fn make_scalar_integer_laplace

Michael Shoemate

October 25, 2024

Proves soundness of make_scalar_integer_laplace in mod.rs at commit f5bb719 (outdated[1]). The function on the resulting measurement takes in a data set x (a single integer), and returns a sample from the Discrete Laplace Distribution centered at x, with a fixed noise scale.

1 Hoare Triple

Preconditions

Compiler-verified

- Argument input_domain, of type AtomDomain<T>
- Argument input_metric, of type AbsoluteDistance<T>
- Argument scale, of type f64
- Generic T must have trait Integer and support saturating cast from IBig (for postprocessing a noisy big integer back to T)
- IBig must be constructable from T (to convert the data into a big integer)

Human-verified

None

Pseudocode

```
def make_scalar_integer_laplace(
    input_domain: AtomDomain[T], input_metric: AbsoluteDistance[T], scale: f64
):
    if scale.is_sign_negative():
        raise ValueError("scale must not be negative")

    # conversion to rational will fail if scale is null
    r_scale = RBig.try_from(scale)

    if scale == 0.0:
        def function(x: T):
            return x
    else:
        def function(x: T):
            release = IBig.from_(x) + sample_discrete_laplace(r_scale)
            # postprocessing
            return T.saturating_cast(release)
```

[1] See new changes with git diff f5bb719..77ce8ef4 rust/src/measurements/laplace/integer/mod.rs

```

    return Measurement(
        input_domain=input_domain,
        function=function,
        input_metric=input_metric,
        output_measure=MaxDivergence(),
        privacy_map=laplace_puredp_map(scale, relaxation=0.0),
    )
```

Postcondition

Theorem 1.1. For every setting of the input parameters `(input_domain, input_metric, scale, T)` to `make_scalar_integer_laplace` such that the given preconditions hold, `make_scalar_integer_laplace` raises an exception (at compile time or run time) or returns a valid measurement. A valid measurement has the following property:

1. (Privacy guarantee). For every pair of elements x, x' in `input_domain` and for every pair (`d_in`, `d_out`), where `d_in` has the associated type for `input_metric` and `d_out` has the associated type for `output_measure`, if x, x' are `d_in`-close under `input_metric`, `privacy_map`(`d_in`) does not raise an exception, and `privacy_map`(`d_in`) $\leq$ `d_out`, then `function`(x), `function`(x') are `d_out`-close under `output_measure`.

2 Proof

Proof. **(Privacy guarantee.)**

`sample_integer_laplace` can only fail due to lack of system entropy. This is usually related to the computer's physical environment and not the dataset. The rest of this proof is conditioned on the assumption that this function does not raise an exception.

Let x and x' be datasets that are `d_in`-close with respect to `input_metric`. Here, the metric is `AbsoluteDistance<T>`.

By the postcondition of `sample_integer_laplace`, the output of the function follows the Discrete Laplace Distribution with scale `scale`.

$$
\begin{aligned}
&\max_{x \sim x'} D_\infty(M(x), M(x')) \\
=&\max_{x \sim x'} \max_{z \in supp(M(\cdot))} \ln\left(\frac{\Pr[M(x) = z]}{\Pr[M(x') = z]}\right) && \text{substitute } D_\infty \\
=&\max_{x \sim x'} \max_{z \in \mathbb{Z}} \ln\left(\frac{\Pr[\mathrm{DLap}(x, b) = z]}{\Pr[\mathrm{DLap}(x', b) = z]}\right) && \text{where } b \text{ is the noise scale} \\
=&\max_{x \sim x'} \max_{z \in \mathbb{Z}} \ln\left(\frac{\frac{\exp^{1/b} - 1}{\exp^{1/b} + 1} \exp\left(-\frac{|x - z|}{b}\right)}{\frac{\exp^{1/b} - 1}{\exp^{1/b} + 1} \exp\left(-\frac{|x' - z|}{b}\right)}\right) && \text{use PMF of Discrete Laplace} \\
=&\max_{x \sim x'} \max_{z \in \mathbb{Z}} \ln\left(\frac{\exp\left(-\frac{|x - z|}{b}\right)}{\exp\left(-\frac{|x' - z|}{b}\right)}\right) \\
=&\max_{x \sim x'} \max_{z \in \mathbb{Z}} \frac{|x' - z| - |x - z|}{b} && \text{exp and ln cancel} \\
\leq&\frac{\max_{x \sim x'} |x - x'|}{b} && \text{by reverse triangle inequality} \\
=&\frac{d_{in}}{b} && \text{by definition of absolute distance}
\end{aligned}
$$

This bound satisfies the postcondition of `laplace_puredp_map`. The saturating conversion to `T` is a post-processing step.

Therefore it has been shown that for every pair of elements $x, x' \in$ `input_domain` and every $d_{Abs}(x, x') \leq$ `d_in` with `d_in` ≥ 0, if x, x' are `d_in`-close then `function`(x), `function`(x') are `privacy_map(d_in)`-close under `output_measure` (the Max-Divergence). $\square$