CSC730: Assignment 9
Huh, wonder what's out there?
Active Learning Based Rare Class Discovery

Kristophor Ray Jensen
Electrical Engineering and Computer Science
South Dakota School of Mines and Technology
Rapid City, United States
0009-0001-7344-349X

I. Introduction

The goal of this assignment is to discover all the classes in the provided datasets with as few queries to the oracle as possible, using the information gained along the way to inform the choice of which point to query next. This is in contrast to the previous assignment [1], where the goal was to build a classifier with high accuracy.

Active learning is a machine learning paradigm that aims to reduce the amount of labeled data required to train a model. In active learning, the model is allowed to query the user for labels of instances that it is uncertain about. This allows the model to focus on the most informative instances, reducing the need for large amounts of labeled data [2].

Rare class discovery is the task of identifying all the classes in a dataset, even if some of the classes are represented by only a few instances. This is particularly challenging when the dataset is imbalanced, with some classes being much rarer than others [3].

The datasets used in this assignment were provided by the instructor. They include the MNIST-C derived dataset and the MNIST-skewed dataset. The MNIST-C dataset is a corrupted version of the original MNIST dataset, while the MNIST-skewed dataset has a skewed distribution of the classes. Importantly, the non-corrupted MNIST dataset used in this assignment has a balanced number of entries for each class.

The requirements for this assignment are as follows [4].

1) Get the provided datasets from D2L. Then for each dataset:
2) Visualize the data w/ labels using 2 or 3-D tSNE.
3) Write your own version of an active learning rare class discovery algorithm.
4) Run your code on the dataset and keep track of the number of classes discovered vs. number of queries.
5) Plot that (# classes discovered vs. # queries).
6) Rerun the same experiment using a random query strategy.
7) Plot the results from the random algorithm on the same plot.

II. Methodology

A. Data Preparation

There were two phases attempted during this assignment to aid in understanding. The first set of algorithms used worked with the following data sets:

- Raw Data
- PCA 2d
- PCA 3d
- PCA 4d
- PCA 5d
- t-SNE 2d
- t-SNE 3D

Many algorithms were run against these data sets and the results are graphed in the appendix.

The second phase of our discovery for this assignment used the following data sets:

- Raw Data
- PCA 10d
- t-SNE 3d
- t-SNE 2d

We will refer the reader to the Jupyter notebook to align the results of the of the first phase to the graphs in the appendix. The training runs took many hours and were reduced to the following algorithms to increase consistency.

B. Active Learning Rare Class Discovery

For the active learning rare class discovery algorithm, we implemented the following query strategies:

- Uncertainty Sampling
- Mahalanobis Uncertainty Sampling
- Random Sampling

These query strategies were stored in the `strategies` list.

Additionally, we used the following classifier models for the active learning algorithm:

- K-Nearest Neighbors with k=10

- Random Forest
- Support Vector Machine

These models were stored in the `models` list.

C. *Visualization*

To gain insights into the structure of the datasets, we visualized the data using 2D and 3D t-SNE projections. The 2D t-SNE plots are shown in Figure 1, and the 3D t-SNE plots are shown in Figure 2. The 3D t-SNE visualization is also provided as a MP4 movie to allow for interactive exploration of the data.

We also generated an animation of the 3D t-SNE visualization to provide a more interactive and dynamic view of the data. This animation allows for a more detailed exploration of the data, revealing potential clusters and outliers that may not be as apparent in static visualizations.

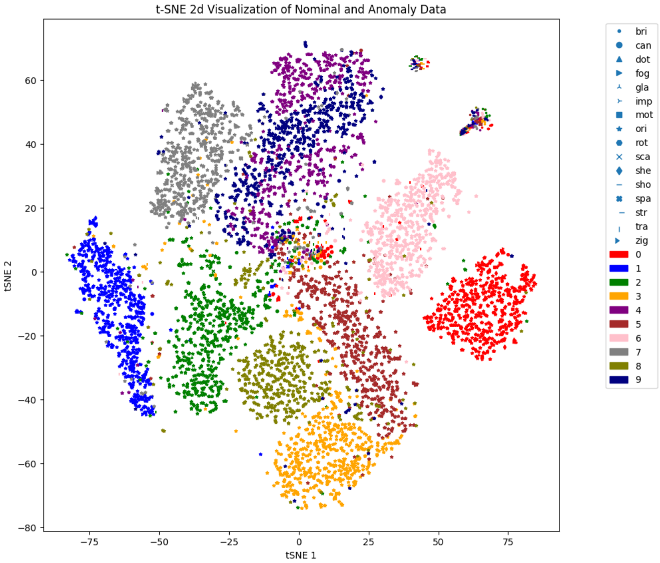

Figure 1. 2D t-SNE Visualization of the Dataset

III. Results and Discussion

A. *Rare Class Discovery Performance*

We ran the active learning rare class discovery algorithm on the provided datasets, tracking the number of classes discovered as a function of the number of queries. Figures 3 to 12 shows the results for both phases of our discovery. The results are consistent with our expectations, showing that the active learning algorithm was able to discover all the classes in the dataset with significantly fewer queries compared to the random query strategy. This demonstrates the effectiveness of the active learning approach in efficiently exploring the data and identifying rare classes. We only ran our experiments to 2000 iterations, but we could have run them for more iterations to see if the active learning algorithm would have discovered all the classes in the dataset with even fewer queries. The total compute time for this assignment exceeded 60 hours of single-core time between two data analysis compute machines.

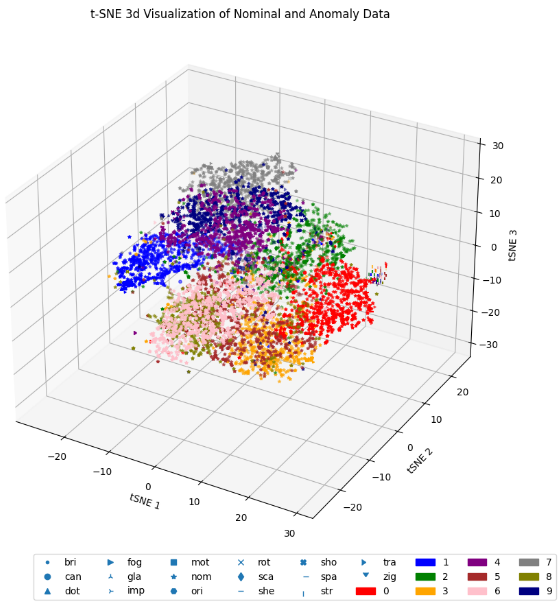

Figure 2. 3D t-SNE Visualization of the Dataset

The results indicate that the active learning algorithm was able to discover all the classes in the dataset with significantly fewer queries compared to the random query strategy. This demonstrates the effectiveness of the active learning approach in efficiently exploring the data and identifying rare classes.

