Clustering Comparison Results Report

Table of Contents

Introduction

This report is about comparing Kmeans algorithm with default setting (only has Euclidean distance and has no spilling setting) of Python Sklearn package with SimCluster+ algorithms(including different distances and range percentiles) of SimMachines company in terms of f1 scores, timing and silhouette score with different number of clusters(i.e k=10/50/100/500/1000)

To better compare these two under different conditions, I set two different parts:
part 1 is Sklearn vs. SImcluster+ apples to apples which have them using the same one-hot-encoded dataset, the same advanced parameters excluding distance and range percentile(it's impossible to be the totally same because SimCluster+ have some of its own parameters, but I keep all parameters that they commonly have the same);
Part 2 is Sklearn vs. Simcluster+ apples to orange which have SimCluster+ deal with original mixed dataset and use its own default advanced parameters setting(excluding distance and range percentiles).

At the same time, I tried both balanced and imbalanced datasets in two parts.

Conclusions

Conclusions based on performance of f1 scores

Note: I tried both threshold based on majority and threshold based on class-ratio for imbalanced dataset just in case you need them, but the conclusion here are based on threshold based on majority by default.

If only comparing Sklearn kmeans with SimCluster+ Euclidean with no range percentiles setting, they have similar performances, but Sklearn kmeans would slightly outperform than SimCluster+ in most of the cases(you can check the gold highlighted content in the f1 scores tables showed in result part).

If only comparing Sklearn kmeans with SimCluster+ Euclidean/Manhattan/One-class under spilling, SimCluster+ can always beat Sklearn kmeans and get much better results(especially for imbalanced dataset). In general, Euclidean with spilling would be the best on balanced dataset while Manhattan with spilling would be the best on imbalanced dataset.

For one-hot-encoded dataset and mixed data, SimCluster+ get similar f1 scores on these two kind of datasets. Even though SimCluster+ performs a little better on one-hot-encoded dataset, it takes too much more time(about 5-10 times in average) to run on it.

Notably, SimCluster+ one-class with spilling outperform all the time with one-hot-encoded-imbalanced dataset. In other cases, SimCluster+ one-class perform not bad but still slightly underperform than other two SimCluster+ distances.

Conclusions based on performance of timings

Note: before I did timing test on Sklearn kmeans, I killed all other running programs to make it less affected by the CPU or laptop configuration.

First, Like I mentioned above, SimCluster+ would be much slower on one-hot-encoding datasets comparing with Sklearn. In other word, Under same conditions(i.e same one-hot-encoding dataset, same INITIALIZATION_NUMBER and MAX_ITERATIONS parameters' setting) with Sklearn, SimCluster+ would be much slower than Sklearn and f1 score only be little better than Sklearn.
But SimCluster+ would have much better timing performance on mixed dataset which is 8-10 faster than that on one-hot-encoding dataset, but still slower than Sklearn kmeans.

Second, generally speaking, no matter for one-hot-encoding datasets or not, or for balanced or imbalanced datasets or not, if ordering timing descendingly, it would be SimCluster+ One-cluster with spilling > SimCluster+ Manhattan with spilling > SimCluster+ Euclidean with spilling > SimCluster+ One-cluster without spilling > SimCluster+ Manhattan without spilling > SimCluster+ Euclidean without spilling > Sklearn kmeans; Spilling methods would be generally slower 8-10 times than sklearn kmeans(please check this statement in timing section in result part below).

Third, timing of sklearn would not as stable as SimCluster+, especially when k is larger(like 500 or 1000), run times would very likely to increase dramatically. For SimCluster+ methods, they look more stable than Sklearn (though one-class would be relatively little non-stable than other SimCluster+ methods).

Conclusions based on performance of silhouette score

I calculated SimCluster+ silhouette score with transformed data gotten by SimCluster+.

Sklearn always get much better silhouette score(around 0.1 and 0.2) while all other SimCluster+ get much lower silhouette score(between -0.8 and -0.1 in general).

But this should be ok, because SimCluster+ has good performance on f1 scores which is more meaningful in business.

Conclusions based on visualization

Strictly speaking, this this is not a conclusion. I used one 2D dataset with different shaped clusters to try to get an intuitive idea about difference between Sklearn and SimCluster+ Euclidean. Based on the visualization, both of them can't correctly cluster the clusters very well but we can still see the difference between sklearn kmeans and SimCluster+ Euclidean. (see 'Additional part: SimCluster vs. Sklearn on visualization' section in result part for details)

Limitations

After Contemplating about time consumption. I used relatively small datasets comparing with the dataset we used in business. So some results may not be representative or general, but I believe they would inspire or give some ideas more or less to the data scientists.

Datasets

There are two original datasets, FICO and Default payment(i.e DP) datasets.

FICO: https://community.fico.com/s/explainable-machine-learning-challenge?tabset-3158a=2
Default payment: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

To better compare the performance, another two datasets with one hot encoding(i.e ohc) are

Dataset Name	Size	Predictors' situation	Target variable's situation	Target variable's ratio(1:0)
FICO(Original)	(9871,25)	num:23 Cat: 2	balanced	52:50
Default Payment(Original)	(24864,25)	num:16 Cat: 9	unbalanced	6:94
FICO(with ohc)	(9871,37)	All real	balanced	52:50
Default Payment(with ohc)	(24864,83)	All real	unbalanced	6:94

built based on original datasets:

Preprocessing

Because I emphasize on comparison, simple preprocessing are done, like removing all-null records, imputing missing values using K nearest neighbor(k=5)

Methods

There are two parts:

Part 1: SimCluster+ vs. kmeans: apples to apples using one hot encoding.
Because Sklearn Kmeans can't directly deal with categorical data like SimCluster+ , I used datasets with one hot encoding in both experiments(I set all variables to 'real' in spec); To keep apples to apple, I used default quantile transformation in Sklearn package for data transformation/normalization which is similar to what we used in SimCluster+, and I set 'INITIALIZATION_NUMBER=10,MAX_ITERATIONS=300' in SimCluster+ which is the same as Sklearn Kmeans.

Part 2: SimCluster+ vs. kmeans: apples to oranges with default SimCluster+.
To perform/investigate our advantage, I put non-one hot encoded datasets in SimCluster+ with all setting by default when keeping Sklearn Kmeans experiment with quantile transformation and one hot encoding.

For each part, I do regular Sklearn Kmeans which means distance='Euclidean', and I do different experiments on SimCluster+ with different distance(distance = 'Euclidean'/ 'Manhattan'/ 'One Class') and spilling(range percentile= '1'/ '0.1')

Details showed below:

Part Name	Experiment object	dataset	Transformation method	Experiement method
	Sklearn Kmeans		Sklearn Quantile Transformer	-Distance = 'Euclidean'; - INITIALIZATION_NUMBE

Part 1:Apples to apples		FICO(with ohc) & DP(with ohc)		R=10,MAX_ITERATIONS= 300
	SimCluster+		SimCluster+ in-built transformer	-Distance = 'Euclidean'/ 'Manhattan'/ 'One Class'; -Range percentile= '1'/ '0.1' - INITIALIZATION_NUMBER=10,MAX_ITERATIONS= 300

Part Name	Experiment object	dataset	Transformation method	Experiement method
Part 2: Apples to oranges	Sklearn Kmeans	FICO(with ohc) & DP(with ohc)	Sklearn Quantile Transformer	-Distance = 'Euclidean'; - INITIALIZATION_NUMBER=10,MAX_ITERATIONS= 300
	SimCluster+	FICO(original) & DP(original)	SimCluster+ in-built transformer	-Distance = 'Euclidean'/ 'Manhattan'/ 'One Class'; -Range percentile= '1'/ '0.1' - INITIALIZATION_NUMBER=10,MAX_ITERATIONS= 300

Results

Reading Guidance

Note: All charts and graphs below are available in 'sklearn_wb_comparison(for develpters).xlsx', if you only want to see those with no explanation, please check out that file. You can see results of different experiments/parts by choosing different tabs below.

