fn make_sized_quantile_score_candidates

Michael Shoemate

April 12, 2023

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

Vetting History

- Pull Request #456

Proves soundness of `make_sized_quantile_score_candidates` in `mod.rs` at commit f5bb719 (outdated[1]). `make_sized_quantile_score_candidates` returns a Transformation that takes a numeric vector database and a vector of numeric quantile candidates, and returns a vector of scores, where higher scores correspond to more accurate candidates.

It is the same as the unsized version, `make_quantile_score_candidates`, but with a sized input domain, and different stability map. The unsized version has a more thorough explanation of the utility function.

1 Hoare Triple

Precondition

- `TIA` (input atom type) is a type with trait `Number`.
- `TOA` (output atom type) is a type with trait `Float`.

Function

```
def make_sized_quantile_score_candidates(size: usize, candidates: List[TIA], alpha: TOA):
    for i in range(len(candidates) - 1):
        assert candidates[i] < candidates[i + 1]

    alpha_num, alpha_den = alpha.into_frac(size=size)
    if alpha_num > alpha_den or alpha_den == 0:
        raise ValueError("alpha must be within [0, 1]")

    # ensures that the function will not overflow
    size * alpha_den

    def function(arg: List[TIA]):
        return score(arg, candidates, alpha_num, alpha_den, size)

    def stability_map(d_in: IntDistance):
        return TOA.inf_cast(d_in // 2).inf_mul(4).inf_mul(alpha_den)

    return Transformation(
```

[1] See new changes with `git diff f5bb719..5fe96270 rust/src/transformations/quantile_score_candidates/mod.rs`

```
        input_domain=SizedDomain(VectorDomain(AtomDomain(TIA)), size),
        output_domain=VectorDomain(AtomDomain(TOA)),
        function=function,
        input_metric=SymmetricDistance(),
        output_metric=InfDifferenceDistance(),
        stability_map=stability_map,
    )
```

Postcondition

For every setting of the input parameters (`size`, `candidates`, `alpha`) to `make_sized_quantile_score_candidates` such that the given preconditions hold, `make_sized_quantile_score_candidates` raises an exception (at compile time or run time) or returns a valid transformation. A valid transformation has the following properties:

1. (Appropriate output domain). For every element v in `input_domain`, `function`(v) is in `output_domain` or raises a data-independent runtime exception.

2. (Domain-metric compatibility). The domain `input_domain` matches one of the possible domains listed in the definition of `input_metric`, and likewise `output_domain` matches one of the possible domains listed in the definition of `output_metric`.

3. (Stability guarantee). For every pair of elements u, v 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 u, v are `d_in`-close under `input_metric` and `stability_map(d_in)` $\leq$ `d_out`, then `function`(u), `function`(v) are `d_out`-close under `output_metric`.

2 Proof

2.1 Appropriate Output Domain

The raw type and domain are equivalent, save for potential nullity in the atomic type. The scalar scorer structurally cannot emit null, because the input argument is non-null.

2.2 Domain-metric compatibility

On the input side, `SymmetricDistance` is well-defined on `VectorDomain`. On the output side, `InfDifferenceDistance` is well-defined on `VectorDomain` consisting of numeric elements.

2.3 Stability Guarantee

Lemma 2.1. If $d_{CO}(X, X') \leq 1$, then $d_\infty(\texttt{function}(X), \texttt{function}(X')) \leq 2$.

Proof. Assume $d_{CO}(X, X') \leq 1$. For convenience, let $s = \texttt{function}(X)$.