B. *Insights from Data Visualization*

The 2D and 3D t-SNE visualizations provided valuable insights into the structure of the datasets. The 3D t-SNE plot, in particular, allowed for a more nuanced understanding of the data, revealing potential clusters and outliers that were not as apparent in the 2D projection.

The interactive 3D t-SNE movie further enhanced our ability to explore the data and gain a deeper understanding of the relationships between the different classes.

Method	**500**	**1000**	**1500**	**2000**
SVC random	45	83	122	134
RandomForest random	58	86	124	139
KNeighbors random	49	81	128	135
SVC uncertainty	98	133	155	157
RandomForest uncertainty	88	133	150	156
KNeighbors uncertainty	47	69	138	142

Table I
Performance comparison of different methods for Data preparation Raw Data

IV. Conclusion

In this assignment, we implemented an active learning rare class discovery algorithm and evaluated its performance on the provided datasets. The results show that the active learning approach was able to discover all the classes in the dataset

Method	500	1000	1500	2000
SVC random	59	99	122	136
RandomForest random	57	96	129	137
KNeighbors random	53	83	125	134
SVC uncertainty	92	135	158	159
RandomForest uncertainty	107	139	155	158
KNeighbors uncertainty	18	81	127	151
SVC mahalanobis uncertainty	59	91	119	131
RandomForest mahalanobis uncertainty	51	79	122	128
KNeighbors mahalanobis uncertainty	54	88	118	130

Table II

Performance comparison of different methods for Data preparation PCA 10D

Method	500	1000	1500	2000
SVC random	54	80	115	125
RandomForest random	53	86	127	139
KNeighbors random	61	84	125	130
SVC uncertainty	92	135	148	150
RandomForest uncertainty	95	126	150	157
KNeighbors uncertainty	92	124	129	137
SVC mahalanobis uncertainty	58	90	128	139
RandomForest mahalanobis uncertainty	63	82	122	135
KNeighbors mahalanobis uncertainty	56	90	130	135

Table III

Performance comparison of different methods for Data preparation TSNE 3D

with significantly fewer queries compared to a random query strategy.

The data visualization techniques, particularly the 3D t-SNE plot and the interactive movie, provided valuable insights into the structure of the datasets and helped inform the development of the active learning algorithm.

Future work could involve exploring alternative query strategies, investigating the impact of different classifier models, and analyzing the performance of the algorithm on a wider range of datasets with varying levels of class imbalance and rarity.

References

[1] R. Loveland, “Assignment_8.pdf,” From SDSMT D2L Website, 2024.

[2] A. Tharwat and W. Schenck, “A survey on active learning: State-of-the-art, practical challenges and research directions,” *Mathematics*, vol. 11, no. 4, p. 820, 2023. [Online]. Available: https://doi.org/10.3390/math11040820.

[3] D. Zhou and J. He, “Rare category analysis for complex data: A review,” *ACM Computing Surveys*, vol. 56, no. 5, pp. 1–35, 2023.

[4] R. Loveland, “Assignment_9.pdf,” From SDSMT D2L Website, 2024.

Method	500	1000	1500	2000
SVC random	49	84	124	134
RandomForest random	49	81	118	129
KNeighbors random	57	86	119	132
SVC uncertainty	84	136	148	151
RandomForest uncertainty	86	117	152	153
KNeighbors uncertainty	77	116	132	138
SVC mahalanobis uncertainty	51	90	121	132
RandomForest mahalanobis uncertainty	73	96	123	129
KNeighbors mahalanobis uncertainty	60	84	113	130

