fn score_candidates_map

Michael Shoemate

November 3, 2025

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

Proves soundness of score_candidates_map in mod.rs at commit f5bb719 (outdated[1]). score_candidates_map returns a specific function that can be used to prove stability of the quantile scoring transformation.

1 Hoare Triple

Precondition

alpha_den > alpha_num.

Function

```python
def score_candidates_map(alpha_num, alpha_den, known_size) -> Callable[[int], int]:
    def stability_map(d_in):
        if known_size:
            return T.inf_cast(d_in // 2).inf_mul(alpha_den)
        else:
            abs_dist_const: u64 = max(alpha_num, alpha_den - alpha_num)  #
            return T.exact_int_cast(d_in).alerting_mul(abs_dist_const)

    return stability_map
```

Postcondition

Theorem 1.1. Define function as follows:

$$\texttt{function}(x) = \text{score\_candidates}_i(x, \texttt{candidates}, \texttt{alpha\_num}, \texttt{alpha\_den}, \texttt{size\_limit})$$

The function calls score_candidates with fixed choices of candidates, alpha_num, alpha_den and size_limit.

If the input domain is the set of all vectors with non-null elements of type TI, input metric is either SymmetricDistance or InsertDeleteDistance, and the postcondition of function is satisfied, then function is stable.

This means that, for any two elements x, x' in the input_domain, where x, x' share the same length if known_length is true, 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, then when x, x' are d_in-close under input_metric and stability_map(d_in) ≤ d_out, function(x), function(x') are d_out-close under output_metric.

The sensitivity of this function differs depending on whether the size of the input vector is known. First, consider the case where the size is unknown.

---

[1]See new changes with `git diff f5bb719..bffbbf2 rust/src/transformations/quantile_score_candidates/mod.rs`

Lemma 1.2. If $d_{Sym}(x, x') = 1$, then $d_\infty(\texttt{function}(x), \texttt{function}(x')) \leq \max(\alpha_{num}, \alpha_{den} - \alpha_{num})$.

Proof. Assume $d_{Sym}(x, x') = 1$.

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

Consider each of the three cases of adding or removing an element in x.

Case 1. Assume x' is equal to x, but with some $x_j < C_i$ added or removed.

$$
\begin{aligned}
&= \alpha_{den} \cdot \max_i ||(1 - \alpha) \cdot \#(x < C_i) - \alpha \cdot \#(x > C_i)| \\
&\qquad - |(1 - \alpha) \cdot (\#(x < C_i) \pm 1) - \alpha \cdot \#(x > C_i)|| \\
&\leq \alpha_{den} \cdot \max_i ||(1 - \alpha) \cdot \#(x < C_i) - \alpha \cdot \#(x > C_i)| \\
&\qquad - (|(1 - \alpha) \cdot \#(x < C_i) - \alpha \cdot \#(x > C_i)| + |\pm (1 - \alpha)|)| && \text{by triangle inequality} \\
&= \alpha_{den} \cdot \max_i |1 - \alpha| && \text{scores cancel} \\
&= \alpha_{den} - \alpha_{num} && \text{since } \alpha \leq 1
\end{aligned}
$$

Case 2. Assume x' is equal to x, but with some $x_j > C_i$ added or removed.

$$
\begin{aligned}
&= \alpha_{den} \cdot \max_i ||(1 - \alpha) \cdot \#(x < C_i) - \alpha \cdot \#(x > C_i)| \\
&\qquad - |(1 - \alpha) \cdot \#(x < C_i) - \alpha \cdot (\#(x > C_i) \pm 1)|| \\
&\leq \alpha_{den} \cdot \max_i ||(1 - \alpha) \cdot \#(x < C_i) - \alpha \cdot \#(x > C_i)| \\
&\qquad - (|(1 - \alpha) \cdot \#(x < C_i) - \alpha \cdot \#(x > C_i)| + |\pm \alpha|)| && \text{by triangle inequality} \\
&= \alpha_{den} \cdot \max_i |\alpha| && \text{scores cancel} \\
&= \alpha_{num} && \text{since } \alpha \geq 0
\end{aligned}
$$

Case 3. Assume x' is equal to x, but with some $x_j = C_i$ added or removed.

$$
\begin{aligned}
&= \alpha_{den} \cdot \max_i ||(1 - \alpha) \cdot \#(x < C_i) - \alpha \cdot \#(x > C_i)| \\
&\qquad - |(1 - \alpha) \cdot \#(x < C_i) - \alpha \cdot \#(x > C_i)|| \\
&= 0 && \text{no change in score}
\end{aligned}
$$

Take the union bound over all cases.

$$
\leq \max(\alpha_{num}, \alpha_{den} - \alpha_{num})
$$

$\square$

Now consider the case where the dataset size is known.

