

def quantile_score_candidates

Michael Shoemate

Christian Covington

Ira Globus-Harrus

September 28, 2022

1 Function

The quantile score function scores each c in a set of candidates C .

$$s_i = -|(1 - \alpha)\#(X < C_i) - \alpha\#(X > C_i)| \quad (1)$$

Where $\#(X < C_i) = |\{x \in X | x < C_i\}|$ is the number of values in X less than C_i , and similarly for other variations of inequalities. The scalar score function can be equivalently stated:

$$s_i = -|(1 - \alpha)\#(X < c) - \alpha\#(X > c)| \quad (2)$$

$$= -|(1 - \alpha)\#(X < c) - \alpha * (|X| - \#(X < c) - \#(X = c))| \quad (3)$$

$$= -|\#(X < c) - \alpha * (|X| - \#(X = c))| \quad (4)$$

It has an intuitive interpretation as $-|candidate_rank - ideal_rank|$, where the absolute distance between the candidate and ideal rank penalizes the score. The ideal rank does not include values in the dataset equal to the candidate. The score is maximized at zero when the candidate rank is equivalent to the rank at the ideal α -quantile.

The scalar scorer is almost equivalent to Smith's[1], but adjusts for a source of bias when there are values in the dataset equal to the candidate. For comparison, we can equivalently write the OpenDP scorer as if there were some α -discount on dataset entries equal to the candidate.

$$\begin{aligned} OpenDP &= -|\#(X < c) + \alpha\#(X = c) - \alpha * |X|| \\ Smith &= -|\#(X < c) + 1\#(X = c) - \alpha * |X|| \end{aligned}$$

Observing that $\#(X \leq c) = \#(X < c) + 1\#(X = c)$.

1.1 Examples

Let $X = \{0, 1, 2, 3, 4\}$ and $\alpha = 0.5$ (median):

$$score(X, 0, \alpha) = -|0 - .5 * (5 - 1)| = -2$$

$$score(X, 1, \alpha) = -|1 - .5 * (5 - 1)| = -1$$

$$score(X, 2, \alpha) = -|2 - .5 * (5 - 1)| = -0$$

$$score(X, 3, \alpha) = -|3 - .5 * (5 - 1)| = -1$$

$$score(X, 4, \alpha) = -|4 - .5 * (5 - 1)| = -2$$

The score is maximized by the candidate at the true median.

Let $X = \{0, 1, 2, 3, 4, 5\}$ and $\alpha = 0.5$ (median):

$$\begin{aligned}
score(X, 0, \alpha) &= -|0 - .5 * (6 - 1)| = -2.5 \\
score(X, 1, \alpha) &= -|1 - .5 * (6 - 1)| = -1.5 \\
score(X, 2, \alpha) &= -|2 - .5 * (6 - 1)| = -0.5 \\
score(X, 3, \alpha) &= -|3 - .5 * (6 - 1)| = -0.5 \\
score(X, 4, \alpha) &= -|4 - .5 * (6 - 1)| = -1.5 \\
score(X, 5, \alpha) &= -|5 - .5 * (6 - 1)| = -2.5
\end{aligned}$$

The two candidates nearest the median are scored equally and highest.
Let $X = \{0, 1, 2, 3, 4\}$ and $\alpha = 0.25$ (first quartile):

$$\begin{aligned}
score(X, 0, \alpha) &= -|0 - .25 * (5 - 1)| = -1 \\
score(X, 1, \alpha) &= -|1 - .25 * (5 - 1)| = -0 \\
score(X, 2, \alpha) &= -|2 - .25 * (5 - 1)| = -1 \\
score(X, 3, \alpha) &= -|3 - .25 * (5 - 1)| = -2 \\
score(X, 4, \alpha) &= -|4 - .25 * (5 - 1)| = -3
\end{aligned}$$

