Case Study 3: Visualizations

Machine Learning

Molin Liu
University of Glasgow
School of Computing Science
Glasgow, Scotland
24736861@student.gla.ac.uk

***Abstract*—This is a report for Machine Learning case study 3, it introduces how we altered the parameters using visualization, and the conclusion we made from our experiments.**

***Index Terms*—audio classification, visualization, hyperparameter**

I. Introduction

In this report we explore the data before we start training the model by using data visualization methods. The problem we try to solve is a audio classification task.

II. Methods

In this case study, we were required to use visualization methods to tuning the parameters. We can only improve our classifier performance by altering the feature extraction parameters.

For the reason above, this report will try to change the parameters respectively, to find out the relationship with the classifier performance.

We also use grid search on the parameters setting, to find out the difference between the best parameter, and the worst one on 5 visualization methods we use in this report.

A. Visualization Methods

In this report, we use PCA, LLE, ISOMAP, tSNE and UMAP to visualize the retrieved data. To find out the impact of these different parameters, we set a series of experiments to test every single parameters and control the rest to be consistent.

Since we use sliding windows to generate the feature vector, the size of window and the size of step are not purely independent variables, that is to say, overlapping is also a critical factor in our experiments. So we will test these 2 parameters together.

The report will organized as below:

- First we use experiments to find out the impact of the parameters;

- then we try to use the conclusion to tuning the parameters to get a good performance;

- performance analysis with the ideal parameter setting generated by grid search;

- test our conclusion.

The following part will illustrate the design of our experiments.

B. Experiments Design

First, we try to find out the impact that overlap exert to the result, so we fix the size of window and other parameters, and set different overlap level to see what will happen.

TABLE I
Parameter setting for overlapping

<table>
<tr><th>id</th><th>size</th><th>step</th><th>over-lap</th><th>deci-mate</th><th>feature range</th><th>window -fn</th><th>feature -fn</th></tr>
<tr><td>1</td><td rowspan="4">1024</td><td>102</td><td>0.1</td><td rowspan="4">1</td><td rowspan="4">(0.0, 1.0)</td><td rowspan="4">boxcar</td><td rowspan="4">cep-strum</td></tr>
<tr><td>2</td><td>307</td><td>0.3</td></tr>
<tr><td>3</td><td>512</td><td>0.5</td></tr>
<tr><td>4</td><td>716</td><td>0.7</td></tr>
</table>

Second, we set different values to the rest parameters respectively, to find out their impacts. Due to the length limitation, the details are emmited, but the parameters setting will be shown as follow.

TABLE II
Parameter setting for size

<table>
<tr><th>id</th><th>size</th><th>step</th><th>over-lap</th><th>deci-mate</th><th>feature range</th><th>window -fn</th><th>feature -fn</th></tr>
<tr><td>1</td><td>1000</td><td>50</td><td rowspan="4">0.05</td><td rowspan="4">1</td><td rowspan="4">(0.0, 1.0)</td><td rowspan="4">boxcar</td><td rowspan="4">cepstrum</td></tr>
<tr><td>2</td><td>3000</td><td>150</td></tr>
<tr><td>3</td><td>5000</td><td>250</td></tr>
<tr><td>4</td><td>7000</td><td>350</td></tr>
</table>

TABLE III
Parameter setting for decimate

<table>
<tr><th>id</th><th>size</th><th>step</th><th>over-lap</th><th>deci-mate</th><th>feature range</th><th>window -fn</th><th>feature -fn</th></tr>
<tr><td>1</td><td rowspan="4">4096</td><td rowspan="4">409</td><td rowspan="4">0.1</td><td>1</td><td rowspan="4">(0.0, 1.0)</td><td rowspan="4">boxcar</td><td rowspan="4">cep-strum</td></tr>
<tr><td>2</td><td>2</td></tr>
<tr><td>3</td><td>3</td></tr>
<tr><td>4</td><td>4</td></tr>
</table>

III. Results

From the experiments result, we find the points sometimes mixing together, and sometime lossing order. Aggregation means the features are too close to classify. In the other hand, sparse points implies that the feature extraction method may not be able to get the traditional feature of a class.

TABLE IV
PARAMETER SETTING FOR WINDOW FN

id	size	step	over-lap	deci-mate	feature range	window -fn	feature -fn
1	4096	409	0.1	1	(0.0, 1.0)	boxcar	cep-strum
2						hamming	
3						hann	
4						black-manharris	

TABLE V
PARAMETER SETTING FOR FEATURE FN

id	size	step	over-lap	deci-mate	feature range	window -fn	feature -fn
1	4096	409	0.1	1	(0.0, 1.0)	boxcar	cepstrum
2							dct
3							dct_phase
4							fft

A. *Overlapping*

Fig. 2 shows the *PCA* with fixed window size, but different overlap levels. From this figure, we can find that the distribution tends to be sparse with the increasement of overlapping levels. Fig. 3 shows the same tendency, the points are gradually loss order. The same situations appear in Fig. 4.

