

DeepL Proに登録して、プレゼン資料を編集しましょう  
詳しくは[www.DeepL.com/pro](https://www.deepl.com/pro?cta=edit-document)をご覧ください。

Minibatch vs Local SGD for

異種分散学習

|  |  |  |
| --- | --- | --- |
| ブレイク・ウッドワース  トヨタテクノス  Institute at Chicago blake@ttic.edu | Kumar Kshitij Patel（クマール・クシティ・パテル  トヨタテクノス  Institute at Chicago kkpatel@ttic.edu  アブストラクト | ネイサン・スレブロ  トヨタテクノス  Institute at Chicago nati@ttic.edu |

ここでは、各マシンが異なるマシン固有の凸型目的に対する確率的勾配推定値にアクセスでき、目的は平均目的に対して最適化することであり、マシンは断続的にしか通信できない。我々は、(i) Minibatch SGD (加速しなくても)は、この設定におけるLocal SGDの既存のすべての分析を支配する、(ii) 加速されたMinibatch SGDは、異質性が高い場合に最適である、(iii) 非異質性領域においてMinibatch SGDを改善するLocal SGDの最初の上界を提示する、と主張する。

# はじめに

最近の機械学習モデルやデータセットは大規模化しているため、分散学習のための優れた手法を開発することが重要になっています。分散型確率的最適化/学習において特に重要な設定は、(1)学習データが単一のノードに集中しているのではなく、多数の並列デバイスに分散していること、(2)このデータが不*均一に*分散していること、つまり各マシンが*異なる分布*からデータを持っていること、(3)デバイス間の通信頻度が限られていること、などが特徴です。目標は，すべてのローカルな分布に対して同時に良好な性能を発揮する単一のコンセンサス予測器を見つけることです [3, 4]．データの不均質性は，分散学習の難易度を著しく高めます．なぜなら，各機器のローカルな目的が全く異なる可能性があるため，ある分布に対して完璧なモデルを作っても，他の分布ではひどい結果になるかもしれないからです．また，デバイス間の通信が限られているため，良好なコンセンサスを得ることがさらに困難になります．

1つの可能なアプローチは、ミニバッチ確率的勾配降下法（SGD）を使用することです。通信の間に、各マシンはローカルデータを使用して1つの大きなミニバッチ確率的勾配を計算します。次に、各マシンはローカルのミニバッチ勾配を平均化し、すべてのマシンのローカル分布からのデータで構成される1つの特別に大きなミニバッチ勾配を得て、これを1回のSGD更新に使用します。Minibatch SGD は、収束率を向上させるために加速することもできます [8, 9]。このアルゴリズムは、シンプルでどこにでもあるものであり、さまざまな環境で非常にうまく機能します。

しかし、最近、もう 1 つのアルゴリズムである Local SGD (Parallel SGD または Federated SGD とも呼ばれる) [14, 20, 27] に大きな関心が寄せられています。このアルゴリズムは、 Minibatch SGD のナイーブなアプローチを改善するものとして提案されています。Local SGD では、各マシンがそれぞれのローカルな目的に対して独立して SGD を実行し、通信するたびに各マシンのローカルな反復処理を平均化します。Local SGD は非常に魅力的なアプローチです。Minibatch SGD とは異なり、各マシンは相互に通信していなくても、ローカル目的に対するローカルモデルの性能を常に向上させているため、更新回数は通信回数から切り離されます。直感的な利点に加えて、Local SGD は多くのアプリケーションで優れた性能を発揮 しています [13, 25, 26]。

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

しかし、凸型の異種目的に対して、Local SGD が Minibatch SGD よりも実際に優れていることを示すことができるでしょうか。また、どのような設定であれば、より良い保証が得られるのでしょうか。これらの質問に答えるためには、Local SGD と Minibach SGD の性能を分析する必要があります。

最近の多くの論文では、異種データ環境における Local SGD の収束特性が分析されています [2, 11, 12, 22 など]。しかし、これから説明するように、これらのLocal SGDの保証はどれも、たとえアクセラレーションがなくても、Minibatch SGDに対して改善を示すことができず、多くの領域でははるかに悪い結果となっています。これは解析の弱点なのでしょうか？この境界を改善して、特定のレジームにおいてLocal SGDがMinibatch SGDよりも実際に優れていることを示すことはできるでしょうか。あるいは、少なくとも同程度にはなるでしょうか？

最近まで、すべてのマシンのデータ分布が同じであるような同種の分散最適化においても、状況は似通っていました。Local SGD に関する分析は数多く発表されていましたが、そのどれもが Minibatch SGD よりも改善されておらず、Minibatch SGD の性能に匹敵するものさえありませんでした。最近になって Woodworth ら [23] は、同種のケースでこの問題に決着をつけ、通信がまれな場合、Local SGD は加速された Minibatch SGD よりも証明的に優れていることを示しましたが、一方で、Local SGD は Minibatch SGD の性能保証に常に一致するわけではなく、状況によっては Minibatch SGD よりも証明的に*悪い*こともあります。

この状況は、より困難で、おそらくより興味深い異種混合環境では、どのように展開するのでしょうか。難易度の高い異種混合環境こそが、Local SGDの真価を発揮する場であり、Local SGDの分析がさらに重要になると指摘する人もいます。では、異質性は、Local SGDとMinibatch SGDの両方、そして両者の比較にどのような影響を与えるのでしょうか。Local SGDの既存の分析で、異機種混合の環境下でMinibatch SGDよりも優れているものはありますか？通信がまれにしか行われない場合でも、Local SGDはMinibatch SGDよりも優れているのでしょうか。一部の人が提案しているように、異種混合という複雑さが加わることで、より洗練された Local SGD アプローチが必要になるのでしょうか。このような困難な環境では、どのような手法が最適なのでしょうか。

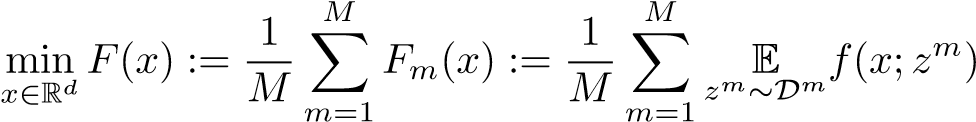
実際、Karimireddyら[11]は最近、異質性がLocal-SGDにとって特に問題となることを主張し、異質性の増加に伴って劣化することを示す下界を証明しました。セクション 4 では、この下界が、異質性のレベルが非常に大きい場合に Local SGD が Minibatch SGD よりも厳密に*悪いことを意味する*ことを説明します。しかし、Karimireddyらの下界をもってしても、単に中程度の異質性の目標に対してLocal SGDがMinibatch SGDよりも改善できるかどうかは明らかではありません。

本論文では、Local SGDが異種環境に適していないというKarimireddyらの見解を拡張する。本論文では、異機種混在データに対するLocal SGDの既存の解析は、異機種混在に近い設定でなければ大幅に改善できないことを証明します。これは、Local SGDにとって残念なことです。なぜなら、異種混合のレベルが非常に低くない限り、アクセラレーションがなくても、また通信頻度に関係なく、Local SGDが確保できる性能はMinibatch SGDよりも本当に悪いということを示しているからです。同時に、Local SGDのより詳細な分析を行い、異質性のレベルが十分に小さい場合（つまり、問題が完全に同質ではないが、少なくとも同質に近い場合）、Local SGDは実際にMinibatch SGDよりも向上することを示しました。これは、非均質な設定において*、*Local SGDがMinibatch SGDよりも改善することを示した初めての結果です。

