セントラルサーバー無料連合学習 片側信頼ソーシャルネットワーク

ChaoyangHe南
カリフォルニア大学chaoyang.he@usc.edu

Conghui Tan
WeBank
tanconghui@gmail.com

ロチェスターの
ハンリンタン大学
htang14@ur.rochester.edu

ミシガン大学
ShuangQiu大学
qiush@umich.edu

Ji Liu
Kwai Inc.
ji.liu.uwisc@gmail.com

概要

連合学習は、現代の機械学習、特にデータのプライバシーに敏感なシナリオにとってますます重要になっています。既存の連合学習は、主に中央サーバーベースのネットワークトポロジを採用しています。ただし、多くのソーシャルネットワークのシナリオでは、集中型の連合学習は適用できません。たとえば、すべてのユーザーを接続する中央エージェントまたはサーバーが存在しない場合や、中央サーバーへの通信コストが手頃でない場合があります。このホワイトペーパーでは、一般的な設定について検討します。1)中央サーバーが存在しない可能性があり、2)ソーシャルネットワークが単方向または片側信頼である（つまり、ユーザーAはユーザーBを信頼しているが、ユーザーBはユーザーAを信頼していない可能性がある） 。この挑戦的で一般的なシナリオを処理するために、Online Push-Sum（OPS)メソッドと呼ばれる中央サーバーのない連合学習アルゴリズムを提案します。厳密な後悔分析も提供されます。これは、連合学習シナリオで信頼できるユーザーとのコミュニケーションからユーザーがどのように利益を得ることができるかについての興味深い結果を示しています。この作業は、一般的なソーシャルネットワークシナリオでの連合学習の基本的なアルゴリズムフレームワークと理論的保証に基づいています。

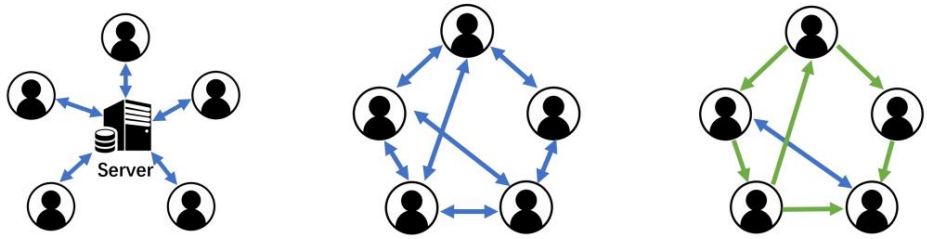
1はじめに

連合学習は、データのプライバシーを保護できるフレームワークとして広く認識されています[1、2、3]。
最先端の連合学習は、集中型ノードが子エージェントから送信された勾配を収集してグローバルモデルを更新する集中型ネットワークアーキテクチャを採用しています。その単純さにもかかわらず、集中型の方法は、特に多数のクライアントが通常関与する連合学習の場合、中央ノードでの通信と計算のボトルネックに悩まされます。さらに、ユーザーのIDのリバースエンジニアリングを防ぐために、ユーザーのプライバシーを保護するために一定量のノイズを勾配に追加する必要があります。これにより、効率と精度が部分的に犠牲になります[4]。

データのプライバシーをさらに保護し、通信のボトルネックを回避するために、分散型アーキテクチャが最近提案されました[5、6]。このアーキテクチャでは、集中型ノードが削除され、各ノードはローカルを交換することによってのみ隣接ノードと通信します（相互信頼）。モデル。
ローカルモデルは大量のデータの集約または混合であり、ローカルグラデーションはプライベートデータサンプルの1つまたはバッチのみを直接反映するため、ローカルモデルの交換は、通常、プライベートグラデーションの送信よりもデータプライバシー保護に適しています。分散型アーキテクチャの利点は、最先端の方法（集中型の方法)よりもよく認識されていますが、通常は相互に信頼できるネットワーク上でのみ実行できます（「信頼」は「情報を送信したい」を意味します） ）。つまり、2つのノード（またはユーザー)は、次の場合にのみローカルモデルを交換できます。

平等な貢献

プレプリント。審査中です。



(a)一元化　(b)分散型（相互信頼）　(c)分散型（片側信頼）

図1 :さまざまなタイプのアーキテクチャ。

相互に信頼します（たとえば、ノードAはノードBを信頼できますが、ノードBがノードAを信頼しない場合、通信できません）。ソーシャルネットワークを考えると、分散型連合学習アルゴリズムを実行するために相互に信頼できるエッジを使用することしかできません。 2つの直接的な欠点は次のとおりです。(1)すべての相互信頼エッジが接続されたネットワークを形成しない場合、連合学習は適用されません。 (2)通信ネットワークからすべての片側エッジを削除すると、通信の効率が大幅に低下する可能性があります。これらの欠点は、次の質問につながります。分散型連合学習フレームワークの下で、片側の信頼エッジをどのように効果的に利用するのでしょうか。

このホワイトペーパーでは、集中型ネットワークが利用できない（たとえば、すべてのユーザーとの接続を構築できる中央ノードが存在しない、または集中型通信コストが手頃でない)ソーシャルネットワークのシナリオを検討します。ソーシャルネットワークについては最小限の仮定をします。連合学習アルゴリズムが実行されると、データは各ユーザーノードでストリーミング方式で送信される可能性があります。ユーザー間の信頼は片側である可能性があり、ユーザーAはユーザーBを信頼しますが、ユーザーBはユーザーAを信頼しない可能性があります。

上記の設定では、次の機能を備えたオンラインプッシュサム（OPS)と呼ばれる分散型学習アルゴリズムを開発します。

- 私たちのアルゴリズムは、典型的な分散型の方法によって課せられたいくつかの制約を取り除き、任意のネットワークトポロジを許可する際により柔軟になります。各ノードは、グローバルトポロジではなく、そのアウトネイバーを知る必要があるだけです。
- アルゴリズムでは、ローカル勾配ではなくモデルのみがクライアント間で交換されます。これスキームは、クライアントのデータプライバシーを公開するリスクを減らすことができます[7]。
- 提案されたアルゴリズムの厳密な後悔分析を提供し、オンライン損失関数の2つのコンポーネント、つまり、クライアントのプライベートデータとクライアント間の内部接続をそれぞれモデル化できる敵対コンポーネントと確率的コンポーネントを区別します。

表記法このホワイトペーパーでは、次の表記法を採用しています。

数ξの場合 t (i)（i） •分布Dの対象となる確率変 t、 ΞN、TとDn、Tを使用して、それぞれ確率変数と分布のセットを示します。

(i)（i） ΞN、T =ξ
1≤t≤Tt1≤i≤n、1≤t≤T 、 Dn、T = Dt1≤i≤n、 。

表記ΞN 、T〜Dn 、 Tはξ〜Dを意味しますt (i)（i)任意の t i∈[n]およびt∈[T]に対して。

- nノードの分散型ネットワークの場合、W∈Rを使用します n×n 混同行列を提示します。ここで、 Wij≥0は、ノードiがノードjに送信する重みです（i、j∈[n]）。 N out = {j∈[n]： Wij >0}およびNin = {k∈[n]： Wki > 0}は、ノードiのin隣接およびout隣接のセットをそれぞれ示すためにも使用されます。

- ノルムk•kは、デフォルトで`2ノルムk•k2を示します。

2関連作業

連合学習の概念は[8]で最初に提案されました。これは、集中化せずにローカルで計算された勾配更新を集約することによって共有モデルを学習する新しい学習設定を提唱しています。

デバイス上の分散データ。連合学習に関する研究の初期の例には、 [9、10]、およびGoogleAIによって投稿された広範なブログ記事[11]も含まれます。統計的課題とシステム課題の両方に対処するために、 [12]と[13]は、連合学習とそれに関連する最適化アルゴリズムのためのマルチタスク学習フレームワークを提案します。これは、初期の作業SDCA [14、15、16]とCOCOA [17、18 、 19]連合学習の設定に。これらの最適化手法の中で、 [8]によって提案されたFederated Averaging（FedAvg)は、通信ラウンドに関して従来の同期ミニバッチSGDを上回り、非IIDおよび不均衡データに収束します。最近の厳密な理論的分析[20、21、22、23]は、 FedAvgが定期的なSGD（「ローカルSGD」とも呼ばれる)を平均化する特殊なケースであることを示しています。 。ただし、片側信頼ネットワーク（非対称トポロジマトリックス)には適用できません。