As for Fig. 5 and Fig. 6, which are little bit different from the figures above, their first sub-figures have the best performance, in other words, the points have been seperated more clearly than other 3 sub-figures.

B. *Size of Window*

In this experiment, we set different sizes of windows in order to get some conlusions from the figures. Fig. 7, Fig. 8 and Fig. 10 show that with the growth of the size, the points are becoming sparse. While we should notice that, when the size is too small, the points are aggregative. So the mid-point would be the best choice.

C. *Feature Fn*

The *PCA* figure Fig. 12 is significantly different from the previous *PCA* figure. These 6 different feature fns has shown many interesting features. The *DCT* and fft seem to be the same, which divide different classes into several part very well. The *FFT_phase* and *DCT_phase* gather the purple class so well, while as for the rest classes, they were all mixed together. The cepstrum is very common comparing to those 4 previous figures. The raw feature_fn is so aggregative to make it hardly seperatable.

The exciting part of this experiment is in *tSNE* and *UMAP*. In Fig. 15, even though there are still some overlap on the green and orange, the *feature_fn DCT* and *FFT* seperate the classes very well. And in Fig. 6, the *DCT* and *FFT* still have a great performance comparing to the rest methods. If permitted, apply classifier on *UMAP* or *tSNE*, the result would have be very impressive.

D. *Window Fn*

In *Window_fn* experiment, from the *PCA* Fig. 17 *LLE* Fig. 18, boxcar has a better performance over other 3 methods, while in other figure, these 4 methods tangle together.

E. *Grid Search*

We apply *Grid Search* to find the best parameters setting in the specific feasible region. Meanwhile, we set up a series of parameter settings and the score distribution in Fig. 1.

F. *Performance Analysis*

From the previous experiments, we have the direction to alter the parameters. Here we simply give the result and then compare it to other parameter setting in Fig. 1.

The parameter we alter from the experiment is shown in Table. VI

TABLE VI
TONED PARAMETER SETTING

size	step	over-lap	deci-mate	feature range	window -fn	feature -fn
4000	200	0.05	1	(0.0, 1.0)	boxcar	dct

The result of our toned parameter shows in Fig. 1.

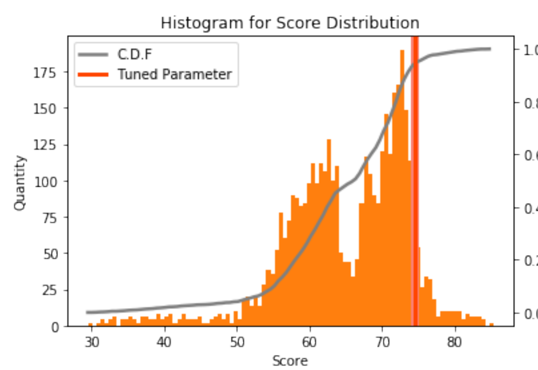

Fig. 1. Distribution of Score
C.D.F

The final score of our tuned parameter is 74.53, which is larger that 94.99% parameters setting.

IV. CONCLUSION

From the experiment results, we can draw the conclusion that, using manifold visualization can help us to have intuitive feelings on high dimension data set.

These figures can not only provide us a perspective of the data, but also can be a feature extraction method, since in some of these figure, the classes have been well seperated, even green region, which is the weak point of our classifier.