For performance metrics, I mainly used F1 score, timing and silhouette score for the following.

For different colors highlighting in table in different k, Gold indicates the better score in terms of Sklearn kmeans and SimCluster+ Euclidean; Yellow indicates the best score in terms of SKlearn kmeans and SimCluster+ Euclidean/Manhattan/One Class without spilling ; Orange indicates the best score in terms of all of the experiments(i.e SKlearn kmeans and SimCluster+ Euclidean/Manhattan/One Class and spilling and non-spilling).

Euclidean/Manhattan/One-class/-original: Simcluster+ with Euclidean/Manhattan/One-class without spilling(i.e range percentile=1)
Euclidean/Manhattan/One-class/-spilling : Simcluster+ with Euclidean/Manhattan/One-class with spilling(i.e range percentile=0.1)

Time unit: second(s)

The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is (b - a) / max(a, b).

The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar.
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

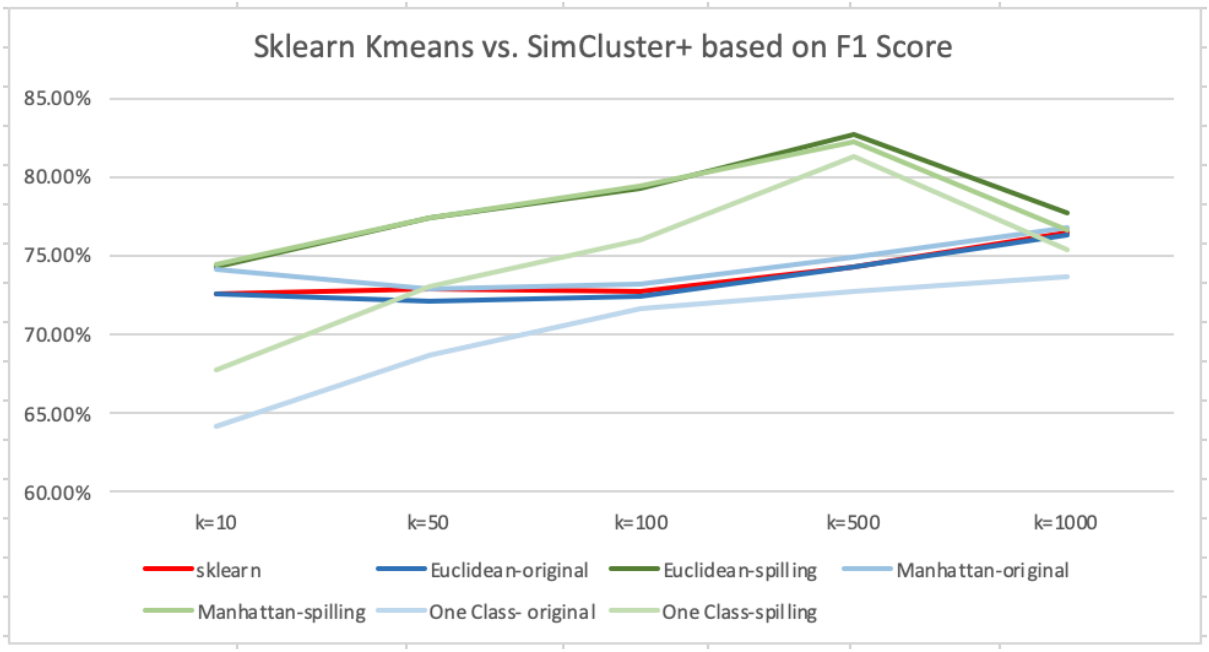
Part 1: SimCluster+ vs. kmeans: apples to apples using one hot encoding

Dataset 1: FICO (with ohc) – balanced dataset

Performance on F1 Score

Table 1: performances of sklearn kmeans vs simcluster+ based on f1 score

	k=10	k=50	k=100	k=500	k=1000
sklearn	72.52%	72.90%	72.68%	74.21%	76.40%
Euclidean-original	72.49%	72.02%	72.37%	74.19%	76.23%
Euclidean-spilling	74.26%	77.45%	79.19%	82.73%	77.72%
Manhattan-original	74.15%	72.81%	73.18%	74.96%	76.78%
Manhattan-spilling	74.48%	77.39%	79.35%	82.20%	76.53%
One Class- original	64.09%	68.61%	71.67%	72.75%	73.71%
One Class-spilling	67.74%	72.99%	76.04%	81.32%	75.40%



For the table, If we only compare Sklearn and SimCluster+ Euclidean, then as Gold highlighted, Sklearn little outperform than SimCluster+ with only +/- 0.2% difference which is not bad;

If we compare Sklearn and different distance in SImCluster+ without spilling, as yellow highlighted, other than k=50 where sklearn is the best, in other cases, SimCluster+ Manhattan is the best.
If we compare Sklearn with all of different distance and range percentile, as orange highlighted, SimCluster+ can always get the best results with spilling on Euclidean or Manhattan distance.

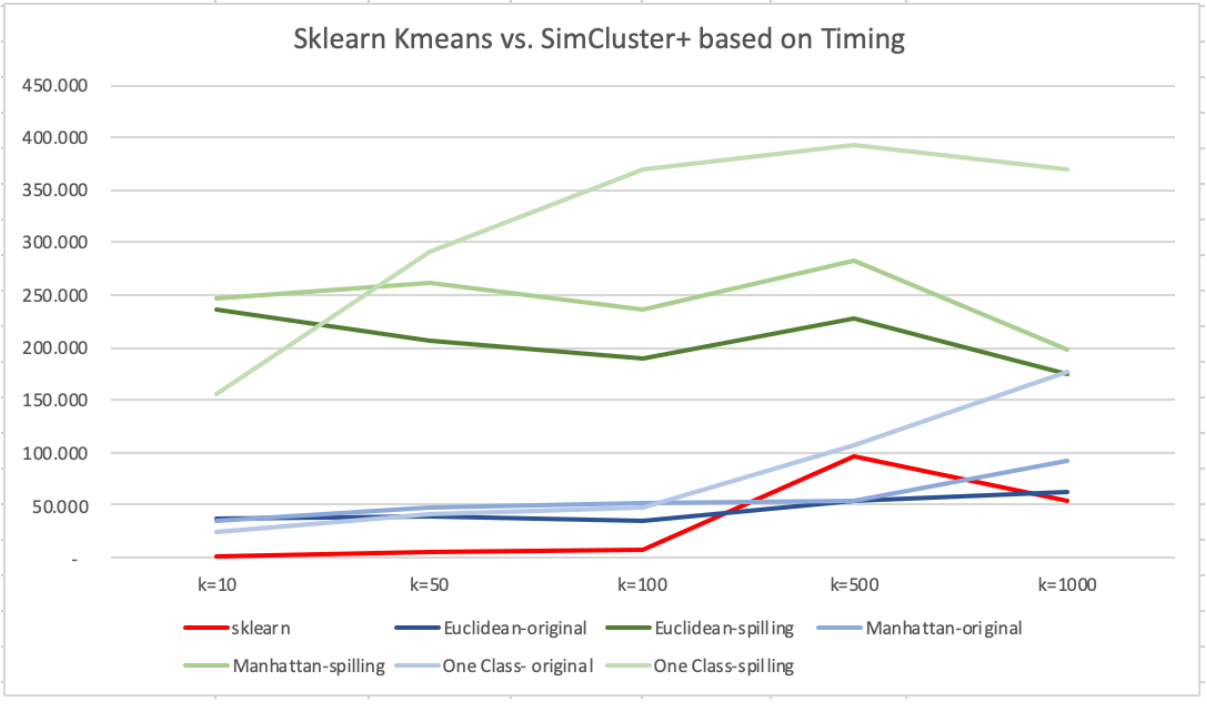
For the visualization, sklearn has similar trend with SimCluster+ Euclidean and Manhattan without spilling excepting one-class, and performances of Euclidean and Manhattan with spilling are all higher than Sklearn, even with one-class with spilling, the performance start to be better than Sklearn after k=50.