さらに、中程度の異質性であっても（あるいはマシン間の異質性を制限しない場合）、加速されたMinibatch SGDは異質な確率的分散最適化に実際に最適であることを示します。これは、我々が示すように、Minibatch SGD とその加速版が問題の異質性に影響されないからです。今回の研究で最も重要な結論は、データが適度に不均質な場合、Accelerated Minibatch SGDが最適で*はなく*、Local SGDがMinibatch SGDよりも確実に劣る領域を特定したことであり、そのために新しい手法が必要になるかもしれません。

# セットアップ

我々は、*M台の*マシンを用いた異種分散型の確率的凸最適化/学習を考える。各マシンは独自のデータ分布*Dm*にアクセスできる。目標は、ローカル目標の平均値の近似最小化を見つけることです。

 (1)

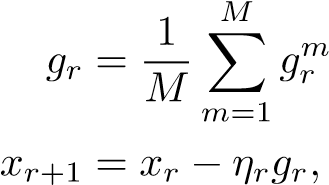
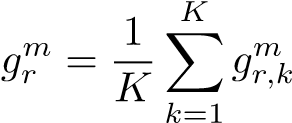
この目的は、例えば、*z* = (zfeatures*,*zlabel)であり、*f*(*x*; *z*)は、インスタンス*z*に対する*x*でパラメータ化された予測子の損失である、という教師付き学習を捉えています。

ここでは，各マシンが自分の分布からのサンプルを用いてローカルな計算を行い，他のマシンと定期的に通信してコンセンサスを得ることができる場合に注目します．このような状況は、例えば、ローカルな計算に比べて通信が高価であり、性能向上のためには通信頻度を制限することが有利な場合に生じます。

具体的には，各マシンが合計*T個の*確率的勾配を計算し，他のマシンとの通信が*R*回に制限されている分散型一次アルゴリズムを考えます．最適化プロセスを*Rラウンドに分割し，*各ラウンドでは，各マシンが*K* = *T/R個の確率的勾配を*計算・処理し，他のマシンと通信することになります．マシン*m*の各確率的勾配は、独立した*zm* ∼ *Dm*に対して∇*f*(*x*; *zm*)で与えられます。

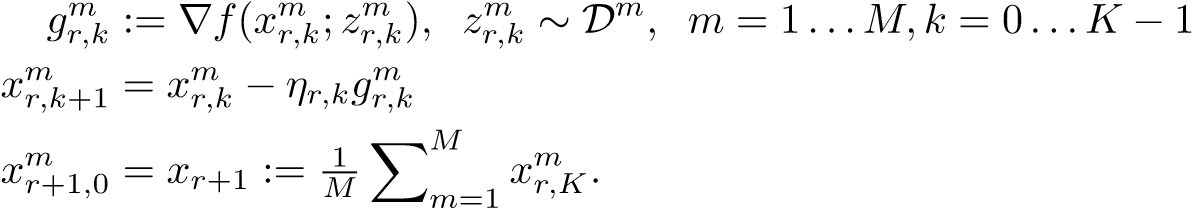
この設定のための単純なアルゴリズムが Minibatch SGD です。各ラウンドにおいて、各マシンは同じ点 *xr* で *K 個の*確率的勾配を計算し、その平均値を通知します。次に、すべてのマシンからの平均値を平均して、全体的な目的の勾配の推定値 *gr* を求めます。この推定値 *gr* は、すべての *KM* 確率的勾配に基づいており、イテレート *xr*+1 を得るために使用されます。全体として、SGDの*R*ステップを実行し、各ステップは、*KM*（非i.i.d.）サンプルのミニバッチに基づいています。要約すると、ある *x*0 で初期化すると、Minibatch SGD は以下のように動作します。

*,*

 の (2)

あるいは、同じ確率的勾配計算と *gr* の集約を実行しますが、単純な SGD 更新をより洗練された更新と慎重に調整された運動量パラメータに置き換えることもできます。本論文では、「Accelerated Minibatch SGD」という用語を使用して、SGDの2つの高速化バージョンを参照しています。これは、凸目的のAC-SA [8]と、強凸目的の多段AC-SA [9]です。疑似コードを含むアルゴリズムの詳細は、付録 C.2 に記載されています。

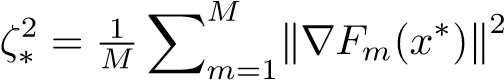
各マシンがラウンド全体を費やして同じポイントで確率的勾配を計算する Minibatch SGD とは異なり、Local SGD では、各マシンがそれぞれの確率的勾配の推定値に基づいてラウンド中にローカルの反復処理を更新できます。各ラウンドは、すべてのマシンで共通の反復処理から始まり、各マシンは自分のローカル目標に 対して SGD の *K* ステップを実行し、最終的な反復処理を伝達します。これらのイテレートは平均化され、次のラウンドの出発点となります。マシン *m* における *r* ラウンドと *K 個の*ローカルステップの後のイテレートを表す ために、すべての *m*∈[*M*] で初期化すると、Local SGD は次のように動作します。

*,*

(3)

別の視点：通信の削減。 この問題を別の視点から見ると、次のようになります。ベースラインとして、各マシンで*T個の*確率的勾配を処理するが、すべてのステップ、つまり*T*回の通信を行い、その結果、*M個の*サンプル（各マシンから1個）のミニバッチを使用して、*T*ステップのSGDを実施すると考えます。このベースラインと同じ性能を、通信回数を減らして達成することはできるでしょうか。通信回数を *T 回で*はなく *R* 回にすると、検討しているモデルに戻ります。上述したすべての方法（サイズ *KM* = *TM/R* のミニバッチを使用した MB-SGD の *R* ステップ、またはマシンごとに *T* ステップと *R の*平均化ステップを使用した Local SGD）により、通信回数を減らすことができます。*R < T* ラウンドで通信量の多いベースラインと同じ精度が得られることを確認することが出発点となりますが、問題は、精度が低下する前に（T = KR を固定した*まま*）R をどれだけ小さくできるかです。*K,M*,*Rに関して）*より良い誤差保証があれば、より小さい*R*を使っても劣化が少なく、また、漸近的に劣化しない最小の*Rは、*誤差保証から直接計算することができます[例えば、6の議論を参照]。

収束保証を証明するために、我々は問題に関するいくつかの仮定に依存しています。解析の中心となるのはパラメータ*ζで*[[1]](#footnote-2)、これはある意味、問題の不均一性のレベルを表しています。これは、コンセンサス最適化に関する文献[4, 15, 16, 19]の典型的な設定であり、実際、Minibatch SGD の分析ではこのパラメータに依存していません。実際、Minibatch SGD の分析ではこのパラメータに依存していません。しかし、比較したい Local SGD の既存の分析 [2, 11, 12] では、このような仮定*が*必要とされており、これから示すように、実際に Local SGD の収束には必要です。先行研究 [11, 12] に従って、次のように定義します。

 (4)

これは、ある意味では、*最適点における*局所勾配の変化を捉えている。*ζ*2 = 0のとき、すべての*Fm*は少なくとも1つの最小化子を共有し、*ζ*2が

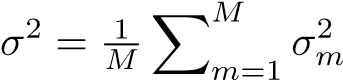
∗∗

が大きい場合、ローカル目標の間には大きな不一致があります。同種の目的（つまり*Fm*＝*F*）は*ζ*2＝0ですが、逆は真ではありません。*ζ*2 = 0であっても、*x* =6 ∗*x*∗の場合、∇*Fm*(*x*)と∇*Fn*(*x*)は異なる可能性があります。 ∗

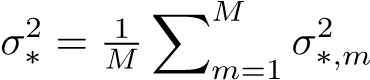
全体を通して、*f*(-; *z*)はすべての*zの*意味に対して*H*平滑であると仮定します。

