As we discussed, I add locality regularizer to the original optimization problem(L1). Then the problem could be expressed as below:

$$min_\beta \frac{1}{2}||y - X\beta||_2^2 + \sum_{i=1}^{n} ||\beta_i||_1$$

$$s.t. \begin{cases} \beta_i = 0, if\ x_i\ is\ far\ from\ y \\ otherwise \end{cases}$$

This method is basically a local L1 regression. There exists two ways to do that. The first is to choose points inside a ε −neighborhood of target point. The second is to choose its k nearest neighbors and do L1 regression. I do the second way. This method is inspired from knn but improves it. I will describe the method in detail.

Parameters: α: parameter for L1 regressor

K: number of regressions we choose to do on data points

K1: number of neighbors we choose to do regression on

K2: number of coefficients we choose from results of regression

Notes about parameters:

1. Parameter K is inspired from OWL regression to accelerate the algorithm, for we do not have to do regression on every points .
2. I consider two selection process in this method. The first is k1 which decide the neighbors we would like to do regression. Then we obtain coefficients for these points and comes the second selection k2. I select k2 largest coefficients and put them into similarity matrix while others are 0. The second selection is taken into consideration because I found if we set k1 to be large, many points would be connected which introduced error. But if k1 is too small, the effect would be like that in knn which decrease the effect from L1 regression. Thus we have to set k1 a little bit larger and k2 smaller to restrict the actual number of connections. And this method indeed works.

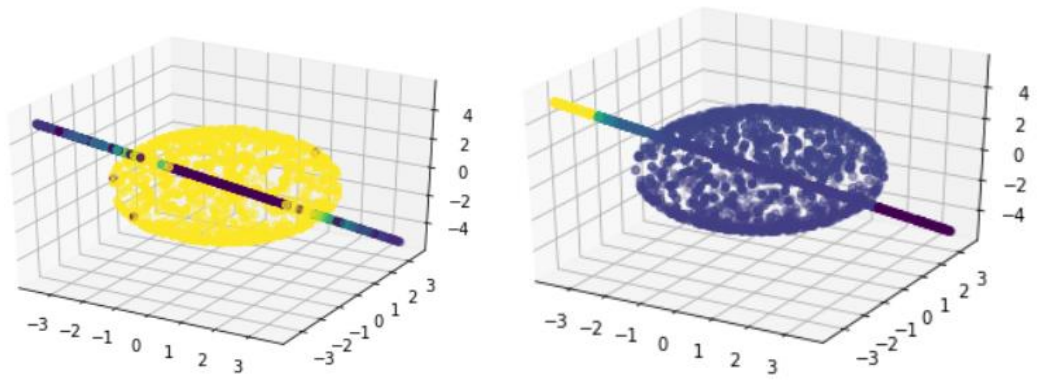
The algorithm is as below:

```
Input: A set of data points $X \in R^{n\times N}$, $K \in \{1, \ldots, N\}$, k1, k2, $\propto$
1.Initialtize $B = 0_{N\times N}$
2.For i in {1,…,K}
3.   Randomly select a index $j_i$ from [N]
4.   Choose k1 nearest neighbors $X_{ji}$ of this index
5.   Obtain $\hat{\beta}$ by regressing $y = X_{ji}$ using L1 regression
6    Select k2 largest coefficients from $\hat{\beta}$ and put them on the corresponding position in $B_{\cdot j_i}$
while other position put 0.
7.   Apply spectral clustering on |B| and obtain partition. Usually the second smallest
eigenvector is enough.
```

Output: Cluster labels.

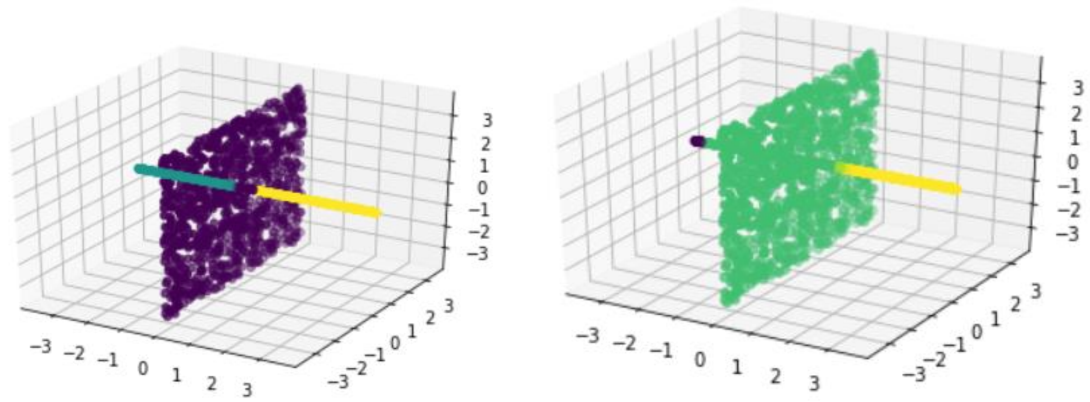
The results are quite good. I try several models and compare results of this method with that from simple Knn. All plots are from the second smallest eigenvector of Laplacian of matrix B.