Maybe because of the spilling feature, all Spilling performances started to drop after k=500.

Performance on Timing

Table 2: performances of sklearn kmeans vs simcluster+ based on timing

	k=10	k=50	k=100	k=500	k=1000
sklearn	0.967	4.384	7.543	96.53	53.795
Euclidean-original	36.466	40.309	36.021	54.658	62.713
Euclidean-spilling	236.03	207.556	189.433	228.647	174.161
Manhattan-original	35.446	48.399	52.14	54.651	92.121
Manhattan-spilling	246.051	260.956	235.586	282.134	197.87
One Class- original	24.597	41.351	48.405	107.362	177.729
One Class-spilling	154.864	290.895	370.038	392.346	370.369

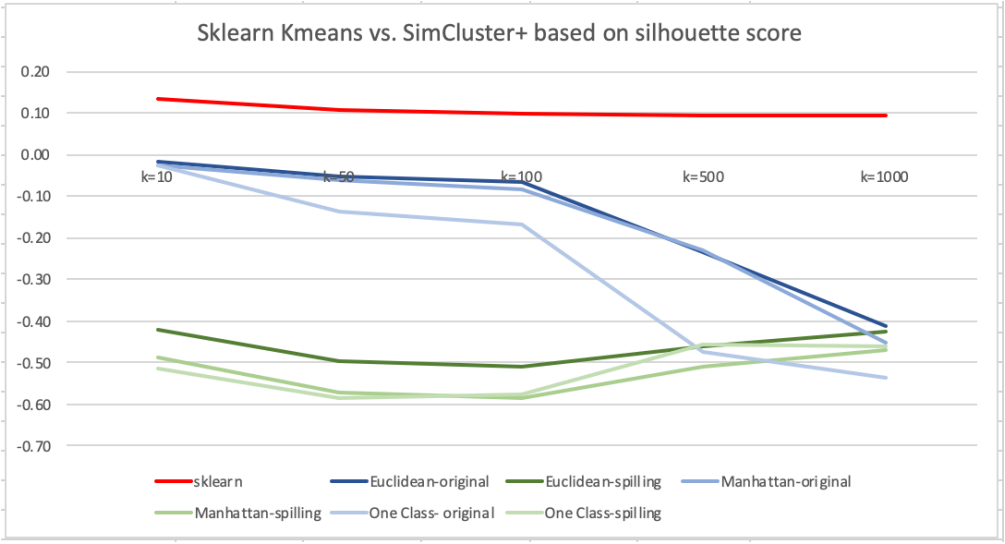


From the table, we can see that Sklearn Kmeans is faster than other SimCluster+ methods excepting k=500 which seems like an anomaly. But since this is a small datasets with only about 10 thousand records, we believe that our SimCluster+ has very conspicuous advantage on very big datasets with millions of records.

From the visualization, we can see that Simcluster+ with spilling would take longer which make sense because they would making more clusters. Simcluster+ Euclidean/Manhattan without spilling are less affected by increasing k than Sklearn.

Performance on silhouette score

Table 3: performances of sklearn kmeans vs simcluster+ based on silhouette score					
	k=10	k=50	k=100	k=500	k=1000
sklearn	0.14	0.11	0.10	0.09	0.09
Euclidean-original	-0.02	-0.05	-0.07	-0.23	-0.41
Euclidean-spilling	-0.42	-0.50	-0.51	-0.46	-0.43
Manhattan-original	-0.02	-0.06	-0.09	-0.23	-0.45
Manhattan-spilling	-0.49	-0.57	-0.58	-0.51	-0.47
One Class- original	-0.03	-0.14	-0.17	-0.47	-0.54
One Class-spilling	-0.51	-0.59	-0.58	-0.46	-0.46



From table and visualization, we can see that Sklearn always keep the best silhouette scores, and SimCluster+ always have worse scores. K=100 is an inflection point and after that scores of SimCluster+ with spilling start to go up and scores of SimCluster+ without spilling starts to drop down.

One-class still shows bad results, but it's maybe because it is designed for dealing with unbalanced data.

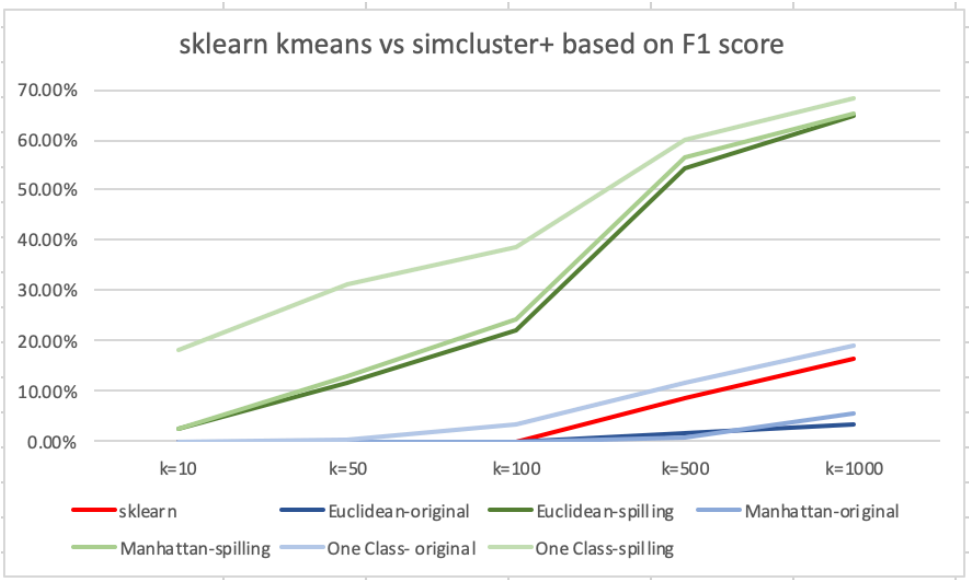
Dataset 2: Default Payment (with ohc) – unbalanced dataset

Performance on F1 Score

I tried two thresholds here, one is threshold=0.5(i.e majority based) and one is threshold=0.6(i.e class ratio based)

1) When threshold =0.5:

Table 1: performances of sklearn kmeans vs simcluster+ based on F1 score					
threshold (majority based)					
	k=10	k=50	k=100	k=500	k=1000
sklearn	0.00%	0.00%	0.00%	8.40%	16.23%
Euclidean-original	0.00%	0.00%	0.00%	1.45%	3.36%
Euclidean-spilling	2.35%	11.73%	22.20%	54.14%	64.73%
Manhattan-original	0.00%	0.00%	0.00%	0.53%	5.69%
Manhattan-spilling	2.23%	12.82%	24.09%	56.33%	65.27%
One Class- original	0.00%	0.27%	3.50%	11.64%	18.94%
One Class-spilling	18.00%	31.09%	38.49%	59.98%	68.28%



From the table, we can see that Sklearn is better when only comparing with SimCluster+ Euclidean without spilling

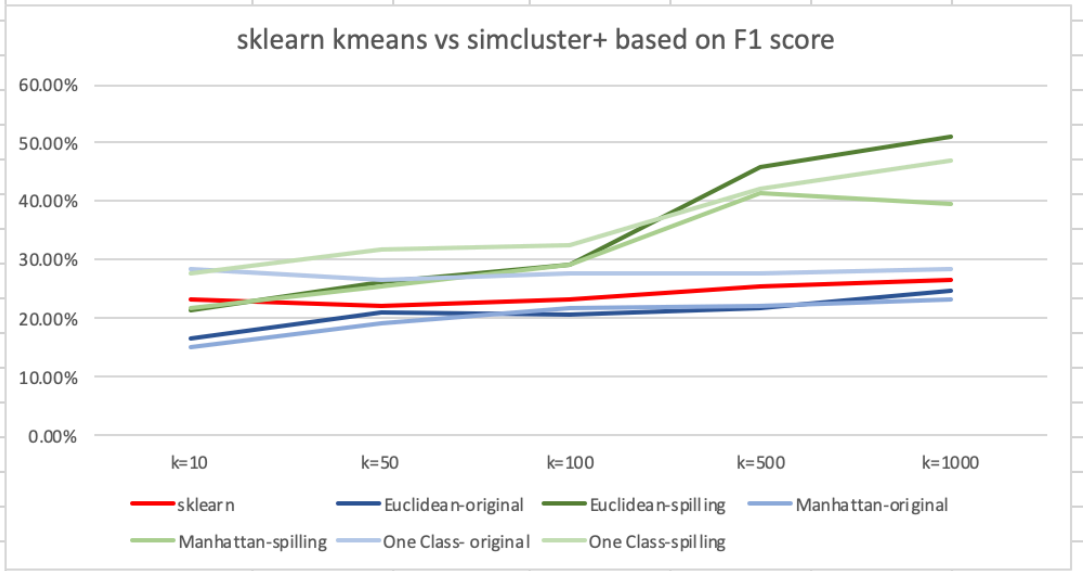
If we compare Sklearn with SimCluster+ with all distances without spilling, SimCluster+ with one-class is the best.

