局所差分プライバシを適用したFederated Learningの安全性評価

松本　茉倫† 　高橋　翼†† 　リュウセンペイ†† 　小口　正人†

† お茶の水女子大学　〒 112–8610 東京都文京区大塚 2-1-1
†† LINE 株式会社　〒 160–0004 東京都新宿区四谷 1-6-1 四谷タワー 23 階
E-mail: †marin@ogl.is.ocha.ac.jp, oguchi@is.ocha.ac.jp, ††{tsubasa.takahashi,sengpei.liew}@linecorp.com

あらまし　クライアントに分散された機微データをプライバシ保護しながら活用し，機械学習モデルの訓練する方法として，局所差分プライバシ (以下 LDP: Local Differential Privacy) を適用した Federated Learning がある．LDP は，プライバシパラメータ ϵ で表される程度に情報の識別性を困難にすることができる一方で，どういった攻撃に対してどの程度の強度があるのかは未知であり，ϵ の決定に必要な判断材料が不足している．そこで本研究では，Federated Learning で送信する勾配の判別可能性を検査し，経験的なプライバシ強度を得ることを考える．このとき，2 つの勾配を判別可能な確率が高くなるほどにランダム化手法のプライバシ強度が十分でなく，逆に判別可能な確率が低くなるほどプライバシ強度が高いことを示すことができる．この検査では，現実的な設定のクライアントとサーバを想定した場合には理論的なプライバシ強度と経験的なプライバシ強度のギャップが大きく，クライアントとサーバが共謀する想定の場合は理論的なプライバシ強度と経験的なプライバシ強度のギャップが小さいことを示す．

キーワード　局所差分プライバシ, Federated Learning

1　はじめに

機械学習や統計分析を行う際，収集するデータの中には個人のプライバシに関する情報が含まれる場合がある．そのため，機微データをプライバシ保護しながら収集・活用する手法が必要とされており，研究が盛んに行われている．中でも，差分プライバシ (以下 DP: Differential Privacy) [1] と呼ばれるプライバシ基準が広く認められてきている．計算機構が (ϵ, δ)-DP を満たす場合，計算機構による出力を公開したとしても，(ϵ, δ) で示される程度に個人のプライバシが厳密に保護される．直感的には，出力を見たとしても出力に加えられた乱数に基づくノイズのために，データセットに任意の個人の情報が含まれていたかの推測が難しくなることが保証される．しかし，標準的な集中型の DP(以下 CDP: Central Differential Privacy) では，信頼できるデータ収集者が正しく DP を満たす計算機構を使用することを前提としているため，データ収集者が信頼できない場合には利用できない．そこで，データ収集者を信頼する必要のない局所差分プライバシ (以下 LDP: Local Differential Privacy) [2] が提案された．LDP では，データ収集者も攻撃者になり得るとして，データ提供者がデータを提供する前にデータへのノイズ付与やランダム化によって自身のプライバシを保護する．直感的には，LDP を適用することによって，どんな 2 つの入力でも ϵ で表される程度に識別を困難にする．

LDP の応用としては，分散強化学習 [3] や生データを集約せずにモデルの更新のみを交換しながら協調的な機械学習を行う Federated Learning [4] が挙げられる．本研究では，クライアントは自身の機微データの勾配を計算し，LDP を保証するようにランダム化してサーバに送信することによる，プライバシ保護型の Federated Learning を考える．このような LDP を適用した Federated Learning を実施する研究として，プライバシ強度を保ちながら有用性を損なわないメカニズムに関する研究 [5]，ランダム化の際にデータの次元に比例して増加するノイズの削減が可能なフレームワークに関する研究 [6]，Membership Inference などのバックドア攻撃とその防御に関する研究 [7] が提案されている．LDP はプライバシパラメータ ϵ で表される程度に個人のデータの識別を困難にすることができる一方で，ϵ をどのように解釈・決定すれば良いかの判断材料が不足している．

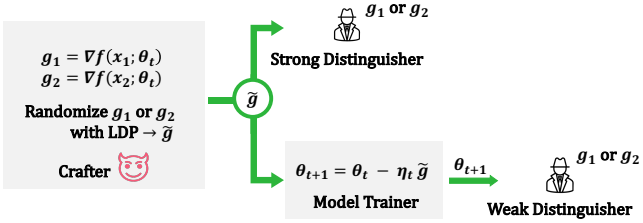


図 1: LDP を適用した Federated Learning のプライバシ強度を検査する方法．*Crafter*(malicious なクライアント) が 2 つの勾配のうち 1 つをランダム化し，*Model Trainer*(サーバ) はランダム化された勾配 $\tilde{g}$ でモデルを更新する．*Distinguisher* は，$\tilde{g}$ または θ_{t+1} からどちらの勾配がランダム化されたのかを予測する．

先行研究 [8] では CDP を保証した機械学習アルゴリズムである DP-SGD(Differential Private Stochastic Gradient Descent) [9] のプライバシ強度を検査した．検査方法は以下の 3 つの工程である．

