# fn counting\_query\_stability\_map

Michael Shoemate

September 1, 2025

This document proves that the implementation of counting\_query\_stability\_map in mod.rs at commit f5bb719 (outdated<sup>1</sup>) satisfies its proof definition.

## 1 Hoare Triple

## Preconditions

## Compiler-verified

- Argument public\_info must be Keys, Lengths or None
- Generic M must implement UnboundedMetric
- Const generic P must be of type usize

## **User-verified**

None

#### Pseudocode

```
def counting_query_stability_map(
      public_info: Literal["Keys"] | Literal["Lengths"] | None,
  ) -> StabilityMap[PartitionDistance[M], LpDistance[P, f64]]:
      if public_info == "Lengths": #
          return StabilityMap.new(lambda _: 0.)
      def norm_map(d_in: f64) -> f64: #
          if P == 1:
              return d_in
10
          if P == 2:
11
              return d_in.inf_sqrt()
12
          raise ValueError("unsupported Lp norm. Must be an L1 or L2 norm.")
14
      def stability_map(d_in: tuple[u32, u32, u32]) -> f64:
15
          10, 11, l_inf = d_in #
16
          10_p = norm_map(f64.from_(10)) #
17
          11_p = f64.from_(11)
18
          l_inf_p = f64.from_(l_inf)
19
          return l1_p.total_min(l0_p.inf_mul(l_inf_p)) #
20
21
      return StabilityMap.new_fallible(stability_map) #
```

<sup>&</sup>lt;sup>1</sup>See new changes with git diff f5bb719..712ba41 rust/src/transformations/make\_stable\_expr/expr\_count/mod.rs

## Postcondition

**Definition 1.1.** For any setting of the input parameters  $\texttt{public\_info}$ , M and P, returns a StabilityMap where for any predicate  $p(\cdot)$  and  $f(x) = [\sum_i \mathbbm{1}_{p(x_{1i})}, \sum_i \mathbbm{1}_{p(x_{2i})}, \dots]$ , if  $d_{\texttt{PartitionDistance} < M >}(x, x') \le d_{in}$ , then  $d_{LP}(f(x), f(x')) \le d_{out}$ , where  $d_{out} = \texttt{StabilityMap.eval}(d_{in})$ .

*Proof.* Since the input metric is PartitionDistance<M>, and M is an unbounded dataset distance metric (with associated distance type u32), the distance type is a tuple of the  $L_0$ ,  $L_1$  and  $L_{\infty}$  distances between the per-partition distances with respect to the input metric M, as shown on 16.

$$d_{LP}(f(x), f(x')) \tag{1}$$

$$= d_{LP} \left( \left[ \sum_{j=1}^{\text{len}(x_1)} \mathbb{1}_{p(x_{1j})}, \sum_{j=1}^{\text{len}(x_2)} \mathbb{1}_{p(x_{2j})}, \dots \right], \left[ \sum_{j=1}^{\text{len}(x_1)} \mathbb{1}_{p(x'_{1j})}, \sum_{j=1}^{\text{len}(x_2)} \mathbb{1}_{p(x'_{2j})}, \dots \right] \right)$$
(2)

$$= \left(\sum_{i}^{\operatorname{len}(x)} \left(\sum_{j}^{\operatorname{len}(x_{i})} \mathbb{1}_{p(x_{ij})} - \mathbb{1}_{p(x'_{ij})}\right)^{P}\right)^{1/P}$$
(3)

Consider two cases. First, when substituting the  $\Delta_0$  and  $\Delta_\infty$  bounds:

$$= \left(\sum_{i}^{\operatorname{len}(x)} \left(\sum_{j}^{\operatorname{len}(x_{i})} \mathbb{1}_{p(x_{ij})} - \mathbb{1}_{p(x'_{ij})}\right)^{P}\right)^{1/P}$$

$$(4)$$

$$\leq \left(\sum_{i}^{\Delta_0} \left(\Delta_{\infty}\right)^P\right)^{1/P} \tag{5}$$

$$=\Delta_0^{1/P}\Delta_{\infty} \tag{6}$$

Alternatively, apply the  $\Delta_1$  bound, considering that distance is maximized when all  $\Delta_1$  contributions are made to the same partition:

$$= \left(\sum_{i}^{\operatorname{len}(x)} \left(\sum_{j}^{\operatorname{len}(x_{i})} \mathbb{1}_{p(x_{ij})} - \mathbb{1}_{p(x'_{ij})}\right)^{P}\right)^{1/P}$$

$$(7)$$

$$\leq (\Delta_1^P)^{1/P} \tag{8}$$

$$=\Delta_1 \tag{9}$$

The sensitivity is no greater than the smaller of these two bounds:

$$d_{LP}(f(x), f(x')) \le \min(\Delta_1, \Delta_0^{1/P} \Delta_\infty)$$
(10)

We now switch to the pseudocode. If partition length is invariant (via public\_info), then  $\Delta_1 = \Delta_{\infty} = 0$ , making the query insensitive. This invariant is reflected in 5, where an insensitive stability map is returned, satisfying the postcondition.

norm\_map on line 8 computes  $x^{1/P}$ , and is used on line 17 to compute  $\Delta^{1/P}$ . Line 20 then returns the bound from equation 10. Therefore, the stability map returned on line 22 also satisfies the postcondition.