 ∀*x,y,z,* (5)

各マシンにおける確率的勾配の分散は、一様に拘束されていると仮定します。

*Ezm*∼Dmk∇*f*(*x*; *zm*) - ∇*Fm*(*x*)k2 ≤ *σ*m2 ∀*x,m* となります。  (6)

または最適な*x*∗でのみ、すなわち

*Ezm*∼Dmk∇*f*(*x*∗; *zm*) - ∇*Fm*(*x*∗)k2 ≤ *σ*∗2*,m. ∀m* とします。  (7)

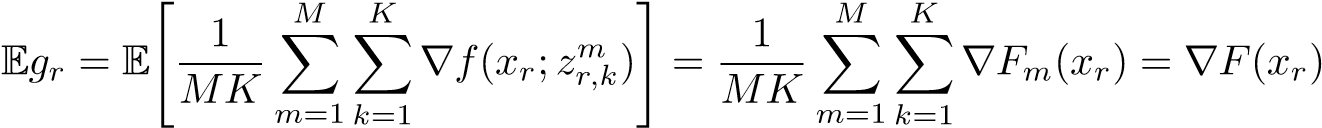
我々は2つの形式の保証を考える。強凸の局所目的に対しては、すべてのマシンの目的に対して、強凸のパラメータ*λ*に依存した保証を考える。

 (8)に加えて、上述した平滑性、異質性境界、分散の他に、初期の副最適性*F*(0) - *F*(*x*∗) ≤ ∆があります。

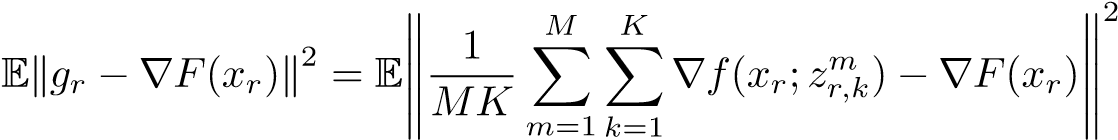
また、弱凸目的に対する保証として、各局所目的*Fm*が凸であること（必ずしも強くなくてもよい）に頼るだけの保証や、上述した平滑性、同質性の境界、分散の他に、最適な*kx*∗k≦*B*のノルムに対する境界も考慮する。

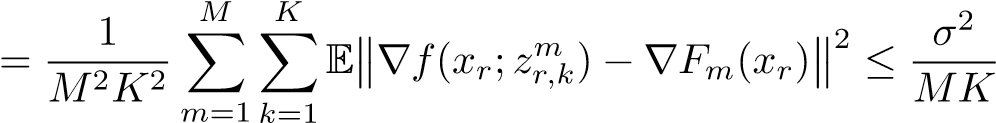
# Minibatch SGDとAccelerated Minibatch SGD

まず始めに、異質な設定におけるミニバッチSGDと加速ミニバッチSGDの最悪のケースのエラーを分析します。簡単な観察によると、目的の不均一性にもかかわらず、ミニバッチ勾配*gr*(2)は∇*F*、すなわち全体的な目的の勾配の不偏の推定値である。

 (9)

したがって、私たちは不偏的な∇*F*の推定値を使って更新しており、（加速）SGDの標準的な分析に訴えることができます。そのためには、これらの推定値の分散を計算します。

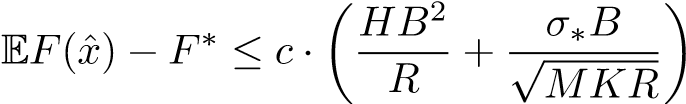
 (10)

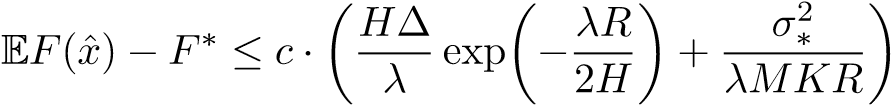
 (11)

興味深いことに、SGDの分析（詳細は付録Cを参照）では、分散は常に2だけ減少しており、異質性*ζ*∗の影響を受けていません。

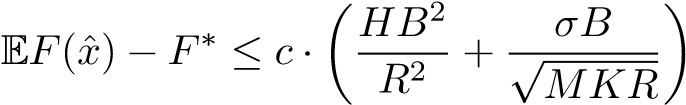
この計算を差し込むと

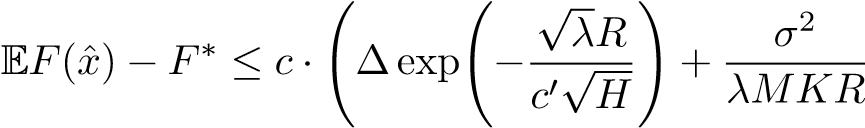
Theorem 1.*Minibatch SGD のイテレートの加重平均は、ある普遍的な定数 c において*

 *凸型の仮定の下で*

 *を、強引な仮定の下で作成します。*

*とAccelerated Minibatch SGDの*[[2]](#footnote-3)*保証*

 *凸型の仮定の下で*

!"を、*強引な仮定の下で実現しました。*

これらの保証の最大の特徴は、*ζ*2から完全に独立していることです。これらの上界は、同種の場合（*ζ*2 = 0）には厳しいことが知られているので [17, 18]∗ 、これは

∗

つまり、どちらのアルゴリズムも問題の不均一性には基本的に影響されず、同種の目的に対しても、任意に不均一な目的に対しても、同じようにパフォーマンスを発揮することができます。

# ヘテロジニアスデータ用のローカルSGD

最近では、Bayoumi ら [2]および Koloskova ら [12]が、異種データ設定の Local SGD を解析しました。凸型の場合の保証と (加速された) Minibatch SGD の分析結果を表 1 にまとめ、付録 A の表 2 には強凸型の場合の比較を示しています。また、異種分散最適化の関連手法である SCAFFOLD [[3]](#footnote-4)[11]も紹介します。

調べてみると、これまでに発表された異機種混在の Local SGD の上界（我々が知っているもの）の中では、Koloskova らの上界が最も厳しく、他を圧倒しています。しかし、この保証でさえ、Minibatch SGD境界に追加項を加えたものであるため、すべてのレジームで悪化し、Minibatch SGDよりも改善されているとは言えません。しかし、これは彼らの分析の弱点を反映したものなのか、それともLocal SGDの真の弱点を反映したものなのでしょうか。実際、Woodworthら[23]は、*同種のケースにおける*Local SGDのより厳しい上界を示しました。この上界は、Koloskovaら[12]（*ζ*∗ = 0の場合）よりも改善され、通信が頻繁に行われない場合にはMinibatch SGDよりも改善さ*れます。*

しかし、Woodworthらの縛りを異種混合の場合にも一般化できるのでしょうか？楽観的に考えれば

結局のところ、Minibatch SGDのレートは、(*ζ∗/R*)2/3の異質性への依存性に依存せず、これらの境界が改善されることを期待しています。*ζ*2に依存しているので、Local SGDの場合もそうなのではないでしょうか？

∗

残念ながら、強凸の場合には*ζ*2/(*λR*2)[[4]](#footnote-5)の依存性があることが既に知られており、Karimireddy et al.のように、より低い*ζ*∗が必要であることが示唆されています。

[11]では、下界

の bound of *ζ*∗*B/R* in convex case.しかし、おそらくKoloskovaらの分析は改善できるだろう∗。

は、この境界に合わせるよりも低次になるでしょうか？異質性がかなり大きければ。一方、Karimireddyらの下限から*HBζ*∗2*B/R/R*の項が可能であれば、*ζ∗＜HBとなり、*Koloskovaらの(ζ∗/R)*2*/3の項ができないレベルまで減速しないと考えられます。

が改善されれば、*ζ*∗ = Ω(*HB/R*)になった時点で、つまり、*ζ*∗が非常に小さくても減速することがわかります。

ここで、Koloskovaらの分析による*ζ*∗への貧弱な依存性が改善されないことを示します。その結果、十分に不均一なデータに対しては、不均一性のレベルが非常に小さくない限り、通信頻度にかかわらず、Local SGDはMinibatch SGDよりも厳密に悪いことになります。

方法／分析 ワーストケース 誤差（＝*EF*(*x*ˆ)-*F*∗ .)