（1） *Crafter* が 1 レコードのみ異なるデータセット D と D' を生成．

（2） *Model Trainer* が D または D' をランダムに学習デー

タとして選び，DP-SGD アルゴリズムで学習したモデルを出力．

（3） *Distinguisher* が学習済みモデルから学習に使われたデータが D と D' のどちらかを当てる．

この試行を十分な回数行い，出力モデルから学習データを正しく予測された確率を算出することで，CDP を保証した機械学習の経験的なプライバシ強度とすることができる．実験では，API のように推論結果しか得られない場合，学習途中のパラメータが得られる場合，*Crafter* が出力結果の差が出やすくするようにデータセットを加工した場合などアクセスレベルに応じた攻撃を行った．強い攻撃であるほど経験的なプライバシ強度は理論値，つまりあらかじめ設定した ϵ に近く，逆にアクセスレベルが最も制限された API の場合は理論値と経験的なプライバシ強度のギャップが大きいことが示された．

ここで，LDP を適用した Federated Learning で想定される攻撃は，信頼できるデータ収集者を必要とする DP-SGD と全く同じであると言えるだろうか．データ収集者を信頼しない LDP では，悪意のあるデータ収集者からの攻撃や自分以外のクライアントからの攻撃も考えられる．本研究では，図 1 のような検査を行うことで，LDP を適用した Federated Learning において，どういった攻撃に対してどの程度の強度があるのかを明らかにする．検査方法は以下の 3 つの工程である．

（1） *Crafter* が 2 つの勾配 g_1, g_2 のうち 1 つをランダム化し，$\tilde{g}$ とする．

（2） *Model Trainer* は $\tilde{g}$ で学習したモデル θ_{t+1} を出力．

（3） *Strong Distinguisher* は $\tilde{g}$ から，*Weak Distinguisher* は θ_{t+1} から，ランダム化された勾配が g_1 と g_2 のどちらかを当てる．

このとき，2 つの勾配を判別可能な確率が高くなるほどにランダム化手法のプライバシ強度が十分でなく，逆に判別可能な確率が低くなるほどプライバシ強度が高いことを示すことができる [10]．本研究は図 2 のように，現実的な設定のクライアントとサーバを想定した場合には理論的なプライバシ強度と経験的なプライバシ強度のギャップが大きく，クライアントとサーバが共謀する想定の場合は理論的なプライバシ強度と経験的なプライバシ強度のギャップが小さいことを示し，LDP を適用した Federated Learning における ϵ の解釈を助けるものである．

2 準　　備

2.1 表 記 法

本論文全体で使用する表記法について記載する．X をレコードのドメイン，レコード $x \in X$ を個人の情報を含むデータ，レコードの集合をデータベース $D = \{x_i\}_{i=1}^n$ とする．データベース $D \in \mathcal{D}$ を引数に取り，乱数に基づくノイズを加えた応答値 $y \in Y$ を返す計算機構を M と置く．ここで取りうるデータベースの集合を $\mathcal{D}$，クエリ応答値にノイズを加えた結果得られる値の集合を Y とした．2 つの同じ大きさのデータベース D, D' において，同一でないレコードの数が 1 つの場合，D, D' は隣接しているという．

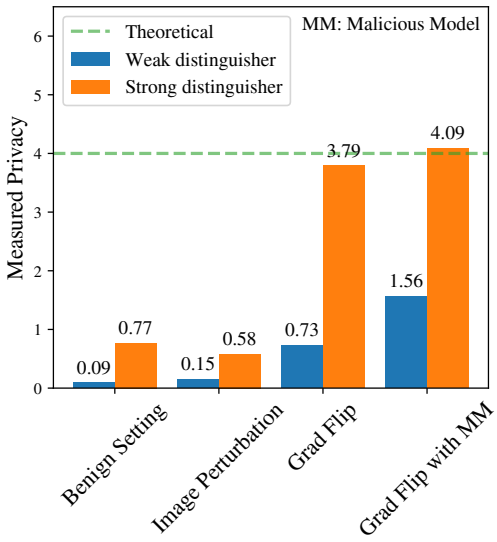


図 2: LDP のプライバシパラメータを $\epsilon = 4$ に設定した Federated Learning の経験的なプライバシ強度．理論的なプライバシ強度 (*Theoretical*) と経験的なプライバシ強度のギャップは，クライアントとサーバが現実的な設定の場合に大きく，クライアントとサーバが共謀する設定の場合は小さい．

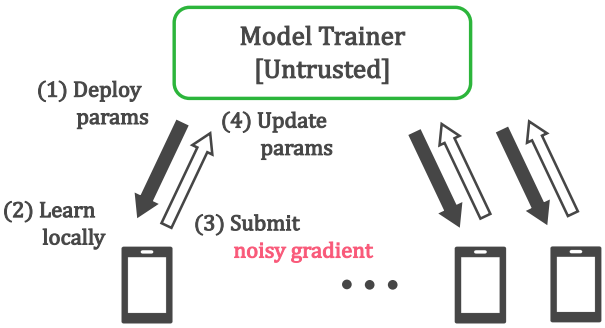


図 3: 本研究で想定する Federated Learning．*Model Trainer* から配布されたパラメータでクライアントは勾配を生成し，LDP を保証するように勾配をランダム化する．*Model Trainer* はクライアントから集めた勾配でパラメータを更新する．

