fn make_vector_float_laplace_cks20

Michael Shoemate

August 18, 2024

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

Proves soundness of `make_vector_float_laplace_cks20` in `mod.rs` at commit f5bb719 (outdated[1]). The function on the resulting measurement takes in a data set `x` (a vector of floats), and returns a sample from the Vector Discrete Laplace Distribution centered at `x`, with a fixed noise scale.

PR History

- Pull Request #490

1 Hoare Triple

Preconditions

- Variable `input_domain`, of type `VectorDomain<AtomDomain<T>>`
- Variable `input_metric`, of type `L1Distance<T>`
- Variable `scale`, of type `QO`
- Variable `k`, of type `Option<i32>`
- Type `T` must have trait `Float` and `CastInternalRational`
- Type `i32` must be constructable from the bit representation of `T` (used to calibrate relaxation)

Pseudocode

```python
def make_vector_float_laplace_cks20(input_domain, input_metric, scale: QO, k):
    if scale < 0:
        raise ValueError("scale must not be negative")

    k, relaxation = get_discretization_consts(k)

    # each value in the input is rounded
    if relaxation != 0:
        if input_domain.size is None:
            raise ValueError("domain size must be known if discretization is not exact")
        relaxation = relaxation.inf_mul(T.inf_cast(input_domain.size))

    if scale == 0.:
        def function(x: list[T]):
```

[1] See new changes with `git diff f5bb719..5ab4bc95 rust/src/measurements/laplace/float/mod.rs`

```python
            return x
    else:
        def function(x: list[T]):
            return [sample_discrete_laplace_Z2k(x_i, scale, k) for x_i in x]

    return Measurement(
        input_domain=input_domain,
        function=function,
        input_metric=input_metric,
        output_measure=MaxDivergence(QO),
        privacy_map=laplace_map(scale, relaxation=relaxation)
    )
```

Postcondition

Theorem 1.1. For every setting of the input parameters `(input_domain, input_metric, scale, k, T)` to
`make_vector_float_laplace_cks20` such that the given preconditions hold,
`make_vector_float_laplace_cks20` 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.)**

The proof assumes the following lemma.

Lemma 2.1. `get_discretization_consts`, `sample_discrete_laplace_Z2k` and `laplace_map` each satisfy their postcondition.

This mechanism can be thought of as a stable transformation from a vector of floating-point numbers to a vector of rationals, followed by a mechanism that adds noise to each rational number.

2.1 Rounding Transformation

The transformation rounds the input to the nearest rational number where the denominator is no greater than 2^k. By the postcondition of `get_discretization_consts`, this rounding changes each value by at most `relaxation`.

$$
\begin{aligned}
&\max_{x\sim x'} d_{L1}(round(x), round(x')) \\
=&\max_{x\sim x'} \sum_{i=1}^{d} |round(x_i) - round(x'_i)| \\
\leq&\max_{x\sim x'} \sum_{i=1}^{d} (|x_i - x'_i| + \texttt{relaxation}) && \text{by postcondition of } \texttt{get\_discretization\_consts} \\
\leq&\texttt{d\_in} + d \cdot \texttt{relaxation} && \text{by definition of } L_1 \text{ distance}
\end{aligned}
$$

Therefore the rounding transformation is $\texttt{d\_in} + d \cdot \texttt{relaxation}$-close under the L^1 distance. The pseudocode multiplies the relaxation term by a factor of d, the dimension of the input vector, when the relaxation term is non-zero.

2.2 Noise Measurement

`sample_discrete_laplace_Z2k` 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 `L1Distance<T>`. By the postcondition of `sample_discrete_laplace_Z2k`, the output of the function follows the Discrete Laplace Distribution with scale `scale`.

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

Therefore it has been shown that for every pair of elements $x, x' \in$ `input_domain` and every $d_{L1}(x, x') \leq$ `d_in` with $\texttt{d\_in} \geq 0$, if x, x' are `d_in`-close then $\texttt{function}(x), \texttt{function}(x')$ are `privacy_map(d_in)`-close under `output_measure` (the Max-Divergence).

2.3 Chained Measurement

The chained map provides a guarantee that the output distributions are at most $\frac{d_{in} + d \cdot \texttt{relaxation}}{b}$-close under the max-divergence. This is consistent with the postcondition of `laplace_map`.

Therefore it has been shown that for every pair of elements $x, x' \in$ `input_domain` and every $d_{L1}(x, x') \leq$ `d_in` with $\texttt{d\_in} \geq 0$, if x, x' are `d_in`-close then $\texttt{function}(x), \texttt{function}(x')$ are `privacy_map(d_in)`-close under `output_measure` (the Max-Divergence).

□