As expected, the score is maximized when $c = 1$.
Let $X = \{0, 1, 2, 3, 4, 5\}$ and $\alpha = 0.25$ (first quartile):

$$\begin{aligned}
score(X, 0, \alpha) &= -|0 - .25 * (6 - 1)| = -1.25 \\
score(X, 1, \alpha) &= -|1 - .25 * (6 - 1)| = -0.25 \\
score(X, 2, \alpha) &= -|2 - .25 * (6 - 1)| = -0.75 \\
score(X, 3, \alpha) &= -|3 - .25 * (6 - 1)| = -1.75 \\
score(X, 4, \alpha) &= -|4 - .25 * (6 - 1)| = -2.75 \\
score(X, 5, \alpha) &= -|5 - .25 * (6 - 1)| = -3.75
\end{aligned}$$

The ideal rank is 1.25. The nearest candidate, 1, has the greatest score, followed by 2, and then 0.

2 Constructor: make_quantile_score_candidates

2.1 Pseudocode

```

1 # type: ignore
2 def make_quantile_score_candidates(candidates: List[TI], alpha: T0):
3     abs_dist_const = max(alpha, 1 - alpha)
4     sup_dist_const = 2 * abs_dist_const
5
6     def function(arg: List[TI]):
7         return score(arg, candidates, alpha)
8
9     def privacy_relation(d_in: u32, d_out: T0):
10         return d_in * sup_dist_const <= d_out
11
12     return Transformation(
13         input_domain=VectorDomain(AllDomain(TI)),
14         output_domain=VectorDomain(AllDomain(T0)),
15         function=function,

```

```

16     input_metric=SymmetricDistance(),
17     output_metric=InfDifferenceDistance(),
18     privacy_relation=privacy_relation,
19 )

```

2.2 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.

2.3 Domain-metric compatibility

On the input side, SymmetricDistance is well-defined on VectorDomains. On the output side, InfDifferenceDistance is well-defined on VectorDomains.

2.4 Stability Guarantee

We first derive the sensitivity when neighbors differ by an addition or removal of a record.

$$\begin{aligned}
\Delta_{Sym} &= \max_{s \sim s'} \max_{ij} |(s_i - s'_i) - (s_j - s'_j)| \\
&\leq 2 \max_{s \sim s'} \max_i |s_i - s'_i| && s \text{ is not monotonic; take looser bound} \\
&\leq 2 \max_{s \sim s'} \max_i |((1 - \alpha)\#(X < C_i) - \alpha\#(X > C_i)) \\
&\quad - ((1 - \alpha)\#(X' < C_i) - \alpha\#(X' > C_i))|
\end{aligned}$$

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.

$$\begin{aligned}
&= 2 \max_{s \sim s'} \max_i |((1 - \alpha)\#(X < C_i) - \alpha\#(X > C_i)) \\
&\quad - ((1 - \alpha)\#(X < C_i) \pm 1) - \alpha\#(X > C_i)| \\
&\leq 2 \max_{s \sim s'} \max_i |((1 - \alpha)\#(X < C_i) - \alpha\#(X > C_i)) \\
&\quad - ((1 - \alpha)\#(X < C_i) - \alpha\#(X > C_i)) + |\pm(1 - \alpha)| && \text{by triangle inequality} \\
&= 2 \max_{s \sim s'} \max_i |1 - \alpha| && \text{scores cancel} \\
&= 2(1 - \alpha) && \text{since } \alpha \leq 1
\end{aligned}$$

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

$$\begin{aligned}
&= 2 \max_{s \sim s'} \max_i |((1 - \alpha)\#(X < C_i) - \alpha\#(X > C_i)) \\
&\quad - ((1 - \alpha)\#(X < C_i) - \alpha(\#(X > C_i) \pm 1))| \\
&\leq 2 \max_{s \sim s'} \max_i |((1 - \alpha)\#(X < C_i) - \alpha\#(X > C_i)) \\
&\quad - ((1 - \alpha)\#(X < C_i) - \alpha\#(X > C_i)) + |\pm \alpha| && \text{by triangle inequality} \\
&= 2 \max_{s \sim s'} \max_i |\alpha| && \text{scores cancel} \\
&= 2\alpha && \text{since } \alpha \geq 0
\end{aligned}$$