If we compare SKlearn with all SimCluster+ methods, we can see that SImCluster+ One-class with spilling outperforms than others and would have big differences with others (especially when k=10 and k=50)

From the visualization, we can see the outperformance of SimCluster+ spilling more obvious: Sklearn starts to have f1 score more than 0 when k=100 while SimCluster+ with spilling methods and one class without spilling method have better result from the beginning.

2) When threshold = 0.06:

threshold (class ratio based)	k=10	k=50	k=100	k=500	k=1000
sklearn	23.08%	21.98%	23.38%	25.63%	26.71%
Euclidean-original	16.60%	21.18%	20.53%	21.93%	24.89%
Euclidean-spilling	21.27%	26.28%	29.12%	45.85%	50.94%
Manhattan-original	14.94%	19.21%	21.68%	22.26%	23.27%
Manhattan-spilling	21.78%	25.36%	29.08%	41.49%	39.42%
One Class- original	28.47%	26.58%	27.53%	27.53%	28.44%
One Class-spilling	27.80%	31.66%	32.66%	42.33%	47.21%



From the table, we can see that if there are only Sklearn and SimCluster+ Euclidean without spilling, Sklearn would always be the best.

If compering Sklearn with SimCluster+ Euclidean/Manhattan/One-class without spilling, SimCluster+ with One-class would always be the best.

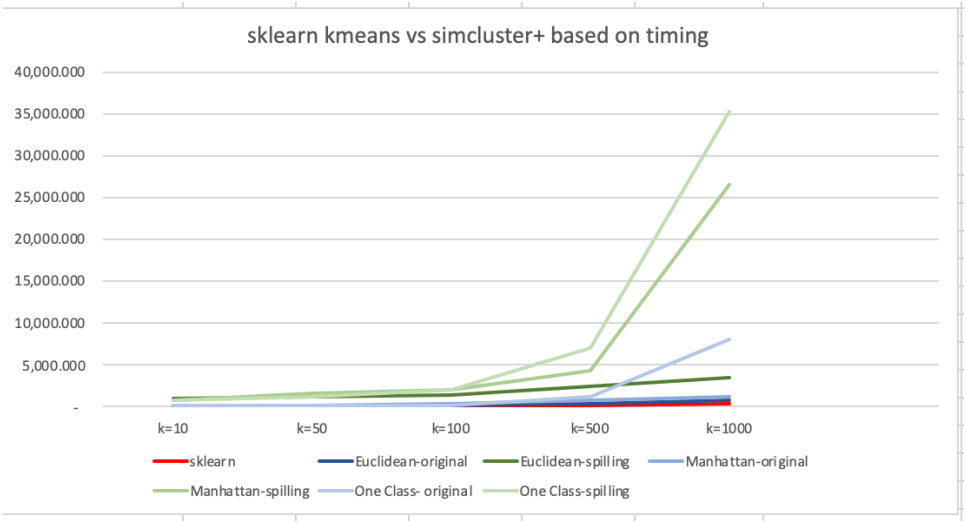
If compering Sklearn with SimCluster+ Euclidean/Manhattan/One-class with spilling, SimCluster+ Euclidean/ One-class with spilling would beat Sklearn and other methods.

From the visualization, we can see that SimCluster+ one-class with/without spilling and other spilling method would much more higher than Sklearn in general, even with SimCluster+ Euclidean/Manhattan without spilling, there is no very big differences between them and Sklearn.

*Once-class have remarkable performance in this case.

Performance on the timing

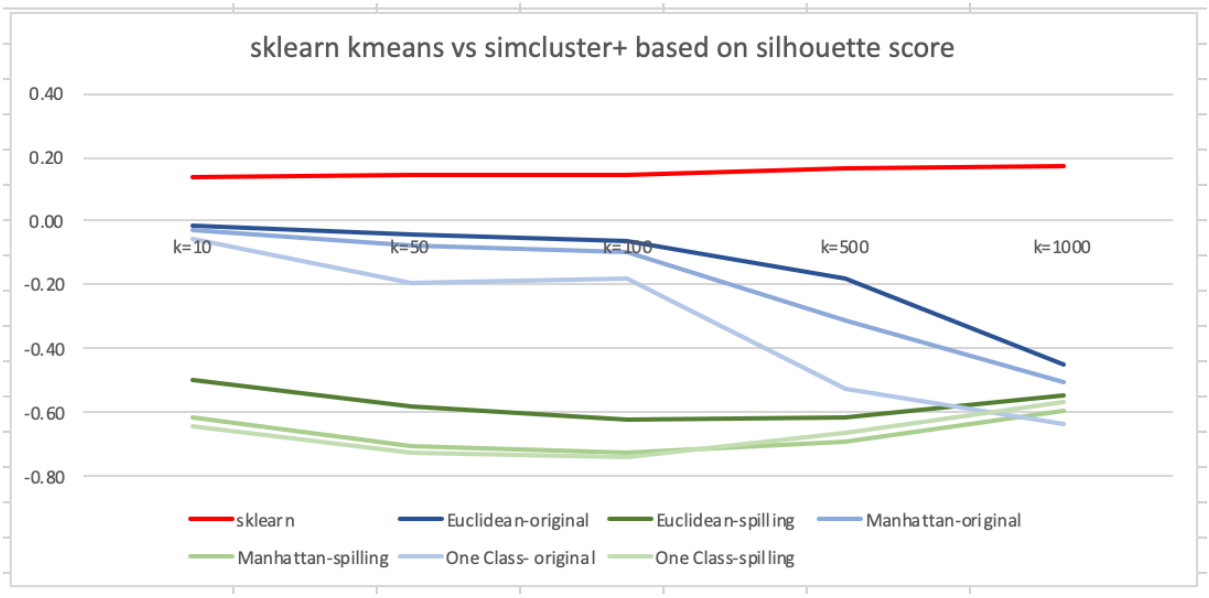
	k=10	k=50	k=100	k=500	k=1000
sklearn	2.387	10.263	19.810	28.105	283.116
Euclidean-original	103.771	177.375	189.199	374.070	669.525
Euclidean-spilling	869.924	1,165.928	1,461.638	2,501.115	3,525.929
Manhattan-original	89.744	95.738	286.189	730.774	1,101.223
Manhattan-spilling	809.511	1,582.206	1,888.195	4,305.151	26,525.207
One Class- original	78.231	135.526	182.957	1,066.166	8,027.365
One Class-spilling	713.028	1,071.536	2,002.625	6,972.527	35,151.433



From the table, we can see that Sklearn always be the fastest. SimCluster+ one-class would take longer than others no matter with spilling or without spilling, especially after k=500.

