fn peel_permute_and_flip

Michael Shoemate

January 24, 2026

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

This document proves soundness of `peel_permute_and_flip` in `mod.rs` at commit e62b0aa2 (outdated[1]). `peel_permute_and_flip` noisily selects the index of the greatest score from a vector of input scores k times without replacement.

1 Hoare Triple

Preconditions

Types consistent with pseudocode.

Pseudocode

```
def peel_permute_and_flip(
    x: list[RBig],
    scale: RBig,
    k: usize,
    replacement: bool,
):
    natural_order = []
    sorted_order = BTreeSet.new()

    for _ in range(min(k, x.len())):
        index = permute_and_flip(x, scale, replacement)  #
        x.remove(index)  #

        # map index on modified x back to original x (postprocessing)
        for del_ in sorted_order:  # |
            if del_ <= index:
                index += 1
            else:
                break

        sorted_order.insert(index)
        natural_order.push(index)

    return natural_order
```

Postcondition

Theorem 1.1. • If replacement is set, returns a sample from $\mathcal{M}_{EM}$ (as defined in MS2023 Definition 4), otherwise returns a sample from $\mathcal{M}_{PF}$ (as defined in MS2023 Lemma 1), k times by peeling, where $\texttt{scale} = \frac{2 \cdot \Delta}{\epsilon}$.

[1] See new changes with `git diff e62b0aa2..2dc7cad rust/src/measurements/noisy_top_k/mod.rs`

- Errors are data-independent, except for exhaustion of entropy.

Proof. The pseudocode applies `permute_and_flip` on line 11 up to k times (no greater than the number of scores), each time removing the selected score on line 12.

Since the index in subsequent selections may be offset by an earlier removal, a data-independent post-processing of the index is performed on line 15.

The only source of error is due to entropy exhaustion. □