分散型学習は、各ワーカーが隣接するワーカーとの通信のみを必要とする典型的な並列戦略です。つまり、（パラメーターサーバー内の)通信のボトルネックが解消されます。分散型学習は、ネットワークの状態が悪い場合にワーカー数が比較的多い場合に、従来の集中型学習よりも優れていることがすでに証明されています[24]。分散型学習アルゴリズムには、主に2つのタイプがあります。固定ネットワークトポロジ[25]と、トレーニング中の時変[26、27]です。 [28、29]は、分散型SGDが集中型アルゴリズムに匹敵する収束率で収束し、通信が少なくて大規模なモデルトレーニングを実現できることを示しています。 [30]は、分散型学習の体系的な分析を提供します。

オンライン学習は何十年にもわたって研究されてきました。オンライン最適化手法の下限は、凸損失関数と強凸損失関数でそれぞれO（√T)とO（log T)であることはよく知られています[31、32]。近年、データ量の増加により、分散型オンライン学習、特に分散型手法が注目されています。これらの作品の例には[33、34、35]が含まれます。特に、 [36]は私たちの論文と同様の問題の定義と理論的結果を共有しています。

ただし、その設定では片側通信が許可されていないため、結果が制限されます。

3問題設定

このホワイトペーパーでは、n個のクライアント（別名、ノード)を使用した連合学習について検討します。各クライアントは、エッジサーバー、またはローカルのプライベートデータとローカルの機械学習モデルxiが保存されているスマートフォンなどの他の種類のコンピューティングデバイスのいずれかです。これらのn個のノードのネットワークのトポロジー構造は、頂点セット[n] = {1、2、。 。 。 、n}およびエッジセットE⊂[n]×[n]。エッジ（u、v)∈Eが存在する場合、ノードuとノードvにネットワーク接続があり、uがvに直接メッセージを送信できることを意味します。

xが反復tでのi番目のノード上のローカルモデルを表すとします。各反復で、ノードiは新しい（i)サンプルを受け取り、現在のモデルxに従ってこの新しいサンプルの予測を計算します（たとえば、オンライン推奨システムでユーにいくつかのアイテムを推奨する場合があります）。その後、その新しいサンプルに関連付けられた損失関数fi、t（•)がノードiによって受信されます。オンライン学習の一般的な目標は、後悔を最小限に抑えることです。これは、ノードの予測によって発生した損失の合計と、グローバル最適モデルxの対応する損失との差として定義されます。

R〜 、

ここでx fi、t（x)が最適なソリューションです。

過去のすべての損失の予想：

(1)

Ξn、 T〜Dn、T （X T ））

xで argminxEΞn、 T〜Dn、T PT =

上記の定式化の利点の1つは、連合学習における非IIDの問題を部分的に解決することです。多くの従来の分散型機械学習方法の基本的な前提は、すべてのノードに保存されているデータサンプルがIIDであるということです。これは、各ユーザーのデバイス上のデータがそのユーザーの好みや習慣と高度に相関しているため、連合学習には当てはまりません。ただし、私たちの定式化では、敵対的な要素を保持するためにIIDの仮定はまったく必要ありません。

確率的コンポーネントのランダムサンプルは独立している必要がありますが、異なる分布から抽出することができます。

最後に、典型的なケースとして、オンライン最適化には確率的最適化（つまり、データサンプルが固定分布から抽出される)とオフライン最適化（つまり、最適化が始まる前にデータがすでに収集されている)も含まれることに注意してください[32]。したがって、私たちの設定は幅広いアプリケーションをカバーしています。

4オンラインプッシュサムアルゴリズム

このセクションでは、混同行列の構築を定義し、提案されたアルゴリズムを紹介します。

4.1混同行列の構築

アルゴリズムの重要なパラメータの1つは、混同行列Wです。Wはネットワークトポロジに依存する行列です。これは、 Gに有向エッジ（i,j)がない場合はWij = 0を意味します。Wijの値が大きい場合、ノードiはノードjにより強い影響を与えます。ただし、 Wでは、ユーザーが既存のエッジに関連付けられた重みを指定できる柔軟性があります。つまり、2つのノード間に物理的な接続がある場合でも、ノードはチャネルの使用を決定できます。たとえば、（i,j） ∈Eであっても、ノードjが信頼できないとユーザーiが判断し、iからjまでのチャネルを除外することを選択した場合、ユーザーはWij =0を設定できます。

もちろん、Wにはまだいくつかの制約があります。Wは行の確率行列である必要があります（つまり、 Wの各エントリは非負であり、各行の合計は1です）。この仮定は、 Wが対称で二重確率行列であると通常仮定する古典的な分散分散最適化の仮定とは異なります（たとえば、 [37]） （つまり、行と列の両方の合計はすべて1です）。すべてのネットワークが二重確率行列を許可するわけではなく（[38]）、二重確率行列を放棄すると最適化にバイアスが生じる可能性があるため、このような要件は非常に制限されます[39、40]。比較として、 Wが行確率的であるという仮定は、各行に少なくとも1つの正のエントリを持つ非負行列（グラフの接続性によってすでに暗示されている)を行確率に簡単に正規化できるため、このような懸念を回避します。連合学習システムは通常、クライアントの数が多いために複雑なネットワークトポロジを伴うことを考えると、この仮定を緩和することは連合学習にとって非常に重要です。さらに、各ノードは、その重みの合計が1であることを確認するだけでよいため、グローバルネットワークトポロジを認識する必要はありません。これは、連合学習システムの実装に大きなメリットをもたらします。一方、 Wを対称にすることで、非対称ネットワークトポロジを使用し、単一側の信頼を採用する可能性が排除されますが、この方法にはそのような制限はありません。

4.2アルゴリズムの説明

提案されたオンラインプッシュサムアルゴリズムは、アルゴリズム1に示されています。アルゴリズム設計は、主にプッシュサムアルゴリズム[41]のパターンに従いますが、ここでは、オンライン設定にさらに一般化します。

アルゴリズムは主に3つのステップで構成されています。

1.ローカル更新 ：各クライアントiは、現在のローカルモデルxを適用して損失関数を取得します。 (i)これに基づいて中間ローカルモデルz t + 1/2 計算されます。

RT := E t = 1 Xn (f)、tfi(x t (xi)) (x t (私) れ(そ

アルゴリズム1オンラインプッシュサム（OPS）アルゴリズム

必要条件：学習率γ、反復回数T、および混同行列W。 (i)0

1： x0を初期化します (私) = 0から (私) = 0、 ω =すべて1 i∈[n]

2： for t = 0、1、...、T − 1 do //すべてのユーザーに対して（たとえば、 i番目のノードi∈[n]）

3： (i)

4： ローカルモデルxを適用し、損失を被る（i） fi、t（x） (私) t t; ξローカルで中間変数を計算します

5： (i) z t + 1 2 (i) (i) (i) −γ∇fi、t x t; ξ= zt t

6： Wizを送信する (私) t + 1 2 、 Weω t (私) すべてのj∈にNアウト

7： アップデート

と (私) t + 1 = X k∈Nini _ Wkiz (に) t + 1 2

ω (私) t + 1 = X k∈Nin Wkiω (k) t

x (私) t + 1 = (i) z t + 1 2 (i) ω t + 1

8： (i)の終了

xをノードiに戻す

9：

2.プッシュ：加重変数Wijz (私) t + 1 2 すべての隣接するjについてjに送信されます。

3.合計：受け取ったすべてのWjiz (j) 1 t + 2 を合計して正規化し、新しいモデルx t+1を取得します。 (i)