Performance on silhouette score

Table 3: performances of sklearn kmeans vs simcluster+ based on silhouette score					
	k=10	k=50	k=100	k=500	k=1000
sklearn	0.14	0.14	0.15	0.16	0.17
Euclidean-original	-0.02	-0.04	-0.06	-0.18	-0.45
Euclidean-spilling	-0.50	-0.58	-0.62	-0.62	-0.55
Manhattan-original	-0.02	-0.07	-0.09	-0.31	-0.50
Manhattan-spilling	-0.62	-0.71	-0.73	-0.69	-0.60
One Class- original	-0.05	-0.19	-0.18	-0.53	-0.64
One Class-spilling	-0.64	-0.72	-0.74	-0.66	-0.57



From the table and visualization, we can see that Sklearn always keep the best silhouette score. SimCluster+ methods have bad performance here.

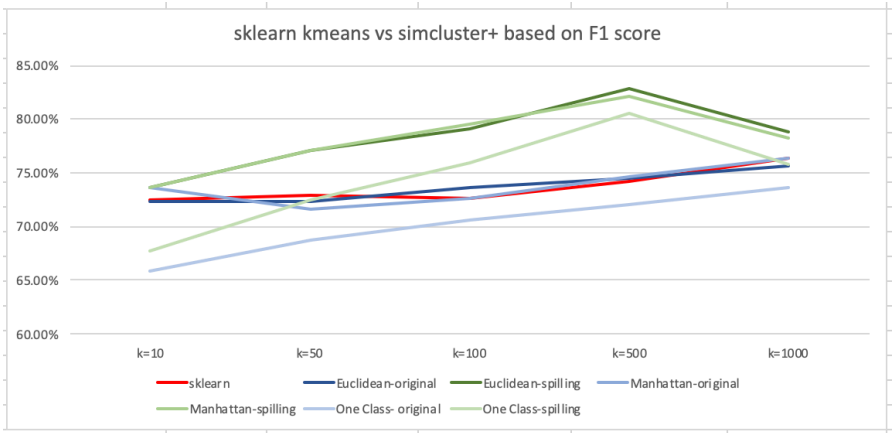
Part 2: SimCluster+ vs. kmeans: apples to oranges with default SimCluster+

Dataset 1: FICO (with NOohc) – balanced dataset

Performance on F1 score

Table 1: performances of sklearn kmeans vs simcluster+ based on F1 score

	k=10	k=50	k=100	k=500	k=1000
sklearn	72.52%	72.90%	72.68%	74.21%	76.40%
Euclidean-original	72.34%	72.39%	73.57%	74.55%	75.71%
Euclidean-spilling	73.68%	77.04%	79.07%	82.86%	78.75%
Manhattan-original	73.64%	71.59%	72.61%	74.69%	76.35%
Manhattan-spilling	73.61%	77.09%	79.54%	82.10%	78.25%
One Class- original	65.82%	68.71%	70.58%	72.01%	73.68%
One Class-spilling	67.78%	72.46%	75.86%	80.53%	75.78%



There is no big difference on f1 scores when using FICO original dataset or FICO one-hot-encoding dataset, and the whole trends didn't change.

Performances of SimCluster+ with FICO(ohc) dataset is little better than that with FICO(NOohc) dataset(generally +/- 1%)

For the table, If we only compare Sklearn and SimCluster+ Euclidean, then as Gold highlighted, Sklearn little outperform than SimCluster+ with only +/- 0.2% difference which is not bad;

If we compare Sklearn and different distance in SImCluster+ without spilling, as yellow highlighted, other than k=10 and k=500 where SimCluster+ Manhattan without spilling is the best, in other cases, Sklearn is the best.

If we compare Sklearn with all of different distance and range percentile, as orange highlighted, SimCluster+ can always get the best results with spilling on Euclidean or Manhattan distance.

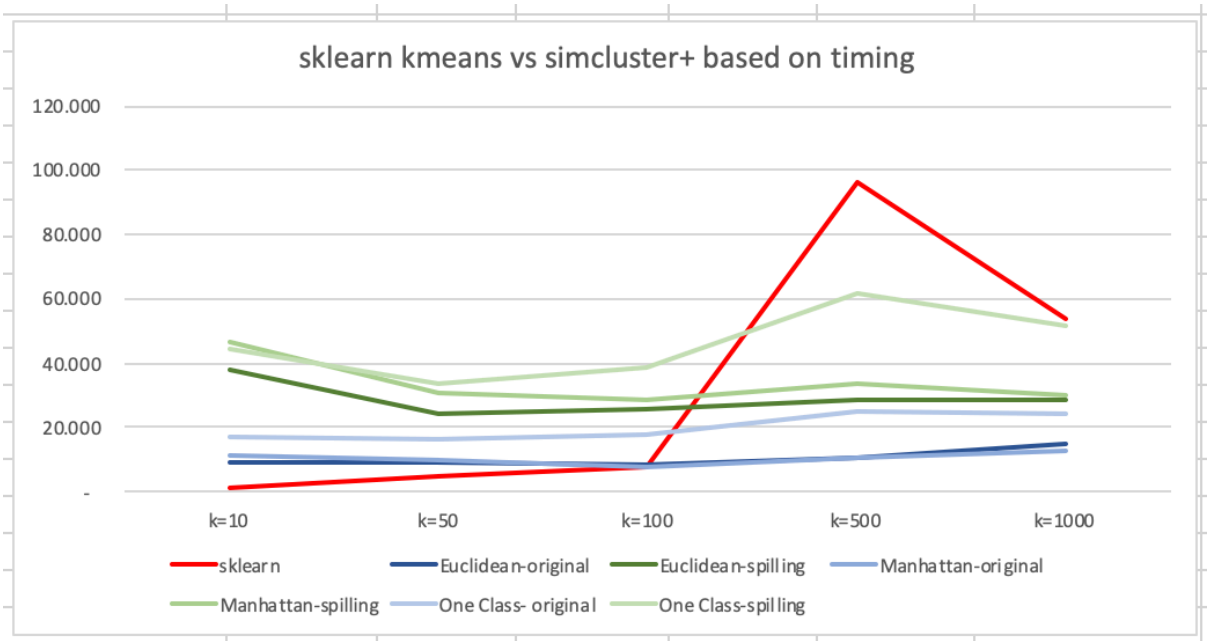
For the visualization, sklearn has similar trend with SimCluster+ Euclidean and Manhattan without spilling excepting one-class, and performances of Euclidean and Manhattan with spilling are all higher than Sklearn, even with one-class with spilling, the performance start to be better than Sklearn after k=50.

Maybe because of the spilling feature, all Spilling performances started to drop after k=500.

Performance on Timing

Table 2: performances of sklearn kmeans vs simcluster+ based on timing

	k=10	k=50	k=100	k=500	k=1000
sklearn	0.967	4.384	7.543	96.53	53.795
Euclidean-original	9.312	9.38	8.033	10.557	14.555
Euclidean-spilling	38.051	23.84	25.978	28.77	28.84
Manhattan-original	11.570	9.724	7.511	10.422	12.507
Manhattan-spilling	46.736	30.946	28.368	33.3	29.969
One Class- original	17.289	16.166	17.554	25.21	24.165
One Class-spilling	44.117	33.454	38.267	61.77	51.632



