mathsf{TopKSign}\left(g_{m,\ n}^t\right) \neq \text{符号}\ (\bar{g}_n^t)\right]$，即 $Z_n^t \sim \mathsf{B}(M_n^t,\ p_{m,\ n}^t)$。于是，通过多数投票产生的符号解码错误可计算为

$$\mathbb{P}\left[\mathsf{sgn}\left(\sum_{m \in \mathcal{M}_n^t} \mathsf{TopKSign}(g_{m,n}^t)\right) \neq \mathsf{sign}\left(\bar{g}_n^t\right)\right] = \mathbb{P}\left[Z_n^t \geq \frac{M_n^t}{2}\right]. \tag{36}$$

应用 Chernoff-Hoeffding 不等式 [39]，我们得到错误概率的界为

$$\mathbb{P}\left[Z_n^t \geq \frac{1}{2}M_n^t\right] \leq \exp\left(-M_n^t \mathsf{KL}\left(\frac{1}{2}||p_{m,n}^t\right)\right), \tag{37}$$

其中，$\mathsf{KL}\left(\frac{1}{2}||p_{m,n}^t\right)$表示参数为 1/2 和$p_{m,n}^t$的两个伯努利随机变量之间的 Kullback-Leibler距离，定义为

$$\mathsf{KL}\left(\frac{1}{2}||p_{m,n}^t\right) = \frac{1}{2}\ln\frac{1}{2p_{m,n}^t} + \frac{1}{2}\ln\frac{1}{2(1-p_{m,n}^t)}. \tag{38}$$

由于$\mathsf{KL}\left(\frac{1}{2}||p_{m,n}^t\right) = \frac{1}{2}\ln\frac{1}{4p_{m,n}^t(1-p_{m,n}^t)}$，这个上界可简化为

$$\mathbb{P}\left[Z_n^t \geq \frac{1}{2}M_n^t\right] \leq \left[4(1-p_{m,n}^t)p_{m,n}^t\right]^{\frac{M_n^t}{2}}. \tag{39}$$

证明至此完成。 □

5.4 定理 1 的证明

证明。 回顾引理 3， M_n^t 表示在迭代 t中，通过 top-K 稀疏化选取第 n个梯度分量的工作节点数量，其分布为

$$\mathbb{P}\left[M_n^t = u\right] = \binom{M}{u}\gamma^u(1-\gamma)^{M-u}. \tag{40}$$

对于 $u \in \{0,1,\ldots,M\}$。然后，由 (7)、(12) 和 (36) 可知，在以$\mathbf{x}^t$为条件时， $f^{t+1}-f^t$ 的期望上界为

$$\begin{aligned}
&\mathbb{E}\left[f^{t+1}-f^t\middle|\mathbf{x}^t\right] \\
&\leq -\delta^t\|\bar{\mathbf{g}}^t\|_1 + \frac{(\delta^t)^2}{2}\|\mathbf{L}\|_1 \\
&\quad + \delta^t\sum_{n=1}^N\left|\bar{g}_n^t\right|\mathbb{P}\left[M_n^t=0\right] - \frac{(\delta^t)^2}{2}\sum_{n=1}^N L_n\,\mathbb{P}\left[M_n^t=0\right] \\
&\quad + 2\delta^t\sum_{n=1}^N\left|\bar{g}_n^t\right|\sum_{u=1}^M\mathbb{P}\left[Z_n^t \geq \frac{M_n^t}{2}\middle|M_n^t=u\right]\mathbb{P}\left[M_n^t=u\right].
\end{aligned} \tag{41}$$

应用引理 4，给定 $M_n^t = u$ 时符号解码错误的概率为

$$\begin{aligned}
\mathbb{P}\left[Z_n^t \geq \frac{1}{2}M_n^t\middle|M_n^t=u\right] &\leq \left[4(1-p_{m,n}^t)p_{m,n}^t\right]^{\frac{u}{2}} \\
&\leq \left[4(1-\tilde{p}_{m,n}^t)\tilde{p}_{m,n}^t\right]^{\frac{u}{2}},
\end{aligned} \tag{42}$$