2.2 ϵ-局所差分プライバシ

CDP はデータ収集者が統計量を公開する際に，隣接データベースの識別不能性を保証することで個人のプライバシを保護する．このとき，信頼できるデータ収集者が正しく DP を満たす計算機構が必要となる．一方で LDP は，データ収集者を信頼せずにデータ提供者が提供の前に自身のデータにノイズを加えることでプライバシを保護する．この場合，各個人が 1 つのデータで構成されるデータベースを所持しており，データそのものという統計量を公開すると捉えることもできる．その場合，隣接データベースはドメイン上の任意のデータであり，CDP と同様に任意の隣接データベースとの識別不能性を保証することでプライバシは保護することができると考える．$\epsilon \in \mathbb{R}^+$ について LDP は以下のように定義される．

定義 1 (ϵ-局所差分プライバシ)．$\forall x, x' \in X$ および，任意の出

力 $y \in Y$ について，

$$\frac{Pr(M(x)=y)}{Pr(M(x')=y)} \leq e^{\epsilon} \tag{1}$$

を満たすとき，計算機構 M は ϵ-LDP を満たすという．

直感的には計算機構 M に x を入力として出力しても，任意のデータ x' を入力とした場合の出力と識別することができないため，本来のデータが何であったかが推測できないことを保証している．

2.3 Federated Learning

Federated Learning [4] は分散型の機械学習手法である．従来の機械学習と Federated Learning の大きな違いはクライアントのデータがサーバや他のクライアントに共有されない点である．Federated Learning の代表的な手法である FedAvg [4] では，以下のように学習する．

（1） 各クライアントがグローバルモデル θ_t をダウンロードし，ローカルデータ x_i を用いてモデルを学習．

（2） 各クライアントは学習後の勾配 $\nabla_{\theta_t} f(x_i)$ をサーバに送信．

（3） サーバは $(\nabla_{\theta_t} f(x_i))_{i\in[n]}$ を平均化して 1 つのグローバルモデルを作成．

Federated Learning に関する研究には，通信コストに関する研究 [11]，セキュリティに関する研究 [12] [13] などが行われている．その他にも，Adam などの最適化手法の応用 [14] やフレームワークやライブラリ [15] [16] など幅広く研究が行われている．

本研究で想定する Federated Learning は図 3 に示すように，信頼できない *Model Trainer*(サーバ) と機微データを所有するクライアントから構成される． まず，*Model Trainer* から配布されたパラメータで，機微データを所有するクライアントが勾配を生成する．次にクライアントは LDP を保証するように勾配をランダム化して *Model Trainer* に送信する．*Model Trainer* はクライアントから集めた勾配を使ってパラメータを更新する．

2.4 Locally Differentially Private Stochastic Gradient Descent

Federated Learning では，クライアントが公開する勾配から元の画像は復元可能であることが指摘されている [17] [18]．元画像の復元を防ぐ手法としては，勾配の LDP を適用したランダム化が挙げられる．本項では，勾配をランダム化するアルゴリズムである LDP- SGD(Locally Differentially Private Stochastic Gradient Descent) [19] [20] について述べる．

アルゴリズム 1 に示した LDP-SGD のクライアント側では，まず勾配のノルムが最大でも L になるようにクリッピングする．次にノルムに比例して表が出やすくなるコインを投げ，裏が出た場合には勾配の符号を逆転させる．最後に，ϵ に比例して表が出る確率が高くなるコインを投げ，表が出た場合には元の勾配との内積が正になるように一様分布からサンプリングし，裏が出た場合には内積が負になるように一様分布からサンプリングする．直感的に説明すると，勾配は図 4 のように表が出れば緑色の範囲からサンプリングされ，裏が出れば白色の範囲からサンプリングされる．

Algorithm 1 LDP-SGD; client-side [20]

Require: Local privacy parameter: ϵ_ℓ, current model: $\theta_t \in \mathbb{R}^d$, ℓ_2-clipping norm: L

1: Compute clipped gradient
$x \leftarrow \nabla\ell(\theta_t; d) \cdot \min\left\{1, \frac{L}{||\nabla\ell(\theta_t; d)||_2}\right\}$

2: $z_i \leftarrow \begin{cases} L \cdot \frac{x}{||x||_2} & w.p. \frac{1}{2} + \frac{||x||_2}{2L} \\ -L \cdot \frac{x}{||x||_2} & otherwise. \end{cases}$

3: Sample $v \sim_u S^d$, the unit sphere in d dimensions.
$\hat{z} \leftarrow \begin{cases} \text{sgn}(\langle z, v\rangle) \cdot v & w.p. \frac{e^{\epsilon_\ell}}{1+e^{\epsilon_\ell}} \\ -\text{sgn}(\langle z, v\rangle) \cdot v & otherwise. \end{cases}$

4: **return** $\hat{z}$

Algorithm 2 LDP-SGD; server-side [20]

Require: Local privacy budget per epoch: ϵ_ℓ, number of epochs: T, parameter set: C

1: $\theta_0 \leftarrow \{0\}^d$
2: **for** $t \in [T]$ **do**
3: Send θ_t to all clients
4: Collect shuffled responses $(\hat{z}_i)_{i\in[n]}$
5: Noisy gradient: $g_t \leftarrow \frac{L\sqrt{\pi}}{2} \cdot \frac{\Gamma(\frac{d-1}{2}+1)}{\Gamma(\frac{d}{2}+1)} \cdot \frac{e^{\epsilon_\ell}+1}{e^{\epsilon_\ell}-1} \left(\frac{1}{n}\sum_{i\in[n]} \hat{z}_i\right)$
6: Update: $\theta_{t+1} \leftarrow \prod_C(\theta_t - \eta_t \cdot g_t)$, where $\prod_C(\cdot)$ is the ℓ_2-projection onto set C, and $\eta_t = \frac{||C||_2\sqrt{n}}{L\sqrt{d}} \cdot \frac{e^{\epsilon_\ell}-1}{e^{\epsilon_\ell}+1}$
7: **end for**
8: **return** $\theta_{priv} \leftarrow \theta_T$

