`NoisePrivacyMap<L1Distance<RBig>, MaxDivergence>` for `ZExpFamily<1>`

Michael Shoemate

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

Proves soundness of the implementation of `NoisePrivacyMap` for `ZExpFamily<1>` in `mod.rs` at commit f5bb719 (outdated[1]).

1 Hoare Triple

Precondition

Compiler-Verified

`NoisePrivacyMap` is parameterized as follows:

- `MI`, the input metric, is of type `L1Distance<RBig>`
- `MO`, the output measure, is of type `MaxDivergence`

User-Verified

None

Pseudocode

```python
# analogous to impl NoisePrivacyMap<L1Distance<RBig>, MaxDivergence> for ZExpFamily<1> in
    Rust
class ZExpFamily1:
    def noise_privacy_map(self) -> PrivacyMap[L1Distance[RBig], MaxDivergence]:
        scale = self.scale
        if scale < RBig.ZERO: #
            raise "scale must not non-negative"

        def privacy_map(d_in: RBig):
            if d_in < RBig.ZERO: #
                raise "sensitivity must be non-negative"

            if d_in.is_zero(): #
                return 0.0

            if scale.is_zero(): #
                return float('inf')

            return f64.inf_cast(d_in / scale) #

        return PrivacyMap.new_fallible(privacy_map)
```

[1] See new changes with `git diff f5bb719..fa860379 rust/src/measurements/noise/distribution/laplace/mod.rs`

Postcondition

Theorem 1.1. Given a distribution `self`, returns `Err(e)` if `self` is not a valid distribution. Otherwise if the output is `Ok(privacy_map)` then `privacy_map` observes the following:

Under the condition that:

- `input_metric` is `MI::default()`
- `output_metric` is `MO::default()`
- `function(x) = x + Z` where `Z` is a vector of iid samples from `self`

For every pair of elements x, x' in `VectorDomain<AtomDomain<IBig>>`, 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`.

Proof. Line 5 rejects `self` if `self` does not represent a valid distribution, satisfying the error conditions of the postcondition.

We now construct the privacy map. First consider the extreme values of the scale and sensitivity parameters. The sensitivity `d_in`, a bound on distances, must not be negative, as checked on line 9. In the case where sensitivity is zero (line 12), the privacy loss is zero, regardless the choice of scale parameter (even zero). This is because the privacy loss when adjacent datasets are always identical is zero. Otherwise, in the case where the scale is zero, the privacy loss is infinite. To avoid a rational division overflow, line 15 returns infinity.

By line 18, both the sensitivity and scale are positive rationals. We directly compute the max divergence over all x, x' in the input domain of big-integer vectors.

$$
\begin{aligned}
&\max_{x \sim x'} D_\infty(M(x), M(x')) && && (1)\\
&= \max_{x \sim x'} \max_{S \subseteq \text{supp}(M(\cdot))} \left[\ln \frac{\Pr[M(x) \in S]}{\Pr[M(x') \in S]} \right] && \text{substitute MaxDivergence} && (2)\\
&\leq \max_{x \sim x'} \max_{y \in \mathbb{Z}^d} \left[\ln \frac{\Pr[M(x) = y]}{\Pr[M(x') = y]} \right] && \text{mass is upper-bounded by point densities} && (3)\\
&= \max_{x \sim x'} \max_{y \in \mathbb{Z}^d} \ln \frac{\prod_{i=1}^d \frac{e^{-1/s}-1}{e^{-1/s}+1} e^{-|x_i - z_i|/s}}{\prod_{i=1}^d \frac{e^{-1/s}-1}{e^{-1/s}+1} e^{-|x'_i - z_i|/s}} && \text{substitute pdf} && (4)\\
&= \max_{x \sim x'} \max_{y \in \mathbb{Z}^d} \ln \prod_{i=1}^d e^{\frac{|x'_i - z_i| - |x_i - z_i|}{s}} && \text{cancel constants} && (5)\\
&= \frac{1}{s} \max_{x \sim x'} \max_{y \in \mathbb{Z}^d} \sum_{i=1}^d |x'_i - z_i| - |x_i - z_i| && \text{simplify via log rules} && (6)\\
&\leq \frac{1}{s} \max_{x \sim x'} \sum_{i=1}^d |x'_i - x_i| && d \text{ applications of reverse triangle inequality} && (7)\\
&= \frac{1}{s} \max_{x \sim x'} ||x' - x||_1 && \text{substitute } L_1 from \text{LpDistance} && (8)\\
&\leq \frac{\texttt{d\_in}}{s} && \text{since } ||x' - x||_1 = d_{L_1}(x, x') \leq \texttt{d\_in} && (9)\\
& && && (10)
\end{aligned}
$$

Line 18 implements this bound with exact division and conservative cast to float. ☐