其中，最后一个不等式由引理 2 得出，即 $p_{m,n}^t \leq \tilde{p}_{m,n}^t = \frac{\sigma_n}{\sqrt{B^t}\left(1+\frac{\epsilon}{\sqrt{\gamma}}\right)|\bar{g}_n^t|} \leq \frac{1}{2}$为得到紧凑的表达式，我们推导出一个更宽松的解码错误概率上界：

$$\begin{aligned}
\mathbb{P}\left[Z_n^t \geq \frac{1}{2}M_n^t\middle|M_n^t=u\right] &\leq \left[4(1-\tilde{p}_{m,n}^t)\tilde{p}_{m,n}^t\right]^{\frac{u}{2}} \\
&\overset{(a)}{\leq} \left[4\tilde{p}_{m,n}^t\right]^{\frac{u}{2}} \\
&\overset{(b)}{\leq} \frac{\tilde{p}_{m,n}^t}{\sqrt{u}},
\end{aligned} \tag{43}$$

其中，(a) 在 $\tilde{p}_{m,n}^t < 1/4$ 时成立；(b) 是因为$\left[4\tilde{p}_{m,n}^t\right]^{\frac{M_n^t}{2}} \leq \frac{\tilde{p}_{m,n}^t}{\sqrt{M_n^t}}$，对于$0 < \tilde{p}_{m,n}^t < \left(2^{M_n^t}\sqrt{M_n^t}\right)^{-\frac{2}{M_n^t-2}}$其极限值为 $\lim_{M_n^t\to\infty}\left(2^{M_n^t}\sqrt{M_n^t}\right)^{-\frac{2}{M_n^t-2}} = 1/4$。根据引理 2 和 3，该符号解码概率的上界为

$$\begin{aligned}
&\sum_{u=1}^M\mathbb{P}\left[Z_n^t \geq \frac{1}{2}M_n^t\middle|M_n^t=u\right]\mathbb{P}\left[M_n^t=u\right] \\
&\qquad \leq \sum_{u=1}^M\frac{\tilde{p}_{m,n}^t}{\sqrt{u}}\binom{M}{u}\gamma^u(1-\gamma)^{M-u} \\
&\qquad = \frac{\sigma_n}{\sqrt{B^t}\left(1+\frac{\epsilon}{\sqrt{\gamma}}\right)|\bar{g}_n^t|}\beta(M,\gamma),
\end{aligned} \tag{44}$$

其中，最后一个等式由定义 $\beta(M,\ \gamma)$ 得出:

$$\beta(M,\gamma) = \sum_{i=1}^M\frac{1}{\sqrt{u}}\binom{M}{u}\gamma^u(1-\gamma)^{M-u}. \tag{45}$$

将（44）代入（41），可将$\mathbb{E}\left[f^{t+1}-f^t\middle|\mathbf{x}^t\right]$的上界改写为紧凑形式:

$$\begin{aligned}
\mathbb{E}\left[f^{t+1}-f^t\middle|\mathbf{x}^t\right] &\leq \alpha(M,\gamma)\,\delta^t\|\bar{\mathbf{g}}^t\|_1 + \alpha(M,\gamma)\frac{(\delta^t)^2}{2}\|\mathbf{L}\|_1 \\
&\quad + 2\delta^t\frac{\|\boldsymbol{\sigma}\|_1}{\sqrt{B^t}\left(1+\frac{\epsilon}{\sqrt{\gamma}}\right)}\beta(M,\gamma),
\end{aligned} \tag{46}$$

其中

$$\alpha(M,\gamma) = 1-(1-\gamma)^M. \tag{47}$$

将步长 $\delta^t = \frac{1}{\sqrt{T\|\mathbf{L}\|_1}}$ 和批量大小代入（T）后，我们得到

$$\begin{aligned}
\mathbb{E}\left[f^{t+1}-f^t\middle|\mathbf{x}^t\right] &\leq -\frac{\alpha(M,\gamma)}{\sqrt{T\|\mathbf{L}\|_1}}\|\bar{\mathbf{g}}^t\|_1 + \frac{\alpha(M,\gamma)}{2T} \\
&\quad + \frac{\beta(M,\gamma)}{T\sqrt{\|\mathbf{L}\|_1}}\cdot\frac{2}{1+\frac{\epsilon}{\sqrt{\gamma}}}\|\boldsymbol{\sigma}\|_1.
\end{aligned} \tag{48}$$