Table IV

Performance comparison of different methods for Data preparation TSNE 2D

Appendix - Large Images

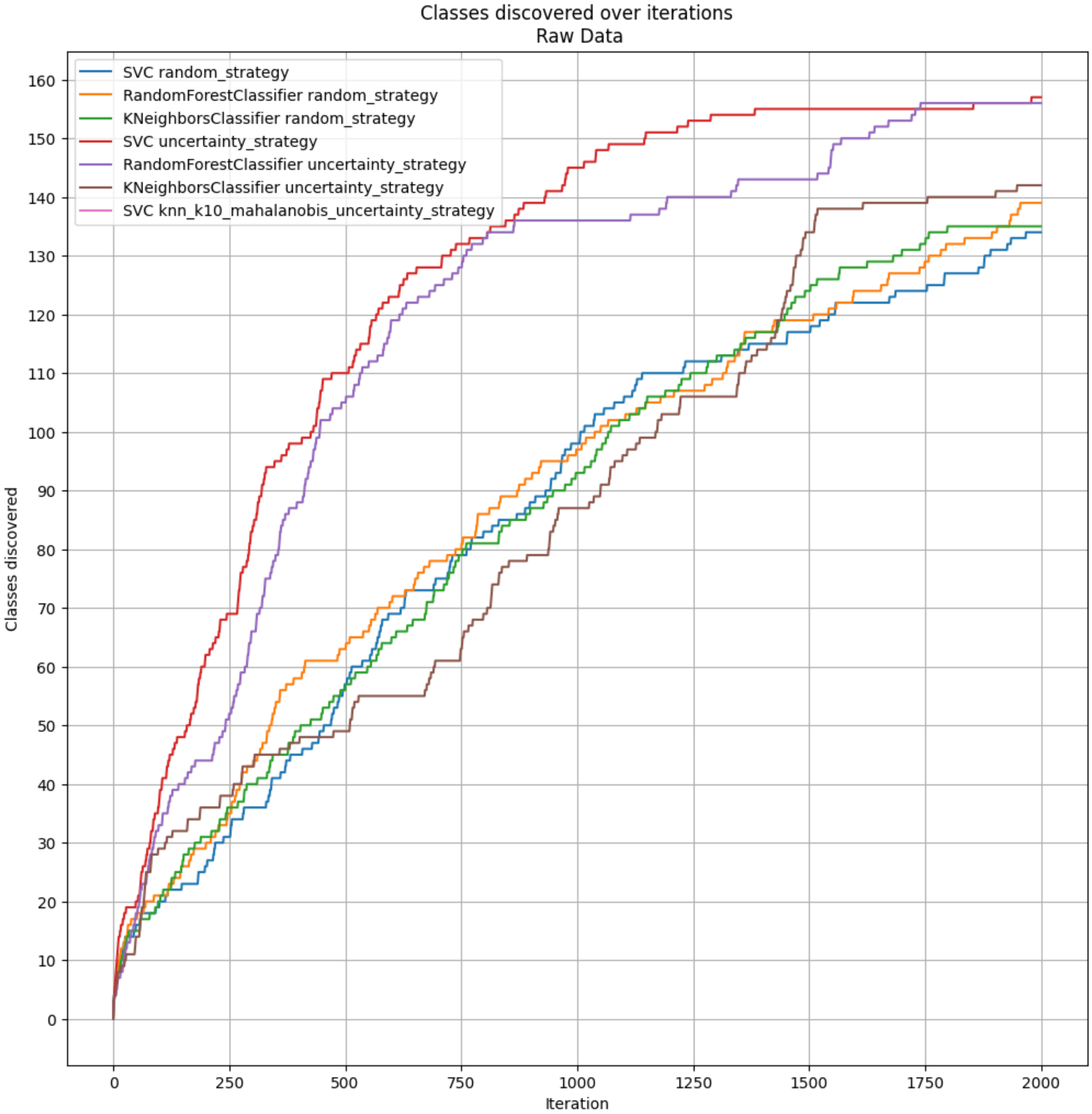

Figure 3. Raw Data Classes Discovered

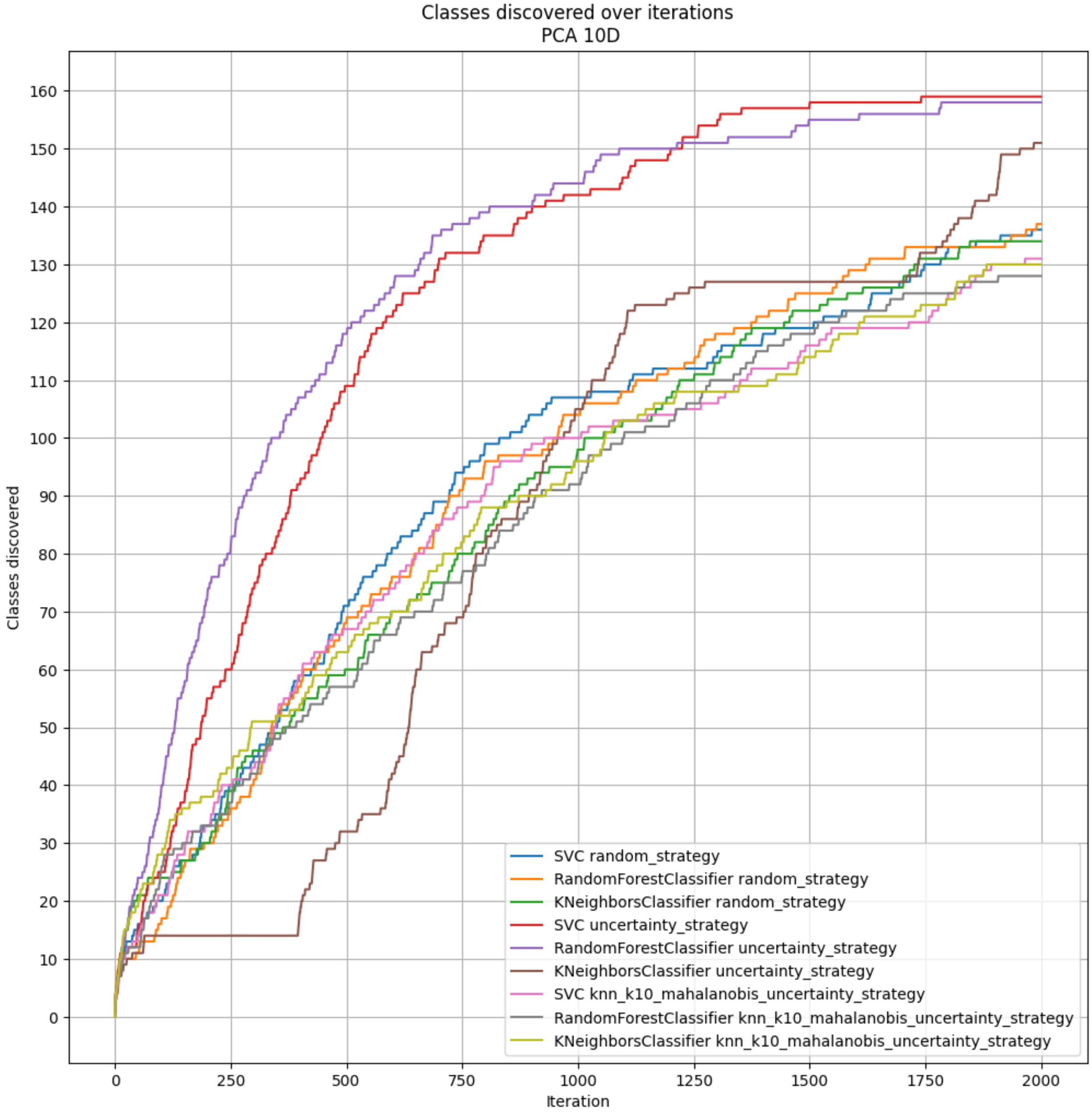

Figure 4. 10D PCA Classes Discovered

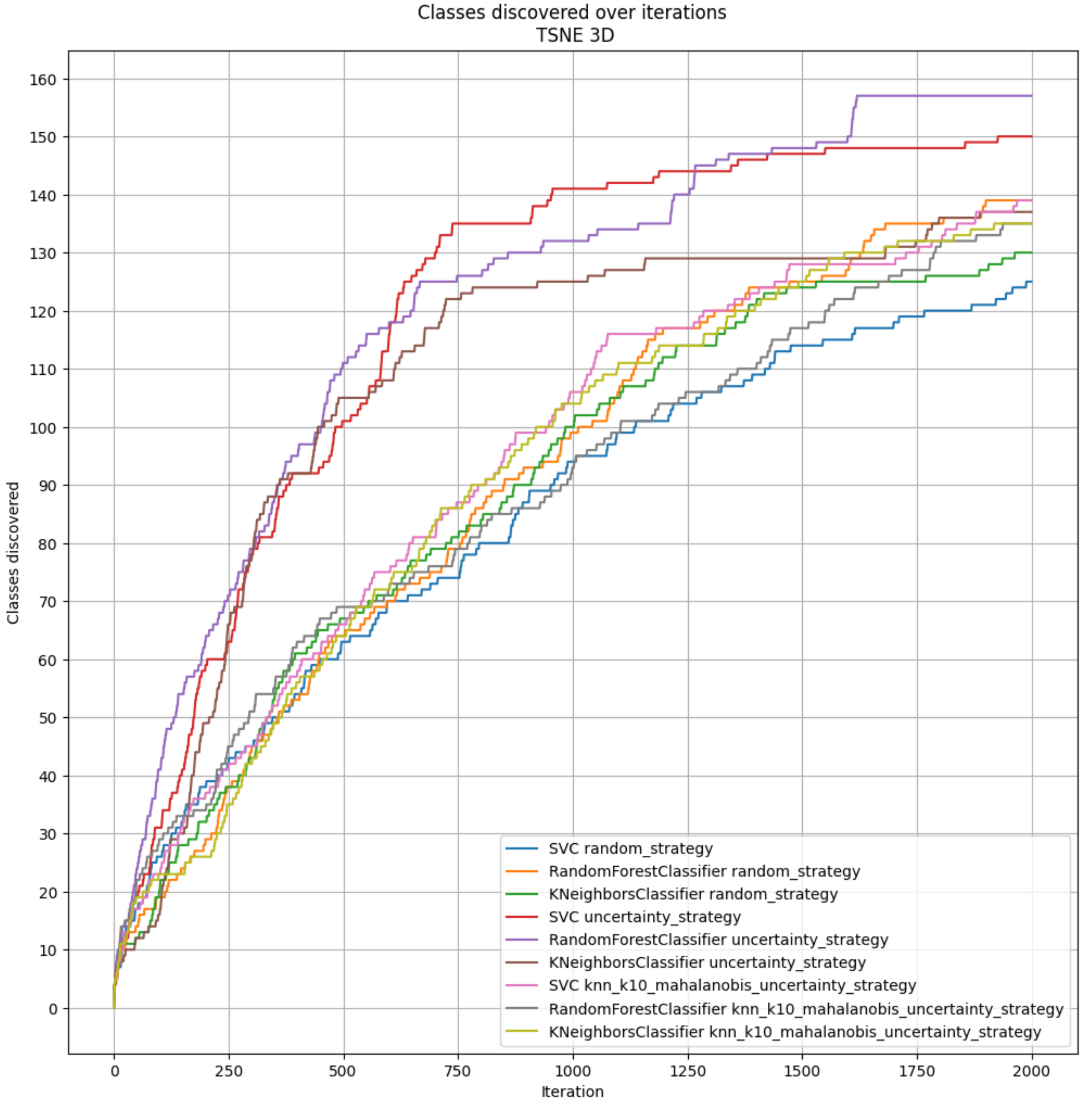