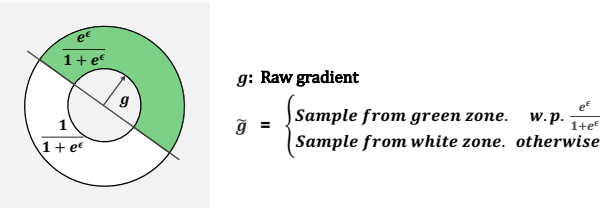


図 4: 勾配のランダム化のイメージ ([19] の図 4 を元に作成)

LDP-SGD には以下の 2 つの特徴がある．

- ランダム化前の勾配のノルムが小さいほど，ランダム化後の勾配の符号は逆転されやすい．
- プライバシパラメータ ϵ を大きく設定すると，ランダム化前の勾配と近い勾配が生成されやすい．

理論的には，LDP-SGD によって ϵ-LDP は保証されるとされているが，どういった攻撃に対してどの程度の強度があるのかは未知であり，ϵ の決定に必要な判断材料が不足している．

3 経験的なプライバシ強度の検査

本節では，LDP を適用した Federated Leaning の経験的なプライバシ強度を検査する方法について説明する．

この検査は以下の 3 つのエンティティで構成される．

***Crafter*:** グローバルモデル θ_t で 2 つの勾配 g_1, g_2 を生成する．どちらか 1 つを LDP-SGD(アルゴリズム 1) によってランダム化し $\tilde{g}$ とする．

***Model Trainer*:** アルゴリズム 2 で *Crafter* から受け取った $\tilde{g}$ を使ってグローバルモデルを更新し，θ_{t+1} とする．

***Distinguisher*:** *Crafter* が勾配の生成に使用したデータ x_1, x_2 を所持し，$\tilde{g}$ または θ_{t+1} からランダム化された勾配が g_1, g_2 のどちらであったかを予測する．本来であれば，*Model Trainer* にしか共有されない $\tilde{g}$ からランダム化された勾配を予測できる強い権限を持った *Distinguisher* を *Strong Distinguisher*，*Model Trainer* が更新したグローバルモデル θ_{t+1} からランダム化された勾配を予測できる *Distinguisher* を *Weak Distinguisher* とする．

Crafter が勾配を生成してから *Distinguisher* がランダム化された勾配を予測するまでを 1 回とした検査を十分な回数行う．2 つの勾配を判別可能であった確率を算出し，判別可能な確率が高くなるほどにランダム化手法の経験的なプライバシ強度が十分でなく，逆に判別可能な確率が低くなるほど経験的なプライバシ強度が高いことを示すことができる．

このような検査方法で経験的なプライバシ強度を測定可能になる理由を次項で説明する．

3.1 仮説検定としての局所差分プライバシ

計算機構 M の入力 x, x' と出力 y について，以下のような仮説検定を考える．

H_0 : 出力 y は入力 x から作られた．

H_1 : 出力 y は入力 x'から作られた．

棄却領域を S，S の補集合を $\bar{S}$ とする．帰無仮説 H_0 が実際には真であるのに棄却した割合 (以下 FP:False Positive Rate) は，$s \in S$ として $Pr(M(x) = s)$ と定義される．そして，帰無仮説 H_0 が実際には偽であるのに棄却されなかった割合 (以下 FN:False Negative Rate) は，$\bar{s} \in \bar{S}$ として $Pr(M(x') = \bar{s})$ と定義される．計算機構 M が ϵ-LDP を保証するとは，以下の条件を満たすと同等である [10].

定理 1 (経験的 ϵ-局所差分プライバシ)**.** $\epsilon \in \mathbb{R}^+$ について，計算機構 M は $\forall x, x' \in X$ および，任意の棄却領域 $S \subseteq Y$ に含まれる値 $s \in S$ に対して次の条件が満たされる場合にのみ，ϵ-LDP を満たす．

$$Pr(M(x) = s) + e^{\epsilon} Pr(M(x') = \bar{s}) \geqq 1 \tag{2}$$

$$e^{\epsilon} Pr(M(x) = s) + Pr(M(x') = \bar{s}) \geqq 1 \tag{3}$$

定理 1 を変形すると，経験的なプライバシ強度 $\epsilon_{empirical}$ は

$$\epsilon_{empirical} = \max \left(\log \frac{1 - FP}{FN}, \log \frac{1 - FN}{FP} \right) \tag{4}$$

と表せる．例えば 1000 回の試行で，実際には x から作られた出力 y を x' から作られたと予想した割合 (=FP) が 0.1，実際には x' から作られた出力 y を x から作られたと予想した割合 (=FN) が 0.2 だった場合，式 4 に FP，FN を代入すると $\epsilon_{empirical} \fallingdotseq 2.0$ となる．

