fn peel_permute_and_flip

Michael Shoemate

September 22, 2025

> 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This document proves soundness of `peel_permute_and_flip` in `mod.rs` at commit e62b0aa2 (outdated[1]).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):
    natural_order = []
    sorted_order = BTreeSet.new()

    for _ in range(min(k, x.len())):
        index = permute_and_flip(x, scale)  #
        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. Returns the index of the max element z_i, where each $z_i \sim \text{Exp}(\text{shift} = x_i, \text{scale} = \texttt{scale})$, k times without replacement.

Proof. The pseudocode applies `permute_and_flip` on line 6 up to k times (no greater than the number of scores), each time removing the selected score on line 7.

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 10. □

[1] See new changes with `git diff e62b0aa2..7b6eb5e rust/src/measurements/noisy_top_k/exponential/mod.rs`