Supplement: Layered Sampling for Robust Optimization Problems

Hu Ding [1] **Zixiu Wang** [1]

1. Experiments

For both Algorithm 1 and 2, we need to compute an initial solution $\tilde{C}$ or $\tilde{h}$ first. For k-median/means clustering, we run the algorithm of Local Search with Outliers from (Gupta et al., 2017) on a small sample of size $O(k)$ to obtain the k initial centers. We do not directly use the k-means + + method (Arthur & Vassilvitskii, 2007) to seed the k initial centers because it is sensitive to outliers. For linear regression, we run the standard linear regression algorithm on a small random sample of size $O(z)$ to compute an initial solution.

Different coreset methods Each of the following methods returns a weighted set as the coreset, and then we run the alternating minimization algorithm k-means-- (Chawla & Gionis, 2013) (or (Shen & Sanghavi, 2019) for linear regression) on it to obtain the solution. For fairness, we keep the coresets from different methods to have the same coreset size for each instance.

- Layered Sampling (`LaySam`). *i.e.,* Algorithm 1 and 2 proposed in this paper.
- Uniform Sampling (`UniSam`). The most natural and simple method is to take a sample S uniformly at random from the input data set P, where each sampled point has the weight $|P|/|S|$.
- Uniform Sampling + Nearest Neighbor Weight (`NN`) (Gupta et al., 2017; Chen et al., 2018). Similar to `UniSam`, we also take a random sample S from the input data set P. For each $p \in P$, we assign it to its nearest neighbor in S; for each $q \in S$, we set its weight to be the number of points assigned to it.
- `Summary` (Chen et al., 2018). It is a method to construct the coreset for k-median/means clustering with outliers by successively sampling and removing points from the original data until the number of the remaining points is small enough.

The above `LaySam`, `UniSam` and `NN` are also used for linear regression in our experiments. We run 10 trials for each case and take the average. All the experimental results were obtained on a Ubuntu server with 2.4GHz E5-2680V4 and 256GB main memory; the algorithms were implemented in Matlab R2018b.

Performance Measures The following measures will be taken into account in the experiment.

- ℓ_1`-loss`: $\mathcal{K}_1^{-z}$ or $\mathcal{LR}_1^{-z}$.
- ℓ_2`-loss`: $\mathcal{K}_2^{-z}$ or $\mathcal{LR}_2^{-z}$.
- `recall/precision`. Let O^* and O be the sets of outliers with respect to the optimal solution and our obtained solution, respectively. `recall` $= \frac{|O \cap O^*|}{|O^*|}$ and `precision` $= \frac{|O \cap O^*|}{|O|}$. Since $|O^*| = |O| = z$, `recall` $=$ `precision` $= \frac{|O \cap O^*|}{z}$.
- `pre-recall`. It indicates the proportion of O^* that are included in the coreset. Let S be the coreset and `pre-recall` $= \frac{|S \cap O^*|}{|O^*|}$. We pay attention in particular to this measure, because the outliers could be quite important and may reveal some useful information (*e.g.,* anomaly detection). For example, as for clustering, if we do not have any prior knowledge of a given biological data, the outliers could be from an unknown tiny species. Consequently it is more preferable to keep such information when compressing the data. More detailed discussion on the significance of outliers can be found in (Beyer & Sendhoff, 2007; Zimek et al., 2012).

Datasets We consider the following datasets in our experiments.

- `syncluster` We generate the synthetic data as follows: Firstly we create k centers with each dimension randomly located in $[0, 100]$. Then we generate the points following standard Gaussian distributions around the centers.

[1]School of Computer Science and Technology, University of Science and Technology of China. Correspondence to: Hu Ding <huding@ustc.edu.cn, http://staff.ustc.edu.cn/~huding/ >.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

- `synregression` Firstly, we randomly set the d coefficients of hyperplane h in $[-5, +5]$ and construct $\mathbf{x_i}$ in $[0, 10]^{d-1}$ by uniform sampling. Then let y_i be the inner product of $(\mathbf{x_i}, 1)$ and h. Finally we randomly perturb each y_i by $\mathcal{N}(0, 1)$.

- `3DSpatial` ($n = 434874$, $d = 4$). This dataset was constructed by adding the elevation information to a 2D road network in North Jutland, Denmark (Kaul et al., 2013).

- `covertype` ($n = 581012$, $d = 10$). It is a forest cover type dataset from Jock A. Blackard (UCI Machine Learning Repository), and we select its first 10 attributes.