Lemma 1.3. If $d_{CO}(x, x') \leq 1$, then $d_\infty(\texttt{function}(x), \texttt{function}(x')) \leq \alpha_{den}$.

Proof. Assume $d_{CO}(x, x') \leq 1$.

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

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 $x'_j > C_i$.

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

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

$$= \alpha_{den}$$

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.

$$\leq \max(\alpha_{num}, \alpha_{den} - \alpha_{num})$$

equivalent to one removal (see `make_quantile_score_candidates`)

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

$$\leq \max(\alpha_{num}, \alpha_{den} - \alpha_{num})$$

equivalent to one addition (see `make_quantile_score_candidates`)

Take the union bound over all cases.

$$d_\infty(x_i, x'_i) \leq \max(\alpha_{den}, \max(\alpha_{num}, \alpha_{den} - \alpha_{num})) = \alpha_{den}$$

since $\max(\alpha, 1 - \alpha) \leq 1$

$\square$

Proof of postcondition. Assume the input domain is the set of all vectors with non-null elements of type `TI`, input metric is either `SymmetricDistance` or `InsertDeleteDistance`, and the postcondition of `function` is satisfied.

First, consider the case where the size is unknown. Take any two members s, s' 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 s, s' are `d_in`-close under `input_metric` and that `stability_map(d_in)` $\leq$ `d_out`.

$$
\begin{aligned}
\texttt{d\_out} &= \max_{s \sim s'} d_\infty(s, s') && \text{where } s = \texttt{function}(x) \\
&= \max_{s \sim s'} \max_i |s_i - s'_i| && \text{by definition of LInfDistance, without monotonicity} \\
&\leq \sum_j^{\texttt{d\_in}} \max_{Z_j \sim Z_{j+1}} \max_i |s_{i,j} - s_{i,j+1}| && \text{by path property } d_{Sym}(Z_i, Z_{i+1}) = 1, x = Z_0 \text{ and } x' = Z_{\texttt{d\_in}} \\
&\leq \sum_j^{\texttt{d\_in}} \max(\alpha_{num}, \alpha_{den} - \alpha_{num}) && \text{by 1.2} \\
&\leq \texttt{d\_in} \cdot \max(\alpha_{num}, \alpha_{den} - \alpha_{num})
\end{aligned}
$$

This formula matches the stability map in the case where the dataset size is unknown.

Now, consider the case where the size is known. Take any two elements s, s' 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 s, s' are `d_in`-close under `input_metric` and that `stability_map(d_in)` $\leq$ `d_out`.

4

$$\texttt{d\_out} = \max_{s \sim s'} d_\infty(s, s')$$

$$= \max_{s \sim s'} \max_i |s_i - s_i'| \qquad \text{by definition of \texttt{LInfDistance}, without monotonicity}$$

$$\leq \sum_j^{\texttt{d\_in}//2} \max_{Z_j \sim Z_{j+1}} \max_i |s_{i,j} - s_{i,j+1}| \quad \text{by path property } d_{CO}(Z_i, Z_{i+1}) = 1, x = Z_0 \text{ and } Z_{\texttt{d\_in}} = x'$$

$$\leq \sum_j^{\texttt{d\_in}//2} \alpha_{den} \qquad \text{by 1.3}$$

$$\leq (\texttt{d\_in}//2) \cdot \alpha_{den}$$

This formula matches the stability map in the case where the dataset size is known.

It is shown that $\texttt{function}(x)$, $\texttt{function}(x')$ are $\texttt{d\_out}$-close under $\texttt{output\_metric}$ for any choice of input arguments. $\qquad\square$