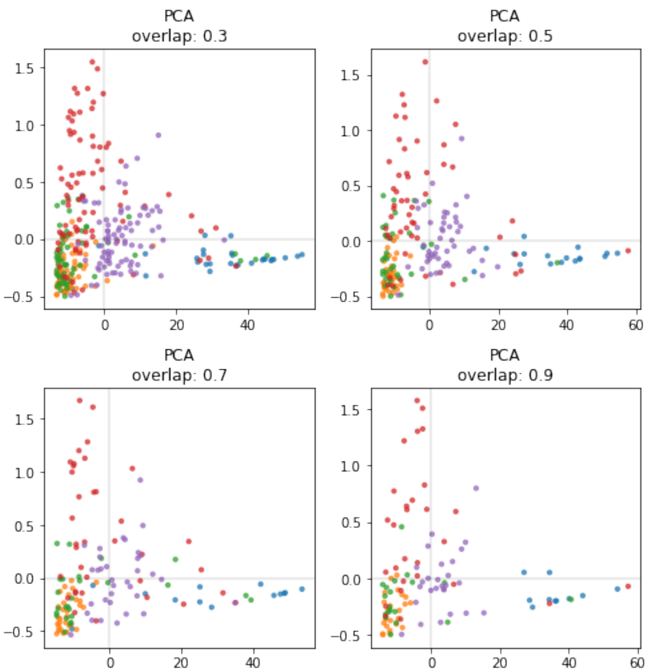

Fig. 2. PCA: Experiment on Overlap
size=1024

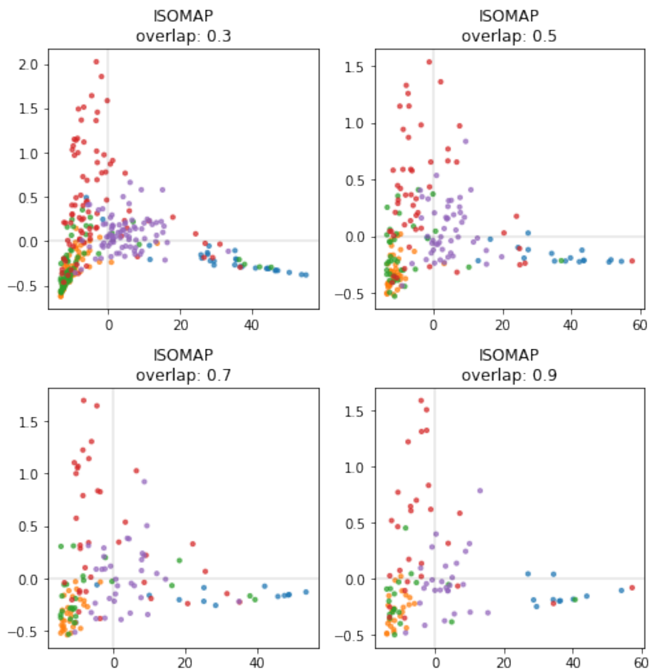

Fig. 4. ISOMAP: Experiment on Overlap
size=1024

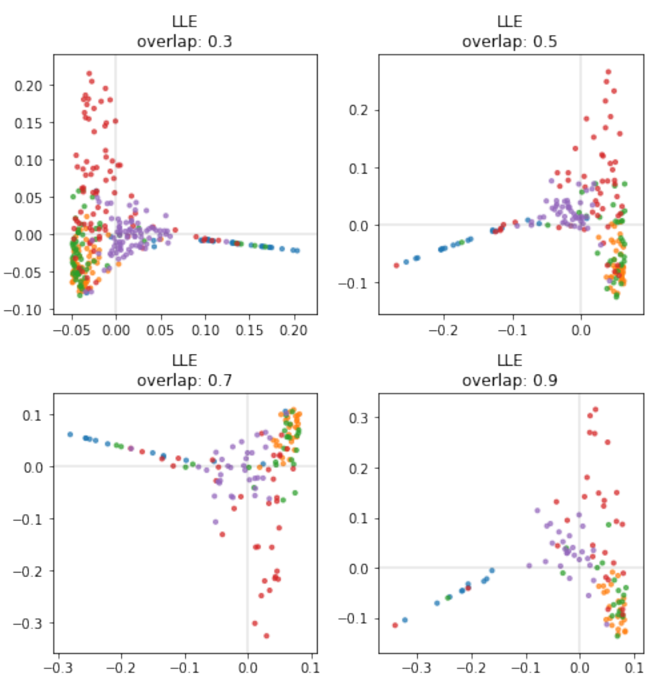

Fig. 3. LLE: Experiment on Overlap
size=1024