4 関 連 研 究

本節では，CDP を保証した機械学習アルゴリズムである DP-SGD の経験的プライバシ強度を検査した研究 [8] [21] について述べる．

Jagielski ら [21] は Membership Inference [22] や 2 つの Poisoning Attack [23] [21] に対し，DP-SGD [9] でプライバシ保護した学習モデルがどの程度耐えうるのか分析を行った．この分析ではデータセット D, D' で学習を行ったとき，出力が設定した集合に含まれるかどうかで，出力から入力を判別できる確率を算出する．$\epsilon = 4$ を設定した場合，Poisoning Attack [21] では $\epsilon_{empirical} = 0.46$ で 8.7 倍のギャップがあり，理論値と経験的プライバシ強度が近づくようなワーストケースを見つけることが課題とされた．

Nasr ら [8] は Jagielski ら [21] の研究を発展させ，DP-SGD の $\epsilon_{empirical}$ を理論値に達する攻撃を示した．この攻撃では，ほとんどのパラメータが更新されないように加工されたデータセットを生成し，その勾配の集合 $\mathcal{G}$ または Watermark が入った勾配を足した $\mathcal{G}'$ のどちらかを学習に使用する．Watermark を入れた部分に注目することで学習データが $\mathcal{G}$ または $\mathcal{G}'$ かを予測する．実験では，$\epsilon = 1, 2, 4, 10$ のいずれに設定した場合でも $\epsilon_{empirical}$ が理論値と一致することを示した．

このような関連研究では，信頼できるデータ収集者 (*Model Trainer*) がいるものとした攻撃を想定している．しかしながら，LDP を適用した Federated Learning で想定される攻撃は，それらとは全く同じであるとは言い難い．データ収集者を信頼しない LDP では，悪意のあるデータ収集者からの攻撃も考えられる．さらに，Federated Learning ではアップデートしたパラメータからだけではなく，ランダム化した勾配そのものから元データを推測される可能性がある．DP-SGD を扱った関連研究 [8] [21] よりも多くの脅威を想定する点が本研究は関連研究と異なっている．

5 提案検査法と実験的評価

5.1 概　　要

本研究の検査ではアクセスレベルによって 4 種類の設定を定義する．

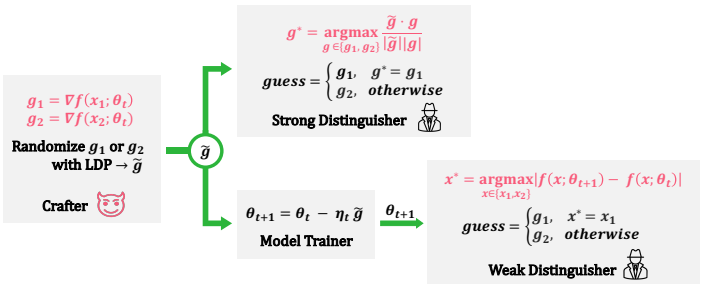
Benign Setting: どのエンティティも悪意のない場合．

Image Perturbation: *Crafter* が画像を加工できる場合．

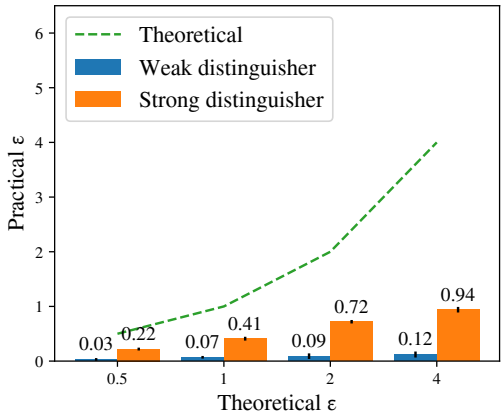
Gradient Flipping: *Crafter* がランダム化前の勾配を加工できる場合．

Gradient Flipping with Malicious Model: *Crafter* と *Model Trainer* が共謀する場合．

このような 4 つの設定で 10000×10 回の検査を行い，10000 回ごとに $\epsilon_{empirical}$ を計算し，10 回の平均を経験的なプライバシ強度とした．使用したデータセットは MNIST と CIFAR-10



(a) 概要



(b) 実験結果 (MNIST)

図 5: Benign Setting: *Crafter*・*Model Trainer*・*Distinguisher* のいずれも悪意のない設定．加工していない 2 つの画像から勾配を生成し判別する．

で，プライバシパラメータ ϵ の値は 0.5, 1, 2, 4 に設定した．

5.2 Benign Setting

Federated Learning における最も現実的な設定として，どのエンティティも悪意のないものであるとする．この設定の概要は図 5(a) に示した．プロトコルは以下のようになる．

（1） *Crafter* が *Model Trainer* から配布されたグローバルモデル θ_t を使って，2 つの画像 x_1, x_2 から勾配 g_1, g_2 を生成．どちらか 1 つをランダム化し $\tilde{g}$ として，*Model Trainer* に送信．

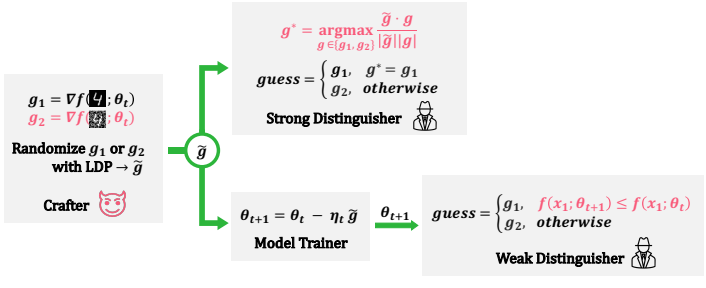
（2） *Model Trainer* が $\tilde{g}$ でグローバルモデルを更新し θ_{t+1} とする．

