fn get_rounding_distance

Michael Shoemate

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

Proves soundness of the implementation of `get_rounding_distance` in `mod.rs` at commit f5bb719 (outdated[1]).

1 Hoare Triple

Precondition

Compiler-Verified

- Generic `T` implements trait `Float`
- Type `i32` implements the trait `ExactIntCast<T.Bits>`, where `T.Bits` is the type of the native bit representation of `T`.

User-Verified

None

Pseudocode

```python
def get_rounding_distance() -> RBig:
    k_min = get_min_k(T) #
    if k < k_min: #
        raise f"k ({k}) must not be smaller than {k_min}"

    # input has granularity 2^{k_min} (subnormal float precision)
    input_gran = x_mul_2k(RBig.ONE, k_min) #

    # discretization rounds to the nearest 2^k
    output_gran = x_mul_2k(RBig.ONE, k) #

    # the worst-case increase in sensitivity due to discretization is
    #     the range, minus the smallest step in the range
    distance = output_gran - input_gran #

    # rounding may occur on all vector elements
    if not distance.is_zero(): #
        if size is None: #
            raise "domain size must be known if discretization is not exact"

        match P:
            case 1:
                distance *= RBig.from_(size)
```

[1] See new changes with `git diff f5bb719..30d1ed6 rust/src/measurements/noise/nature/float/utilities/mod.rs`

```
24             case 2:
25                 distance *= RBig.try_from(f64.inf_cast(size).inf_sqrt())
26             case _:
27                 raise f"norm ({P}) must be one or two"
28 
29     return distance
```

Postcondition

Theorem 1.1. Let D denote the space of `size`-dimensional vectors whose elements are in $\mathbb{Z}2^{k_{min}}$, where $2^{k_{min}}$ is the smallest distance between adjacent non-equal values in `T`. Let round_k be a function that rounds each element to the nearest multiple of 2^k, with ties rounding down. Return $\max_{x,x' \in D} ||\text{round}_k(x) - \text{round}_k(x')||_P - ||x - x'||_P$, the increase in the sensitivity due to rounding.

Proof. We first consider the increase in sensitivity due to rounding one element. The greatest increase in sensitivity occurs when one value x is rounded down, and x' rounded up. The greatest round down occurs when x is offset from the output grid by 2^{k-1}, and the greatest round up occurs when x' is offset from the output grid by $2^{k-1} + 2^{k_{min}}$.

$$
\begin{aligned}
&\max_{x,x'} |\text{round}_k(x) - \text{round}_k(x')| && (1)\\
&\leq \max_{x,x'} |(x - 2^{k-1}) - (x' + 2^{k-1} - 2^{k_{min}})| && (2)\\
&= \max_{x,x'} |(x - x') - 2 \cdot 2^{k-1} + 2^{k_{min}}| && (3)\\
&= \max_{x,x'} |x - x'| + 2^k - 2^{k_{min}} && (4)\\
& && (5)
\end{aligned}
$$

Now apply this same logic to a vector of rounded values.

$$
\begin{aligned}
&\max_{x,x'} ||\text{round}_k(x) - \text{round}_k(x')||_P && && (6)\\
&\leq \max_{x,x'} ||x - x' + r||_P && \text{where } r \text{ is a vector of } 2^k - 2^{k_{min}} && (7)\\
&\leq \max_{x,x'} ||x, x'||_P + ||r||_P && \text{triangle inequality} && (8)\\
&= \max_{x,x'} ||x, x'||_P + n^{1/P} \cdot (2^k - 2^{k_{min}}) && && (9)\\
& && && (10)
\end{aligned}
$$

Substituting into the return criteria of the postcondition, the return value is:

$$
\begin{aligned}
&\max_{x,x'} ||\text{round}_k(x) - \text{round}_k(x')||_P - ||x - x'||_p && (11)\\
&\leq \max_{x,x'} ||x, x'||_P + n^{1/P} \cdot (2^k - 2^{k_{min}}) + \max_{x,x'} ||x, x'||_P && (12)\\
&= n^{1/P} \cdot (2^k - 2^{k_{min}}) && (13)
\end{aligned}
$$

We now focus on showing correctness of the implementation. By the postcondition of `get_min_k` on line 2, `min_k` is the k such that the smallest distance between adjacent non-equal values of type `T` is 2^k (the distance between subnormals).

Line 3 ensures that k is not too small, as any smaller k would result in unused precision in the noise sample due to the output being rounded to the nearest `T`. This check is not necessary for privacy; it prevents wasted performance.

Since the precondition for x_mul_2k that $k \neq$ i32.MIN is satisfied on line 3, then by the postcondition of x_mul_2k on line 7, input_gran is the distance between subnormals, 2^k. Similarly, output_gran on line 10 is the distance between adjacent values in the rounded space.

The greatest possible increase in distances between rounded values is thus $2^k - 2^{k_{min}}$, as defined on line 14.

When $k = k_{min}$, the rounding is a no-op and $2^k - 2^{k_{min}}$ is zero, so line 17 skips the vector calculations. Otherwise line 18 ensures the vector size n is known, and the following lines increase the distance by a factor of $n^{1/P}$, resulting in a conservative upper estimate of the expected bound. $\square$