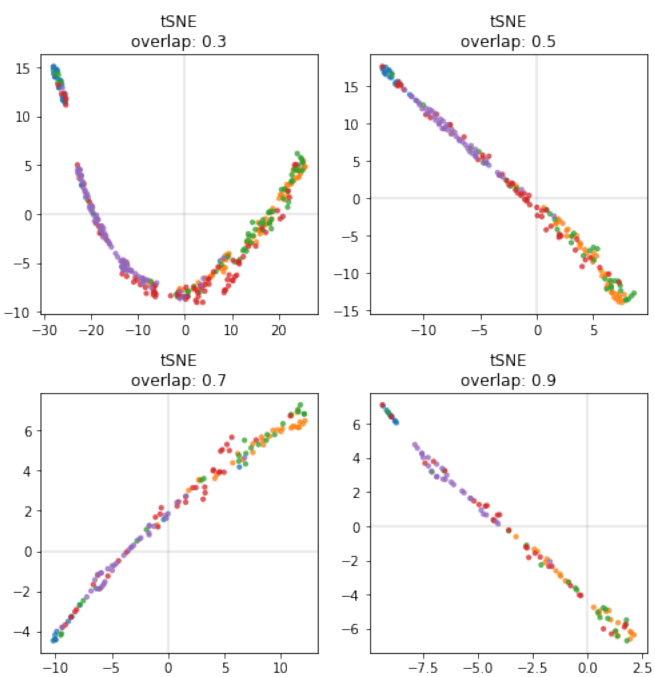

Fig. 5. tSNE: Experiment on Overlap
size=1024

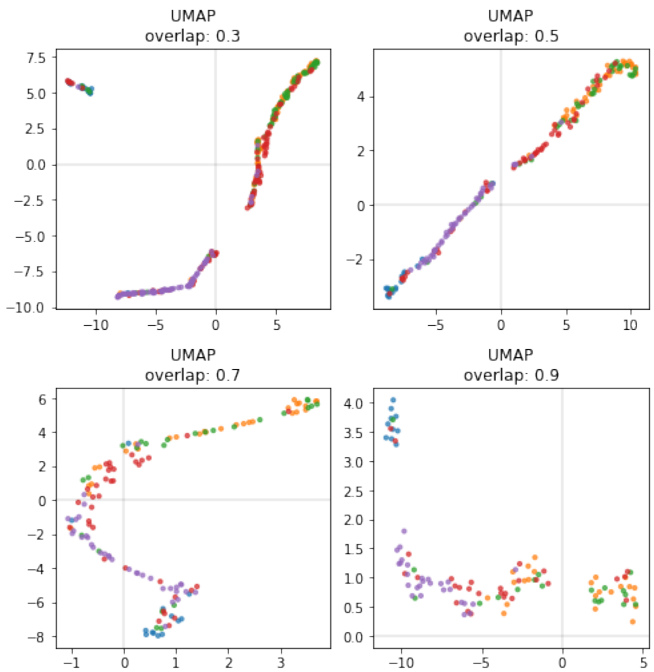

Fig. 6. UMAP: Experiment on Overlap
size=1024

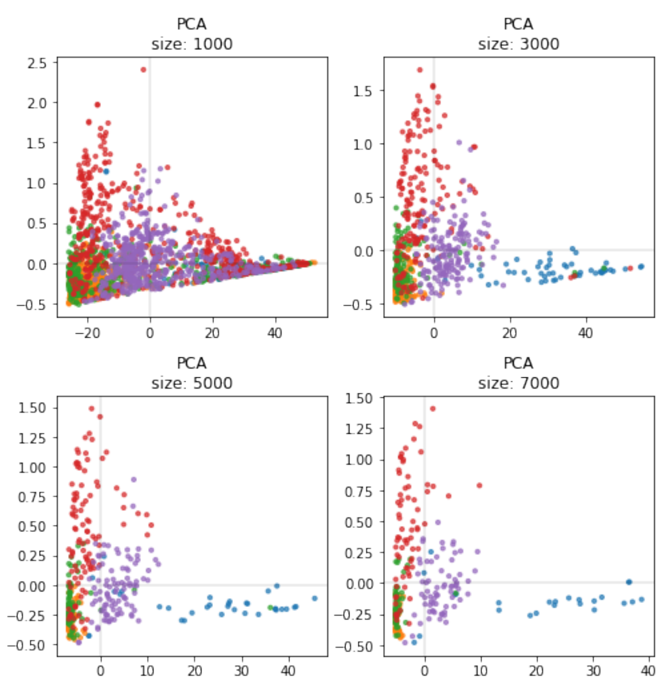

Fig. 7. PCA: Size Experiment
overlap=0.05

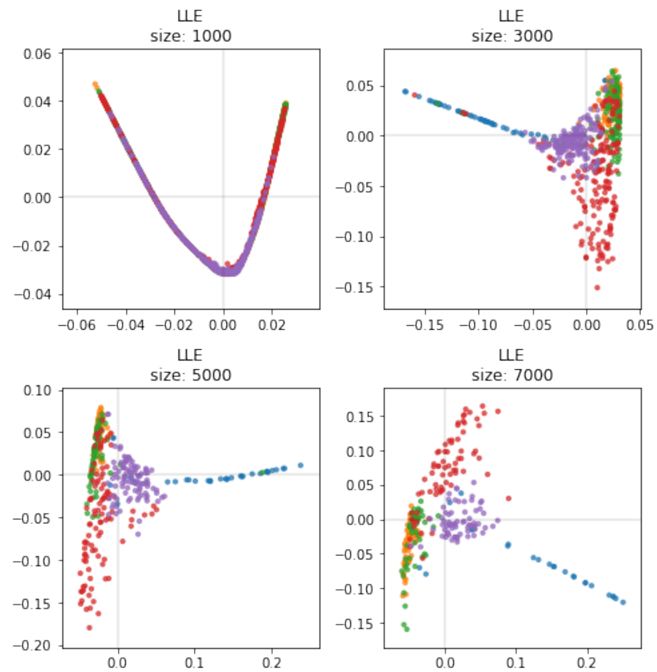

