fn make_vec

Michael Shoemate

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

Proves soundness of the implementation of `make_vec` in `mod.rs` at commit f5bb719 (outdated[1]).

This transformation simply wraps an input scalar in a singleton vec. The output metric then becomes an Lp distance.

1 Hoare Triple

Precondition

Compiler-Verified

- Generic `T` implements trait `Number`
- Generic `Q` implements trait `Number`

User-Verified

None

Pseudocode

```python
def make_vec(
    input_space: tuple[AtomDomain[T], AbsoluteDistance[Q]],
) -> Transformation[
    AtomDomain[T], VectorDomain[AtomDomain[T]], AbsoluteDistance[Q], LpDistance[P, Q]
]:
    input_domain, input_metric = input_space
    return Transformation.new(
        input_domain,
        VectorDomain.new(input_domain).with_size(1),
        lambda arg: [arg],
        input_metric,
        LpDistance.default(),
        lambda d_in: d_in,
    )
```

Postcondition

Theorem 1.1.

Theorem 1.2. For every setting of the input parameters (`input_space, T, Q`) to `make_vec` such that the given preconditions hold, `make_vec` raises an exception (at compile time or run time) or returns a valid transformation. A valid transformation has the following properties:

[1] See new changes with `git diff f5bb719..bae48343 rust/src/transformations/scalar_to_vector/mod.rs`

1. (Appropriate output domain). For every element x in `input_domain`, `function`(x) is in `output_domain` or raises a data-independent runtime exception.
2. (Stability 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_metric`, if x, x' are `d_in`-close under `input_metric`, `stability_map(d_in)` does not raise an exception, and `stability_map(d_in)` $\leq$ `d_out`, then `function`(x), `function`(x') are `d_out`-close under `output_metric`.

Proof. The function is infallible, and the output domain trivially follows, since all output vectors are length-one. For all x in the input domain, the output of `make_vec` is a vector of length 1, so the output domain is trivially valid. The function is 1-stable because:

$$\begin{aligned}
&\texttt{d_out} && (1)\\
&= \max_{x \sim x'} d_{Lp}(f(x), f(x')) && (2)\\
&= \max_{x \sim x'} (\sum_i (x_i - x'_i)^p)^{1/p} && (3)\\
&= \max_{x \sim x'} ((x_1 - x'_1)^p)^{1/p} && (4)\\
&= \max_{x \sim x'} |x_1 - x'_1| && (5)\\
&= \max_{x \sim x'} d_{Abs}(x_1, x'_1) && (6)\\
&= 1 \cdot \texttt{d_in} && (7)\\
& && (8)
\end{aligned}$$

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_metric`, if x, x' are `d_in`-close under `input_metric`, `stability_map(d_in)` does not raise an exception, and `stability_map(d_in)` $\leq$ `d_out`, then `function`(x), `function`(x') are `d_out`-close under `output_metric`. □