補助変数zは、説明を明確にするためにアルゴリズムで使用されますが、実際 (私) 1 t + 2 および z t + 1 (i) アルゴリズムで使用されます。実際、彼らはの（i)実装では簡単に削除できることに注意してください。さらに、別の変数ω （i）（i） zωt +1。 t + 1 t + 1 の正規化係数であるも導入されます

Wはこの設定では二重確率行列ではなく、 iが受け取る総重みが1に等しくない可能性があるため、プッシュサムアルゴリズムで重要な役割を果たします。正規化係数ωの導入 （i)はアルゴリズムに役立ちます。 Wが二重に（i)確率的ではないことによってもたらされる問題を回避します。さらに、 Wが二重確率になると、 ω≡1 （i)とxが[36]によって提案されていることを簡単に確認できます。

t (i) 任意のiとtについて、アルゴリズム1は分散オンライン勾配法≡ztに還元されます。 t

アルゴリズムでは、勾配fi、t（x t ; ξt） [4]でエンコードされたローカルデータは、ローカルモデル (私) の更新にのみ使用されます。隣接ノードが交換するのは、ローカルモデルのみに限定されます。

4.3後悔分析

このサブセクションでは、OPSアルゴリズムの後悔限界分析を提供します。スペースに限りがあるため、詳細証明は補足資料に委ねられています。便宜上、最初に

Fi、t（x） := E ξi、t Di、t fi、t（x; ξi、t）。

分析を実行するには、次の仮定が必要です。仮定1.このホワイトペーパーでは、次の仮定を行います。（1）トポロジグラフGは強く関連しています。 Wは確率的な行です。 （2）任意のi∈ [n]およびt∈ [T]の場合、損失関数fi、t（x; ξi、t）はxで凸です。 （3）問題領域は、任意の2つのベクトルxとyに対して、常にkx − yk≤Rになるように制限されます。 （4）期待される勾配∇Fi、t（•）のノルムは有界です。つまり、任意のi、 t、およびxに対して≤G2です。 （5） k∇Fi、t（x）kもσによって制限されるような定数G>0が存在する勾配分散

2 2 2 、すなわち、

Ek∇fi、t（x; ξi、t） − ∇Fi、t（x） kξi、t Di、t 22≤σ _ 。

ここで、定数Gは、敵対的なコンポーネントの上限を提供します。一方、 σは確率的要素によってもたらされる確率的要素の大きさを測定します。 σ= 0の場合、問題の設定は単純に通常の分散型オンライン学習に戻ります。任意の2つのノード間で情報を交換できるようにするには、強力な接続性の前提が必要です。

凸性とドメイン境界の仮定に関しては、 [31]などのオンライン学習の文献では非常に一般的です。

これらの仮定を備えたので、収束結果を提示する準備が整いました。定理2。

$$\sqrt{n}R\gamma = \sigma\sqrt{1+nC2+} \quad G\sqrt{nC1T}, \tag{2}$$

OPSの後悔は、次の要因によって制限されます。

$$RT \leq OnGR\sqrt{T} + \sigma R1 + pnC2\sqrt{nT}, \tag{3}$$

ここで、 C1とC2は、付録で定義されている2つの定数です。

n =1およびσ= 0の場合、問題の設定が通常のオンライン最適化に還元される場合、暗黙の悔恨限界O（GR√T)はオンライン最適化の下限と正確に一致することに注意してください[31]。さらに、私たちの結果は、完全に接続されたネットワークのq=0である集中型オンライン学習の収束率とも一致します。したがって、OPSアルゴリズムはTに最適に依存していると結論付けることができます。

この境界はノード数nに線形依存しますが、理解しやすいです。まず、後悔をすべてのノードの損失の合計と定義しました。 nを大きくすると、後悔が自然に大きくなります。第二に、私たちの連合学習の設定は、IIDの仮定がここでは成り立たないという点で、典型的な分散学習とは異なります。各ノードには、まったく異なる分布から抽出される可能性のある個別のローカルデータが含まれています。したがって、ノードを追加しても、既存のクライアントの後悔を減らすのに役立ちません。

さらに、各ワーカーのモデルxの差は、次の定理を使用して制限できることも証明します。

定理3.γを（ 2)として設定すると、各ワーカーのモデルxの違いにより、後悔よりも収束率が速くなります。

$$\frac{1}{T}\sum_{t=0}^{T} \sum_{i=1}^{n} \left(x_t^{(i)} - z_t\right)^2 \leq O\left(\frac{nGR+nR\sigma}{T}\right).$$

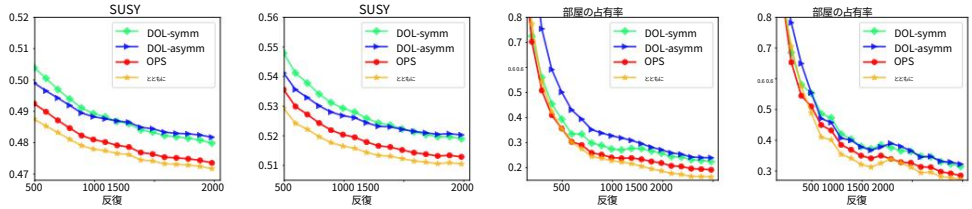
したがって、すべてのクライアントのデバイスのモデルは、最終的にレートO（1 / T)で同じモデルに収束します。

4.4プライバシー保護

この作業の主な焦点は、非対称ソーシャルネットワークを使用した実際のシナリオに対処するためのソリューションを提供することですが、提案されたアルゴリズムには、プライバシー保護に関していくつかの利点もあります。まず、前述したように、OPSは分散型で実行され、勾配やトレーニングサンプルの代わりにモデルを交換します。これは、プライバシー漏洩のリスクを軽減するのに効果的であることがすでに証明されています[42]。次に、OPSは分散型で非対称的な方法で実行されます。これらの特性は、 [43]などの多くの攻撃方法に困難をもたらします。他のクライアントのデータを推測するために、攻撃者は攻撃が注入された後の他のノードの反応を観察する必要があります。これは、接続が片側の場合は不可能です。攻撃はネットワーク全体に広がり、最終的に攻撃者に戻りますが、攻撃者はグローバルを認識していないため、近隣から受信した情報がすでに攻撃の影響を受けているかどうかを区別するのは困難です。トポロジー。

5つの実験

提案されたオンラインプッシュサム（OPS)法のパフォーマンスを、分散型オンライン勾配法（DOL)および集中型オンライン勾配法（COL)のパフォーマンスと比較し、さまざまなネットワークサイズとネットワークトポロジ密度でのOPSの有効性を評価します。設定。



（a)確率的= 100%（b)確率的= 50%（c)確率的= 100%（n = 128、#Neighb。= 32）（n = 128、#Neighb。= 32）（n = 20、#Neighb 。=10）（d)確率= 50%（n = 20、#Neighb。= 10）

図2 :OPS、DOL（分散型オンライン学習)およびCOL（集中型)の比較
オンライン学習）

5.1実装と設定

2乗の`2ノルム正則化を使用したオンラインロジスティック回帰を検討します。fi、 t （x; ξi、t） = log 1 + exp −yi、tA> i、tx +ここで、正則化係数λ> 0は固定されます。ネットワークトポロジとデータセットの詳細は、セクション5.3で紹介されます。関数fi、tの確率的バージョンは、ランダムな単一サンプル（Ai、t、 yi、t)にエンコードされます。平均PTを測定することで学習パフォーマンスを評価します

損失 $\frac{1}{n}\sum_{i=1}^{n} f_{i,t}(x_{i,t};\xi_{i,t})$ 。アルゴリズムの実装はPython 3.7.0およびTorch 1.0.1.post2に基づいています。実験の学習率は、Python 3.7.0およびTorch 1.0.1.post2に基づいています。実験の学習率は、最適な収束を保証するように調整されています。実験は、Python 3.7.0およびscikit-learn0.20.3に基づいています。ソースコードは、連合学習用のオープンソースの研究ライブラリであるFedML [44]を使用して開発されています。詳細については、 FedMLのソースコードを参照してください。