（3） *Strong Distinguisher* が $\tilde{g}$ とランダム化前の 2 つの勾配 g_1, g_2 とそれぞれコサイン類似度を計算し，類似度の高い勾配をランダム化された勾配と予測する．

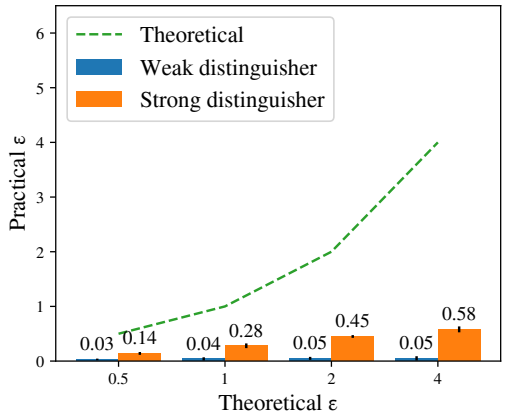
$$g^* = \underset{g \in \{g_1, g_2\}}{\arg\max} \frac{\tilde{g} \cdot g}{|\tilde{g}||g|}$$

$$guess = \begin{cases} g_1 & g^* = g_1 \\ g_2 & otherwise \end{cases}$$

（4） *Weak Distinguisher* が x_1, x_2 の損失 f を計算し，モデル更新前後の差分を比較することで更新に使用された勾配を予測する．



(a) 概要



(b) 実験結果 (MNIST)

図 6: Image Perturbation: *Crafter* が画像を加工できる設定．*Crafter* が画像に摂動を加える．

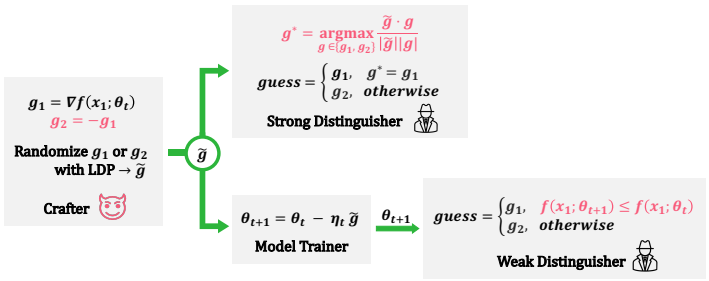
$$x^* = \underset{x \in \{x_1, x_2\}}{\arg\max} |f(x; \theta_{t+1}) - f(x; \theta_t)|$$

$$guess = \begin{cases} g_1 & x^* = x_1 \\ g_2 & otherwise \end{cases}$$

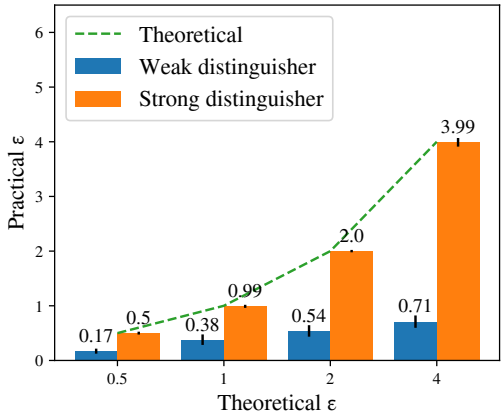
図 5(b) の実験結果は，縦軸が測定した ϵ(経験的なプライバシ強度)，横軸が設定した ϵ(理論的なプライバシ強度) である．最も現実的なこの設定では，$\epsilon = 4$ を設定しても経験的なプライバシ強度が最大でも 0.94 と理論値とのギャップが大きい．これは判別に成功した確率で表すと，68.1%程度となる．*Strong Distinguisher* は *Weak Distinguisher* よりも $\epsilon_{empirical}$ は高くなっており，ランダム化した勾配 $\tilde{g}$ を使って更新したパラメータ θ_{t+1} よりも，ランダム化した勾配 $\tilde{g}$ そのものを判別する方が容易であることが示された．

5.3 Image Perturbation

この設定では，*Crafter* によって元データが加工される場合を想定する．*Crafter* が所持しているデータ x_1 に摂動 (Perturbation) を加えることで，勾配を判別しやすくしようとする．摂動を加える手法は広く知られている，FGSM(Fast Gradient Sign Method) [24] を用いた．FGSM は，入力画像に対し入力画像に関する損失の勾配を使用して，その損失を最大化する新しい画像を作成する．有名な例として，摂動が加えられたパンダの画像がテナガザルとして分類される攻撃がある．この設定の概要は図 6(a) に示した．プロトコルは以下のようになる．

