fn cdp_delta

Michael Shoemate

October 27, 2025

Proves soundness of `fn cdp_delta` in `cdp_delta.rs` at commit 0b8f4222 (outdated[1]). This proof is an adaptation of subsection 2.3 of [CKS20].

1 Bound Derivation

Definition 1.1. (Privacy Loss Random Variable). Let $M : \mathcal{X}^n \to \mathbb{Y}$ be a randomized algorithm. Let $x, x' \in \mathcal{X}^n$ be neighboring inputs. Define $f : \mathcal{Y} \to \mathbb{R}$ by $f(y) = log\left(\frac{\mathbb{P}[M(x)=y]}{\mathbb{P}[M(x')=y]}\right)$. Let $Z = f(M(x))$, the privacy loss random variable, denoted $Z \leftarrow PrivLoss(M(x)||M(x'))$.

Lemma 1.2. [CKS20] Let $\epsilon, \delta \geq 0$. Let M: $\mathcal{X}^n \to \mathcal{Y}$ be a randomized algorithm. Then M satisfies (ϵ, δ)-differential privacy if and only if

$$\delta \geq \mathop{\mathbb{E}}_{Z \leftarrow PrivLoss(M(x)||M(x'))}[max(0, 1 - e^{\epsilon - Z})] \tag{1}$$

$$\tag{2}$$

for all $x, x' \in \mathcal{X}^n$ differing on a single element.

Proof. Fix neighboring inputs $x, x' \in \mathcal{X}^n$. Let $f : \mathcal{Y} \to \mathbb{R}$ be as in 1.1. For notational simplicity, let $Y = M(x)$, $Y' = M(x')$, $Z = f(Y)$ and $Z' = -f(Y')$. This is equivalent to $Z \leftarrow PrivLoss(M(x)||M(x'))$. Our first goal is to prove that

$$\sup_{E \subset \mathcal{Y}} \mathbb{P}[Y \in E] - e^{\epsilon}\mathbb{P}[Y' \in E] = \mathbb{E}[max\{0, 1 - e^{\epsilon - Z}\}]. \tag{3}$$

For any $E \subset \mathcal{Y}$, we have

$$\mathbb{P}[Y' \in E] = \mathbb{E}[\mathbb{I}[Y' \in E]] = \mathbb{E}[\mathbb{I}[Y \in E]e^{-f(Y)}]. \tag{4}$$

This is because $e^{-f(y)} = \frac{\mathbb{P}[Y=y]}{\mathbb{P}[Y'=y]}$.

Thus, for all $E \subset \mathcal{Y}$, we have

$$\mathbb{P}[Y \in E] - e^{\epsilon}\mathbb{P}[Y' \in E] = \mathbb{E}\left[\mathbb{I}[Y \in E](1 - e^{\epsilon - f(Y)})\right] \tag{5}$$

Now it is easy to identify the worst event as $E = \{y \in \mathcal{Y} : 1 - e^{\epsilon - f(y)} > 0\}$. Thus

$$\sup_{E \subset Y} \mathbb{P}[Y \in E] - e^{\epsilon}\mathbb{P}[Y' \in E] = \mathbb{E}\left[\mathbb{I}[1 - e^{\epsilon - f(Y)} > 0](1 - e^{\epsilon - f(Y)})\right] = \mathbb{E}[max\{0, 1 - e^{\epsilon - Z}\}] \tag{6}$$

□

[1] See new changes with `git diff 0b8f4222.. rust/src/combinators/measure_cast/zCDP_to_approxDP/cdp_delta/cdp_delta.rs`

Theorem 1.3. [CKS20] Let $M : \mathcal{X}^n \to \mathcal{Y}$ be a randomized algorithm. Let $\alpha \in (1, \infty)$ and $\epsilon \geq 0$. Suppose $D_\alpha(M(x)||M(x')) \leq \tau$ for all $x, x' \in \mathcal{X}^n$ differing in a single entry.[2] Then M is (ϵ, δ)-differentially private for

$$\delta = \frac{e^{(\alpha-1)(\tau-\epsilon)}}{\alpha - 1}\left(1 - \frac{1}{\alpha}\right)^{\alpha} \tag{7}$$