对 $t \in [T]$ 应用裂项和方法，我们得到 $f^0 - f^\star$ 的下界为

$$\begin{aligned}
f^0 - f^\star &\geq f^0 - \mathbb{E}\left[f^T\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{T-1} f^t - f^{t+1}\right] \\
&\geq \mathbb{E}\left[\sum_{t=0}^{T-1}\left\{\frac{\alpha(M,\gamma)}{\sqrt{T\|\mathbf{L}\|_1}}\|\bar{\mathbf{g}}^t\|_1 - \frac{\alpha(M,\gamma)}{2T}\right.\right. \\
&\qquad \left.\left. - \frac{\beta(M,\gamma)}{T\sqrt{\|\mathbf{L}\|_1}}\frac{2}{\left(1+\frac{\epsilon}{\sqrt{\gamma}}\right)}\|\boldsymbol{\sigma}\|_1\right\}\right] \\
&= \alpha(M,\gamma)\left\{\sqrt{\frac{T}{\|\mathbf{L}\|_1}}\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|\bar{\mathbf{g}}^t\|_1\right] - \frac{1}{2}\right. \\
&\qquad \left. - \frac{\beta(M,\gamma)}{\alpha(M,\gamma)\sqrt{\|\mathbf{L}\|_1}}\frac{2}{1+\frac{\epsilon}{\sqrt{\gamma}}}\|\boldsymbol{\sigma}\|_1\right\}.
\end{aligned} \tag{49}$$

通过重新整理式（49），我们得出

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|\bar{\mathbf{g}}^t\|_1\right] \le \frac{1}{\sqrt{T}}\left[\sqrt{\|\mathbf{L}\|_1}\left(\frac{f^0-f^\star}{\alpha(M,\gamma)}+\frac{1}{2}\right)+\frac{\beta(M,\gamma)}{\alpha(M,\gamma)}\frac{2}{1+\frac{\epsilon}{\sqrt{\gamma}}}\|\boldsymbol{\sigma}\|_1\right]. \tag{50}$$

证明至此完成。 □

5.5 推论1的证明

证明。 当 $\gamma \ll 1$ 时，式（20）中的收敛上界可表示为

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|\bar{\mathbf{g}}^t\|_1\right] \le \frac{1}{\sqrt{T}}\left(\frac{f^0-f^\star}{M\gamma}\sqrt{\|\mathbf{L}\|_1}+\frac{2\sqrt{\gamma}}{\epsilon}\|\boldsymbol{\sigma}\|_1\right) \triangleq h(\gamma). \tag{51}$$

由式（51），对 γ 求导可得

$$\frac{\partial}{\partial\gamma}h(\gamma)=\frac{1}{\sqrt{T}}\left(-\frac{f^0-f^\star}{M\gamma^2}\sqrt{\|\mathbf{L}\|_1}+\frac{1}{\epsilon\sqrt{\gamma}}\|\boldsymbol{\sigma}\|_1\right). \tag{52}$$

然后令其等于零 $\frac{\partial}{\partial\gamma}h(\gamma)\big|_{\gamma=\gamma^\star}=0$，可得到驻点 $\gamma^\star$，即

$$\gamma^\star=\left(\frac{\epsilon\left(f^0-f^\star\right)}{M}\frac{\sqrt{\|\mathbf{L}\|_1}}{\|\boldsymbol{\sigma}\|_1}\right)^{\frac{2}{3}}. \tag{53}$$

为验证 $h(\gamma)$ 的凸性，我们计算 $h(\gamma)$ 的二阶导数，得到

$$\frac{\partial^2}{\partial\gamma^2}h(\gamma)=\frac{1}{\sqrt{T}}\left(\frac{2\left(f^0-f^\star\right)}{M\gamma^3}\sqrt{\|\mathbf{L}\|_1}-\frac{1}{2\epsilon\gamma^{\frac{3}{2}}}\|\boldsymbol{\sigma}\|_1\right), \tag{54}$$

