fn approximate_to_tradeoff

Aishwarya Ramasethu, Yu-Ju Ku, Jordan Awan, Michael Shoemate

July 4, 2025

This proof resides in **"contrib"** because it has not completed the vetting process.

1 Hoare Triple

Preconditions

Compiler-verified

- Argument `param` of type `(f64, f64)`

User-verified

None

Pseudocode

```python
def appproximate_to_tradeoff(
    param: tuple[f64, f64]
) -> tuple[Callable[[RBig], RBig], RBig]:
    epsilon, delta = param

    exp_eps = epsilon.with_rounding(Down).exp()  #
    exp_eps = RBig.try_from(exp_eps)

    exp_neg_eps = (-epsilon).with_rounding(Up).exp()  #
    exp_neg_eps = RBig.try_from(exp_neg_eps)

    fixed_point = (RBig(1) - delta) / (RBig(1) + exp_eps)

    if fixed_point >= RBig(1, 2):
        raise ValueError("fixed point of tradeoff curve must be less than 1/2")

    def tradeoff(alpha: RBig) -> RBig:  #
        t1 = RBig(1) - delta - exp_eps * alpha
        t2 = exp_neg_eps * (RBig(1) - delta - alpha)
        return max(max(t1, t2), RBig(0))

    return tradeoff, fixed_point
```

Postcondition

Theorem 1.1. Given a pair of epsilon and delta, the pseudocode returns an error if epsilon or delta are invalid, otherwise returns the corresponding symmetric nontrivial f-DP tradeoff curve with conservative arithmetic, as well as the fixed point c where $c = f(c)$.

Proof. We start with the following alternative definition of DP from Awan and Vadhan 2023.

Definition 1.2 (Awan and Vadhan 2023, Definition 2.2)**.** Let $\epsilon > 0$ and $\delta \geq 0$, and define

$$f_{\epsilon,\delta}(\alpha) = \max(0, 1 - \delta - \exp(\epsilon)\alpha, \exp(-\epsilon)(1 - \delta - \alpha)). \tag{1}$$

Then we say that a mechanism M satisfies (ϵ, δ)-DP if it satisfies $f_{\epsilon,\delta}$-DP.

The definition assumes $\alpha \in [0, 1]$. For more information about tradeoff functions, see Awan and Vadhan 2023.

On lines 6 and 9, arithmetic is computed in a manner which results in over-estimates of the privacy loss, and therefore a tradeoff curve that bows further away from $1-\alpha$. The results of these constants are converted exactly into rationals to be used in the tradeoff function. The function defined on line 17 implements the formula in Definition 1.2 with exact fractional arithmetic. The implementation of the tradeoff function in the pseudocode is the corresponding symmetric nontrivial f-DP tradeoff curve with conservative arithmetic, satisfying the postcondition.

We now show that the second return value of the pseudocode is the fixed point c of the tradeoff function. The fixed-point c of the tradeoff function is $(1 - \delta)/(1 + e^\epsilon)$, because $f_{\epsilon,\delta}(c) = c$:

$$f_{\epsilon,\delta}((1 - \delta)/(1 + e^\epsilon)) \tag{2}$$
$$= \max(0, 1 - \delta - \exp(\epsilon)((1 - \delta)/(1 + e^\epsilon)), \exp(-\epsilon)(1 - \delta - ((1 - \delta)/(1 + e^\epsilon)))) \tag{3}$$
$$= \max(0, (1 - \delta)/(1 + e^\epsilon), (1 - \delta)/(1 + e^\epsilon)) \tag{4}$$
$$= (1 - \delta)/(1 + e^\epsilon) \tag{5}$$

As shown, the tradeoff function, when invoked with the fixed point c, returns the fixed point c.
Therefore both return values satisfy their respective requirements specified in the postcondition. □

References

Awan, Jordan and Salil Vadhan (2023). "Canonical Noise Distributions and Private Hypothesis Tests". In: *The Annals of Statistics* 51.2, pp. 547–572.