Figure 5. 3D t-SNE Classes Discovered

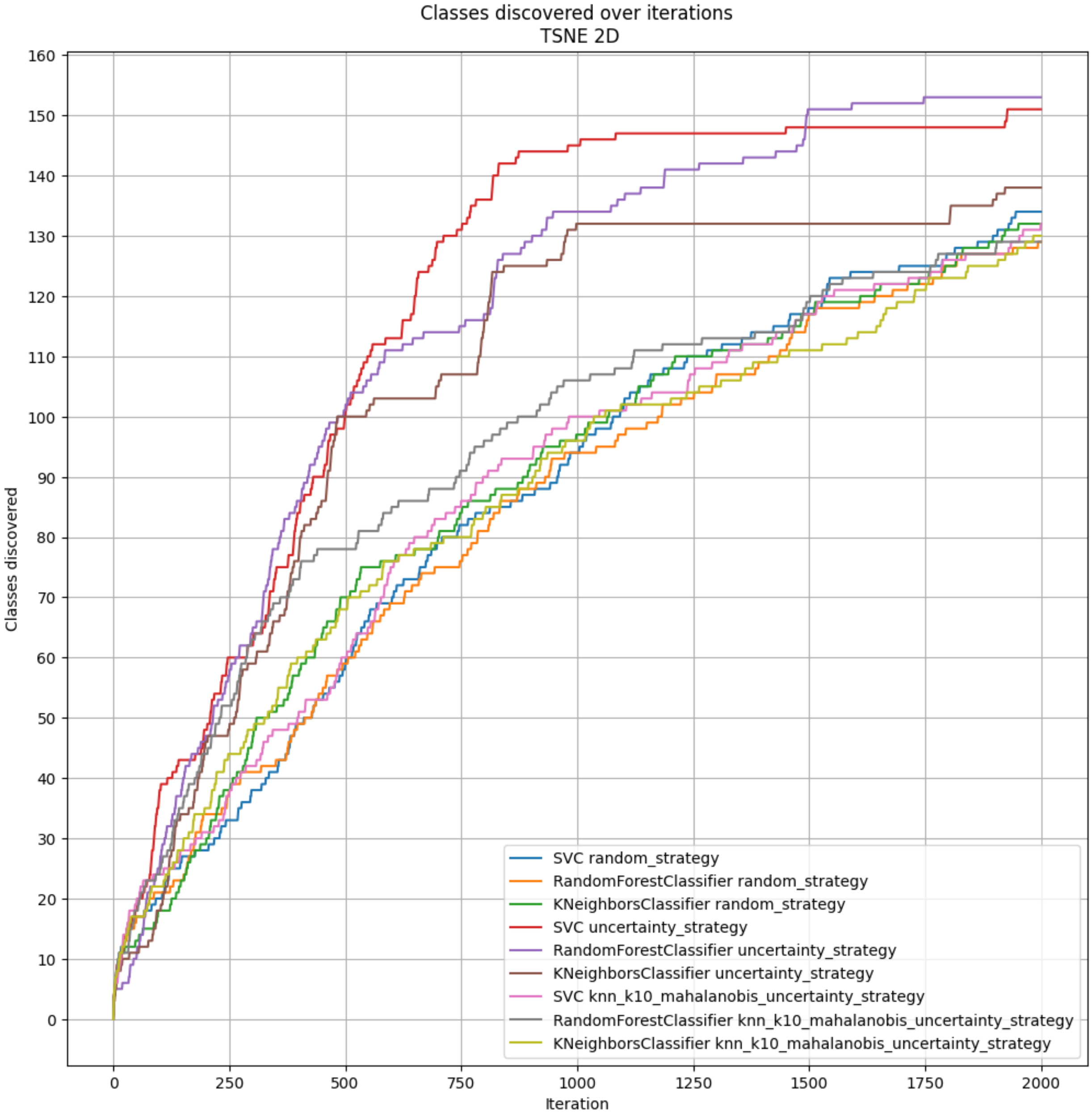

Figure 6. 2D t-SNE Classes Discovered

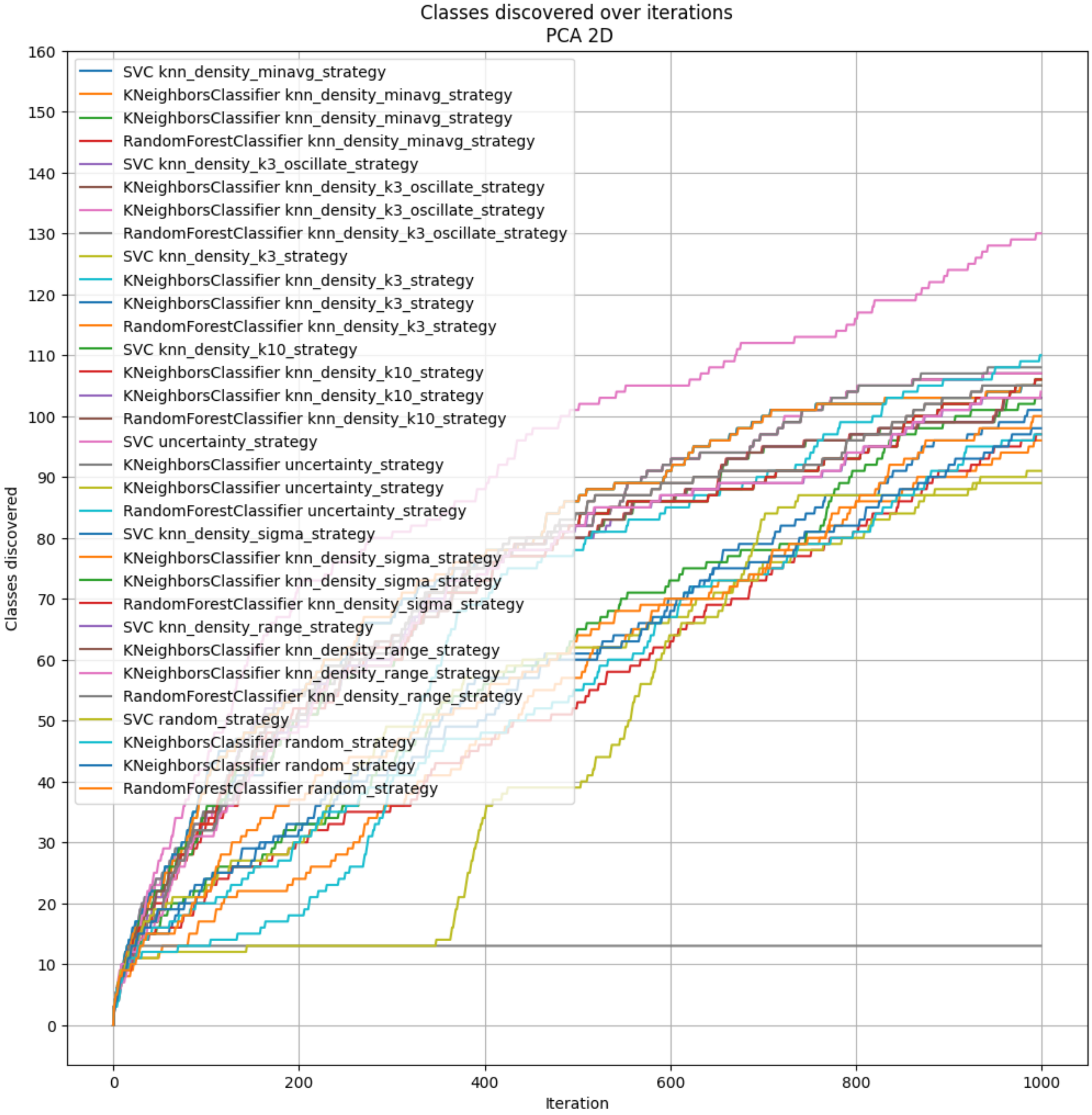

Figure 7. 2D PCA Classes Discovered on original algorithms

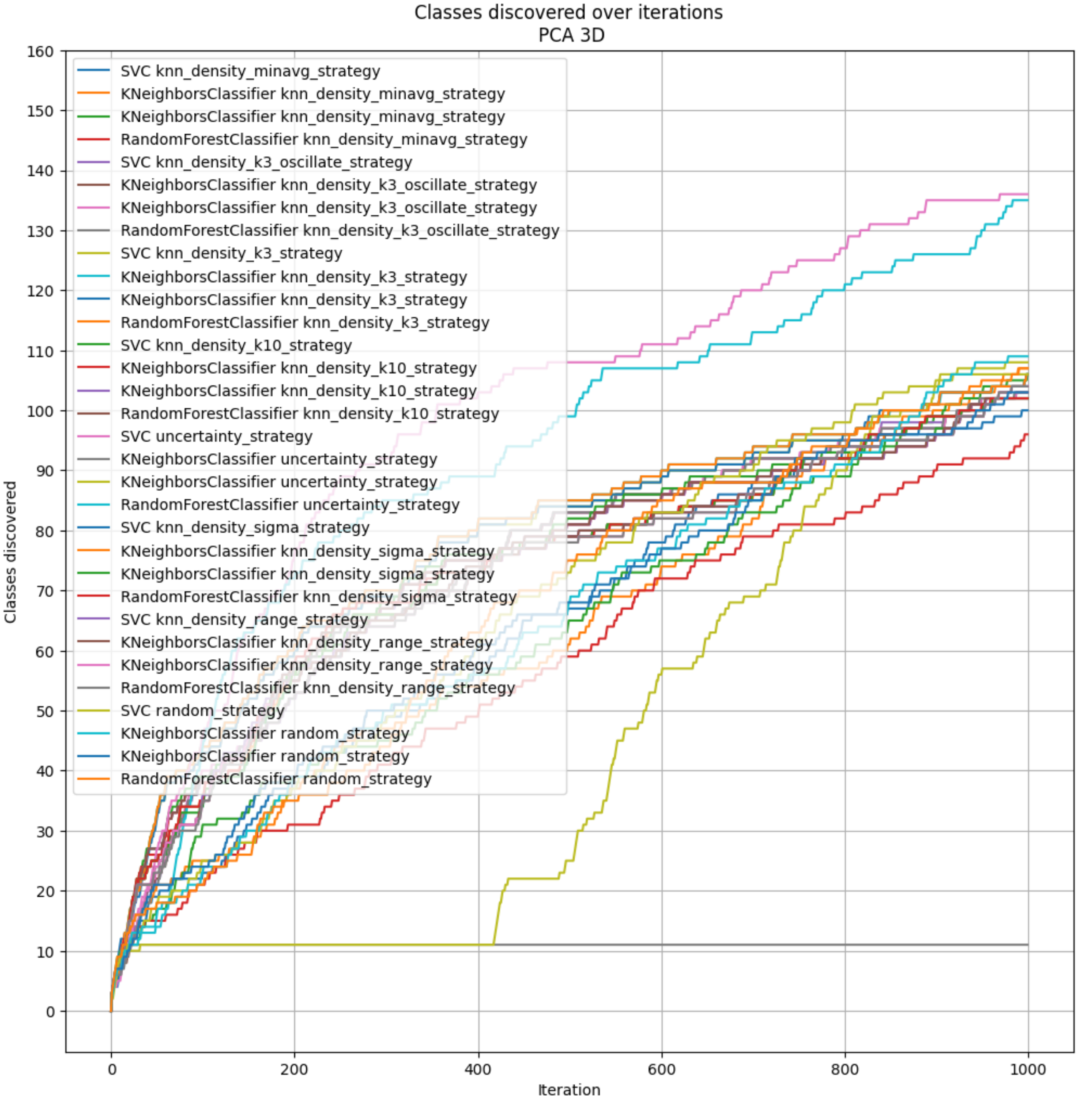

Figure 8. 3D PCA Classes Discovered on original algorithms