将 $\gamma^\star$ 代入式（54），可得

$$\frac{\partial^2}{\partial\gamma^2}h(\gamma)\Big|_{\gamma=\gamma^\star}=\frac{3M}{2\epsilon^2\left(f^0-f^\star\right)}\frac{\|\boldsymbol{\sigma}\|_1^2}{\sqrt{\|\mathbf{L}\|_1}}>0, \tag{55}$$

这意味着 $\gamma=\gamma^\star$ 是局部凸的。此外，我们可以很容易地验证，当 $\gamma>\gamma^\star$ 时，$\frac{\partial}{\partial\gamma}h(\gamma)>0$，并且 极限 $\lim_{\gamma\to\infty}\frac{\partial}{\partial\gamma}h(\gamma)=0$。因此，可以验证，当 $\gamma\ll 1$ 时，$\gamma^\star$ 是 $h(\gamma)$ 的最优稀疏度，证毕。 □

5.6 定理 2 的证明

证明。按照算法2中的更新规则，我们可以将 (41) 改写为

$$\begin{aligned}&\mathbb{E}\left[f^{t+1}-f^t\middle|\mathbf{x}^t\right]\\ &\le -\delta^t\|\bar{\mathbf{g}}^t\|_1+\frac{\left(\delta^t\right)^2}{2}\|\mathbf{L}\|_1\\ &\quad+\delta^t\sum_{n=1}^{N}\left|\bar{g}_n^t\right|\mathbb{P}\left[M_n^t=0\right]-\frac{\left(\delta^t\right)^2}{2}\sum_{n=1}^{N}L_n\,\mathbb{P}\left[M_n^t=0\right]\\ &\quad+2\delta^t\sum_{n=1}^{N}\left|\bar{g}_n^t\right|\sum_{u=1}^{M}\mathbb{P}\left[\tilde{Z}_n^t\ge\frac{M_n^t}{2}\middle|M_n^t=u\right]\mathbb{P}\left[M_n^t=u\right],\end{aligned} \tag{56}$$

其中，$\tilde{Z}_n^t$ 是式 (35) 中 Z_n^t 的 RandKSign(·) 版本。由于 RandK 算子会均匀地选择梯度分量，我们得到

$$\mathbb{P}\left[M_n^t=0\right]=(1-\gamma)^M. \tag{57}$$

此外，我们可以将幅值阈值$\rho_{m,\,n}^t(\gamma)$ 设为零$\rho_{m,\,n}^t(\gamma)=0$，从而忽略该阈值。由式（32）可得，单个工作节点情况下符号解码错误的概率为

$$\mathbb{P}\left[\mathsf{RandKSign}\left(g_n^t\right)\ne\mathsf{sign}\left(\bar{g}_n^t\right)\right]\le\frac{\sigma_n}{\sqrt{B^t\left|\bar{g}_n^t\right|}}. \tag{58}$$

将式（58）代入式（43），可得符号错误概率为

$$\mathbb{P}\left[\tilde{Z}_n^t\ge\frac{M_n^t}{2}\right]\le\frac{\sigma_n}{\sqrt{M_n^tB_n^t\left|\bar{g}_n^t\right|}}. \tag{59}$$

我们想强调的是，该符号翻转错误概率与 [7] 中的结果相同。代入式（57）和式（59），可将式（56）改写为

$$\begin{aligned}\mathbb{E}\left[f^{t+1}-f^t\middle|\mathbf{x}^t\right]&\le\alpha(M,\gamma)\,\delta^t\|\bar{\mathbf{g}}^t\|_1+\alpha(M,\gamma)\,\frac{\left(\delta^t\right)^2}{2}\|\mathbf{L}\|_1\\&\quad+\beta(M,\gamma)\frac{2\delta^t}{\sqrt{B^t}}\|\boldsymbol{\sigma}\|_1.\end{aligned} \tag{60}$$