Fig. 8. LLE: Size Experiment
overlap=0.05

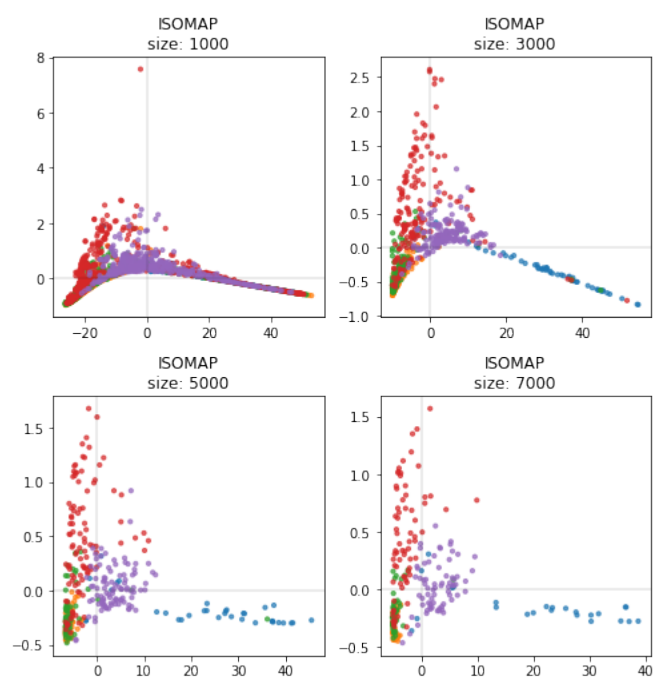

Fig. 9. ISOMAP: Size Experiment
overlap=0.05

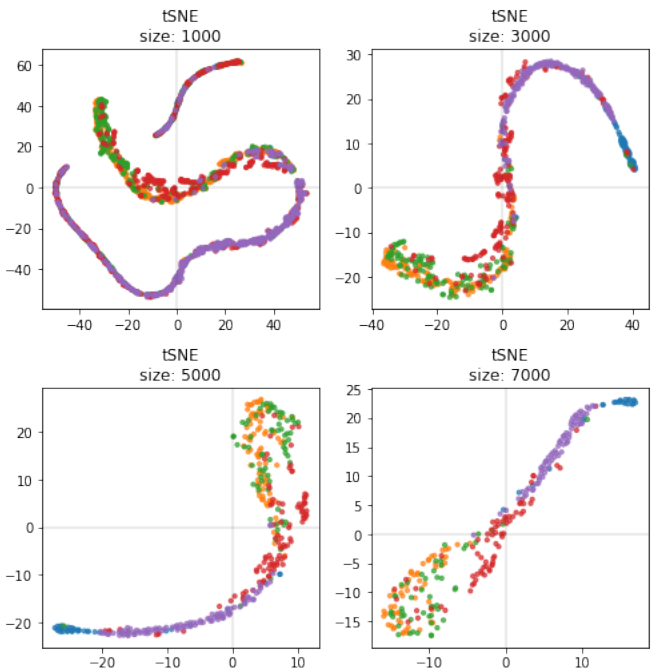

Fig. 10. tSNE: Size Experiment
overlap=0.05

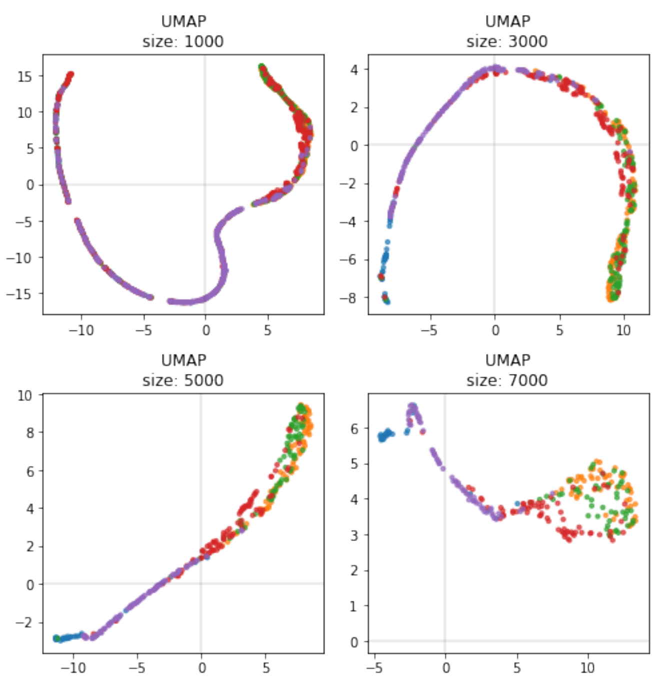