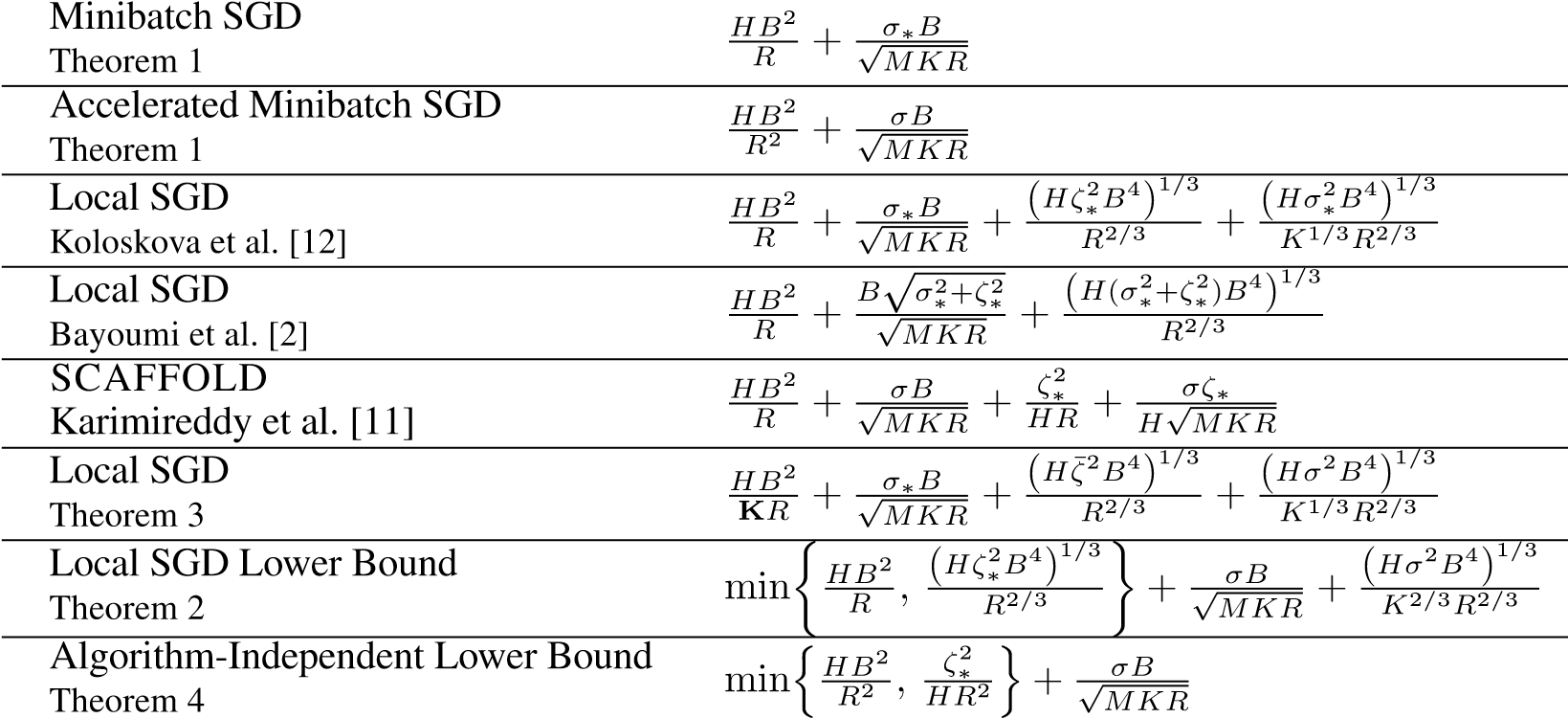
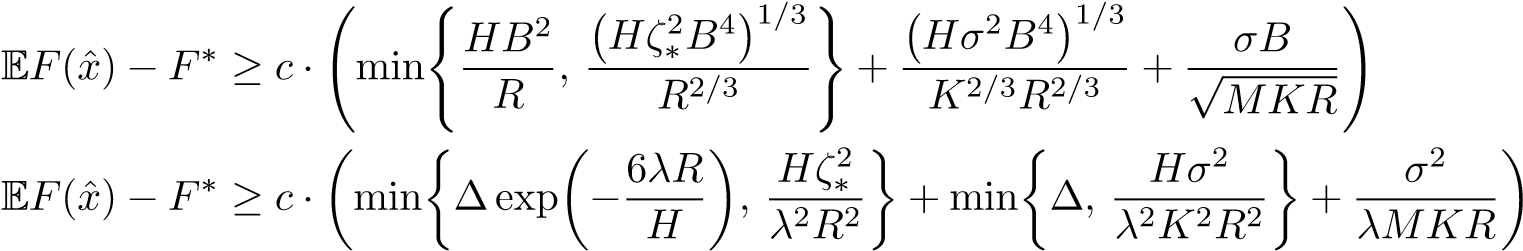


表 1: 凸型の仮定の下での保証。*ζ*¯の定義と議論については(12)を参照。

Theorem 2.*任意の M、K、R に対して、ゼロで初期化され、任意の固定ステップサイズ η を使用した Local SGD が、少なくとも以下のサブ最適性を持つような 4 次元の目的が存在します。*



*それぞれ、（H*≧16*λの場合）凸型と強凸型の仮定の下で*

これは、付録DでWoodworthら[23]の同種の場合の下界と同様のアプローチで証明されており、Karimireddyら[11]の異種目的の下界と概念的には似ています。Koloskovaら[12]も下界を証明しているが、これは特に1-強凸目的の場合であり、強凸パラメータの重要な役割を不明瞭にしている。

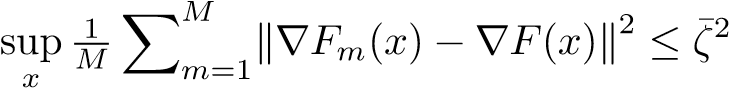
凸型の場合、この下界は、Koloskovaら[12]の上界とよく似ています。

*ζ*2の役割を強調するために*H* = *B* = *σ*2 = 1の場合に注目すると、唯一のギャップは1/3*R*2/3) vs *1*/(*K*2/3*R*2/3)の間にあり、このギャップはホモジニアスの場合にも存在します∗ (i) *1*/(*K*

