fn permute_and_flip

Michael Shoemate

December 12, 2025

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

This document proves soundness of `permute_and_flip` [1] in `mod.rs` at commit e62b0aa2 (outdated[1]). `permute_and_flip` noisily selects the index of the greatest score from a vector of input scores.

1 Hoare Triple

Preconditions

Types consistent with pseudocode.

Pseudocode

```
def permute_and_flip(x: list[RBig], scale: RBig, replacement: bool):
    if scale.is_zero():  #
        return max(range(x.len()), key=lambda i: x[i])

    # begin nonzero scale
    x_max = max(x)
    permutation = list(range(x.len()))

    sequence = range(0, len(x)) if replacement else repeat(0)

    for left in sequence:
        right = left + sample_uniform_uint_below(x.len() - left)
        # fisher-yates shuffle up to left
        permutation.swap(left, right)

        candidate = permutation[left]
        if sample_bernoulli_exp((x_max - x[candidate]) / scale):
            return candidate

    raise "at least one x[candidate] is equal to x_max"
```

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), where `scale` $= \frac{2 \cdot \Delta}{\epsilon}$.
- Errors are data-independent, except for exhaustion of entropy.

Proof. By swapping elements on line 13, an online Fisher-Yates shuffle is applied up to and including index `left`.

[1] See new changes with `git diff e62b0aa2..dbb0d19 rust/src/measurements/noisy_top_k/mod.rs`

Substituting `scale` $= \frac{2\Delta}{\epsilon}$, the argument to `sample_bernoulli_exp` is then $\frac{\epsilon}{2\Delta}(q_* - q_r)$, which is non-negative, satisfying the precondition of `sample_bernoulli_exp`. Therefore by the postcondition of `sample_bernoulli_exp`, the response is a sample from $\text{Bern}(\exp(-x))$, where $x = \frac{\epsilon}{2\Delta}(q_* - q_r)$. Therefore the response is a sample from $\text{Bern}(\exp(\frac{\epsilon}{2\Delta}(q_r - q_*)))$, which is equivalent to Algorithm 1 in [1].

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

References

[1] Ryan McKenna and Daniel Sheldon. Permute-and-flip: A new mechanism for differentially private selection, 2020.