Fig. 11. UMAP: Size Experiment
overlap=0.05

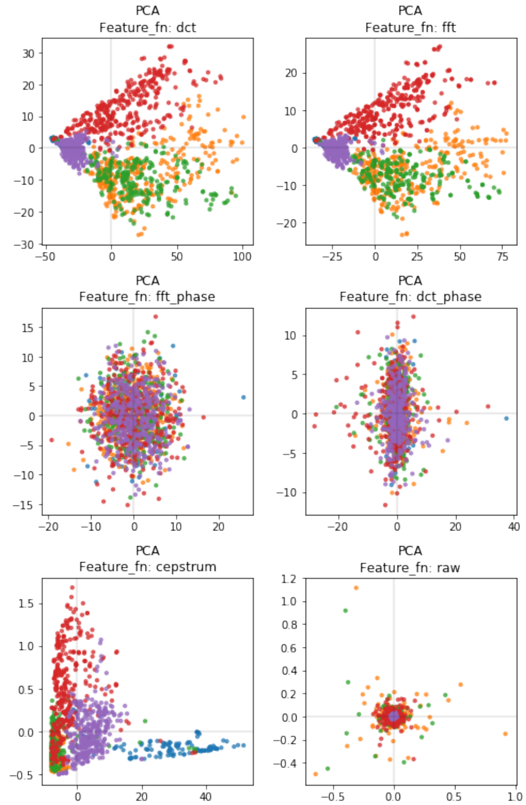

Fig. 12. PCA: Feature fn Experiment

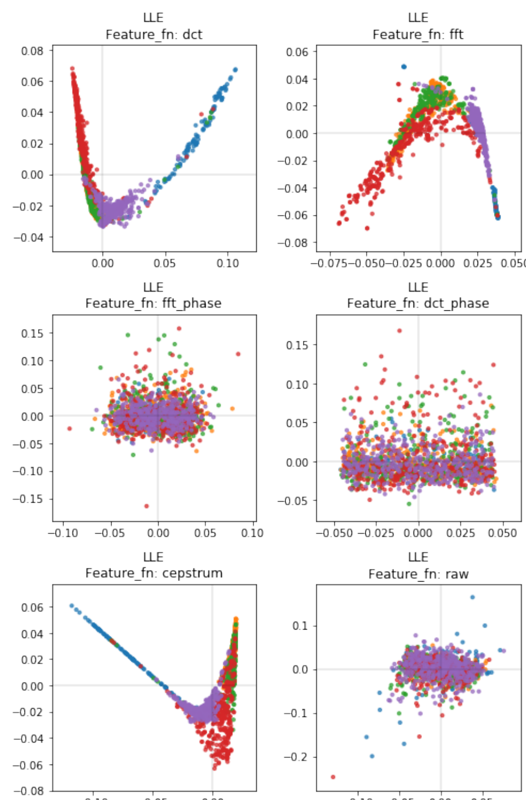

Fig. 13. LLE: Feature fn Experiment

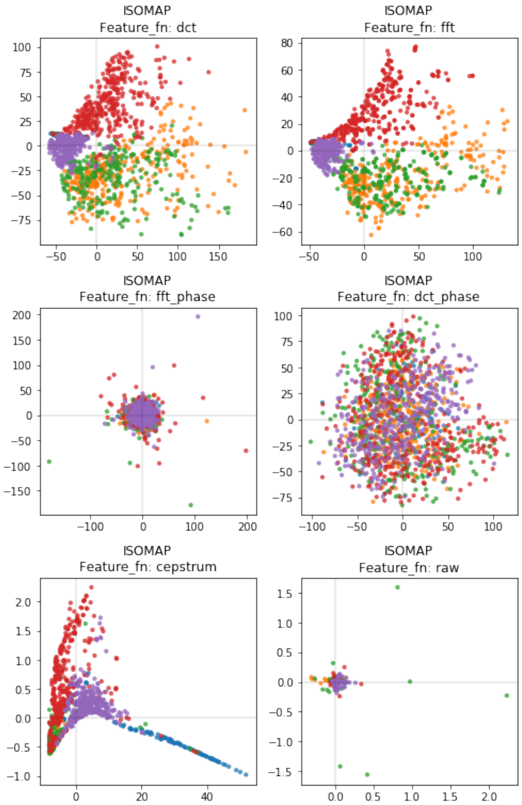

Fig. 14. ISOMAP: Feature fn Experiment