使用与推导定理 1 时相同的学习率和批量大小，式（60）可改写为

$$\begin{aligned}\mathbb{E}\left[f^{t+1}-f^t\middle|\mathbf{x}^t\right]&\le-\frac{\alpha(M,\gamma)}{\sqrt{T\|\mathbf{L}\|_1}}\|\bar{\mathbf{g}}^t\|_1+\frac{\alpha(M,\gamma)}{2T}\\&\quad+\frac{2\beta(M,\gamma)}{T\sqrt{\|\mathbf{L}\|_1}}\|\boldsymbol{\sigma}\|_1.\end{aligned} \tag{61}$$

对整个迭代过程应用裂项和，最终得到

$$\begin{aligned}f^0-f^\star&\ge f^0-\mathbb{E}\left[f^T\right]\\&=\mathbb{E}\left[\sum_{t=0}^{t-1}f^t-f^{t+1}\right]\\&\ge\mathbb{E}\left[\sum_{t=0}^{T-1}\left\{\frac{\alpha(M,\gamma)}{\sqrt{T\|\mathbf{L}\|_1}}\|\bar{\mathbf{g}}^t\|_1-\frac{\alpha(M,\gamma)}{2T}\right.\right.\\&\qquad\left.\left.-\frac{2\beta(M,\gamma)}{T\sqrt{\|\mathbf{L}\|_1}}\|\boldsymbol{\sigma}\|_1\right\}\right]\\&=\alpha(M,\gamma)\left\{\sqrt{\frac{T}{\|\mathbf{L}\|_1}}\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|\bar{\mathbf{g}}^t\|_1\right]-\frac{1}{2}\right.\\&\qquad\left.-\frac{2\beta(M,\gamma)}{\alpha(M,\gamma)\sqrt{\|\mathbf{L}\|_1}}\|\boldsymbol{\sigma}\|_1\right\}.\end{aligned} \tag{62}$$

重新整理式（62），可得定理2中的表达式，至此证明完成。 □

6 结论

我们提出了一种名为S^{3}GD-MV的通信高效分布式学习算法。根据多数投票原则，S^{3}GD-MV结合稀疏化和符号量化，带来两项协同增益：首先，它显著降低了

通信成本，优于当前最先进的算法。其次，在根据模型大小 N 和工作节点数量 M适当选择稀疏化参数 K 后，该算法的收敛速率快于现有方法。我们通过收敛速率分析和仿真，从理论和实证两方面展示了这些协同增益。

一项有前景的未来研究是，探究稀疏化与符号量化在联邦学习问题中的协同增益，并展示这一更一般设定下的收敛速率。预计这将拓展S^{3}GD-MV的潜在应用范围。探究该算法对对抗攻击的鲁棒性也很有意义。