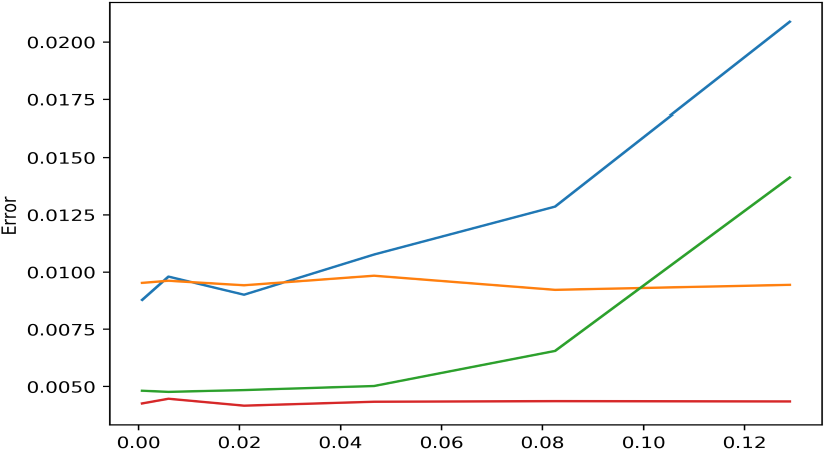
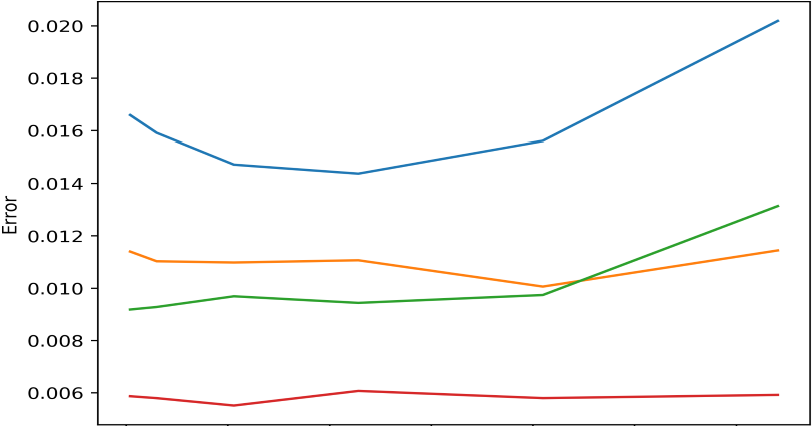
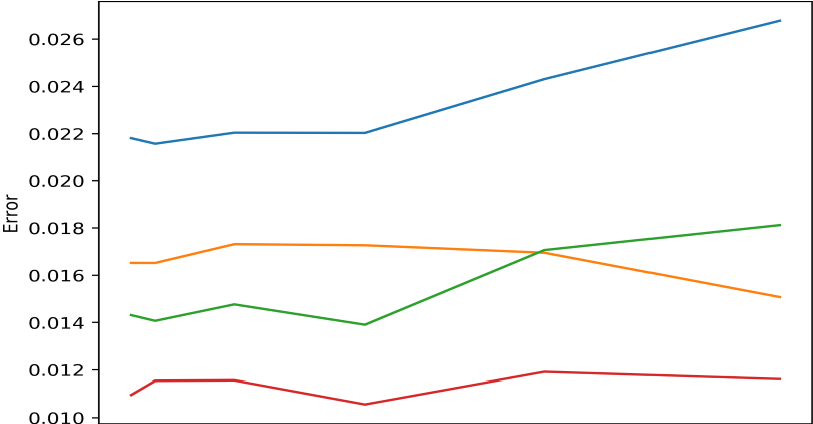
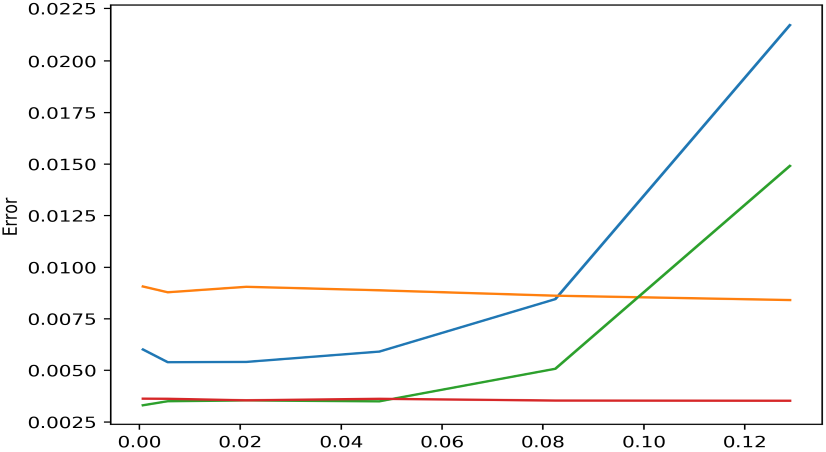
[23]と、(ii)別の項 .その結果、*ζ*∗2√≥*1/R* ⇒ (*ζ∗/R*)2/3 ≥*1/R*の場合、下界はLocal SGDが少なくとも*1/R* +の誤差を持つことを示しています。

これは非常に驚くべきことです。Local SGD は、異種混合環境において Minibatch SGD を改善するものとして提案されることが多いのですが、わずかな異種混合性でも Local SGD は大きく悪化することがわかります。さらに、各ラウンドの継続時間 *K* を増加させることは、Local SGD にとって Minibatch SGD よりも有益であると考えられがちですが、下限値を見ると、異種混合環境における Local SGD の助けにはほとんどならないことがわかります。付録 A では、強く凸の場合についても質的に同様の比較を行っています。

下限値は、Koloskova らの分析を根本的に改善することができないことを示しており、したがって、中程度の異質性であっても、より強い仮定なしに Minibatch SGD を改善することはできません。異質な環境で Minibatch SGD を改善するには、異質性測定*ζ*∗に修正を加え、ローカル目標の勾配の差を（*x*∗だけでなく）あらゆる場所で制限することにします。

 (12)

*ζ*¯2=0は*Fm*=*F*(無関係な加法定数まで)の場合のみであるため、この量は同質性を正確に捉えています。*ζ*¯2の観点からLocal SGDを分析すると、異質（*ζ*¯2が大きい）な設定から同質（*ζ*¯2 = 0）な設定へとスムーズに移行することがわかります。



*K*

=50

*K*

=25

*K*

=10

*K*

=100

*⇣*

2

⇤

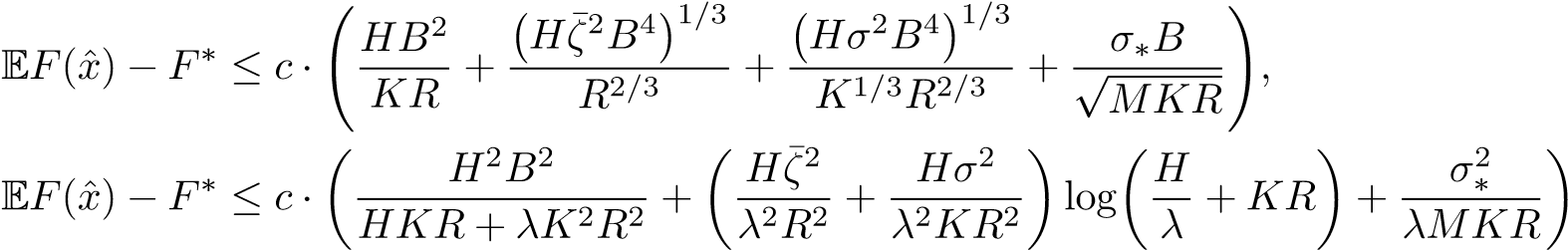
*⇣*

2

⇤

図1：MNISTの偶数桁と奇数桁の間のバイナリロジスティック回帰。*p*∈{0*,*20*,*40*,*60*,*80*,*100}に対して、*M*=25個の機械のそれぞれに、*p*%のデータをタスク*m*から、(100-*p*)%のデータをすべてのタスクの混合から割り当てました。*R*と*K*のいくつかの選択について、誤差（4回の実行の平均）と*p*の各選択から得られる*ζ*∗2の値をプロットしました。両方のアルゴリズムについて、*K*、*R*、*ζ*∗の各選択に対して最適な固定ステップサイズを使用しました。詳細は付録Fに記載しています。

