Diversity Maximization in the Presence of Outliers (Supplementary File)

Daichi Amagata

Osaka University
amagata.daichi@ist.osaka-u.ac.jp

Space Complexity

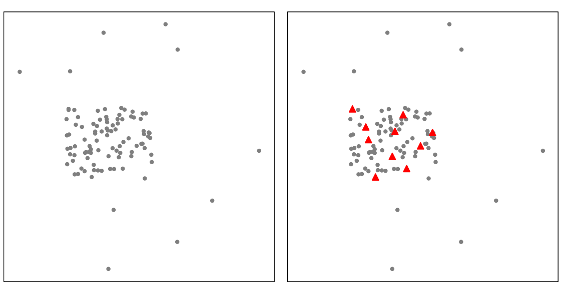
GREEDY. As it always maintains X and S, its space complexity is $O(n)$.

STREAMING. As it can maintain only S, its space complexity is $O(k)$. (We can input X' in a streaming manner.)

CORESET. As with STREAMING, its space complexity is $O(k)$. If it maintains C, where $|C| = O(z) \geq z + k$, its space complexity becomes $O(z)$.

Additional Experiment

Result obtained by STREAMING. Figure 1(a) illustrates an example of X consisting of points including outliers[1], whereas Figure 1(b) shows a diverse set obtained by STREAMING. (As this X contains a small number of points, CORESET returns the same solution.) From Figure 1, STREAMING also returns only inliers, different from GMM (see Figure 2 of our main paper).



(a) Example of X (b) STREAMING

Figure 1: Result set obtained by STREAMING ($k = 10$)

Result of GMM and PODS19. Table 1 clearly shows that most points in S computed by GMM and PODS19 are outliers. This result demonstrates that simply running an existing algorithm for the problem of Max-Min diversification *without* outliers does not work.

Standard deviation result. Table 2 reports the standard deviation w.r.t. $div(S)$ of GREEDY, STREAMING, and CORESET. We used the default parameter setting. CORESET has a larger standard deviation than the others, and this result is actually reasonable. The coreset C has a much smaller number of points than $n = |X|$, thus $div(S)$ of CORESET tends to depend on the first random point of S. Since GREEDY and STREAMING use X, they do not have this tendency.

Table 1: Average number of outliers in S ($k = 100$)

Algorithm	FCT	Household	KDD99	Mirai
GMM	99.00	99.00	99.00	99.00
PODS19	92.35	91.55	87.40	84.05

Table 2: Standard deviation of $div(S)$

Algorithm	FCT	Household	KDD99	Mirai
GREEDY	0.483	0.460	1.211	1.639
STREAMING	0.217	0.240	0.564	5.344
CORESET	2.345	1.506	4.307	4.444

Impact of success probability p. Table 3 shows the average $div(S) = \min_{x,x' \in S} dist(x, x')$ and running time [msec] of CORESET with different p. (Note that CORESET did not return any outliers for these values of p.)

We see that $div(S)$ with $p = 0.9$ is smaller than those with $p = 0.95$ and $p = 0.99$. On the other hand, the running time becomes longer as p becomes larger. For the running time, this result is reasonable, since a smaller p constructs a coreset with a smaller size. (Recall that the time complexity of COREST is $O(kc)$, where c is the coreset size.) This result is also reasonable for $div(S)$. Given a larger p, a coreset contains more points in X, so $\min_{x,x' \in S} dist(x, x')$ tends to be larger.

[1]This is the same set as that in Figure 2 of our main paper.

Table 3: CORESET's average $div(S)$ and running time [msec] ($k = 100$ and $z = 200$)

p	FCT		Household		KDD99		Mirai	
	$div(S)$	Time	$div(S)$	Time	$div(S)$	Time	$div(S)$	Time
0.90	48.996	1.369	36.916	1.553	73.690	3.253	101.880	30.500
0.95	50.158	5.323	38.369	5.823	77.064	8.523	106.352	97.760
0.99	51.425	13.422	39.294	27.239	80.153	54.130	107.272	476.063