- `skin` ($n = 245057$, $d = 3$). The skin dataset is collected by randomly sampling B, G, R values from face images of various age groups and we select its first three dimension (Bhatt & Dhall).

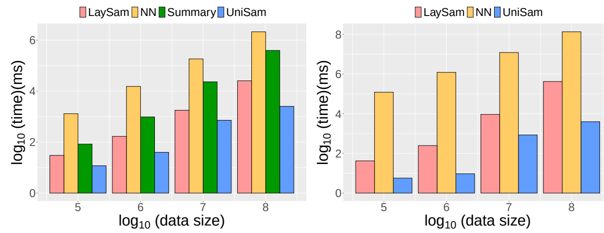
- `SGEMM` ($n = 241600$, $d = 15$). It contains the running times for multiplying two 2048 x 2048 matrices using a GPU OpenCL SGEMM kernel with varying parameters (Ballester-Ripoll et al., 2017).

- `PM2.5` ($n = 41757$, $d = 11$). It is a data set containing the PM2.5 data of US Embassy in Beijing (Liang et al., 2015).

For each dataset, we randomly pick z points to be outliers by perturbing their locations in each dimension. We use a parameter σ to measure the extent of perturbation. For example, we consider the Gaussian distribution $\mathcal{N}(0, \sigma)$ and uniform distribution $[-\sigma, +\sigma]$. So the larger the parameter σ is, the more diffused the outliers will be. For simplicity, we use the notations in the form of `[dataset]-[distribution]-`σ to indicate the datasets, *e.g.*, `syncluster-gauss-`σ.

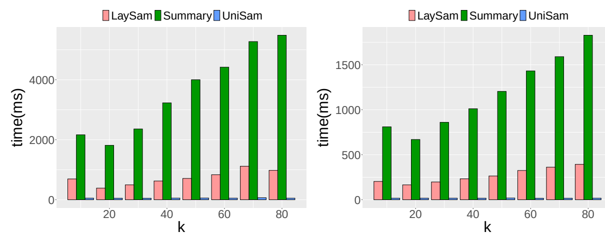
1.1. Coreset Construction Time

We fix the coreset size and vary the data size n of the synthetic datasets, and show the coreset construction times in Figure 1(a) and 1(b). It is easy to see that the construction time of `NN` is larger than other construction times by several orders of magnitude. It is not out of expectation that `UniSam` is the fastest (because it does not need any operation except uniform random sampling). Our `LaySam` lies in between `UniSam` and `Summary`.



(a) (b)

Figure 1. Coreset construction time *w.r.t.* data size. (a) Clustering: coreset size=10^4, $d = 5$ and $k = 10$; (b) Linear Regression: coreset size=10^4 and $d = 20$.



(a) (b)

Figure 2. Construction time *w.r.t.* k. (a) `syncluster` with data size $n = 10^6$, $d = 20$; (b) `covertype`.

We also study the influence of k (for clustering) on the construction time by testing the synthetic datasets and the real-world dataset `covertype` (see Figure 2(a) and 2(b)).

1.2. Performance

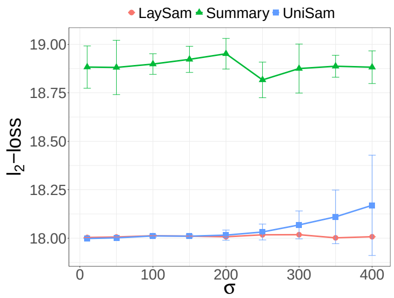
Figure 3(a)-3(e) show the performances of clustering on ℓ_2`-loss` with different σs. The results on ℓ_1`-loss` are very similar (more results are summarized in Table 1 and 2). We can see that `LaySam` outperforms the other methods in terms of both synthetic and real-world datasets. Moreover, its performance remains quite stable (with small standard deviation) and is also robust when σ increases. `UniSam` works well when σ is small, but it becomes very instable when σ rises to large. Both `LaySam` and `UniSam` outperform `Summary` on most datasets.

Similar comparison of the performance for linear regression are shown in Figure 4(a)-4(c).

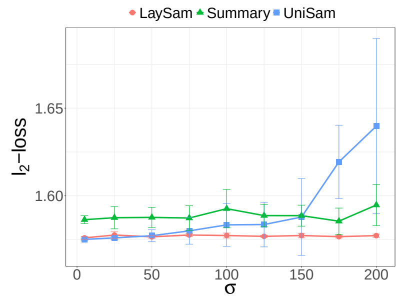
The three coreset methods achieve very close values of `recall` and `precision`. But `UniSam` has much lower `pre-recall` than those of `Summary` and `LaySam`.