$$\begin{aligned}
d_\infty(s_i, s'_i) &= \max_i |s_i - s'_i| && \text{by definition of } d_\infty \\
&= \max_i |\text{abs_diff}(\alpha_{den} \cdot \min(\#(X < C_i), l), \alpha_{num} \cdot \min(|X| - \#(X = C_i), l)) && \text{by definition of } \texttt{function} \\
&\qquad \text{abs_diff}(\alpha_{den} \cdot \min(\#(X' < C_i), l), \alpha_{num} \cdot \min(|X'| - \#(X' = C_i), l))| \\
&= \alpha_{den} \cdot \max_i || \min(\#(X < C_i), l) - \alpha \cdot \min(|X| - \#(X = C_i), l)| \\
&\qquad | \min(\#(X' < C_i), l) - \alpha \cdot \min(|X'| - \#(X' = C_i), l)|| \\
&= \alpha_{den} \cdot \max_i ||\#(X < C_i) - \alpha \cdot (|X| - \#(X = C_i))| \\
&\qquad |\#(X' < C_i) - \alpha \cdot (|X| - \#(X' = C_i))||
\end{aligned}$$

Consider each of the four cases of changing a row in X.

Case 1. Assume X' is equal to X, but with some $X_j < C_i$ replaced with $X'_j > C_i$.

$$\begin{aligned}
&= 2 \cdot \alpha_{den} \cdot \max_i ||(1-\alpha) \cdot \#(X < C_i) - \alpha \cdot \#(X > C_i)| \\
&\qquad - (1-\alpha) \cdot (\#(X < C_i) - 1) - \alpha \cdot (\#(X > C_i) + 1)|| && \text{by definition of } \texttt{function} \\
&\le 2 \cdot \alpha_{den} \cdot \max_i ||(1-\alpha) \cdot \#(X < C_i) - \alpha \cdot \#(X > C_i)| \\
&\qquad - (|(1-\alpha) \cdot \#(X < C_i) - \alpha \cdot \#(X > C_i)| + |1|)| && \text{by triangle inequality} \\
&= 2 \cdot \alpha_{den} \cdot \max_i |1| && \text{scores cancel} \\
&= 2 \cdot \alpha_{den}
\end{aligned}$$

Case 2. Assume X' is equal to X, but with some $X_j > C_i$ replaced with $X'_j < C_i$.

$$= 2 \cdot \alpha_{den} \qquad \text{by symmetry, follows from Case 1.}$$

Case 3. Assume X' is equal to X, but with some $X_j \neq C_i$ replaced with C_i.

$$\le 2 \cdot \max(\alpha_{num}, \alpha_{den} - \alpha_{num}) \qquad \text{equivalent to one removal (see make}$$

Case 4. Assume X' is equal to X, but with some $X_j = C_i$ replaced with $X'_j \neq C_i$.

$$\le 2 \cdot \max(\alpha_{num}, \alpha_{den} - \alpha_{num}) \qquad \text{equivalent to one addition (see mak}$$

Take the union bound over all cases.

$$d_\infty(s_i, s'_i) \le \max(2 \cdot \alpha_{den}, 2 \cdot \max(\alpha_{num}, \alpha_{den} - \alpha_{num})) = 2 \cdot \alpha_{den} \qquad \text{since } \max(\alpha, 1-\alpha) \le 1$$

□

Take any two elements X, X' in the `input_domain` and any 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`. Assume X, X' are `d_in`-close under `input_metric` and that `stability_map(d_in)` $\le$ `d_out`.

$$\begin{aligned}
\texttt{d_out} &= \max_{X \sim X'} d_{IDD}(s, s') \\
&= \max_{X \sim X'} \max_{ij} |(s_i - s'_i) - (s_j - s'_j)| && \text{by definition of InfDifferenceDistance} \\
&\le 2 \max_{X \sim X'} \max_i |s_i - s'_i| && s \text{ is not monotonic; take looser bound} \\
&\le 2 \sum_j^{\texttt{d_in}//2} \max_{Z_j \sim Z_{j+1}} \max_i |s_{i,j} - s_{i,j+1}| && \text{by path property where } d_{CO}(Z_i, Z_{i+1}) = 1, X = Z_0 \text{ and } Z_{\texttt{d_in}} = X' \\
&\le 2 \sum_j^{\texttt{d_in}//2} 2 \cdot \alpha_{den} && \text{by 2.1} \\
&\le 4(\texttt{d_in}//2) \cdot \alpha_{den}
\end{aligned}$$

It is shown that `function`(X), `function`(X') are `d_out`-close under `output_metric`.