Proof. Fix neighboring $x, x' \in \mathcal{X}^n$ and let $Z \leftarrow PrivLoss(M(x)||M(x'))$. We have

$$\mathbb{E}[e^{(\alpha-1)Z}] = e^{(\alpha-1)D_\alpha(M(x)||M(x'))} \leq e^{(\alpha-1)\tau} \tag{8}$$

By 1.2, our goal is to prove that $\delta \geq \mathbb{E}[max\{0, 1 - e^{\epsilon - Z}\}]$. Our approach is to pick $c > 0$ such that $max\{0, 1 - e^{\epsilon - Z}\} \leq ce^{(\alpha-1)z}$ for all $z \in \mathbb{R}$. Then

$$\mathbb{E}[max\{0, 1 - e^{\epsilon - Z}\}] \leq \mathbb{E}[ce^{(\alpha-1)z}] \leq ce^{(\alpha-1)\tau}. \tag{9}$$

We identify the smallest possible value of c:

$$c = \sup_{z \in \mathbb{R}} \frac{max\{0, 1 - e^{\epsilon - z}\}}{e^{(\alpha-1)z}} = \sup_{z \in \mathbb{R}} e^{z - \alpha z} - e^{\epsilon - \alpha z} = \sup_{z \in \mathbb{R}} f(z) \tag{10}$$

where $f(z) = e^{z-\alpha z} - e^{\epsilon - \alpha z}$. We have

$$f'(z) = e^{z-\alpha z}(1-\alpha) - e^{\epsilon - \alpha z}(-\alpha) = e^{-\alpha z}(\alpha e^{\epsilon} - (\alpha - 1)e^{z}) \tag{11}$$

Clearly $f'(z) = 0 \iff e^z = \frac{\alpha}{\alpha-1}e^{\epsilon} \iff z = \epsilon - log(1 - 1/\alpha)$. Thus

$$\begin{aligned}
c &= f(\epsilon - log(1 - 1/\alpha)) && (12)\\
&= \left(\frac{\alpha}{\alpha-1}e^{\epsilon}\right)^{1-\alpha} - e^{\epsilon}\left(\frac{\alpha}{\alpha-1}e^{\epsilon}\right)^{-\alpha} && (13)\\
&= \left(\frac{\alpha}{\alpha-1}e^{\epsilon} - e^{\epsilon}\right)\left(\frac{\alpha}{\alpha-1}e^{-\epsilon}\right)^{\alpha} && (14)\\
&= \frac{e^{\epsilon}}{\alpha-1}\left(1 - \frac{1}{\alpha}\right)^{\alpha} e^{-\alpha\epsilon}. && (15)
\end{aligned}$$

Thus

$$\mathbb{E}[max\{0, 1 - e^{\epsilon - Z}\}] \leq \frac{e^{\epsilon}}{\alpha - 1}\left(1 - \frac{1}{\alpha}\right)^{\alpha} e^{-\alpha\epsilon}e^{(\alpha-1)\tau} = \frac{e^{(\alpha-1)(\tau-\epsilon)}}{\alpha-1}\left(1 - \frac{1}{\alpha}\right)^{\alpha} = \delta \tag{16}$$

□

Corollary 1. [CKS20] Let $M : \mathcal{X}^n \to \mathcal{Y}$ be a randomized algorithm. Let $\alpha \in (1, \infty)$ and $\epsilon \geq 0$. Suppose $D_\alpha(M(x)||M(x')) \leq \tau$ for all $x, x' \in \mathcal{X}^n$ differing in a single entry. Then M is (ϵ, δ)-differentially private for

$$\epsilon = \tau + \frac{\ln(1/\delta) + (\alpha - 1)\ln(1 - 1/\alpha) - \ln(\alpha)}{\alpha - 1} \tag{17}$$