1. A sphere and a line



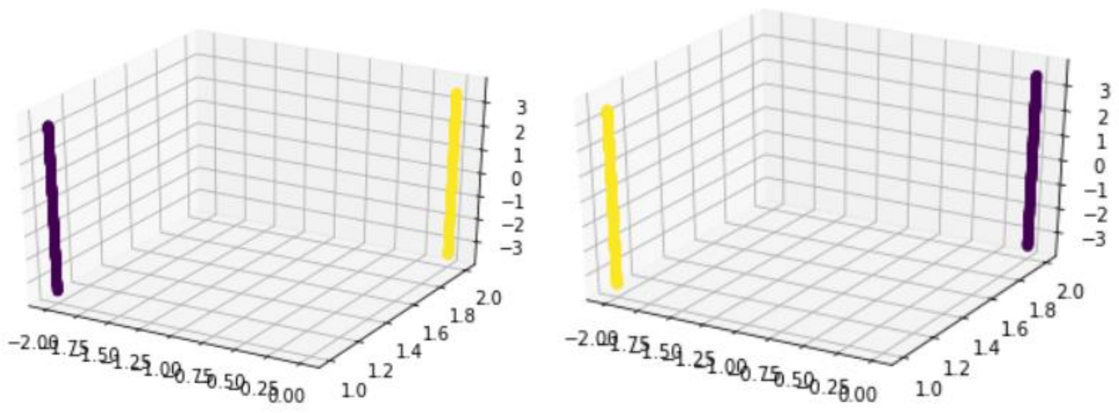
Left from new method while right from KNN and the following results all show in this order. It is obvious, new method almost separate the line and sphere, but the simple KNN works bad.

2. A line and a vertical plane



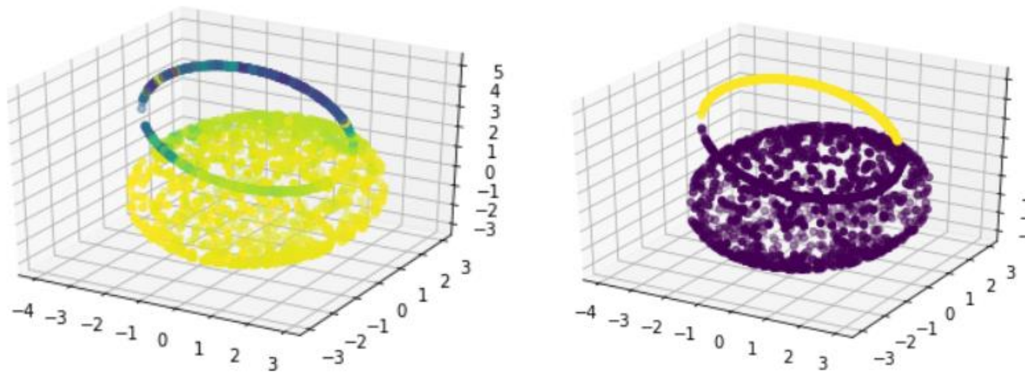
The new method could distinguish the plane though the line is split into two parts but it is still easy to combine them together to obtain the complete line. The KNN works worse.