Theorem 3.の*とき**、Local SGD の反復処理の平均は、凸型および強凸型の仮定の下で、それぞれ次のことを保証します。*

*.*

付録Eでは、この定理を証明します。これは、Local SGD、または異種分散最適化のためのその他の手法を分析して、任意の異種領域で Minibatch SGD よりも優れていることを示した初めての例です[[5]](#footnote-6)。*ζ*¯=0のとき、定理3はWoodworthら[23, Theorem 2]のLocal SGDの同種分析に還元されます。この分析では、その場合、*K* & Rのときに改善が見られることがすでに示されています。*定理*3は、ζ *> ¯0*のときに滑らかに劣化し、ζ¯2のときにもLocal SGDがMinibatch SGDよりも改善することを示します。*1/Rの*場合、つまり異質性が低くても正の値を示す場合にも、Local SGDがMinibatch SGDよりも改善することがわかります。この収束率を*ζ*¯2ではなく*ζ*2で確保できるかどうかはまだ不明である。

∗

実験的証拠 最後に、定理2では、*最悪のケースでζ*2が非常に小さくない限り、Local SGDはMinibatch SGDよりも劣ることが証明されていますが、「通常の」異種問題に対しては、Local SGDがその最悪のケースの誤差が示すよりも良い性能を発揮するのではないかと期待するかもしれません。しかし、MNISTを対象とした単純なバイナリ∗ロジスティック回帰実験によると、この挙動は最悪のケースを大幅に超えている可能性があります。この結果は図1に示されていますが、*ζ*\*が非常に小さく、*Kが*大きくない限り、Local SGDはMinibatch SGDよりも性能が低いことがわかります。最後に、Minibatch SGDの性能は、理論的に予測されているように、経験的に*ζ*∗の影響をほとんど受けないことも確認しています。

# 高速化されたMinibatch SGDは、高度に異質なデータに最適です。

前のセクションでは、*ζ*2 ≥ *HB/R* のとき（凸の場合）、Local SGD は Minibatch SGD よりも厳密に悪いことを示しました。しかし、もしかしたら別の方法でAccelerated Minibatch SGDよりも改善できるかもしれません。結局のところ、Accelerated Minibatch SGDの収束率は、通信ラウンドあたりのローカル計算量である*K*を増加させることで、部分的にしか∗改善できません。*K* → ∞の場合でも、Accelerated Minibatch SGDは*1/R*2またはexp(-*λR*)の準最適性しか保証されていません。このレートは改善できるのでしょうか？ここで、その答えは異質性のレベルに依存することを示します。異質性が十分に小さくなければ、どのアルゴリズムも最適である加速ミニバッチSGDよりも改善できません。下限を示すために、Carmonら[5]に倣って次のように定義します。

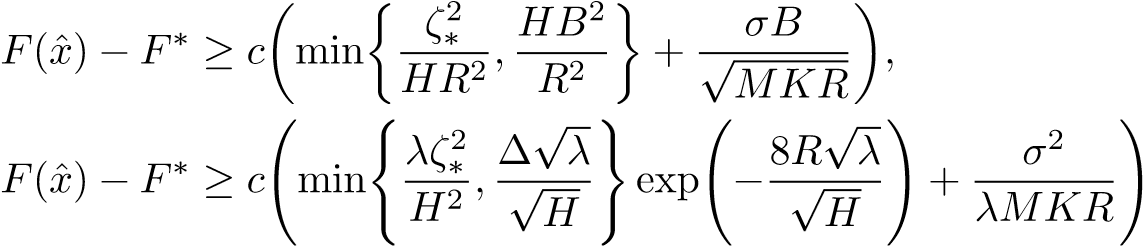
定義1 (Distributed Zero-respecting algorithm).supp(*v*)={*i*∈[*d*] : *vi* 6= 0}*とし、πm*(*t,m*0)*をtの前にマシンmとm*0が*通信した最後の時間とする。最適化アルゴリズムは、m番目のマシンに対するt番目のクエリが、以下の**条件を満たす場合、distributed zero-respectingであると言います。*

supp(*xmt* ) ⊆ [ supp(∇*f*(*xms* ; *zsm* )) ∪ [ supp(∇*f*(*xms* 0;*zsm*0*)).*

*s<tm06 =*m s*≤*πm*(t,*m*0)*

つまり、あるマシンのイテレートの座標は、そのマシンが見た勾配がゼロではない場合にのみ、ゼロではないということです。これは、Minibatch SGD、Accelerated Minibatch SGD、Local SGD、座標降下法など、多くの一次最適化アルゴリズムを網羅しています。

定理4*任意のM、K、Rに対して、任意の分散型ゼロ尊重アルゴリズムの出力が凸型と強凸型の場合にそれぞれ準最適となるような、凸型と強凸型の仮定を満たす2つの2次目的が存在する（H*≧7*λの場合）。*

*.*

これは、Woodworth and Srebro [24]と同様の手法を用いて、付録Gで証明されています。

Arjevani and Shamir [1]を参照してください。ただし、ハードインスタンスのパラメータ*ζ*∗を制御し、それが最終的な下界にどのような影響を与えるかについては注意が必要です。

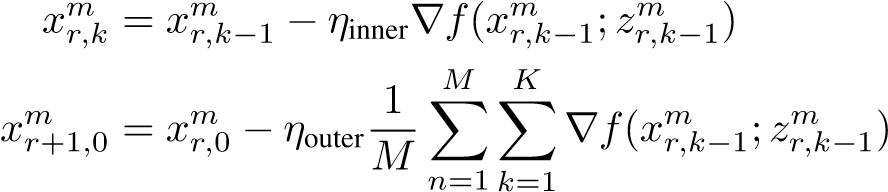
補論 1.*Accelerated Minibatch SGDは、凸型の場合にはζ*∗ ≥ *HBのときに最適であり、強凸型の場合には対数倍まで最適です。*

このように、Accelerated Minibatch SGDは、*ζ*＊が大きい場合には最適ですが、*ζ*＊が小さい場合には、下界が一致せず、改善できる可能性があります。実際、凸型の場合に注目すると、√

*HB/R* ≤ *ζ*∗ ≤ *HB という*実質的な領域で、Accelerated Minibatch SGD を改善することができる場合とできない場合があります。この領域で改善することは可能なのか、可能な場合はどのようなアルゴリズムで改善するのか。前のセクションから、Local SGDはそのようなアルゴリズムで*はない*ことがわかっています。したがって、今後の重要な課題は、*ζ*∗ が拘束されているが重要ではない場合に何ができるか、また、この領域で Accelerated Minibatch SGD を上回ることができる新しいアルゴリズムは何か、ということです。

# 内側と外側のステップサイズ

Minibatch SGD と Local SGD の関係を理解し、これらを改善する方法を考える際には、 両者の間を補間する統一的な手法を検討することが有益です。これは、あるステップサイズでローカルにSGDのステップを踏み、その後、マシンが通信する際に、結果的に異なる2つ目のステップサイズでステップを踏みます。このようなデュアルステップサイズのアプローチは、KarimireddyらによってFEDAVGとしてすでに発表され、分析されている[11]。ここでは，これらの2つのステップサイズを，それぞれ「内側」と「外側」のステップサイズと呼び，以下のように考える．

*m*∈[*M],k*∈[*K*]である。

(13)

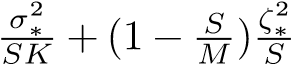
*m*∈[*M],r*∈[*R*])