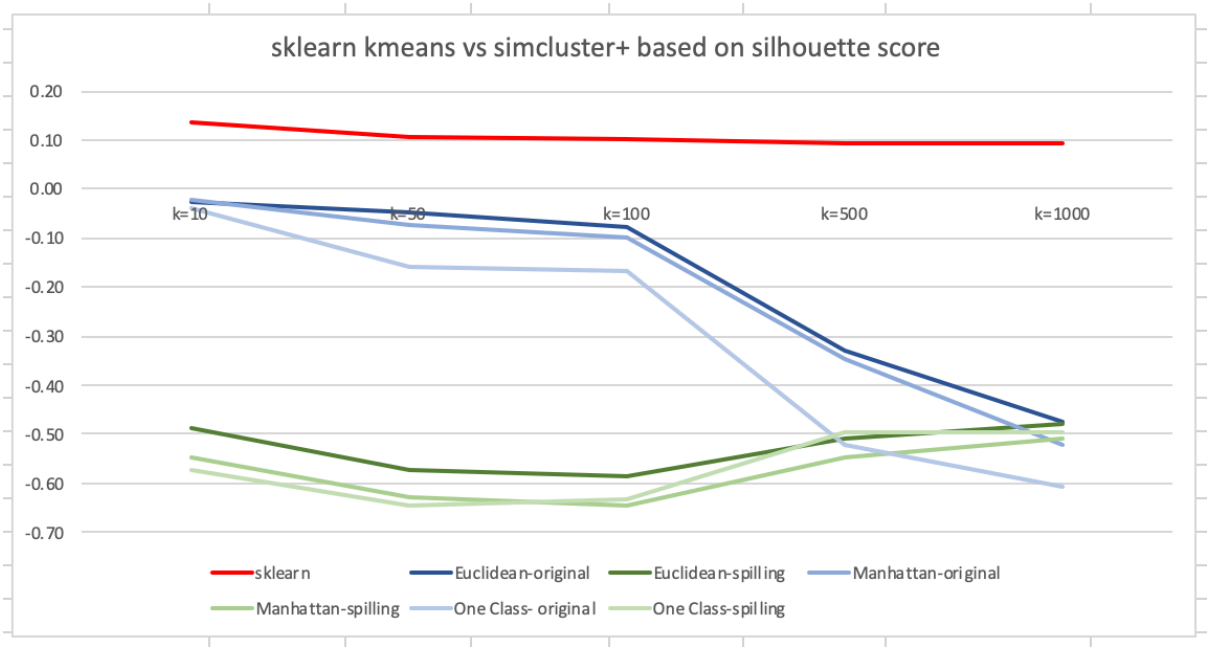
Comparing with timing Performance of SimCluster+ with FICO(with ohc) showed in page 4, timing performance of SimCluster+ with FICO(with NO ohc) showed here is significantly better – about 5-10 times less than before. One of the reason could be iteration times and initialization parameters changed(in the previous experiment, to keep apple to apple, we set these parameters the same with Sklearn kmeans which are larger), another reason would be the time difference of SimCluster+ between dealing with categorical data directly(i.e encoding itself) and dealing with one-hot-encoded categorical data(treated as real actually).

Comparing SimCluster+ performance here with the previous performance of SimCluster+ on FICO(with ohc) dataset, we can see that though a little drop on f1 scores, time reduced a lot, and most important thing is SimCluster+ with spilling can beat Sklearn in either of these experiments.

Performance on silhouette score

Table 3: performances of sklearn kmeans vs simcluster+ based on silhouette score

	k=10	k=50	k=100	k=500	k=1000
sklearn	0.14	0.11	0.10	0.09	0.09
Euclidean-original	-0.02	-0.05	-0.08	-0.33	-0.48
Euclidean-spilling	-0.49	-0.57	-0.59	-0.51	-0.48
Manhattan-original	-0.02	-0.07	-0.10	-0.35	-0.52
Manhattan-spilling	-0.55	-0.63	-0.64	-0.55	-0.51
One Class- original	-0.04	-0.16	-0.17	-0.52	-0.61
One Class-spilling	-0.57	-0.64	-0.63	-0.50	-0.50



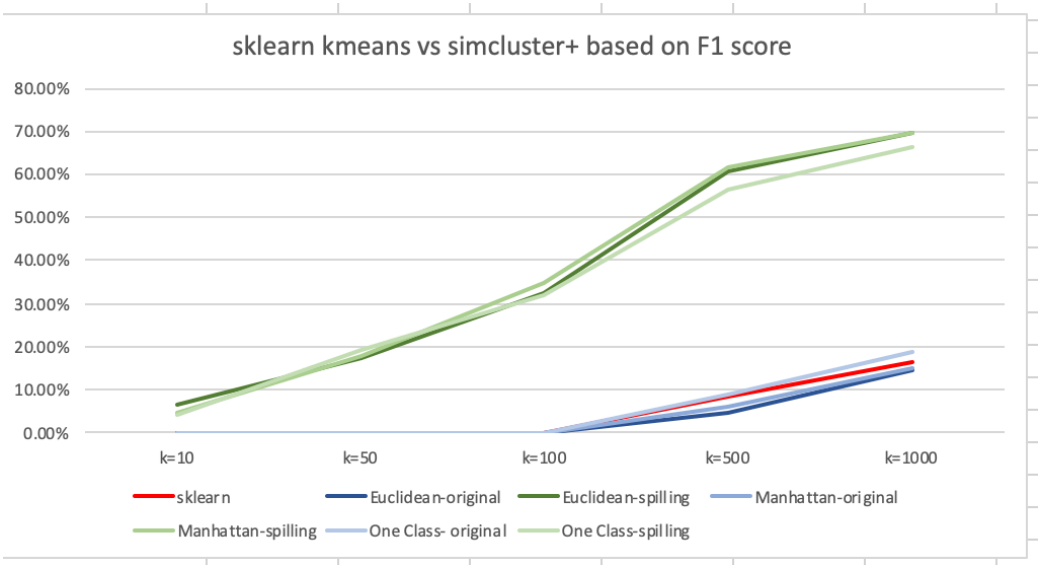
As we can see above, Sklearn kmeans keep having the best silhouette scores. SimCluster+ methods have much worse performances here, and it's even little worse comparing with the previous silhouette scores of SimCluster+ with FICO(with ohc) dataset.

Dataset 2: Default Payment (with NOohc) – unbalanced dataset

Performance on F1 Score

1) When threshold = 0.5

threshold (majority based)					
Table 1: performances of sklearn kmeans vs simcluster+ based on F1 score					
	k=10	k=50	k=100	k=500	k=1000
sklearn	0.00%	0.00%	0.00%	8.40%	16.23%
Euclidean-original	0.00%	0.00%	0.00%	4.47%	14.56%
Euclidean-spilling	6.45%	17.14%	32.46%	60.72%	69.73%
Manhattan-original	0.00%	0.00%	0.00%	5.80%	14.83%
Manhattan-spilling	4.73%	17.66%	34.78%	61.86%	69.75%
One Class- original	0.00%	0.00%	0.00%	8.69%	18.68%
One Class-spilling	4.12%	19.05%	32.04%	56.53%	66.34%



From the table, we can see that If we only compare Sklearn and SimCluster+ Euclidean, then as Gold highlighted, Sklearn little outperform than SimCluster+ Euclidean without spilling when k=500 and k=1000

If we compare Sklearn and different distance in SimCluster+ without spilling, as yellow highlighted, **SimCluster+ one-class** with no spilling is outperform than Sklearn and other methods.

If we compare Sklearn with all of different distance and range percentile, as orange highlighted, SimCluster+ can always get the best results with spilling on Euclidean/ Manhattan/one-class distance.

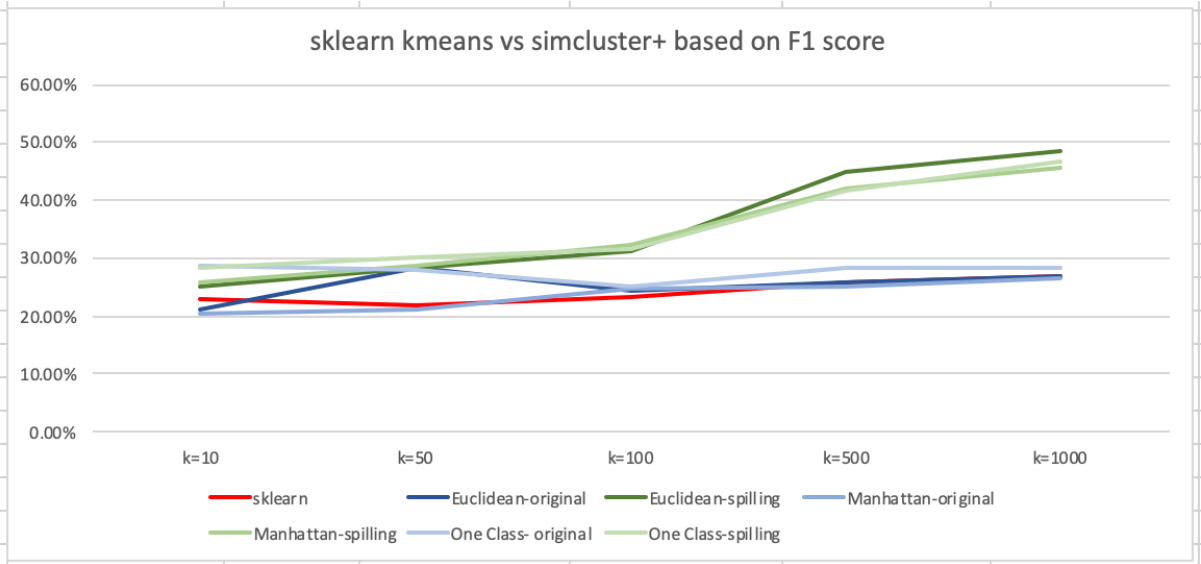
For the visualization, sklearn has similar trend with SimCluster+ Euclidean/Manhattan/ one-class without, and performances of Euclidean/Manhattan/One-class with spilling are all much higher than Sklearn.