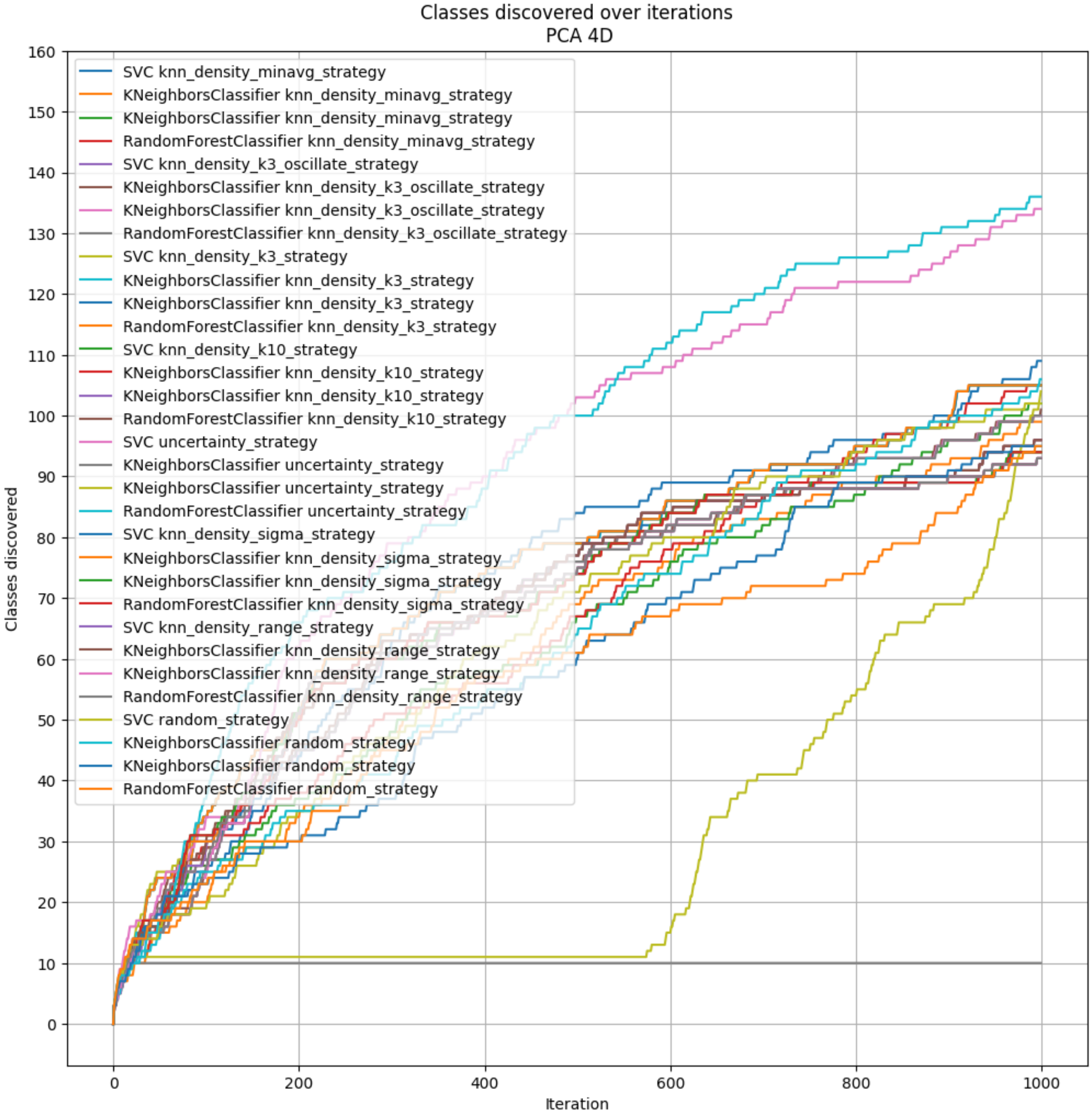

Figure 9. 4D PCA Classes Discovered on original algorithms

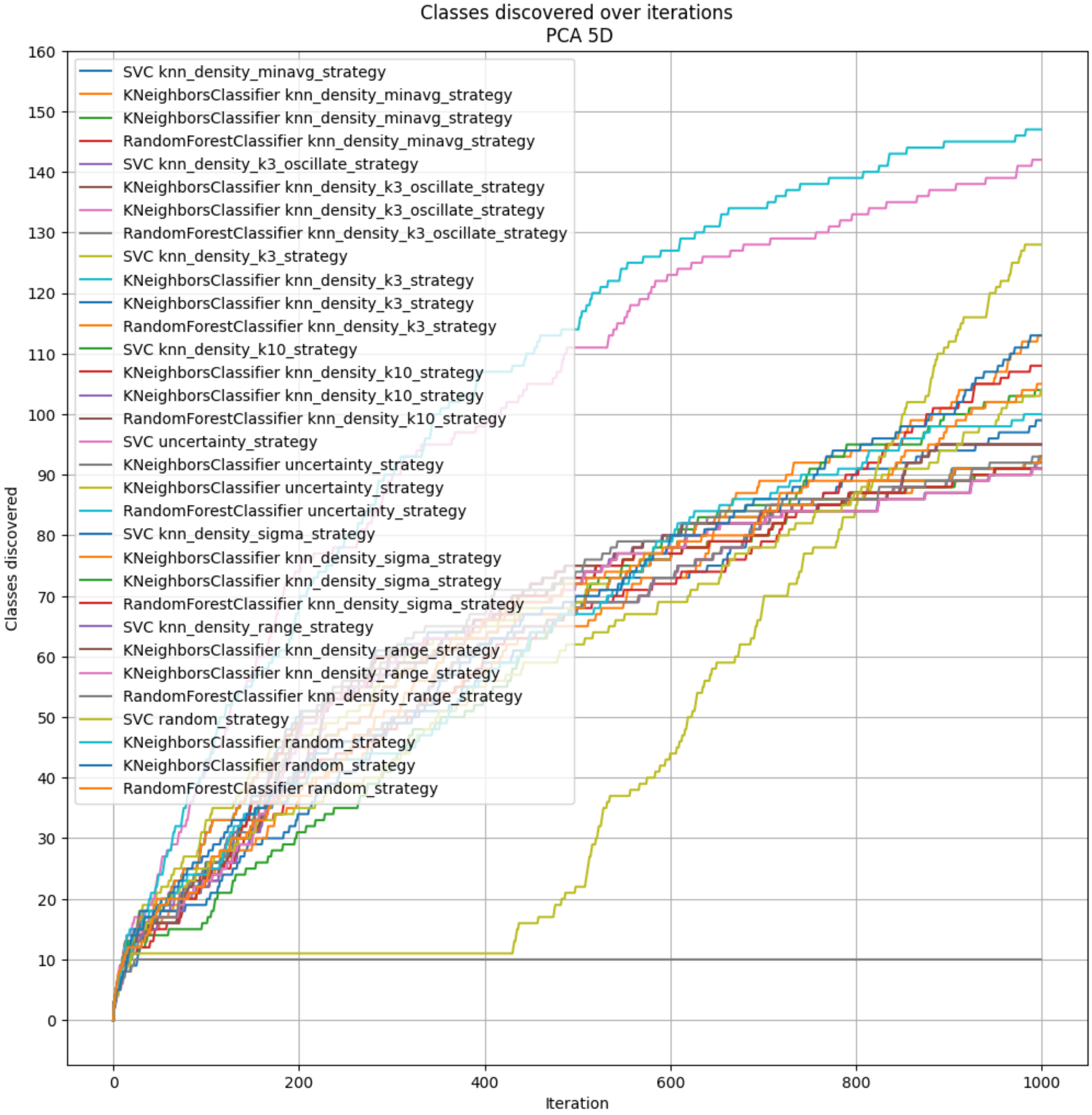

Figure 10. 5D PCA Classes Discovered on original algorithms

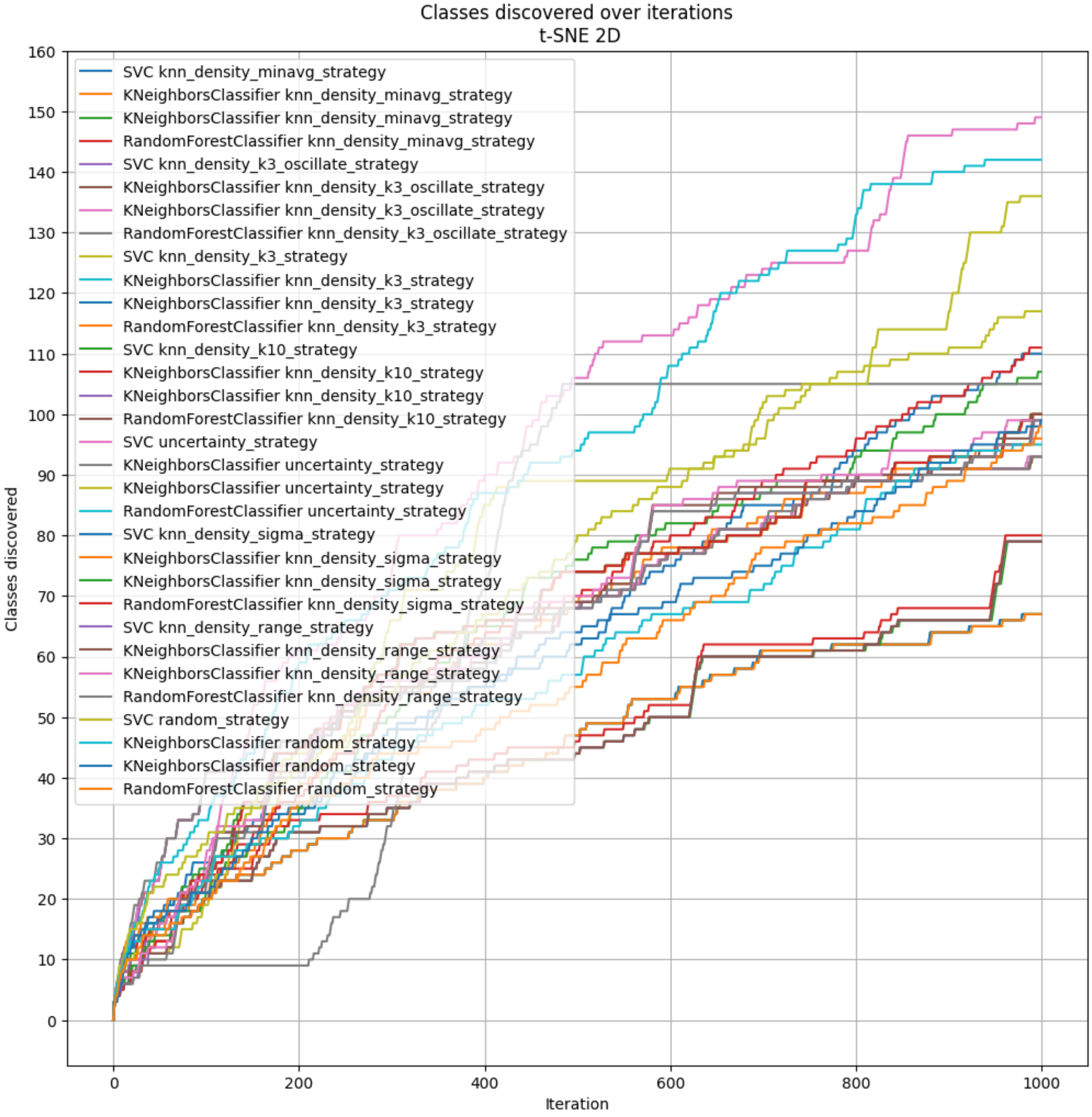