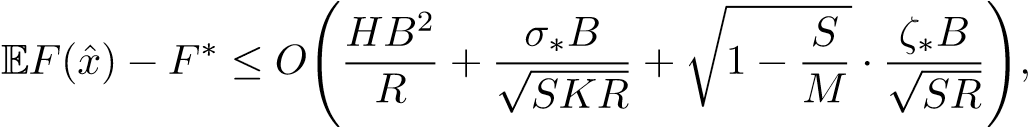
*η*inner = 0 を選択すると、これはステップサイズ *η*outer を持つ Minibatch SGD と同等になり、*η*inner = *η*outer を選択すると Local SGD が回復します。したがって、ステップサイズを最適に選択した場合、このアルゴリズムは、Minibatch SGDとLocal SGDの両方が定理1と3の最小値を達成するのと、常に同等の性能を発揮します（前者は*η*inner = 0を選択し、後者は*η*inner = *η*outerを選択します）。この分析は、付録Bで述べたように、KarimireddyらのFEDAVGの分析よりも優れています。重要な問題は、この方法が、・・・を選択することによって、両方の選択肢よりも*改善できる*かどうかである。

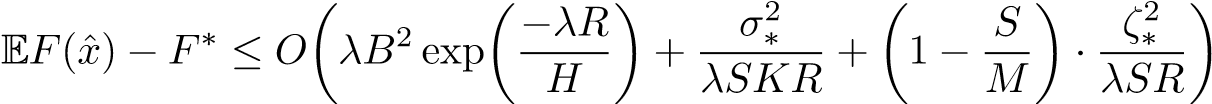
残念ながら、既存の分析ではこの問題を解決できません。

# 各ラウンドでマシンのサブセットを使用

最近Karimireddyら[11]によって定式化され研究されたこの問題設定の興味深い修正は、Federated Learning[10]の典型的なケースのように、各ラウンドで*S* ≤ *Mの*マシンのランダムなサブセットのみが使用されるケースである。同種のケースでは，すべてのマシンが同一であるため，このことに違いはない．しかし、異機種混合の場合、すべてのマシンが各ラウンドに参加するということは、各ラウンドで目的(1)の*すべての*成分の推定値を得ることができるということであり、各ラウンドで異なる*S < Mの*成分を見るだけで、すべてのMの成分の平均値を最小化したいと考えるのとは大きく異なる。

この設定でも Minibatch SGD を考えることはできますが、各ラウンドでは *SK 個の*確率的勾配に基づく勾配推定値を使用します。これは、全体的な勾配∇*F*(*xr*)の不偏な推定値であり、分散（*x*∗では分散）を伴い、以下のことが保証されます。

 (14)

*.* (15)

同様の保証は、*σ*と*ζ*¯を*σ*∗と*ζ*∗に置き換えたAccelerated Minibatch SGDにも適用できます。保証(14)および(15)は、Minibatch SGD (すなわち、*η*inner = 0)に対しても有効であり、したがって、インナー/アウター・ステップサイズ・アップデート(13)に対しても有効です。これらの保証は、Karimireddyらがインナー/アウターのステップサイズの変化について示したものよりも改善されていますが、この設定の下でいくつかのレジームにおいてMinibatch SGDよりも利点を示すSCAFFOLDも提示しています（付録Bの議論を参照）。

# ディスカッション

同種環境と異種環境の両方で Local SGD を分析する大規模な取り組みにもかか わらず、ほとんどすべての取り組みにおいて、Local SGD が Minibatch SGD よりも優れていることを示すことができませんでした。実際、我々が知る限り、どのような状況でも Minibatch SGD よりも優れていることを示すことができたのは Woodworth ら [23]だけであり、彼らの研究はより簡単な同種の設定に限定されています。このことから、同種の設定よりも大幅に困難な異種のケースでは何が起こるのか、また、（加速）Minibatch SGDの非常にナイーブなアプローチから逸脱する必要があるのではないかという疑問が生じました。本論文では、異種混合環境におけるLocal SGDとMinibatch SGDの慎重な分析と比較を行い、同種混合環境から異種混合環境に移行すると、Local SGDの性能が著しく低下することを示しました。例えば、凸型のケースでは、Local SGD は、Minibatch SGD と比べて ... でない限り、厳密に悪い結果となります。これは、LocalがMinibatch SGDよりも優れているのは、実際には同種または同種に近い環境であり、（追加の仮定をしない限り）高度に異種の環境ではないことを示しています。また、Woodworthらが示したのと同様のメリットをLocal SGDでも示し、*ζ*¯2が有界で*1/R*よりも小さい場合には、ほぼ同質の体制に拡張しています。これは、非均質な体制において、Local SGDがMinibatch SGDよりも有利であることを示した初めての分析です。反対に、異質性が高い*ζ∗ > HBの場合は、*加速ミニバッチSGDが

が最適である。しかし、この方法では、中程度の異質性の体制が残されています。

加速された）Minibatch SGDで改善することが可能かもしれないが、Local SGDは使用し*ない*場合、この体制のために他の方法を考案することはできますか？

# より広範な影響

分散型最適化は、大規模な機械学習の成功と実現可能性に大きな影響を与える重要な問題です。近年、最適化コミュニティは、一見シンプルで魅力的なアルゴリズムである Local SGD を分析し、理解することに多大な努力を払ってきました。Woodworth らの最近の研究 [23] は、こうした努力の多くが無駄であったことを示しており、その結果、おなじみの Minibatch SGD よりも改善されていない保証が得られました。しかし、彼らの分析は、同種のケースに限定されています。異種データに対する分散学習を理解することは、例えば、異なるサーバやデバイスが異なる特性を持つデータを持っている場合など、ごく自然に発生する設定であるため、重要である。我々の分析は、(i)注目すべき重要な領域と重要な問題、(ii)どのような種類の保証が進歩の助けになるか、(iii)どのような領域が新しいアルゴリズムの開発を必要とするかを明らかにすることで、この分野を明確にすることができる。このように焦点を絞ることで、開発を大幅に改善・加速することができ、分散型学習が使用される場所（それはますますあらゆる場所に広がっている）において、実用上の重要な意味を持つことになります。

他の多くの分散最適化に関する論文と同様に、ここでは異なるマシン間でコンセンサスを得ることに焦点を当てています。なぜなら、異なる分布に対して単一の解（例えば、単一の予測子や単一のモデル）にこだわることは、最適ではない可能性があるからです。また、異常値分布に対しては非常に悪い結果となる可能性があり、非典型的なユーザー、少数派グループ、または利用可能な学習データが少ないグループに不均衡に影響を与える可能性があります。そのため、分布ごとに異なる予測子を学習し、共通点を活用してパフォーマンスを向上させるという、異なる複数のアプローチで分散学習を行うことが望ましい場合があります。これは、より公平で、より効率的な方法です。本論文では、既存の文献との整合性を図るために、コンセンサスの獲得に焦点を当てていますが、ここで開発したアイデアは、多元的なアプローチの研究にも適用可能です。

# 謝辞と資金提供の開示

本研究は，NSF/BSF award 1718970，NSF-DMS 1547396，およびGoogle Faculty Research Awardによって部分的にサポートされています．BWは，Google PhD Fellowshipの支援を受けています．