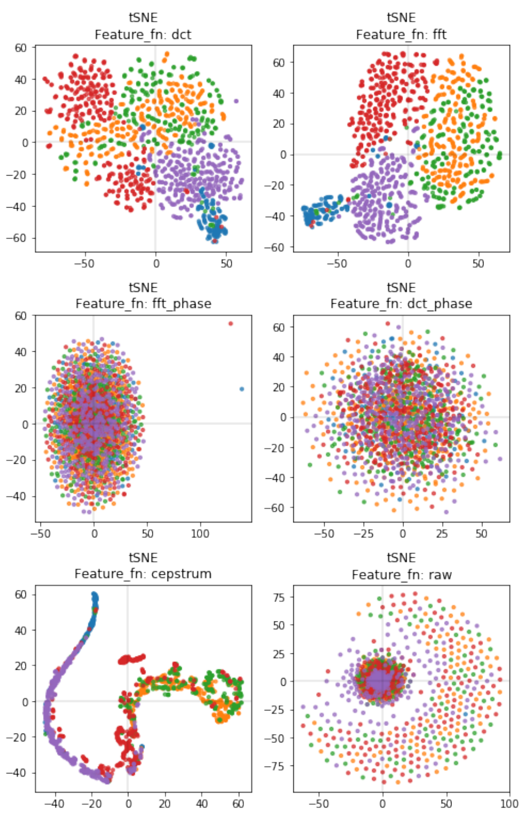

Fig. 15. tSNE: Feature fn Experiment

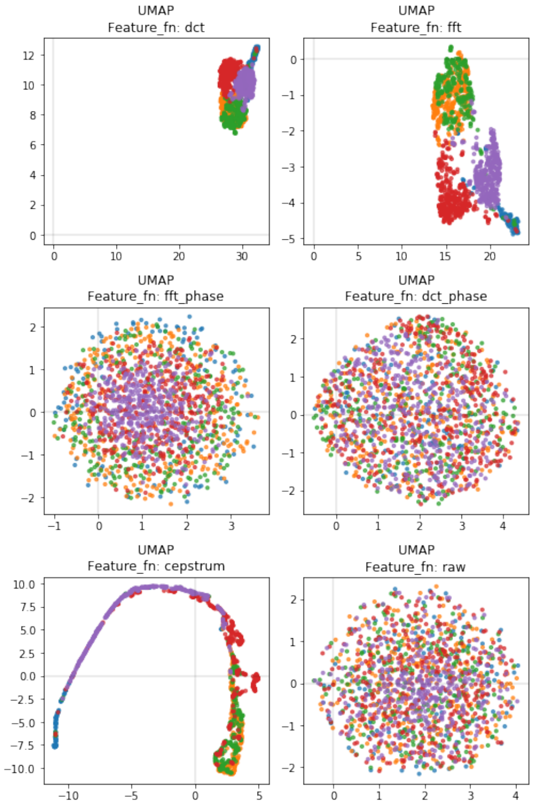

Fig. 16. UMAP: Feature fn Experiment

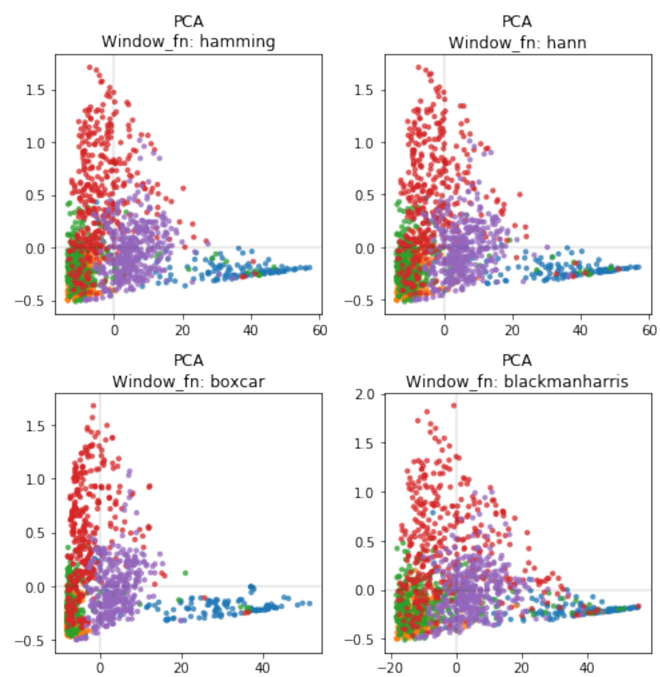

Fig. 17. PCA: Window fn Experiment

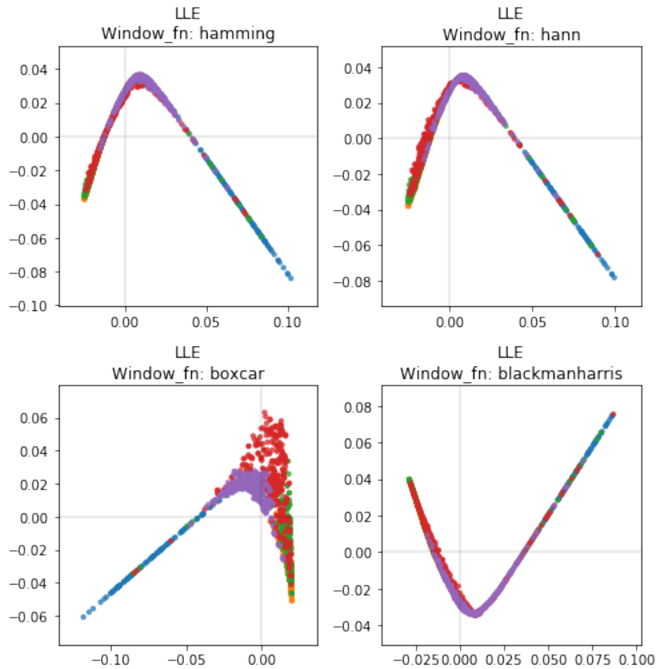