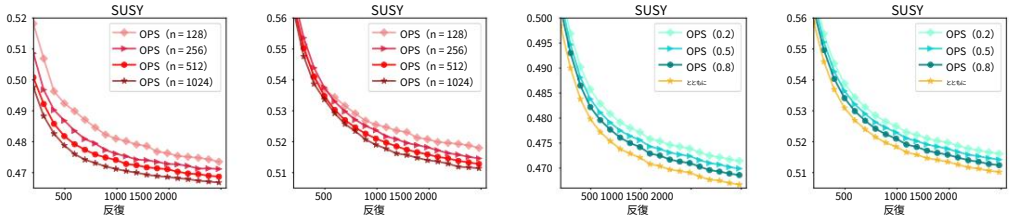
データセット。実験は、SUSY[2]とRoom-Occupancy[3]の2つの実際の公開データセットで実行されました。SUSYとRoom-Occupancyはどちらも大規模な二項分類データセットであり、それぞれ5,000,000サンプルと20,566サンプルが含まれています。各データセットは、確率的データと敵対的データの2つのサブセットに分割されます。確率的データは、サンプルの一部（たとえば、データセット全体の50%)をノードにランダムかつ均一に割り当てることによって生成されます。敵対的なデータは、残りのデータセットを実行してn個のクラスターを生成し、すべてのクラスターをノードに割り当てることによって生成されます。以前に分析したように、分散された確率的データのみがノード内通信によってモデルのパフォーマンスを向上させることができます。各ノードについて、この事前に取得されたデータは、オンライン学習をシミュレートするためにストリーミングデータに変換されます。

5.2DOLおよびCOLとの比較

OPSをDOLおよびCOLと比較するために、 SUSYおよびRoom-Occupancyに対して、それぞれ128ノードおよび20ノードのネットワークサイズが選択されています。 COLの場合、その混同行列Wは完全に接続されています（二重確率行列）。DOLとOPSの場合、公平な比較を維持するために、同じネットワークトポロジと同じ行の確率行列（非対称混同行列)で実行されます。このような非対称の混乱は、各ノードのネイバーの数を固定の上限よりも小さいランダムな値として設定し、ネットワーク全体の強力な接続を保証することによって構築されます（SUSYデータセットの場合、この上限のネイバー数は32に設定されます。一方、Room-Occupancyデータセットには10が設定されています）。 DOLは通常、ネットワークが対称で二重確率の混同行列である必要があるため、比較のために2つの設定でDOLを実行します。最初の設定では、対称性と二重確率行列の仮定を満たすために、すべての一方向接続が混同行列で削除され、行の確率的混同行列が二重確率行列に縮退します。この設定は、図2でDOL-Symmとラベル付けされています。別の設定では、DOLは、送信ウェイトが受信ウェイトと等しいかどうかを考慮せずに、各ノードが受信モデルを単純に集約する非対称ネットワークで実行するように強制されます。 DOL-Asymmは、図2でこの設定にラベルを付けるために使用されます。

図2に示すように、2つのデータセットの両方で、OPSは行の確率的混同行列でDOL-Symmよりも優れています。これは、単方向通信を組み込むことでモデルのパフォーマンスを向上できることを示しています。言い換えれば、OPSは片側の信頼でより良いパフォーマンスを獲得します

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#SUSY
[3] https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+



（a)確率= 100%　（b)確率= 50%　（c)確率= 100%　（d)確率的= 50%

図3：さまざまなネットワークサイズと密度の評価

連合学習の設定の下でのネットワーク。 OPSはDOL-Asymmよりもうまく機能します。 DOL-Asymmは追加の一方向接続を利用しますが、場合によっては、そのパフォーマンスはDOL-Symmよりもさらに劣ります（たとえば、図2a）。この現象は、その単純な集約パターンに起因する可能性が最も高く、二重確率行列の仮定を削除すると、DOL-Asymmのパフォーマンスが低下します。これらの2つの観察結果は、行確率的混同行列におけるOPSの有効性を確認します。これは、理論的分析と一致しています。

図2cと図2dを比較すると、確率的成分の比率を増やすと、平均損失（後悔)が小さくなることもわかります。 OPSは、情報交換がCOLよりもはるかに少ない、まばらに接続されたネットワークで機能するため、OPSがCOLよりもわずかに悪いパフォーマンスを達成することは合理的です。すべての実験でベースラインとしてCOLを使用します。

実験では、実際の実行時間ではなく、反復回数のみが考慮されます。実際の実行時間を表示することは冗長です。中央ノードでのネットワークの輻輳が原因で、集中型の方法では反復ごとにより多くの時間が必要になるため、OPSは通常、実行時間の点でCOLよりも優れています。

5.3さまざまなネットワークサイズでの評価

図3aおよび3bは、さまざまなネットワークサイズでのOPSの評価をまとめたものです（SUSYデータセットでは、128、256、512、1024が設定されています）。上限のネイバー番号は、その影響を分離するために、異なるネットワークサイズ間で同じ値に調整されます。ご覧のとおり、すべてのデータセットで、さまざまなネットワークサイズでの平均損失（後悔)曲線は小規模に近いものです。これらの観察結果は、OPSがネットワークサイズに対して堅牢であることを示しています。さらに、ネットワークサイズが大きいほど平均損失（後悔)は小さくなります（つまり、 n = 1024ネットワークサイズの曲線は他のネットワークサイズよりも小さくなります）。これは、より多くのノードによって提供される確率的サンプルが自然に収束を加速できることも示しています。スペースの制限により、他のデータセットの結果は付録に延期されます。

5.4ネットワーク密度の評価

また、さまざまなネットワーク密度でのOPSのパフォーマンスを評価します。 SUSYデータセットのネットワークサイズを512に固定します。ネットワーク密度は、ネットワークのサイズに対するノードごとの上限ランダムネイバー数の比率として定義されます（たとえば、SUSYで比率が0.5の場合、各ノードの上限ネイバー番号として256が設定されることを意味します）。 。図3cおよび3dから、ネットワーク密度が増加するにつれて、平均損失（後悔)が減少したことがわかります。この観察結果は、提案されたOPSアルゴリズムがさまざまなネットワーク密度でうまく機能し、より密度の高い行確率行列からより多くの利益を得ることができることも証明しています。この利点は直感的にも理解できます。連合学習ネットワークでは、より多くのユーザーと通信すると、ユーザーのモデルのパフォーマンスが向上します。部屋の占有率の結果も付録に延期されます。

5.5ローカルオンライン勾配降下法との比較

通信の必要性を正当化するために、OPSをローカルオンライン勾配降下法（ローカルOGD)と比較します。この場合、すべてのノードが他のノードと通信せずにローカルモデルをトレーニングします。結果は付録に記載されています。

6結論

片側の信頼を備えた分散型連合学習は、幅広い問題を解決するための有望なフレームワークです。この論文では、この設定のためにオンラインプッシュサムアルゴリズムを開発しました。これは、複雑なネットワークトポロジを処理でき、最適な収束率を備えていることが証明されています。後悔に基づくオンライン問題の定式化も、そのアプリケーションを拡張します。提案されたOPSアルゴリズムをさまざまな実験でテストしましたが、その効率は経験的に正当化されています。

より広い影響

この論文で提案された連合学習フレームワークは、情報を共有する意欲が一方的なアプリケーションで機械学習モデルを協調的にトレーニングするための効果的なツールを提供します。
このようなアプリケーションは、実際の生活にはたくさんあります。さらに、前述したように、このようなフレームワークには、ユーザーのプライバシーをより適切に保護する可能性がありますが、プライバシー保護は、このような情報化時代における最も重要な問題の1つであると考えています。ただし、場合によっては、片側のコミュニケーションによって、人々の間の情報の非対称性が高まる可能性があります。一部のユーザーは常に他のユーザーよりも早く情報を知るようになりますが、これは一種の人間の不平等と見なされる可能性があります。

参考文献

[1] Jakub Konecn ˇ y、H Brendan McMahan、Felix X Yu、PeterRichtárik、Ananda Theertha Suresh、およびDaveBacon。連合学習：コミュニケーション効率を改善するための戦略。 arXiv preprint arXiv：1610.05492、2016。