3. Two separate lines



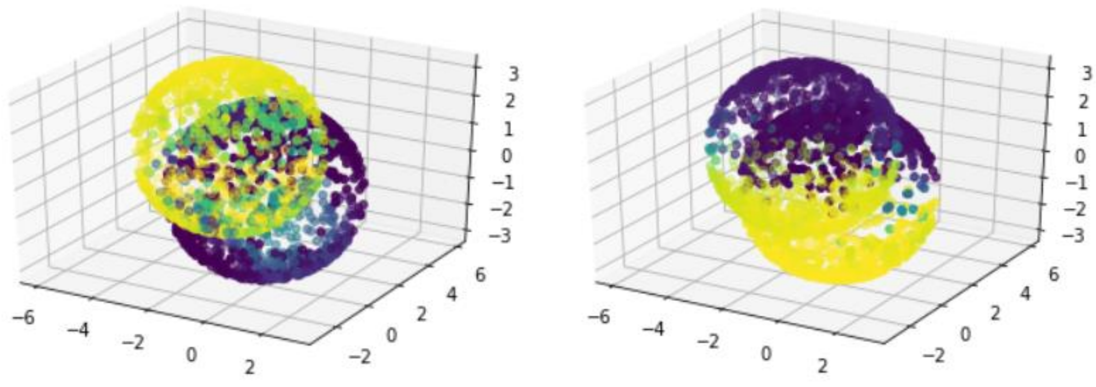
Two methods both separate two lines which means the new method might inherit properties from KNN.

4. A sphere and a circle



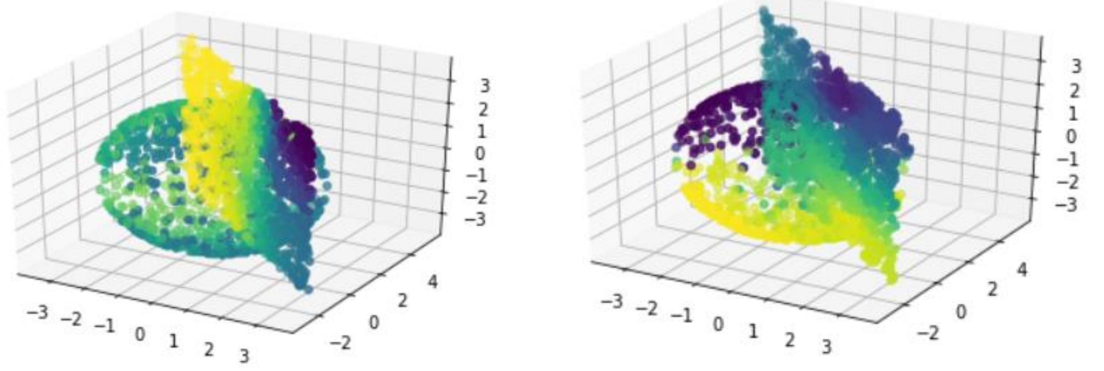
We could discover an obvious circular figure in the first plot and the sphere is of same color everywhere while the second only shows some part.

5. Two spheres



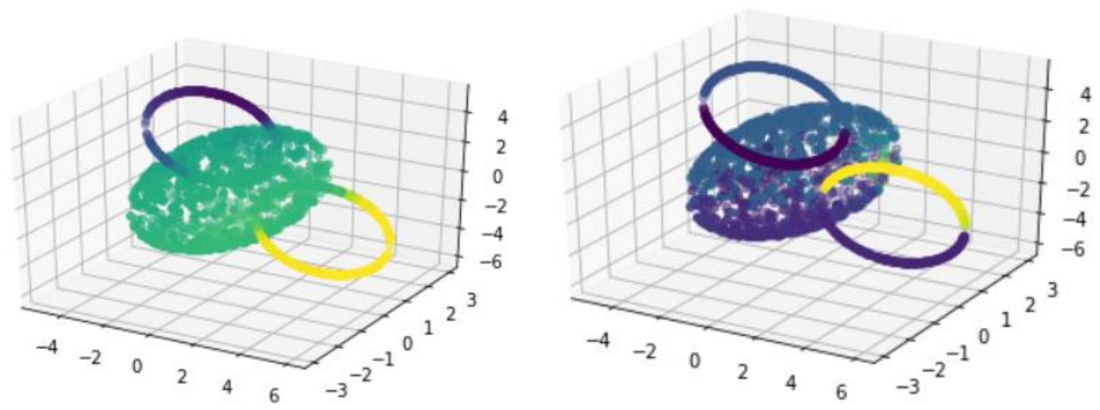
The new method distinguish two figures completely which seems to be a perfect result.

6. A sphere and a plane



We could discover a whole sphere in the first plot. And more adjustment of parameters should be done.

7. A sphere and two circles



In the first plot, we could observe 3 figures which improves a lot from the second plot.

To sum up, this new method indeed improve the performance of simple KNN and L1 regression on subspace clustering.