Comparing with the previous experiment of SimCLuster+ on Default Payment (with ohc)(aka. Imbalanced dataset), excepting one-class, Euclidean/Manattan have better performance no matter with or without spilling.

For Simcluster+ One-Class, it can have better performance when use Default Payment(without ohc) (aka. Imbalanced dataset), especially with spilling.

2) When threshold = 0.06:

threshold (majority based)					
Table 1: performances of sklearn kmeans vs simcluster+ based on F1 score					
	k=10	k=50	k=100	k=500	k=1000
sklearn	23.08%	21.98%	23.38%	25.63%	26.71%
Euclidean-original	21.25%	28.31%	24.23%	25.68%	26.94%
Euclidean-spilling	25.22%	28.38%	31.24%	44.76%	48.37%
Manhattan-original	20.54%	21.12%	24.77%	25.00%	26.43%
Manhattan-spilling	25.77%	28.68%	32.24%	42.18%	45.49%
One Class- original	28.86%	27.96%	24.99%	28.14%	28.17%
One Class-spilling	28.36%	30.21%	31.45%	41.65%	46.86%



From the table, we can see that if we only compare Sklearn with SimCluster+ Euclidean without spilling, excepting k=10, SimCluster+ Euclidean without spilling beats Sklearn kmeans in all other situations.

If we compare Sklearn with SimCluster+ Euclidean/Manhattan/One-class without spilling, we can see that the result is the same with above, which is, SimCluster+ beats Sklearn in most of cases.

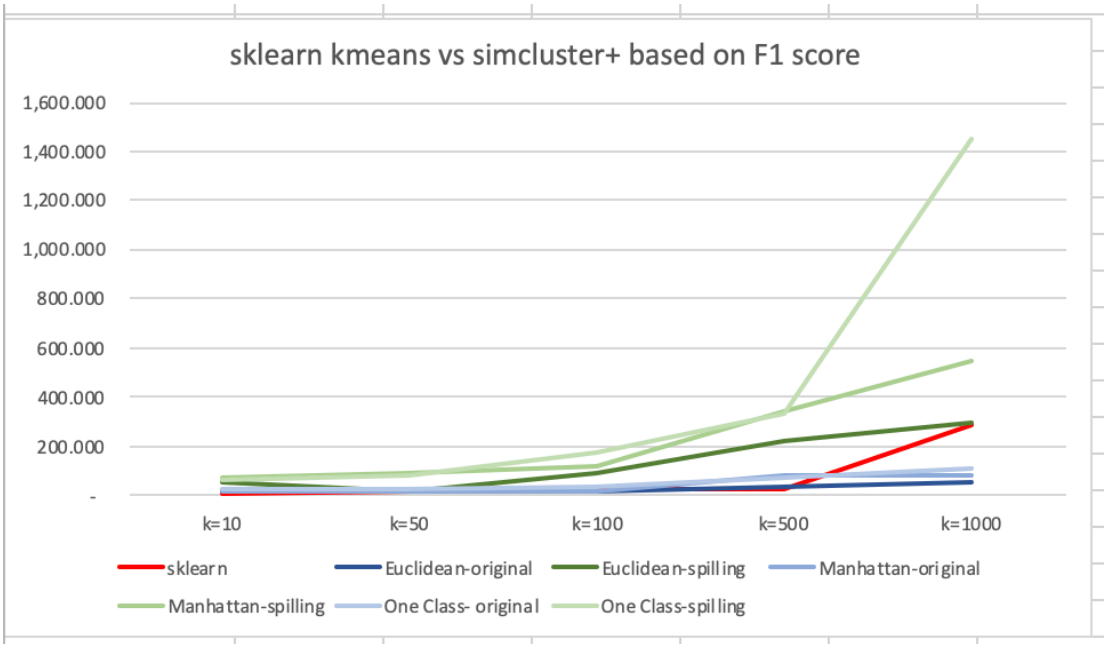
If we compare Sklearn with all SimCluster+ methods, we can see that SimCluster+ different distances with spilling give us the best resutls in different k.

Noticeably, one-class with spilling has very good performance.

Comparing with the previous experiment of SimCluster+ using Default Payment(with ohc), all SimCluster+ methods's performance improve a little in generally (+/- 2%)

Performance on Timing

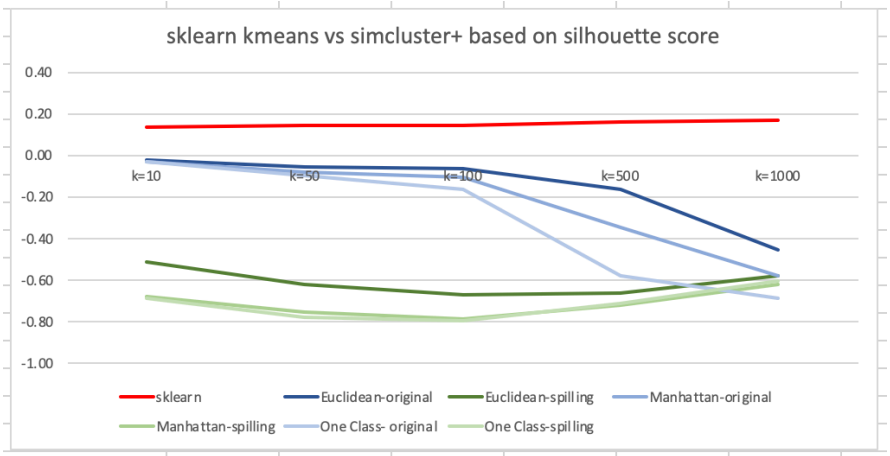
Table 2: performances of sklearn kmeans vs simcluster+ based on timing					
	k=10	k=50	k=100	k=500	k=1000
sklearn	2.387	10.263	19.810	28.105	283.116
Euclidean-original	9.997	10.055	13.623	29.325	50.349
Euclidean-spilling	55.872	13.331	87.375	222.597	292.612
Manhattan-original	10.84	13.331	15.45	76.266	78.87
Manhattan-spilling	68.668	88.420	121.308	341.473	549.701
One Class- original	22.231	28.574	28.808	71.193	105.889
One Class-spilling	65.052	80.748	172.781	332.737	1453.832



We can see that the advantages of Sklearn sklearn is not very obvious here comparing with the experiment we did before with SimCluster+ using Default Payment (with ohc) datasets. Sklearn kmeans would much be much slower at k=500 while the timing of SimCluster+ methods are slowly increase(excepting SimCluster+ spilling) and always keeping fast.

Performance on silhouette score

Table 3: performances of sklearn kmeans vs simcluster+ based on silhouette score					
	k=10	k=50	k=100	k=500	k=1000
sklearn	0.14	0.14	0.15	0.16	0.17
Euclidean-original	-0.03	-0.06	-0.07	-0.16	-0.45
Euclidean-spilling	-0.51	-0.62	-0.67	-0.66	-0.58
Manhattan-original	-0.03	-0.08	-0.11	-0.35	-0.58
Manhattan-spilling	-0.68	-0.75	-0.78	-0.72	-0.62
One Class- original	-0.03	-0.09	-0.16	-0.57	-0.68
One Class-spilling	-0.69	-0.78	-0.79	-0.71	-0.60



Like what we see before, Sklearn still keep higher Silhouette scores.