Proof. This follows by rearranging 1.3. □

Corollary 2. Let $M : \mathcal{X}^n \to \mathcal{Y}$ be a randomized algorithm satisfying ρ-concentrated differential privacy. Then M is (ϵ, δ)-differentially private for any $0 < \delta \leq 1$ and

$$\epsilon = \inf_{\alpha \in (1,\infty)} \alpha\rho + \frac{\ln(1/\delta) + (\alpha - 1)\ln(1 - 1/\alpha) - \ln(\alpha)}{\alpha - 1} \tag{18}$$

Proof. This follows from 1 by taking the infimum over all divergence parameters α. □

[2] This is the definition of (α, τ)-Rényi differential privacy.

2 Pseudocode

Precondition

None.

Implementation

```python
def cdp_delta(rho: float, eps: float) -> float:
    """The Rust code may be easier to follow due to more commenting."""
    if rho.is_sign_negative():
        raise ValueError(f"rho ({rho}) must be non-negative")

    if eps.is_sign_negative():
        raise ValueError(f"epsilon ({eps}) must be non-negative")

    if rho.is_zero() or eps.is_infinite():
        return 0.0

    if rho.is_infinite():
        return 1.0

    a_max = eps.inf_add(1.0).inf_div((2.0).neg_inf_mul(rho)).inf_add(2.0)

    a_min = 1.01

    while True:
        diff = a_max - a_min

        a_mid = a_min + diff / 2.0

        if a_mid == a_max or a_mid == a_min:
            break

        # calculate derivative
        deriv = (2.0 * a_mid - 1.0) * rho - eps + a_mid.recip().neg().ln_1p()

        if deriv.is_sign_negative():
            a_min = a_mid
        else:
            a_max = a_mid

    # calculate delta
    a_1 = a_max.inf_sub(1.0)
    ar_e = a_max.inf_mul(rho).inf_sub(eps)

    try:
        t1 = a_1.inf_mul(ar_e)

    except OpenDPException:

        # if t1 is negative, then handle negative overflow by making t1 larger: the most negative finite float
        # making t1 larger makes delta larger, so it's still a valid upper bound
        if a_1.is_sign_negative() != ar_e.is_sign_negative():
            t1 = 1.7976931348623157e308  # f64::MIN
        else:
            raise

    t2 = a_max.inf_mul(a_max.recip().neg().inf_ln_1p())

    delta = t1.inf_add(t2).inf_exp().inf_div((a_max.inf_sub(1.0)))

```

```
    # delta is always <= 1
    delta.min(1.0)
```

Postcondition

Theorem 2.1. For any possible setting of ρ and ϵ, `cdp_delta` either returns an error, or a δ such that any ρ-differentially private measurement is also (ϵ, δ)-differentially private.

3 Proof

Proof. The code always finds an $\alpha_* \approx$ `a_max` ≥ 1.01. Since `a_max` $\in (1, \infty)$, then by 1, any ρ-differentially private measurement is also $(\epsilon(\texttt{a\_max}), \delta)$-differentially private. Define $\delta_{cons}(\alpha)$ as a "conservative" function for computing $\delta(\epsilon)$, where floating-point arithmetic is computed with conservative rounding such that $\delta_{cons}(\alpha) \geq \delta(\alpha)$ for $\forall \alpha \in (1, \infty)$. Since `delta` $= \delta_{cons}(\texttt{a\_max}) \geq \delta(\texttt{a\_max})$, then any $(\epsilon, \delta(\texttt{a\_max}))$-differentially private measurement is also $(\texttt{epsilon}, \delta)$-differentially private. □

References

[CKS20] Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *CoRR*, abs/2004.00010, 2020.