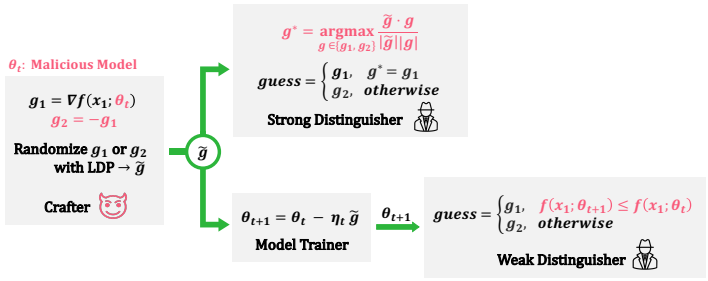


(a) 概要

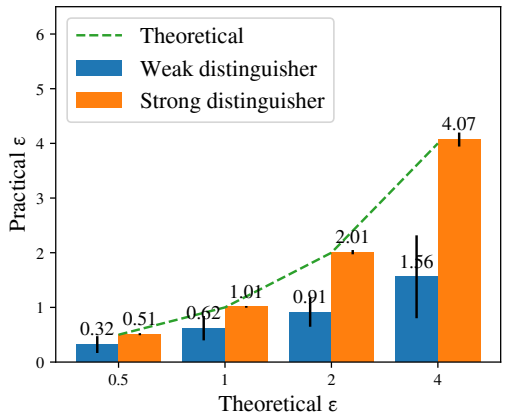


(b) 実験結果 (MNIST)

図 7: Gradient Flipping: *Crafter* がランダム化前の勾配を加工できる設定．*Crafter* がランダム化前の勾配を反転する．



(a) 概要



(b) 実験結果 (MNIST)

図 8: Gradient Flipping with Malicious Model:*Crafter* と *Model Trainer* が共謀する設定．

（1） *Crafter* がグローバルモデル θ_t を使って，x_1 と x_1 に摂動を加えた x_2 から勾配 g_1, g_2 を生成．どちらか 1 つをランダム化し $\tilde{g}$ として，*Model Trainer* に送信．

（2） *Model Trainer* が $\tilde{g}$ でグローバルモデルを更新し θ_{t+1} とする．

（3） *Strong Distinguisher* が $\tilde{g}$ とランダム化前の 2 つの勾配 g_1, g_2 とそれぞれコサイン類似度を計算し，類似度の高い勾配をランダム化された勾配と予測する．

（4） *Weak Distinguisher* がモデル更新後に x_1 の損失が増えていれば摂動が加えられた画像の勾配 g_2 がランダム化されたと予測する．

$$guess = \begin{cases} g_1 & f(x_1;\theta_{t+1}) \leqq f(x_1;\theta_t) \\ g_2 & otherwise \end{cases}$$

実験結果を表す図 6(b) から，加工された画像の勾配をランダム化しても判別可能な確率は上がらず，経験的なプライバシ強度と理論的なプライバシ強度のギャップは埋まっていないことが分かる．

5.4 Gradient Flipping

この設定では，*Crafter* によって勾配が加工される場合を想定する．判別可能性を高めるための勾配の加工として最も単純な方法としては，勾配の符号の反転が挙げられる [25]．符号が反転された勾配が学習に使われると，損失が大きくなりやすくなるためである．2.4 項にあるように，LDP-SGD では ϵ を大きく設定すると勾配の符号は変わりにくくなるため，このような加工が有効であると考える．この設定の概要は図 7(a) に示した．プロトコルは以下のようになる．

（1） *Crafter* がグローバルモデル θ_t を使って，画像 x_1 から勾配 g_1，g_1 の符号を反転させた g_2 を生成．どちらか 1 つをランダム化して $\tilde{g}$ として，*Model Trainer* に送信．

（2） *Model Trainer* が $\tilde{g}$ でグローバルモデルを更新し θ_{t+1} とする．

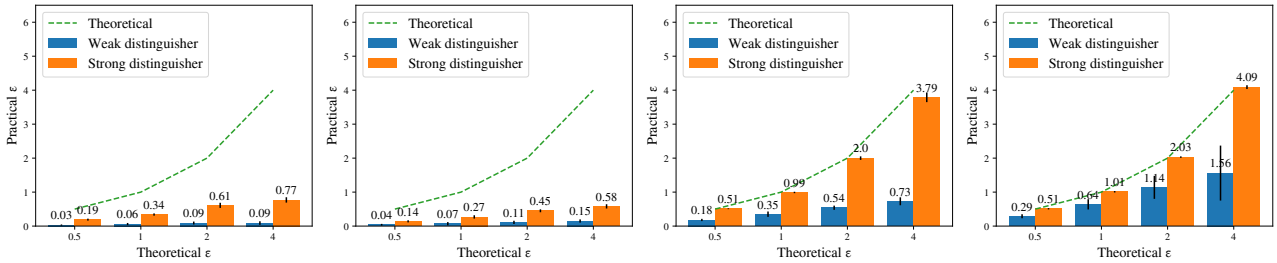
（3） *Strong Distinguisher* が $\tilde{g}$ とランダム化前の 2 つの勾配 g_1, g_2 とそれぞれコサイン類似度を計算し，類似度の高い勾配をランダム化された勾配と予測する．

（4） *Weak Distinguisher* がモデル更新後に x_1 の損失が増えていれば，符号を反転させた勾配 g_2 がランダム化されたと予測する．

図 7(b) は実験結果のグラフである．2 つの勾配のうち，1 つの勾配の符号を反転した場合，Benign Setting と比べて 2 つの勾配の判別可能な確率が向上することが分かる．特に，*Strong Distinguisher* では $\epsilon = 4$ を設定した場合に $\epsilon_{empirical}$ は 3.99 となり，ほとんど理論値となる．