And comparing SimCluster+ in DP(with ohc) with SimCLuster+ in DP(without ohc), silhouette scores be even little worse.

Additional part: SimCluster vs. Sklearn on visualization

To get a visualized idea about the difference between sklearn kmeans and SimCluster+ Euclidean, I used a 2D datasets with different shaped clusters. Details about the datasets:

Dataset 3: Clusterable data with different shapes and ground truths					
shape	(2309,2)				
n_clusters	7				
intro	6 different shaped clusters(e.g circular, linear) with 1 cluster for outliers (see figure 1)				

Statistics results of Sklearn kmeans and SimCluster+:

	sklearn	workbench					
		Euclidean		Manhattan		One class	
		original	spilling	original	spilling	original	spilling
homoegneity	0.537	0.494	0.609	0.491	0.597	0.424	0.502
completeness	0.571	0.527	0.546	0.524	0.535	0.515	0.498
v measure score	0.554	0.51	0.576	0.507	0.564	0.465	0.5

Visualization of correct clustering results:

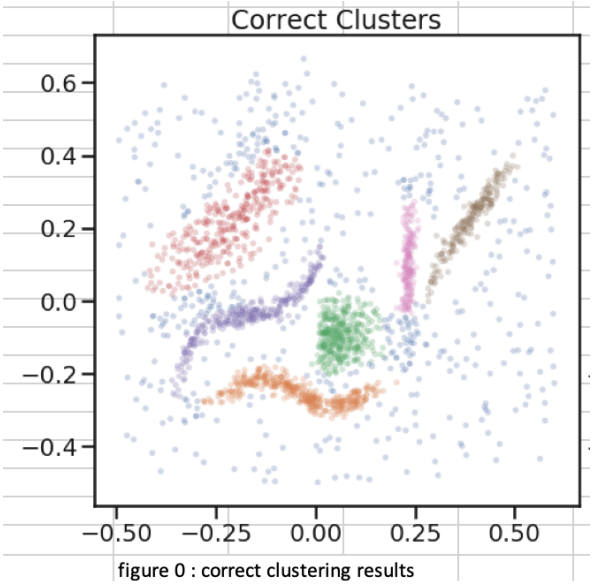
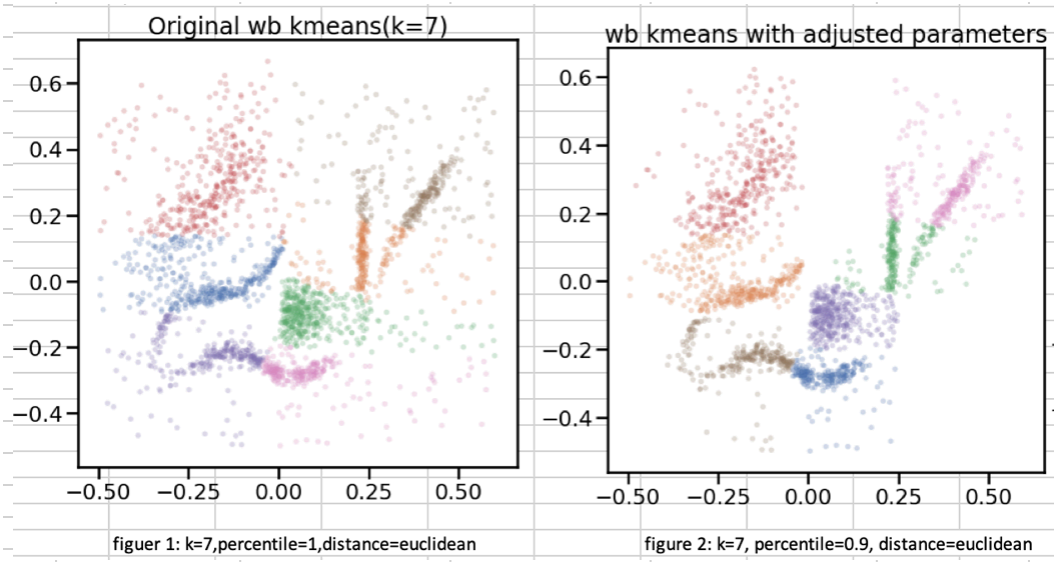


figure 0 : correct clustering results

Visualization of SimCluster+ and sklearn:



figuer 1: k=7,percentile=1,distance=euclidean

figure 2: k=7, percentile=0.9, distance=euclidean

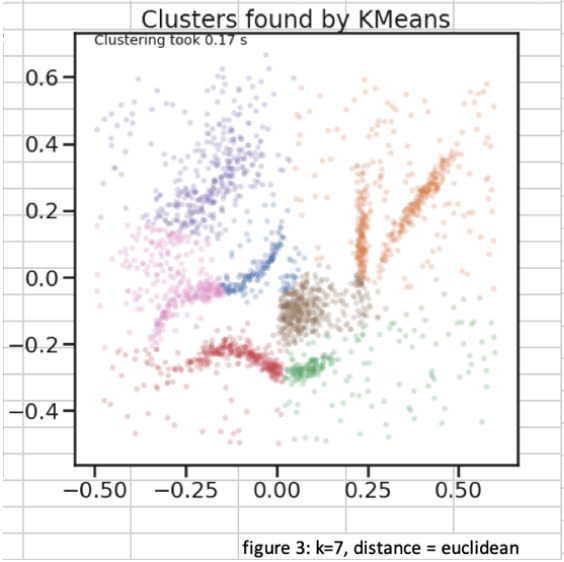


figure 3: k=7, distance = euclidean

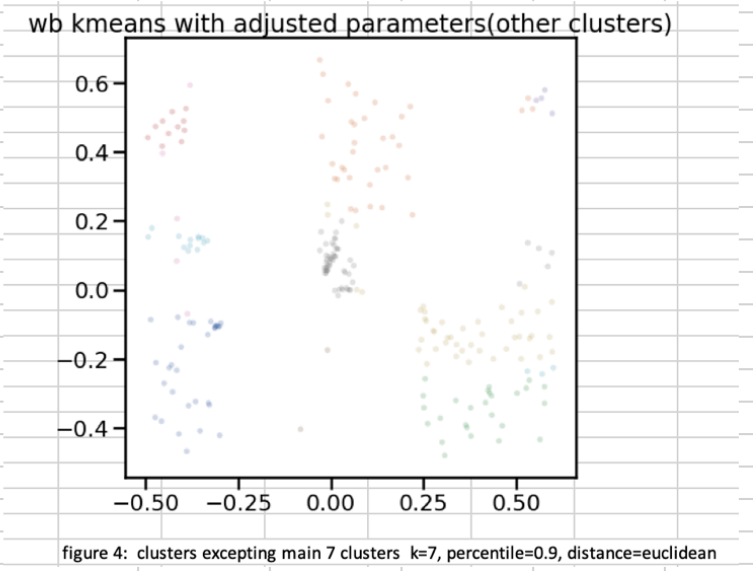


figure 4: clusters excepting main 7 clusters k=7, percentile=0.9, distance=euclidean

from the table, we know that SimCluster+ Euclidean with spilling have the best performance(i.e highest v measure score), so here I posted the visualization of it.

Figure 0 shows the correct clustering result.

Comparing figure 1 with figure 2, we can clear see the spilling effect on the figure 2, which shows that SimCluster+ indeed exclude the outlisers, to better shows it, I made figure 4 especially for those excluded outliers.

Figure 3 is made by sklearn kmeans, we can see that SimCluster+ Euclidean is visually different from SKlearn, they clustered different in chunks. The mainly difference is the way to cluster the right two line-shaped clusters -- sklearn treat them the same while SimCluster+ Euclidean split them into two clusters though they look not correct but at least it indicates that we relatively better recognized clusters.