参考文献

[1] C. Park 和 N. Lee，“S^{3}GD-MV：带多数投票的稀疏符号SGD，实现通信高效的分布式学习”，提交至 *IEEE* 国际信息论研讨会（*ISIT*），2023年2月。[2] M. Zinkevich、M. Weimer、L. Li 和 A. Smola，“并行随机梯度下降”，载于神经信息处理系统进展会议论文集，第23卷，2010年12月，第2595–2603页。[3] L. Bottou，“利用随机梯度下降进行大规模机器学习”，载于国际计算统计会议论文集（*COMPSTAT*），2010年9月，第177–186页。[4] J. Chen、X. Pan、R. Monga、S. Bengio 和 R. Jozefowicz，“重新审视分布式同步随机梯度下降”，*arXiv*预印本 *arXiv:1604.00981,* 2016, [在线]。链接：https://arxiv.org/abs/1604.00981。[5] J. Yu、Z. Wang、V. Vasudevan、L. Yeung、M. Seyedhosseini 和 Y. Wu，“Coca：对比式图像描述模型是图文基础模型”，*arXiv*预印本 *arXiv:2205.01917*, 2022, [在线]。链接：https://arxiv.org/abs/2205.01917。[6] D. Alistarh、D. Grubic、J. Li、R. Tomioka 和 M. Vojnovic，“QSGD：通过梯度量化和编码实现通信高效的随机梯度下降”，载于神经信息处理系统进展会议论文集，第30卷，2017年12月，第1709–1720页。[7] J. Bernstein、Y.-X. Wang、K. Azizzadenesheli 和 A. Anandkumar，“signSGD：面向非凸问题的压缩优化”，载于国际机器学习会议论文集，2018年7月，第560–569页。[8] S. U. Stich、J.-B. Cordonnier 和 M. Jaggi，“带内存的稀疏化随机梯度下降”，载于神经信息处理系统进展会议论文集，第31卷，2018年12月，第4448–4459页。[9] D. Basu、D. Data、C. Karakus 和 S. Diggavi，“Qsparse-local-SGD：结合量化、稀疏化和本地计算的分布式随机梯度下降”，载于神经信息处理系统进展会议论文集，第32卷，2019年，第14 695–14 706页。[10] W. Wen、C. Xu、F. Yan、C. Wu、Y. Wang、Y. Chen 和 H. Li，“Tern-Grad：利用三值梯度减少分布式深度学习中的通信”，载于神经信息处理系统进展会议论文集，第30卷，2017年12月，第1–13页。[11] F. Sattler、S. Wiedemann、K.-R. M¨uller 和 W. Samek，“基于非独立同分布数据的稳健且通信高效的联邦学习”，*IEEE* 神经网络与学习系统汇刊，第31卷，第9期，第3400–3413页，2019年9月。[12] C. Renggli、S. Ashkboos、M. Aghagolzadeh、D. Alistarh 和 T. Hoefler，“SparCML：面向机器学习的高性能稀疏通信”，载于国际高性能计算、网络、存储与分析会议论文集，2019年11月，第1–15页。[13] F. Seide、H. Fu、J. Droppo、G. Li 和 D. Yu，“1位随机梯度下降及其在语音深度神经网络数据并行分布式训练中的应用”，载于国际语音通信协会年会论文集（*INTERSPEECH*），2014年9月，第1058–1062页。[14] N. Strom，“利用商用GPU云计算实现可扩展的分布式深度神经网络训练”，载于国际语音通信协会年会论文集（*INTERSPEECH*），2015年9月，第1488–1492页。[15] D. Carlson、V. Cevher 和 L. Carin，“受限玻尔兹曼机的随机谱下降”，载于国际人工智能与统计学会议论文集，第38卷，2015年5月，第111–119页。[16] S. Zheng、Z. Huang 和 J. Kwok，“带误差反馈的通信高效分布式分块动量随机梯度下降”，载于神经信息处理系统进展会议论文集，第32卷，2019年，第11 446–11 456页。