[2]バージニア•スミス、チャオカイ•チェン、マジアル•サンジャビ、アミート•S•タルウォーカー。フェデレーションマルチタスク学習。ニューラル情報処理システムの進歩、4424〜4434ページ、2017年。

[3] Qiang Yang、Yang Liu、Tianjian Chen、およびYongxinTong。連合機械学習：概念とアプリケーション。インテリジェントシステムとテクノロジーに関するACMトランザクション（TIST）、10（2）：12、2019 。

[4] RezaShokriとVitalyShmatikov。プライバシーを保護するディープラーニング。コンピュータと通信のセキュリティに関する第22回ACMSIGSAC会議の議事録、 1310〜1321ページ。
ACM、2015年。

[5] Paul Vanhaesebrouck、AurélienBellet、およびMarcTommasi。ネットワークを介したパーソナライズされたモデルの分散型共学習。人工知能と統計に関する国際会議（AISTATS）、2017年。

[6] AurélienBellet、Rachid Guerraoui、Mahsa Taziki、およびMarcTommasi。パーソナライズされたプライベートピアツーピア機械学習。人工知能と統計に関する国際会議、473〜481ページ、2018年。

[7]青野義典、林拓也、王李華、森内志保ほか。相加的準同型暗号化によるプライバシー保護ディープラーニング。 IEEE Transactions on Information Forensics and Security、13（5）：1333–1345、2017年。

[8] H.ブレンダン•マクマハン、アイダー•ムーア、ダニエル•ラマージュ、セス•ハンプソン、ブレイス•アグエラ•イ•アルカス。分散型データからのディープネットワークの通信効率の高い学習。 arXiv：1602.05629 [cs]、2016年2月。arXiv：1602.05629。

[9]ヤクブ•コネクニー、ブレンダン•マクマハン、ダニエル•ラマージュ。フェデレーション最適化：データセンターを超えた分散最適化。 arXiv：1511.03575 [cs、math]、2015年11月。arXiv： 1511.03575。

[10] JakubKonecný、H。Brendan McMahan、Felix X. Yu、PeterRichtárik、Ananda Theertha Suresh、およびDaveBacon。連合学習：コミュニケーション効率を改善するための戦略。 arXiv：1610.05492 [cs]、2016年10月。arXiv：1610.05492。

[11]ブレンダン•マクマハンとダニエル•ラマージュ。 Google AIブログ：連合学習：コラボレーション
一元化されたトレーニングデータを使用しない機械学習、2017年4月。

[12]バージニア•スミス、チャオカイ•チェン、マジアル•サンジャビ、アミート•S•タルヴァル。フェデレーションマルチタスク学習。ニューラル情報処理システムの進歩、4424〜4434ページ、2017年。

[13]セバスチャン•カルダス、バージニア•スミス、アミート•タルヴァル。フェデレーションカーネル化マルチタスク学習。システムと機械学習に関する会議、2018年3ページ。

[14] ShaiShalev-ShwartzとTongZhang。正則化された損失最小化のための確率的二重座標上昇法。 Journal of Machine Learning Research、14（2月）：567–599、2013年。

[15]天宝ヤン。通信のための取引計算：分散確率的二重座標上昇。ニューラル情報処理システムの進歩、629〜637ページ、2013年。

[16] Tianbao Yang、Shenghuo Zhu、Rong Jin、およびYuanqingLin。分散確率の分析二重座標上昇。 arXiv preprint arXiv：1312.1031、2013。

[17]マーティン•ジャギ、バージニア•スミス、マーティン•タカック、ジョナサン•ターホルスト、サンジェイ•クリシュナン、トーマス•ホフマン、マイケル•I•ジョーダン。通信効率の高い分散デュアル座標上昇。ニューラル情報処理システムの進歩、3068〜3076ページ、2014年。

[18] Chenxin Ma、Virginia Smith、Martin Jaggi、Michael I. Jordan、PeterRichtárik、 MartinTakác。分散プライマル-デュアル最適化における追加と平均化。 arXiv：1502.03508 [cs]、 2015年2月。arXiv：1502.03508。

[19]バージニア•スミス、シモーネ•フォルテ、チェンシン•マ、マーティン•タカック、マイケル•I•ジョーダン、マーティン•ジャギ。 CoCoA：通信効率の高い分散最適化のための一般的なフレームワーク。 arXiv：1611.02189 [cs]、2016年11月。arXiv：1611.02189。

[20]セバスチャンU.スティッチ。ローカルSGDは高速に収束し、ほとんど通信しません。 2018年9月。

[21] JianyuWangとGauriJoshi。 Cooperative SGD：通信効率の高いSGDアルゴリズムの設計と分析のための統合フレームワーク。 2018年8月。

[22] Hao Yu、Sen Yang、およびShenghuoZhu。より高速な収束とより少ない通信で並列に再起動されたSGD：モデル平均化が深層学習に機能する理由を解明します。 arXiv：1807.06629 [cs、math]、2018年7月。arXiv：1807.06629。

[23]タオ•リン、セバスチャン•U•スティッチ、マーティン•ジャギ。大きなミニバッチを使用せず、ローカルSGDを使用します。arXiv：1808.07217 [cs、stat]、2018年8月。arXiv：1808.07217。

[24] Xiangru Lian、Ce Zhang、Huan Zhang、Cho-Jui Hsieh、Wei Zhang、およびJiLiu。分散型アルゴリズムは集中型アルゴリズムよりも優れたパフォーマンスを発揮できますか？分散型並列確率的勾配降下法のケーススタディ。ニューラル情報処理システムの進歩、5330〜5340ページ、 2017年。

[25] Lie He、An Bian、およびMartinJaggi。コーラ：分散型線形学習。ニューラル情報処理システムの進歩、ページ4541–4551、2018。

[26]アンジェリア•ネディックとアレックス•オルシェフスキー。時間変化する有向グラフの分散最適化。自動制御に関するIEEEトランザクション、60（3）：601–615、2015年。

[27] Xiangru Lian、Wei Zhang、Ce Zhang、およびJiLiu。非同期分散型並列確率的勾配降下法。機械学習に関する国際会議、2018年。

[28] Tianyu Wu、Kun Yuan、Qing Ling、Wotao Yin、およびAliH.Sayed。非同期性と遅延を伴う分散型コンセンサス最適化。ネットワークを介した信号および情報処理に関するIEEEトランザクション、PP：1–1、2017年4月。

[29] Zebang Shen、Aryan Mokhtari、Tengfei Zhou、Peilin Zhao、およびHuiQian。より効率的な確率的分散学習に向けて：より速い収束とまばらなコミュニケーション。 JenniferDyとAndreasKrauseの編集者、Proceedings of the 35th International Conference on Machine Learning、Proceedings of Machine Learning Research、pages 4624–4633、 Stockholmsmässan、Stockholm Sweden、2018年7月10〜15日。PMLR。

[30] Youjie Li、Mingchao Yu、Songze Li、Salman Avestimehr、Nam Sung Kim、AlexanderSchwing 。 Pipe-sgd：分散型ディープネットトレーニング用の分散型パイプラインsgdフレームワーク。
S. Bengio、H。Wallach、H。Larochelle、K。Grauman、N。Cesa-Bianchi、およびR. Garnettの編集者、 Advances in Neural Information Processing Systems 31、8056〜8067ページ。 Curran Associates、 Inc.、2018年。

[31] Elad Hazanetal。オンライン凸最適化の概要。 Foundations and Trends R in Optimization、2（3-4）：157–325、2016年。

[32] Shai Shalev-Shwartzetal。オンライン学習とオンライン凸最適化。財団と
Trends R in Machine Learning、4（2）：107–194、2012。

[33] Michael Kamp、Mario Boley、Daniel Keren、Assaf Schuster、およびIzchakSharfman。
動的モデル同期による通信効率の高い分散オンライン予測。
機械学習とデータベースでの知識発見に関する欧州合同会議、 623〜639ページ。 Springer、2014年。