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

$$\begin{aligned}
&= 2 \max_{s \sim s'} \max_i |((1 - \alpha) \#(X < C_i) - \alpha \#(X > C_i))| \\
&\quad - |(1 - \alpha) \#(X < C_i) - \alpha \#(X > C_i)| \\
&= 0 \qquad \qquad \qquad \text{no change in score}
\end{aligned}$$

Take the union bound over all cases.

$$\Delta_{Sym} = 2 \max(\alpha, 1 - \alpha)$$

3 Constructor: make_sized_quantile_score_candidates

3.1 Pseudocode

```

1 # type: ignore
2 def make_sized_quantile_score_candidates(size: usize, candidates: List[TI], alpha: T0):
3     abs_dist_const = 1
4     inf_diff_dist_const = 2 * abs_dist_const
5
6     def function(arg: List[TI]):
7         return score(arg, candidates, alpha)
8
9     def privacy_relation(d_in: u32, d_out: T0):
10         return d_in * inf_diff_dist_const <= d_out
11
12     return Transformation(
13         input_domain=SizedDomain(VectorDomain(AllDomain(TI)), size),
14         output_domain=VectorDomain(AllDomain(T0)),
15         function=function,
16         input_metric=SymmetricDistance(),
17         output_metric=InfDifferenceDistance(),
18         privacy_relation=privacy_relation,
19     )

```

3.2 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.

3.3 Domain-metric compatibility

On the input side, SymmetricDistance is well-defined on sized VectorDomains. On the output side, InfDifferenceDistance is well-defined on VectorDomains.

3.4 Stability Guarantee

We first derive the sensitivity when neighbors differ by changing one record.

$$\begin{aligned}
\Delta_{CO} &= \max_{s \sim s'} \max_{i,j} |(s_i - s'_i) - (s_j - s'_j)| \\
&\leq 2 \max_{s \sim s'} \max_i |s_i - s'_i| \qquad \qquad \qquad s \text{ is not monotonic; take looser bound} \\
&\leq 2 \max_{s \sim s'} \max_i |((1 - \alpha) \#(X < C_i) - \alpha \#(X > C_i))| \\
&\quad - |(1 - \alpha) \#(X' < C_i) - \alpha \#(X' > C_i)|
\end{aligned}$$

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 $X'_j > C_i$.

$$\begin{aligned}
&= 2 \max_{s \sim s'} \max_i |((1 - \alpha)\#(X < C_i) - \alpha\#(X > C_i)) \\
&\quad - (1 - \alpha)(\#(X < C_i) - 1) - \alpha(\#(X > C_i) + 1)| \\
&\leq 2 \max_{s \sim s'} \max_i |((1 - \alpha)\#(X < C_i) - \alpha\#(X > C_i)) \\
&\quad - ((1 - \alpha)\#(X < C_i) - \alpha\#(X > C_i)) + |1|| \quad \text{by triangle inequality} \\
&= 2 \max_{s \sim s'} \max_i |1| \quad \text{scores cancel} \\
&= 2
\end{aligned}$$

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

$$= 2 \quad \text{by symmetry, follows from Case 1.}$$

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

$$\leq 2 \max(\alpha, 1 - \alpha) \quad \text{equivalent to one removal [2.4](#)}$$

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

$$\leq 2 \max(\alpha, 1 - \alpha) \quad \text{equivalent to one addition [2.4](#)}$$

Take the union bound over all cases.

$$\Delta_{CO} = \max(2, 2 \max(\alpha, 1 - \alpha)) = 2 \quad \text{since } \max(\alpha, 1 - \alpha) \leq 1$$

Represent sensitivity where neighbors differ by a symmetric distance of one instead of change one.

$$\Delta_{Sym} = 1 \quad \text{since one edit is an addition and removal}$$

References

- [1] Adam Smith. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 813–822, New York, NY, USA, 2011. Association for Computing Machinery.