References

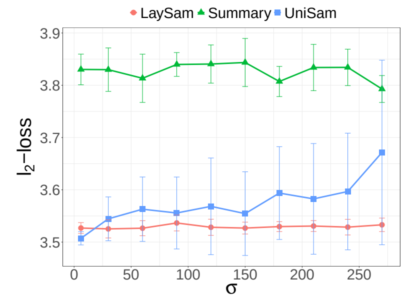
Arthur, D. and Vassilvitskii, S. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth*



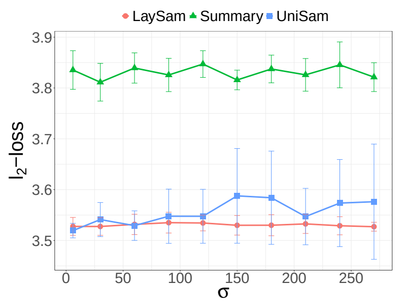
(a) `syncluster-gauss-`σ, with $d = 20$, $k = 10$,$z = 2 \cdot 10^4$ and coreset size= $5 \cdot 10^4$.



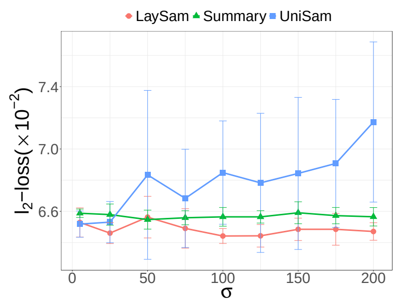
(b) `3DSpatial-gauss-`σ, with $k = 5$, $z = 4000$ and coreset size= 10^4.



(c) `covertype-gauss-`σ, with $k = 20$, $z = 10^4$ and coreset size= $3 \cdot 10^4$.

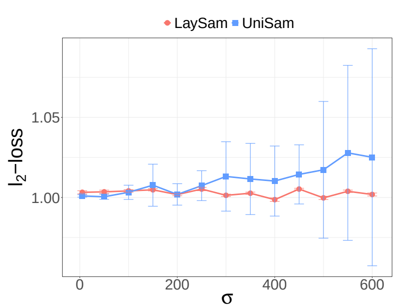


(d) `covertype-uniform-`σ, with $k = 20$, $z = 10^4$ and coreset size= $3 \cdot 10^4$.

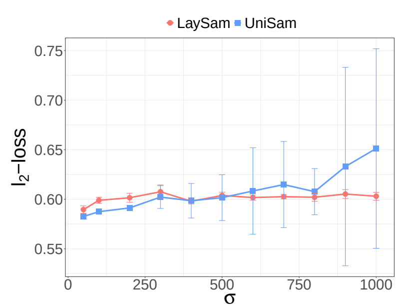


(e) `skin-uniform-`σ, with $k = 40$, $z = 3000$ and coreset size= 10^4.

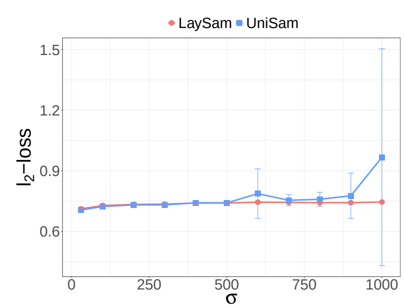
Figure 3. Clustering: ℓ_2`-loss` and stability *w.r.t.* σ.



(a) `synregression-gauss-`σ, with $d = 20$, $z = 10^4$ and coreset size=$2 \cdot 10^4$.



(b) `SGEMM-gauss-`σ, with $z = 4000$ and coreset size=10^4.



(c) `PM2.5-uniform-`σ, with $z = 1000$ and coreset size=3000.

Figure 4. Linear Regrssion: ℓ_2`-loss` and stability *w.r.t.* σ.

(a) `syncluster-gauss-`σ, with $d = 20$, $k = 10$,$z = 2 \cdot 10^4$ and coreset size= $5 \cdot 10^4$.

σ	20			100			200		
Coreset	LaySam	UniSam	Summary	LaySam	UniSam	Summary	LaySam	UniSam	Summary
ℓ_1-loss	3.976	3.976	4.073	3.976	3.976	4.067	3.977	4.048	4.074
ℓ_2-loss	18.01	18	18.9	18.01	18.01	18.84	18.01	18.14	18.91
Prec	1	1	1	1	1	1	1	1	1
Pre-Rec	1	0.0506	1	1	0.0499	1	1	0.0503	1

(b) `covertype-gauss-`σ, with $k = 20$, $z = 10^4$ and coreset size= $3 \cdot 10^4$.