[34] ShahinShahrampourとAliJadbabaie。ミラー降下を使用した動的環境での分散オンライン最適化。自動制御に関するIEEEトランザクション、63（3）：714–725、 2017年。

[35] Soomin Lee、Angelia Nedic、およびMaximRaginsky。分離不可能なグローバル目標を持つ分散型オンライン最適化のための二重平均化を調整します。ネットワークシステムの制御に関するIEEEトランザクション、5（1）：34–44、2016年。

[36] Yawei Zhao、Chen Yu、Peilin Zhao、およびJiLiu。分散型オンライン学習：自分のデータを共有せずに他の人のデータを活用して、世界的な傾向を追跡します。 arXiv preprint arXiv：1901.10593、2019 。

[37]ジョン•ダッチ、アレフ•アガルワル、マーティン•J•ウェインライト。分散最適化のための二重平均化：収束分析とネットワークスケーリング。自動制御に関するIEEEトランザクション、57（3）：592–606、2011。

[38]バーマン•ガレシファードとホルヘ•コルテス。有向グラフが二重確率隣接行列を認めるのはいつですか？ 2010
American Control Conferenceの議事録、2440〜2445ページ。 IEEE、 2010年。

[39] S Sundhar Ram、Angelia Nedic、およびVenugopalVVeeravalli。凸最適化のための分散確率的劣勾配射影アルゴリズム。 Journal of Optimization Theory and Applications、 147（3）：516–545、2010年。

[40] KonstantinosITsianosとMichaelGRabbat。通信遅延下での凸型最適化のための分散デュアル平均化。 2012
American Control Conference（ACC）、 1067〜1072ページ。 IEEE、2012年。

[41] Konstantinos I Tsianos、Sean Lawlor、およびMichaelGRabbat。凸最適化のためのプッシュサム分散デュアル平均化。 2012年のIEEE51stIEEE Conference on Decision and Control（CDC）、5453〜5458ページ。 IEEE、2012年。

[42] AurélienBellet、Rachid Guerraoui、Mahsa Taziki、およびMarcTommasi。パーソナライズされたプライベートピアツーピア機械学習。 arXiv preprint arXiv：1705.08435、2017。

[43]ミラッド•ナスル、レザ•ショクリ、アミール•フマンサドル。ディープラーニングの包括的なプライバシー分析：パッシブおよびアクティブホワイトボックス推論攻撃の下でのスタンドアロンおよび連合学習。 arXiv preprint arXiv：1812.00910、2018。

[44] Chaoyang He、Songze Li、Jinhyun So、Mi Zhang、Hongyi Wang、Xiaoyang Wang、Praneeth Vepakomma、Abhishek Singh、Hang Qiu、Li Shen、Peilin Zhao、Yan Kang、Yang Liu、Ramesh Raskar、Qiang Yang、Murali Annavaram 、およびSalmanAvestimehr。 Fedml：連合機械学習のための研究図書館とベンチマーク。 arXivプレプリントarXiv：2007.13518、2020。

[45]アンジェリア•ネディックとアレックス•オルシェフスキー。時間変化する有向グラフの分散最適化。
自動制御に関するIEEEトランザクション、60（3）：601–615、2014。

[46]アンジェリア•ネディックとアレックス•オルシェフスキー。確率的勾配降下法は、時変有向グラフ上の強く凸な関数をプッシュします。自動制御に関するIEEEトランザクション、61（12）：3936–3947、 2016年。

[47]マフムード•アスランとマイケル•ラッバート。非同期劣勾配-プッシュ。 arXiv preprint arXiv：1803.08950、2018。

[48] Mahmoud Assran、Nicolas Loizou、Nicolas Ballas、およびMichaelRabbat。分散ディープラーニングのための確率的勾配降下法。 arXiv preprint arXiv：1811.10792、2018。

付録

表記法 :以下では、証明に次の表記法を使用しています

…
、

• ∇ft（Xt） ≥ h∇F1、tx • Xt ≥ hx 私

、 、 …、
私

• Gt ≥ ∇f1、t（xt;ξ

ここでは、最初に証明定理2を提示し、次に定理3の証明とともにいくつかの重要な補題を提示します。

7定理2と定理3の証明

次の定理は、定理2を証明するための鍵です。定理4。ステップサイズγ> 0のオンラインプッシュサムアルゴリズムの場合、次のようになります。

nR2
RT≤G2TnyC1 + σ _ 2Tγ（1 + nC2）+ 2c 、 (4)

どこ

8Cq 2Cq C1 := + 1、 C2 := δmin
（1 − q） δmin（1 − q）は後の見出語で定義され、
る定数です。

証拠。損失関数fi、t（・）は凸であると想定されるため、次のようになります。

--nFt（x ）

(i) ;
–fi、t x パッ

(i)
≤EtXn _ D∇fi、t x t ;パッ 、 − x AND

—
、x t
D∇fi、t x D∇fi、t x
| ———————— tと ———————— | | ———————— tと ———————— |

—
、 。

I2tの場合、

—
、
D∇fi、t x

n —
= — —
c n

—
、

n c —
= —— —— + kzt − x k — — —
n n

2 _

n c —
= —— —— + kzt − x k − kzt + 1 − x k
n n

2 _

n —
_ _ _ _ Etγ2G2≤2γ γ2σ2 + kzt -x — − kzt + 1 − x k
k +
n

Et *、YXn (i) 登録 (zt E + Et Xn − x Xn (i) nV fi、t x t ξ

COLの場合、 x = ztであるため、 I1t =0であることに注意してください。したがって、DOLの場合、 I1tをバインドするには、差x − ztを制限する（補題8を使用）。

$$\text{そして}_{i=1}^{n} D\nabla f_{i,t}(x_t^{(私)};\text{バツ}_t^{(i)}, x_t^{(i)}) - \bar{z}_t E$$

$$=\text{そして}_{i=1}^{n} D\nabla F_{i,t}(x_t^{(i)}), x_t^{(私)}) - \bar{z}_t E$$

$$\leq E_t \text{Xn}_{i=1} \left(\alpha \nabla F_{i,t}(x_t^{(i)}) \right)^2 + \frac{1}{a} \left(x_t^{(私)} - \bar{z}_{現在} \right)^2 。$$

上記の不等式をt=1からt=Tまで合計すると、次のようになります。

$$\text{バツ}_{t=1}^{T} \text{そして}_{i=1}^{n} D\nabla f_{i,t}(x_t^{(私)};\text{バツ}_t^{(i)}, x_t^{(私)}) - \bar{z}_t E$$

$$= \text{X}_{t=1}^{T} \text{そして}_{i=1}^{n} D\nabla F_{i,t}(x_t^{(私)}, x_t^{(私)}) - \bar{z}_t E$$

$$\text{Xn} \leq \text{X}_{t=1}^{T} \text{そして}_{i=1} \left(\alpha \nabla F_{i,t}(x_t^{(i)}) \right)^2 + \frac{1}{a} \left(x_t^{(私)} - \bar{z}_{現在} \right)^2$$

$$= \text{X}_{t=1}^{T} \alpha E_t \|\nabla F_t(X_t)\|_F^2 + \frac{1}{a} E_t \|X_t - z_t\|_F^2$$

$$\leq \alpha \text{X}_{t=1}^{T} E_t \|\nabla F_t(X_t)\|_F^2 + \frac{4c\,2C^2\,2^{2q}}{\|G_t\|\alpha\delta_{2\min}(1-q)^2} \text{X}_{t=1}^{T} \text{そして}_F^2$$

$$\leq \alpha \text{X}_{t=1}^{T} E_t \|\nabla F_t(X_t)\|_F^2 + \frac{4c\,2C^2\,2^{2q}}{\alpha\delta_{2\min}(1-q)^2} \text{X}_{t=1}^{T} E_t \|\nabla F_t(X_t)\|_F^2 + n\sigma^2 。$$

