1 Summary of The Article

In this paper, the authors tackle the challenge of performing differentially private linear regression without requiring users to specify complex hyperparameters or data bounds.

They introduce an algorithm called TukeyEM, which builds upon previous work on differentially private linear regression and extends it in two key ways. First, they replace the median with a multidimensional analogue based on Tukey depth to estimate high-depth models. Second, they adapt a technique based on propose-test-release (PTR) to construct an algorithm that doesn't need domain bounds for the exponential mechanism.

The authors demonstrate through empirical evaluation that TukeyEM achieves competitive performance with, and often exceeds, non-privately tuned baseline algorithms across various synthetic and real datasets.

Algorithm 1 TukeyEM

Require: Features matrix $X \in \mathbb{R}^{n \times d}$, label vector $y \in \mathbb{R}^n$, number of models m, privacy parameters ε and δ

1: Evenly and randomly partition X and y into subsets $\{(X_i, y_i)\}_{i=1}^m$
2: **for** $i = 1, \ldots, m$ **do**
3: Compute OLS estimator $\beta_i \leftarrow (X_i^T X_i)^{-1} X_i^T y_i$
4: **end for**
5: **for** dimension $j \in [d]$ **do**
6: $\{\beta_{i,j}\}_{i=1}^m \leftarrow$ projection of $\{\beta_i\}_{i=1}^m$ onto dimension j
7: $(S_{j,1}, \ldots, S_{j,m}) \leftarrow \{\beta_{i,j}\}_{i=1}^m$ sorted in nondecreasing order
8: **end for**
9: Collect projected estimators into $S \in \mathbb{R}^{d \times m}$, where each row is nondecreasing
10: **for** $i \in [m/2]$ **do**
11: Compute volume of region of depth $\geq i$, $V_i \leftarrow \prod_{j=1}^d (S_{j,m-(i-1)} - S_{j,i})$
12: **end for**
13: **if** PTRCheck$(V, \varepsilon/2, \delta)$ **then**
14: $\hat{\beta} \leftarrow$ RestrictedTukeyEM$(V, S, m/4, \varepsilon/2)$
15: **return** $\hat{\beta}$
16: **else**
17: **return** $\perp$
18: **end if**

2 Initial Implementation

We have started to implement by ourselves the algorithm proposed step-by-step, in order to understand each part in its details. So far, we have implemented up to line 12 of Algorithm 1.

As a first testing dataset we use the Synthetic one proposed in the paper: the result of `sklearn.datasets.make_regression` with $n = 22000$ and $d = 11$. We split it into m subsets and fit m OLS estimators. Lines 5 to 12 are then simple manipulations done with `numpy`.

3 Planned Follow-up

- **Finish the Algorithm:** Finish the programming part of the TukeyEM algorithm, mainly focusing on completing the parts about RestrictedTukeyEM and PTRCheck as mentioned in Algorithm 1.
- **Test the Algorithm:** Test how well TukeyEM works by using it on different kinds of data, including both made-up and real-world data, and see how it compares to other algorithms that don't protect privacy.
- **Optimization** Look for ways to make TukeyEM run faster and work better, particularly in calculating the Tukey depth and handling the PTR mechanism more efficiently.

References

[1] Kareem Amin, Matthew Joseph, Mónica Ribero, and Sergei Vassilvitskii. Easy differentially private linear regression. *arXiv preprint arXiv:2208.07353*, 2022.