σ	20			100			200		
Coreset	LaySam	UniSam	Summary	LaySam	UniSam	Summary	LaySam	UniSam	Summary
ℓ_1-loss	1.781	1.781	1.868	1.779	1.785	1.871	1.786	1.799	1.853
ℓ_2-loss	3.501	3.501	3.818	3.505	3.523	3.874	3.515	3.576	3.767
Prec	1	1	1	1	1	1	1	1	1
Pre-Rec	1	0.0524	1	1	0.0511	1	1	0.0547	1

(c) `skin-uniform-`σ, with $k = 40$, $z = 3000$ and coreset size= 10^4.

σ	20			100			200		
Coreset	LaySam	UniSam	Summary	LaySam	UniSam	Summary	LaySam	UniSam	Summary
ℓ_1-loss	0.1906	0.1906	0.1925	0.1883	0.1902	0.1914	0.1954	0.2032	0.1957
ℓ_2-loss($\times 10^{-2}$)	6.46	6.531	6.579	6.449	6.849	6.565	6.471	7.173	6.565
Prec	0.9993	0.9995	0.9993	1	1	1	1	1	1
Pre-Rec	0.9997	0.0403	1	1	0.0403	1	1	0.0437	1

Table 1. Performance on clustering with outliers.

(a) `synregression-gauss-`σ, with $d = 20$, $z = 10^4$ and coreset size=$2 \cdot 10^4$.

σ	20		300		600	
Coreset	LaySam	UniSam	LaySam	UniSam	LaySam	UniSam
ℓ_1-loss	0.7996	0.7982	0.7989	0.8007	0.7987	0.7998
ℓ_2-loss	1.003	0.9993	1.001	1.01	1.002	1.025
Prec	0.9305	0.931	0.9953	0.9953	0.9975	0.9978
Pre-Rec	0.9535	0.0108	0.9968	0.0093	0.9975	0.0098

(b) `PM2.5-uniform-`σ, with $z = 1000$ and coreset size=3000.

σ	20		500		1000	
Coreset	LaySam	UniSam	LaySam	UniSam	LaySam	UniSam
ℓ_1-loss	0.6292	0.6244	0.637	0.6368	0.6349	0.6791
ℓ_2-loss	0.7179	0.7075	0.744	0.7473	0.7456	0.8819
Prec	0.9188	0.921	0.99	0.9901	0.9918	0.9917
Pre-Rec	0.9714	0.0732	0.999	0.0725	1	0.0734

Table 2. Performance on linear regression with outliers.

Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, pp. 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 978-0-898716-24-5.

Ballester-Ripoll, R., Paredes, E. G., and Pajarola, R. Sobol tensor trains for global sensitivity analysis. *CoRR*, abs/1712.00233, 2017.

Beyer, H.-G. and Sendhoff, B. Robust optimization–a comprehensive survey. *Computer methods in applied mechanics and engineering*, 196(33-34):3190–3218, 2007.

Bhatt, R. and Dhall, A. Skin segmentation dataset. *UCI Machine Learning Repository*.

Chawla, S. and Gionis, A. k-means--: A unified approach to clustering and outlier detection. In *SDM*, 2013.

Chen, J., Azer, E. S., and Zhang, Q. A practical algorithm for distributed clustering and outlier detection. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pp. 2253–2262, 2018.

Gupta, S., Kumar, R., Lu, K., Moseley, B., and Vassilvitskii, S. Local search methods for k-means with outliers. *Proc. VLDB Endow.*, 10(7):757–768, March 2017. ISSN 2150-8097. doi: 10.14778/3067421.3067425.

Kaul, M., Yang, B., and Jensen, C. Building accurate 3d spatial networks to enable next generation intelligent transportation systems. volume 1, 06 2013. doi: 10.1109/MDM.2013.24.

Liang, X., Zou, T., Guo, B., Li, S., Zhang, H., Zhang, S., Huang, H., and Chen, S. Assessing beijing's pm 2.5 pollution: severity, weather impact, apec and winter heating. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 471:20150257, 10 2015. doi: 10.1098/rspa.2015.0257.

Shen, Y. and Sanghavi, S. Iterative least trimmed squares for mixed linear regression. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 6076–6086, 2019.

Zimek, A., Schubert, E., and Kriegel, H.-P. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, 2012.