Figure 11. 2D t-SNE Classes Discovered on original algorithms

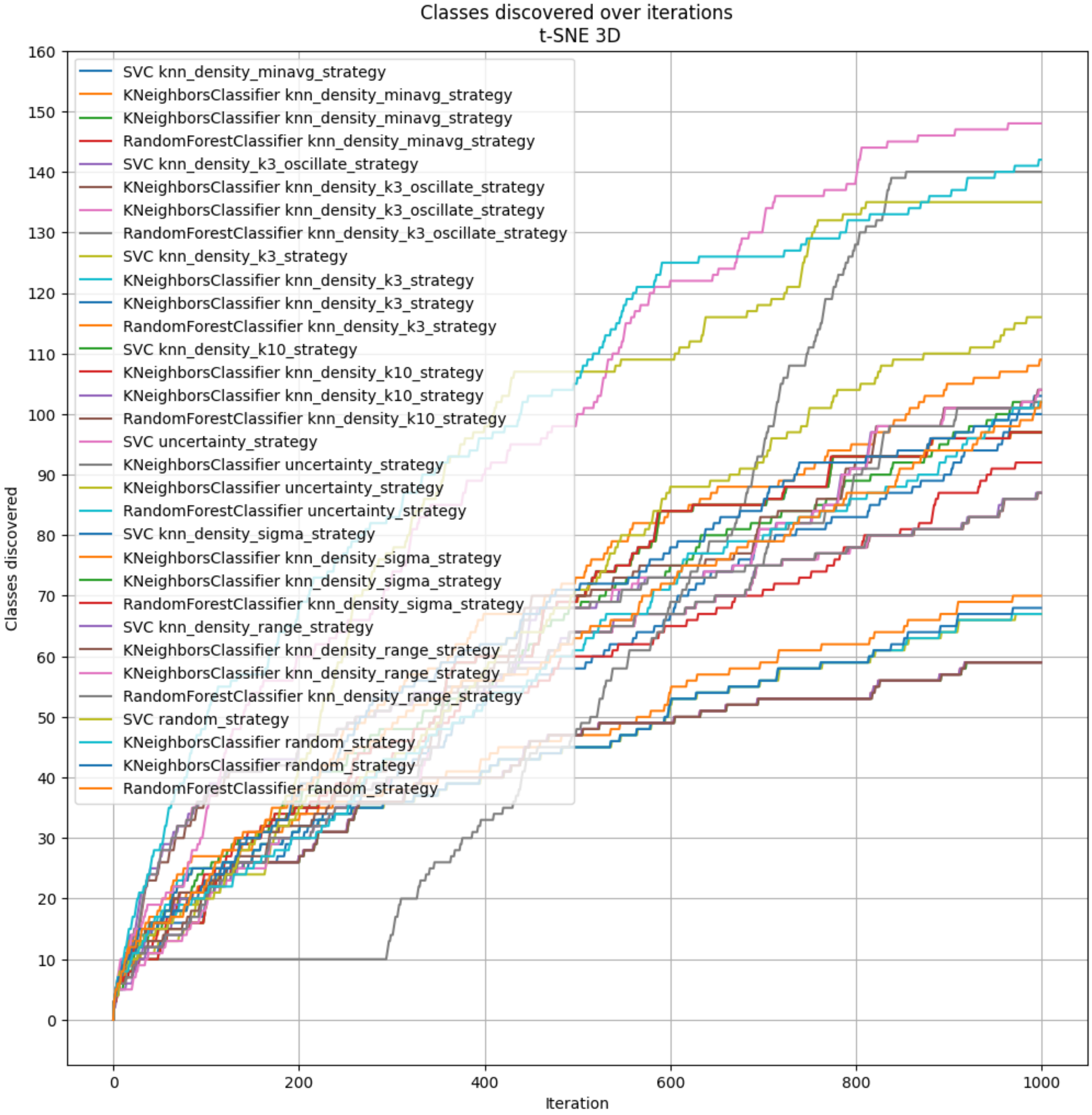

Figure 12. 3D t-SNE Classes Discovered on original algorithms

Python Requirements

```
asttokens==2.4.1
colorama==0.4.6
comm==0.2.2
contourpy==1.2.1
cycler==0.12.1
debugpy==1.8.1
decorator==5.1.1
executing==2.0.1
fonttools==4.51.0
graphviz==0.20.3
ipykernel==6.29.4
ipython==8.23.0
jedi==0.19.1
joblib==1.4.0
jupyter_client==8.6.1
jupyter_core==5.7.2
kiwisolver==1.4.5
matplotlib==3.8.4
matplotlib-inline==0.1.7
nest-asyncio==1.6.0
numpy==1.26.4
packaging==24.0
pandas==2.2.2
parso==0.8.4
pillow==10.3.0
platformdirs==4.2.0
prompt-toolkit==3.0.43
psutil==5.9.8
pure-eval==0.2.2
Pygments==2.17.2
pyparsing==3.1.2
python-dateutil==2.9.0.post0
pytz==2024.1
pywin32==306
pyzmq==26.0.0
scikit-learn==1.4.2
scipy==1.13.0
six==1.16.0
stack-data==0.6.3
threadpoolctl==3.4.0
tornado==6.4
traitlets==5.14.2
tzdata==2024.1
wcwidth==0.2.13
```