impl TopKMeasure for MaxDivergence

Michael Shoemate

August 19, 2025

1 Hoare Triple

Precondition

Compiler-verified

- Method `noisy_top_k` *Types consistent with pseudocode.*
- Method `privacy_map` *Types consistent with pseudocode.*

Caller-verified

- Method `random_variable`
  - `x` elements are non-null.
  - `scale` is non-null and non-negative.
- Method `privacy_map`
  - `d_in` is non-null and positive.
  - `scale` is non-null and positive.

Pseudocode

```
class MaxDivergence(TopKMeasure):
    ONE_SHOT = False
    RV = ExponentialRV

    @staticmethod
    def random_variable(shift: FBig, scale: FBig) -> ExponentialRV:
        return ExponentialRV(shift=shift, scale=scale)

    @staticmethod
    def privacy_map(d_in: f64, scale: f64, k: usize) -> f64:
        if d_in < 0:
            raise ValueError("input distance must be non-negative")

        if scale.is_zero():
            return f64.INFINITY

        return d_in.inf_div(scale).inf_mul(f64.inf_cast(k))
```

Postcondition

Theorem 1.1. The implementation is consistent with all associated items in the `TopKMeasure` trait.

1. Method `random_variable`: Returns the index of the top element z_i, where each $z_i \sim \text{DISTRIBUTION}(\text{shift} = y_i, \text{scale} = \texttt{scale})$, and each $y_i = -x_i$ if `negate`, else $y_i = x_i$, k times with removal.

2. Method `privacy_map`: For any x, x' where $d_{\text{in}} \geq d_{\text{Range}}(x, x')$, return $d_{\text{out}} \geq D_{\text{self}}(f(x), f(x'))$, where $f(x) = \text{noisy\_top\_k}(x = x, k = 1, \text{scale} = \text{scale})$.

Definition 1.2. A random variable follows the Exponential distribution if it has density

$$f(x) = \frac{1}{\beta} e^{-z} \tag{1}$$

where $z = \frac{x-\mu}{\beta}$, μ is the shift (location) parameter and β is the scale parameter.

Proof of valid associated type: `RV`. The associated type `RV` is defined as `ExponentialRV`, which represents a random variable following the Exponential distribution 1.2. The compiler verifies that `ExponentialRV` implements the `InverseCDF` trait. □

Proof of postcondition: `random_variable`. The preconditions of `exponential_noisy_max` are met, therefore by the postcondition of `exponential_top_k`, the postcondition of `random_variable` is satisfied. □

Proof of postcondition: `privacy_map`. By Theorem 1 of [1], $\mathcal{M}_{\text{Exponential}}(x)$ satisfies ϵ-DP. □

References

[1] Ryan McKenna and Daniel Sheldon. Permute-and-flip: A new mechanism for differentially private selection, 2020.