α=を選択すると、 $\delta_{\min} \geq \frac{2\gamma C_q}{(1-q)}$が得られます。

$$\text{バツ}_{t=1}^{T} \text{そして}_{i=1}^{n} D\nabla f_{i,t}(x_t^{(私)};\text{バツ}_t^{(i)}, x_t^{(私)}) - \bar{z}_t E \leq \frac{}{8n\gamma C_T q G_2 \delta_{\min}} + \frac{}{2(1-q)^2 \delta_{\min}(1-q)}$$

だから私たちは

$$\text{バツ}_{t=1}^{T} \text{そして}_{i=1}^{n} f_{i,t}(z_t;\text{バツ}_t^{(i)}, \text{バツ}_t^{(i)}) - nF(x^*)$$

$$8n\gamma C_T q G_2 \leq \delta_{\min} \frac{}{(1-q)} + \frac{}{2\gamma C_q \sigma^2 T \delta_{\min}} + \frac{n}{2n\gamma(1-q)} \text{バツ}_{t=1}^{T} \gamma G_2 + \frac{\gamma^2\sigma^2}{n} + E_t \|z_t - \bar{x}\|^2 - E_t \|z_{t+1} - \bar{x}\|^2$$

$$\leq G_2 T n\gamma + \frac{8C_q}{(1-q)} + 1 + \sigma\delta_{\min}(1-q) + 2T\gamma 1 + \delta_{\min} \frac{2nC_q}{(1-q)} + \frac{n}{2c} \text{バツ}_{t=1}^{T} \text{そして}\|z_t - \bar{x}\|^2 - E_t \|z_{t+1} - \bar{x}\|^2$$

$$\leq G_2 T n\gamma + \frac{8C_q}{(1-q)} + \frac{2nC_q}{(1-q)} \sigma\delta_{\min}(1-q) + 2T\gamma\delta_{\min} + 1 + \frac{}{} + \frac{nR^2}{2c}$$

$$= C_1 n G_2 T\gamma + (1 + nC_2)\sigma^2 T\gamma + 2\gamma \frac{nR^2}{} 。$$

定理2は、$\gamma = \frac{\sqrt{nR}}{\sqrt{(1+nC_2)\sigma^2 + nC_1 G_2 T}}$を設定することで簡単に検証できることに注意してください。 □

次に、補題8の証明のために2つの補題を提示します。次の2つの補題の証明は、既存の文献[45、46、47、48]にあります。

補題5.仮定1の下で、定数δmin > 0が存在するため、任意のtに対して、次のことが成り立ちます。

$$\sum_{j=1}^{n} [W_t > W_{t-1} \cdots W_0 >]_{ij} \geq \delta_{\min} \geq \frac{1}{n}, \quad \forall i \tag{5}$$

ここで、 W_tは行確率行列です。

補題6.仮定1の下では、任意のtに対して、確率ベクトルψ（t)と2つの定数C=4およびq=1 − n <1が常に存在し、s≤tを満たす任意のsに対して、次の不等式が成り立ちます。

$$[W_t > W_{t-1} \cdots W_{s+1} > W_s >]_{ij} - \psi_i(t) \leq C q^{t-s}, \forall i, j$$

ここで、 W_tは行確率行列であり、ψ（t)はψi（t)その i番目のエントリです。

補題7.2つの非負のシーケンスが与えられた$\{a_t\}_{t=1}^{\infty}$および$\{b_t\}_{t=1}^{\infty}$その満足

$$a_t = \sum_{s=1}^{t} \rho^{t-s} b_s, \tag{6}$$

ρ∈[0,1)の場合、次のようになります。

$$D_k := \sum_{t=1}^{\infty} a_t^2 \leq \frac{1}{(1-\rho)^2} \sum_{s=1}^{\infty} b_s^2.$$

証拠。定義から、

$$S_k = \sum_{t=1}^{\infty} \sum_{s=1}^{t} \rho^{t-s} b_s = \sum_{s=1}^{\infty} \sum_{t=s}^{\infty} \rho^{t-s} b_s = \sum_{s=1}^{\infty} \sum_{t=0}^{\infty} \rho^t b_s \leq \sum_{s=1}^{\infty} \frac{b_s}{1-\rho}, \tag{7}$$

$$\begin{aligned}
D_k &= \sum_{t=1}^{\infty} \sum_{s=1}^{t} \rho^{t-s} b_s \sum_{r=1}^{t} \rho^{t-r} b_r \\
&= \sum_{t=1}^{\infty} \sum_{s=1}^{t} \sum_{r=1}^{t} \rho^{2t-s-r} b_s b_r \\
&\leq \sum_{t=1}^{\infty} \sum_{s=1}^{t} \sum_{r=1}^{t} \rho^{2t-s-r} \frac{b_s^2 + b_r^2}{2} \\
&= \sum_{t=1}^{\infty} \sum_{s=1}^{t} \sum_{r=1}^{t} \rho^{2t-s-r} b_s^2 \\
&\leq \frac{1}{1-\rho} \sum_{t=1}^{\infty} \sum_{s=1}^{t} \rho^{t-s} b_s^2 \\
&\leq \frac{1}{(1-\rho)^2} \sum_{s=1}^{\infty} b_s^2.
\end{aligned}$$

□

上記の3つの補題に基づいて、次の補題を得ることができます。

補題8.仮定1の下で、アルゴリズム1の更新規則は、次の不等式につながります。

$$\sum_{k=0}^{T} \rho^{T-t} \left(\frac{1}{n} - \frac{1}{n} + \frac{\gamma_t}{2}\right)^2 \leq \frac{4\gamma_0^2 C^2 \delta_2^2}{(1-q)^2} + \sum_{t=0}^{T} \|G_t\|_F^2, \tag{8}$$

ここで、 γはステップサイズ、 C = 4、 δmin≥n − n 、 q = 1 − nは定数です。 Gtは、時間tでの確率的勾配の行列です（たとえば、 i番目の列は時間tでのノードiの確率的勾配ベクトルです）。

証拠。 OPSの更新規則は、Zt + 1 = （ Zt −γGt)Wωt +1 = W> ωtXt+

1 = Zt + 1 [diag（ωt+ 1）]-1と

して定式化できます。

ここで、 Wは行確率行列です。 Xt = [x t (1)、x t (2)、...、x t （n）]は、各列が x t (i) (i)。 Gtは勾配の行列であり、その各列はzでの確率的勾配です。 t ノードi。 Zt = [z t (1)、...、 t z （n）（i）]は、各列がzである行列です。 t 。

X0 =Oおよびω0 =1と仮定すると、次のようになります。

$$Zt + 1 = （ Zt − γGt)W = \quad − γ Xt \sum_{s=0} GsWt − s + 1 、 \tag{8}$$

$$\overline{zt} + 1 = zt \overline{−} γgt = \overline{\quad} = − xt \sum_{s=0} γ\overline{gs} 、 \tag{9}$$

$$ωt+1 = Wt + 1>ω0、 \tag{10}$$

ここで、 $\overline{xt}$ = Xt1はn個のノード上のすべての変数の平均であり、 gt = Gt1は平均化された勾配です。 Wは行確率行列であるため、W1=1になります。

ωt+1の場合、補題6によれば、次のように分解します。

$$ωt+1 = Wt+1>ω0 = [Wt+ 1> --ψ（t)1>]ω0 +ψ（t)1>ω0 = [Wt + 1> --ψ（t)1>] 1 +nψ（ t）、 \tag{11}$$

ω0 =1なので。

一方、補題5によると、

$$ω^{(私)}_{t+1} = [Wt + 1> 1]> ei = Xn \sum_{j=1} [Wt + 1>] ij ≥ nδmin 、 \tag{12}$$

ここで、 eiは、i番目のエントリのみが1で、他のエントリは0であるベクトルです。

次の用語をさらに制限する必要があります

$$(i) \quad x − \overline{zt} +1 = γt+ 1 \quad \frac{と^{(i)}_{t+1}}{ωt + 1^{(j)}} − \overline{zt} + 1$$

$$= γXt \sum_{s=0} \frac{GsWt − s + 1ei}{1> Wt + 1ei} − \frac{Gs1}{n}$$

$$= γXt \sum_{s=0} \frac{nGsWt − s + 1ei − Gs11> Wt + 1ei \quad (i)}{nωt + 1} 、$$