Fig. 18. LLE: Window fn Experiment

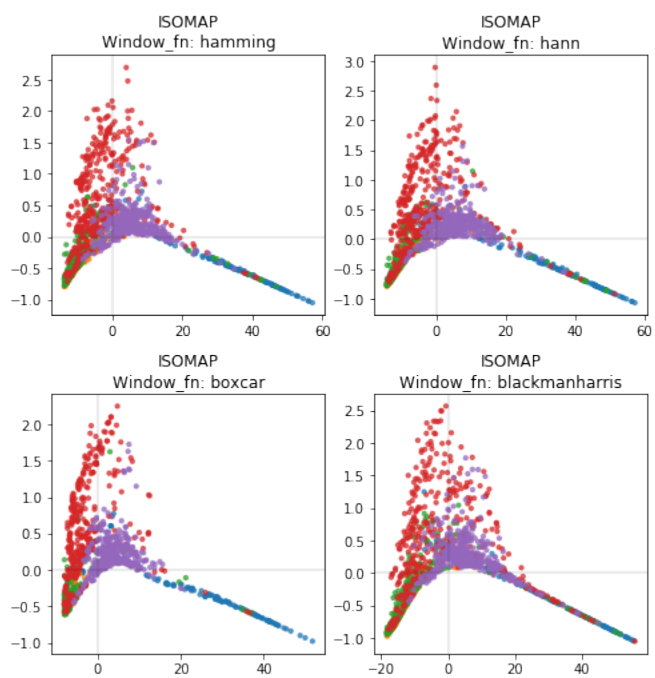

Fig. 19. ISOMAP: Window fn Experiment

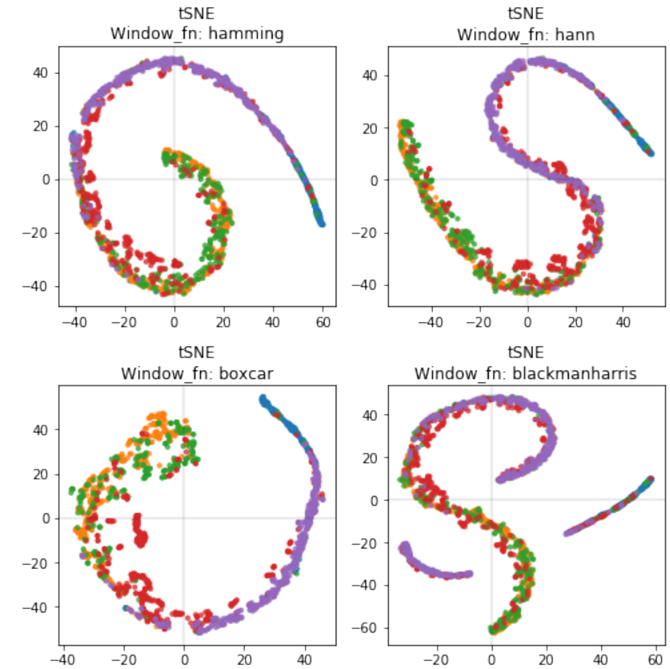

Fig. 20. tSNE: Window fn Experiment

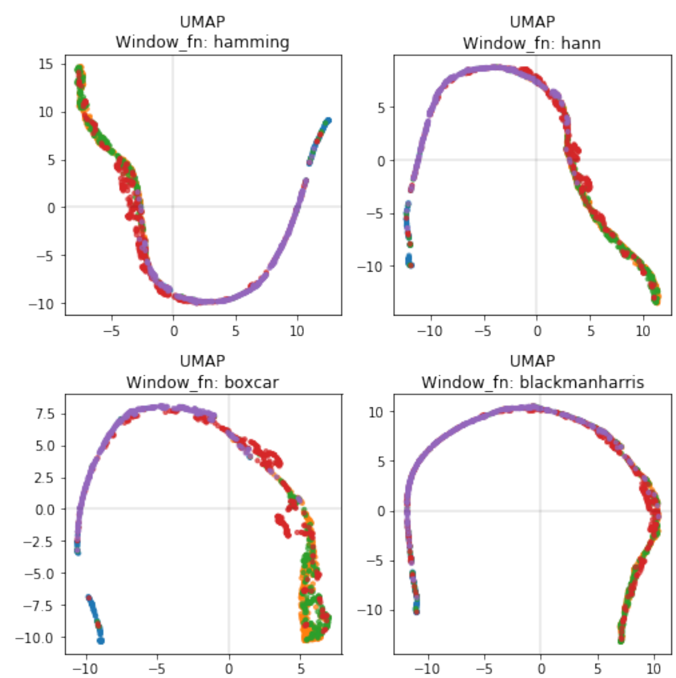

Fig. 21. UMAP: Window fn Experiment