# リファレンス

1. Yossi Arjevani and Ohad Shamir.分散した凸型学習と最適化の通信複雑性。In *Advances in neural information processing systems*, pages 1756-1764, 2015.
2. Ahmed Khaled Ragab Bayoumi, Konstantin Mishchenko, and Peter Richtarik.同一および異種データ上でのローカルsgdのタイトな理論。*人工知能と統計に関する国際会議*では、ページ4519-4529、2020年。
3. Dimitri P Bertsekas, John N Tsitsiklis.*並列・分散計算：数値計算法* 第23巻.Prentice hall Englewood Cliffs, NJ, 1989.
4. Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein.交互作用乗数法による分散型最適化と統計的学習.*機械学習の基礎とトレンド*R, 3(1):1-122, 2011.
5. Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford.Lower bounds for finding stationary points i. *arXiv preprint arXiv:1710.11606*, 2017.url [https://arxiv.org/abs/ 1710.11606.](https://arxiv.org/abs/1710.11606)
6. Andrew Cotter, Ohad Shamir, Nati Srebro, and Karthik Sridharan.加速された勾配法によるより良いミニバッチアルゴリズム。J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*

*24*, pages 1647-1655.カラン・アソシエイツ社、2011年。URL [http://papers.nips.cc/paper/](http://papers.nips.cc/paper/4432-better-mini-batch-algorithms-via-accelerated-gradient-methods.pdf)

[4432-[]better-[]mini-[]batch-[]algorithms-[]via-[]accelerated-[]gradient](http://papers.nips.cc/paper/4432-better-mini-batch-algorithms-via-accelerated-gradient-methods.pdf)-[]methods. [pdf.](http://papers.nips.cc/paper/4432-better-mini-batch-algorithms-via-accelerated-gradient-methods.pdf)

1. Aaron Defazio, Francis Bach, and Simon Lacoste-Julien.Saga: 非強凸の複合目的をサポートする高速な増分勾配法．In *Advances in neural information processing systems*, pages 1646-1654, 2014.
2. サイード・ガディミ、グァンホイ・ラン強く凸な確率的複合最適化のための最適な確率的近似アルゴリズム i:一般的なアルゴリズムのフレームワークです。*SIAM Journal on Optimization*, 22(4):1469-1492, 2012.
3. Saeed Ghadimi and Guanghui Lan.強く凸な確率的複合最適化のための最適な確率的近似アルゴリズム、II: 縮小手順と最適アルゴリズム。*SIAM Journal on Optimization*, 23(4):2061-2089, 2013.
4. Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings,

Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett,

Adrià Gascón、Badih Ghazi、Phillip B. Gibbons、Marco Gruteser、Zaid Harchaoui、Chaoyang

He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri

Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmiˇ Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer

Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn

Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao.Advances and open problems in federated learning, 2019.

1. Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh.SCAFFOLD: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019.
2. Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U Stich.統一された理論で、トポロジーとローカルアップデートが変化する分散型SGDの理論を提案します。
3. Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi.Don't use large mini-batches, use local sgd. *arXiv preprint arXiv:1808.07217*, 2018.
4. H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communicationefficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.
5. アンジェリア・ネディックとアスマン・オズダグラー。マルチエージェント最適化のための分散副勾配法.*IEEE Transactions on Automatic Control*, 54(1):48-61, 2009.
6. Angelia Nedic, Asuman Ozdaglar, and Pablo A Parrilo.マルチエージェントネットワークにおける制約付きコンセンサスと最適化.*IEEE Transactions on Automatic Control*, 55(4):922-938, 2010.
7. アルカディイ・セメノビッチ・ネミロフスキーとデビッド・ボリソビッチ・ユーディン。最適化における問題の複雑さと手法の効率性.1983.
8. ユリイ・ネステロフ凸最適化の入門講座：基礎編.2004.
9. S Sundhar Ram, Angelia Nedic, and Venugopal V Veeravalli.Distributed stochastic subgradient´ projection algorithms for convex optimization.*最適化理論と応用のジャーナル*, 147(3):516-545, 2010.
10. セバスチャン・U・スティッヒLocal sgd converges fast and communicates little. *arXiv preprint arXiv:1805.0967*, 2018.URL [https://arxiv.org/abs/1805.09767。](https://arxiv.org/abs/1805.09767)
11. セバスチャン・U・スティッヒ。(stochastic) gradient methodの統一的な最適解析. *arXiv preprint arXiv:1907.04232*, 2019.
12. Jianyu Wang氏とGauri Joshi氏。Cooperative sgd:A unified framework for the design and analysis of communication-efficient sgd algorithms. *arXiv preprint arXiv:1808.07576*, 2018.
13. Blake Woodworth, Kumar Kshitij Patel, Sebastian U Stich, Zhen Dai, Brian Bullins, H Brendan McMahan, Ohad Shamir, and Nathan Srebro.Is local sgd better than minibatch sgd? *arXiv preprint arXiv:2002.07839*, 2020.
14. Blake E Woodworth and Nati Srebro.複合目的を最適化するためのタイトな複雑さの境界。D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages

3639–3647.Curran Associates, Inc., 2016. URL [http://papers.nips.cc/paper/ 6058-[]tight-[]complex-[]bounds-[]for-[]optimizing-[]composition-[]](http://papers.nips.cc/paper/6058-tight-complexity-bounds-for-optimizing-composite-objectives.pdf)objectives.[pdf.](http://papers.nips.cc/paper/6058-tight-complexity-bounds-for-optimizing-composite-objectives.pdf)

1. Jian Zhang, Christopher De Sa, Ioannis Mitliagkas, and Christopher Ré.並列sgd。When does averaging help? *arXiv preprint arXiv:1606.07365*, 2016.
2. Fan Zhou and Guojing Cong.On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization.In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3219-3227.人工知能に関する国際合同会議機構、2018年7月。 doi: 10.24963/ijcai.2018/447.URL [https://doi.org/10.24963/ijcai.2018/447.](https://doi.org/10.24963/ijcai.2018/447)
3. Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola.並列化された確率的勾配降下法.並列化された確率的勾配降下法...*神経情報処理システムの進歩*, ページ 2595-2603, 2010.

1. ∗ 2

   同様に計算すると、*x*での分散は*σ∗/MKと*なります。 [↑](#footnote-ref-2)
2. この分析は、おそらく*σ*∗の観点からも述べることができますが、既存の研究からは容易には導かれません。 [↑](#footnote-ref-3)
3. Karimireddyら[11]も、異種混合のLocal SGDの変形(FEDAVGと呼ぶ)を解析していますが、付録Bで説明するように、結局はMinibatch SGDと本質的に等価です。この保証は、(3)のLocal SGDの性能ではなく、Minibatch SGDの緩やかな上界であるため、表には含めていません。Karimireddy らは、各ラウンドでマシンのランダムなサブセットのみが使用されるという、 より一般的なフレームワークを検討しています。ここでの表では、各ラウンドですべての マシンが使用されるという我々の設定に適用される分析を示しています。詳細は付録Bを参照してください。 [↑](#footnote-ref-4)
4. Karimireddyら[11, Theorem II]が述べているように、この下界は彼らのFEDAVG法に対するものであり、セクション6の(13)と同じである。しかし、彼らの下界は、ステップサイズ・パラメータの最適な選択により、*ζ*∗依存性を回避することができ、下界は成立しないので、修飾されるべきである（セクション6と付録Bを参照）。より正確には、彼らの下界は「伝統的な」Local SGD、すなわちセクション6の表記法では*η*inner = *η*outer、Karimireddyらの表記法では*ηg* = 1のときのものである。 [↑](#footnote-ref-5)
5. Karimireddyら[11]は、各反復においてマシンのランダムなサブセットのみが利用可能であり、*ζ*∗が十分に小さいという設定において、Minibatch SGDよりも改善するSCAFFOLDの保証を確立しました。ここでは、各反復においてすべてのマシンが使用される、セクション 2 の分散最適化の設定を参照します。 [↑](#footnote-ref-6)