5.5 Gradient Flipping with Malicious Model

2.4 項で説明したように，LDP-SGD ではランダム化前の勾配のノルムが小さいほど符号が反転されやすくなってしまう．この性質を利用して，*Crafter* と *Model Trainer* が共謀し，ノルムが小さい勾配が生成されにくくするような設定を考える．



(a) Benign Setting (b) Image Perturbation (c) Gradient Flipping (d) Gradient Flipping with MM

図 9: CIFAR-10 データセットで勾配をランダム化した場合の経験的なプライバシ強度

この設定では *Model Trainer* が損失が大きくなるグローバルモデルを *Crafter* に配布し，*Crafter* が Gradient Flipping と同じように勾配の反転を行う．この設定の概要は図 8(a) に示した．プロトコルは以下のようになる．

（1） *Model Trainer* が全て正解ラベルが同じ画像のみを使って，グローバルモデル θ_t を生成し，*Crafter* に配布．

（2） *Crafter* がグローバルモデル θ_t の生成に使われたラベルとは別のラベルを持つ画像 x_1 から勾配 g_1，g_1 の符号を反転させた g_2 を生成．どちらか 1 つをランダム化して $\tilde{g}$ として，*Model Trainer* に送信．

（3） *Model Trainer* が $\tilde{g}$ でグローバルモデルを更新し θ_{t+1} とする．

（4） *Strong Distinguisher* が $\tilde{g}$ とランダム化前の 2 つの勾配 g_1, g_2 とそれぞれコサイン類似度を計算し，類似度の高い勾配をランダム化された勾配と予測する．

（5） 5.4 項の Gradient Flipping と同様に，*Weak Distinguisher* がモデル更新後に x_1 の損失が増えていれば，符号を反転させた勾配 g_2 がランダム化されたと予測する．

図 8(b) の実験結果から，*Crafter* と *Model Trainer* が共謀することで，勾配の符号を反転させる Gradient Flipping よりも，理論的なプライバシ強度と経験的なプライバシ強度のギャップが小さくなる．この設定の *Strong Distinguisher* では，全ての設定値 $\epsilon = 0.5, 1, 2, 4$ で経験的なプライバシ強度が理論値に達した．この場合の $\epsilon_{empirical} = 4.07$ は，約 98.2%の確率で判別に成功した，と言い換えることができる．

CIFAR-10 で実験した結果は図 9 に示した．

6 議　　論

前節の実験結果から，経験的なプライバシ強度が理論値に達するためには，*Crafter* と *Model Trainer* が共謀し，*Crafter* が真逆の方向の勾配を生成して，さらにランダム化前の勾配を持つ *Distinguisher* がコサイン類似度を取ることが必要であると分かった．しかしながら，このような設定は現実的と言えるだろうか．例えば，Intel SGX [26] [27] のような高いレベルのセキュリティで保護された環境がサーバ内にあるのであれば，各クライアントによってランダム化された勾配を他のクライアントが見ることは不可能になり，経験的なプライバシ強度は理論値に達することは難しくなると推測される．すなわち，*Crafter*・*Model Trainer*・*Distinguisher* の設定にいくつか制約を加えることで，理論的には $\epsilon = 4$ で実装したシステムあっても $\epsilon_{empirical} = 1$ 程度の強度を達成できる，といった ϵ の緩和が可能になると考える．

7 結　　論

本研究では，LDP を適用した Federated Learning において，どういった攻撃に対してどの程度の強度があるのかを明らかにする検査を行った．その結果，現実的な設定のクライアントとサーバを想定した場合には理論的なプライバシ強度と経験的なプライバシ強度のギャップが大きく，逆にクライアントとサーバが共謀するような設定だとしても，LDP-SGD では理論値と同等のプライバシ強度が保たれていることを示した．

文　　献

[1] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[2] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[3] Hajime Ono and Tsubasa Takahashi. Locally private distributed reinforcement learning. *arXiv preprint arXiv:2001.11718*, 2020.

[4] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[5] Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984*, 2018.

[6] Ruixuan Liu, Yang Cao, Masatoshi Yoshikawa, and Hong Chen. Fedsel: Federated sgd under local differential privacy with top-k dimension selection. In *International Conference on Database Systems for Advanced Applications*, pages 485–501. Springer, 2020.

[7] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. Local and central differential privacy for robustness and privacy in federated learning. *arXiv preprint arXiv:2009.03561*, 2020.

[8] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papemoti, and Nicholas Carlin. Adversary instantiation: Lower bounds for differentially private machine learning.

In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 866–882. IEEE, 2021.

[9] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[10] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015.

[11] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pages 2021–2031. PMLR, 2020.

[12] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pages 634–643. PMLR, 2019.

[13] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948. PMLR, 2020.

[14] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[15] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, Pedro PB de Gusmão, and Nicholas D Lane. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.

[16] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

[17] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

[18] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346, 2021.

[19] John C Duchi, Michael I Jordan, and Martin J Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201, 2018.

[20] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *arXiv preprint arXiv:2001.03618*, 2020.

[21] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? *Advances in Neural Information Processing Systems*, 33:22205–22216, 2020.

[22] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[23] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[24] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[25] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE, 2019.

[26] Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R Savagaonkar. Innovative instructions and software model for isolated execution. *Hasp@ isca*, 10(1), 2013.

[27] Victor Costan and Srinivas Devadas. Intel sgx explained. *Cryptology ePrint Archive*, 2016.