fn truncate_id_bound

Michael Shoemate

June 4, 2025

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

Proves soundness of `truncate_id_bound` in `mod.rs` at commit f5bb719 (outdated[1]). `truncate_id_bound` returns an uppper bound on dataset distances in terms of the symmetric distance metric.

1 Hoare Triple

Precondition

Caller Verified

- `id_bound.by` = `truncate.by`

Function

```python
def truncate_id_bound(
    id_bound: Bound,
    truncation: Bound,
    total_ids: Optional[int],
) -> Bound:
    # Once truncated, max contributions when grouped by "over" are bounded
    row_bound = Bound.by(truncation.by)

    # In each group, the worst-case row contributions is the
    # the number of ids contributed (known from id_bound)
    # times the number of rows contributed under each id (known from truncation),
    num_ids, num_rows = id_bound.per_group, truncation.per_group
    if num_ids is not None and num_rows is not None:
        row_bound = row_bound.with_per_group(num_ids.inf_mul(num_rows)) #

    # Worst case number of groups contributed is the
    # total number of ids contributed (total_ids)
    # times the number of groups contributed under each id (known from truncation).
    num_groups_via_truncation = None  #
    if total_ids is not None and truncation.num_groups is not None:
        num_groups_via_truncation = total_ids.inf_mul(truncation.num_groups)

    # Alternatively, the number of groups contributed may be known outright from id_bound.
    # Use the smaller of the two if both are known.
    num_groups = option_min(num_groups_via_truncation, id_bound.num_groups)
    if num_groups is not None:
        row_bound = row_bound.with_num_groups(num_groups) #

    return row_bound
```

[1] See new changes with `git diff f5bb719..d584b91 rust/src/transformations/make_stable_lazyframe/truncate/mod.rs`

Postcondition

Theorem 1.1 (Postcondition)**.** Let g vary over groups when partitioned by `id_bound.by`. If for any two datasets x, x' we have that

$$||d_{\text{SymId}}(\texttt{function}(x)_g, \texttt{function}(x')_g)||_\infty \leq \texttt{id\_bound.per\_group},$$
$$||d_{\text{SymId}}(\texttt{function}(x)_g, \texttt{function}(x')_g)||_0 \leq \texttt{id\_bound.num\_groups},$$
$$d_{\text{SymId}}(\texttt{function}(x), \texttt{function}(x')) \leq \texttt{total\_ids},$$

and `function` truncates a dataset such that,

$$\max_{id} ||d_{\text{Sym}}(\texttt{function}(x)_{id,g}, \texttt{function}(x')_{id,g})||_\infty \leq \texttt{truncation.per\_group},$$
$$\max_{id} ||d_{\text{Sym}}(\texttt{function}(x)_{id,g}, \texttt{function}(x')_{id,g})||_0 \leq \texttt{truncation.num\_groups},$$

then we have that

$$||d_{\text{Sym}}(\texttt{function}(x)_g, \texttt{function}(x')_g)||_\infty \leq \texttt{row\_bound.per\_group},$$
$$||d_{\text{Sym}}(\texttt{function}(x)_g, \texttt{function}(x')_g)||_0 \leq \texttt{row\_bound.num\_groups},$$

where `row_bound` denotes the return value.

Proof. Assume the preconditions are met, as well as the conditions of the postcondition.

We first prove the per-group bound.

$$\begin{aligned}
&||d_{\text{Sym}}(\texttt{function}(x)_g, \texttt{function}(x')_g)||_\infty \\
&\leq ||d_{\text{SymId}}(\texttt{function}(x)_g, \texttt{function}(x')_g)||_\infty \cdot \max_{id} ||d_{\text{Sym}}(\texttt{function}(x)_{id,g}, \texttt{function}(x')_{id,g})||_\infty \\
&\leq \texttt{id_bound.per_group} \cdot \texttt{truncation.per_group} \\
&= \texttt{row_bound.per_group} \qquad \text{by line 14}
\end{aligned}$$

We now prove the number of groups bound. There are two ways to bound the number of contributed groups. We first reason by the total number of identifiers.

$$\begin{aligned}
&||d_{\text{Sym}}(\texttt{function}(x)_g, \texttt{function}(x')_g)||_0 \\
&\leq d_{\text{SymId}}(\texttt{function}(x), \texttt{function}(x')) \cdot \max_{id} ||d_{\text{Sym}}(\texttt{function}(x)_{id,g}, \texttt{function}(x')_{id,g})||_0 \\
&\leq \texttt{total_ids} \cdot \texttt{truncation.num_groups} \\
&= \texttt{num_groups_via_truncation} \qquad \text{by line 19}
\end{aligned}$$

We can also directly bound the number of contributed groups by `id_bound.num_groups`:

$$\begin{aligned}
&||d_{\text{Sym}}(\texttt{function}(x)_g, \texttt{function}(x')_g)||_0 \\
&= ||d_{\text{SymId}}(\texttt{function}(x)_g, \texttt{function}(x')_g)||_0 \\
&\leq \texttt{id_bound.num_groups}
\end{aligned}$$

This is a valid upper bound on the number of contributed groups, because truncation is applied independently to each group.

Therefore, we take the minimum of the two upper bounds.

$$\begin{aligned}
&||d_{\text{Sym}}(\texttt{function}(x)_g, \texttt{function}(x')_g)||_0 \\
&\min(\texttt{num_groups_via_truncation}, \texttt{id_bound.num_groups}) \\
&= \texttt{row_bound.num_groups} \qquad \text{by line 27}
\end{aligned}$$

In the implementation, any of the input bounds could be missing. When a bound is missing, any output bounds that require it are not claimed. □