ここで、2番目の等式は（8）、（9）、および（10)によるものです。次の用語をバインドするようになります

$$Xt \sum_{s=0} \frac{nGsWt − s + 1ei − Gs11> Wt + 1ei \quad (i)}{nωt + 1}$$

$$≤ \frac{1}{n2δmin} \sum_{s=0} Xt \quad nGsWt − s + 1ei − Gs11> Wt + 1ei 、$$

ここで、最初の不等式は（12)に従います。したがって、上記の結果を組み合わせると、

$$\sum_{i=1} Xn \quad x^{(私)}_{t+1} − \overline{zt} + 1 \quad 2 \quad γ ≤ \frac{2}{n4δ2} \sum_{i=1} Xn \sum_{s=0} Xt \quad nGsWt − s + 1ei − Gs11> Wt + 1ei \quad 2$$

$$2γ ≤ \frac{2}{n4δ2} \sum_{s=0} Xt \quad nGsWt − s + 1 − Gs11> Wt + 1 \quad F$$

ここで、2番目の不等式はPnによるものです $\sum_{i=1} \|Ae_i\|_2^2 = \|A\|_{2F}$。

次の用語を制限することは残っています

$$\sum_{s=0}^{t} \|G_s W_{t-s+1} - G_{s11} > W_{t+1}\|_F^2$$

$$= \sum_{s=0}^{t} \|G_s W_{t-s+1} - G_{s1}[1 > (W_{t+1} - \psi(t)1>) > + n\psi(t) >]\|_F^2$$

$$= \sum_{s=0}^{t} \|G_s [W_{t-s+1} - 1\psi(t) >] - G_{s11} > [W_{t+1} - 1\psi(t) >]\|_F^2$$

$$\leq \sum_{s=0}^{t} \|G_s [W_{t-s+1} - 1\psi(t) >]\|_F + \sum_{s=0}^{t} \|G_{s11} > [W_{t+1} - 1\psi(t) >]\|_F \|_2$$

$$\leq n \sum_{s=0}^{t} \|G_s\|_k \|[W_{t-s+1} - 1\psi(t) >]\|_F + \sum_{s=0}^{t} \|G_{s11} > \|_k \|[W_{t+1} - 1\psi(t) >]\|_F \|_2$$

$$\leq \eta_2 \sum_{s=0}^{t} \|G_s\|_k \|[W_{t-s+1} - 1\psi(t) >]\|_F + \sum_{s=0}^{t} \|G_s\|_k \|[W_{t+1} - 1\psi(t) >]\|_F \|_2$$

$$\leq \eta_2 \sum_{s=0}^{t} nC_{q^{t-s+1}} \|G_s\|_F + \sum_{s=0}^{t} nC_{q^{t+1}} \|G_s\|_F \|_2$$

$$\leq 4n4C^2 \frac{q^{t-s}}{\eta_2} \sum_{s=0}^{t} \|G_s\|_F \|_2$$

ここで、3番目の不等式は$\|k_{11} > \|_F = n$によるものであり、4番目の不等式は補題6によるものであり、$\|A\|_F \leq n \cdot \max_{i,j} |A_{ij}|$ $A \in R^{n \times n}$の場合。

したがって、上記のすべての不等式を組み合わせると、次のようになります。

$$\sum_{i=1}^{n} \left\| x^{(i)} - z_{t+1} \bar{t}_{t+1} \right\|_2^2 \leq \frac{4\gamma_2 C^2 \leq \delta_2^2 q}{\text{分}} \sum_{s=0}^{t} q^{t-s} \|G_s\|_F \|_2 。$$

補題7を使用すると、

$$\sum_{t=0}^{T} \sum_{s=0}^{t} q^{t-s} \|G_s\|_F \|_2 \leq \frac{1}{(1-q)^2} \sum_{t=0}^{T} \|G_t\|_k^2 、$$

これは

$$\sum_{t=0}^{T} \sum_{i=1}^{n} \left\| x_{t+1}^{(i)} - z_{t+1} \bar{t}_{t+1} \right\|_2^2 \text{分} \frac{4c^2 C^2 q^2}{(1-q)^2} \sum_{t=0}^{T} \|G_t\|_k^2 \leq \delta_2^2 、$$

これで証明が完成します。 □

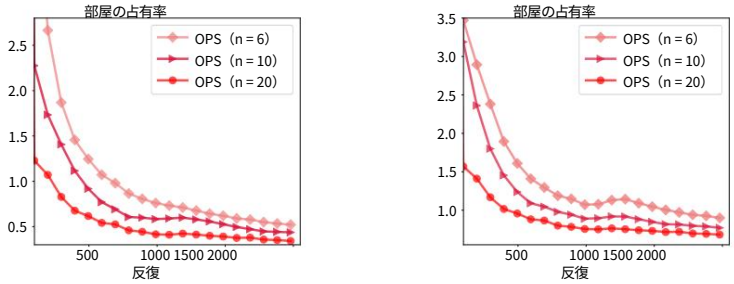
実際、定理3は、γを適切な値として設定することによる補題8の結果です。

8追加の実験結果

8.1部屋占有率データセットの評価

スペースの制限により、セクション5.3および5.4ではSUSYデータセットの実験結果のみを示します。部屋の占有率に関する関連するプレゼントを図4と図5に示します。

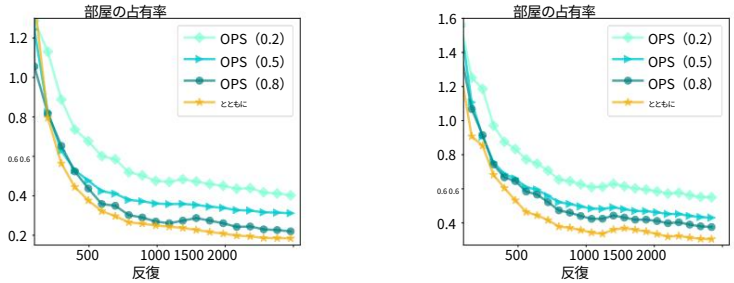
図4では、ネットワーク内のクライアントの数を6から20まで変化させています。図5では、ネットワーク密度を変化させています。すべての結果はSUSYの結果と一致しています。



(a)確率= 100%
#Neighbors = 2

(b)確率= 50%
#Neighbors = 2

図4 :ネットワークサイズの評価



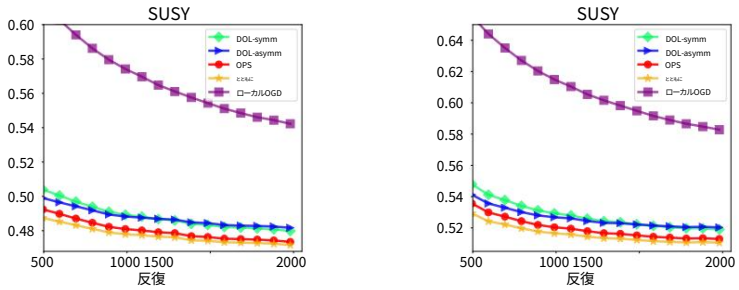
(a)確率= 100%

(b)確率= 50%

図5 :ネットワーク密度の評価

8.2ローカルオンライン勾配降下法との比較

通信の必要性を正当化するために、OPSをローカルオンライン勾配降下法（ローカルOGD)と比較します。この場合、すべてのノードが他のノードと通信せずにローカルモデルをトレーニングします。図2の設定に基づいて、敵対的要素と確率的要素のさまざまな比率で実験を実行します。図6に示すように、コミュニケーションが後悔を減らすのに役立つことを経験的に証明します。さらに、確率的要素の比率が増加するにつれて、OPSの後悔はさらに減少します。これはまた、確率的要素がコミュニケーションから利益を得ることができるが、敵対的要素はそうではないことを経験的に証明しています。



(a)確率= 100%（n = 128、#Neighbors = 32）

(b)確率= 50%（n = 128、#Neighbors = 32）

図6 :OPSとローカルOGDの比較。