[17] S. P. Karimireddy、Q. Rebjock、S. Stich 和 M. Jaggi，“误差反馈修正了signSGD及其他梯度压缩方案”，载于《国际机器学习会议论文集》，2019年6月，第3252–3261页。[18] K. Mishchenko、E. Gorbunov、M. Tak´a˘c 和 P. Richt´arik，“利用压缩梯度差进行分布式学习”，arXiv预印本 arXiv:1901.*09269,* 2019, [在线]。链接：https://arxiv.org/abs/1901.09269。[19] E. Gorbunov、K. P. Burlachenko、Z. Li 和 P. Richt´arik，“Marina：通过压缩加速非凸分布式学习”，载于《国际机器学习会议论文集》，2021年7月，第3788–3798页。[20] N. Shlezinger、M. Chen、Y. C. Eldar、H. V. Poor 和 S. Cui，“UVeQFed：面向联邦学习的通用向量量化”，《IEEE 信号处理汇刊》，第69卷，第500–514页，2020年。[21] V. Gandikota、D. Kane、R. K. Maity 和 A. Mazumdar，“vqsgd：向量量化随机梯度下降”，载于《国际人工智能与统计学会议论文集》，2021年，第2197–2205页。[22] D. Jhunjhunwala、A. Gadhikar、G. Joshi 和 Y. C. Eldar，“面向通信高效联邦学习的模型更新自适应量化”，载于《IEEE 国际声学、语音与信号处理会议论文集（ICASSP）》，2021年6月，第3110–3114页。[23] Y. Mao、Z. Zhao、G. Yan、Y. Liu、T. Lan、L. Song 和 W. Ding，“采用自适应量化的通信高效联邦学习”，载于《国际机器学习会议论文集》，2022年7月，第22 802–22 838页。[24] A. F. Aji 和 K. Heafield，“用于分布式梯度下降的稀疏通信”，arXiv预印本 arXiv:1704.*05021,* 2017, [在线]。链接：https://arxiv.org/abs/1704.05021。[25] S. Shi、X. Chu、K. C. Cheung 和 S. See，“理解分布式深度学习中的Top-K稀疏化”，arXiv预印本 arXiv:1911.*08772,* 2019, [在线]。链接：https://arxiv.org/abs/1911.08772。[26] D. Alistarh、T. Hoefler、M. Johansson、N. Konstantinov、S. Khirirat 和 C. Renggli，“稀疏化梯度方法的收敛性”，载于《神经信息处理系统进展会议论文集》，第31卷，2018年12月，第5973–5983页。[27] J. Wangni、J. Wang、J. Liu 和 T. Zhang，“面向通信高效分布式优化的梯度稀疏化”，载于《神经信息处理系统进展会议论文集》，第31卷，2018年12月，第1299–1309页。[28] Y. Lin、S. Han、H. Mao、Y. Wang 和 W. J. Dally，“深度梯度压缩：降低分布式训练的通信带宽”，arXiv预印本 arXiv:1712.*01887,* 2017, [在线]。链接：https://arxiv.org/abs/1712.01887。[29] D. Rothchild、A. Panda、E. Ullah、N. Ivkin、I. Stoica、V. Braverman、J. Gonzalez 和 R. Arora，“FetchSGD：采用草图技术的通信高效联邦学习”，载于《国际机器学习会议论文集》，2020年7月，第8253–8265页。[30] J. Jiang、F. Fu、T. Yang 和 B. Cui，“SketchML：利用数据草图加速分布式机器学习”，载于《国际数据管理会议论文集》，2018年，第1269–1284页。[31] P. Han、S. Wang 和 K. K. Leung，“面向高效联邦学习的自适应梯度稀疏化：一种在线学习方法”，载于《IEEE 国际分布式计算系统会议（ICDCS）》，2020年11月，第300–310页。[32] Z. Zhang 和 C. Wang，“MIPD：面向分布式深度神经网络训练的自适应梯度稀疏化框架”，《IEEE 并行与分布式系统汇刊》，第33卷，第11期，第3053–3066页，2022年。[33] H. M. Mahmoud、R. Modarres 和 R. T. Smythe，“快速选择算法分析：一种顺序统计量算法”，《RAIRO-理论与信息应用》，第29卷，第4期，第255–276页，1995年。[34] A. Shanbhag、H. Pirk 和 S. Madden，“大规模并行硬件上的高效top-k查询处理”，载于《国际数据管理会议论文集》，2018年5月，第1557–1570页。[35] S. Li 和 T. Hoefler，“面向分布式深度学习的近乎最优稀疏全归约”，载于《ACM SIGPLAN 并行编程原理与实践研讨会论文集》，2022年4月，第135–149页。[36] Y. LeCun、L. Bottou、Y. Bengio 和 P. Haffner，“应用于文档识别的基于梯度的学习”，《IEEE会刊》，第86卷，第11期，第2278–2324页，1998年。[37] A. Krizhevsky、G. Hinton 等，“从微小图像中学习多层特征”，硕士论文，计算机科学系，多伦多大学，加拿大安大略省多伦多，2009年。[38] K. He、X. Zhang、S. Ren 和 J. Sun，“用于图像识别的深度残差学习”，载于《IEEE计算机学会计算机视觉与模式识别会议论文集》，2016年，第770–778页。[39] W. Hoeffding，“有界随机变量之和的概率不等式”，《美国统计学会杂志》，第58卷，30，1963 301，13第。–。